Web Development Using PHP

DECAP777

Edited by
Dr. Pawan Kumar

LOVELY
PROFESSIONAL
UNIVERSITY

LOVELY
PROFESSIONAL
UNIVERSITY

Web Development Using PHP

Edited By:
Dr. Pawan Kumar

Title: WEB DEVELOPMENT USING PHP

Author's Name: Dr. Tarandeep Kaur

Published By : Lovely Professional University

Publisher Address: Lovely Professional University, Jalandhar Delhi GT road, Phagwara - 144411

Printer Detail: Lovely Professional University

Edition Detail: (I)

ISBN: 978-81-19334-31-5

CONTENT

Unit 01: Introduction with Language Basics

CONTENTS

Objectives

After studying this unit, you will be able to:

- Know about the features and history of PHP
- Investigate the data types, variables and constants in PHP.
- Explore the lexical structure of PHP.
- Understand the concept of operators in PHP.
- Recognize the importance of commenting in PHP.
- Empirically run PHP on your system.

Introduction

PHP is a simple yet powerful language designed for creating HTML content.PHP is a server scripting language, and a powerful tool for making dynamic and interactive Web pages.PHP is a widely used, free, and efficient alternative to competitors such as Microsoft's ASP.It is an open-source, interpreted, and object-oriented scripting language that can be executed at the server-side. It is well suited for web development and is used to develop web applications (an application that executes on the server and generates the dynamic page.).

PHP was created by Rasmus Lerdorf in 1994 but appeared in the market in 1995. PHP 8.1.2 is the latest version of PHP, which was released on 20th January; 2022.W3Techs reports that, as of January 2022, "PHP is used by 78.1% of all the websites whose server-side programming language we know". PHP version 7.4 is the most used version. Support for version 7.3 was dropped on 6 December 2021.

PHP Mascot

The mascot of the PHP project is the elePHPant (Figure 1), a blue elephant with the PHP logo on its side, designed by Vincent Pointer in 1998."The (PHP) letters were forming the shape of an elephant if viewed in a sideways angle". The elePHPant is sometimes differently colored when in plush toy form.Many variations of this mascot have been made over the years. Only the elePHPants based on the original design by Vincent Pointer are considered official by the community.

Figure 1: PHP Mascot

1.1 What Does PHP Do?

PHP can be used in three primary ways:

- Server-side Scripting: PHP was originally designed to create dynamic web content, and it is still best suited for thattask. To generate HTML, you need the PHP parser and a web server through which to send thecoded documents. PHP has also become popular for generating XML documents, graphics, Flashanimations, PDF files, and so much more.
- Command-line Scripting: PHP can run scripts from the command line, much like Perl, awk, or the Unix shell. You mightuse the command-line scripts for system administration tasks, such as backup and log parsing;even some CRON job type scripts can be done this way (nonvisual PHP tasks).
- Graphical Scripting: Using PHP-GTK, you can write full-blown, cross-platform GUI applications in PHP. PHP runs on all major operating systems, from Unix variants including Linux, FreeBSD, Ubuntu,Debian, and Solaris to Windows and Mac OS X. It can be used with all leading web servers, including Apache, Microsoft IIS, and the Netscape/iPlanet servers.

Notes: The language itself is extremely flexible.

1.2 Characteristics of PHP

PHP stands for Hypertext Preprocessor. It originally stood for Personal Home Page, but it now stands for the recursive initialism PHP: Hypertext Preprocessor.PHP is an interpreted language, i.e., there is no need for compilation. The other characteristics of PHP are:

- Faster than other scripting languages, for example, ASP and JSP.
- A server-side scripting language, which is used to manage the dynamic content of the website.

- Can be embedded into HTML.
- An object-oriented language.
- Simple and easy to learn language.
- PHP files have extension ".php“.
- PHP is a server-side scripting language, which is used to design the dynamic web applications with MySQL database.
- Handles dynamic content, database as well as session tracking for the website.
- Can create sessions in PHP.
- Can access cookies variable and also set cookies.
- Helps to encrypt the data and apply validation.
- Supports several protocols such as HTTP, POP3, SNMP, LDAP, IMAP, and many more.
- Using PHP language, you can control the user to access some pages of your website.
- As PHP is easy to install and set up, this is the main reason why PHP is the best language to learn.
- PHP can handle the forms, such as - collect the data from users using forms, save it into the database, and return useful information to the user. For example: Registration form.

PHP is a very popular language because of its simplicity and open source. There are many important features of PHP as listed below (Figure 2):

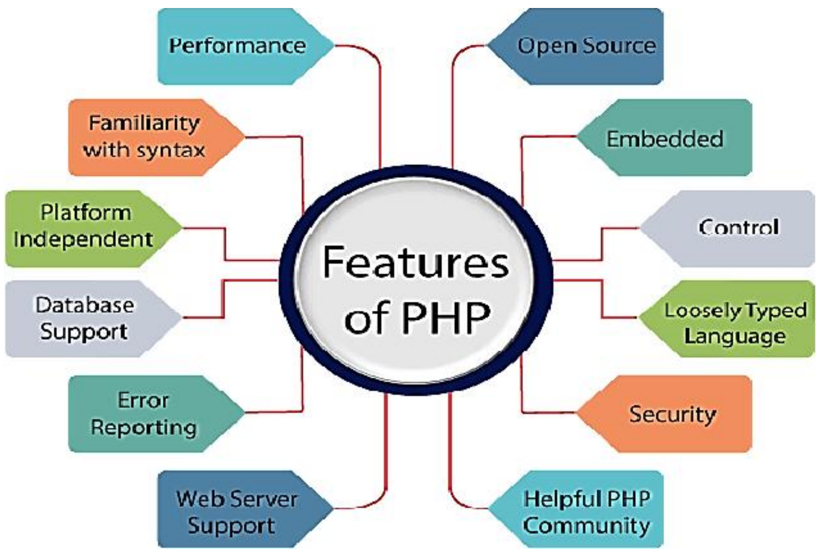


Figure 2:Characteristics of PHP

Performance:PHP script is executed much faster than those scripts which are written in other languages such as JSP and ASP. PHP uses its own memory, so the server workload and loading time is automatically reduced, which results in faster processing speed and better performance.

Open Source: PHP source code and software are freely available on the web. You can develop all the versions of PHP according to your requirement without paying any cost. All its components are free to download and use.

Familiarity with syntax: PHP has easily understandable syntax. Programmers are comfortable coding with it.

Embedded: PHP code can be easily embedded within HTML tags and script.

Platform Independent: PHP is available for WINDOWS, MAC, and LINUX& UNIX operating system. A PHP application developed in one OS can be easily executed in other OS also.

Control: Different programming languages require long script or code, whereas PHP can do the same work in a few lines of code. It has maximum control over the websites like you can make changes easily whenever you want.

Database Support: PHP supports all the leading databases such as MySQL, SQLite, ODBC, etc.

Loosely Typed Language: PHP allows us to use a variable without declaring its datatype. It will be taken automatically at the time of execution based on the type of data it contains on its value.

Error Reporting: PHP has predefined error reporting constants to generate an error notice or warning at runtime. E.g., E_ERROR, E_WARNING, E_STRICT, E_PARSE.

Security: PHP is a secure language to develop the website. It consists of multiple layers of security to prevent threads and malicious attacks.

Web servers Support:PHP is compatible with almost all local servers used today like Apache, Netscape, Microsoft IIS, etc.

Helpful PHP Community: It has a large community of developers who regularly updates documentation, tutorials, online help, and FAQs. Learning PHP from the communities is one of the significant benefits.

1.3 PHP Web Execution

PHP code is usually processed on a web server by a PHP interpreter implemented as a module, a daemon or as a Common Gateway Interface (CGI) executable (Figure 3). On a web server, the result of the interpreted and executed PHP code – which may be any type of data, such as generated HTML or binary image data– would form the whole or part of an HTTP response.

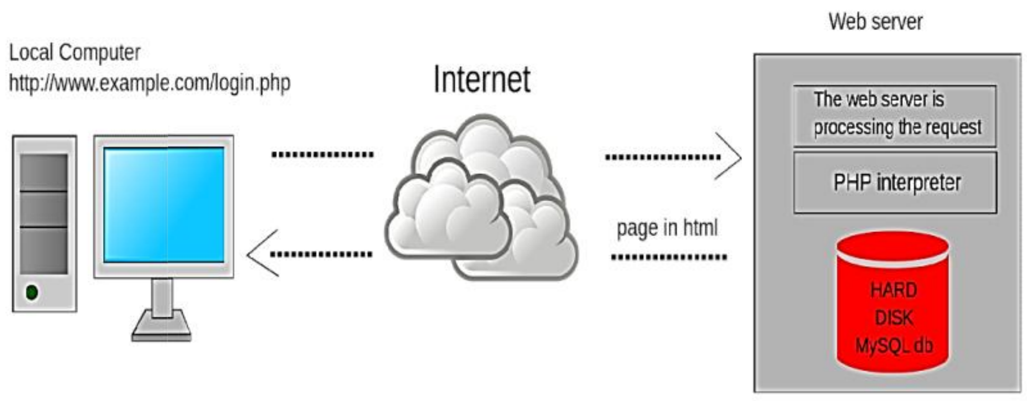


Figure 3: PHP Web Execution Scenario

Various web template systems, web content management systems, and web frameworks exist which can be employed to orchestrate or facilitate the generation of that response. Additionally, PHP can be used for many programming tasks outside the web context, such as standalone graphical applications and robotic drone control. The PHP code can also be directly executed from the command line.PHP is widely used in web development nowadays. PHP can develop dynamic websites easily. But you must have the basic the knowledge of following technologies for web development as well:

- HTML-HTML is used to design static webpage.
- CSS-CSS helps to make the webpage content more effective and attractive
- JavaScript- JavaScript is used to design an interactive website.
- Ajax
- XML and JSON
- jQuery

1.4 History of PHP

PHP development began in 1994 when Rasmus Lerdorf wrote several Common Gateway Interface (CGI) programs in C, which he used to maintain his personal homepage. He extended them to work with web forms and to communicate with databases, and called this implementation "Personal Home Page/Forms Interpreter" or PHP/FI.PHP/FI could be used to build simple, dynamic web applications. To accelerate bug reporting and improve the code, Lerdorf initially announced the release of PHP/FI as "Personal Home Page Tools (PHP Tools) version 1.0" on the Usenet discussion group on June 8, 1995.This release already had the basic functionality that PHP has today. This included Perl-like variables, form handling, and the ability to embed HTML. The syntax resembled that of Perl, but was simpler, more limited and less consistent.

The early PHP was not intended to be a new programming language, and grew organically, with Lerdorf noting in retrospect: "I don't know how to stop it, there was never any intent to write a programming language [...] I have absolutely no idea how to write a programming language, I just kept adding the next logical step on the way." A development team began to form and, after months of work and beta testing, officially released PHP/FI 2 in November 1997.The fact that PHP was not originally designed, but instead was developed organically has led to inconsistent naming of functions and inconsistent ordering of their parameters. In some cases, the function names were chosen to match the lower-level libraries which PHP was "wrapping", while in some very early versions of PHP the length of the function names was used internally as a hash function, so names were chosen to improve the distribution of hash values.

1.5 Versions of PHP

The different versions of PHP have been developed as listed below:

PHP 3 and 4: This is an example of PHP code for the WordPress content management system.ZeevSuraski and Andi Gutmans rewrote the parser in 1997 and formed the base of PHP 3, changing the language's name to the recursive acronym PHP: Hypertext Pre-processor.Afterwards, the public testing of PHP 3 began, and the official launch came in June 1998. Suraski and Gutmans then started a new rewrite of PHP's core, producing the Zend Engine in 1999. They also founded Zend Technologies in Ramat Gan, Israel.On 22 May 2000, PHP 4, powered by the Zend Engine 1.0, was released. By August 2008, this branch had reached version 4.4.9. PHP 4 is now no longer under development and nor are any security updates planned to be released.

PHP 5: On 1 July 2004, PHP 5 was released, powered by the new Zend Engine II. PHP 5 included new features such as improved support for object-oriented programming, the PHP Data Objects (PDO) extension (which defines a lightweight and consistent interface for accessing databases), and numerous performance enhancements.In 2008, PHP 5 became the only stable version under development. Later, static binding had been missing from previous versions of PHP, and was added in version 5.3.Many high-profile open-source projects ceased to support PHP 4 in new code from February 5, 2008, because of the GoPHP5 initiative, provided by a consortium of PHP developers promoting the transition from PHP 4 to PHP 5.Over time, PHP interpreters became available on most existing 32-bit and 64-bit operating systems, either by building them from the PHP source code, or by using pre-built binaries. For PHP versions 5.3 and 5.4, the only available Microsoft Windows binary distributions were 32-bit IA-32 builds, requiring Windows 32-bit compatibility mode while using Internet Information Services (IIS) on a 64-bit Windows platform. PHP version 5.5 made the 64-bit x86-64 builds available for Microsoft Windows. The official security support for PHP 5.6 ended on 31 December 2018.

PHP 6 and Unicode:PHP received mixed reviews due to lacking native Unicode support at the core language level.In 2005, a project headed by Andrei Zmievski was initiated to bring native Unicode support throughout PHP, by embedding the International Components for Unicode (ICU) library, and representing text strings as UTF-16 internally.Since this would cause major changes both to the internals of the language and to user code, it was planned to release this as version 6.0 of the language, along with other major features then in development.However, a shortage of developers who understood the necessary changes, and performance problems arising from conversion to and from UTF-16, which is rarely used in a web context, led to delays in the project.

As a result, a PHP 5.3 release was created in 2009, with many non-Unicode features back-ported from PHP 6, notably namespaces. In March 2010, the project in its current form was officially abandoned, and a PHP 5.4 release was prepared containing most remaining non-Unicode features from PHP 6, such as traits and closure re-binding. The initial hopes were that a new plan would be formed for Unicode integration, but by 2014 none had been adopted.

PHP 7: During 2014 and 2015, a new major PHP version was developed, PHP 7. The numbering of this version involved some debate among internal developers. While the PHP 6 Unicode experiment had never been released, several articles and book titles referenced the PHP 6 name, which might have caused confusion if a new release were to reuse the name. After a vote, the name PHP 7 was chosen.

The foundation of PHP 7 is a PHP branch that was originally dubbed PHP next generation (phpng). It was authored by Dmitry Stogov, Xinchen Hui and Nikita Popov, and aimed to optimize PHP performance by refactoring the Zend Engine while retaining near-complete language compatibility.

By 14 July 2014, WordPress-based benchmarks, which served as the main benchmark suite for the phpng project, showed an almost 100% increase in performance. The changes from phpng make it easier to improve performance in future versions, as more compact data structures and other changes are seen as better suited for a successful migration to a just-in-time (JIT) compiler. Because of the significant changes, the reworked Zend Engine was called Zend Engine 3, succeeding Zend Engine 2 used in PHP 5.

The major versions of PHP are allowed to break backward-compatibility of code and therefore PHP 7 presented an opportunity for other improvements beyond phpng that require backward-compatibility breaks.In particular, it involved the following changes:

- Many fatal or recoverable-level legacy PHP error mechanisms were replaced with modern object-oriented exceptions.
- The syntax for variable dereferencing was reworked to be internally more consistent and complete, allowing the use of the operators ->, [], (),{}, and ::, with arbitrary meaningful left-side expressions.
- Support for legacy PHP 4-style constructor methods was deprecated.
- The behaviour of the foreach statement was changed to be more predictable.
- Constructors for the few classes built-in to PHP which returned null upon failure were changed to throw an exception instead, for consistency.
- Several unmaintained or deprecated server application programming interfaces (SAPIs) and extensions were removed from the PHP core, most notably the legacy mysql extension.
- The behaviour of the list() operator was changed to remove support for strings.
- Support was removed for legacy ASP-style delimiters <% and %> and <script language="php"> ... </script>.
- An oversight allowing a switch statement to have multiple default clauses was fixed.
- Support for hexadecimal number support in some implicit conversions from strings to number types was removed.
- The left-shift and right-shift operators were changed to behave more consistently across platforms.
- Conversions between floating-point numbers and integers were changed (e.g. infinity changed to convert to zero) and implemented more consistently across platforms.
- PHP 7 also included new language features. Most notably, it introduced return type declarations for functions which complement the existing parameter type declarations, and support for the scalar types (integer, float, string, and Boolean) in parameter and return type declarations.

PHP 8:PHP 8 was released on November 26, 2020. PHP 8 is a major version and has breaking changes from previous versions. The new features and notable changes include:

- Just-in-time compilation
- Addition of the match expression
- Type changes and additions

- Syntax changes and additions
- Standard library changes and additions

PHP 8.1: PHP 8.1 was released on November 25, 2021. It included several improvements, such as enumerations (also called "enums"), readonly properties and array unpacking with string keys.

- Support for `readonly` properties was added.
- After the introduction of array unpacking in PHP 7.4 with consecutive numbered keys, PHP 8.1 introduced support for array unpacking with string keys.
- PHP 8.1 added support for using `new` in initializes.
- A new syntax was added for creating callables.
- PHP 8.1 brought support for Pure Intersection Types, after in the introduction of union types in PHP 8.0.

1.6 Applications of PHP

PHP is one of the most widely used language over the web.PHP performs system functions, that is, from files on a system it can create, open, read, write, and close them.PHP can handle forms, that is, gather data from files, save data to a file, through email you can send data, return data to the user.You add, delete, modify elements within your database through PHP. It allows to access cookies variables and set cookies.Using PHP, you can restrict users to access some pages of your website. Also, it can encrypt the data (Figure 4).

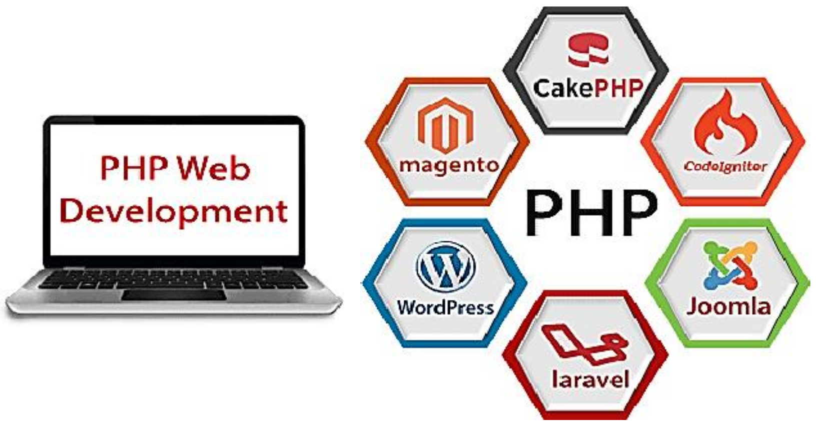


Figure 4: PHP Based Applications

CakePHP: CakePHP is an open-source web framework. It is intended to make developing, deploying and maintaining applications much easier. It follows the model–view–controller (MVC) approach and is written in PHP, modeled after the concepts of Ruby on Rails, and distributed under the MIT License. CakePHP uses well-known software engineering concepts and software design patterns, such as convention over configuration, model–view–controller, active record, association data mapping, and front controller. CakePHP is licensed under the MIT license which makes it perfect for use in commercialapplications.

CodeIgniter: It is intended to make developing, deploying and maintaining applications much easier. CodeIgniter is loosely based on the popular model–view–controller (MVC) development pattern. While controller classes are a necessary part of development under CodeIgniter, models and views are optional. CodeIgniter can be also modified to use Hierarchical Model View Controller (HMVC) which allows the developers to maintain modular grouping of Controller, Models and View arranged in a sub-directory format.

Joomla: Joomla also spelled Joomla! (With an exclamation mark) and sometimes abbreviated as J! is a free and open-source content management system (CMS) for publishing web content on websites. Web content applications include discussion forums, photo galleries, e-Commerce and user communities and numerous other web-based applications. Joomla is developed by a community of volunteers supported with the legal, organizational and financial resources of Open-Source Matters, Inc.

Joomla is written in PHP, uses object-oriented programming techniques and software design patterns, and stores data in a MySQL database. It has a software dependency on the Symfony PHP framework. Joomla includes features such as page caching, RSS feeds, blogs, search, and support for language internationalization. It is built on a model–view–controller web application framework that can be used independently of the CMS.

Laravel: Laravel is a powerful MVC PHP framework, designed for developers who need a simple and elegant toolkit to create full-featured web applications. Laravel was created by Taylor Otwell. Laravel is a web application framework with expressive, elegant syntax.

WordPress: WordPress (WP, WordPress.org) is a free and open-source content management system (CMS) written in PHP and paired with a MySQL or MariaDB database. The features include plugin architecture and a template system, referred to within WordPress as Themes. WordPress was originally created as a blog-publishing system but has evolved to support other web content types including more traditional mailing lists and forums, media galleries, membership sites, learning management systems (LMS) and online stores. One of the most popular content management system solutions in use, WordPress is used by 42.8% of the top 10 million websites as of October 2021.

WordPress was released on May 27, 2003, by its founders, American developer Matt Mullenweg and English developer Mike Little, as a fork of b2/cafelog. The software is released under the GPLv2 (or later) license. In order to function, WordPress has to be installed on a web server, either part of an Internet hosting service like WordPress.com or a computer running the software package WordPress.org in order to serve as a network host in its own right.[10] A local computer may be used for single-user testing and learning purposes.

Magento: Magento 2 is a CMS (Content Management System), developed by Varien Inc. It is an open-source software which is a very useful software for online business. Magento 2 is developed in PHP and Zend framework. Magento 2 is currently the largest E-commerce platform in the world. Magento 2 is known for easy customization and extension of its functionalities.

1.7 PHP Environment Setup

In order to develop and run PHP Web pages three vital components need to be installed on your computer system.

- Web Server: PHP will work with virtually all Web Server software, including Microsoft's Internet Information Server (IIS) but then most often used is freely available Apache Server. Download Apache for free here − https://httpd.apache.org/download.cgi
- Database: PHP will work with virtually all database software, including Oracle and Sybase but most commonly used is freely available MySQL database. Download MySQL for free here− https://www.mysql.com/downloads/
- PHP Parser: In order to process PHP script instructions a parser must be installed to generate HTML output that can be sent to the Web browser.

PHP Parser Installation: Before you proceed it is important to make sure that you have proper environment setup on your machine to develop your web programs using PHP.

- Type the following address into your browser's address box.

  http://127.0.0.1/info.php

- If this displays a page showing your PHP installation related information, then it means you have PHP and Web server installed properly. Otherwise, you must follow given procedure to install PHP on your computer.
- You can install and configure PHP over the following four platforms:
  - PHP Installation on Linux or Unix with Apache
  - PHP Installation on Mac OS X with Apache

- PHP Installation on Windows NT/2000/XP with IIS
- PHP Installation on Windows NT/2000/XP with Apache

1.8 Lexical Structures in PHP

Lexical structure of a programming language is the set of basic rules that governs how you write programs in that language. The lowest-level syntax of the language and specifies such things as what variable names look like, what characters are used for comments, and how program statements are separated from each other.

Basic PHP Syntax

A PHP script can be placed anywhere in the document. A PHP script starts with <?php and ends with ?>:

```
<? php
// PHP code goes here
?>
```

The default file extension for PHP files is ".php". A PHP file normally contains HTML tags, and some PHP scripting code.

We have an example of a simple PHP file, with a PHP script that uses a built-in PHP function "echo" to output the text "Hello World!" on a web page:

```
<! DOCTYPE html>
<html>
<body>
<h1>My first PHP page</h1>
<?php
echo "Hello World!";
?>
</body>
</html>
```

Notes:PHP statements end with a semicolon (;).

Expressions are combinations of tokens: The smallest building blocks of PHP are the indivisible tokens, such as numbers (3.14159), strings (.two.), variables ($two), constants (TRUE), and the special words that make up the syntax of PHP itself like if, else, while, for and so forth.

PHP Semicolon

The PHP statements end with a semicolon (;). A semicolon is used to mark end of a statement and is mandatory.

```
$a = 34;
$b = $a * 34 - 34;
echo $a;
```

Here, we have three different PHP statements:

- The first is an assignment. It puts a value into the $a variable.
- The second one is an expression. The expression is evaluated and the output is given to the $b variable.
- The third one is a command. It prints the $a variable.

Escaping to PHP

The PHP parsing engine needs a way to differentiate PHP code from other elements in the page. The mechanism for doing so is known as 'escaping to PHP'. There are four ways to do this:

Canonical PHP tags: The most universally effective PHP tag style is −

<? php...?>

If you use this style, you can be positive that your tags will always be correctly interpreted.

Short-open (SGML-style) tags: Short or short-open tags look like this−

<?...?>

Short tags are, as one might expect, the shortest option You must do one of two things to enable PHP to recognize the tags. You can choose the --enable-short-tags configuration option when you're building PHP. Set the short_open_tag setting in your php.ini file to on. This option must be disabled to parse XML with PHP because the same syntax is used for XML tags.

ASP-style Tags: ASP-style tags mimic the tags used by Active Server Pages to delineate code blocks. ASP-style tags look like this−

<%...%>

To use ASP-style tags, you will need to set the configuration option in your php.ini file.

HTML script Tags: HTML script tags look like this −

<script language = "PHP">...</script>

Commenting PHP Code

A comment is the portion of a program that exists only for the human reader and stripped out before displaying the programs result. There are two commenting formats in PHP:

- Single-line comments: They are generally used for short explanations or notes relevant to the local code. Here are the examples of single line comments.

```
<! DOCTYPE html>
<html>
<body>
<?php
// This is a single-line comment
# This is also a single-line comment
?>
</body>
</html>
```

Notes: Everything that follows the # character is ignored by the PHP interpreter.

- Multi-lines comments: They are generally used to provide pseudo code algorithms and more detailed explanations when necessary. The multiline style of commenting is the same as in C. Here is an example of multi-lines comments:

```
<! DOCTYPE html>
<html>
<body>
<?php
/*
```

```
This is a multiple-lines comment block
that spans over multiple
lines
*/
?>
</body>
</html>
```

Statements in PHP

The statements are expressions terminated by semicolons. A statement in PHP is any expression that is followed by a semicolon (;). Any sequence of valid PHP statements that is enclosed by the PHP tags is a valid PHP program. Here is a typical statement in PHP, which in this case assigns a string of characters to a variable called $greeting:

```
$greeting = "Welcome to PHP!”.
```

Whitespace in PHP

PHP is whitespace insensitive. The whitespace is the stuff you type that is typically invisible on the screen, including spaces, tabs, and carriage returns (end-of-line characters).White space in PHP is used to separate tokens in PHP source file. It is used to improve the readability of the source code.

```
public $isRunning;
```

White spaces are required in some places; for example between the access specified and the variable name. In other places, it is forbidden. It cannot be present in variable identifiers.

```
$a=1;
$b = 2;
$c = 3;
```

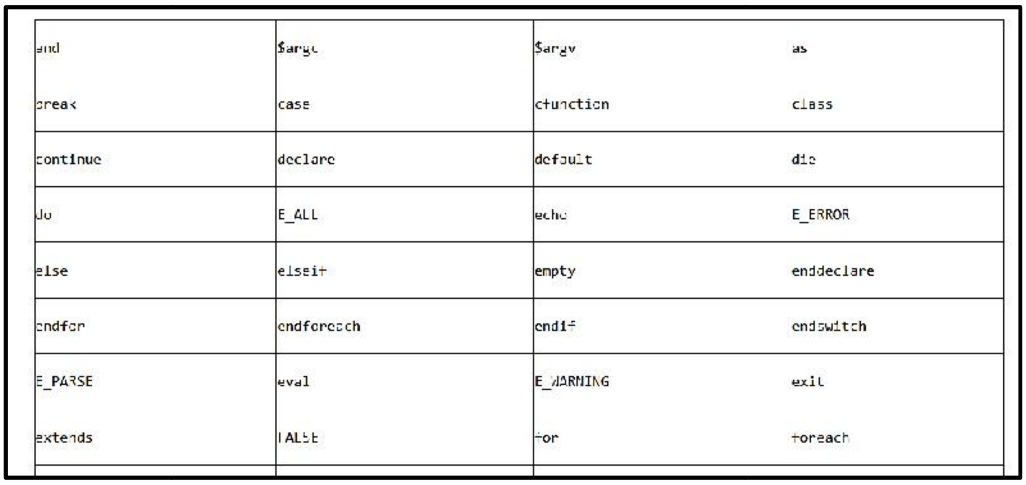
The amount of space put between tokens is irrelevant for the PHP interpreter. It is based on the preferences and the style of a programmer.

```
$a = 1;
$b = 2; $c = 3;
$d
=
4;
```

We can put two statements into one line. Or one statement into three lines. However, source code should be readable for humans. There are accepted standards of how to lay out your source code.

Keywords in PHP

PHP is case sensitive.However, PHPkeywords (e.g. if, else, while, echo, etc.), classes, functions, and user-defined functions are not case-sensitive.A keyword is a word reserved by the language for its core functionality you cannot give a variable, function, class, or constant the same name as a keyword. Some of the keywords in PHP are displayed below (Figure 5):



and	$argc	$argv	as
break	case	cfunction	class
continue	declare	default	die
do	E_ALL	echo	E_ERROR
else	elseif	empty	enddeclare
endfor	endforeach	endif	endswitch
E_PARSE	eval	E_WARNING	exit
extends	FALSE	for	foreach

Figure 4: List of Keywords in PHP

Example: In the following example, all the keywords ECHO, echo or EcHo are equal and legal.

```
<! DOCTYPE html>
<html>
<body>
<? php
ECHO "Hello World!<br>";
echo "Hello World!<br>";
EcHo "Hello World!<br>";
?>
</body>
</html>
```

Variables in PHP

A variable is an identifier, which holds a value. In programming we say that we assign a value to a variable. Technically speaking, a variable is a reference to a computer memory, where the value is stored. In PHP language, a variable can hold a string, a number, or various objects like a function or a class. Variables can be assigned different values over time.Variables in PHP consist of the $ character, called a sigil, and a label. A label can be created from alphanumeric characters and an underscore _ character. A variable cannot begin with a number. The PHP interpreter can then distinguish between a number and a variable more easily.

Example: $Value

$value2

$company name

These were valid PHP identifiers.

$12Val

$exx$

$first-name

These were examples of invalid PHP identifiers.

Variables in PHP are case-sensitive. That is, $name, $NAME, and $NaME are three different variables.

```
<? php
$number = 10;
$Number = 11;
$NUMBER = 12;
echo $number, $Number, $NUMBER;
echo "\n";
?>
```

Example 1: Look at the example below; only the first statement will display the value of the $color variable! This is because $color, $COLOR, and $co LOR is treated as three different variables:

```
<! DOCTYPE html>
<html>
<body>
<?php
$color = "red";
echo "My car is " . $color . "<br>";
echo "My house is " . $COLOR . "<br>";
echo "My boat is " . $coLOR . "<br>";
?>
</body>
</html>
```

Example 2:

```
<! DOCTYPE html>
<html>
<body>
<?php
$color = "red";
echo "My car is " . $color . "<br>";
echo "My house is " . $COLOR . "<br>";
$color= "blue";
echo "My hair is " . $color . "<br>";
echo "My boat is " . $coLOR . "<br>";
echo "My boat is " . $color . "<br>";
?>
</body>
</html>
```

Example 3:

```
<html>
<body>
<?php
        $capital = 67;
print("Variable capital is $capital<br>");
print("Variable CaPiTaL is $CaPiTaL<br>");
        ?>
</body>
</html>
```

Braces

Braces make Blocks. Although statements cannot be combined like expressions, you can always put a sequence of statements anywhere a statement can go by enclosing them in a set of curly braces.Here both statements areequivalent −

```
if (3 == 2 + 1)
print("Good - I haven't totally lost my mind.<br>");
if (3 == 2 + 1)
{
print("Good - I haven't totally");
print("lost my mind.<br>");
}
```

Constants

Constant is an identifier for a value which cannot change during the execution of the script. By convention, constant identifiers are always uppercase.Example:

```
<? php
define ("SIZE", 300);
define ("EDGE", 100);
#SIZE = 100;
echo SIZE;
echo EDGE;
echo "\n";
?>
```

In the script, we define two constants, that is,

```
define ("SIZE", 300);
define ("EDGE", 100);
```

Literal in PHP

A literal is any notation for representing a value within the PHP source code. Technically, a literal is assigned a value at compile time, while a variable is assigned at runtime.

```
$age = 29;
```

```
$nationality = "Hungarian";
```

1.9 Data Types in PHP

PHP String: A string is a sequence of characters, like "Hello world!". A string can be any text inside quotes. You can use single or double quotes. Example

```
<? php
$str = "Hello world!";
$strs = 'Web Development using PHP';
echo $str;
echo "<br>";
echo $strs;
?>
```

PHP Integer: An integer data type is a non-decimal number between -2,147,483,648 and 2,147,483,647. The following are the rules for integers:

- An integer must have at least one digit.
- An integer must not have a decimal point.
- An integer can be either positive or negative.
- Integers can be specified in: decimal (base 10), hexadecimal (base 16), octal (base 8), or binary (base 2) notation.

PHP Float:A float (floating point number) is a number with a decimal point or a number in exponential form. In the following example $a is a float. The PHP var_dump() function returns the data type and value:

```
<? php
$a = 15.18;
var_dump($a);
?>
```

PHP Boolean:A Boolean represents two possible states: TRUE or FALSE.

```
$x = true;
$y = false;
```

Booleans are often used in conditional testing.

PHP Array: Arrays are named and indexed collections of other values. An array stores multiple values in one single variable. In the following example $courses is an array. The PHP var_dump() function returns the data type and value:

```
<? php
$courses=array ("DataStructures","Data Communications","PHP","Python");
var_dump($courses);
?>
```

PHP Object: Objects − are instances of programmer-defined classes, which can package up both other kinds of values and functions that are specific to the class.

- Classes and objects are the two main aspects of object-oriented programming.
- A class is a template for objects, and an object is an instance of a class.
- When the individual objects are created, they inherit all the properties and behaviors from the class, but each object will have different values for the properties.

PHP NULL Value: Null is a special data type which can have only one value: NULL. A variable of data type NULL is a variable that has no value assigned to it.

```
<? php
$a="How are you";
$b=null;
var_dump($a);
var_dump($b);
?>
```

PHP Resource: The special resource type is not an actual data type. It is the storing of a reference to functions and resources external to PHP. A common example of using the resource data type is a database call.

1.10 Variables in PHP

Variable is an identifier, which holds a value. In programming we say that we assign a value to a variable. Technically speaking, a variable is a reference to a computer memory, where the value is stored. In PHP language, a variable can hold a string, a number, or various objects like a function or a class. Variables can be assigned different values over time. The syntax of declaring a variable in PHP is given below:

```
$variable name=value;
```

Variables in PHP consist of the $ character, called a sigil, and a label. A variable must start with a dollar ($) sign, followed by the variable name.A label can be created from alphanumeric characters and an underscore _ character. A variable name must start with a letter or underscore (_) character.

As PHP is a loosely typed language, so we do not need to declare the data types of the variables. It automatically analyses the values and makes conversions to its correct datatype.

- After declaring a variable, it can be reused throughout the code.
- Assignment Operator (=) is used to assign the value to a variable.
- The value of a variable is the value of its most recent assignment.
- Variables are assigned with the = operator, with the variable on the left-hand side and the expression to be evaluated on the right.
- Variables can, but do not need, to be declared before assignment.
- Variables in PHP do not have intrinsic types- a variable does not know in advance whether it will be used to store a number or a string of characters.
- Variables used before they are assigned have default values.
- PHP does a good job of automatically converting types from one to another when necessary.
- PHP variables are Perl-like.

Rules for Declaring PHP Variables

- A PHP variable name cannot contain spaces.
- One thing to be kept in mind that the variable name cannot start with a number or special symbols.
- PHP variables are case-sensitive, so $name and $NAME both are treated as different variable.

Creating (Declaring) PHP Variables

The following example illustrates the creation or declaration of variables in PHP. In the following example, a string variable ($txt) and two other variables $x and $y have been declared.

```
<? php
$txt="Hello PHP Students";
```

```
$x=15;
$y=12.5;
echo $txt;
echo $x;
?>
```

PHP Variables Scope

PHP has three different variable scopes:

- Local: A variable declared within a function has a LOCAL SCOPE and can only be accessed within that function.
- Global: A variable declared outside a function has a GLOBAL SCOPE and can only be accessed outside a function.
- Static: When a function is completed/executed, all its variables are deleted. However, sometimes we want a local variable NOT to be deleted. We need it for a further job.

Output Variables in PHP

1. Echo statement is often used to output data to the screen.

```
<?php
$txt = “lpu.in";
echo “This is $txt!";
?>
```

2. Print statement can be used to display the variables. The following example shows how to output text and variables with the print statement:

```
<?php
$str="Learn language";
$str1="I am open to learn anything new";
#$r=20;
#$z=45;
print "<h2>" . $str . "</h2>";
print "<h2>" . $str1 . "</h2>";
print "<h2>" . $str."  and” . $str1. "</h2>";
#echo $r+$z;
?>
```

Echo vs Print Statements

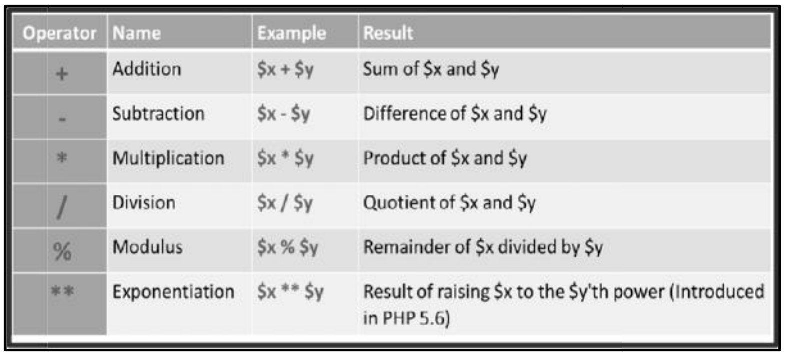
Echo and print are the same. They are both used to output data to the screen.The differences are small:

- Echo has no return value while print has a return value of 1 so it can be used in expressions.
- Echo can take multiple parameters (although such usage is rare) while print can take one argument.
- Echo is marginally faster than print.

1.11 PHP Operators

The operators are used to perform operations on variables and values.PHPoperator is a symbolthat isused to perform operations on the operands. The various types of operators are:

- **PHP Arithmetic Operators:** PHP arithmetic operators are used to perform common arithmetic operations such as addition, subtraction, etc. with numeric values (Figure 6).



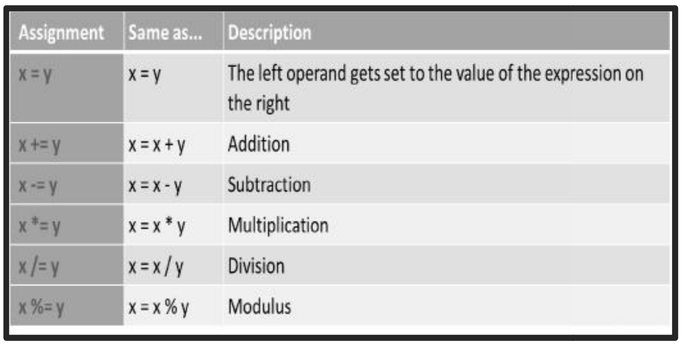
Operator	Name	Example	Result
+	Addition	$x + $y	Sum of $x and $y
-	Subtraction	$x - $y	Difference of $x and $y
*	Multiplication	$x * $y	Product of $x and $y
/	Division	$x / $y	Quotient of $x and $y
%	Modulus	$x % $y	Remainder of $x divided by $y
**	Exponentiation	$x ** $y	Result of raising $x to the $y'th power (Introduced in PHP 5.6)

Figure 5: Arithmetic Operators in PHP

Example: Write a program to add 2 numbers?

```
<? php
$num1= 20;
$num2=30;
$num= $num1+$num2;
echo $num . "</br>";
$numberS=10+50;
echo $numberS;
#Here + is the operator and 20, 3 are the operands.
?>
```

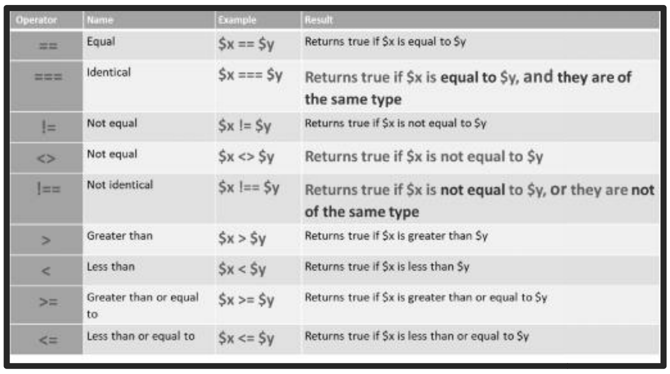
- **PHP Assignment Operators:**The basic assignment operator in PHP is "=" (Figure 7).



Assignment	Same as...	Description
x = y	x = y	The left operand gets set to the value of the expression on the right
x += y	x = x + y	Addition
x -= y	x = x - y	Subtraction
x *= y	x = x * y	Multiplication
x /= y	x = x / y	Division
x %= y	x = x % y	Modulus

Figure 6: Assignment Operators in PHP

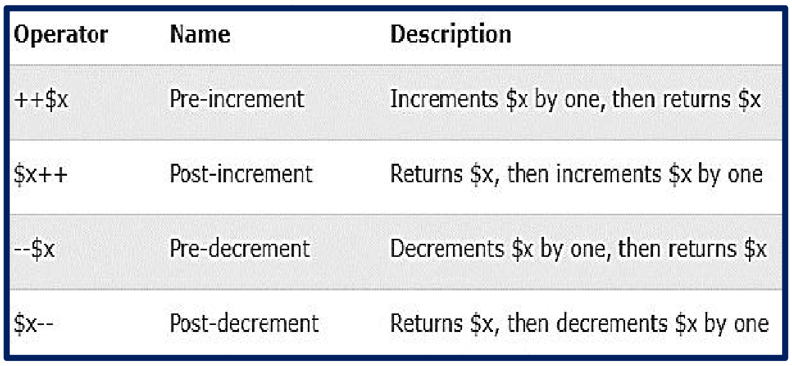
PHP Comparison Operators: The comparison operators allow comparing two values, such as number or string (Figure 8).



Operator	Name	Example	Result
==	Equal	$x == $y	Returns true if $x is equal to $y
===	Identical	$x === $y	Returns true if $x is **equal to** $y, and **they are** of **the same type**
!=	Not equal	$x != $y	Returns true if $x is not equal to $y
<>	Not equal	$x <> $y	Returns true if $x is not equal to $y
!==	Not identical	$x !== $y	Returns true if $x is **not equal** to $y, or they are **not of the same type**
>	Greater than	$x > $y	Returns true if $x is greater than $y
<	Less than	$x < $y	Returns true if $x is less than $y
>=	Greater than or equal to	$x >= $y	Returns true if $x is greater than or equal to $y
<=	Less than or equal to	$x <= $y	Returns true if $x is less than or equal to $y

Figure 7: Comparison Operators in PHP

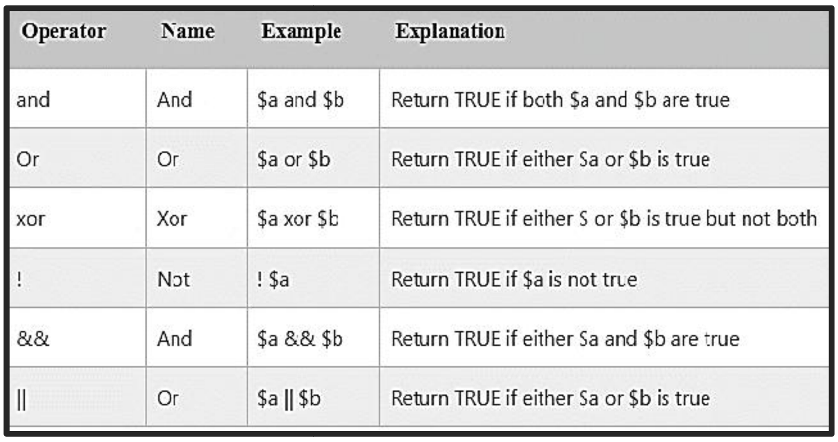
Incrementing/Decrementing Operators: The increment and decrement operators are used to increase and decrease the value of a variable (Figure 9).



Operator	Name	Description
++$x	Pre-increment	Increments $x by one, then returns $x
$x++	Post-increment	Returns $x, then increments $x by one
--$x	Pre-decrement	Decrements $x by one, then returns $x
$x--	Post-decrement	Returns $x, then decrements $x by one

Figure 8: Incrementing/Decrementing Operators in PHP

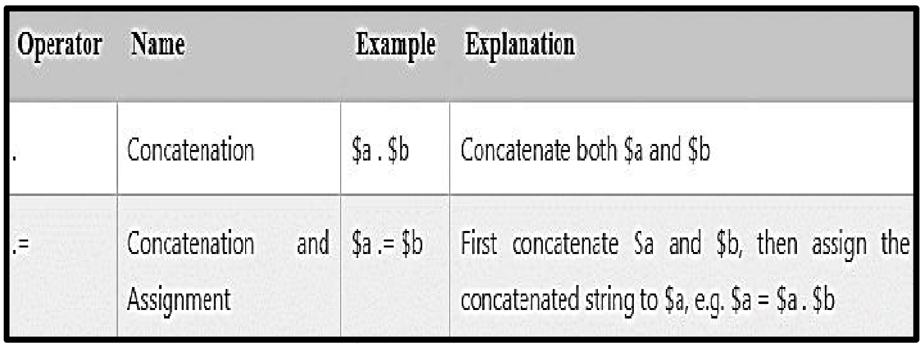
Logical Operators: PHP logical operators are used to combine conditional statements (Figure 9).



Operator	Name	Example	Explanation
and	And	$a and $b	Return TRUE if both $a and $b are true
Or	Or	$a or $b	Return TRUE if either Sa or $b is true
xor	Xor	$a xor $b	Return TRUE if either S or $b is true but not both
!	Not	! $a	Return TRUE if $a is not true
&&	And	$a && $b	Return TRUE if either Sa and $b are true
||	Or	$a || $b	Return TRUE if either Sa or $b is true

Figure 9: Logical Operators in PHP

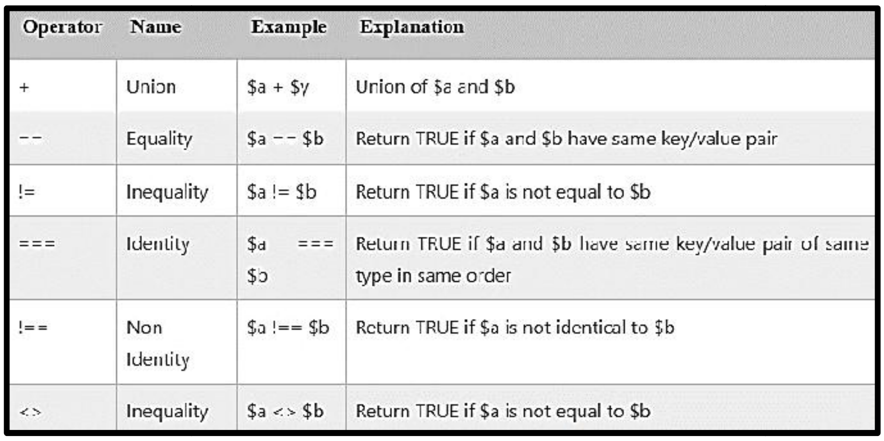
String Operators: These operators are used to perform the operation on strings (Figure 11). There are two string operators in PHP, which are given below:



Operator	Name	Example	Explanation
.	Concatenation	$a . $b	Concatenate both $a and $b
.=	Concatenation and Assignment	$a .= $b	First concatenate Sa and $b, then assign the concatenated string to $a, e.g. $a = $a . $b

Figure 10: String Operators in PHP

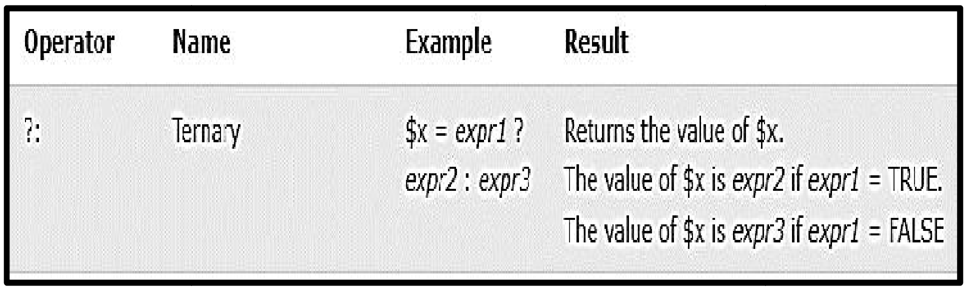
Array Operators: The array operators are used in case of array (Figure 12). Basically, these operators are used to compare the values of arrays.



Operator	Name	Example	Explanation
+	Union	$a + $y	Union of $a and $b
==	Equality	$a == $b	Return TRUE if $a and $b have same key/value pair
!=	Inequality	$a != $b	Return TRUE if $a is not equal to $b
===	Identity	$a === $b	Return TRUE if $a and $b have same key/value pair of same type in same order
!==	Non Identity	$a !== $b	Return TRUE if $a is not identical to $b
<>	Inequality	$a <> $b	Return TRUE if $a is not equal to $b

Figure 11: Array Operators in PHP

Conditional Operator: The conditional operator first evaluates an expression for a true or false value and then executes one of the two given statements depending upon the result of the evaluation (Figure 13).



Operator	Name	Example	Result
?:	Ternary	$x = *expr1* ? *expr2* : *expr3*	Returns the value of $x. The value of $x is *expr2* if *expr1* = TRUE. The value of $x is *expr3* if *expr1* = FALSE

Figure 12: Conditional Operators in PHP

1.12 PHP Constants

PHP constants are name or identifier that can't be changed during the execution of the script except for magic constants, which are not really constants. The constants are similar to the variable except once they defined; they can never be undefined or changed. They remain constant across the entire program. PHP constants follow the same PHP variable rules. For example, it can be started with a letter or underscore only.Conventionally,PHP constants should be defined in uppercase letters. The PHP constants can be defined by 2 ways:

1. Using define () function: Use the define() function to create a constant. It defines constant at run time. The syntax of define () function is:

```
define (name, value, case-insensitive)
```

where:

- Name: It specifies the constant name.
- Value: It specifies the constant value.
- Case-insensitive: Specifies whether a constant is case-insensitive. Default value is false. It means it is case sensitive by default.Example:

```
<?php
define ("MESSAGE","HelloJavaTpoint PHP");
echo MESSAGE;
?>
```

Example: <?php

```
define("SIZE", 300);
define("EDGE", 100);
#SIZE = 100;
echo SIZE;
echo EDGE;
echo "\n";
?>
```

In the script, we define two constants.

```
Define ("SIZE", 300);
Define ("EDGE", 100);
```

2. Using const keyword: PHP introduced a keyword const to create a constant. The const keyword defines constants at compile time. It is a language construct, not a function. The constant defined using const keyword is case-sensitive.

```
<? php
Const MESSAGE="Hello const by Java point PHP";
EchoMESSAGE;
?>
```

Constant () function

PHP constants can also be used using Constant() function. There is another way to print the value of constants using constant () function instead of using the echo statement.The syntax for the following constant function:

```
Constant (name)
<?php
define("MSG", "JavaTpoint");
        Echo MSG, "</br>";
        echo constant("MSG");
//both are similar
        ?>
```

Constant vs Variables in PHP

Table 1: Constants vs Variables

Constants	Variables
Constants do not follow any variable scoping rules, and they can be defined and accessed anywhere.	Variables can be declared anywhere in the program, but they follow variable scoping rules.
Constants do not follow any variable scoping rules, and they can be defined and accessed anywhere.	Variables can be declared anywhere in the program, but they follow variable scoping rules.
By default, constants are global.	Variables can be local, global, or static.
Once the constant is defined, it can never be redefined.	A variable can be undefined as well as redefined easily.
A constant can only be defined using	A variable can be defined by simple assignment

define() function. It cannot be defined by any simple assignment.	(=) operator.
There is no need to use the dollar ($) sign before constant during the assignment.	To declare a variable, always use the dollar ($) sign before the variable.

PHP Magic Constants

Magic constants are the predefined constants in PHP which get changed on the basis of their use. They start with double underscore (__) and ends with double underscore.They are similar to other predefined constants but as they change their values with the context, they are called magic constants. There are nine magic constants in PHP. In which eight magic constants start and end with double underscores (__).

- __LINE__
- __FILE__
- __DIR__
- __FUNCTION__
- __CLASS__
- __TRAIT__
- __METHOD__
- __NAMESPACE__
- ClassName::class

All of the constants are resolved at compile-time instead of run time, unlike the regular constant. Magic constants are case-insensitive.

- **__LINE__:** It returns the current line number of the file, where this constant is used.

```
Example: <? php
echo "<h3>Example for __LINE__</h3>";
// print Your current line number, that is, 4
echo "You are at line number " . __LINE__ . "<br><br>";
?>
```

- **__FILE__:** This magic constant returns the full path of the executed file, where the file is stored. If it is used inside the include, the name of the included file is returned.

```
Example: <?php
echo "<h3>Example for __FILE__</h3>";
//print full path of file with .php extension
echo __FILE__ . "<br><br>";
?>
```

- __DIR__: Returns the full directory path of the executed file. The path returned by this magic constant is equivalent to dirname(__FILE__). This magic constant does not have a trailing slash unless it is a root directory.

```
Example: <?php
echo "<h3>Example for __DIR__</h3>";
//print full path of directory where script will be placed
echo __DIR__ . "<br><br>";
```

```
//below output will equivalent to above one.
echo dirname(__FILE__) . "<br><br>";
?>
```

- **__FUNCTION__:**These magic constant returns the function name, where this constant is used. It will return blank if it is used outside of any function.
- **__CLASS__:** It returns the class name, where this magic constant is used. __CLASS__ constant also works in traits.

1.13 Comments in PHP

A comment is the portion of a program that exists only for the human reader and stripped out before displaying the programs result. There are two commenting formats in PHP:

Single-line comments:They are generally used for short explanations or notes relevant to the local code. Here are the examples of single line comments.

```
<! DOCTYPE html>
<html>
<body>
<?php
// This is a single-line comment
# This is also a single-line comment
?>
</body>
</html>
```

Example: Everything that follows the # character is ignored by the PHP interpreter.

Multi-lines comments: They are generally used to provide pseudo code algorithms and more detailed explanations when necessary. The multiline style of commenting is the same as in C. An example of multi-lines comments is as follows:

```
<! DOCTYPE html>
<html>
<body>
<? php
/*
This is a multiple-lines comment block
That spans over multiple
lines
*/
?>
</body>
</html>
```

1.14 Running PHP

PHP is the most popular and widely used server-side scripting language for web development. However, it requires a web server to run even a locally developed webpage. There is various web server software for setting up our local webserver. Amongst them, PHP XAMPP and WampServer are the most popular.

While WampServer is only available for the Windows platform, XAMPP is a cross-platform application that can run on Windows, Linux, and macOS.

XAMPP

XAMPP is a free and open-source cross-platform web server solution stack package developed by Apache Friends, consisting mainly of the Apache HTTP Server, MariaDB database, and interpreters for scripts written in the PHP and Perl programming languages.Since most actual web server deployments use the same components as XAMPP, it makes transitioning from a local test server to a live server possible.XAMPP's ease of deployment means a WAMP or LAMP stack can be installed quickly and simply on an operating system by a developer, with the advantage that common add-in applications such as WordPress and Joomla! can also be installed with similar ease using Bitnami. It can be acronym as XAMPP Apache + MariaDB + PHP + Perl (Table 2). But later on, MariaDB was replaced by MySQL in 2015.

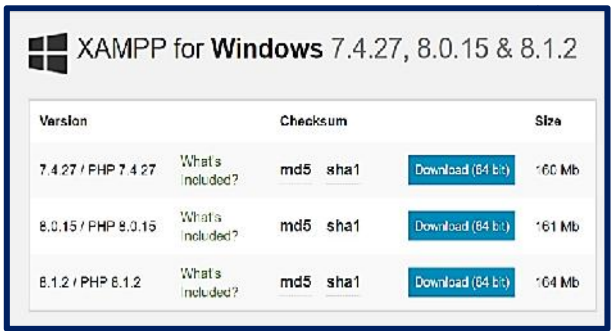
Table 2: XAMPP Full Form

Letter	Meaning
X	An ideographic letter referring to cross-platform
A	Apache, or its expanded form, Apache HTTP Server
M	MariaDB (formerly MySQL
P	PHP
P	Perl

XAMPP eases the way in which WAMP webserver stack can be deployed and instantiated. Later some common packaged applications that could be easily installed were provided by Bitnami.Officially, XAMPP's designers intended it for use only as a development tool, to allow website designers and programmers to test their work on their own computers without any access to the Internet. To make this as easy as possible, many important security features are disabled by default.XAMPP has the ability to serve web pages on the World Wide Web. A special tool is provided to password-protect the most important parts of the package.XAMPP also provides support for creating and manipulating databases in MariaDB and SQLite among others.Once XAMPP is installed, it is possible to treat a localhost like a remote host by connecting using an FTP client. Using a program like FileZilla has many advantages when installing a content management system (CMS) like Joomla or WordPress. It is also possible to connect to localhost via FTP with an HTML editor.

Download XAMPP

In order to download XAMPP, go to https://www.apachefriends.org/download.html (Figure 14 - Figure 15)



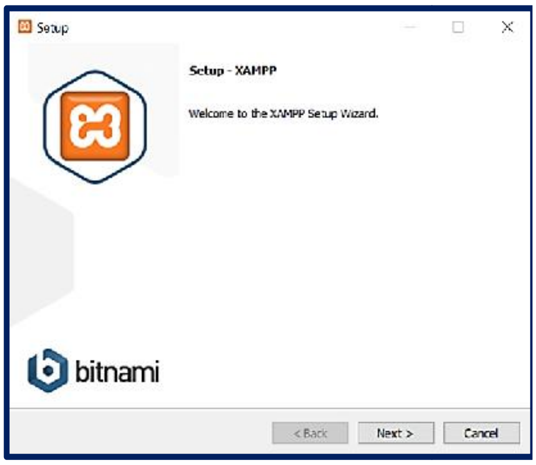
XAMPP for **Windows** 7.4.27, 8.0.15 & 8.1.2

Version		Checksum			Size
7.4.27 / PHP 7.4.27	What's Included?	md5	sha1	Download (64 bit)	160 Mb
8.0.15 / PHP 8.0.15	What's Included?	md5	sha1	Download (64 bit)	161 Mb
8.1.2 / PHP 8.1.2	What's Included?	md5	sha1	Download (64 bit)	164 Mb

Figure 13: XAMPP Download

Installation of XAMPP

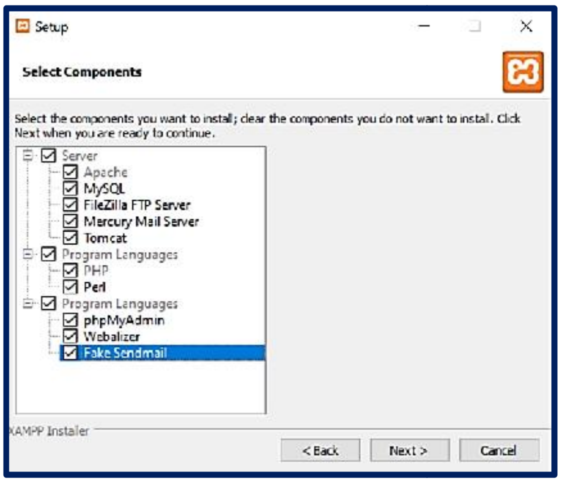
On completing the download of the setup file, begin the installation process and, in the "Select Components" section, select all the required components(Figure 15-Figure 22).



Setup - XAMPP

Welcome to the XAMPP Setup Wizard.

Figure 14: XAMPP Installation Screen

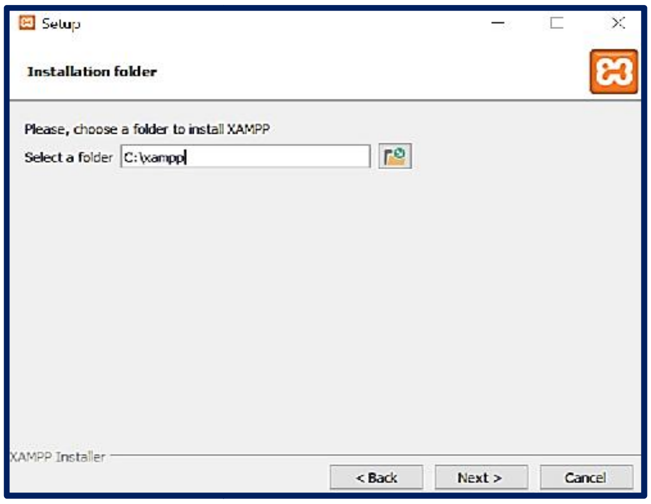


Select Components

Select the components you want to install; clear the components you do not want to install. Click Next when you are ready to continue.

- ☑ Server
  - ☑ Apache
  - ☑ MySQL
  - ☑ FileZilla FTP Server
  - ☑ Mercury Mail Server
  - ☑ Tomcat
- ☑ Program Languages
  - ☑ PHP
  - ☑ Perl
- ☑ Program Languages
  - ☑ phpMyAdmin
  - ☑ Webalizer
  - ☑ Fake Sendmail

Figure 15: Select Components to Install

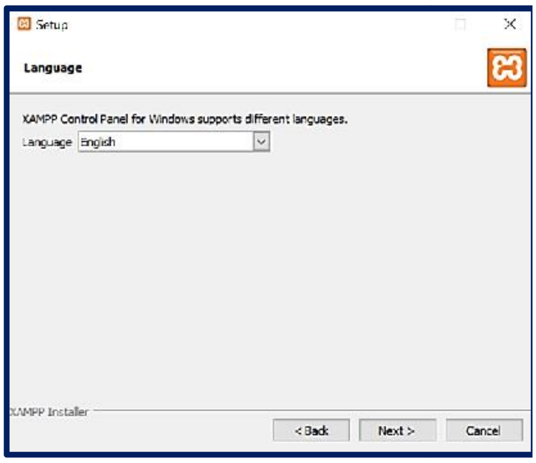


Installation folder

Please, choose a folder to install XAMPP

Select a folder C:\xampp

Figure 16: Installation Folder

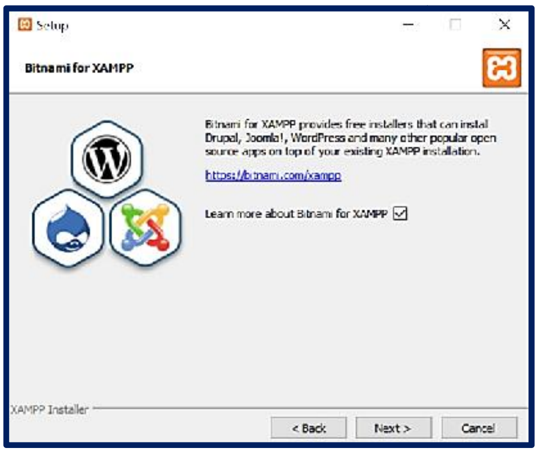


Language

XAMPP Control Panel for Windows supports different languages.

Language English

Figure 17: Language Selection



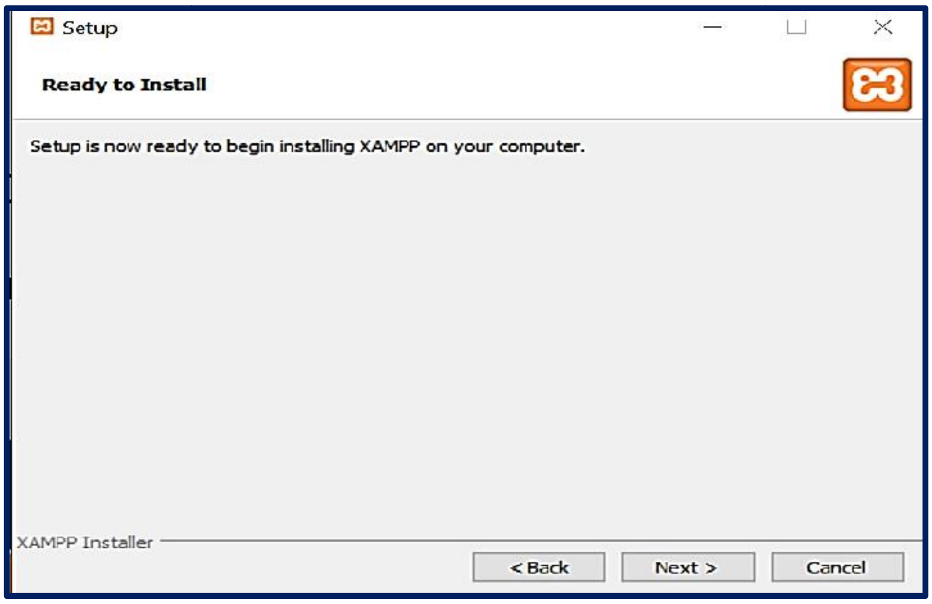
Bitnami for XAMPP

Bitnami for XAMPP provides free installers that can install Drupal, Joomla!, WordPress and many other popular open source apps on top of your existing XAMPP installation.

https://bitnami.com/xampp

Learn more about Bitnami for XAMPP ☑

Figure 18: Bitnami XAMPP



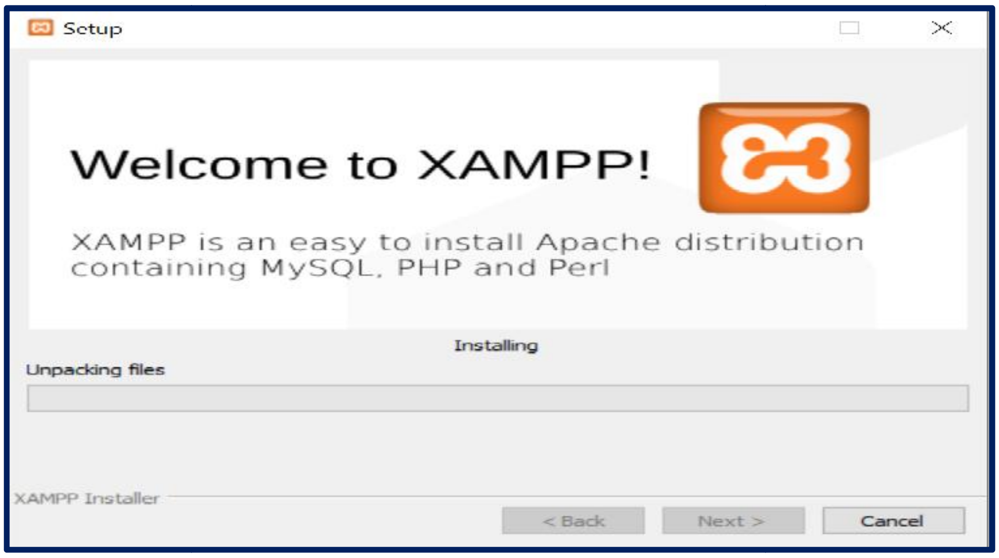
Setup

Ready to Install

Setup is now ready to begin installing XAMPP on your computer.

XAMPP Installer

Figure 19: Installation Selection Option



Setup

Welcome to XAMPP!

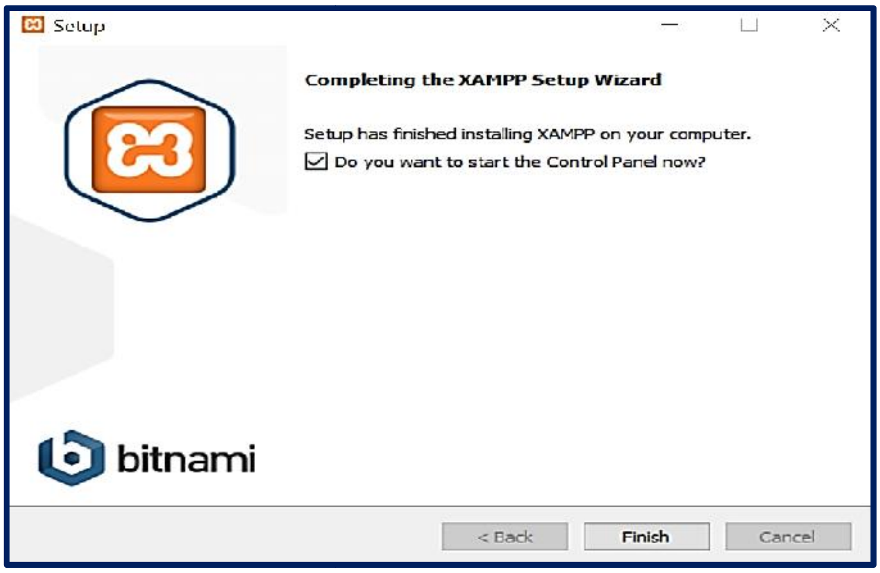
XAMPP is an easy to install Apache distribution containing MySQL, PHP and Perl

Installing

Unpacking files

XAMPP Installer

Figure 20: Installation Start Screen


Setup

Completing the XAMPP Setup Wizard

Setup has finished installing XAMPP on your computer.

☑ Do you want to start the Control Panel now?

Figure 21: XAMPP Installation Finish Screen

1.15 How to Run a PHP Code Using XAMPP?

- Before running a PHP script, you must know where to write it.
- In the XAMPP directory, there exists a folder called "htdocs". This is where all the programs for the web pages will be stored.
- Now, to run a PHP script:
- Go to "C:\xampp\htdocs" and inside it, create a folder. Let's call it "test". It's considered good practice to create a new folder for every project you work on.
- Inside the test folder, create a new text file and name it "index.php" and write an initial PHP script.
- Now, to see the script output, open the XAMPP control panel and start Apache to host the local webserver, where our script will be running. (Make sure any other services like IIS are stopped).
- Now navigate to your browser and type in "localhost/test/" in the address bar to view the output.

Summary

- PHP is a server scripting language, and a powerful tool for making dynamic and interactive Web pages. It is widely used, free, and efficient alternative to competitors such as Microsoft's ASP.
- PHP is an open-source, interpreted, and object-oriented scripting language that can be executed at the server-side.
- PHP stands for Hypertext Pre-processor. It originally stood for Personal Home Page, but it now stands for the recursive initialism PHP: Hypertext Pre-processor. PHP is an interpreted language, i.e., there is no need for compilation.
- PHP performs system functions, that is, from files on a system it can create, open, read, write, and close them. PHP can handle forms, that is, gather data from files, save data to a file, through email you can send data, return data to the user.

- WordPress (WP, WordPress.org) is a free and open-source content management system (CMS) written in PHP and paired with a MySQL or MariaDB database.
- PHP code is usually processed on a web server by a PHP interpreter implemented as a module, a daemon or as a Common Gateway Interface (CGI) executable. On a web server, the result of the interpreted and executed PHP code – which may be any type of data, such as generated HTML or binary image data– would form the whole or part of an HTTP response.

Keywords

- ***PHP:*** PHP is a server scripting language, and a powerful tool for making dynamic and interactive Web pages. PHP is a widely used, free, and efficient alternative to competitors such as Microsoft's ASP.
- ***PHP Parser:***In order to process PHP script instructions a parser must be installed to generate HTML output that can be sent to the Web browser.
- ***Lexical structure:*** Lexical structure of a programming language is the set of basic rules that governs how you write programs in that language.
- ***Magic Constants:***PHP constants are name or identifier that can't be changed during the execution of the script except for magic constants, which are not really constants. The constants are similar to the variable except once they defined, they can never be undefined or changed. They remain constant across the entire program.
- ***CakePHP:***CakePHP is an open-source web framework. It is intended to make developing, deploying and maintaining applications much easier. It follows the model–view–controller (MVC) approach and is written in PHP, modeled after the concepts of Ruby on Rails, and distributed under the MIT License.
- ***Magento:*** Magento 2 is a CMS (Content Management System), developed by Varien Inc. It is open-source software which is very useful software for online business.

SelfAssessment

1. The _______________ syntax of the language and specifies such things as what variable names look like, what characters are used for comments, and how program statements are separated from each other.

A. Highest-level
B. Lowest-level
C. Slow-level
D. Fast-level

2. What does PHP stand for?

A. Pretext Hypertext Processor
B. Hypertext Preprocessor
C. Preprocessor Home Page
D. Personal Home Paging

3. PHP is an example of ___________ scripting language.

A. Browser-side

B. Client-side

C. In-side

D. Server-side

4. PHP mascot is a/an

A. Elephant

B. Fish

C. Cat

D. Rhinoceros

5. Which of the following is NOT an application of PHP?

A. PHP can handle forms, that is, gather data from files, save data to a file, through email you can send data, return data to the user.

B. Allows to add, delete, modify elements within your database through PHP.

C. Can access cookies variables and set cookies.

D. You cannot restrict users to access some pages of your website.

6. Which of the following corresponds to the characteristic(s) of PHP?

A. Loosely typed language

B. Web server support

C. Platform independence

D. All of the above

7. Which of the following is the default file extension of PHP?

A. .hphp

B. .php

C. .xml

D. .html

8. Which of the following starts with __ (double underscore) in PHP?

A. Inbuilt constants

B. User-defined constants

C. Magic constants

D. Default constants

9. _____________ is a cross-platform application that can run on Windows, Linux, and macOS.

A. HAMP

B. Hadoop

C. XAMPP

D. WampServer

10. Variables in PHP are case-sensitive.
A. True
B. False
11. The PHP var_dump() function returns the ___________.
A. Value only
B. Data type only
C. Data type and value both
D. None of the above

12. _____________ operatoris used to assign the value to a variable.
A. Logical
B. Assignment
C. Unary
D. Equivalent

13. In PHP, the variables start with _____?
A. &
B. $
C. #
D. %

14. Which of the following is not a variable scope in PHP?
A. Extern
B. Local
C. Static
D. Global

15. PHP ____________ operators are used to combine conditional statements.
A. Assignment
B. Arithmetic
C. Array
D. Logical

Answers for Self Assessment

1.	B	2.	B	3.	D	4.	A	5.	D
6.	D	7.	B	8.	C	9.	C	10.	A
11.	C	12.	B	13.	A	14.	A	15.	D

Review Questions

1. What are the different data types supported by PHP?
2. Define Magic constants and their significance.
3. Indicate the different ways of defining constants in PHP.
4. Write a short note on:
   a. Comments in PHP
   b. Constants in PHP
   c. Arithmetic operators
5. Define the lexical structure of PHP.
6. Discuss the need for operators in PHP with the help of examples.
7. How variables are declared in PHP? Illustrate with examples.
8. Compare constants and variables in PHP.
9. What are variables in PHP? Describe the rules for declaring and using variables in PHP?
10. How PHP is executed on the Web?

Further Readings

- Joy of PHP: A Beginner's Guide by Alan Forbes.
- Programming PHP: Creating Dynamic Web Pages by Kevin Tatroe and Peter MacIntyre.
- Learn PHP in One Day and Learn It Well by Jamie Chan.
- Web Technologies: A Computer Science Perspective by Jackson, PearsonEducation India.

Web Links

- https://www.w3schools.com/php/
- https://en.wikipedia.org/wiki/PHP
- https://www.php.net/manual/en/language.oop5.basic.php
- https://youtu.be/hx38tnlYGlA
- https://youtu.be/C9Tg1h53Q8s
- https://youtu.be/52YfLZApZ8I

Unit 02: Statements in PHP

CONTENTS

Objectives

After studying this unit, you will be able to:

- Learn about the control statements in PHP.
- Know about the flow statements such as If, If-else, If-elseif-else etc.
- Explore the loops in PHP such as while, do-while, for loop.
- Learn about embedding PHP in web pages.

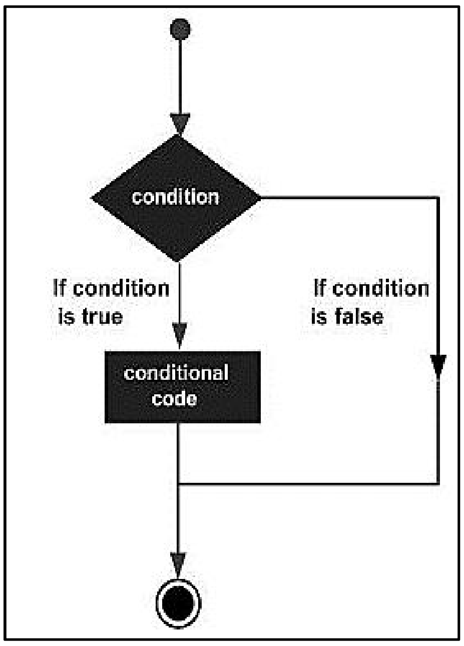
Introduction

One of the main reasons for using scripting languages such as PHP is to build logic and intelligence into the creation and deployment of web based data. In order to be able to build logic into PHP based scripts, it is necessary for the script to be able to make decisions and repeat tasks based on specified criteria. For example, it may be necessary to repeat a section of a script a specified number of times, or perform a task only if one or more conditions are found to be true. In programming terms this is known as flow control and looping. In the simplest terms, this involves some standard scripting structures provided by languages such as PHP to control the logic and overall behavior of a script.Each of these structures provides a simple and intuitive way to build intelligence into scripts.

2.1 Conditional Statements in PHP

Sometimes when you write code, you want to perform different actions for different decisions. You can use conditional statements in your code to do this. The conditional statements are the set of commands used to perform different actions based on different conditions.You can use conditional statements in your code to make your decisions. PHP supports following three decision making statements:

- if...else statement: Use this statement if you want to execute a set of code when a condition is true and another if the condition is not true.



- Elseif statement: Used with the if...else statement to execute a set of code if one of the several condition is true.
- Switch statement: Used if you want to select one of many blocks of code to be executed, use the Switch statement. The switch statement is used to avoid long blocks of if..else if..else code.

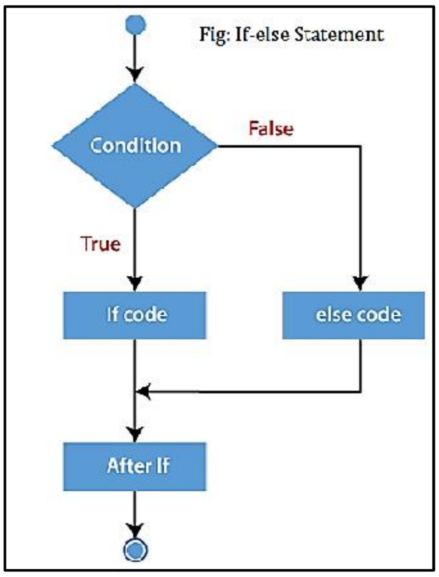
2.2 PHP If Statement

PHP if statement allows conditional execution of code. It is executed if condition is true. The If statement is used to executes the block of code exist inside the if statement only if the specified condition is true.

Syntax:

```
if (condition)
{
```

```
//code to be executed
}
```



You might have a script that checks if Boolean value is true or false, if variable contains number or string value, if an object is empty or populated, etc. The condition can be anything you choose, and you can combine conditions together to make for actions that are more complicated.

Lab Exercise: Write a program to find the smaller of two numbers?

Compound If Statements: You may also compound the statements using elseif to have multiple conditions tested insequence. You should use this construction if you want to select one of many sets of lines toexecute.

```
<?php
$num1 = 12;
$num2 = 15;
$num3=20;
if ( ( $num1 <$num2) && ($num2<$num3) )//Using Compound If Statement
{
  echo “$num1 is 12, $num2 is 15 and $num3 is 20";
}
?>
```

2.3 PHP If-else Statement

PHP if-else statement is executed whether condition is true or false. If-else statement is slightly different from if statement. It executes one block of code if the specified condition is true and another block of code if the condition is false.The syntax if-else statement is:

```
if (condition)
{
statements_1
}
else
```

```
{
statements_2
}
```

Lab Exercise: Write a program to decide whether a student has passed an exam with a pass mark of 57?Write a program to check whether a number is even or odd?

2.4 PHP if...elseif...else Statement

PHP if...elseif...else Statement is a special statement used to combine multiple if?.else statements. So, we can check multiple conditions using this statement.The if...elseif...else statement executes different codes for more than two conditions. The syntax:

```
if (condition1) {
//code to be executed if condition1 is true
}
elseif (condition2) {
//code to be executed if condition2 is true
}
elseif (condition3) {
//code to be executed if condition3 is true
....
}
else {
//code to be executed if all given conditions are false
}
```

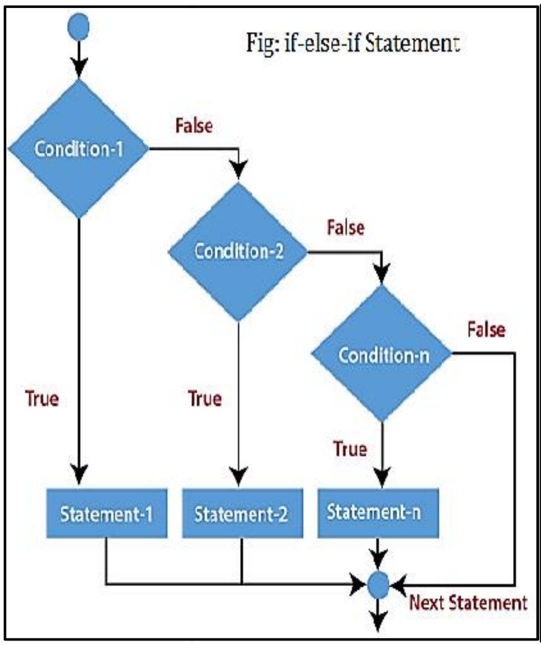


Fig: if-else-if Statement

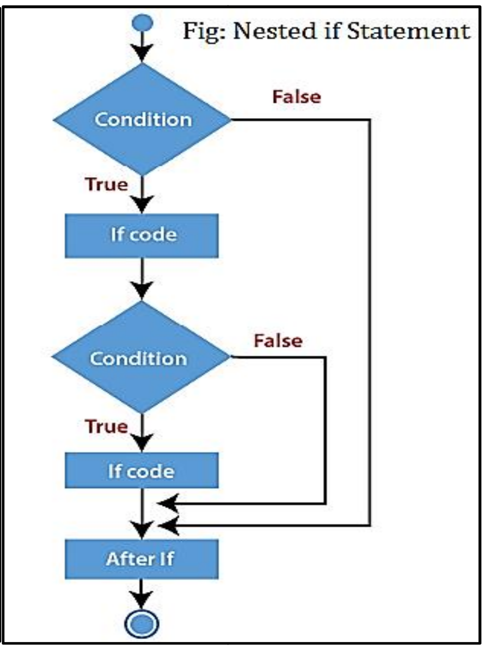


Fig: Nested if Statement

2.5 PHP Nested If

The Nested if contains the if block inside another if block. Here, the inner if statement executes only when specified condition in outer if statement is true. The syntax for the Nested if is:

```
if (condition) {
//code to be executed if condition is true
if (condition) {
//code to be executed if condition is true
}
}
```

Task: Write a program to check whether a person is eligible to vote or not? Write a program to find the smallest of three numbers.

2.6 Switch Statement in PHP

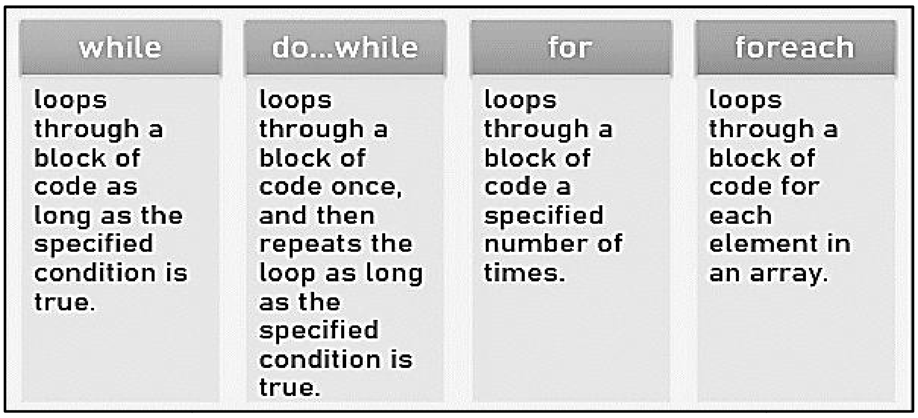
Switch statements work the same as if statements. However, the difference is that they can check for multiple values. Of course you do the same with multiple if..else statements, but this is not always the best approach.A switch statement allows a program to evaluate an expression and attempt to match the expression's value to a case label. If a match is found, the program executes the associated statement. One can use the switch statement to select one of many blocks of code to be executed. Syntax:

```
Switch (expression)
{
Case label1:
        code to be executed if expression = label1;
        break;
Case label2:
        code to be executed if expression = label2;
        break;
        default:
Code to be executed
        if expression is different
        from both label1 and label2;
        }
```

Task: Write a program to check the day of a week?

2.7 Loop Statements in PHP

Often when you write code, you want the same block of code to run over and over again a certain number of times. So, instead of adding several almost equal code-lines in a script, we can use loops. The loops are used to execute the same block of code again and again, as long as a certain condition is true.

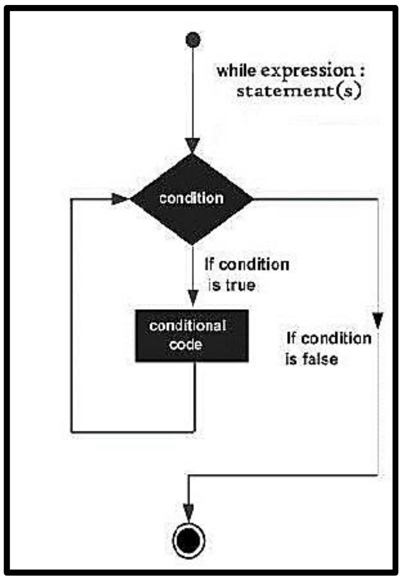


while	do...while	for	foreach
loops through a block of code as long as the specified condition is true.	loops through a block of code once, and then repeats the loop as long as the specified condition is true.	loops through a block of code a specified number of times.	loops through a block of code for each element in an array.

2.8 PHP while Loop

The while loop executes a block of code as long as the specified condition is true. In other words, while loop executes a block of code repeatedly until the condition is FALSE. Once the condition gets FALSE, it exits from the body of loop. The syntax for the while loop is:

```
While (condition is true)
{
Code to be executed;
}
```



While loop is also called as an Entry control loops because the condition is checked before entering the loop body. This means that first the condition is checked. If the condition is true, the block of code will be executed.

Notes: While loop should be used if the number of iterations is not known.

2.9 PHP Nested While Loop

The while loop inside another while loop in PHP, it is known as nested while loop.In case of inner or nested while loop, nested while loop is executed fully for one outer while loop. If outer while loop is to be executed for 3 times and nested while loop for 3 times, nested while loop will be executed 9 times (3 times for 1st outer loop, 3 times for 2nd outer loop and 3 times for 3rd outer loop).

2.10 PHP Infinite While Loop

If we pass TRUE in while loop, it will be an infinite loop. The syntax for an infinite loop is:

```
While(true)
{
//code to be executed
}
```

Example: The following example indicates an infinite loop.

```
<?php
  while (true) {
    echo "Hello Javatpoint!";
    echo "</br>";
  }
?>
```

2.11 PHP do-while Loop

PHP do-while loops through a block of code once, and then repeats the loop as long as the specified condition is true. The do-while loop will always execute the block of code once, it will then check the condition, and repeat the loop while the specified condition is true. The syntax for it is:

```
Do
{
Code to be executed;
}
While (condition is true);
```

Whilevs. do-while Loop

- Do-while loop is very much similar to the while loop except the condition check.
- The main difference between both loops is that while loop checks the condition at the beginning, whereas do-while loop checks the condition at the end of the loop.

The Table 1lists the difference between while and do-while loop.

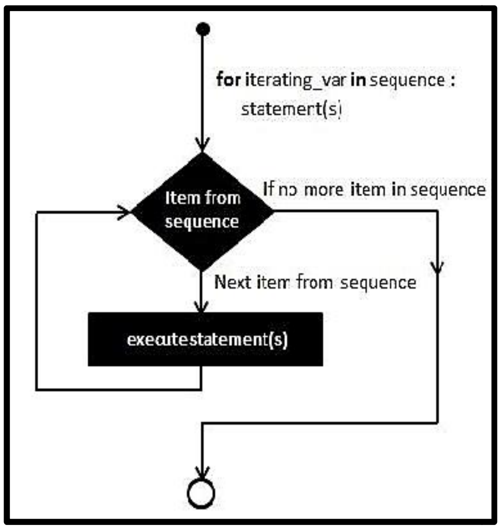
while loop	do while loop
The while loop is also named as entry control loop.	The do-while loop is also named as exit control loop.
The body of the loop does not execute if the condition is false.	The body of the loop executes at least once, even if the condition is false.
Condition checks first, and then block of statements executes.	Block of statements executes first and then condition checks.
This loop does not use a semicolon to terminate the loop.	Do-while loop use semicolon to terminate the loop.

Table 1: while vs. do-while Loop

2.12 PHP for Loop

PHP for loops through a block of code a specified number of times. The for loop is used when you know in advance how many times the script should run. The syntax is:

```
For (init counter; test counter; increment counter)
{
Code to be executed for each iteration;
}
```



Parameters:

- Init counter: Initialize the loop counter value.
- Test counter: Evaluated for each loop iteration. If it evaluates to TRUE, the loop continues. If it evaluates to FALSE, the loop ends.
- Increment counter: Increases the loop counter value.

Lab Exercise: Write a program to display the numbers 1 to 10?

2.13 PHP Infinite for Loop

All three parameters are optional, but semicolon (;) is must to pass in for loop. If we don't pass parameters, it will execute infinite.

```
<? php
  $i = 1;
  for (;;)                    //infinite loop
    {
echo $i++;
     echo "</br>";
  }
?>
```

2.14 PHP Nested for Loop

The for loop inside for loop in PHP is known as nested for loop. The inner for loop executes only when the outer for loop condition is found true.In case of inner or nested for loop, nested for loop is executed fully for one outer for loop. If outer for loop is to be executed for 3 times and inner for loop for 3 times, inner for loop will be executed 9 times (3 times for 1st outer loop, 3 times for 2nd outer loop and 3 times for 3rd outer loop).

2.15 PHP foreach Loop

The PHP for each loop is used to traverse array elements. The syntax used foreach loop is:

```
Foreach( $array as $var )
{
        //code to be executed
}
```

2.16 Embedding PHP in Web Pages

Although it is possible to write and run stand alone PHP programs, most PHP code is embedded in HTML or XML files. This is, after all, why it was created in the first place. The processing such documents involves replacing each chunk of PHP source code with the output it produces when executed.Because a single file contains PHP and non-PHP source code, we need a way to identify the regions of PHP code to be executed. PHP provides four different ways to do this:

- First, and preferred, method is XML.
- Second method is SGML.
- Third method is based on ASP tags.
- Fourth method uses the standard HTML <script> tag; this makes it easy to edit pages with enabled PHP using a regular HTML editor.

2.17 Embedding PHP in Web Pages- XML Style

XML Style: The way in which you normally embed PHP in a document is through what is commonly called a PHP statement, but is actually an XML directive. PHP is designed to be compatible with many standards and commonly used scripting approaches, making it easier to code. An XML processing directive is a command embedded in an XML document for some application to process.All XML processing directives take the form of:

```
<?appname information for application ?>
```

Notice that the statement opens and closes with nested brackets and question marks, and that the name of the application being referenced must come right after the first question mark, with no space before it. The information for the application can be a short statement, or, in the case of PHP, can go on for pages and pages.

PHP statement normally uses this form of inclusion in HTML documents. It looks like this:

```
<?php some phpstatements ?>
```

Line breaks are allowed within the XML processing directive, so the following is also legal.

```
<?php
        Somephp statements
        And some more statements
?>
```

If you want to write standards compliant code, or embed PHP in XML documents, you need to use this format. If you are not working with XML or don’t need to adhere to standards, there are other approaches. Each may be appropriate within certain server situations.Since PHP is processed by the server, it can literally go anywhere in an HTML document in question. The code is processed before the Web browser sees the code, so the Web browser never sees the PHP. If working with XML, you need to make sure that any server-side XML processing is done after the PHP processing, or code the document with the knowledge that the PHP code cannot violate the syntax rules of the application you are working with. This means that XML documents containing PHP should still be well-formed and valid.

2.18 Embedding PHP in Web Pages- SGML Style

SGML Style: The "classic" style of embedding PHP comes from SGML instruction processing tags. To use this method, simply enclose the PHP in <? and ?>. Here's the "Hello world" example again:

```
<? echo "Hello, world"; ?>
```

This style, known as short tags, is the shortest and least intrusive, and it can be turned off so as to not clash with the XML PI (Process Instruction) tag in the php.ini initialization file. Consequently, if you want to write fully portable PHP code that you are going to distribute to other people (who might have short tags turned off), you should use the longer <?php ... ?> style, which cannot be turned off. If you have no intention of distributing your code, you don't have an issue with telling people who want to use your code to turn on short tags, and you are not planning on mixing XML in with your PHP code, then using this tag style is okay.

2.19 Embedding PHP in Web Pages- ASP Style

ASP Style: Realizing that some old code editors might have problems with PHP, the authors of PHP included two additional ways to include PHP code in a document. One mimicks ASP tags. The other uses standard HTML <script> tags. The ASP tags are not enabled by default. To turn them on you need to find the asp tags flag in the php.ini file, or, if working from the source, build PHP with the —enable-asp-tags option.

It is not necessary for the server to actually process ASP tags as PHP tags, since the point of the compatibility with ASP tags is to allow users to work with editors that understand ASP tags but not PHP tags. When you are done with the document, you can just do a global search and replace to replace all ASP tags with PHP tags before moving the documents to production. The ASP tags take the following form:

```
<% code goes here %>
```

Like PHP tags, ASP tags can go anywhere in the document.

2.20 Embedding PHP in Web Pages- Script Style

Script Style: The final method of distinguishing PHP from HTML involves a tag invented to allow client-side scripting within HTML pages, the <script> tag. You might recognize it as the tag in which JavaScript is embedded. Since PHP is processed and removed from the file before it reaches the browser, you can use the <script> tag to surround PHP code. In order to use this method, simply specify "php" as the value of the language attribute of the tag:

```
<script language="php"> echo "Hello, world"; </script>
```

This method is most useful with HTML editors that work only on strictly legal HTML files and do not yet support XML processing commands.

Summary

- PHP is a server scripting language, and a powerful tool for making dynamic and interactive Web pages. It is widely used, free, and efficient alternative to competitors such as Microsoft's ASP.

- PHP is an open-source, interpreted, and object-oriented scripting language that can be executed at the server-side.
- Control statements are used in PHP to build logic and intelligence into the creation and deployment of web-based data.
- PHP if-else statement is executed whether condition is true or false. If-else statement is slightly different from if statement. It executes one block of code if the specified condition is true and another block of code if the condition is false.
- Else if statement is a conditional statement that is used with the if...else statement to execute a set of code if one of the several conditions is true.
- A switch statement allows a program to evaluate an expression and attempt to match the expression's value to a case label. If a match is found, the program executes the associated statement.
- Loop statements in PHP can be used if you want the same block of code to run over and over again a certain number of times.
- A do-while loop will always execute the block of code once, it will then check the condition, and repeat the loop while the specified condition is true.

Keywords

- ***Conditional statements:***Conditional statements are the set of commands used to perform different actions based on different conditions.
- ***Switch statement:***Switch statement is used if you want to select one of many blocks of code to be executed, use the Switch statement. The switch statement is used to avoid long blocks of if..else code.
- ***Compound If statement:***Compound If statements help to compound the statements using elseif to have multiple conditions tested insequence.
- ***PHP do-while Loop:*** PHP do-while loops through a block of code once, and then repeats the loop as long as the specified condition is true.
- ***XML Directive:*** An XML processing directive is a command embedded in an XML document for some application to process.

SelfAssessment

1. _____________ statements are the set of commands used to perform different actions based on different conditions.

A. Assignment
B. Operational
C. Conditional
D. Indictive

2. A ____________ statement allows a program to evaluate an expression and attempt to match the expression's value to a case label.

A. Case
B. Labelling
C. Levelling
D. Switch

3. _____________ statement is executed whether condition is true or false. It executes one block of code if the specified condition is true and another block of code if the condition is false.

A. Do-while
B. While
C. If-else
D. Switch

4. Two broad types of control structures are branching and

A. Looping
B. Structuring
C. Coding
D. None of the above

5. Which of the following is/are decision making statement(s) in PHP?

A. If...else statement
B. Else if
C. Switch
D. All of the above

6. ___________ statement loops through a block of code for each element in an array.

A. For
B. For each
C. Each
D. Each block

7. The loop that executes a block of code repeatedly until the condition is false. Once the condition gets false, it exits from the body of loop.

A. While
B. Switch
C. Case
D. If

8. ____________ loop is called as an entry control loop because the condition is checked before entering the loop body.

A. Do-while
B. If
C. For each
D. While

9. Which of the following loops uses semicolon to terminate the loop?

A. while

B. do-while loop
C. for
D. for each

10. Supposing that the developer has to develop a program and knows in advance how many times the script should run. The developer will be using
A. Switch
B. For each
C. For loop
D. None of the above

11. In the XML style, you normally embed PHP in a document through what is commonly called a PHP statement, but is actually an ____________.
A. XML header
B. XML footer
C. XML docs
D. XML directive

12. Line breaks are allowed within the XML processing directive.
A. True
B. False

13. ______________ tag has been invented to allow client-side scripting within HTML pages.
A. .aspx
B. .php
C. Script
D. Embed

14. SGML stands for
A. Simple Generalized Mark-up Language
B. Standard Generalized Mark-up Language
C. Simple General Machine Language
D. Standardized General Machine Language

15. An XML processing directive is a command embedded in an XML document for some application to process.
A. True
B. False

Answers forSelfAssessment

1.	C	2.	D	3.	C	4.	A	5.	D

6.	B	7.	A	8.	C	9.	B	10.	C
11.	D	12.	A	13.	C	14.	B	15.	A

Review Questions

1. Explain the following:
    a. XML Style
    b. SGML Style
2. What are ways of embedding PHP in web pages?
3. Explain the switch statement in PHP using examples?
4. What are control statements? Illustrate at least 2 with examples.
5. Differentiate while vs. do-while loop using examples.
6. What are the different conditional statements used in PHP?
7. Which different looping constructs are used in PHP? Elaborate using examples.
8. Write a short note on:
    a. If statement
    b. If-else statement
    c. Nested if
9. Indicate the working of a nested while and infinite while loop.
10. What is the significance of using control statements in PHP?

Further Readings

- Programming PHP: Creating Dynamic Web Pages by Kevin Tatroe and Peter MacIntyre.
- PHP Solutions by David Powers, Apress, 2007.
- PHP and MySQL Web Development by Luke Welling, Sams 2001.
- Programming PHP by RusmusLerdorf,Oreilly Publishers, 2013.

Web Links

- Control Statements In PHP (c-sharpcorner.com)
- Control Statement in PHP | Top 4 Control Statement in PHP with Syntax (educba.com)
- PHP: Control Structures - Manual
- PHP Control Structures and Loops: if, else, for, foreach, while, and More (tutsplus.com)
- PHP Control Structures: If else, Switch Case (guru99.com)
- https://wpshout.com/learning-php-for-wordpress-development-control-flow-basics/

Unit 03: Functions in PHP

CONTENTS

Objectives

After studying this unit, you will be able to:

- Understand built in functions of PHP.
- Understand more built-in functions of PHP.
- Analyse the practical implementation of certain built-in functions.
- Explore functions in PHP.
- Understand defining and calling user defined functions in PHP.
- Learn about managing parameters in user defined functions, user defined functions with return values.

Introduction

The real power of PHP comes from its functions. PHP function is a piece of code that can be reused many times. PHP functions are similar to other programming languages. A function is a piece of code which takes one more input in the form of parameter and does some processing and returns a value.

Advantages of Using Functions in PHP

- Code Reusability: PHP functions are defined only once and can be invoked many times, like in other programming languages.
- Less Code: It saves a lot of code because you don't need to write the logic many times. By the use of function, you can write the logic only once and reuse it.
- Easy to Understand: PHP functions separate the programming logic. So, it is easier to understand the flow of the application because every logic is divided in the form of functions.
- Using functions, you can create a single function, with one if statement that can be used for each blank string you need to check.
- Using a function means there's less code for you to write; and it's more efficient.

Defining a Function

A function is just a segment of code, separate from the rest of your code. You separate it because it's nice and handy, and you want to use it not once but over and over. It is a chunk of code that you think is useful, and want to use again. The functions save you from writing the code over and over.

3.1 PHP Built-in Functions

PHP has more than 1000 built-in functions, and in addition you can create your own custom functions. There are built-in functions but PHP gives you option to create your own functions as well.

PHP Array Functions	PHP OpenSSL Functions
PHP bzip2 Functions	PHP Password Functions
PHP Calendar Functions	PHP Queue Functions
PHP Class/Object Functions	PHP Sequence Functions
PHP Character Functions	PHP Session Functions
PHP Date & Time Functions	PHP String Functions
PHP Deque Functions	PHP Thread Functions
PHP Directory Functions	PHP Threaded Functions
PHP Error Handling Functions	PHP Tokenizer Functions
PHP File System Functions	PHP URL's Functions
PHP Function Handling Functions	PHP XML Functions
PHP MySQL Functions	PHP XML Parsing Functions
PHP Network Functions	PHP XML Reader Functions
PHP ODBC Functions	PHP XML Writer Functions

3.2 PHP Array Functions

PHP array is an ordered map (contains value on the basis of key). It is used to hold multiple values of similar type in a single variable. There is large number of array functions in PHP (Table 1).

- *array () Function:*The array() function is used to create a PHP array. This function can be used to create indexed arrays or associative arrays. PHP arrays could be single dimensional or multi-dimensional.This function was first introduced as part of core PHP 4.0.0. The syntax to create PHP indexed arrays:

```
$a = array (value1, value2, value3, ...)
```

The syntax to create PHP associative arrays is:

```
$a = array(key1 => value1, key2 => value2...)
```

Where,

- Key (Optional): Specifies the key, of type numeric or string. If not set, an integer key is generated, starting at 0.
- Value (Required): Specifies the value.
  - Creating an empty PHP array

```
<?php
$cars = array();     //Array function
print_r($cars); //print_r() prints information about a variable in a more human-readable way
?>
```

- *Creating an Indexed array*

```
<? php
  $cars = array("Mercedes", "BMW", "Chevrolet");
print_r($cars);
?>
```

Array Functions	Description
array()	Creates an array
array_change_key_case()	Changes all keys in an array to lowercase or uppercase
array_chunk()	Splits an array into chunks of arrays
array_column()	Returns the values from a single column in the input array
array_combine()	Creates an array by using the elements from one "keys" array and one "values" array
array_count_values()	Counts all the values of an array
array_diff()	Compare arrays, and returns the differences (compare values only)
array_diff_assoc()	Compare arrays, and returns the differences (compare keys and values)
array_diff_key()	Compare arrays, and returns the differences (compare keys only)
array_diff_uassoc()	Compare arrays, and returns the differences (compare keys and values, using a user-defined key comparison function)
array_diff_ukey()	Compare arrays, and returns the differences (compare keys only,

	using a user-defined key comparison function)
array_fill()	Fills an array with values
array_fill_keys()	Fills an array with values, specifying keys
array_flip()	Flips/Exchanges all keys with their associated values in an array
array_intersect()	Compare arrays, and returns the matches (compare values only)
array_intersect_assoc()	Compare arrays and returns the matches (compare keys and values)
array_intersect_key()	Compare arrays, and returns the matches (compare keys only)
array_intersect_uassoc()	Compare arrays, and returns the matches (compare keys and values, using a user-defined key comparison function)
array_intersect_ukey()	Compare arrays, and returns the matches (compare keys only, using a user-defined key comparison function)
array_key_exists()	Checks if the specified key exists in the array
array_keys()	Returns all the keys of an array
array_map()	Sends each value of an array to a user-made function, which returns new values
array_merge()	Merges one or more arrays into one array
array_merge_recursive()	Merges one or more arrays into one array recursively
array_multisort()	Sorts multiple or multi-dimensionalarrays
array_pad()	Inserts a specified number of items, with a specified value, to an array
array_pop()	Deletes the last element of an array
array_product()	Calculates the product of the values in an array
array_push()	Inserts one or more elements to the end of an array
array_rand()	Returns one or more random keys from an array
array_reduce()	Returns an array as a string, using a user-defined function
array_replace()	Replaces the values of the first array with the values from following arrays
array_replace_recursive()	Replaces the values of the first array with the values from following arrays recursively
array_reverse()	Returns an array in the reverse order

Table 1: Different Types of Array Functions

- *PHP array_change_key_case() Function:* This function changes case of all key of an array. Noteably, it changes case of key only. It changes the case of all keys of the passed array and returns an array with all keys either in lower case or upper case based on the option passed. The syntax for it is:

```
array array_change_key_case (array $array [,int $case= CASE_LOWER])
```

Example:

```
<?php
   $cars=array("BMW"=>"550000","MERCEDES"=>"250000","CHEVROLET"=>"200000");
print_r(array_change_key_case($cars,CASE_LOWER));
?>
```

- *PHP array_chunk() Function:* By using array_chunk() method, you can divide array into many parts. The array_chunk() function takes an array as input and split that array into smaller chunks of the given size. The last chunk may contain a smaller number of elements than passed size based on the multiplicity factor of the total numbers available in the array. It returns a multidimensional numerically indexed array, starting with zero, with each dimension containing size elements. The syntax is:

```
array array_chunk (array $input, int $size);
```

Example:

```
<?php
$cars=array("BMW"=>"550000","MERCEDES"=>"250000","CHEVROLET"=>"200000“,
"Maruti"=>"50000");
print_r(array_chunk($cars,2));
?>
```

- *PHP count () Function:* Counts all elements in an array. The syntax is:

```
int count(mixed $array_or_countable);
```

Example: <?php

```
$courses=array("Data Structures","PHP","Python","Software Engineering", "Algorithm
Designing", "C Programming", "Web Programming");
echo count($courses); // Will display the count of the elements in an array
?>
```

- *PHP sort () Function:* Sorts all the elements in an array. The syntax is,

```
bool sort( $array );
```

Example: The example below illustrates the use of sort() function.

```
<?php
   $courses=array("Data Structures","PHP","Python","Software Engineering", "Algorithm
Designing", "C Programming", "Web Programming");
   sort($courses);
   foreach( $courses as $c) {
   echo "$c<br />"; }
   ?>
```

The example below illustrates the use of sort() function.

```
<?php
$courses=array("Data Structures","PHP","Python","Software Engineering", "Algorithm Designing", "C Programming", "Web Programming");
sort($courses);
foreach( $courses as $c) {
    echo "$c<br />";  }
?>
```

- *PHP array_reverse() Function:* Returns an array containing elements in reversed order. The syntax is:

  array array_reverse (array $array);

Example:

```
<?php
$cars=array("BMW","MERCEDES","PAJERO","DUSTER", "ECOSPORT");
$reversecars=array_reverse($cars);
foreach($cars as $c)  {
echo "$c <br />";  }
echo "<br/>";
foreach( $reversecars as $rc ) {
echo "$rc<br />";  }    ?>
```

- *PHP array_search() Function:* Searches the specified value in an array. It returns key if search is successful. It has the following syntax:

  array_search(item, array $arrayname);

Example:

```
<?php
$cars=array("BMW","MERCEDES","PAJERO","DUSTER", "ECOSPORT");
$key=array_search("PAJERO",$cars);
echo $key;
?>
```

3.3 PHP Calendar Functions

Calendar extension presents a series of functions to simplify converting between different calendar formats.The intermediaries or standard it is based on is the Julian Day Count. The Julian Day Count is a count of days starting from January 1st, 4713 B.C. In order to convert between calendar systems, you must first convert to Julian Day Count, then to the calendar system of your choice.

- *cal_days_in_month():* Returns the number of days in a month for a specified year and calendar. The syntax is:

  cal_days_in_month ($calendar, $month, $year);

Example:

```
<?php
$num = cal_days_in_month(CAL_GREGORIAN, 2, 2007);
echo "There was $num days in Jan 2007";
?>
```

- *cal_from_jd():* Converts a Julian day count into a date of a specified calendar. The syntax is:cal_from_jd ($jd, $calendar);

//This function converts the Julian day given in jd into a date of the specified calendar.

Return Value: It returns an array containing calendar information like month, day, year, day of week, abbreviated and full names of weekday and month and the date in string form "month/day/year".

Example:

```
<?php
$input = unixtojd(mktime(0, 0, 0, 8, 16, 2016));
print_r(cal_from_jd($input, CAL_GREGORIAN));
?>
```

Output:

```
Array (  [date] > 8/16/2016   [month] > 8   [day] > 16   [year] > 2016   [dow] > 2   [abbrevdayname] >
Tue   [dayname] > Tuesday   [abbrevmonth] > Aug   [monthname] > August)
```

- ***cal_to_jd():*** Converts a date to Julian day count. The syntax is:

  cal_to_jd ($calendar, $month, $day, $year);

Return Value: The length in days of the selected month in the given calendar.Example:

```
<?php
$d = cal_to_jd(CAL_GREGORIAN,11,03,2007);
echo $d;
?>
```

- *easter_days():* Returns the number of days after March 21, on which Easter falls for a specified year. If no year is specified, the current year is assumed. The syntax is:

  easter_days ([$year [, $method]]);

Example:

```
<?php
  echo easter_days()."\n <br/>";  //Easter for year 2022 falls on 17th April
  echo easter_days(2021)."\n <br/>";
  echo easter_days(1913);
?>
```

3.4 PHP Class/Object Functions

Such functions allow you to obtain information about classes and instance objects. You can obtain the name of the class to which an object belongs, as well as its member properties and methods. The objects are instances of programmer-defined classes, which can package up both other kinds of values and functions that are specific to the class. The classes and objects are the two main aspects of object-oriented programming. A class is a template for objects, and an object is an instance of a class.When the individual objects are created, they inherit all the properties and behaviors from the class, but each object will have different values for the properties.

- *class_exists():* Checks if the given class have been defined. Returns TRUE if class_name is a defined class, FALSE otherwise. The syntax is:

  class_exists ($class_name);

Return Value: Returns TRUE if class_name is a defined class, FALSE otherwise.

Example:

```
<?php
if (class_exists('HelloWorld')) {
$helloworld = new HelloWorld();  }
?>
```

- *get_class_methods():* Gets the class methods names. Returns an array of method names defined for the class specified by class_name. In case of an error, it returns NULL. The syntax is:get_method ($class_name);

Return Value: Returns an array of method names defined for the class specified by class_name. In case of an error, it returns NULL.

- *get_class():* Gets the name of the class of the given object. The syntax is: get_class ($object);

Return Value: Returns the name of the class of which object is an instance. Returns FALSE if object is not an object.

3.5 PHP Character Functions

The functions provided by this extension check whether a character or string falls into a certain character class according to the current locale.When called with an integer argument these functions behave exactly like their C counterparts fromctype.

Installation: Beginning with PHP 4.2.0 these functions are enabled by default.

- ctype_alnum(): Checks if all of the characters in the provided string, text, are alphanumeric.

Syntax: ctype_alnum ($text);

Return Value: Returns TRUE if every character in text is either a letter or a digit, FALSE otherwise.

Example: The example below indicates the use of the function:

```
<?php
  $cars = array('Mercedes', 'swift!!#$car');
  foreach ($cars as $cr)
  {
    if (ctype_alnum($cr))
    {
      echo "$cr consists of all letters or digits. \n <br/>";
    }
    else {
```

```
                echo "$cr does not have all letters or digits. \n";
              }
            }
    ?>
```

- ctype_alpha(): Checks if all of the characters in the provided string, text, are alphabetic.

Syntax: ctype_alpha ($text);

Return Value: Returns TRUE if every character in text is a letter, FALSE otherwise.

Example: The following example showcases the use of the function:

```
<?php
  $cars = array('Mercedes', 'swift!!#$car');
 foreach ($cars as $crs) {
    if (ctype_alpha($crs)){
      echo "$crs consists of all letters and is alphabetic only. \n";
    }
    else {
      echo "$crs does not have all letters. \n";}}
?>
```

- ctype_digit(): Checks if all of the characters in the provided string, text, are numerical. It checks only 1...9.

Syntax: ctype_digit ($text);

Return Value: Returns TRUE if every character in text is a decimal digit, FALSE otherwise.

Example: The following example showcases the use of the function:

```
<?php
  $cars = array('Mercedes123', 'swift!!#$car', '123', "1345");
  foreach ($cars as $num) {
    if (ctype_digit($num)) {
      echo "$num consists of all digits. \n <br/>";}
    else {
      echo "$num does not have all digits. \n <br/>";}}
?>
```

- ctype_lower(): Checks if all of the characters in the provided string, text, are lowercase letters.

Syntax: ctype_lower ($text);

Return Value: Returns TRUE if every character in text is a lowercase letter in the current locale.

Example: The following example showcases the use of the function:

```
<?php
  $cars = array('Mercedes123', 'swift!!#$car', 'TARAN', "DESIgnER", "taran");
 foreach ($cars as $crs) {
```

```
                    if (ctype_lower($crs))
                    {
                      echo "$crs consists of all lowercase letters. \n <br/>";
                    }
                    else {
                      echo "$crs does not have all lowercase letters. \n <br>";
                    } }
        ?>
```

- ctype_upper(): Checks if all of the characters in the provided string, text, are uppercase characters.

Syntax: ctype_upper ($text);

Return Value: Returns TRUE if every character in text is an uppercase letter in the current locale.

Example: The following example indicates the use of the function:

```
                <?php
                 $cars = array('Mercedes123', 'swift!!#$car', 'TARAN', "DESIgnER", "taran");
                foreach ($cars as $crs) {
                    if (ctype_upper($crs))
                    {
                      echo "$crs consists of all uppercase letters. \n <br/>";
                    }
                    else {
                      echo "$crs does not have all uppercase letters. \n <br>";
                    } }
        ?>
```

- ctype_print(): Checks if all of the characters in the provided string, text, are printable.

Syntax: ctype_print ($text);

Return Value: Returns TRUE if every character in text will actually create output (including blanks). Returns FALSE if text contains control characters or characters that do not have any output or control function at all.

Example: The following example indicates the use of the function:

```
                <?php
                  $cars = array('Mercedes123', 'swift!!#$car', '123', "1345");
                  foreach ($cars as $num){
                      if (ctype_print($num)) {
                      echo "$num consists of all printable characters. \n <br>";
                   }
                    else {
                      echo "$num does not have all printable characters. \n";}}
                ?>
```

3.6 PHP Date & Time Functions

The date and time functions allow you to get the date and time from the server where your PHP scripts are running. You can use these functions to format the date and time in many different ways.

checkdate() Function*:* It accepts the month, day, year of a date as parameters and, verifies whether it is a Gregorian date or not. This function was first introduced in PHP Version 4 and works with all the later versions.

Syntax: checkdate (int $month, int $day, int $year)

Return Values: Returns a boolean value. This value is true if the given date is valid and, false if it is invalid.

Example: The following examples indicates the use of this function:

```
<?php
// PHP program to demonstrate the checkdate() function
$month = 12;
$day = 31;
$year = 2022;
// returns a boolean value after validation of date
var_dump(checkdate($month, $day, $year));
?>
<?php
$month = 12;
$day = 36;                    //Output- bool(false)
$year = 2022;
// returns a boolean value after validation of date
var_dump(checkdate($month, $day, $year));
?>
```

Example: The program below checks if the date is a valid one or not in case of a leap year and non-leap year.

```
<?php
// PHP program to demonstrate the checkdate() function in case of leap year
$month = 2;
$day = 29;
$year = 2020;
// returns a boolean value after validation of date leap year
var_dump(checkdate($month, $day, $year));
echo "<br/>";
$month = 2;
$day = 29;
$year = 2022;
// returns a boolean value after validation of date non-leap year
var_dump(checkdate($month, $day, $year));
```

```
?>
```

date_create() Function*:* This function is used to create a DateTime object by using a date/time string and timezone. The default value of the date/time string is current date/time. An alias of the DateTime::__construct, a constructor of the DateTime class. Where, a DateTime class represents date and time in PHP. It accepts a date time string and time zone (optional) as parameters and, creates a DateTime object accordingly.This function was first introduced in PHP Version 5.2.0 and, works with all the later versions.

Syntax: date_create([$date_time, $timezone]);

Return Values: Returns the created DateTime object which specifies a date.

date_format() Function*:* An alias of DateTime::format() function. It accepts a DateTime object and a format string (representing a desired date/time format) as parameters, formats the object in the specified format and, returns the result. This function was first introduced in PHP Version 5.2.1 and works in all the later versions.

Syntax: date_format($date_time_object, $format)

Return Values: Returns the formatted date string.

Here, we are creating a DateTime object and formatting it.

```
<?php
  //Creating a DateTime object
  $dob = date_create("16-02-2022");
  //formatting the date/time object
  $fmt = date_format($dob, "y-d-m");
print("Date in yy-dd-mm format: ".$fmt);
?>
```

This will produce following result:

```
Date in yy-dd-mm format: 22-16-02
```

date_date_set() Function*:* An alias of the DateTime::setDate(). Using this, you can (re)set the date of a DateTime object. This function was first introduced in PHP Version 5.2.0 and, works with all the later versions.

Syntax: date_date_set($object, $year, $month, $day)

Return Values: Returns DateTime object with modified value. Incase of the failure, this function returns boolean value false.

Example: The program demonstrates the usage of date_date_set function.

Example:

```
<?php
 //Creating a date
 $date = new DateTime();
//Setting the date
date_date_set($date, 2022, 10, 02);
print("Date: ".date_format($date, "Y/m/d"));
?>
```

Output:

Date: 2022/10/02

date_modify(): An alias of DateTime::modify(). This function is used to modify the date in a DateTime object. It alters the time stamp of the given object. This function was first introduced in PHP Version 5.2.0 and, works with all the later versions.

Syntax: date_modify($object, $modify)

Return Values: Returns the DateTime object with modified value. Incase of the failure, this function returns the boolean value false.

Example:

```
          <?php
            //Modifying the date
            $date = date_modify(new DateTime(), "+15 day");
            //Will add 15 days to the current system date
          print("Date: ".date_format($date, "Y/m/d"));
?>
```

Output:

Date: 2022/03/22

- date() function: Accepts a format string as a parameter, formats the local date/time in the specified format and returns the result. This function was first introduced in PHP Version 4 and, works with all the later versions.

Syntax: date ($format, $timestamp)

Return Values: Returns the current local time/date in the specified format.

Example:

```
<?php
$day =date('d/m/y');
echo $day; //presents the current date
?>
```

3.7 PHP Error Handling Functions

Error is the fault or mistake in a program. It can be several types. An error can occur due to wrong syntax or wrong logic. The type of mistakes or condition of having incorrect knowledge of the code. Also, creating a custom error handler in PHP is quite simple. We can also create a function that can be called when an error has been occurred in PHP. The various types of errors in PHP but it contains basically four main types of errors. These are:

Parse error or Syntax Error: It is the type of error done by the programmer in the source code of the program. The syntax error is caught by the compiler. After fixing the syntax error the compiler compiles the code and execute it. Parse errors can be caused dues to unclosed quotes, missing or Extra parentheses, Unclosed braces, Missing semicolon etc.Example:

Example:

```
<?php
        $x = "geeks";
        y = "Computer science";
```

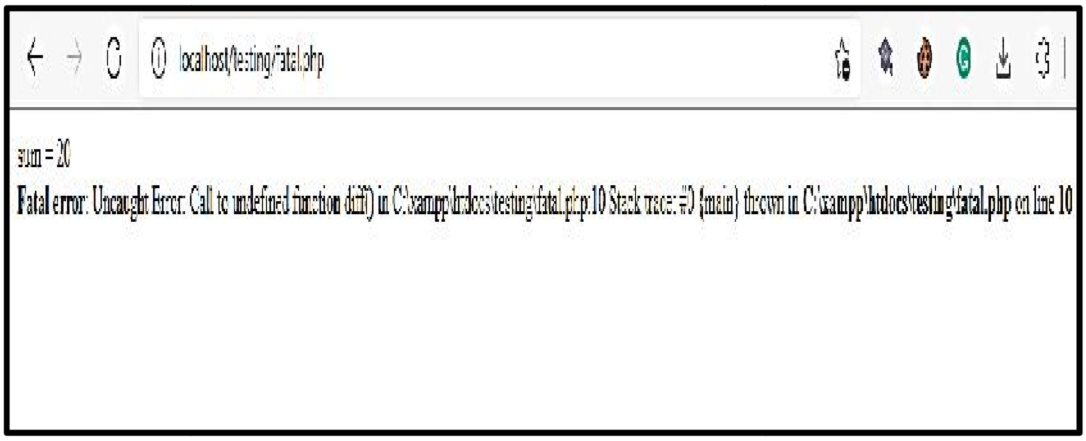
```
//Error: In above program, $ sign is missing in line 3 so it gives an error message
echo $x;
echo $y;
?>
```

Fatal Error: Type of error where PHP compiler understand the PHP code but it recognizes an undeclared function. This means that function is called without the definition of function.

Example:

```
<?php
        function add($x, $y)
        {
          $sum = $x + $y;
          echo "sum = ". $sum;
        }
        $x = 0;
        $y = 20;
        add($x, $y);
        diff($x, $y);
        ?>
```

Error: In line 12, function is called but the definition of function is not available. So, it gives error.



```
localhost/testing/fatal.php

sum = 20
Fatal error: Uncaught Error: Call to undefined function diff() in C:\xampp\htdocs\testing\fatal.php:10 Stack trace: #0 {main} thrown in C:\xampp\htdocs\testing\fatal.php on line 10
```

Warning Errors: The main reason of warning errors is including a missing file. This means that the PHP function call the missing file. Example:

Example:

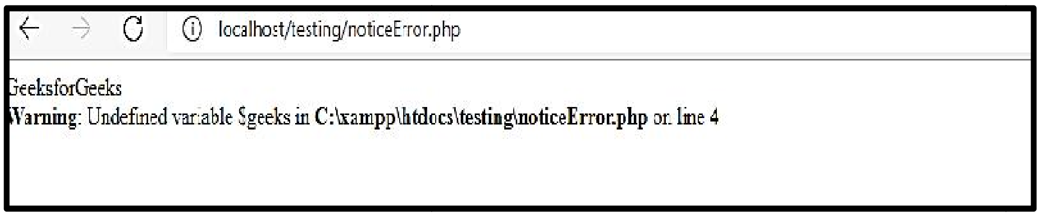
```
<?php
$x = "GeeksforGeeks";
include ("gfg.php");
echo $x . "Computer science portal";
?>
```

Error: This program calls an undefined file gfg.php which are not available. So, it produces error.

Notice Error: It is similar to warning error. It means that the program contains something wrong but it allows the execution of script. Example:

Example:

```
<?php
$x = "GeeksforGeeks";
echo $x;
echo $geeks;
?>
```



```
localhost/testing/noticeError.php

GeeksforGeeks
Warning: Undefined variable $geeks in C:\xampp\htdocs\testing\noticeError.php on line 4
```

Ways to handle PHP Errors:

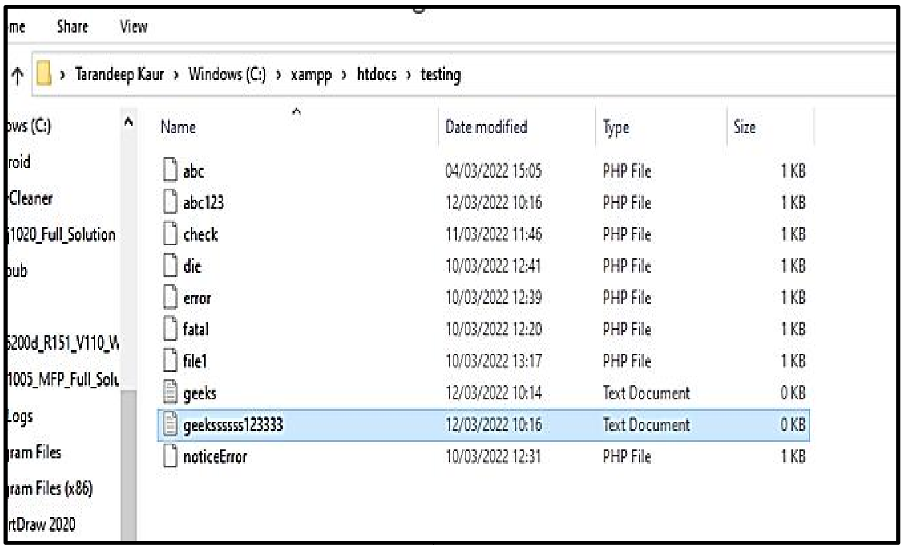
- Using die() method
- Custom Error Handling

1. Using die () function: die () function prints a message and exit from current script.

Syntax: die ($message) //will display a run-time error message.

Example:

```
<?php
// Php code showing default error handling
$file = fopen("geek12333.txt", "w");
?>
```



Tarandeep Kaur › Windows (C:) › xampp › htdocs › testing

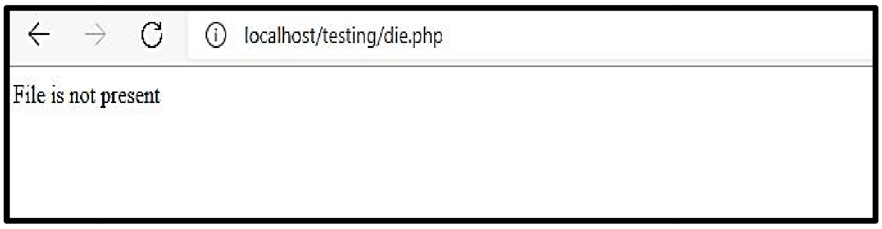
Name	Date modified	Type	Size
abc	04/03/2022 15:05	PHP File	1 KB
abc123	12/03/2022 10:16	PHP File	1 KB
check	11/03/2022 11:46	PHP File	1 KB
die	10/03/2022 12:41	PHP File	1 KB
error	10/03/2022 12:39	PHP File	1 KB
fatal	10/03/2022 12:20	PHP File	1 KB
file1	10/03/2022 13:17	PHP File	1 KB
geeks	12/03/2022 10:14	Text Document	0 KB
geekssssss123333	12/03/2022 10:16	Text Document	0 KB
noticeError	10/03/2022 12:31	PHP File	1 KB

Notes: Run the above code and geeks.txt file is not present then it will display a run-time error message.

Example:

```
<?php        // PHP code to check errors
```

```
// If file is not present, then exit from script
if( !file_exists("geeks.txt") )
{
die("File is not present");
}
// If file is present then continue
else {
    $file = fopen("geeks.txt", "w");
    }
?>
```



localhost/testing/die.php

```
File is not present
```

3.8 PHP File System Functions

basename( ) Function*:* An inbuilt function which is used to return the base name of a file if the path of the file is provided as a parameter to the basename() function.

Syntax: string basename ($path, $suffix)

Return Value: This function returns the basename of the file whose path has been given as a parameter by the user.

```
<?php
$path = “xampp/htdocs/testing/error.php";
// basename() function to show filename along with extension
echo basename($path);
?>
```

chmod( ) Function*:* Used to change the mode of a specified file to a specific mode given by the user.Thechmod() function changes the permissions of the specified file and returns true on success and false on failure.

Syntax: bool chmod (string $filename, int $mode)

The chmod() function in PHP accepts two parameters which are filename and mode.

$filename: It specifies the file whose permissions need to be changed.

$mode: It is used to specify the new permissions. The $mode parameters consist of four numeric values where the first value is always zero, the second value specifies permissions for the owner, the third value specifies permissions for the owner’s user group and the fourth value specifies permissions for everybody else.

There are three possible values and to set multiple permissions the following values can be added.

1 = execute permissions

2 = write permissions

4 = read permissions

file_exists( ) Function*:* An inbuilt function which is used to check whether a file or directory exists or not.The path of the file or directory you want to check is passed as a parameter to the file_exists() function which returns True on success and False on failure.

Syntax: file_exists($path)

Parameters: The file_exists() function in PHP accepts only one parameter $path. It specifies the path of the file or directory you want to check.

Return Value: It returns true on success and False on failure.

3.9 PHP Math Functions

abs() Function: An inbuilt function in PHP which is used to return the absolute (positive) value of a number. The abs() function in PHP is identical to what we call modulus in mathematics. The modulus or absolute value of a negative number is positive.

Syntax: number abs(value)

Parameters: The abs() function accepts single parameter value which holds the number whose absolute value you want to find.

Return Value: It returns the absolute value of the number passed to it as argument.

```
<?php
echo (abs(-6)); //Output: 6
?>
```

ceil() Function*:* Ceil() function rounds fractions up.

Syntax: float ceil (float $value)

Example:

```
<?php
echo (ceil(3.3)."<br/>");// 4
echo (ceil(7.333)."<br/>");// 8
echo (ceil(-4.8)."<br/>");// -4
?>
```

floor() Function*:* floor() function rounds fractions down.

Syntax: float floor (float $value)

Example:

```
    <?php
    echo (floor(3.3)."<br/>");// 3
    echo (floor(7.333)."<br/>");// 7
    echo (floor(-4.8)."<br/>");// -5
?>
```

sqrt() function: sqrt() function returns square root of given argument.

Syntax: float sqrt (float $arg)

Example:

```
<?php
```

```
        echo (sqrt(16)."<br/>");  // 4
        echo (sqrt(25)."<br/>");  // 5
        echo (sqrt(7)."<br/>");  // 2.6457513110646
?>
```

3.10 PHP MySQL Functions

MySQL is an open-source relational database management system (RDBMS). It is the most popular database system used with PHP. MySQL is developed, distributed, and supported by Oracle corporation. The data in a MySQL database are stored in tables which consists of columns and rows.

- MySQL is a database system that runs on a server.
- MySQL is ideal for both small and large applications.
- MySQL is very fast, reliable, and easy to use database system.It uses standard SQL
- MySQL compiles on a number of platforms.
- Various database function such as connect (), close (), ping () etc.

3.11 PHP Set Functions

Set is a sequence of unique values, and this implementation can use the same hash table as Ds\Map where the values are used as keys, and the mapped value is ignored.

Strengths:

- Values can be any type, including objects.
- Supports array syntax (square brackets).
- Insertion order is preserved.
- Automatically frees allocated memory when its size drops low enough.
- Add (), remove() and contains() are all O(1).

3.12 PHP Session Functions

PHP Sessions functions allow you to access data across multiple pages in the same session. The functions such as session_start, session_abort, session_status, session_cache_expire, session_commit, session_decode, session_destroy, session_name, session_reset etc.

3.13 PHP String Functions

chr() Function*:* Returns a character from the specified ASCII value.The ASCII value can be specified in decimal, octal, or hex values. The octal values are defined by a leading 0, while hex values are defined by a leading 0x.

Syntax: chr(ascii)

The example below illustrates the use of char() function.

```
<?php
echo chr(65) . "<br>";
echo chr(062) . "<br>";
echo chr(0x75) . "<br>";
?>
```

strlen() Function:This function returns the length of the string or the number of characters in the string including whitespaces.

Syntax: strlen($str)

Example:

The example below illustrates the use of strlen() function.

```
        <?php
        $s = "Good Day Ahead";
        // use of strlen
        echo "Total string length is: ". strlen($s);
?>
```

- *trim() Functions:* Removes whitespace and other predefined characters from both sides of a string.
  - ltrim()- Removes whitespace or other predefined characters from the left side of a string.
  - rtrim()- Removes whitespace or other predefined characters from the right side of a string.

3.14 PHP URL Functions

Such functions deal with URL strings: encoding, decoding, and parsing.

- get headers() Function- This function can fetch all headers sent by the server in response to an HTTP request.
- get_meta_tags() Function: This function can extract all meta tag content attributes from a file and can return an array.

3.15 PHP XML Functions

The SimpleXML extension functions provides the toolset to convert XML to an object. These objects deals with normal property selectors and array iterators.

PHP XML Reader Functions: XMLReader extension is used to read/retrieve the contents of an XML document i.e. using the methods of the XMLReader class you can read each node of an XML document.

- XMLReader::close----This function is used to close the current XMLReader object.
- XMLReader::getAttribute----This function is used to retrieve the value of a named attribute.
- XMLReader::isValid----This Is used to determine whether the parsed XML document is valid.

PHP XML Writer Functions: XMLWriter extension internally has libxmlxmlWriter API and is used to write/create the contents of an XML document. The XML documents generated by this are non-cached and forward-only.

3.16 PHP User-defined Functions

We can declare and call user-defined functions easily. The syntax is:

```
function functionname()
{
        //code to be executed
}
```

Notes: Function name must be start with letter and underscore only like other labels in PHP. It can't be start with numbers or special symbols. Function names are NOT case-sensitive.

There are two parts of making use of functions:

- Creating a PHP Function
- Calling a PHP Function

Did you know?

You hardly need to create your own PHP function because there are already more than 1000 of built-in library functions created for different area and you just need to call them according to your requirement.

3.17 Creating and calling a PHP Function

It is very easy task to create your own PHP function. Suppose you want to create a PHP function which will simply write a simple message on your browser when you will call it.

Notes: While creating a function its name should start with keyword function and all the PHP code should be put inside {and} braces

Example: The following example creates a function called RunningPHP() and then calls it just after creating it.

```
        <?php
           /* Defining a PHP Function */
             function RunningPHP()
            {
               echo "Learning web development using PHP";
             }
             /* Calling a PHP Function */
        RunningPHP();
?>
```

Lab Exercise: Write a program to print a number using if and for condition in Functions? Write a program to find the sum of five numbers? Write a program to find the factorial of a number using Functions in PHP?

3.18 PHP User-defined Functions with Parameters

PHP gives you option to pass your parameters inside a function. You can pass as many as parameters your like. These parameters work like variables inside your function. We can pass the information in PHP function through arguments which is separated by comma.An argument is just like a variable. The arguments are specified after the function name, inside the parentheses. You can add as many arguments as you want, just separate them with a comma.PHP supports call by value (default), call by reference, default argument values and variable-length argument list.

PHP Call by Value

- ***Passing Single Parameter***

Example: The following example illustrates the creation of a user-defined function with parameters:

```
          <?php
          //A program to verify if a number is odd or even number
          function _verifyNumber($num)
          {
           if ($num%2==0)
                    echo "The entered number $num is an even number";
          else
                    echo "The entered number $num is an odd number";
          }
          _verifyNumber(15);
?>
```

Task: Write a program to illustrate the use of arithmetic operators by passing parameters to a function?

```
<?php
     function using Func($num1, $num2) {
       $sum = $num1 + $num2;
       echo "Sum of the two numbers is: $sum";
       echo "<br/>";
       $sub= $num1-$num2;
       echo "The difference of the two numbers is: $sub";
       echo "<br/>";
$mul= $num1*$num2;
       echo "The multiplication of the two numbers is: $mul";
       echo "<br/>";
       $div= $num1/$num2;
       echo "The division of the two numbers is: $div";
     }
usingFunc(100, 50);
?>
```

Lab Exercise: Write a program to find the factorial of a number by passing parameters?

- ***Passing Multiple Parameters***

The following example illustrates the creation of user-defined functions and passing multiple parameters.

Example:

```
<?php
        function StudentRecord($name,$age,$course)
        {
        echo "The student $name is $age years old and is taking $course course<br/>";
        }
        StudentRecord("Taran",35, "Data Structures");
        StudentRecord("Karan",37, "Data Communications");
        StudentRecord("Gurfateh",40, "Wireless Networks");
?>
```

PHP Call by Reference

The value passed to the function doesn't modify the actual value by default (call by value). But we can do so by-passing value as a reference.By default, the value passed to the function is call by value. In order to pass value as a reference, you need to use ampersand (&) symbol before the argument name.

Example: Write a program to perform string concatenation using call by reference?

```
<?php
function concat(&$string2)
{
   $string2= $string2.'World';
}
$string1 = 'Web Development using PHP ';
concat($string1);
echo $string1;
?>
<?php
        function addFive($num)
        {
           $num += 15;
        }
        function addSix(&$num)
        {
           $num += 36;
        }
        $actualnum = 16;
addFive($actualnum);
        echo "The actual value is: $actualnum<br/>";
addSix($actualnum);
        echo "The actual value is: $actualnum<br/>";
?>
```

Default Argument Values

We can specify a default argument value in function. While calling PHP function if you don't specify any argument, it will take the default argument.

Example:

```
<?php
function PHPLearn($course="PHP")
  {
        echo "Learning $course<br/>";
   }
        PHPLearn("Software Engineering");
        PHPLearn();        //passing no value
        PHPLearn("Python");
?>
```

PHP User-defined Functions with Return Value

A function can return a value using the return statement in conjunction with a value or object. The return stops the execution of the function and sends the value back to the calling code.You can return more than one value from a function using return array (1, 2, 3, and 4).

Notes: The return keyword is used to return a value from a function.

PHP Function: Returning Value

Example:

```
<?php
function square($num)
{
return $num*$num;
}
echo "Square of the number is: " .square(7);
?>
```

Example: Following example takes two integer parameters and add them together and then returns their sum to the calling program.

```
<?php
    function addFunction($num1, $num2)
   {
      $sum = $num1 + $num2;
      return $sum;
    }
    $return_value = addFunction(15, 60);
    echo "Returned value from the function : $return_value";
```

```
?>
```

Task: Write a program to find the factorial of a number using user-defined function with return value?

```
<?php
        function factorial($n)
        {
          if ($n < 0)
            return -1; /*Wrong value*/
          if ($n == 0)
            return 1; /*Terminating condition*/
          return ($n * factorial ($n -1));
        }
        echo factorial(5);
?>
```

Variable Length Parameters

PHP supports variable length argument function. It means you can pass 0, 1 or n number of arguments in function. To do so, you need to use 3 ellipses (dots) before the argument name.The3 dot concept is implemented for variable length argument since PHP 5.6.

Example:

```
        <?php
        function add(...$numbers)
        {
          $sum = 0;
          foreach ($numbers as $num)
          {
            $sum= $sum+$num;
          }
          return $sum;
        }
        echo add(4, 6, 10, 12);
?>
```

Summary

- PHP is a server scripting language, and a powerful tool for making dynamic and interactive Web pages. It is widely used, free, and efficient alternative to competitors such as Microsoft's ASP.
- The real power of PHP comes from its functions. PHP function is a piece of code that can be reused many times. PHP functions are similar to other programming languages.
- A function is just a segment of code, separate from the rest of your code. You separate it because it's nice and handy, and you want to use it not once but over and over. It is a chunk of code that you think is useful, and want to use again.

- PHP has more than 1000 built-in functions, and in addition you can create your own custom functions.
- The date and time functions allow you to get the date and time from the server where your PHP scripts are running.
- PHP sessions functions allows you to access data across multiple pages in the same session. The functions.

Keywords

- ***Arrays in PHP:*** PHP array is an ordered map (contains value on the basis of key). It is used to hold multiple values of similar type in a single variable.
- ***Array() function:*** The array() function is used to create a PHP array. This function can be used to create indexed arrays or associative arrays.
- ***MySL:*** MySQL is an open-source relational database management system (RDBMS). It is the most popular database system used with PHP. MySQL is developed, distributed, and supported by Oracle corporation.
- ***Error:*** Error is the fault or mistake in a program. It can be several types. An error can occur due to wrong syntax or wrong logic. The type of mistakes or condition of having incorrect knowledge of the code.
- ***Parse error or Syntax Error:*** It is the type of error done by the programmer in the source code of the program. The syntax error is caught by the compiler. After fixing the syntax error the compiler compiles the code and execute it.

SelfAssessment

1. PHP array is an ___________ map and contains value on the basis of key.

A. Isolated
B. Ordered
C. Unordered
D. Emergent

2. A class is a template for objects, and an object is an instance of a class.

A. True
B. False

3. Which of the following function gets the name of the class of the given object?

A. get_class()
B. name_class()
C. classname_get()
D. none of the above

4. ____________ function checks if all of the characters in the provided string, text, are lowercase letters.

A. type_lower()
B. lower()

C. str_lower()
D. ctype_lower()

5. Which of the following is used to create a PHP array?
A. arr() function
B. ary() function
C. array() function
D. arrays() function

6. Ceil() function rounds fractions down and floor() function rounds fractions up.
A. True
B. False

7. The _________ function changes the permissions of the specified file and returns true on success and false on failure.
A. chdir()
B. chmod()
C. permit()
D. chperm()

8. Which of the following is the use of strlen() function in PHP?
A. The strlen() function returns the type of string.
B. The strlen() function returns the length of string.
C. The strlen() function returns the value of string.
D. The strlen() function returns both value and type of string.

9. Which of the following function is used to get the ASCII value of a character in PHP?
A. val()
B. asc()
C. ascii()
D. chr()

10. Syntax errors are caught by the compiler.
A. True
B. False

11. __________ error occurs when the PHP compiler understands the PHP code but it recognizes an undeclared function. This means that function is called without the definition of function.
A. Parse
B. Semantic
C. Fatal

D. Syntax

12. Which of the following is/are error(s) that can be handled by PHP error handling functions?

A. Parse errors

B. Fatal error

C. Syntax Error

D. All of the above

13. Which of the following statement is/are true for the date and time functions in PHP?

A. Allow you to get the date and time from the server where your PHP scripts are running.

B. Allow you to format the date and time in many different ways.

C. Allow you to modify the date.

D. All of the above

14. ____________ extension is used to read/retrieve the contents of an XML document.

A. XMLReader

B. XMLReading

C. ReadXML

D. RetrieveXML

15. In PHP call by reference, in order to pass value as a reference, you need to use _____________ symbol before the argument name.

A. hash (#)

B. ampersand (&)

C. sigil ($)

D. power (^)

Answers for Self Assessment

1.	B	2.	A	3.	A	4.	D	5.	C
6.	B	7.	B	8.	B	9.	D	10.	A
11.	C	12.	D	13.	D	14.	A	15.	B

Review Questions

1. What are built-in functions? Explain the significance of them.
2. Write a short note on:
   a. PHP Character functions
   b. PHP Array functions
3. How error handling in PHP does occurs? Discuss the functions used for it.

4. Discuss the role of string functions in PHP.
5. Differentiate function call by value and call by reference.
6. What are the user-defined functions? Discuss with examples.
7. How user-defined functions calls work with single parameters and multiple parameters?

Further Readings

- PHP Solutions by David Powers, Apress, 2007.
- Programming PHP by RusmusLerdorf, Oreilly Publishers, 2013.
- Creating dynamic Web sites with PHP and MySQL (PDF 20P) by Ashraful Alam.
- Practical PHP Programming by Paul Hudson.

Web Links

PHP Math Functions | Find Out How To Perform Math Functions In PHP (educba.com)

PHP User Defined Functions | Working of the PHP User Defined Function (educba.com)

https://www.w3schools.com/php/php_functions.asp

https://www.tutorialspoint.com/php-user-defined-functions

https://www.php.net/manual/en/functions.user-defined.php

PHP: defined - Manual

php eBook (riptutorial.com)

https://youtu.be/uK3kM7BJmKs

Unit 04: PHP Functions

CONTENTS

Objectives

After studying this unit, you will be able to:

- Understand the concept of global and super global variables in PHP.
- Explore the variable functions in PHP.
- Learn about the anonymous functions in PHP.
- Investigate the different date and time functions in PHP.

Introduction

Variable is an identifier, which holds a value. In programming, we say that we assign a value to a variable. Technically speaking, a variable is a reference to a computer memory, where the value is stored.In PHP language, a variable can hold a string, a number, or various objects like a function or a class. Variables can be assigned different values over time.The syntax of declaring a variable in PHP is given below:

```
$variablename=value;
```

Variables in PHP consist of the $ character, called a sigil, and a label. A variable must start with a dollar ($) sign, followed by the variable name.A label can be created from alphanumeric characters and an underscore _ character. A variable name must start with a letter or underscore (_) character.The value of a variable is the value of its most recent assignment. The variables are assigned with the = operator, with the variable on the left-hand side and the expression to be evaluated on the right.Variables can, but do not need, to be declared before assignment.

4.1 PHP Variable Scopes

The scope of a variable determines which part of the code can access it. The locations where the variable can be accessible determine the scope of the variable. In PHP, variables have four types of scopes:

- Local
- Static
- Global
- Function parameters

4.2 Local variables

When you define a variable inside a function, you can only access that variable within the function. And it's said that the variable is local to the function. The following example defines the say() function that displays the 'Hi' message:

Example:

```
<?php
function say() {
    $message = 'Hi';
    echo $message; }
say();
?>
```

*Output:*Hi

Inside the say () function, we define the $message variable. The $message variable is a local variable. And you cannot access it from the outside of the say() function.Also, the $message variable only exists during the execution of the say() function. Once the say() function ends, the $message variable won't exist anymore.

4.3 Static Variables

A static variable retains its value between function calls. Also, a static variable is only accessible inside the function. To define a static variable, you use the static keyword. Example

Example:

```
<?php
function get_counter() {
   static $counter = 1;
   return $counter++;
}
echo get_counter() .  '<br>'; // 1
echo get_counter() .  '<br>'; // 2
echo get_counter() .  '<br>'; // 3
?>
```

Output:

```
1
2
3
```

4.4 Global Variables

Global variables refer to any variable that is defined outside of the function. These variables can be accessed from any part of the script i.e. inside and outside of the function and can be declared just like other variable but it must be declared outside of function definition. The syntax is:

$variable_name = data; or $variable = your_value;

Example: $SampleVar = “Hi!!!! i am a global variable";

How Global Variables Work in PHP?

As the global variables are declared globally to use anywhere in the application, these variables can be accessed from inside or outside of the function as well. We just declare them like other variables, but in order to access them we need to follow some standards defined by the PHP. In order to use them inside the function and how to declare them.

Declaring Global Variables Examples: In this example, we are declaring a global variables and trying to print and concatenate them:

Example:

```
<?php
// Declaring global variable
$x = "Geeks";
$y = "for";
$z = "Geeks";
// Display value::  Concatenating String
echo $x.$y.$z;
?>
```

*Output:*GeeksforGeeks

Example:

```
<?php
//declaring global variable
$gvar1 = "Hello Friends ";
$gvar2 = "I am learning ";
$gvar3 = "PHP  ";
```

```
$gvar4 = "Global Variables today !!!!!!";
//printing result here
echo $gvar1.$gvar2.$gvar3.$gvar4.$gvar5;
?>
```

*Output:*Hello Friends I am learning PHP Global Variables today !!!!!!

Storing Global Variables

It is important to mention of how to store our global variable. There are ways for storing global variables using global array. In a global array, PHP maintains an array where it stores all the global variable that we defined in an application. By the use of this array, we can access this variable in and out of the script.

Accessing Global Variable inside Function

The ways to access the global variable inside functions are:

- Using global keyword
- Using array GLOBALS[var_name]: It stores all global variables in an array called $GLOBALS[var_name]. Var_name is the name of the variable. This array is also accessible from within functions and can be used to perform operations on global variables directly.

Using Global Keyword: In contrast to local variables, a global variable can be accessed in any part of the program. However, in order to be modified, a global variable must be explicitly declared to be global in the function in which it is to be modified. This is accomplished, conveniently enough, by placing the keyword GLOBAL in front of the variable that should be recognized as global. Placing this keyword in front of an already existing variable tells PHP to use the variable having that name.

Example:

```
<?php
  $gvar = 15;
  function increment()
  {
    GLOBAL $gvar;
    $gvar++;
     print "Gvar is $gvar";
  }
increment();
?>
```

*Output:*Gvar is 16

Global Array: In PHP we use an array to access this global variable. Like any other programming language, it maintains the history of the global variable in an array. If we want to access any particular element or variable from the array then we have to pass the exact name of the variable in order to access them. The syntax is:

```
$GLOBALS['variable_name']
```

It stores all global variables in an array called $GLOBALS[var_name]. Var_name is the name of the variable.

This array is also accessible from within functions and can be used to perform operations on global variables directly.

Example:

```
<?php
//declaring global variable
$gvar1= "Hello Friends ";
$gvar2= "I am learning ";
$gvar3= "PHP ";
$gvar4= "Global Variables today !!!!!!";
//declaring function
function demoglobe()
{
echo $GLOBALS['gvar1']."<br>";              //Using Global Array
echo $GLOBALS['gvar2']."<br>";
echo $GLOBALS['gvar3']."<br>";
echo $GLOBALS['gvar4']."<br>";
}
demoglobe();  //Call to the function
//printing result here
echo $gvar1.$gvar2.$gvar3.$gvar4;
?>
```

Output:

```
Hello Friends
I am learning
PHP
Global Variables today !!!!!!
Hello Friends I am learning PHP Global Variables today !!!!!!
```

Example:

An example which is using both Global Keyword and Global Array:

```
<?php
// Demonstrate how to declare global variable
// Declaring global variable
$str1 = "Geeks";
$str2 = "for";
$str3 = "Geeks";
$var1 = 15;
$var2 = 100;
$var3;
$var4;
function concatenate() // Using global keyword
```

```
{
   global $str1, $str2, $str3;
   return $str1.$str2.$str3;
}
function add()       {          // Using GLOBALS['var_name']
   $GLOBALS['var3'] = $GLOBALS['var1'] + $GLOBALS['var2'];
}
function mult()      {          // Using GLOBALS['var_name']
   $GLOBALS['var4'] = $GLOBALS['var1'] * $GLOBALS['var2'];
}
// Print result
echo concatenate();
echo"\n";
echo "<br/>";
add();
echo $var3;
echo "<br/>";
mult();
echo $var4;
?>
```

Output:

```
GeeksforGeeks
115
1500
```

Conclusively, by using the global variable we can access variables in our whole application these are useful when we have to use same value in the whole application for example username, password and so many other details depend on the requirement.

4.5 Super Global Variables

Superglobals were introduced in PHP 4.1.0, and are built-in variables that are always available in all scopes.Several predefined variables in PHP are "superglobals", which means that they are always accessible, regardless of scope - and you can access them from any function, class or file without having to do anything special. The superglobal variables provide information about the PHP script's environment. The different PHP superglobal variables are:

- $GLOBALS- Returns an array that contains global variables. The variable names are used to select which part of the array to access.
- $_SERVER- Returns data about the webserver environment.
- $_GET- Return data from GET requests.
- $_POST- Return data from POST requests.
- $_COOKIE- Return data from HTTP cookies
- $_FILES- Return data from POST file uploads.
- $_ENV- Return information about the script's environment.
- $_REQUEST- Return data from the HTTP request

- $_SESSION- Return variables registered in a session

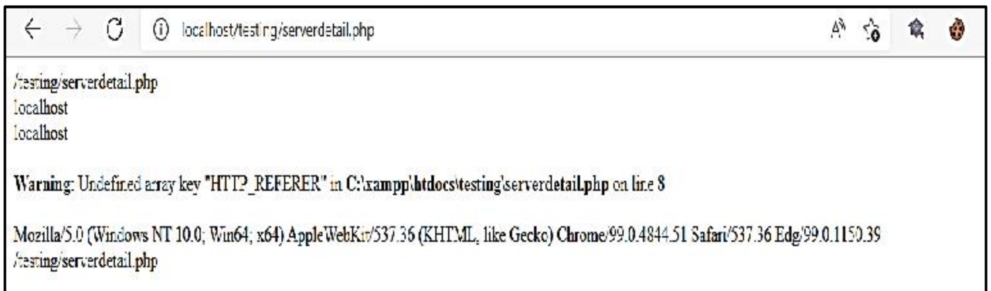
PHP $GLOBALS: $GLOBALS is a PHP super global variable which is used to access global variables from anywhere in the PHP script (also from within functions or methods).PHP stores all global variables in an array called $GLOBALS[index]. The index holds the name of the variable.

**PHP $_SERVER:**$_SERVER is a PHP super global variable which holds information about headers, paths, and script locations. The entries in this array are created by the web server.

- $_SERVER['PHP_SELF']: Returns the filename of the currently executing script.
- $_SERVER['SERVER_NAME']: Returns the name of the host server.
- $_SERVER['SERVER_ADDR']: Returns the IP address of the host server.
- $_SERVER['SCRIPT_FILENAME']: Returns the absolute pathname of the currently executing script.
- $_SERVER['REQUEST_METHOD']: Returns the request method used to access the page (such as POST).
- $_SERVER['SERVER_SOFTWARE']: Returns the server identification string (such as Apache/2.2.24).

Example:

```
<?php
echo $_SERVER['PHP_SELF'];
echo "<br>";
echo $_SERVER['SERVER_NAME'];
echo "<br>";
echo $_SERVER['HTTP_HOST'];
echo "<br>";
echo $_SERVER['HTTP_REFERER'];
echo "<br>";
echo $_SERVER['HTTP_USER_AGENT'];
echo "<br>";
echo $_SERVER['SCRIPT_NAME'];
?>
```


localhost/testing/serverdetail.php

```
/testing/serverdetail.php
localhost
localhost

Warning: Undefined array key "HTTP_REFERER" in C:\xampp\htdocs\testing\serverdetail.php on line 8

Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/99.0.4844.51 Safari/537.36 Edg/99.0.1150.39
/testing/serverdetail.php
```

PHP $_REQUEST:PHP $_REQUEST is used to collect data after submitting an HTML form.

$_SESSION: An alternative way to make data accessible across the various pages of an entire website is to use a PHP Session.A session creates a file in a temporary directory on the server where registered session variables and their values are stored. This data will be available to all pages on the site during that visit.The location of the temporary file is determined by a setting in the php.ini file called session.save path.A PHP session is easily started by making a call to the session start()function.This function first checks if a session is already started and if none is started then it starts one.It is recommended to put the call to session start () at the beginning of the page.

$Files- File Upload: There is one global PHP variable called $_FILES. This variable is an associate double dimension array and keeps all the information related to uploaded file.

- $_FILES['file']['tmp_name'] − the uploaded file in the temporary directory on the web server.
- $_FILES['file']['name'] − the actual name of the uploaded file.
- $_FILES['file']['size']− the size in bytes of the uploaded file.
- $_FILES['file']['type']− the MIME type of the uploaded file.
- $_FILES['file']['error']− the error code associated with this file upload.

4.6 Function Parameters

Function parameters are local to the function. Therefore, function parameters can only be accessible inside the function. An example is:

Example:

```
<?php
function sum($items)
{
  $total = 0;
foreach($items as $item) {
    $total += $item;
  }
  return $total;
}
// $items cannot be accessible here
echo sum([10,20,30]);
?>
```

Output: 60

4.7 Variable Functions in PHP

PHP supports the concept of variable functions. If a variable name has parentheses appended to it, PHP will look for a function with the same name as whatever the variable evaluates to, and will attempt to execute it. Among other things, this can be used to implement callbacks, function tables, and so forth. The variable functions won't work with language constructs such as echo, print, unset (), isset(), empty(), include, require and the like. You can use the utilize wrapper functions to make use of any of these constructs as variable functions.

PHP functions specifically for handling PHP variables. Such functions are used for performing variable related functionalities. The common functionalities applied on PHP variables include:

- Getting the value of PHP variables
- PHP Variable type handling.
- printing functions.
- Importing/Exporting PHP variables.
- Other variable functions

Getting the Value of PHP variables:For simply getting the value of the variable, we can use its name followed by a $ sign, and we can use the value for any purpose, like assigning it to another variable, returning it to the browser or for any other thing. But, PHP includes a certain list of variable functions that are dedicated for the special purposes, that is, for getting specified type value of a PHP variable. There are appropriate functions for each PHP supported datatype, for returning variable values in that particular type. These functions are,

- Interval ()– function accepts variable reference and base parameters to return integer by converting variable type with respect to the base.
- boolval()– function returns Boolean value of a given variable. And, it required only value or variable reference.
- floatval()– returns float value. And similarly, PHP contains an alias function doubleval().
- strval()– returns string value.

PHP Variable Type Handling

- get_resource_type()– returns type of the resource data to indicate, whether it is file resource or database resource or any other thing, and it returns FALSE while invoking with non-resource data.
- Get type ()/set type ()– Such functions are PHP datatype conversion, and, these are getters and setters of PHP variable data types.

Importing/Exporting PHP Variables

- import_request_variables()– This function is deprecated as of PHP version 5.3.0. Its purpose is to import PHP $_REQUEST variables into the global scope, where register_globals directive of the php.ini file is disabled.
- var_export()– This function accepts an optional return type parameter, which will be FALSE by default, and exporting variable data to the browser. While setting the optional parameter with TRUE, then, this function will return the array to be stored in a variable.

Printing Functions:These functions are used to display output to the browser instead of returning something to be stored into a variable.

- print_r()– It will print an array of data to the browser.
- var_dump () – It will print the array of data with more information to the browser. For example, it will print data with its length, type information.

Other PHP variable Functions

- get_defined_vars()– This function is used for getting an array of all defined variable of a PHP including global variables.
- empty ()– This function will be used in PHP if/else statements, as like as, is array(), is_numeric() and etc. This is for checking whether the given variable is empty or not.
- isset()– PHP isset() function is used to check whether a variable is defined or not, and no matter, if no values initialized with it.
- unset()– It will clear the value set with a variable.
- seralize()/unseralize()– seralize() converts value of a variable into storable data format, whereas,unseralize() is used to convert storable data format back to its original form.

The variable functions allow you to use a variable like a function. If the name of a variable has parentheses (with or without parameters in it) in front of it, PHP parser tries to find a function whose name corresponds to value of the variable and executes it. Such a function is called variable function. This feature is useful in implementing callbacks, function tables etc.

Example:

```
<?php
function hello()
{
 echo "Hello World";
}
$var="Hello";
$var();
?>
```

Output: Hello World

In the above example, value of a variable matches with function of name. The function is thus called by putting parentheses in front of variable.

Example:

```
<?php
        $f = 'strlen';
        echo $f('Hello');
        ?>
```

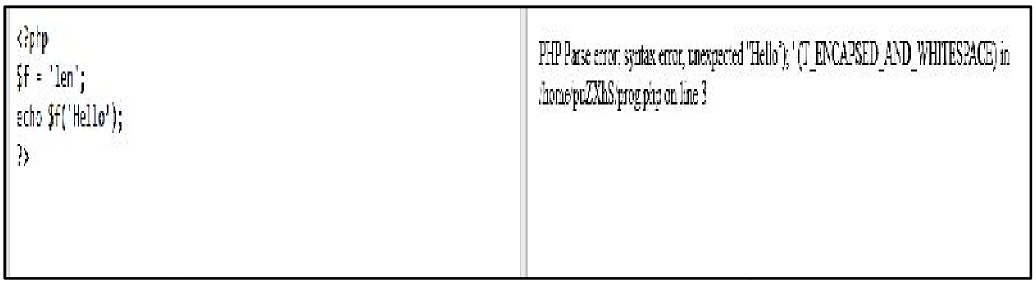
Output: 5

Here in the above code, firstly we will define a variable *$f* and initialize its value to the 'strlen' literal string.Second, use the $f as a function by passing the string 'Hello' to it.When PHP sees *$f(),* it looks for the *strlen()* function. Because the *strlen()* is a built-in function, PHP just invokes it.

However, if PHP cannot find the function name, it'll raise an error. For example:

Example:

```
<?php
$f = 'len';
echo $f('Hello');
?>
```



```
<?php
$f = 'len';
echo $f('Hello');
?>
```

```
PHP Parse error: syntax error, unexpected 'Hello');' (T_ENCAPSED_AND_WHITESPACE) in /home/puZXbS/prog.php on line 3
```

4.8 Using Variable Functions to Call a Method

The variable functions allow you to call the methods of an object. The syntax for calling a method using a variable function is as follows:

```
$this->$variable($arguments)
```

Example:

```
<?php
class Str {
          private $s;
          public function __construct(string $s) //Constructor with argument
                    {   $this->s = $s;    }   //Call to the current variable
          public function lower()
                    {   return mb_strtolower($this->s, 'UTF-8');     }
          public function upper()
                    {   return mb_strtoupper($this->s, 'UTF-8');     }
          public function title()
          {  returnmb_convert_case($this->s, MB_CASE_TITLE, 'UTF-8'); }
          public function convert(string $format)          {
                    if (!in_array($format, ['lower', 'upper', 'title'])) {                     throw
new Exception('The format is not supported'); }
                    return $this->$format();     }   }?>
```

In the above example, firstly we definea Str class that has three methods for converting a string to lowercase, uppercase, and title case. Secondly, we can define the convert () method that accepts a string. If the format argument is not one of the method names: lower, upper, and title, the convert () method will raise an exception. Otherwise, it'll call the corresponding method lower (), upper () or title().

4.9 Using Variable Functions to Call a Static Method

Example:

```
<?php
class myclass   {
  function welcome($name)
  {
    echo "Welcome $name";   }
}
$obj=new myclass();
$f="welcome";
$obj->$f("Amar");
?>
```

Output: Welcome Amar

Notes: We can use the className::$variable() to call a static method of a class.

A static method can be also called by variable method technique:

Example:

```
<?php
class myclass
{
  static function welcome($name){
    echo "Welcome $name";
  }
}
$f="welcome";
myclass::$f("Amar");
?>
```

Output: Welcome Amar

4.10 Variable Functions with Arguments

Example:

```
<?php
function add($x, $y)
{
  echo $x+$y;
}
$var="add";
$var(10,20);
?>
```

Output:30

4.11 Anonymous Functions in PHP

Anonymous function is a function without any user defined name. Such a function is also called closure or lambda function. Sometimes, you may want a function for one time use. The closure is an anonymous function which closes over the environment in which it is defined. You need to specify use keyword in it. The most common use of anonymous function to create an inline callback function.

Anonymous functions allow the creation of functions which have no specified name. They are most useful as the value of callable parameters, but they have many other uses.Anonymous functions are implemented using the Closure class. The syntax is:

```
$var=function ($arg1, $arg2) {return $val;};
```

There is no function name between the function keyword and the opening parenthesis.There is a semicolon after the function definition because anonymous function definitions are expressions. The function is assigned to a variable, and called later using the variable's name.When passed to another function that can then call it later, it is known as a callback. You can return it from within an outer function so that it can access the outer function's variables. This is known as a closure.

Example:

```
<?php
$var = function ($x) {return pow($x,3);};
echo "cube of 3 = " . $var(3);
?>
```

Output:

```
Cube of 3=27
```

4.12 Anonymous Function as Callback

In following example, an anonymous function is used as argument for a built-in usort() function. The usort() function sorts a given array using a comparison function.

Example:

```
<?php
$arr = [10,3,70,21,54];
usort ($arr, function ($x , $y) {   return $x > $y;   });
foreach ($arr as $x)
{
  echo $x . "\n";
}
?>
```

Output:

```
3 10 21 54 70
```

4.13 Anonymous Function as Closure

Closure is also an anonymous function that can access variables outside its scope with the help of use keyword. The closures can also be used as the values of variables; PHP automatically converts such expressions into instances of the Closure internal class. Also, assigning a closure to a variable uses the same syntax as any other assignment, including the trailing semicolon.

Example: Anonymous function variable assignment

```
<?php
$message = function($name)
{
printf("We are learning  %s\r\n", $name);
   echo "<br/>";
};
$message('World');
$message('PHP');
?>
```

Output:

```
We are learning World
We are learning PHP
```

4.14 PHP Date and Time Functions

The PHP date and time functions allow you to get the date and time from the server where your PHP scripts are running. You can use these functions to format the date and time in many different ways.

- **checkdate() Function:** Accepts the month, day, year of a date as parameters and, verifies whether it is a Gregorian date or not. The syntax is:

  checkdate (int $month , int $day , int $year)

Return Values: Returns a boolean value. This value is true if the given date is valid and, false if it is invalid.

PHP Version: Was first introduced in PHP Version 4, and works with all the later versions.

```
<?php
var_dump(checkdate(11, 07, 1989));
  echo "<br/>";
var_dump(checkdate(02, 31, 2022));
                                          //checkdate (int $month , int $day , int
$year)
  echo "<br/>";
  $bool = (checkdate(06, 03, 1999));
  print($bool);
   echo "<br/>";
  /*
var_dump(checkdate(02, 29, 2020));  //To check the leap year
   echo "<br/>";
var_dump(checkdate(02, 28, 2019));
    echo "<br/>"; */
?>
```

Output:

```
bool(true)
bool(false)
1
```

Example: In the following example, we are trying to verify the dates of leap year(s):

```
<?php
var_dump(checkdate(02, 30, 2004));
  echo "<br/>";
```

```
var_dump(checkdate(02, 28, 2008));
  echo "<br/>";
var_dump(checkdate(02, 29, 2021));  //Checking leap year dates
  echo "<br/>";
var_dump(checkdate(02, 29, 2020)); //Checking leap year dates
?>
```

Output:

```
bool(false)
bool(true)
bool(false)
bool(true)
```

- **date_create():** Returns new DateTime object: An alias of the DateTime::__construct, a constructor of the DateTime class. Where, a DateTime class represents date and time in PHP. It accepts a date time string and time zone (optional) as parameters and, creates a DateTime object accordingly. The syntax is:

  date_create([$date_time, $timezone]);

Return Values: Returns the created DateTime object.

PHP Version: This function was first introduced in PHP Version 5.2.0 and, works with all the later versions.

- **date_format() Function:**Thedate_format() function formats a given date. An alias of DateTime::format() function. It accepts a DateTime object and a format string (representing a desired date/time format) as parameters, formats the object in the specified format and, returns the result. The syntax is:

  string date_format(object, format);

Return Value: The date_format() function returns a string which represents the date formatted according to the specified format on successful formatting otherwise it returns false on failure.

Example: We are creating a DateTime object, formatting it, and printing the result.

```
<?php
 //Date string
          $date_str = "2022-09-25 13:09:08";
 //Creating a DateTime object
          $dob = date_create($date_str);
//formatting the date to print it
          $format = date_format($dob, "d-m-Y H:i:s");
          print($format);
?>
```

*Output:*25-09-2022 13:09:08

Example:

```
<?php
// using date_create() function to create a DateTime object
            $date=date_create("2022-03-15");
// using date_format() function to format date
            echo date_format($date, "Y/m/d H:i:s");
?>
```

*Output:*2022/03/15 00:00:00

Example: Following example formats a DateTime object as date and time separately.

```
<?php
  $dateStr = '16-02-2022 14:02:41 IST';
  $dto = date_create($dateStr);
$date = date_format($dto, 'y-m-d');
print("Date: $date");
  $time = date_format($dto, 'H:i:s');
  echo "<br/>";
print("Time: $time");
?>
```

Output:

```
Date: 22-02-16
Time: 14:02:41
```

- **date_date_set() Function:** An alias of the DateTime::setDate(). Using this, you can (re)set the date of a DateTime object.

Syntax: date_date_set($object, $year, $month, $day)

Return Values: Returns DateTime object with modified value. Incase of failure, this function returns boolean value false.

PHP Version: This function was first introduced in PHP Version 5.2.0 and, works with all the later versions.

Example: Demonstrate the usage of date_date_set function.

```
<?php
  //Creating a date
  $date = new DateTime();
  //Setting the date
date_date_set($date, 2022, 10, 02);
print("Date: ".date_format($date, "Y/m/d"));
?>
```

Output:

Date: 2022/10/02

Example: illustrating the use of date function:

```
<?php
       $d = date("D");  //Takes current system date
       if ($d == "Fri")
          echo "Have a nice weekend!";
elseif ($d == "Mon")
          echo "Have a nice Sunday!";
else
          echo "Have a nice day!";
     ?>
```

Output:

```
Have a nice day!
```

Example: The following example creates a DateTime object and modifies its date using the date_date_set() function.

```
<?php
  //Date string
  $date_str = "07-03-2022";
  //Creating a DateTime object
  $dob = date_create($date_str);
print("Original Date: ".date_format($dob, "Y/m/d"));
print("\n");
  //Setting the date
  $date = date_date_set($dob, 2022, 11, 21);   //21st November, 2022
print("Modified Date: ".date_format($date, "m/d/y"));
?>
```

Output:

```
Original Date: 2022/03/07 Modified Date: 11/21/22
```

Notes: While invoking this function if you pass the day and month values exceeding their range, they will be added to their parent values.

```
<?php
  //Creating a date
  $date = new DateTime();
  //Setting the date
```

```
date_date_set($date, 2019, 15, 17);
print("Date: ".date_format($date, "Y/m/d"));
?>
```

Since we have set the month value as 15. Three months are added to the appropriate date:

```
Date: 2020/03/17
```

- **date_modify():** An alias of DateTime::modify(). This function is used to modify the date in a DateTime object. It alters the time stamp of the given object.

Syntax: date_modify($object, $modify)

Return Values: Returns the DateTime object with modified value. Incase of failure, this function returns the boolean value false.

PHP Version: This function was first introduced in PHP Version 5.2.0 and, works with all the later versions.

Example:

```
<?php
  //Modifying the date
  $date = date_modify(new DateTime(), "+15 day");
  //Will add 15 days to the current system date
print("Date: ".date_format($date, "Y/m/d"));
?>
```

Output:

```
Date: 2022/03/31
```

- **date_sunrise():** Returns the time of sunrise for a given day/ location. It accepts a timestamp representing the given day and returns the sunrise time on that particular day.

Syntax:date_sunrise($timestamp, [$format, $latitude, $longitude]

Return Values: Returns the time of the sunrise in desired format. Incase of failure, it returns the boolean value false.

PHP Version: This function was first introduced in PHP Version 5.0 and, works with all the later versions.

- **date_sunset():** Returns the time of sunset for a given day/ location. It accepts a timestamp representing the given day and, returns the sunset time on that particular day.

Syntax:date_sunset($timestamp, [$format, $latitude, $longitude])

Return Values: Returns the time of the sunset in desired format. Incase of failure, it returns the boolean value false.

PHP Version: This function was first introduced in PHP Version 5.0 and, works with all the later versions.

- **date_time_set():** Sets the time. An alias of the DateTime::setTime() function. Using this, you can (re)set the time of a DateTime object.

Syntax: date_time_set($object, $hours, $minutes, $seconds, $microseconds).

Return Values: Returns the DateTime object with modified (time) value. Incase of failure, this function returns the boolean value false.

PHP Version: This function was first introduced in PHP Version 5.2.0 and, works with all the later versions.

Example:

```
<?php
  //Creating a date
  $date = new DateTime();
  //Setting the date-time
date_time_set($date, 5, 20, 45);
print("Date: ".date_format($date, "Y/m/d H:i:s"));
?>
```

Output:

```
Date: 2022/05/11 05:20:45
```

- **date() Function:** Accepts a format string as a parameter, formats the local date/time in the specified format and returns the result.

Syntax: date ($format, $timestamp)

Return Values: Returns the current local time/date in the specified format.

PHP Version: This function was first introduced in PHP Version 4 and, works with all the later versions.

- **Get date() Function:** Used to get information about a particular date/time. It accepts an optional parameter specifying the time stamp of which you need the info about. If you haven't passed any argument, this function returns information about the current local time.

Syntax: getdate([$timestamp])

Return Values: Returns an array containing the information about the given time/date.

PHP Version: This function was first introduced in PHP Version 4 and, works with all the later versions.

Example:

```
<?php
  $date = date("D M d Y"); //Day Month Day Year
print("Date: ".$date);
  $info = getdate();
  echo "<br>";
print_r($info);
?>
```

Output:

```
Date: Wed Mar 16 2022
Array ( [seconds] => 21 [minutes] => 37 [hours] => 8 [mday] => 16 [wday] => 3 [mon] => 3
[year] => 2022 [yday] => 74 [weekday] => Wednesday [month] => March [0] =>
1647419841 )
```

- **date_timezone_get() Function:** An alias of DateTime::getTimezone. It accepts a DateTime object as a parameter and returns the timezone object relative to the given date/time (object).

Syntax: date_timezone_get($object)

Return Values: Returns a DateTimeZone object. Incase of failure it returns the boolean value false.

PHP Version: This function was first introduced in PHP Version 5.2.1 and, works with all the later versions.

- **date_timezone_set() Function:** Accepts a DateTime object and a timezone object as parameters and, sets the specified timezone to the given DateTime.

Syntax: date_timezone_set($object, $time zone)

Return Values: Returns a DateTime object. Incase of failure it returns the boolean value false.

PHP Version: This function was first introduced in PHP Version 5.2.0 and, works with all the later versions.

- **gettimeofday() Function:** Returns the current time of the day. By default, this function returns the current time as an array. If you pass the boolean value true as an argument, this function returns the time as floating point number.

Syntax: gettimeofday($return_float)

Return Values: Returns the current time. By default this value will be an array with keys: sec, usec, minuteswest, dsttime. If you set the return_float value to true, the time will be returned as floating-point value.

PHP Version: This function was first introduced in PHP Version 4 and, works with all the later versions.

Example:

```
<?php
  $time = gettimeofday();
print_r($time);
?>
```

Output:

```
Array ([sec] => 1646720009 [usec] => 119598 [minuteswest] => 0 [dsttime] => 0)
```

- **date_add() Function:** An alias of DateTime::add(). It accepts a DateTime object as parameters and a DateInterval object, adds the specified interval to the given DateTime.

Syntax: date_add($object, $interval)

Return Values: Returns a DateTime object with added interval. In case of failure, this function returns the boolean value false.

PHP Version: This function was first introduced in PHP Version 5.3.0 and, works with all the later versions.

- **date_sub() Function:** Accepts a DateTime object and a DateInterval object, subtracts the specified interval to the given DateTime.

Syntax: date_sub($object, $interval)

Return Values: Returns a DateTime object, subtracting the given interval from it. In case of failure, this function returns the boolean value false.

PHP Version: This function was first introduced in PHP Version 5.3.0 and, works with all the later versions.

- **date_create_from_format() Function:** An inbuilt function in php which is used to parses a time string according to a specified format. This function accepts three parameters and returns new DateTime in success or false on failure.

Syntax: date_create_from_format ($format, $time, $timezone)

Return Value: This function returns a new DateTime instance on success or FALSE on failure.

Example:

```
<?php
// Declare a date in given format
$date = date_create_from_format('D-M-Y', 'monday-Feb-2022');
// Output date in given format
echo date_format($date, 'y-n-j');
?>
```

Output:

```
22-2-14
```

Summary

- A variable is a reference to a computer memory, where the value is stored. In PHP language, a variable can hold a string, a number, or various objects like a function or a class. Variables can be assigned different values over time.
- Variables in PHP consist of the $ character, called a sigil, and a label. A variable must start with a dollar ($) sign, followed by the variable name.
- When you define a variable inside a function, you can only access that variable within the function.
- Global variables refer to any variable that is defined outside of the function. These variables can be accessed from any part of the script i.e. inside and outside of the function and can be declared just like other variable but it must be declared outside of function definition.
- Superglobals were introduced in PHP 4.1.0, and are built-in variables that are always available in all scopes. Several predefined variables in PHP are "superglobals", which means that they are always accessible, regardless of scope - and you can access them from any function, class or file without having to do anything special.
- PHP date and time functions allow you to get the date and time from the server where your PHP scripts are running. You can use these functions to format the date and time in many different ways.
- The superglobal variables provide information about the PHP script's environment.

Keywords

Variable: Variable is an identifier, which holds a value.

Global Variables:Global variables refer to any variable that is defined outside of the function. These variables can be accessed from any part of the script i.e. inside and outside of the function and can be declared just like other variable but it must be declared outside of function definition.

PHP $_REQUEST: PHP $_REQUEST is used to collect data after submitting an HTML form.

Global Array: Ina global array, PHP maintains an array where it stores all the global variable that we defined in an application. By the use of this array, we can access this variable in and out of the script.

Gettimeofday () Function:Returns the current time of the day. By default, this function returns the current time as an array.

Closure:Closure is also an anonymous function that can access variables outside its scope with the help of use keyword. The closures can also be used as the values of variables; PHP automatically converts such expressions into instances of the Closure internal class.

Date & Time Functions:PHP date and time functions allow you to get the date and time from the server where your PHP scripts are running.

SelfAssessment

1. Which of the following statement(s) is/are true for a variable in PHP?

A. In PHP, a variable is an identifier which holds a value.

B. In PHP language, a variable can hold a string, a number, or various objects like a function or a class.

C. Variable is a reference to a computer memory, where the value is stored.

D. All of the above.

2. In PHP, a variable name must start with a letter or __________ character.

A. Underscore (_)

B. Hash (#)

C. Power (^)

D. Double quotes (")

3. In order to define a static variable, the ______________ keyword is used.

A. Stat

B. Non-variable

C. Static

D. No change

4. A PHP session is easily started by making a call to the _________ function.

A. session_go()

B. session_init()

C. session_start()

D. session_due()

5. PHP maintains an array where it stores all the global variable that we defined in an application. Such array is called __________ array.

A. Static

B. Global

C. Random

D. Fixed

6. $_SERVER is a PHP super global variable which holds information about _______.

A. Headers

B. Paths

C. Script locations

D. All of the above

7. Which of the following statement(s) is valid for global variables?
A. Global variables refer to any variable that is defined outside of the function.
B. Global variables can be accessed from any part of the script, that is, inside and outside of the function.
C. Global variables can be declared just like other variable but it must be declared outside of function definition.
D. All of the above

8. __________ function accepts variable reference and base parameters to return integer by converting variable type with respect to the base.
A. Intern ()
B. Internal ()
C. Include ()
D. Interval ()

9. ___________ function will print the array of data with more information to the browser such as its length, type information.
A. var_dump()
B. dump()
C. echo()
D. printer()

10. ___________ function in PHP converts value of a variable into storable data format, whereas, ________ is used to convert storable data format back to its original form.
A. store (), unstore()
B. seralize(), unseralize()
C. convert (), unconvert()
D. varst(), dvarst()

11. Anonymous function is a function without any user defined name. Such a function is also called ___________ function.
A. Lambda
B. Variable
C. Anon
D. Close

12. The date_format() function returns ___________ which represents the date formatted according to the specified format on successful formatting otherwise it returns false on failure.
A. String
B. Character

C. ASCII
D. Float

13. Which function is used to return the date information?
A. date()
B. getdate()
C. time_date()
D. None of the above

14. The checkdate() function accepts _____________ as parameters and, verifies whether it is a Gregorian date or not.
A. Month of a date
B. Day of a date
C. Year of a date
D. All of the above

15. Which of the following statement is true for date_timezone_set() function?
A. It accepts a DateTime object as a parameter.
B. It accepts a timezone object as parameter.
C. It sets the specified timezone to the given DateTime
D. All of the above

Answers forSelf Assessment

1. D	2. A	3. C	4. C	5. B
6. D	7. D	8. D	9. A	10. B
11. A	12. A	13. B	14. D	15. D

Review Questions

1. Explain the significance of date and time functions in PHP?
2. Write a short note on:
   a. Global array
   b. Global keyword
3. Discuss the different date and time functions in PHP?
4. What are Anonymous functions and how Callback works?
5. What are variables functions in PHP?
6. Considering the importance of variable functions, explain the functionalities applied on PHP variables?
7. Discuss the concept of super global variables with examples.

Further Readings

- PHP Solutions by David Powers, Apress, 2007.
- Programming PHP by RusmusLerdorf, Oreilly Publishers, 2013.
- Creating dynamic Web sites with PHP and MySQL (PDF 20P) by Ashraful Alam.
- Practical PHP Programming by Paul Hudson.

Web Links

- PHP Global Variable | How Global Variable works in PHP? (educba.com)
- PHP - Global Variables (tutorialspoint.com)
- PHP - Functions (tutorialspoint.com)
- PHP Anonymous functions (tutorialspoint.com)
- PHP - Date & Time (tutorialspoint.com)
- PHP Global Variables - Superglobals (w3schools.com)
- https://youtu.be/YZ3BPzMU3WQ
- https://youtu.be/kUs8bAaab70
- https://youtu.be/B5wMcJC0x5w

Dr. Tarandeep Kaur, Lovely Professional University

Unit 05: PHP Strings and Math Functions

CONTENTS

Objectives

After studying this unit, you will be able to:

- Know about the string functions in PHP.
- Explore the working of PHP strings.
- Analyze the use of various string functions in PHP.
- Compare strings in PHP.
- Learn about the use of Math functions in PHP.

Introduction

The term "string" refers to a series of characters. PHP supports a number of data types, including strings. The alphanumeric characters are allowed in string variables.When these conditions are met, strings are generated. It is possible to create a variable and assign it to string characters. With the echo argument, using PHP Strings directly.

The string functions in PHP are language constructs that aid in the capture of terms. These are used to modify a string or query knowledge about a string (some do both).The length (string) function is the most basic example of a string function.The length of a string literal is returned by this function:str(len)

5.1 Importance of Using String Functions in PHP

The string function can be used for various operations such as:

- Finding length of a string
- Counting number of words in string
- Reversing a string
- Searching text in string
- Replacing text in a string
- Converting lowercase into title case
- Converting a whole string into UPPERCASE and lowercase

- Comparing string

Finding Length of a String: The length of a string can be found by using the most basic type of built-in function.The length of a string can be determined using a predefined function in PHP.strlen() function returns the length of any string. It's most widely used in input fields where the user can only type a certain number of characters.

Syntax: Strlen(String);

Example:

```
<?php
echo strlen('Learning string functions in PHP');
        //Will return the length of given string
?>
```

Output:32

```
<?php
$s = "Good Day Ahead";
        // Illustrate the use of strlen
echo "Total string length is: ". strlen($s);
?>
```

*Output:*Total string length is: 14

Counting Number of Words in String: A built-in function can be used to display the number of words in a given string(). The validation of input fields is also possible with this feature.

Syntax: Str_word_count(String)

Example:

```
<?php
echo str_word_count('We are taking online class of PHP');
        //returns the number of words in a String
?>
```

Output: 7

```
<?php
$s = "Good Day Everyone";
        //Use of word count() function
echo "Total string length is: ". str_word_count($s);
?>
```

Output:Total string length is: 3

Reversing a String: The Strrev() is a function that reverses a string and can be used to get the reverse version of any string.

Syntax: Strrev(String)

Example:

```
<?php
```

```
echo strrev('Welcome to string functions in php');
            //returns the reverse of the string passed
?>
```

Output:php nisnoitcnufgnirtsotemocleW

Searching Text in String: Strpos() allows you to check for specific text within a string. It works by simply matching a string's basic text. If a match is found, the location is returned. It will return "False" if the item is not found at all. It is widely used to validate input fields such as email addresses.

Syntax: Strops(String,text);

Example:

```
<?php
echo strpos('We are learning about string functions in PHP','string');
            //Returns the location of the word being searched
?>
```

Output: 22

Replacing Text in a String: Strreplace() is a built-in function that allows you to replace specific text in a string.

Syntax: Str_replace((string to be replaced,text,string)

Example:

```
<?php
echo str_replace('python', 'php', 'Learning python is fun');
            //Replaces desired word
?>
```

Output:Learning php is fun

```
<?php
   //replacement from the front of string
   $r1 = substr_replace("Hello Everyone", "Day", 6);
   echo $r1;
   //replacement from end of the string
   $r2 = substr_replace("All the best", "Great", -4);
   echo $r2;
?>
```

Output:Hello DayAll the Great

Converting Lowercase into Title Case: Ucwords() is a function that converts the first letter of each word to uppercase.

Syntax: Ucwords(string);

Example:

```
<?php
```

```
echo ucwords('welcome to the world of learning PHP');
?>
```

Output: Welcome To The World Of Learning PHP

Converting a Whole String into UPPERCASE and lowercase: There are two functions related to converted whole string into UPPERCASE and lowercase.

- Strtoupper() used to convert all string characters into upper.
- Strtolower() used to convert all string characters into lower.

Syntax:

```
Strtoupper(String);
Strtolower(String);
```

Example:

```
<?php
echo strtoupper('welcome to the world of learning PHP');
echo "<br/>";
echo strtolower('Welcome to THE WORLD OF LEARNING PHP');
?>
```

Output: WELCOME TO THE WORLD OF LEARNING PHP

welcome to the world of learning php

Comparing String:strcmp can be used to compare two strings () and produces the output that is either greater than, less than, or equal to zero. It returns greater than zero if string 1 is greater than string 2. It returns less than zero if string 1 is less than string 2. If the strings are equal, it returns zero.

Syntax: Strcmp(string1,string2)

Example:

```
<?php
echo strcmp("hi","hi");
echo '<br>';
?>
```

//Output: 0

Example:

```
<?php
echo strcmp("hi hello","hi");
echo '<br>';
?>
```

Output: 6

Example:

```
<?php
echo strcmp("hi","hi hello");
echo '<br>';
?>
```

Output: -6

5.2 Other String Functions

There are several other functions that can perform different operations related to the strings.

chr() Function: This function returns a character from the specified ASCII value.The ASCII value can be specified in decimal, octal, or hex values. The octal values are defined by a leading 0, while hex values are defined by a leading 0x.

Syntax: chr(ascii)

Example:

```
<?php
echo chr(65) . "<br>";
echo chr(062) . "<br>";
echo chr(0x75) . "<br>";
?>
```

Output:

```
A
2
U
```

chop() Function: The chop() function removes the whitespaces or other predefined characters from the right end of a string.

Syntax: chop(string)

Example:

```
<?php
echo chop("Hello    World ");
            //Chops whitespace from the right of the string
?>
```

Output: Hello World

substr() Function:A built-in function in PHP that is used to extract a part of string.The first argument is the string itself, second argument is the starting index of the substring to be extracted and the third argument is the length of the substring to be extracted.

Syntax: substr($str, $start, $length)

Example:

```
<?php
echo substr("Good Morning",5,2);
```

```
//substr($str, $start, $length)
?>
```

Output: Mo

substr_count() Function: It counts the number of times a substring occurs in the given string.Substr_count() is a case-sensitive function., which means it treats uppercase and loweSyntax

Syntax of the substr_count() is given below, which accepts four parameters, two string, and two integer type values.

```
substr_count($string, $substring)
```

Example:

```
<?php
  $s = "Good Mood Leads to Good Day";
  echo substr_count($s,"Good");
?>
```

Output: 2

ucfirst() Function: This function is used to convert the first character of the string into the uppercase.

Syntax: Contains only one string parameter.

```
ucfirst( $string)
```

Example:

```
<?php
  $st1 = "hello dear how r you?";
  echo ucfirst($st1);
  echo "<br/>";
  echo ucwords($st1);
?>
```

Output: Hello dear how r you?

Hello Dear How R You?

lcfirst() Function: This function converts the first character of a string to lowercase.

Syntax: lcfirst(string)

Example:

```
<?php
  $st1 = "HELLO DEAR HOW R YOU?";
  echo lcfirst($st1);
  echo "<br/>";
  //echo ucwords($st1);
?>
```

Output: hELLO DEAR HOW R YOU?

str_split() Function:str_split() function splits a string into an array.

Syntax: str_split(string,length)

Example:

```
<?php
print_r(str_split("Good",2));
?>
```

Output: Array ([0] => Go [1] => od)

Strcasecmp() Function: The strcasecmp() function compares two string.

Syntax: strcasecmp(string1,string2)

Example:

```
<?php
echo strcasecmp("Good Day","good day");
?>
```

Output: 0

trim() Function:trim() function removes whitespace and other predefined characters from both sides of a string.

- ltrim()- Removes whitespace or other predefined characters from the left side of a string.
- rtrim()- Removes whitespace or other predefined characters from the right side of a string.

Parameter	Description
string	Required. Specifies the string to check
Charlist	Optional. Specifies which characters to remove from the string. If omitted, all of the following characters are removed: "\0" - NULL "\t" - tab "\n" - new line "\x0B" - vertical tab "\r" - carriage return " " - ordinary white space

Example: Trim Function example:

```
<?php
$str = "Hello World!";
echo $str . "<br>";
echo trim($str,"Hed!");
?>
```

Output: Hello World!

lloWorl

Trimming white spaces: Write a program to remove whitespaces from both sides of a string?

```
<?php
$str = " Hello World! Hii        ";
echo "Without trim: " . $str;
echo "<br>";
echo "With trim: " . trim($str);
?>
```

Output:Without trim: Hello World! Hii

With trim: Hello World! Hii

Lab Exercise: Write a program to remove newlines (\n) from both sides of the string.

5.3 Math Functions in PHP

Math functions offer the inbuilt functionalities of PHP. These provide a mechanism in which a developer can do some sort of math calculations or similar stuff. Using such functions, provides a quick hands-on for development without writing a long piece of code.

Range of PHP Math Functions

The range of PHP math functions is within integer and float types. The range of integer data type in PHP for a 32-bit computer is -2,147,483,647 to 2,147,483,647. Any number smaller than -2,147,483,647 or any number greater than 2,147,483,647 or any number smaller than -2,147,483,647 is considered as float.

Abs () Function: This function was introduced in PHP 4+ version and returns the absolute value of the number. It is used to return the absolute (positive) value of a number and is identical to what we call modulus in mathematics. The modulus or absolute value of a negative number is positive.

Syntax: number abs(value)

Parameters: Accepts single parameter value which holds the number whose absolute value you want to find.

Return Value: Return type of the function is float or integer number depending upon what type of argument passed in the function.

Example:

```
<?php
echo (abs(-9));                    //Output: 9
?>
```

ceil() Function: Ceil() function rounds fractions up.

Syntax: float ceil (float $value)

Example:

```
<?php
echo (ceil(3.3)."<br/>");// 4
echo (ceil(7.333)."<br/>");// 8
echo (ceil(-4.8)."<br/>");// -4
?>
```

floor() Function:floor() function rounds fractions down.

Syntax: float floor (float $value)

Example:

```
<?php
echo (floor(3.3)."<br/>");              // 3
echo (floor(7.333)."<br/>");                    // 7
echo (floor(-4.8)."<br/>");             // -5
?>
```

sqrt () Function: sqrt() function returns square root of given argument.

Syntax: float sqrt (float $arg)

Example:

```
<?php
echo (sqrt(16)."<br/>");     // 4
echo (sqrt(25)."<br/>");      // 5
echo (sqrt(7)."<br/>");      // 2.6457513110646
?>
```

Example:

sqrt() by declaring variables

```
<?php
$five = sqrt(25);
echo $five;
echo "<br/>";
$four = sqrt(16);
echo $four;
?>
```

Output:

```
5
4
```

pi() Function: This function was introduced in PHP 4+ version. It returns value of a PI and its return type is a float.

Example:

```
<?php
echo(pi() . "<br>");
?>
```

Output: 3.1415926535898

pow() Function: This function was introduced in PHP 4+ version. It accepts two arguments say x and y. It calculates x raised to the power of y. The return type is integer or float which depends upon the nature of the argument.

Example:

```
<?php
echo(pow(2,3) . "<br>");
echo(pow(2,4) . "<br>");
echo(pow(5,6) . "<br>");
echo(pow(3,5));
?>
```

Output:

```
8
16
15625
243
```

log() Function: This function was introduced in PHP 4+ version. It accepts two arguments say x and y where x is a number and y is the logarithm of a number to base.If y is not passed then the default value 'e' is assumed. Its return type is float.

Example:

```
<?php
echo(log(2.718) . "<br>");
echo(log(2) . "<br>");
echo(log(1) . "<br>");
echo(log(0));                //Infinite
?>
```

Output:

```
0.99989631572895
0.69314718055995
0
-INF
```

log10() Function: It was introduced in PHP 4+ version and accepts one argument says x where x is a number whose base 10 logarithm needs to be calculated. The return type is float.

Example:

```
<?php
echo(log10(656) . "<br>");
echo(log10(455) . "<br>");
echo(log10(145) . "<br>");
?>
```

Output:

```
2.8169038393757
2.6580113966571
2.161368002235
```

round() Function: This function was introduced in PHP 4+ version. It rounds a number and expects three parameters where the first parameter is number, the second parameter is for precision and the third argument is for mode. The only first argument is mandatory.

Example:

```
<?php
echo(round(3.35) . "<br>");
echo(round(-2.35) . "<br>");
echo(round(5) . "<br>");
?>
```

Output:

```
3
-2
5
```

max() Function: This function was used to find the numerically maximum value in an array or the numerically maximum value of several specified values. It can take an array or several numbers as an argument and return the numerically maximum value among the passed parameters.

Return type is not fixed, it can be an integer value or a float value based on input.

Syntax: max (array_values) or max(value1, value2, ...)

Parameters: This function accepts two different types of arguments which are explained below:

- array_values: It specifies an array containing the values.
- value1, value2, …: It specifies two or more than two values to be compared.

Example:

```
<?php
echo (max(12, 4, 62, 97, 26));
?>
```

Output: 97

min() Function: This function returns the lowest value in an array, or the lowest value of several specified values.

Syntax: min (array_values);   or min(value1,value2,...);

Parameters

- array_values: Required. Specifies an array containing the values
- value1,value2,...: Required. Specifies the values to compare (must be at least two values)

Example:

```
<?php
echo(min(2,4,6,8,10) . "<br>");
echo(min(22,14,68,18,15) . "<br>");
echo(min(array(4,6,8,10)) . "<br>");
echo(min(array(44,16,81,12)));
?>
```

Output:

```
2
14
```

```
4
12
```

decbin() Function: The decbin() function converts decimal number into binary. It returns binary number as a string.

Syntax: string decbin (int $number)

Example:

```
<?php
echo (decbin(2)."<br/>");
echo (decbin(10)."<br/>");
echo (decbin(22)."<br/>");
?>
```

Output:

```
10
1010
10110
```

dechex() Function: It converts decimal number into hexadecimal. It returns hexadecimal representation of given number as a string.

Syntax: string dechex (int $number)

Example:

```
<?php
echo (dechex(2)."<br/>");
echo (dechex(10)."<br/>");
echo (dechex(22)."<br/>");
?>
```

Output:

```
2
a
16
```

Base convert() Function: This function converts a number from one number base to another.

Syntax: base convert(number,frombase,tobase);

Parameter

- Number Required. Specifies the number to convert
- Frombase: Required. Specifies the original base of number. Has to be between 2 and 36, inclusive. Digits in numbers with a base higher than 10 will be represented with the letters a-z, with “a meaning 10”, “b meaning 11” and “z meaning 35”.
- Tobase: Required. Specifies the base to convert to. Has to be between 2 and 36, inclusive. Digits in numbers with a base higher than 10 will be represented with the letters a-z, with a meaning 10, b meaning 11 and z meaning 35

Task:

1.Convert an octal number to a decimal number?

2.Convert a hexadecimal number to octal number?

3.Convert an octal number to a hexadecimal number?

Lab Exercise:

Write a PHP script to find the maximum and minimum marks from the set of arrays?

$marks1 = array(360,310,310,330,313,375,456,111,256);

$marks2 = array(350,340,356,330,321);

$marks3 = array(630,340,570,635,434,255,298);

Expected Output :

Maximum marks:

Minimum marks:

```
<?php
$marks1 = array(360,310,310,330,313,375,456,111,256);
$marks2 = array(350,340,356,330,321);
$marks3 = array(630,340,570,635,434,255,298);
$max_marks = max(max($marks1),max($marks2),max($marks3));
$min_marks = min(min($marks1),min($marks2),min($marks3));
echo "Maximum marks : ".$max_marks."\n";
echo "Minimum marks : ".$min_marks."\n";
?>
```

Lab Exercise:

Write a PHP script which rounds the following values with 1 decimal digit precision?

Sample values and Output :

1.65 --> 1.7

1.65 --> 1.6

-1.54 --> -1.5

```
<?php
echo round( 1.65, 1, PHP_ROUND_HALF_UP)."\n";   //  1.7
echo "<br/>";
echo round( 1.65, 1, PHP_ROUND_HALF_DOWN)."\n"; //  1.6
echo "<br/>";
echo round(-1.54, 1, PHP_ROUND_HALF_EVEN)."\n"; // -1.5
?>
```

Summary

- PHP is a server scripting language, and a powerful tool for making dynamic and interactive Web pages. It is widely used, free, and efficient alternative to competitors such as Microsoft's ASP.

- The term "string" refers to a series of characters. PHP supports a number of data types, including strings.
- Alphanumeric characters are allowed in string variables. When these conditions are met, strings are generated.
- It is possible to create a variable and assign it to string characters.
- The string functions in PHP are language constructs that aid in the capture of terms. These are used to modify a string or query knowledge about a string (some do both).
- Math functions offer the inbuilt functionalities of PHP. These provide a mechanism in which a developer can do some sort of math calculations or similar stuff.
- strcmp can be used to compare two strings () and produces the output that is either greater than, less than, or equal to zero. It returns greater than zero if string 1 is greater than string 2. It returns less than zero if string 1 is less than string 2.

Keywords

PHP: PHP is a server scripting language, and a powerful tool for making dynamic and interactive Web pages. PHP is a widely used, free, and efficient alternative to competitors such as Microsoft's ASP.

chop() Function:The chop() function removes the whitespaces or other predefined characters from the right end of a string.

abs() Function:This function was introduced in PHP 4+ version and returns the absolute value of the number. It is used to return the absolute (positive) value of a number.

Ucwords() Function:Ucwords() is a function that converts the first letter of each word to uppercase.

pow() Function: This function accepts two arguments say x and y. It calculates x raised to the power of y. The return type is integer or float which depends upon the nature of the argument.

trim() Function:trim() function removes whitespace and other predefined characters from both sides of a string.

chr() Function:This function returns a character from the specified ASCII value. The ASCII value can be specified in decimal, octal, or hex values. The octal values are defined by a leading 0, while hex values are defined by a leading 0x.

substr_count() Function: It counts the number of times a substring occurs in the given string. Substr_count() is a case-sensitive function., which means it treats uppercase and loweSyntax

floor() Function:floor() function rounds fractions down.

SelfAssessment

1. Which of the following function converts a string to all uppercase?
A. upper ()
B. uppercase ()
C. struppercase()
D. strtoupper()

2. String values in PHP must be enclosed within
A. Double Quotes
B. Single Quotes
C. Both (a) and (b)
D. None of the above

3. Which of the following is the use of strlen() function in PHP?
A. The strlen() function returns the type of string
B. The strlen() function returns the length of string
C. The strlen() function returns the value of string
D. The strlen() function returns both value and type of string

4. Which of the following function is used to get the ASCII value of a character in PHP?
A. val()
B. asc()
C. ascii()
D. chr()

5. _____________ function removes whitespace and other predefined characters from the left side of a string.
A. strim()
B. lefttrim()
C. ltrim()
D. lftrim()

6. The statement that aptly indicates the feature(s) for substr_count() function:
A. It counts the number of times a substring occurs in the given string.
B. It is a built-in function.
C. It is a case-sensitive function.
D. All of the above

7. Which of the following is the correct way to print "Hello World" in PHP?
A. "Hello World";
B. write ("Hello World");
C. echo "Hello World";
D. None of the above

8. The _____________ function is used used to get the reverse version of any string.
A. strrev()
B. strrevs()
C. rev()
D. revstrr()

9. Which of the following statement(s) true regarding the abs() function in PHP?
A. It is an inbuilt function in PHP which is used to return the absolute (positive) value of a number.
B. It is identical to what we call modulus in mathematics.

C. Accepts single parameter value which holds the number whose absolute value you want to find.
D. All of the above

10. The baseconvert() function is used to convert a:
A. String argument into floats
B. String argument into numbers
C. String argument into Boolean
D. String argument into arrays

11. The ______________ function returns square root of given argument.
A. sqrt()
B. squareroot()
C. sqroot()
D. square()

12. ______________ function rounds the fractions up while __________ function rounds the fractions down.
A. floor(), ceil()
B. ceil(), floor()
C. RoundUp(), RoundDown()
D. FracUp(), FracDown()

13. PHP offers 1000's of built-in functions. The functions that offer mechanism for the developer to do some sort of mathematical calculations or similar stuff are:
A. Math functions
B. Formula functions
C. Calculate functions
D. Compute functions

14. The pi() function in PHP was introduced in
A. PHP 5 version
B. PHP 7 version
C. PHP 8 version
D. PHP 4+ version

15. Which of the following statement is/are true for pow() function in PHP?
A. It was introduced in PHP 4+ version.
B. It accepts two arguments say x and y. It calculates x raised to the power of y.
C. The return type is integer or float which depends upon the nature of the argument.
D. All of the above

Answers for Self Assessment

1.	D	2.	C	3.	B	4.	D	5.	C
6.	D	7.	C	8.	A	9.	D	10.	B
11.	A	12.	B	13.	A	14.	D	15.	D

Review Questions

1. What are String functions? Explain any two.
2. Write a PHP script to find the sum of square root of 125 and 16.
3. Discuss the significance of using the math functions in PHP.
4. Explore the concept of in-built functions taking examples from string and math functions in PHP.
5. Which functions are used to find the length of a string and concatenate it?
6. Discuss the working of the following functions:
   a. decbin
   b. dechex
   c. log()
   d. pow()
   e. pi()
7. Write a short note on:
   a. String functions in PHP
   b. Math functions in PHP

Further Readings

- The Joy of PHP: A Beginner's Guide by Alan Forbes.
- Programming PHP: Creating Dynamic Web Pages by Kevin Tatroe and Peter MacIntyre.
- Learn PHP in One Day and Learn It Well by Jamie Chan.
- Web Technologies: A Computer Science Perspective by Jackson, Pearson Education India.

Web Links

- https://www.w3schools.com/php/
- https://www.educba.com/php-math-functions/?source=leftnav
- https://www.w3schools.com/pHp/php_ref_math.asp
- https://wpshout.com/php-math-functions/

Unit 06: Arrays in PHP

CONTENTS

Objectives

After studying this unit, you will be able to:

- Learn about arrays and their different types in PHP.
- Explorethe traversal of different types of arrays.
- Know aboutthe different array functions in PHP.
- Understand converting between arrays and variables.
- Implement the sorting of arrays in PHP.

Introduction

An array is a series of elements of the same type placed in contiguous memory locations that can be individually referenced by adding an index to a unique identifier. The arrays in PHP are a type of data structure that allow us to store multiple elements of similar data type under a single variable thereby saving us the effort of creating a different variable for every data.

Arrays are helpful to create a list of elements of similar types, which can be accessed using their index or key. For example, we can store 5 values of type int in an array without having to declare 5 different variables, each one with a different identifier. Instead of that, using an array we can store 5 different values of the same type, int for example, with a unique identifier.

Arrays can be used to hold multiple values of similar type in a single variable.Suppose we want to store five names and print them accordingly. This can be easily done by the use of five different string variables. But if instead of five, the number rises to a hundred, then it would be really difficult for the user or developer to create so many different variables.

Here, array comes into play and helps us to store every element within a single variable and also allows easy access using an index or a key. PHP array is an ordered map (contains value on the basis of key).

array () function is used to create a PHP array. This function can be used to create indexed arrays or associative arrays. PHP arrays could be single dimensional or multi-dimensional.The array() function was first introduced as part of core PHP 4.0.0. The syntax to create PHP indexed arrays:

```
$a = array(value1, value2, value3, ...)
```

The syntax to create PHP associative arrays:

```
$a = array(key1 => value1, key2 => value2...)
```

Where

- Key (Optional): Specifies the key, of type numeric or string. If not set, an integer key is generated, starting at 0.
- Value (Required): Specifies the value.

6.1 Storing Data in Arrays

Arrays are used to store data. The scalar variables are also the basis for assigning values to elements of a array. Using the array name as well as the index in parentheses, reference the values according to the operator (=) followed by values, then pass these values on to the assignment operator (=).

Caution: Storing a value in an array will create the array if it didn't already exist, but trying to retrieve a value from an array that hasn't been defined won't create the array.

An easier way to initialize an array is to use the array() construct, which builds an array from its arguments. This builds an indexed array, and the index values (starting at 0) are created automatically. In order to create an associative array with array(), use the => symbol to separate indices (keys) from values.You can specify an initial key with => and then a list of values.The values are inserted into the array starting with that key, with subsequent values having sequential keys.If the initial index is a nonnumeric string, subsequent indices are integers beginning at 0. Thus, the following code is probably a mistake. Also, in order to construct an empty array, pass no arguments to array() like $addresses = array();When we store a value in an array it will create the array if it didn't already exist, but trying to retrieve a value from an array that hasn't been defined yet won't create the array. For example:

Example:

```
<?php
// $postcode not defined before this point
echo $postcode[0];                  // prints nothing
echo$postcode;           // prints nothing
$postcode[0] = 'spam@fresherscloud.net';
echo$postcode;           // prints "Array"
?>
```

Notes: Using simple assignment to initialize an array in your program leads to code like this:

```
<?php
$postcode[0] = 'spam@fresherscloud.net';
```

```
$postcode[1] = 'abuse@example.com';
$postcode[2] = 'root@example.com';
echo $postcode[0] . "</br>";
echo $postcode[1] . "</br>";
echo $postcode[2];
?>
```

6.2 Adding Values to the End of an Array

The syntax, is used to insert more values into the end of an existing indexed array. Example:

Example:

```
<?php
$friend = array('Harish', 'Pinky');
$friend[] = 'Tushar'; // $friend[2] is 'Tushar'
echo $friend[2];
?>
```

*Output:*Tushar

This construct assumes the array's indexes are numbers and assigns elements into the next available numeric index, starting from 0.

6.3 Types of Arrays in PHP

There are three types of arrays in PHP:

- Indexed or Numeric Arrays: An array with a numeric index where values are stored linearly.
- Associative Arrays: An array with a string index where instead of linear storage, each value can be assigned a specific key.
- Multidimensional Arrays: An array which contains single or multiple arrays within it and can be accessed via multiple indices.

6.4 Indexed or Numeric Arrays

These types of arrays can be used to store any type of elements, but an index is always a number. By default, the index starts at zero. There are 2 different ways to create indexed arrays.

Way 1:

```
<?php
$cars = array("Volvo", "BMW", "Toyota");
echo "I like " . $cars[0] . ", " . $cars[1] . " and " . $cars[2] . ".";
?>
```

Output:I like Volvo, BMW and Toyota.

Way 2: Indexed arrays can be created in two different ways as shown in the following example:

```
<?php
// One way to create an indexed array
$names= array("Taran", "Gurfateh", "Seerat", "Anhad", "Ravinder");
```

```
// Accessing the elements directly
echo "Accessing the 1st array elements directly:\n";
echo $names[2], "\n";
echo $names[0], "\n";
echo $names[4], "\n";
// Second way to create an indexed array
$naming[0] = "TARAN";
$naming[1] = "GURFATEH";
$naming[2] = "SEERAT";
$naming[3] = "ANHAD";
echo "<br/>";
// Accessing the elements directly
echo "Accessing the 2nd array elements directly:\n";
echo $naming[2], "\n";
echo $naming[0], "\n";
echo $naming[3], "\n";
?>
```

Output:

```
Accessing the 1st array elements directly: SeeratTaran Ravinder
Accessing the 2nd array elements directly: SEERAT TARAN ANHAD
```

6.5 Associative Arrays

These types of arrays are similar to the indexed arrays but instead of linear storage, every value can be assigned with a user-defined key of string type. The associative arrays use named keys that you assign to them.There are two ways to create an associative array:

$age=array(“Taran”=>”35",”Gurfateh”=>”37",”Ravu”=>”43");

Or:

$age[‘Taran’]=“35";

$age[‘Gurfateh’]=”37";

$age[‘Ravu’]=”43";

Creating Associative Arrays

Example:

```
<?php
// One way to create an associative array
$names= array("Taran"=>"Tara", "Gurfateh"=>"Guru", "Raju"=>"Rani", "Seerat"=>"Doll",
            "Ravinder"=>"Ravu");

// Second way to create an associative array
$naming["Taran"]="Tara";
$naming["Gurfateh"]="Guru";
$naming["Raju"] = "Rani";
```

```
$naming["Seerat"] = "Doll";
$naming["Ravinder"] = "Ravu";

echo "Accessing the elements directly:\n"; // Accessing the elements directly
echo $naming["Taran"], "\n";
echo $naming["Gurfateh"], "\n";
echo $naming["Raju"], "\n";
echo $naming["Seerat"], "\n";
echo $naming["Ravinder"], "\n";
?>
```

Output:

```
Accessing the elements directly: Tara Guru Rani Doll Ravu
```

```
<?php
$age=array("Taran"=>"35","Gurfateh"=>"37","Ravinder"=>"43");
echo "Taran is " . $age['Taran'] . " years old.";
?>
```

Output:

```
Taran is 35 years old.
```

6.6 Multidimensional Arrays

Arrays that store another array at each index instead of a single element. In other words, we can define multi-dimensional arrays as an array of arrays. Every element in this array can be an array and they can also hold other sub-arrays within. The arrays or sub-arrays in multidimensional arrays can be accessed using multiple dimensions.

Example:

```
<?php
// Defining a multidimensional array
$details=array(array("name"=>"Taran","mob"=>"5678", "email"=>"taran@gmail.com", ),
array("name"=>"Gurfateh","mob"=>"1234","email"=>"gur@gmail.com",),
array("name"=>"Ravinder","mob"=>"9999","email"=>"rav@gmail.com", ));
// Accessing elements
echo "Taran email-id is: " . $details[0]["email"], "\n <br/>";
echo "Gurfateh mobile number is: " . $details[2]["mob"];
?>
```

Output:

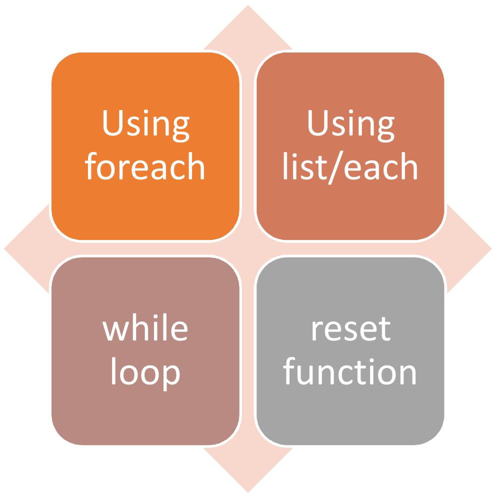
```
Taran email-id is: taran@gmail.com
Gurfateh mobile number is: 9999
```

Scope of Arrays in PHP

When declaring a regular array of local scope (within a function, for example), if we do not specify otherwise, its elements will not be initialized to any value by default, so their content will be undetermined until we store some value in them. The elements of global and static arrays, on the other hand, are automatically initialized with their default values, which for all fundamental types this means they are filled with zeros. In both cases, local and global, when we declare an array, we have the possibility to assign initial values to each one of its elements by enclosing the values in braces { }.

6.7 Traversing Arrays in PHP

PHP provides several ways to traverse arrays using both array iteration functions and language constructs: array_walk, array_map, array_filter, for each, list/each, and for loops.



As you iterate over the array, each function is used to return the current key-value pair and advance the array pointer. list function is used to assign the key and value to variables. While loop ends when the last element in the array is reached. Reset function is invoked to restore the array pointer to the first element in the array.

- **Using foreach:**PHP's foreach loop provides a convenient way to iterate over arrays. There are two forms: one uses both the key and value of each array entry while the other uses only the value. Firstly, we demonstrate the key => value form:

Example:

```
<?php
$cars = ['BMW', 'Maruti', 'Porsche', 'Mercedes' => 'XT4'];
foreach ($cars as $carrrr => $val)
{
   echo "$carrrr => $val \n";
}
?>
```

Output:

```
0 => BMW 1 => Maruti 2 => Porsche Mercedes => XT4
```

The body of the loop uses echo to display the key and value of each array element in turn. In this example, foreach uses only the value of the array elements:

Example:

```
<?php
```

```
$cars = ['BMW', 'Maruti', 'Porsche', 'Mercedes' => 'XT4'];
foreach ($cars as $val) {
echo "$val \n";          }
?>
```

Output:

BMW Maruti Porsche XT4

Using list/each: Another common way to traverse arrays uses a while loop with the list language construct and the each function. The following example achieves the same result as the first foreach example shown above:

Example:

```
<?php
$cars = ['BMW', 'Maruti', 'Porsche', 'Mercedes' => 'XT4'];
reset($cars);                          // reset array pointer
while ( list($key, $val) = each($cars) ) {
   echo "$key => $val \n";
}
?>
```

Output: 0 => BMW 1 => Maruti 2 => Porsche Mercedes => XT4

Using for: A for loop can be used to iterate over numerically indexed arrays.

Example:

```
<?php
$cars= ['BMW', 'Mercedes', 'Creta', 'Breeza'];
for ($i=0, $len=count($cars); $i<$len; $i++) {
   echo "$cars[$i] \n";
}
?>
```

Output:

BMW Mercedes Creta Breeza

6.8 <u>Traversing Different Types of Arrays in PHP</u>

1. Traversing Indexed Arrays: Indexed arrays can be used to store any type of elements, but an index is always a number. By default, the index starts at zero. We can traverse an indexed array using loops in PHP. We can loop through the indexed array in two ways. First by using for loop and secondly by using foreach.

Example:

```
<?php
// Creating an indexed array
$names= array("Taran", "Gurfateh", "Seerat", "Anhad", "Ravinder");
// One way of Looping through an array using foreach
echo "Looping using foreach: \n";
```

```
foreach ($names as $valuing) {
    echo $valuing. "\n <br/>"; }
// count() function is used to count the number of elements in an array
$countelements = count($names);
echo "\n The number of elements are $countelements \n <br/>";
// Another way to loop through the array using for
echo "Looping using for: \n";
for($n = 0; $n < $countelements; $n++){
    echo $names[$n], "\n";
}
?>
```

Output:

```
Looping using foreach: Taran
Gurfateh
Seerat
Anhad
Ravinder
The number of elements are 5
Looping using for: TaranGurfatehSeeratAnhad Ravinder
```

2. Traversing Associative Arrays: Associative arrays are similar to the indexed arrays but instead of linear storage, every value can be assigned with a user-defined key of string type. Associative arrays use named keys that you assign to them. We can traverse associative arrays in a similar way did in numeric arrays using loops. We can loop through the associative array in two ways. First by using for loop and secondly by using for each.

Example:

```
<?php
// Creating an associative array
$names= array("Taran"=>"Tara", "Gurfateh"=>"Guru", "Raju"=>"Rani", "Seerat"=>"Doll", "Ravinder"=>"Ravu");
// Looping through an array using foreach
echo "Looping using foreach: \n <br/>";
foreach ($names as $valuing => $valuing_value) {
    echo "Husband is ".$valuing." and Wife is ".$valuing_value."\n <br/>";    } //echo "<br/>";
// Looping through an array using for
echo "\nLooping using for: \n";
$keys = array_keys($namesarray_two);
$counters = count($namesarray_two);
for($i=0; $i< $counters; ++$i) {
    echo $keys[$i] . ' ' . $namesarray_two[$keys[$i]] . "\n"; }
?>
```

```
Output:
Looping using foreach:
Husband is Taran and Wife is Tara
Husband is Gurfateh and Wife is Guru
Husband is Raju and Wife is Rani
Husband is Seerat and Wife is Doll
Husband is Ravinder and Wife is Ravu
```

3. Traversing Multidimensional Arrays:An array of arrays. Every element in this array can be an array and they can also hold other sub-arrays within. Arrays or sub-arrays in multidimensional arrays can be accessed using multiple dimensions. We can traverse through the multidimensional array using for and foreach loop in a nested way. That is, one for loop for the outer array and one for loop for the inner array.

Example:

```
<?php
// Defining a multidimensional array
$multi=array("Taran"=>array("mob"=>"123","email"=>"taran@gmail.com",),
"Taran" =>array("mob" => "1234", "email" => "mount@gmail.com",   ),
"Johny" =>array("mob" => "4567", "email" => "johny@gmail.com",   ));
// Using for and foreach in nested form
$keys = array_keys($multi);
for($i = 0; $i< count($multi); $i++) {
  echo $keys[$i] . "\n";
  foreach($multi[$keys[$i]] as $key => $value) {
    echo $key . " : " . $value . "\n";   }
  echo "\n";
}
?>
```

Output:

```
Taran mob : 1234 email : mount@gmail.com Johny mob : 4567 email : johny@gmail.com
```

6.9 PHP Array Functions

- **array () Function:**array() function is used to create a PHP array. This function can be used to create indexed arrays or associative arrays.

Example:

```
1. <?php
  $a = array("Hi", "Bye", "All");
print_r($a);
?>
```

Output: Array ([0] => Hi [1] => Bye [2] => All)

2. <?php

```
  $a = array("A" => "Apple", "B" => "Banana", "C" => "Cactus");
  print_r($a);
?>
```

Output : Array ([A] => Apple [B] => Banana [C] => Cactus)

Table 1: Different Array Functions

Array Functions	Description
array()	Creates an array
array_change_key_case()	Changes all keys in an array to lowercase or uppercase
array_chunk()	Splits an array into chunks of arrays
array_column()	Returns the values from a single column in the input array
array_combine()	Creates an array by using the elements from one "keys" array and one "values" array
array_count_values()	Counts all the values of an array
array_diff()	Compare arrays, and returns the differences (compare values only)
array_diff_assoc()	Compare arrays, and returns the differences (compare keys and values)
array_diff_key()	Compare arrays, and returns the differences (compare keys only)
array_diff_uassoc()	Compare arrays, and returns the differences (compare keys and values, using a user-defined key comparison function)
array_diff_ukey()	Compare arrays, and returns the differences (compare keys only, using a user-defined key comparison function)
array_fill()	Fills an array with values
array_fill_keys()	Fills an array with values, specifying keys
array_flip()	Flips/Exchanges all keys with their associated values in an array
array_intersect()	Compare arrays, and returns the matches (compare values only)
array_intersect_assoc()	Compare arrays and returns the matches (compare keys and values)
array_intersect_key()	Compare arrays, and returns the matches (compare keys only)
array_intersect_uassoc()	Compare arrays, and returns the matches (compare keys and values, using a user-defined key comparison function)

array_intersect_ukey()	Compare arrays, and returns the matches (compare keys only, using a user-defined key comparison function)
array_key_exists()	Checks if the specified key exists in the array
array_keys()	Returns all the keys of an array
array_map()	Sends each value of an array to a user-made function, which returns new values
array_merge()	Merges one or more arrays into one array
array_merge_recursive()	Merges one or more arrays into one array recursively
array_multisort()	Sorts multiple or multi-dimensionalarrays
array_pad()	Inserts a specified number of items, with a specified value, to an array
array_pop()	Deletes the last element of an array
array_product()	Calculates the product of the values in an array
array_push()	Inserts one or more elements to the end of an array
array_rand()	Returns one or more random keys from an array
array_reduce()	Returns an array as a string, using a user-defined function
array_replace()	Replaces the values of the first array with the values from following arrays
array_replace_recursive()	Replaces the values of the first array with the values from following arrays recursively
array_reverse()	Returns an array in the reverse order

- **array_change_key_case() Function:** This function changes the case of all key of an array. Notably, it changes case of key only. It changes the case of all keys of the passed array and returns an array with all keys either in lower case or upper case based on the option passed.

Syntax: array array_change_key_case (array $array [,int $case= CASE_LOWER])

Example:

```
<?php
$cars=array("BMW"=>"550000","MERCEDES"=>"250000","CHEVROLET"=>"200000");
print_r(array_change_key_case($cars,CASE_LOWER));
?>
```

Output:

```
Array ( [bmw] => 550000 [mercedes] => 250000 [chevrolet] => 200000 )
```

- **array_chunk() Function:** By using array_chunk() method, you can divide array into many parts. The array_chunk() function takes an array as input and split that array into smaller chunks of the given size. The last chunk may contain less number of elements than passed size based on the multiplicity factor of the total numbers available in the array. It returns a multidimensional numerically indexed array, starting with zero, with each dimension containing size elements.

Syntax:

```
array array_chunk ( array $input, int $size );
```

Example:

```
1.<?php
$cars=array("BMW"=>"550000","MERCEDES"=>"250000","CHEVROLET"=>"200000", "Maruti"=>"50000");
print_r(array_chunk($cars,2));
?>
// Output: Array ( [0] => Array ( [0] => 550000 [1] => 250000 ) [1] => Array ( [0] => 200000 [1] => 50000 ) )
```

```
2. <?php
  $in = array('Hello', 'Hi', 'How', 'Are', 'You');
print_r(array_chunk($in, 3));
?>
// Output:
Array ( [0] => Array ( [0] => Hello [1] => Hi [2] => How ) [1] => Array ( [0] => Are [1] => You ) )
```

- **count() Function:** This function counts all elements in an array.

Syntax: int count(mixed $array_or_countable);

Example:

```
<?php
$courses=array("Data Structures","PHP","Python","Software Engineering", "Algorithm Designing", "C Programming", "Web Programming");
echo count($courses);  // Will display the count of the elements in an array
?>
Output: 7
```

- **sort() Function:** This function sorts all the elements in an array.

  Syntax: bool sort ($array);

Example:

```
<?php
$courses=array("Data Structures","PHP","Python","Software Engineering", "Algorithm Designing", "C Programming", "Web Programming");
sort($courses);
foreach( $courses as $c)  {
```

```
            echo "$c<br />";  }
    ?>
```

Output:

```
Algorithm Designing
C Programming
Data Structures
PHP
Python
Software Engineering
Web Programming
```

- **array_reverse() Function:** This function returns an array containing elements in reversed order.

Syntax: array array_reverse (array $array);

Example:

```
<?php
$cars=array("BMW","MERCEDES","PAJERO","DUSTER", "ECOSPORT");
$reversecars=array_reverse($cars);
foreach( $cars as $c )
{
  echo "$c <br />";
}
echo "<br/>";
foreach( $reversecars as $rc )
{
  echo "$rc<br />"; }     ?>
```

Output:

```
BMW
MERCEDES
PAJERO
DUSTER
ECOSPORT

ECOSPORT
DUSTER
PAJERO
MERCEDES
BMW
```

- **array_search() Function:** This function searches the specified value in an array. It returns key if search is successful.

Syntax: array_search(item, array $arrayname);

Example:

```
<?php
$cars=array("BMW","MERCEDES","PAJERO","DUSTER", "ECOSPORT");
$key=array_search("PAJERO",$cars);
echo $key;
?>
```

Output: 2

- **array_intersect() Function:**PHParray_intersect() function returns the intersection of two array.

  Syntax: array array_intersect (array $array1 , array $array2)

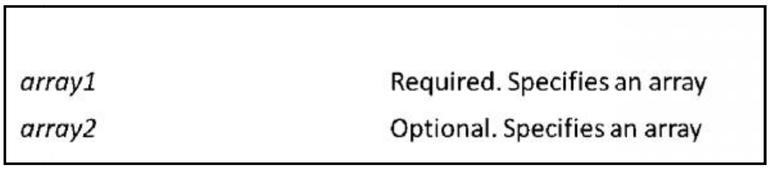
Example:

```
<?php
$a1=array("good","day","morning");
$a2=array("Good","morning");
$res=array_intersect($a1,$a2);
print_r($res);
?>
```

Output : Array ([2] => morning)

- **array_merge() Function:**Thearray_merge() function merges one or more arrays into one array.

  Syntax: array_merge(array1, array2, array3, ...)



array1	Required. Specifies an array
array2	Optional. Specifies an array

Example:

```
<?php
$arr1=array("Hello","Dear");
$arr2=array("Good","Day");
print_r(array_merge($arr1,$arr2));
?>
```

Output:

Array ([0] => Hello [1] => Dear [2] => Good [3] => Day)

- **array_unique() Function:**array_unique() function removes duplicate values from an array. If two or more array values are the same, the first appearance will be kept and the other will be removed

  Syntax: array_unique(array)

Example:

```
<?php
```

```
$arr=array("good","Day","good");
print_r(array_unique($arr));
?>
```

Output: Array ([0] => good [1] => Day)

- **array_column() Function:**array_column() function returns the values from a single column of the input array and identified by the column_key.The function array_column returns an array of values representing a single column from the input array.

  Syntax: array_column(array, column_key, index_key)

Example:

```
<?php
// An array that represents a possible record set returned from a database
$a = array (
array( 'roll' => 2,  'f_name' => 'Great',  'l_name' => 'India', ),
array(  'roll' => 4,  'f_name' => 'Hello', 'l_name' => 'Hi',),
array(  'roll' => 3,  'f_name' => 'Good',  'l_name' => 'Morning',)
);
$l_names = array_column($a, 'l_name');
print_r($l_names);
?>
```

Output: Array ([0] => Kumar [1] => Sharma [2] => Morning)

- **array_diff() Function:**This function compares array1 against one or more other arrays passed to it and returns the values in array1 that are not present in any of the other arrays.

  Syntax: array array_diff(array $array1, array $array2 [, array $array3 ...]);

Example:

```
<?php
  $arr1 = array("Black", "Red", "Green");
  $arr2 = array("Orange", "Red", "Black");
print_r(array_diff($arr1, $arr2));
?>
```

Output: Array ([2] => Green)

6.10 Converting Between Arrays and Variables

PHP provides two functions that convert between arrays and variables.

- extract()
- compact()

For instance, this array:

```
$person = array('name' => "Fred", 'age' => 35, ‘friend' => "Betty");
```

Can be converted to, or built from, these variables:

```
$name = "Fred";
```

```
$age = 35;
$wife = "Betty";
```

Creating an Array from Variables

Extract () function automatically creates local variables from an array. The indices of the array elements become the variable names:

```
$person = array('name' => "Fred", 'age' => 35, 'wife' => "Betty");
extract($person);
```

Conditions:

If a variable created by the extraction has the same name as an existing one, the variable's value is overwritten with that from the array.You can modify extract()'s behavior by passing a second argument.The most useful value is EXTR_PREFIX_ALL, which indicates that the third argument to extract() is a prefix for the variable names that are created.This helps ensure that you create unique variable names when you use extract ().

Example:

```
<?php
$shape = "round";
$array = array('cover' => "bird", 'shape' => "rectangular");
extract($array, EXTR_PREFIX_ALL, "book");
echo "Cover: {$book_cover}, Book Shape: {$book_shape}, Shape: {$shape}";
?>
```

Output:

```
Cover: bird, Book Shape: rectangular, Shape: round
```

compact() function is the reverse of extract(). It creates an associative array whose keys are the variable names and whose values are the variable's values.Any names in the array that do not correspond to actual variables are skipped.

Example:

```
<?php
$color = "indigo";
$shape = "curvy";
$floppy = "none";
$a = compact("color", "shape", "floppy");
//or
//$names = array("color", "shape", "floppy");
//$a = compact($names);
?>
```

6.11 Sorting Arrays

PHP has several functions that deal with sorting arrays.Sorting arrays is the process of arranging elements in ascending and descending. PHP provides functions to sort elements of indexed and associative arrays.

The indexed array may contain elements that can be numeric or string. The sorting of numeric elements arranged them in numeric ascending and descending. While sorting string elements arranged them in alphabetical ascending and descending orders.

The sorting of associative arrays including arranging elements according to the keys or values. There are certain pre-conditions which are:

- Some sort based on the array keys, whereas others by the values: $array['key'] = 'value';
- Whether or not the correlation between the keys and values are maintained after the sort, which may mean the keys are reset numerically (0,1,2 ...)
- The order of the sort: alphabetical, ascending (low to high), descending (high to low), natural, random, or user defined.

Notes: These entire sort functions act directly on the array variable itself, as opposed to returning a new sorted array. If any of these sort functions evaluates two members as equal then they retain their original order.

There are some in-built functions relevant to sorting in arrays:

- sort (): Sorting Indexed Arrays in Ascending Order.
- rsort(): Sorting Indexed Arrays in Descending Order.
- asort(): Sorting Associative Arrays in Ascending Order by Value.
- arsort(): Sorting Associative Arrays in Descending Order by Value.
- ksort(): Sorting Associative Arrays in Ascending Order by Key.
- krsort(): Sorting Associative Arrays in Descending Order by Key.

Sorting Indexed Arrays in Ascending Order- sort()

- **sort () Function:** You can arrange the elements of an indexed array in ascending order using the sort() in PHP.

Example:

The below example contains the alphabetical elements and the function arranges them in ascending alphabetically.

```
<?php
$vehicles = array("Cycle", "Bike", "Car", "Bolero");
//Sort Indexed Array in Ascending Order
sort($vehicles);
//Traversing elements of an indexed array
foreach ($vehicles as $veh)
{
   echo $veh;
   echo "<br>";
}
?>
```

Output:

```
Bike
Bolero
Car
Cycle
```

Example:

Similarly, you can sort the numerically indexed array element in ascending numerically using the sort() of PHP.

```
<?php
$numbers = array(1, 9, 3, 0, 13, 7);
//Sort Indexed Array in Ascending Order
sort($numbers);
//traversing elements of an indexed array
foreach ($numbers as $num)
{
   echo $num;
   echo "<br>";
}
?>
```

Output:

```
0
1
3
7
9
13
```

- **rsort() Function: The i**ndexed array can be arranged in descending order using the rsort() function in PHP.

Example:

The below example contains the indexed array with alphabetical elements. The function arranges the elements in ascending order alphabetically.

```
<?php
$vehicles = array("Cycle", "Bike", "Car", "Bolero");
//Sort Indexed Array in Descending Order
rsort($vehicles);
//traversing elements of an indexed array
foreach ($vehicles as $veh)
{
   echo $veh;
   echo "<br>";
}
?>
```

Output:

```
Cycle
Car
Bolero
Bike
```

Example:

You can perform sorting over the numerically indexed array elements in descending order. It uses the same sort() function to arrange elements in descending order numerically in PHP.

```
<?php
$numbers = array(1, 9, 3, 0, 13, 7);
//Sort Indexed Array in Descending Order
rsort($numbers);
//traversing elements of an indexed array
foreach ($numbers as $num)
{
  echo $num;
  echo "<br>";
}
?>
```

Output:

```
13
9
7
3
1
0
```

- **asort() Function:** It arranges the elements of an associative array in ascending order by value. The below example contains the associative array that uses the function for sorting:

Example:

```
<?php
$assoCars = array("Cycle" => 2, "Bike" => 5, "Car" => 9);
//Sort Associative Array in Ascending Order by Value
asort($assoCars);
//traversing elements of an associative array
foreach ($assoCars as $key => $aCars)
{
  echo "Key is: ".$key.", "."Value is: ".$aCars;
  echo "<br>";
}
?>
```

Output:

```
Key is: Cycle, Value is: 2
Key is: Bike, Value is: 5
Key is: Car, Value is: 9
```

- **arsort() Function:** You can arrange the elements of an associative array in descending order using the arsort() in PHP. The following example sorts the elements in descending according to the value as given below:

Example:

```
<?php
$assoCars = array("Cycle" => 2, "Bike" => 5, "Car" => 9);
//Sort Associative Array in Ascending Order by Value
arsort($assoCars);
//traversing elements of an associative array
foreach ($assoCars as $key => $val)
{
        echo "Key is: ".$key.", "."Value is: ".$val;
  echo "<br>";
}
?>
```

Output:

```
Key is: Car, Value is: 9
Key is: Bike, Value is: 5
Key is: Cycle, Value is: 2
```

- **ksort() Function:** The elements of an associative array can be arranged in ascending order using the ksort(). It sorts the elements in ascending order according to the key.

Example:

```
<?php
$assoCars = array("Cycle" => 2, "Bike" => 5, "Car" => 9);
//Sort Associative Array in Ascending Order by Key
ksort($assoCars);
//traversing elements of an associative array
foreach ($assoCars as $key => $val)
{
  echo "Key is: ".$key.", "."Value is: ".$val;
  echo "<br>";
}
?>
```

Output:

```
Key is: Bike, Value is: 5
Key is: Car, Value is: 9
Key is: Cycle, Value is: 2
```

Summary

- An array is a series of elements of the same type placed in contiguous memory locations that can be individually referenced by adding an index to a unique identifier.
- The arrays in PHP are a type of data structure that allow us to store multiple elements of similar data type under a single variable thereby saving us the effort of creating a different variable for every data.
- Arrays are helpful to create a list of elements of similar types, which can be accessed using their index or key.
- Indexed arrays can be used to store any type of elements, but an index is always a number. By default, the index starts at zero.
- PHP provides several ways to traverse arrays using both array iteration functions and language constructs: array_walk, array_map, array_filter, foreach, list/each, and for loops.
- Sorting arrays is the process of arranging elements in ascending and descending. PHP provides functions to sort elements of indexed and associative arrays.

Keywords

- ***Arrays:*** An array is a series of elements of the same type placed in contiguous memory locations that can be individually referenced by adding an index to a unique identifier.
- ***Indexed arrays:*** The indexed arrays can be used to store any type of elements, but an index is always a number. By default, the index starts at zero.
- ***Associative arrays:*** Associative arrays are similar to the indexed arrays but instead of linear storage, every value can be assigned with a user-defined key of string type. Associative arrays use named keys that you assign to them.
- ***Multidimensional array:*** Arrays that store another array at each index instead of a single element. In other words, we can define multi-dimensional arrays as an array of arrays.
- ***Foreach Loop:***PHP's foreach loop provides a convenient way to iterate over arrays. There are two forms: one uses both the key and value of each array entry while the other uses only the value.

SelfAssessment

1. The statement that is true for PHP arrays is

A. An array is a series of elements of the same type placed in contiguous memory locations.
B. An array can be individually referenced by adding an index to a unique identifier.
C. An array is a type of data structure that allows us to store multiple elements of similar data type under a single variable.
D. All of the above

2. By default, the index of array in php starts from ______?

A. 0
B. 1

C. -1

D. 2

3. An array which contains single or multiple arrays within it and can be accessed via multiple indices is

A. single dimensional array

B. multi-dimensional array

C. Both A and B

D. None of the above

4. ____________ construct helps to initialize an array, building it from its arguments.

A. array()

B. initarray()

C. helloarray()

D. arry()

5. The array with a string index where instead of linear storage, each value can be assigned a specific key is

A. Indexed array

B. Multi-dimensional array

C. Associative array

D. None of the above

6. Which of the following is also called as an array of arrays?

A. Multi-dimensional array

B. Single dimensional array

C. Two-way array

D. Three-way array

7. An array is a series of elements of the same type placed in _____________ memory locations that can be individually referenced by adding an index to a unique identifier.

A. Contiguous

B. Non-contiguous

C. Temporary

D. Indefinite

8. The indexed arrays used to store any type of elements have default index starting at _________.

A. One

B. Zero

C. Two

D. Infinite

9. In array traversal, the ____________ function is invoked to restore the array pointer to the first element in the array.

A. Reach
B. First
C. Reset
D. Support

10. The indexed arrays can be used to store any type of elements, but an index is always a_________.

A. String
B. Date
C. Alphanumeric
D. Number

11. Array_merge function merges __________ into one array.

A. Two arrays
B. Three arrays
C. More than three arrays
D. Merges array with a string

12. Which function displays structured information about variables including its type and value?

A. Asort( )
B. Ksort( )
C. Sort ( )
D. Var_dump( )

13. Which of the following statement represents arrays in PHP?

A. Arrays are helpful to create a list of elements of similar types.
B. Arrays can be accessed using their index or key.
C. Array is an ordered map.
D. All of the above.

14. ____________ function takes an array as input and split that array into smaller chunks of the given size.

A. array_break()
B. array_chunk()
C. array_crop()
D. array_cut()

15. Sorting of ___________ arrays including arranging elements according to the keys or values.

A. Associative

B. Numeric

C. Indexed

D. Definite

Answers forSelf Assessment

1.	D	2.	A	3.	B	4.	A	5.	C
6.	A	7.	A	8.	B	9.	C	10.	D
11.	A	12.	D	13.	D	14.	B	15.	A

Review Questions

1. Discuss the different functions used for sorting the arrays in PHP?
2. What are the asort() and rsort() functions in PHP used for?
3. Write a short note on:
    a. Sorting arrays in PHP
    b. Traversing arrays in PHP
4. What are arrays and their different types in PHP?
5. How the indexed and associative arrays are traversed in PHP?
6. Explain the traversal of indexed, associative and multidimensional arrays?

Suppose you want to add any element at the end of an array. How this can be done? Explain.

Further Readings

- The Joy of PHP: A Beginner's Guide by Alan Forbes.
- Programming PHP: Creating Dynamic Web Pages by Kevin Tatroe and Peter MacIntyre.
- Learn PHP in One Day and Learn It Well by Jamie Chan.
- Web Technologies: A Computer Science Perspective by Jackson, Pearson Education India.

Web Links

- PHP | Arrays - GeeksforGeeks
- www.w3schools.com
- PHP - Wikipedia
- www.php.net
- www.tutorialrepublic.com
- https://youtu.be/z4GE46lCv_U
- https://youtu.be/zBaHBmZLDxY
- https://youtu.be/u-KgdtNnIlI

Unit 07: Working with Databases in PHP

CONTENTS

Objectives

After studying this unit, you will be able to:

- learn about database and its role.
- explore the usage of PHP to access database.
- performing database operations using mysqli.
- explore the advanced database techniques.
- learn about the database validation techniques.

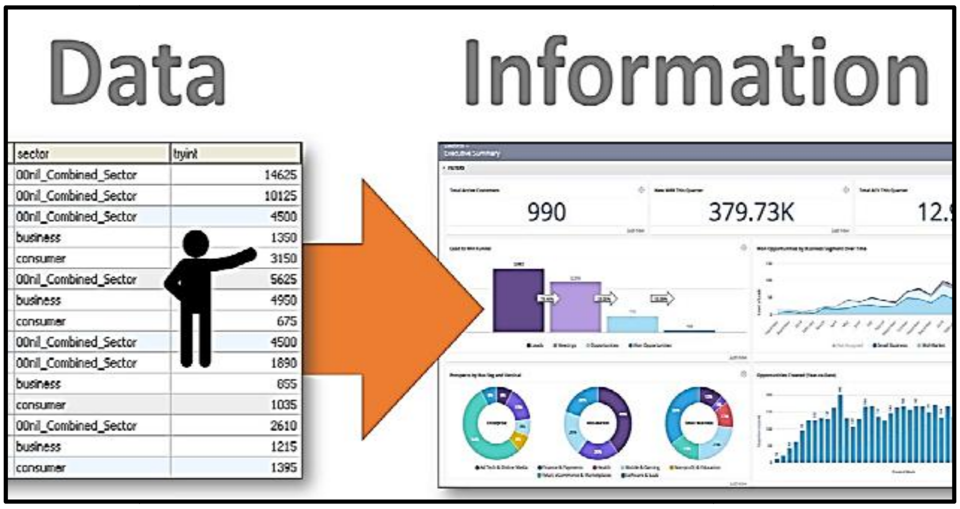
Introduction

Data are units of information, often numeric, that are collected through observation. In a more technical sense, data are a set of values of qualitative or quantitative variables about one or more persons or objects, while a datum (singular of data) is a single value of a single variable. Data is the raw material from which useful information is derived. It is defined as raw facts or observations. Data is a collection of unorganized facts but can be made organized into useful information (Figure 1). A set of values of qualitative or quantitative variables about one or more persons or objects. Data is employed in scientific research, businesses management (e.g., sales data, revenue, profits, stock price), finance, governance (e.g., crime rates, unemployment rates, literacy rates), human organizational activity (e.g., censuses of the number of homeless people by non-profit organizations).

Data is basically a collection of raw facts and figures we have a rough material that can be possessed by any computing machine that actually pertains to data. Example, we have few numbers with us, but which number it is referring to, it is unknown. We have no idea if it is specifying somebody's age, or it is specifying some currency what it is specifying we don't know. It is just a number. Okay, so actually the collection of facts, which can be in different kind of conclusions can be drawn from it that is actually called as data.

Information pertains to a systematic and meaningful form of data. It is the knowledge acquired through study or experience. The information helps human beings in their decision-making. Information can be said to be data that has been processed in such a way as to increase the knowledge of the person who uses the data.

The process of converting the facts into meaningful information.Data processing refers to the process of performing specific operations on a set of data or a database.As illustrated in **Error! Reference source not found.**1, Input one input two input three there are so many different kinds of data processing stages that are involved, like pre-processing is there storage is there, then manipulations are there analysis is there and we have so many different kinds of stages involves sorting is also involved over here, and then an output is generated so this output is the meaningful information or we can say organized data.



sector	tryint
00nil_Combined_Sector	14625
00nil_Combined_Sector	10125
00nil_Combined_Sector	4500
business	1350
consumer	3150
00nil_Combined_Sector	5625
business	4950
consumer	675
00nil_Combined_Sector	4500
00nil_Combined_Sector	1890
business	855
consumer	1035
00nil_Combined_Sector	2610
business	1215
consumer	1395

Figure 1: Conversion of Raw Data into Information

7.1 Data to Database

Data is encountered in our day-to-day life everyday. For example, the age of a person, price of potato, number of students in a class, pin code of a city etc. Such data is easy to remember all information for a few individuals. It is easy to tell accurately the age, height, complexion, income, educational qualification, residential address, etc. of your close friends. However, it is difficult to memorize all such information for a large number of individuals.

Example, every year about one lakh students take admission in National Open School (NOS). If you are asked to memorize records of date of birth, subjects offered and postal address of all these students, it will not be possible for you. In order to deal with such problems, we construct a database where in such situations, all information about students can be arranged in a tabular form. Several records can be maintained. A database helps to arrange all the information about students in a tabular form. It keeps all the records.

7.2 Database

A database is an organized collection of facts and information, such as records on employees, inventory, customers, and potential customers. Database is a system intended to organize, store, and retrieve large amounts of data easily. It consists of an organized collection of data for one or more uses, typically in digital form. One way of classifying databases involves the type of their contents, for example: bibliographic, document-text, statistical. digital databases are managed using database management systems, which store database contents, allowing data creation and maintenance, and search and other access.

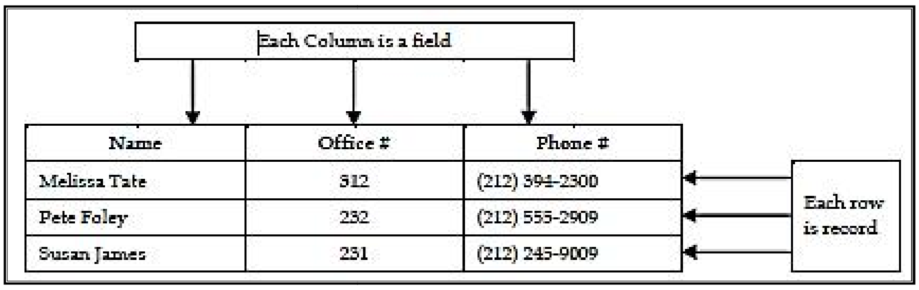
A database is a collection of information that is organized so that it can easily be accessed, managed, and updated. In one view, databases can be classified according to types of content: bibliographic, full-text, numeric, and images.

In computing, databases are sometimes classified according to their organizational approach. The most prevalent approach is the relational database, a tabular database in which data is defined so that it can be reorganized and accessed in a number of different ways. A distributed database is one that can be dispersed or replicated among different points in a network. An object-oriented programming database is one that is congruent with the data defined in object classes and subclasses. The computer databases typically contain aggregations of data records or files, such as sales transactions, product catalogs and inventories, and customer profiles. Typically, a database manager provides users the capabilities of controlling read/write access, specifying report generation, and analyzing usage.

A database is typically designed so that it is easy to store and access information. A good *database is crucial* to any company or organization. This is because the **database** stores all the important details about the company such as employee records, transactional records, salary details.Database management systems are essential for businesses because they offer an efficient way of handling large amounts and multiple types of data. The ability to access data efficiently allows companies to make informed decisions quicker.

7.3 Database Organization

A database is a place to store information. That information could be sales figures, an inventory of computers you own, a list of key customers, timesheet information; the possibilities are endless. By having the database on your site, you can easily share the information with people who need access to it. The data in a database is organized as Tables. The best way to think of a database is as a table(Figure 2). The columns in the table are called fields. There are many types of fields you can create: text, numeric, currency, names, dates, to name a few. The rows in the table are called records. The records contain all the database information.



Name	Office #	Phone #
Melissa Tate	312	(212) 394-2300
Pete Foley	232	(212) 555-2909
Susan James	231	(212) 245-9009

Figure 2: Database as a Table

7.4 Database Views

A database is much more powerful than a simple table. With a table, you can only look at the information in the one way that the table is designed. A database allows you to create many views of your information. Each view allows you to:

- o Select the fields displayed in the view.
- o Pick the order in which the fields appear.
- o Set up a filter to control which records appear in the view.
- o Sort the order in which the records appear.
- o Total up fields in the view.
- o View fields from more than one database.

A database also allows you to control how views are printed. You can create and print reports using a number of professionally designed templates. The real power of your database is that it is located in your web office. This allows you to easily share the information in the database with any member of your site, or even with guest users. It also lets other members add information to the database. Of course, you control access to all of this with database permissions.When you create your own database, you determine how the information appears, what the names of the fields are, what the views look like, and who has access to the database. If you already have information that you would like in your database, you can import that information into the database. You can also export information from your database for use in another program, such as Microsoft Excel®.

7.5 Using PHP to Access Database

There are two ways to access databases from PHP:

- Use a database-specific extension;
- Use the database-independent PEAR DB library.

MySQL extension's function names, parameters, error handling, and so on are completely different from those of the other database extensions. If you want to move your database from MySQL to PostgreSQL, it will involve significant changes to your code. The PEAR DB, on the other hand, hides the database-specific functions from you; moving between database systems.

7.6 MySQL Database

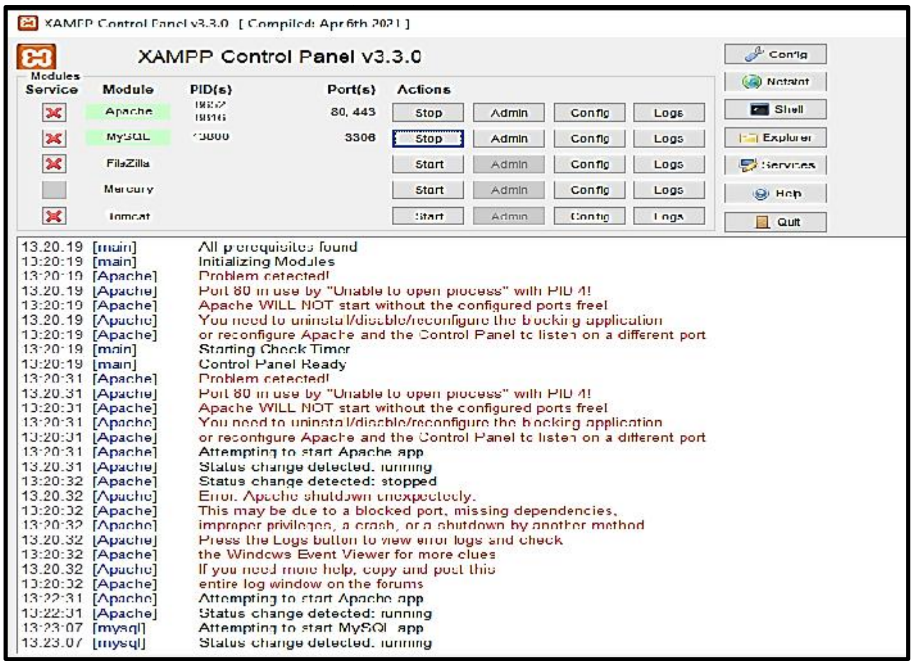
With PHP, you can connect to and manipulate databases. MySQL is the most popular database system used with PHP. MySQL is a database system used on the web that runs on a server. MySQL is ideal for both small and large applications. MySQL is very fast, reliable, and easy to use. MySQL uses standard SQL and compiles on a number of platforms. MySQL is free to download and use. It is developed, distributed, and supported by Oracle Corporation. MySQL is named after co-founder Monty Widenius's daughter: My.

The data in a MySQL database are stored in tables. PHP combined with MySQL are cross-platform (you can develop in Windows and serve on a Unix platform). PHP 5 and later can work with a MySQL database using: MySQLi extension (the "i" stands for improved).

Create a Connection to a MySQL Database

- Start XAMPP (Figure 3)
- Start MySQL
- Open browser> type localhost/phpMyAdmin

localhost / 127.0.0.1 | phpMyAdmin 5.1.1


XAMPP Control Panel v3.3.0 [Compiled: Apr 6th 2021]
XAMPP Control Panel v3.3.0
Config
Netstat
Shell
Explorer
Services
Help
Quit

```
Modules
Service  Module     PID(s)       Port(s)   Actions
         Apache     [illegible]  80, 443   Stop   Admin  Config  Logs
         MySQL      [illegible]  3306      Stop   Admin  Config  Logs
         FileZilla                         Start  Admin  Config  Logs
         Mercury                           Start  Admin  Config  Logs
         Tomcat                            Start  Admin  Config  Logs
```

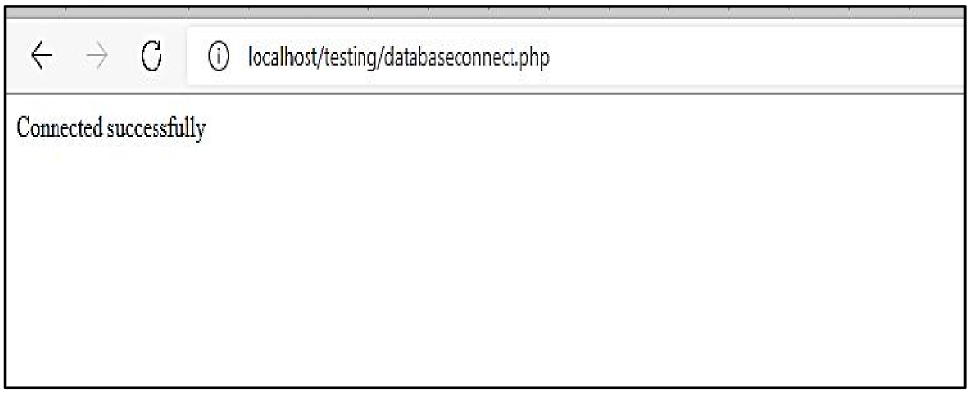
```
13:20:19  [main]     All prerequisites found
13:20:19  [main]     Initializing Modules
13:20:19  [Apache]   Problem detected!
13:20:19  [Apache]   Port 80 in use by "Unable to open process" with PID 4!
13:20:19  [Apache]   Apache WILL NOT start without the configured ports free!
13:20:19  [Apache]   You need to uninstall/disable/reconfigure the blocking application
13:20:19  [Apache]   or reconfigure Apache and the Control Panel to listen on a different port
13:20:19  [main]     Starting Check-Timer
13:20:19  [main]     Control Panel Ready
13:20:31  [Apache]   Problem detected!
13:20:31  [Apache]   Port 80 in use by "Unable to open process" with PID 4!
13:20:31  [Apache]   Apache WILL NOT start without the configured ports free!
13:20:31  [Apache]   You need to uninstall/disable/reconfigure the blocking application
13:20:31  [Apache]   or reconfigure Apache and the Control Panel to listen on a different port
13:20:31  [Apache]   Attempting to start Apache app...
13:20:31  [Apache]   Status change detected: running
13:20:32  [Apache]   Status change detected: stopped
13:20:32  [Apache]   Error: Apache shutdown unexpectedly.
13:20:32  [Apache]   This may be due to a blocked port, missing dependencies,
13:20:32  [Apache]   improper privileges, a crash, or a shutdown by another method.
13:20:32  [Apache]   Press the Logs button to view error logs and check
13:20:32  [Apache]   the Windows Event Viewer for more clues
13:20:32  [Apache]   If you need more help, copy and post this
13:20:32  [Apache]   entire log window on the forums
13:22:31  [Apache]   Attempting to start Apache app...
13:22:31  [Apache]   Status change detected: running
13:23:07  [mysql]    Attempting to start MySQL app...
13:23:07  [mysql]    Status change detected: running
```

Figure 3: XAMPP Control Panel

7.7 Connect to MySQL

The following code creates a connection to a MySQL Database through Code- "databaseconnect.php" as listed below (Figure 4):

```php
<?php
$servername = "localhost";
$username = "root";
//$password = "password";
// Create connection
//$conn = new mysqli($servername, $username, $password);
$conn = new mysqli($servername, $username, "" );
// Check connection
if ($conn->connect_error) {
die("Connection failed: " . $conn->connect_error);
}
echo "Connected successfully";
?>
```



```
localhost/testing/databaseconnect.php

Connected successfully
```

Figure 4: MySQL Database Connected Successfully

7.8 Create Database

The following code illustrates the creation of a database "Dnew" to a MySQL Database through code- "databaseCreate.php" (Figure 5).

```
<?php
//Connecting to database and then creating a database
$c1=mysqli_connect("localhost","root","");
if(!$c1)
die("not connected");
echo "connected";
$q="Create database Dnew";
if(mysqli_query($c1,$q))                {
echo "Database created";    }
else                    {
echo "database not created";            }
mysqli_close($c1);
?>
```

Notes: Check in the interface if database is created

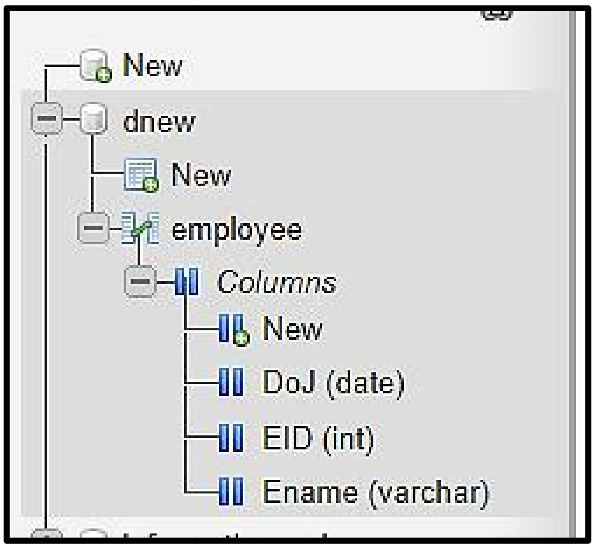


Figure 5: New Database Created "dnew"

7.9 Creating a Table in a Database

There are 2 ways to create table, either directly through browser-based MySQL admin or through code.

1. Through PHPmyadmin:
   - Click on the New Option
   - Enter the table name and columns, specifying the column names and data types (Figure 6).

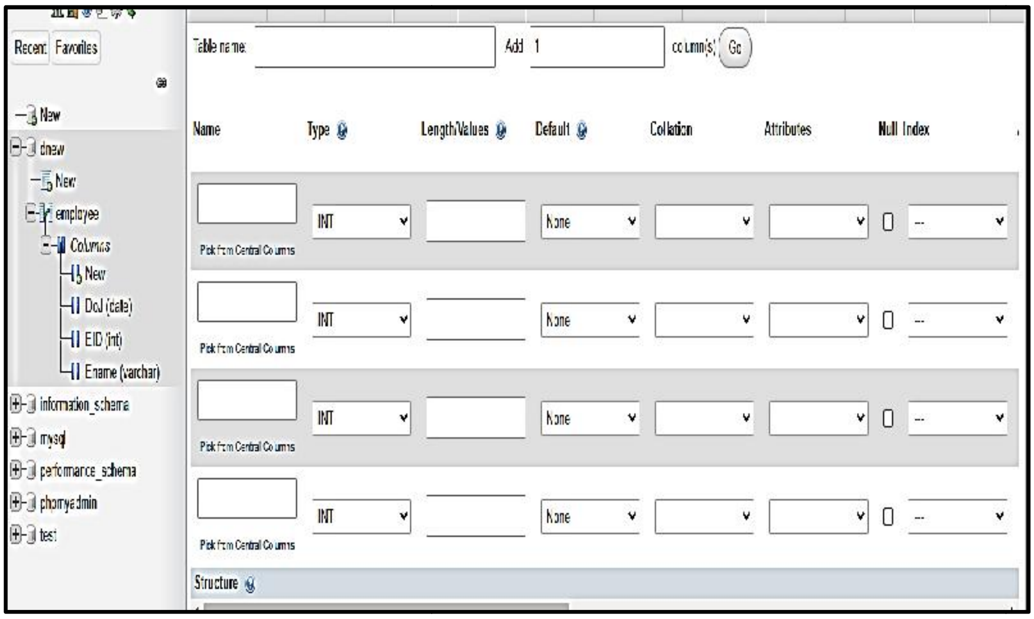
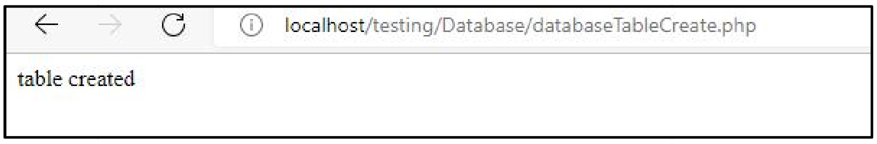


Figure 6: Table Creation a dnew Database

2. Through Code: You can create a code file in PHP as listed below:

```
<?php
$con=mysqli_connect("localhost","root","","Dnew");
$q1="Create table student1(Name varchar(20),Roll integer(4))";
if(mysqli_query($con,$q1))
```

```
{
echo “table created";
}
else
{echo "not created";
}
mysqli_close($con);
?>
```


localhost/testing/Database/databaseTableCreate.php

```
table created
```

7.10 Inserting Data in a Database

Under this, a user input can be retrieved and then the record can be inserted to the table specified.

1. Design a HTML form to get the user input: Save it as insert.html (Figure 7).
2. Design the PHP page to insert the user input.

Example:

```
<form name=f1 action="insertn.php" method="POST">
Name<input type="text" name="Name"><br>
Regno<input type="text" name="Roll"><br>
<input type="Submit" value="Add Values">
</form>
```

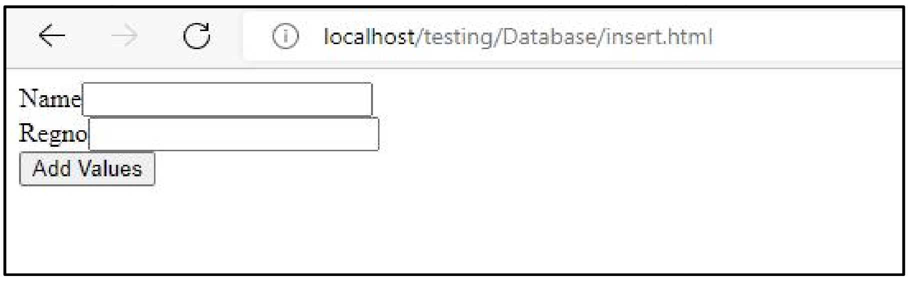


Figure 7: Form Creation to Get User Input and Insertion in the Database

Example:

```
<?php
        $n=$_POST["Name"];
        $r=$_POST["Roll"];
$c=mysqli_connect("localhost","root","","dnew"); //database name and connection
        if(!$c)
        die("cannot connect."<br>");
        echo "connected";
        echo "<br>";
//Insert record to the table student1 through the HTML form
```

```
                $s="insert into student1(Name,Roll)value('$n','$r')";
                if(mysqli_query($c,$s))     {
                echo "Record inserted";     }
                else        {
                echo"Problem in command";           }
                mysqli_close($c);
        ?>
```

7.11 Updating Records in the Database

Example:

```
<?php
$c=mysqli_connect("localhost","root","","dnew");
if(!$c)
die("cannot connect"."<br>");
echo "connected";
echo "<br>";
$s="update student1 set Name='Gurfateh' where Roll=123";
if(mysqli_query($c,$s))
{
echo "Record updated";
}
else{echo"Problem in command";}
mysqli_close($c);
?>
```

Example:Deleting the Data in Database

```
<?php
$c=mysqli_connect("localhost","root","",“dnew");
if(!$c)
die("cannot connect"."<br>");
echo "connected";
echo "<br>";
$s="delete from student1 where Roll=123";
if(mysqli_query($c,$s))
{
echo "Record Deleted";
}
else{echo"Problem in command";}
mysqli_close($c);
```

```
?>
```

Lab Exercise: Illustrate the selection of data from a database.

7.12 Advance Database Techniques

How we can connect PHP to MySQL?

- PHP 5 and later can work with a MySQL database using:
  - o MySQLi extension (the 'i' is abbreviation for improved)
  - o PDO (PHP Data Objects)

Connection to MySQL using PDO

PDO, also called "PHP Data Objects," is a PHP extension used for connecting to MySQL databases.First, create a PHP script inside the Apache web root directory for connecting to a MySQL database using PDO.Unlike MySQLi, PDO is only object-oriented and supports a number of different database types that use PHP, such as MySQL, MSSQL, Informix, and PostgreSQL.One of the most important features they both support is prepared statements, which accelerates the time needed for MySQL to execute the same query multiple times.

Which one should we use MySQLi or PDO?

Both MySQLi and PDO have their recompenses:

- o PDO will work with 12 different database systems, whereas MySQLi will only work with MySQL databases.
- o So, if you have to shift your project to use alternative database, PDO makes the process easy. You only have to change the connection string and a few queries. With MySQLi, you will need to rewrite the complete code — queries included.
- o Both are object-oriented, but MySQLi also offers a procedural API.

Connection to MySQL using PDO (testPDO.php)

Example:

```
<?php
$servername = "localhost";
$username = "root";
$password = "";
try {
$conn= new PDO("mysql:host=$servername;dbname=dnew",$username, $password);
//Set the PDO error mode to exception
$conn->setAttribute(PDO::ATTR_ERRMODE,PDO::ERRMODE_EXCEPTION);
echo "Connected successfully";        }
catch(PDOException $e) {
      echo "Connection failed: ". $e->getMessage();       }
?>
```

Output:

```
Connected succesfully
```

How to Insert Form Data into Database using PHP

Step 1: Creating a HTML Form

We are going to store data in database which is submitted through HTML form. First, we create an HTML form that need to take user input from keyboard. An HTML form is a document which stores information of a user on a web server using interactive controls. An HTML form contains different kind of information such as username, password, contact number, email id etc.The elements that are used in an HTML form are check box, input box, radio buttons, submit buttons etc. With the help of these elements, the information of a user is submitted on the web server.The form tag is used to create an HTML form.

Syntax: `<form> Form Elements... </form>`

To pass the values to next page, we use the page name with the syntax. We can use either GET or POST method to send the data to the server.

```
<form action=other_page.php  method= POST/GET>
 Form Elements...
</form>
```

Example:

```
<!DOCTYPE html>          //student1.html
<html lang="en">
<head>
<title>Student Details</title>
</head>
<body><center>
<h1>Storing Form data in Database</h1>
<form action="formDatabase.php" method="post">
<p>     <label for="Name">Student Name:</label>
        <input type="text" name="first_name" id="firstName"></p>
<p>     <label for="Roll">Roll No:</label>
        <input type="text" name="roll_no" id="roll"></p>
<input type="submit" value="Submit">
</form>
</center></body>
</html>
```

Step 2: Database Connection and Command Execution

The collection of related data is called a database. XAMPP stands for cross-platform, Apache, MySQL, PHP, and Perl. It is among the simple light-weight local servers for website development. In PHP, we can connect to database using localhost XAMPP web server.

Example:

```
<?php                           //formDatabase.php
$servername = "localhost";
$username = "root";
$password = "";
$dbname = "dnew";
// Create connection
$conn = new mysqli($servername,$username, $password, $dbname);
// Check connection
if ($conn->connect_error) {
        die("Connection failed: “ . $conn->connect_error);   }
$sqlquery = "INSERT INTO student1 VALUES           ('Seerat', 76)"
if ($conn->query($sql) === TRUE) {
        echo "record inserted successfully";
}
else {
        echo "Error: " . $sql . "<br>" . $conn->error;
}
```

Retrieving the Form Data

We can also collect the form data submitted through HTML form. The PHP $_POST method is a PHP super global variable which is used to collect data after submitting the HTML form.

Example:

```
<form name=f1 action="form.php" method="POST">//formsample.html
Name<input type="text" name="Name"><br>
Regno<input type="text" name="Roll"><br>
<input type="Submit" value="Add Values">
</form>

//Insertion Code
<!DOCTYPE html>                         //form.php
<html><head>
<title>Insert Page page</title>
</head><body><center>
<?php
// Taking all 2 values from the form data(input)
$names = $_POST["Name"];
$rolls = $_POST["Roll"];
$conn = mysqli_connect("localhost", "root", "", "dnew");
// Check connection
if(!$conn)
```

```
            die("cannot connect"."<br>");
            echo "connected <br>";          //echo "<br>";
//Performing insert query execution
$sql = "INSERT INTO student1(Name,Roll) VALUE('$names','$rolls')";
if(mysqli_query($conn, $sql)) {
echo "<h3>data stored in a database successfully.";
            echo nl2br("\n$names\n $rolls\n ");      }
else {    echo "ERROR: Hush! Sorry $sql. ". mysqli_error($conn);
 }
mysqli_close($conn); // Close connection
?>
</center></body>
</html>
```

7.13 Database Validation Techniques

Validation techniques validate the required fields, checking whether the field is filled or not in the proper way. Validation means checking the input submitted by the user. There are two types of validation are available in PHP as mentioned follow:

- Client-side Validation: Validation is performed on the client machine web browsers.
- Server-side Validation: After submitted by data, the data has sent to a server and perform validation checks in server machine.Table 1lists the validation rules.

Table 1: Some of Validation Rules for Field

Field	Validation Rules
Name	Should require letters and white-spaces
Email	Should require @ and .
Website	Should require a valid URL
Radio	Must be selectable at least once
Check Box	Must be checkable at least once
Drop Down menu	Must be selectable at least once

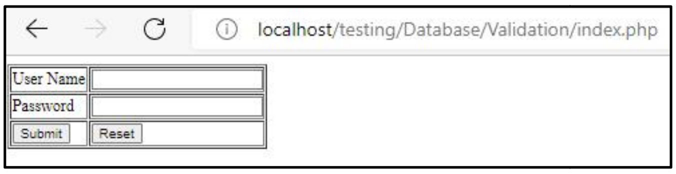
The following program involve whether a username and password entered by the user indicate if the user is valid or invalid user. Here, we will create three files, namely,

- connect.php
- index.php
- user.php

In the first case, “connect.php” will establish the connection to the database. In “index.php”, the form to obtain user input has been designed and propagates an action to user.php file where the database validation is done.

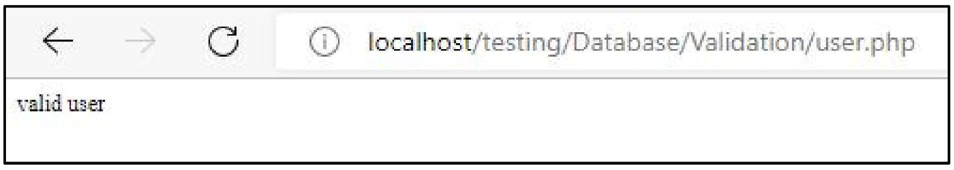
Example:

```
<?php                    //connect.php
// Create connection
$con=mysqli_connect("localhost","root","","dnew");
// Check connection
if (mysqli_connect_errno())
  {
  echo "Failed to connect to MySQL: " .mysqli_connect_error();
  }
?>

<html><head>             //index.php
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
<title>Login Form</title></head><body>
<form method="post" action="user.php" name="contactForm">
<table border="1" >
<tr>
<td><label for="username">User Name</label></td>
<td><input type="text" name="username" id="username"></td></tr>
<tr>
<td><label for="userpassword">Password</label></td>
<td><input type="password" name="userpassword" id="userpassword"></td></tr>
<tr>
<td><input type="submit" value="Submit"/>
<td><input type="reset" value="Reset"/></tr>
</table>
</form></body>
</html>
```



```
<?php                    //user.php
//Grab User submitted information
$username= $_POST["username"];
$userpassword1 = $_POST["userpassword"];
include('connect.php');          //Connect to the database
```

```
//Fetching user record from database
$result=mysqli_query($con,"select username, userpassword from users where username='$username'");
//checking if record selected
if($result === FALSE) {
die('no record fetched'.mysql_error());
}
$row =mysqli_fetch_array($result);
if($userpassword1===$row['userpassword'])
{
echo "valid user";      }
else {
echo "invalid user";         }
mysqli_close($con);
?>
```



```
valid user
```

Summary

- Data are units of information, often numeric, that are collected through observation.
- Data are a set of values of qualitative or quantitative variables about one or more persons or objects, while a datum (singular of data) is a single value of a single variable.
- Data is a collection of unorganized facts but can be made organized into useful information.
- Information pertains to a systematic and meaningful form of data. It is the knowledge acquired through study or experience.
- By having the database on your site, you can easily share the information with people who need access to it. The data in a database is organized as tables.
- MySQL is the most popular database system used with PHP. MySQL is a database system used on the web that runs on a server.
- MySQL uses standard SQL and compiles on a number of platforms. MySQL is free to download and use. It is developed, distributed, and supported by Oracle Corporation.
- PHP combined with MySQL are cross-platform (you can develop in Windows and serve on a Unix platform). PHP 5 and later can work with a MySQL database using: MySQLi extension (the "i" stands for improved).

Keywords

Data Processing:The process of converting the facts into meaningful information.Data processing refers to the process of performing specific operations on a set of data or a database.

Database:A database is a collection of information that is organized so that it can easily be accessed, managed, and updated. In one view, databases can be classified according to types of content: bibliographic, full-text, numeric, and images.

MySQL:MySQL is the most popular database system used with PHP. MySQL is a database system used on the web that runs on a server.

Validation:Validation techniques validate the required fields, checking whether the field is filled or not in the proper way. Validation means checking the input submitted by the user.

PDO:PDO, also called "PHP Data Objects," is a PHP extension used for connecting to MySQL databases.

Self Assessment

1. The process of converting the facts into meaningful information is termed as

A. Data processing

B. Data converging

C. Data analyzing

D. Data deduplication

2. A(n) ______________ database is one that can be dispersed or replicated among different points in a network.

A. optical

B. analytical

C. distributed

D. back-end

3. MySQL database is of type

A. Access Jet Database Engine

B. Object-relational DBMS

C. General Public

D. Single platform database

4. A database is defined as "A collection of ____________ data items that can be processed by one or more application programs".

A. interrelated

B. non-related

C. sampling

D. factual

5. The database classification can be done on the basis of type of their content(s) like,

A. bibliographic

B. document-text

C. statistical

D. all of the above

6. The rows in the database table are called as ____________ that contain all the database information.

A. Attributes

B. records

C. lines

D. links

7. ______________database is one that is congruent with the data defined in object classes and subclasses.

A. Convergent

B. Object-oriented programming

C. Class

D. Insecure

8. Columns in the database table are called____________.

A. records

B. aids

C. fields

D. links

9. MySQL is the most popular database system used with PHP. Which of the following statement(s) is/are true for MySQL?

A. MySQL is a database system that runs on a server

B. MySQL uses standard SQL

C. MySQL compiles on a number of platforms

D. All of the above

10. The MySQL extension's function names, parameters, error handling, and so on are completelydifferent from those of the other database extensions.

A. True

B. False

11. In MySQLi extension, the 'i' is abbreviation that stands for

A. impact

B. included

C. injected

D. improved

12. PDO is a PHP extension used for connecting to MySQL databases. Here, PDO stands for

A. PHP Data Obstacles

B. PHP Database Orientation

C. PHP Database Objects

D. PHP Data Objects

13. HTML form is a document which stores information of a user on a web server using interactive controls.

A. True

B. False

14. Which of the following statement(s) is/are true?
A. XAMPP stands for cross-platform, Apache, MySQL, PHP, and Perl.
B. XAMPP is among the simple light-weight local servers for website development.
C. In PHP, we can connect to database using localhost XAMPP web server.
D. All of the above

15. PHP ___________ method is a PHP super global variable which is used to collect data after submitting the HTML form.
A. $_POSTING
B. $_POST
C. $_POSTAGE
D. $_SUPER

Answers for Self Assessment

1.	A	2.	C	3.	B	4.	A	5.	D
6.	B	7.	B	8.	C	9.	D	10.	A
11.	D	12.	D	13.	A	14.	D	15.	B

Review Questions

1. What is PDO? Explain how MySQL connection can be done using PDO?
2. What is database? Explain the database organization.
3. How PHP can be used to access database?
4. Explain database views?
5. How to insert form data into database using PHP?
6. Discuss the database validation?
7. Illustrate through an example to validate a user credentials.
8. How do we access a database using PHP? Write the steps of database connectivity withPHP?
9. Discuss the advance database techniques.

Further Readings

- The Joy of PHP: A Beginner's Guide by Alan Forbes.
- Programming PHP: Creating Dynamic Web Pages by Kevin Tatroe and Peter MacIntyre.
- Learn PHP in One Day and Learn It Well by Jamie Chan.
- Web Technologies: A Computer Science Perspective by Jackson, Pearson Education India.

Web Links

PHP MySQL Connect to database (w3schools.com)

How to Connect Database to PHP? | Learn to Connect Database to PHP (educba.com)

Accessing SQL Server Databases from PHP - TechNet Articles - United States (English) - TechNet Wiki (microsoft.com)

https://youtu.be/Uy5pN1c9KUU

https://youtu.be/2gwXRo-FKW0

https://youtu.be/nN2mIbspaFI

Unit 08: Working with Graphics in PHP

CONTENTS

Objectives

After studying this unit, you will be able to:

- Learn about working with Graphics in PHP.
- Explore embedding an image in web page using PHP.
- Learn about creating and drawing images in PHP.
- Learn about embedding text in the images.
- Explore fonts in PHP.
- Understand the scaling of images and color handling using PHP.

Introduction

PHP makes it very easy to do many things needed on a website, among which is to embed or create an image. The ability to generate an image in PHP can be useful if you want to do things like create CAPTCHA images, or even design a banner or logo on the fly the way some free blogging software do.

There are two types of computer graphics, vector and raster. The raster graphics are the representation of images as an array of pixels. The vector graphics use geometrical primitives such as points, lines, curves or polygons to represent images. The primitives are created using mathematical equations.The Cairo Graphics Library takes a vector approach to graphics, allowing smaller size, infinite zooming, and moving, scaling, and rotating without degrading image quality.

8.1 Embedding an Image in a Page

The command to place an image is constant. You will use the same format every time. Now might be a good time to talk about where to store everything on your web server because you are starting to call for additional items to fill up your home page. Until now, all you did was put text on the page.At this point in your HTML, it is a good idea for you to place whatever images you are going to use in a subdirectory called "images". That means place the image in a directory (to be called "images") under the directory where your web pages are located (which would be the "root" directory for your site).

Format for Placing an Image

<IMG SRC="image.gif" ALT="some text" WIDTH=32 HEIGHT=32>

- <img> tag defines an image in an HTML page.
- <img> tag has two required attributes: src and alt.
- Images are not technically inserted into an HTML page, images are linked to HTML pages. The <img> tag creates a holding space for the referenced image. To link an image to another document, simply nest the <img> tag inside <a> tags.Table *1*lists the image element parameter values.

Table 1: Image Element Parameters

Attribute	Value	Description
align	top bottom middle left right	Specifies the alignment of an image according to surrounding elements
alt	text	Specifies an alternate text for an image
height	pixels	Specifies the height of an image
src	url	Specifies the url of an image
width	pixels	Specifies the width of an image
hspace	pixels	Specifies the whitespace on left and right side of an image
vspace	pixels	Specifies the whitespace on top and bottom of an image

Image Formats for the Web

Image file formats are standardized means of organizing and storing digital images. The image files are composed of digital data in one of these formats that can be rasterized for use on a computer display or printer. An image file format may store data in uncompressed, compressed, or vector formats.Table 2 lists the best file types.

Table 2: Best file types for these General Purposes

	Photographic Images	**Graphics, including Logos or Line Art**
Properties	Photos are continuous tones, 24-bit colour or 8-bit Gray, no text, few lines and edges	Graphics are often solid colours, with few colours, up to 256 colours, with text or lines and sharp edges
For Unquestionable Best Quality	TIF or PNG (lossless compression and no JPG artifacts)	PNG or TIF (lossless compression, and no JPG artifacts)
Smallest File Size	JPG with a higher Quality factor can be decent.	TIF LZW or GIF or PNG (graphics/logos without gradients normally permit indexed colour of 2 to 16 colours for smallest file size)
Maximum Compatibility (PC, Mac, Unix)	TIF or JPG	TIF or GIF

Worst Choice	256 color GIF is very limited color, and is a larger file than 24 -bit JPG	JPG compression adds artifacts, smears text and lines and edges

Major considerations to choose the necessary file type include:

- Compression quality- Lossy for smallest files (JPG), or Lossless for best quality images (TIF, PNG).
- Full RGB color for photos (TIF, PNG, JPG), or Indexed Color for graphics (PNG, GIF, TIF).
- 16-bit color (48-bit RGB data) is sometimes desired (TIF and PNG).
- Transparency or Animation is used in graphics (GIF and PNG).
- Documents- line art, multi-page, text, fax, etc. - this will be TIF.
- CMYK color is certainly important for commercial pre-press (TIF).

A common misconception is that there is a mixture of text and graphics flowing across a single HTTP request. When we view a page, we see a single page containing such a mixture. It is important to understand that a standard web page containing text and graphics is created through a series of HTTP requests from the web browser, each answered by a response from the web server. Each response can contain one and only one type of data, and each image requires a separate HTTP request and web server response. When a page that contains some text and two images, it has taken three HTTP requests and corresponding responses to construct this page.To embed a PHP-generated image in an HTML page, imagine that the PHP script that generates the image is actually the image. Thus, we can have cake.php and truck.php scripts that embed the images.

Example:

```
<html>          //image.html
<head>
<title>Example Page</title>
</head>
<body>
<h2>This page contains two images:</h2>
<imgsrc="cake.jpg" alt="Image 1" />
<imgsrc="truck.jpg" alt="Image 2" />
</body>
</html>
```

Example:

```
<?php          //image123.php
echo '
<html>
<head>
<title>Example Page</title>
</head>
<body>
<h2>This page contains two images:</h2>
<imgsrc="cake.php" alt="Cake Image" />
<imgsrc="truck.php" alt="Truck Image" />
```

```
</body>
</html>
```

In the above examples, instead of referring to real images on your web server, the <img> tags now refer to the PHP scripts that generate and return image data as indicated below:

Example:

```
<?php                //cake.php
echo '
<html>
<head>
<title>Example Page</title>
</head>
<body>
<h2>This page contains cake image:</h2>
<imgsrc="cake.jpg" />
</body>
</html>
';
?>
```

Example:

```
<?php                          //truck.php
echo '
<html>
<head>
<title>Example Page</title>
</head>
<body>
<h2>This page contains truck image:</h2>
<imgsrc="truck.jpg" />
</body>
</html>
';
?>
```

Did You Know?

It's possible to embed an image into a web page.The entire image is in the web page source code. The image can have no URL to an external file that needs to be pulled in.

Example:

```
<?php                //apple.php
```

```
echo '
<!DOCTYPE html>
<html>
<head>
<title>HTML img Tag</title>
</head>
<body>
<imgsrc="apple.jpg" alt="Seeing an Apple" width="500" height="150">
</body>
</html>
';
?>
```

Activating an Image: Turning an Image into a Link

To make an image into a click-able hyperlink, simply replace the hyperlink text with some HTML image code. Images can have a relative path (/images/sample-image.gif), or they can have an absolute path (http://hyperlinkcode.com/images/sample-image.gif).

```
<a href="http://www.hyperlinkcode.com"><imgsrc="http://hyperlinkcode.com/
images/sample-image.gif"></a>
```

Example:

```
//Embedding Image as a Link (newimageTry.php)
<?php
echo '
<!DOCTYPE html>
<html>
<head>
<title>HTML img Tag</title>
</head>
<body>
<img src="https://th.bing.com/th/id/OIP.4fAh9-
yxTtVDBQIIoWi35AHaEo?pid=ImgDet&rs=1" alt="Cloud Computing" width="500"
height="180">
</body>
</html>
';
?>
```

Example:

```
Image as a Hyperlink          //apple1.php
<?php
echo '
<html><head><title> HTML img Tag</title></head>
```

```
<body>
<a
href="https://th.bing.com/th/id/R.72638d880f4bfa8cba105e189fdda4eb?rik=H0ETBzc%2f
O8oGaQ&riu=http%3a%2f%2fpngimg.com%2fuploads%2fphp%2fphp_PNG18.png&ehk=
xcypO1gi5ecKt%2f%2btOGbfko3qA%2fucsGkdT7UikIh3cxw%3d&risl=&pid=ImgRaw&r=
0">
<img
src="https://th.bing.com/th/id/R.72638d880f4bfa8cba105e189fdda4eb?rik=H0ETBzc%2f
O8oGaQ&riu=http%3a%2f%2fpngimg.com%2fuploads%2fphp%2fphp_PNG18.png&ehk=
xcypO1gi5ecKt%2f%2btOGbfko3qA%2fucsGkdT7UikIh3cxw%3d&risl=&pid=ImgRaw&r=
0" alt="PHP Logo" width="400" height="300">
</a>
</body></html>
';
?>
```

8.2 Graphic Design (GD) Extension

GD library is used for dynamic image creation. The structure of a graphics program is as follows:

- From PHP, we use with the GD library to create GIF, PNG or JPG images instantly from our code.
- This allows us to do things such as create charts on the fly, create an anti-robot security image, create thumbnail images, or even build images from other images.
- PHP is not limited to create just HTML output.
- PHP can also be used to create and manipulate image files in a variety of different image formats, including GIF, PNG, JPEG, WBMP, and XPM.

Creating and Drawing Images

PHP can output image streams directly to a browser. You will need to compile PHP with the GD library of image functions for this to work. GD and PHP may also require other libraries, depending on which image formats you want to work with. You can use the image functions in PHP to get the size of JPEG, GIF, PNG, SWF, TIFF and JPEG2000 images.

Changing the Output Format

There are many image formats out there for many different uses. A format stores an image in a lossless or lossy format; with lossy formats you suffer some image degradation but save disk space because the image is saved using fewer bytes. A lossless format preserves the image exactly, pixel for pixel. You can break formats down into static images and movie clips.

Within either category there are standards (static formats and clip codecs) which may be proprietary standards (developed and controlled by one company), or open standards (which are community or consortium-controlled). The open standards generally outlive any one particular company and will always be royalty-free and freely obtained by the viewer. The proprietary formats may only work with a specific video card, or while the codec may be free, the viewer may cost.

As you may have deduced, generating an image stream of a different type requires only two changes to the script: send a different Content-Type and use a different image-generating function.An image is a rectangle of pixels of various colors. The colors are identified by their position in the palette, an array of colors. Each entry in the palette has three separate color values—one for red, one for green, and one for blue.Each value ranges from 0 (this color not present) to 255 (this color at full intensity). The image files are rarely a straightforward dump of the pixels and the palette.

Imagesetpixel(image, x, y, color) : Imagesetpixel( ) sets the color of a specified pixel.

Basic Drawing Functions

Instead, various file formats (GIF, JPEG, PNG, etc.) have been created that attempt to compress the data somewhat to make smaller files. The different file formats handle image transparency, which controls whether and how the background shows through the image, in different ways.

Functions for drawing lines:

Image line (image, start_x, start_ y, end_x, end_ y, color);

Image dashed line (image, start_x, start_ y, end_x, end_ y, color);

Functions for drawing rectangles

Imagerectangle(image, tlx, tly, brx, bry, color);

Imagefilledrectangle(image, tlx, tly, brx, bry, color);

Functions for drawing polygons

Image polygon (image, points, number, color);

Image filled polygon (image, points, number, color);

PHP imagecreate() Function

The imagecreate() function is an inbuilt function in PHP which is used to create a new image. This function returns the blank image of given size. The function returns the given image in a specific size. We need to define the width and height of the required image.In general, imagecreatetruecolor() function is used instead of imagecreate() function because imagecreatetruecolor() function creates high quality images. The syntax is:

Imagecreate($width, $height)

Where,

- $width: It is mandatory parameter which is used to specify the image width.
- $height: It is mandatory parameter which is used to specify the image height.

Notes: Image create() function returns the resource identifier of an image on successful execution of the program and FALSE on a failed attempt.

PHP imagecolorallocate() Function

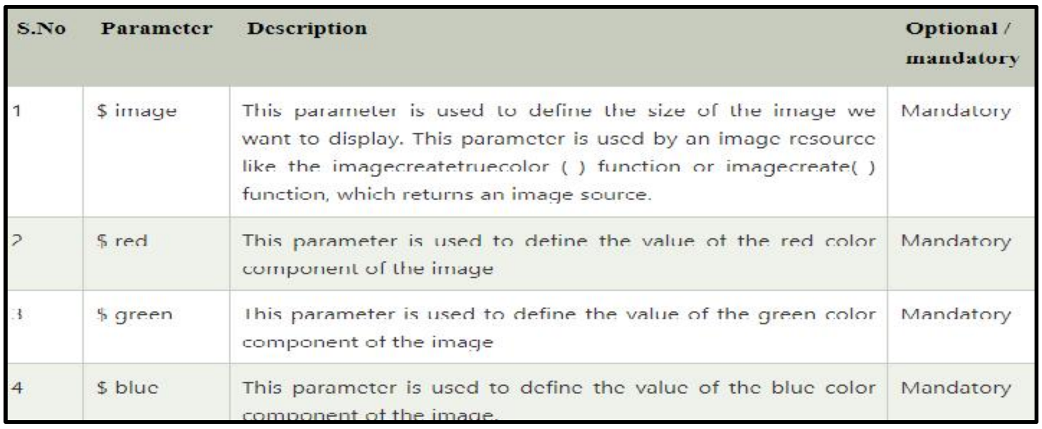
Imagecolorallocate( ) function is another inbuilt PHP function mainly used to implement a new color to an image. It returns the color of an image in an RGB format (RED GREEN BLUE).

Syntax:

Imagecolorallocate($ image , $red , $green , $blue)

This function accepts four parameters as listed in the Table 3 below:

Table 3: imagecolorallocate() Parameters



S.No	Parameter	Description	Optional / mandatory
1	$ image	This parameter is used to define the size of the image we want to display. This parameter is used by an image resource like the imagecreatetruecolor ( ) function or imagecreate( ) function, which returns an image source.	Mandatory
2	$ red	This parameter is used to define the value of the red color component of the image	Mandatory
3	$ green	This parameter is used to define the value of the green color component of the image	Mandatory
4	$ blue	This parameter is used to define the value of the blue color component of the image.	Mandatory

Notes: The image color allocate ( ) function returns the identifier of the new color on successful execution of the program. If the color of the image is not specified or does not contain any color, it will return a false on a failed attempt.

Example: Write a program making use of image color allocate (), image create true color, and image fill functions.

```
<?php                //imgAlloc.php
// to define the size of the image
$image= image create true color (400, 200);
// to define the background color of the image
$background color= image color allocate($image, 0, 150, 2);
// to fill the selected color in the background
Image fill ( $image , 0 , 0 , $background color ) ;
header('Content-Type: image/jpeg');
//The Content-Type header is used to indicate the media type of the //resource. The media type is a string sent along with the file indicating the //format of the file.
Image png($image);
Image destroy($image);
?>
```

Imagedashedline() Function

This function is used to draw a dashed line. The syntax is:

Imagedashedline (GdImage $image, int $x1, int $y1, int $x2, int $y2, int $colour)where,

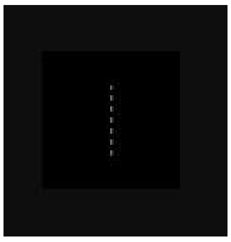
- Image: A GdImage object, returned by one of the image creation functions, such as imagecreatetruecolor().
- x1: Upper left x coordinate.
- y1: Upper left y coordinate 0, 0 is the top left corner of the image.
- x2: Bottom right x coordinate.
- y2: Bottom right y coordinates.
- Colour: The fill colour. A colour identifier created with imagecolourallocate().

Return Values: Returns true on success or false on failure.

Example: Program making use of imagedashedline() Function.

```
<?php                //dashedl.php
// Create a 100x100 image
$im = imagecreatetruecolor(100, 100);
$white = imagecolorallocate($im, 75, 150, 200);
// Draw a vertical dashed line
Imagedashedline($im, 50, 25, 50, 75, $white);
// Save the image
header('Content-Type: image/jpeg');
imagepng($im);
imagedestroy($im);
?>
```

The output is as follows:



ImageFilledRectangleFunction

This function is used to draw a filled rectangle. The syntax is:

Imagefilledrectangle(GdImage $image, int $x1, int $y1, int $x2, int $y2, int$colour)

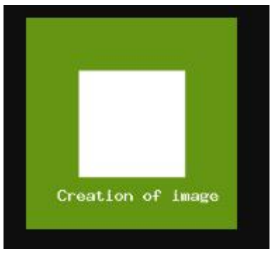
Where:

- x1: x-coordinate for point 1.
- y1: y-coordinate for point 1.
- x2: x-coordinate for point 2.
- y2: y-coordinate for point 2.

Example: PHP program to display the basic use of image colour allocate( ) with image filled rectangle( ) function

```
<?php
$img = imageCreate(200,200);
$bgcolor = imageColorAllocate($img,100,150,20);//bg
$imgcolor =imageColorAllocate($img,255,255,255);//image
ImageFilledRectangle($img,50,50,150,150,$imgcolor);
ImageString($img,5,30,160,"Creation of image",$imgcolor);
Header('Content-Type: image/png');
ImagePNG($img);
?>
```

The output for the above example is:



Imagepolygon() Function

This function is used to draw a polygon. The syntax is:

Imagepolygon(GdImage $image, array $points, int $color)

or

imagepolygon(GdImage $image, array $points, int $num_points, int $color)

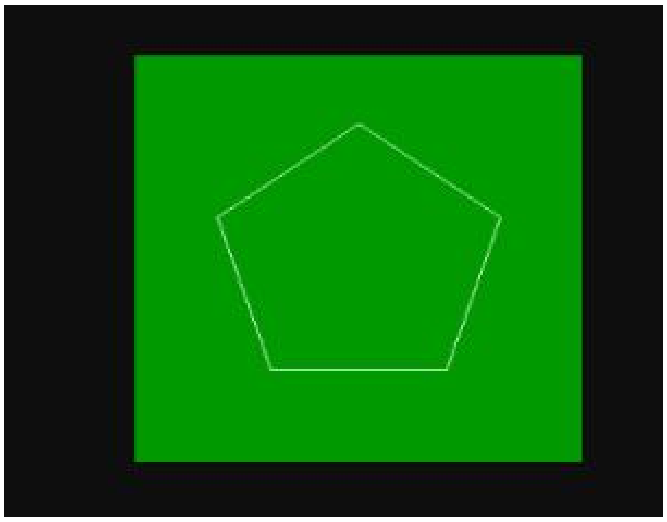
Where:

- Array $points: An array containing the polygon's vertices
- $num_points: Total number of points (vertices), which must be at least 3.

Example: PHP program to display the use of imagecreate( ) function to draw a polygon

```
<?php
// Set the vertices of polygon
$values = array( 150, 50, // Point 1 (x, y)
        55, 119, // Point 2 (x, y)
        91, 231, // Point 3 (x, y)
        209, 231, // Point 4 (x, y)
245, 119,  // Point 5 (x, y)
              290,140);
$image = imagecreatetruecolor(300, 300); // Create the size of image or blank image
$background_color = imagecolorallocate($image,  0, 153, 0); // Set background color of image
// Fill background with above selected color
imagefill($image, 0, 0, $background_color);
// Allocate a color for the polygon
$col_poly = imagecolorallocate($image, 255, 255, 255);
imagepolygon($image, $values,5, $col_poly);             // Draw the polygon
header('Content-type: image/png');                      // Output the picture to the browser
imagepng($image);                        ?>
```

The output is as follows:Here in this program, we have declared various variables like $image to define the size of the image that we require, $background color to define the color of background we require, $text color to define the color of text we require, an array $values to set the coordinates of the polygon we need to declare, and we have used image polygon ( ) function to display the polygon we want to display as an image, to display the output of the image we have used an inbuilt PHP command header and imagepng to display on browser.



Imageellipse() Function

This function is used to draw an ellipse. The syntax is:

Imageellipse(GdImage $image, int $center_x, int $center_y, int $width, int $height, int $color)

Where:

- center_x: x-coordinate of the center.

- center_y: y-coordinate of the center.
- Width: The ellipse width.
- Height: The ellipse height.

Imagefilledellipse() Function

This function draws a filled ellipse. It returns true on success or false on failure. The syntax is:

Imagefilledellipse(GdImage $image, int $centre x, int $centery, int $width, int $height, int $color)

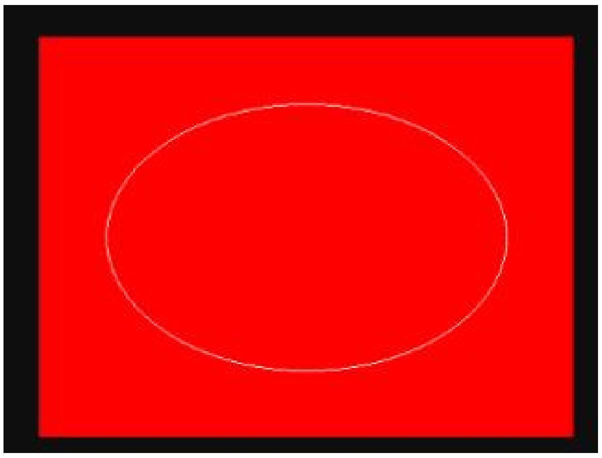
Where:

- center_x: x-coordinate of the center.
- center_y: y-coordinate of the center.
- width: The ellipse width.
- height: The ellipse height.
- image: A GdImage object, returned by one of the image creation functions, such as imagecreatetruecolor().
- Colour: The fill colour. A colour identifier created with image colour allocate().

Example: PHP program to display the use of imagefilledellipse() function

```
<?php                          //ellipse123.php
// Create a blank image.
$image = imagecreatetruecolor(400, 300);
// Select the background color.
$bg = imagecolorallocate($image, 255, 0, 0);
// Fill the background with the color selected above.
imagefill($image, 0, 0, $bg);
// Choose a color for the ellipse.
$col_ellipse = imagecolorallocate($image, 255, 255, 255);
// Draw the ellipse.
imageellipse($image, 200, 150, 300, 200, $col_ellipse);
// Output the image.
header("Content-type: image/png");
imagepng($image);
?>
```

The output is as follows:



Imagerotate() Function

This function rotates an image with a given angle. It rotates the image using the given angle in degrees. The syntax is:

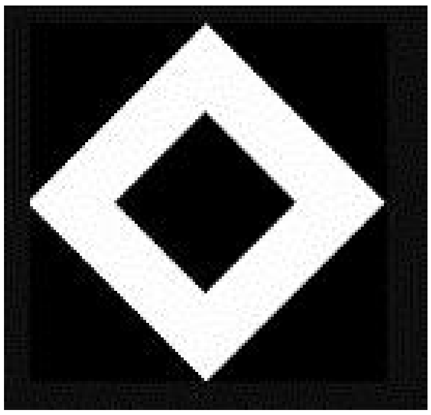
Imagerotate(GdImage $image, float $angle, int $background_color, bool $ignore_transparent = false)

Where: $ignore_transparent is an unused parameter

Example: Program illustrating the use of imagerotate() Function

```
<?php                          //imagerotate.php
$image = imagecreate(200, 200);
$white = imagecolorallocate($image, 0xFF, 0xFF, 0xFF);
$black = imagecolorallocate($image, 0x00, 0x00, 0x00);
imagefilledrectangle($image, 50, 50, 150, 150, $black);
$rotated = imagerotate($image,45,$black);
header("Content-Type: image/png");
imagepng($rotated);
?>
```

The output is as follows:



8.3 Working with Images: Adding Text

It is possible to add text to the images.Various in-built functions can be used.

Images With Text: imagestring() Function

This function draws a string horizontally. It draws a string at the given coordinates. The syntax is:

Imagestring(GdImage $image, GdFont|int $font, int $x, int $y, string $string, int $color)

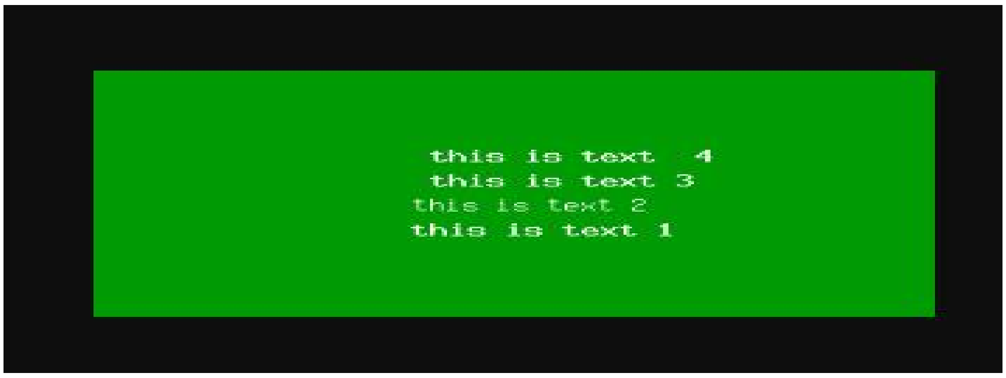
Where:

- Image: A GdImage object, returned by one of the image creation functions, such as imagecreatetruecolor().
- Font: Can be 1, 2, 3, 4, 5 for built-in fonts in latin2 encoding (where higher numbers corresponding to larger fonts) or GdFont instance, returned by imageloadfont().
- x: x-coordinate of the upper left corner.
- y: y-coordinate of the upper left corner.
- string: The string to be written.
- color: A color identifier created with imagecolorallocate().

Program using imagestring() Function //imgText.php

```
<?php
// to define the size of the image
  $image= imagecreate(400,200);
// to define the background color of the image
  $backgroundcolor= imagecolorallocate($image,0, 155, 2);
// to define the text color of the image
  $textcolor = imagecolorallocate($image,255, 255, 255);
// These functions will define the character to be displayed on the screen
Imagestring($image,13,151,121,"this is text 1", $textcolor);
Imagestring($image,4,151,101,"this is text 2", $textcolor);
Imagestring($image,10,151,81," this is text 3", $textcolor);
Imagestring($image,13,151,61," this is text  4", $textcolor);
Header("Content-Type: image/png");
Imagepng($image);
Imagedestroy($image);
?>
```

The output is as follows:



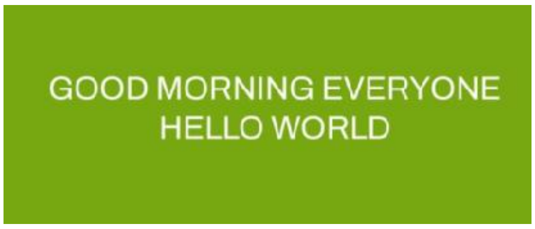
```
 this is text  4
 this is text 3
this is text 2
this is text 1
```

Here in this program, we have declared various variables like $image to define the size of the image that we require, $background color to define the color of background we require, $text color to define the color of text we require. We have used the image string ( ) function to declare the string we want to display as an image. To display the output of the image, we have used an inbuilt PHP command header and imaging to display on the browser.

Example: Program using image string() Function

```
<?php                    //imageCre.php
//to define the size of the image
$image= imagecreate(400, 200);
//to define the background color of the image
$backgroundcolor= imagecolorallocate($image, 0, 150, 2);
//to define the text color of the image
$textcolor= imagecolorallocate($image, 255, 255, 255);
//function which will define character to be displayed on screen
Imagestring($image, 5, 180, 100,  "GOOD MORNING EVERYONE", $textcolor);
Imagestring($image, 3, 160, 120,  "HELLO WORLD", $textcolor);
Header("Content - Type: image/png");
Imagepng($image);
?>
```

The output is as follows:



Here in this program, we have declared various variables like $image to define the size of the image that we require, $background color to define the color of background we require, $text color to define the color of text we require. We have used the image string ( ) function to declare the string we want to display as an image. In order to display the output of the image, we have used an inbuilt PHP command header and imagepng to display on the browser.

8.4 Fonts in PHP

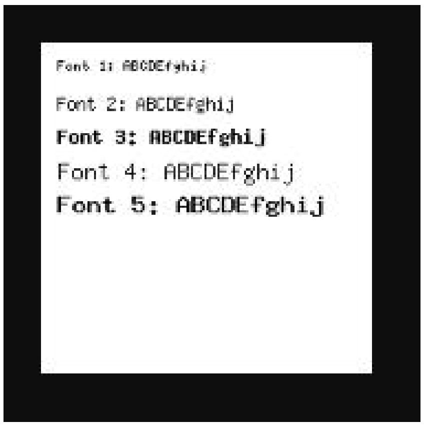
We can specify the fonts for the text added to the images.

Setting Text Fonts

Example: <?php   //imagefont.php

```
$image = imagecreate(200, 200);
$white = image color allocate($image, 0xFF, 0xFF, 0xFF); //background color by default
$black = imagecolorallocate($image, 0x00, 0x00, 0x00);
Imagestring($image, 1, 10, 10, "Font 1: ABCDEfghij", $black);
Imagestring($image, 2, 10, 30, "Font 2: ABCDEfghij", $black);
Imagestring($image, 3, 10, 50, "Font 3: ABCDEfghij", $black);
Imagestring($image, 4, 10, 70, "Font 4: ABCDEfghij", $black);
Imagestring($image, 5, 10, 90, "Font 5: ABCDEfghij", $black);
header("Content-Type: image/png");
imagepng($image);
?>
```

The output is as follows:



Setting Image Text Fonts: imagefontheight() Function

This function is used to get the font height. The syntax is:

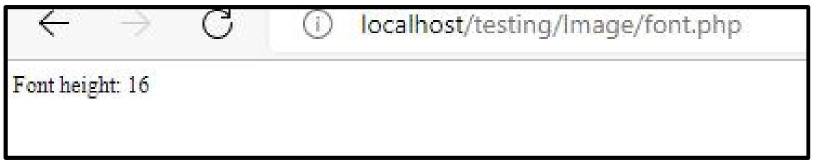
Imagefontheight(GdFont|int $font)

It returns the pixel height of a character in the specified font. It returns the pixel height of the font.

Example:

#1: Using imagefontheight() on built-in fonts

```
<?php                //font.php
        echo 'Font height: '. imagefontheight(4);
?>
```

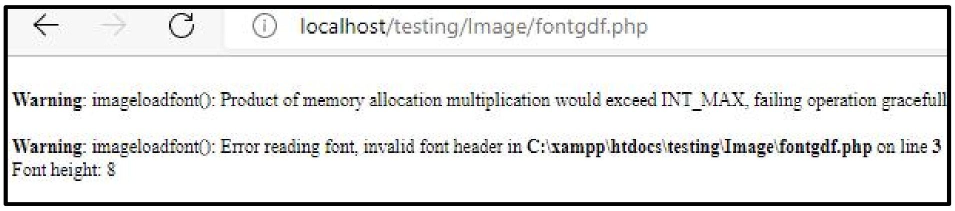


```
Font height: 16
```

Example:

#2: Using imagefontheight() together with imageloadfont()

```
<?php                        //fontgdf.php
// Load a .gdf font
$font = imageloadfont('anonymous.gdf');
echo 'Font height: ' .imagefontheight($font);
?>
```



```
Warning: imageloadfont(): Product of memory allocation multiplication would exceed INT_MAX, failing operation gracefull

Warning: imageloadfont(): Error reading font, invalid font header in C:\xampp\htdocs\testing\Image\fontgdf.php on line 3
Font height: 8
```

Image Text Fonts: imagefontwidth() Function

This function helps to get the font width. The syntax is:

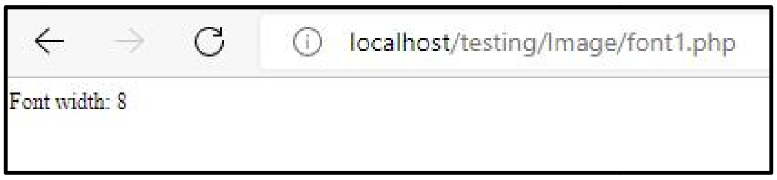
Imagefontwidth(GdFont|int $font)

It returns the pixel width of a character in font.

Example:

#1 Using imagefontwidth() on built-in fonts

```
<?php                //font1.php
echo 'Font width: ' .imagefontwidth(4);
?>
```

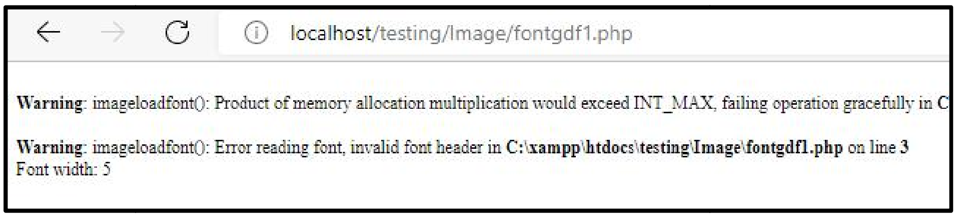


```
localhost/testing/Image/font1.php
Font width: 8
```

Example:

#2 Using imagefontwidth() together with imageloadfont()

```
<?php                          //fontgdf1.php
// Load a .gdf font
$font = imageloadfont('anonymous.gdf');
echo 'Font width: ' .imagefontwidth($font);
?>
```



```
localhost/testing/Image/fontgdf1.php

Warning: imageloadfont(): Product of memory allocation multiplication would exceed INT_MAX, failing operation gracefully in C

Warning: imageloadfont(): Error reading font, invalid font header in C:\xampp\htdocs\testing\Image\fontgdf1.php on line 3
Font width: 5
```

Images With Text- imagettftext() Function

In PHP, five fonts are built-in, and you can load additional fonts through the imageloadfont()function. The TrueType is an outline font standard; it provides more precise control over the rendering of the characters. In order to add text in a TrueType font to an image, use imagettftext() The imagettftext() function is used to write text to the image using TrueType fonts.

Syntax:

Imagettftext(GdImage $image,float $size, float $angle,int $x, int $y, int $color,string $font_filename, string $text, array $options= [])

or

imagettftext (resource $image , $size , $angle , $x , $y , $color , $fontfile , $text)

Where:

- size: The font size in points.
- angle: The angle in degrees, with 0 degrees being left-to-right reading text. Higher values represent a counter-clockwise rotation. For example, a value of 90 would result in bottom-to-top reading text.
- x: The coordinates given by x and y will define the basepoint of the first character (roughly the lower-left corner of the character). This is different from the imagestring(), where x and y define the upper-left corner of the first character. For example, "top left" is 0, 0.
- y: The y-ordinate. This sets the position of the fonts baseline, not the very bottom of the character.

- The angle in degrees, with 0 degrees being left-to-right reading text. Higher values represent a counter-clockwise rotation. For example, a value of 90 would result in bottom-to-top reading text.

8.5 Scaling Images

The images can be scaled using various in-built functions.The imagescale() function is used for scaling the images.

Imagescale() Function

- Scale an image using the given new width and height. The syntax is:

Imagescale (GdImage $image, int $width, int $height = -1,int $mode = IMG_BILINEAR_FIXED)

Where:

- Image:A GdImage object, returned by one of the image creation functions, such asimagecreatetruecolour().
- Width: The width to scale the image to.
- Height: The height to scale the image to. If omitted or negative, the aspect ratio will be preserved.
- Mode: One of IMG_NEAREST_NEIGHBOUR, IMG_BILINEAR_FIXED, IMG_BICUBIC,IMG_BICUBIC_FIXED or anything else (will use two pass).

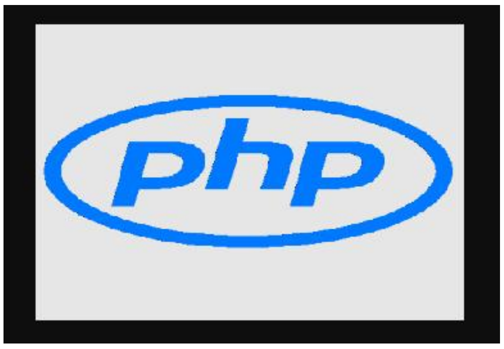
Notes: Unlike many of other image functions, image scale() does not modify the passed image; instead, a new image is returned.

Example:

```
<?php    //imagescale.php
// Load image file
$image = imagecreatefrompng('PHP1.png');  //source of image
// Use imagescale() function to scale the image
$img = imagescale($image, 500, 400 );
// Output image in the browser
header("Content-type: image/png");
imagepng($img);
?>
```

Original Image:

Scaled Image:



There are other two ways to change the size of an image: ImageCopyResized() function andImageCopyResampled() function can be used.Both the functions take the same arguments:

```
ImageCopyResized(dest, src, dx, dy, sx, sy, dw, dh, sw, sh);
ImageCopyResampled(dest, src, dx, dy, sx, sy, dw, dh, sw, sh);
```

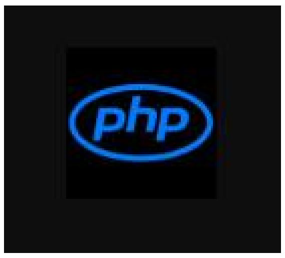
- The destand src parameters are image handles. The point (dx,dy) is the point in the destination image where the region will be copied.
- The point (sx,sy) is the upperleft corner of the source image.
- The sw, sh, dw, and dh parameters give the width and height of the copy regions in the source and destination.

Notes: The imagecopy resized () function is fast but crude, and may lead to jagged edges in your new images. The image copy resampled() function is slower, but features pixel interpolation to give smooth edges and clarity to the resized image.

Resizing with ImageCopyResampled( )

```
Example: <?php                          //scale.php
$src = ImageCreateFromPNG('PHP1.png');//source of image
//Retrieve width and height of the image
$width = ImageSx($src); //inbuilt function used to return width of given image.
$height = ImageSy($src); //inbuilt function used to return height of given image.
$x = $width/16;   //Provide the resizing dimensions
$y = $height/16;
$dst = ImageCreateTrueColor($x,$y);
ImageCopyResampled($dst,$src,0,0,0,0,$x,$y,$width,$height);
header('Content-Type: image/png');
ImagePNG($dst);
?>
```

Original Image: **After Image size has been reduced by 1/16:**



8.6 Color Handling of Images

In order to create a true color image, we can use the ImageCreateTrueColor() function:

```
Syntax: $image = ImageCreateTrueColour(width, height);
```

We can use ImagecolorAllocate() to allocate a color for an image. The syntax is:

```
Imagecolorallocate( GdImage $image, int $red, int $green, int $blue)
```

```
Example: <?php                          //color.php
$im = imagecreate(100, 100);
```

```
// sets background to red
$background = imagecolorallocate($im, 255, 0, 0);
// Save the image
header('Content-Type: image/jpeg');
imagepng($im);
imagedestroy($im);
?>
```

Another function Imagealphan blending () can be used with syntax:

Imagealphablending(resource $image, bool $blendmode)

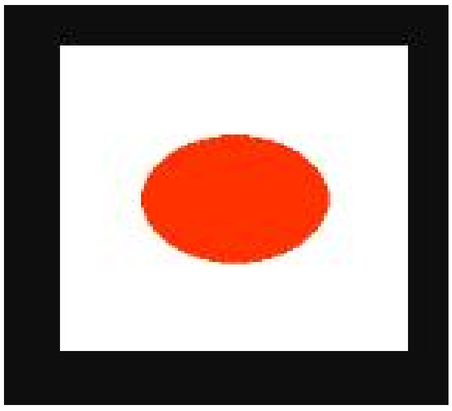
This offers two different modes of drawing on truecolor images: The blendingmode determines how much of the underlying color should be allowed to shine through.In the non-blending mode, the drawing color is copied literally with its alpha channel information, replacing the destination pixel.

Example:

Program depicting a simple orange ellipse on a white background

```
<?php                          //orange.php
$im = ImageCreateTrueColour(150,150);        //(w,h)
$white = ImageColourAllocate($im,255,255,255);                //(r,g,b)
ImageAlphaBlending($im, false);
ImageFilledRectangle($im,0,0,150,150,$white);          //(x1,y1,x2,y2)
$red=ImageColourResolveAlpha($im,255,50,0,50);       //(r,g,b,alpha)
ImageFilledEllipse($im,75,75,80,63,$red);       //(x,y,w,h,c)
header('Content-Type: image/png');
ImagePNG($im);
?>
```

The output for the above example is:



Summary

- PHP makes it very easy to do many things needed on a website, among which is to embed or create an image.

- The ability to generate an image in PHP can be useful if you want to do things like create CAPTCHA images, or even design a banner or logo on the fly the way some free blogging software do.
- PHP cleans up the image when the script ends, but, if you wish to manually deallocate the memoryused by the image, calling ImageDestroy($image)forces PHP to get rid of theimage immediately.The standard rectangular or Cartesian (named after René Descartes) has two perpendicular real number lines (axes) which divide the plane into four quadrants. The place where thenumber lines cross is the origin (0, 0) and the numbering is positive to the right and up; negative to the left and down.
- GD has an existing API, and PHP tries to follows its syntax and function-naming conventions. So, if you are familiar with GD from other languages, such as C or Perl, you can easily use GD with PHP.
- One of the most basic features of computers today is the ability to edit graphics. Manytimes, you need to build web applications that take image data from users and scale itdown to a format that can easily be displayed on your website.
- There are two types of computer graphics, vector and raster.
- Imagecolor allocate ( ) function is another inbuilt PHP function mainly used to implement a new color to an image. It returns the color of an image in an RGB format (RED GREEN BLUE).
- GD library is used for dynamic image creation. From PHP, we use with the GD library to create GIF, PNG or JPG images instantly from our code.

Keywords

Raster graphics:The raster graphics are the representation of images as an array of pixels.

Vector graphics:The vector graphics use geometrical primitives such as points, lines, curves or polygons to represent images. The primitives are created using mathematical equations.

Cairo Graphics Agreement: The Cairo Graphics Library takes a vector approach to graphics, allowing smaller size, infinite zooming, and moving, scaling, and rotating without degrading image quality.

Image:An image is a rectangle of pixels that have various colors. The colors are identified bytheir position in the palette, an array of colors.

IMG:It stands for “image.” It announces to the browser that an image will go here on the page.

WIDTH:It stands for just that, the width of the image in pixels. It can range from 1 pixel to, well, just about any number, but generally will be less than the width of the web browser.

HEIGHT:It stands for, as you might guess, the height of the image in pixels. Again, the heightcan be just about anything, but generally will be less than the height of the web browser.

SelfAssessment

1. ____________ Graphics are the representation of images as an array of pixels.

A. Raster
B. Vector
C. Line
D. Coordinate

2. ____________ graphics use geometrical primitives such as points, lines, curves or polygons to represent images.

A. Raster

B. Vector

C. Line

D. Coordinate

3. Which of the following is an attribute in the <img> tag?

A. atl

B. agt

C. sco

D. src

4. Image files are composed of digital data in one of these formats that can be rasterized for use on a computer display or printer. An image file format may store data in __________ format(s).

A. Uncompressed

B. Compressed

C. Vector

D. All of the above

5. Which of the following is/are major consideration(s) to choose the necessary file type?

A. Compression quality-

B. Transparency

C. Documents

D. All of the above

6. A standard web page containing text and graphics is created through a series of HTTP requests from the web browser, each answered by a response from the web server.

A. True

B. False

7. GD library is used for dynamic image creation. GD here stands for

A. Graph Designing

B. Graphic Design

C. Graphic Doppler

D. Graph Doppling

8. An image is a rectangle of ____________ of various colours.

A. Pixels

B. Values

C. Formats

D. Vertices

9. Imagecreatetrue color() function is used instead of imagecreate() function because imagecreatetrue color() function creates high quality images.
A. True
B. False

10. Imagecolor allocate ( ) function is another inbuilt PHP function mainly used to implement a new color to an image. It returns the color of an image in an RGB format which stands for
A. Red, Green, Black
B. Red, Green Brown
C. Red, Green, Blue
D. Red, Green, Blended

11. In the imagepolygon() function syntax as "imagepolygon(Gd Image $image, array $points, int $num_points, int $colour)", the parameter that defines the total number of points or vertices in the polygon is
A. array $points
B. $num_points
C. $image
D. $colour

12. In order to add the text in a TrueType font to an image, the function that can be used is:
A. Imagefontheight()
B. Imagettf()
C. Imagetext()
D. Imagettftext()

13. The function imagefontheight () retrieves the font height and returns the ___________ height of a character in the specified font.
A. Pixels
B. Format
C. vertice
D. upper

14. _____________ Function helps to scale an image using the given new width and height.
A. Imagescalar()
B. Scaleimage()
C. Imagescale()
D. imgscl()

15. Which of the following function(s) can be used to resize the image in PHP?
A. ImageCopyResized()
B. ImageCopyResampled()

C. Imagescale()

D. All of the above

Answers for Self Assessment

1.	A	2.	B	3.	D	4.	D	5.	D
6.	A	7.	B	8.	A	9.	A	10.	C
11.	B	12.	D	13.	A	14.	C	15.	D

Review Questions

1. What is the meaning of Graphics? How it is used with PHP? Explain with example.
2. How an image embedding in a page with PHP?
3. How an Image converts into a Link? Explain with example.
4. Write a short note on:
   a. Colour handling of images
   b. Scaling images
5. Why we use GD library? Explain the GD extensions with example
6. What are the basic concepts of graphics? Explain briefly.
7. What is the process to create and draw the images with graphics?
8. How the images embedded with text? Explain with example.
9. What is the meaning of scaling of image? How does it perform in Graphics? Explain withexample.
10. How the colour handlings proceed in graphics? Write a program to create a red rectangle on a white background.

Further Readings

- The Joy of PHP: A Beginner's Guide by Alan Forbes.
- PHP: The Complete Reference by Steven Holzner. 2017.
- Learning PHP by Ramesh Bangia, 2012.
- Learn PHP in One Day and Learn It Well by Jamie Chan.
- Learning PHP, MySQL & JavaScript with j Query, CSS & HTML5by Robin Nixon, 2015.

Web Links

- How To Embed Images In PHP Mail - Simple Examples (code-boxx.com)
- PHP: Hypertext Preprocessor
- Php | Graphics Code Examples (happycodings.com)
- Creating Simple Graphics Program in PHP (c-sharpcorner.com)
- https://blogs.oracle.com/oswald/entry/scaling_images_in_php_done

Unit 09: Working with PDF in PHP

CONTENTS

Objectives

After studying this unit, you will be able to:

- learn about the pdf extensions available in PHP.
- know regarding the pdf documents and pages.
- carry out putting text in PDF pages using PHP.
- insert images and graphics in PDF with PHP.
- perform navigation in PDF pages using PHP.
- design headers and footers in PDF using PHP.

Introduction

PDF stands for Portable Document Format. As the name implies, it is a data format that can be used to describe documents. Adobe, the developers of PDF, market software to create, edit and visualize PDF files. Because the specifications of the file format are publicly available and meanwhile even became an official ISO-standard, a lot of other companies develop PDF-related software as well. In pre-press, PDF is commonly used as a format to exchange data, either complete pages that need to be printed or advertisements that needs to be included in a publication. The file format is also popular for soft proofing and reviewing content, because there are applications that allow you to make annotations on the PDF pages.

9.1 PDF Extensions

A common problem when developing a web application is having producing a high-quality PDF out of an existing layout/view/template. Perhaps for a reporting engine, an invoice, a receipt, or any number of other situations.Often, this involves using somewhat cryptic output primitives and creating the PDF by hand.

PDF has a unique number of advantages:

- **It is a cross platform standard:** This means that somebody can create a PDF file on a Unix workstation and you can open it on a Mac or PC and still see the document just like it was intended to be viewed.
- **PDF files can be device independent:** They can be printed on a cheap ink jet printer as well as on an expensive imagesetter. This does not necessarily mean that the output will be optimized for each device. A lot depends on the way the document is created.
- **PDF files are compact**: PDF supports a number of sophisticated compression algorithms as well as a clever file structure to keep the file size of PDF files down to an absolute minimum.
- **PDF files can contain multimedia elements** like movies or sound as well as hypertext elements like bookmarks, links to e-mail addresses or web pages and thumbnail views of pages.
- **PDF supports security:** The creator of a PDF file can set various security options. It is possible to lock a PDF so it can only be opened with a password. It is also possible to forbid changing the content of a PDF or disable the option to print a PDF file.
-

Did you Know?

The PDF format is so powerful and has been adopted around the world because it acts as a container for various types of content.

Adobe's Portable Document Format (PDF) provides a popular way to get a consistent look, both on screen and when printed, for documents.PHP has several libraries for generating PDF documents.The FPDF library is a set of PHP code you include in your scripts with the require function, so it doesn't require any server-side configuration or support, meaning you can use it even without support from your host.

http://www.fpdf.org/en/download.php can be used to download the fpdf library (v 1.81, ZIP).

You need to download the FPDF class from the FPDF website and include it in your PHP script.

```
require('fpdf/fpdf.php');
```

9.2 FPDF Library

FPDF is a PHP class which allows to generate PDF files with pure PHP, that is to say without using the PDFlib library. F from FPDF stands for Free: you may use it for any kind of usage and modify it to suit your needs. FPDF has other advantages: high level functions. Here is a list:

- Choice of measure unit, page format and margins
- Page header and footer management
- Automatic page break
- Automatic line break and text justification
- Image support (JPEG, PNG and GIF)
- Colors
- Links
- TrueType, Type1 and encoding support
- Page compression

9.3 Documents and Pages in PDF

A PDF document is made up of a number of pages. Each page contains text and/or images.

FPDF('P', 'in', 'Letter');

- o P or L
- o Units of font pt, mm, ct, in
- o Paper size A3, A4, A5, Legal, Letter

Addpage(orientation, paper size, rotation)

- o P or L
- o A3, A4, A5, Letter, Legal
- o Multiple of 90. Default is 0

Cell(float w , float h , string txt , mixed border , int ln , string align , boolean fill , mixed link)

or

Cell(width, height, string, border, ln, align, fill, link)

- o W = Cell width. If 0, the cell extends up to the right margin. .
- o H = Cell height. Default value: 0.
- o Txt = String to print. Default value empty
- o Border = (1,0), (L,T,R,B)
- o In = 0 to the right, 1 beginning of next line, 2 below
- o Align = L,R,C
- o Fill = cell background to be painted true or false
- o Mixed = url returned by addlink()

Notes: Cell extends upto the right margin. The cell height. Default is zero.

String to be print. Default is empty.

It can be 0 or 1 for frame. It can L, R, T and B

Current position should go after the call 0 to the right, 1 beginning of the next line, 2 below.

Position of text L,C and R

Background of cell true or false. Default is false

Link returned by addlink()

Ln(height) line break

Height of break. Default is height of last printed cell

9.4 <u>Putting Text in PDF Pages Using PHP</u>

Text: There are many options for changing the appearance and layout of text.

Coordinates: The origin (0,0) in a PDF document with the FPDF library is in the top-left corner of the defined page.All of the measurements are specified in points, millimeters, inches, or centimeters. A point (the default) is equal to 1/72 of an inch, or 0.35 mm

Text Attributes: There are three common ways to alter the appearance of text: bold, underline, and italics.The following functions can be used:

- Setfont(string family , string style , float size);
  - o where, Family= Courier (fixed width), Helvetica or Arial (Sans Serif), Times (Serif), Symbol (Symbolic), Zapfdingbats (Symbolic)
  - o Style = empty, B, I, U
  - o Size = font size in point.
- setDrawColor(r,g,b)where,

- o Red component (0 to 255)
- o Green component (0 to 255)
- o Blue component (0 to 255)

- AliasNbPages([string alias]): Defines an alias for the total number of pages.
- Addfont() Function: Used to add a font. We have an existing library of fonts. We can add any fontto a PDF document.

  Addfont(string family , string style , string file)

  - o where, Family = font family name.
  - o Style = empty, B, I, U
  - o File = The font definition file.
- Settextcolor()

  Settextcolor(int r , int g, int b)

  - o r = red component (0 – 255)
  - o g = green component (0 – 255)
  - o b = blue component (0 – 255)
- setFillColor(r,g,b)
  - o Red component (0 to 255)
  - o Green component (0 to 255)
  - o Blue component (0 to 255)

Example:
```
<?php                                //pdf1.php
require("../fpdf184/fpdf.php");      // path to fpdf.php
//reference to the FPDF library with the require function
$pdf = new FPDF();         //created a new instance of the FPDF object.
$pdf->addPage(); //After you have created new instance of FPDF object, you will need to add
//at least one page to object, so the AddPage method is called.
$pdf->setFont("Arial", 'B', 16);     //Text style as Bold
//You need to set the font for the output you are about to generate with the //SetFont call
$pdf->cell(40, 10, "Hello Out There!");
//using the cell method call, you can place the output on your created document.
$pdf->output();    //To send all your work to the browser, simply use the Output method.
?>
```

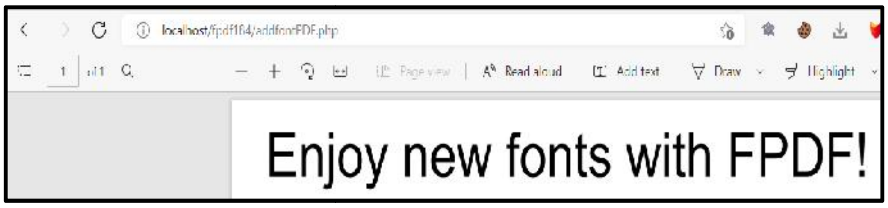
Example:
```
<?php                      //pdf3.php
  require("../fpdf184/fpdf.php");
  $pdf = new FPDF();
  $pdf->AddPage();
  $pdf->SetFont("Arial", "I", 12);     //Text style as Italic
  $pdf->Cell(40, 10, "Hello World");
  $pdf->Output();
?>
```

Example:<?php //pdf2.php

```
require('../fpdf184/fpdf.php');
// Instantiate and use the FPDF class
$pdf = new FPDF();
//Add a new page
$pdf->AddPage();
// Set the font for the text
$pdf->SetFont('Arial', 'U', 18);          //Text as Underline
// Prints a cell with given text
$pdf->Cell(60,20,'Hello GeeksforGeeks!');
// return the generated output
$pdf->Output();
?>
```

Example: Putting Text in PDF Pages Using PHP- Example for addfont() (Figure 1).

```
<?php                //addfontPDF.php
//define('FPDF_FONTPATH', '.');
require '../fpdf184/fpdf.php';
$pdf = new FPDF();
//$pdf->addFont('Calligrapher', '', 'calligra.php');
//$pdf->addFont('courier', '', 'courier.php');
$pdf->addFont('helvetica', '', 'helvetica.php');
$pdf->addPage();
//$pdf->setFont('courier', '', 35);
$pdf->setFont('helvetica', '', 45);
$pdf->cell(0, 10, 'Enjoy new fonts with FPDF!');
$pdf->output
();
?>
```



Enjoy new fonts with FPDF!

Figure 1: Adding Fonts in PDF

Example: Putting Text in PDF Pages Using PHP- Example for settextcolor() andsetfillcolor()

```
<?php                //fillcolo.php
```

```
require('../fpdf184/fpdf.php');
// Instantiate and use the FPDF class
$pdf = new FPDF();
//Add a new page
$pdf->AddPage();
// Set the font for the text
$pdf->settextcolor(255,0,0);
$pdf->SetFont('Arial', 'U', 18);          //Text as Underline
$pdf->setfillcolor(255,255,0);
// Prints a cell with given text
$pdf->Cell(60,20,'Hello World',0,0,0,1);
// return the generated output
$pdf->Output();
?>
```

9.5 Inserting Images and Graphics in PDF

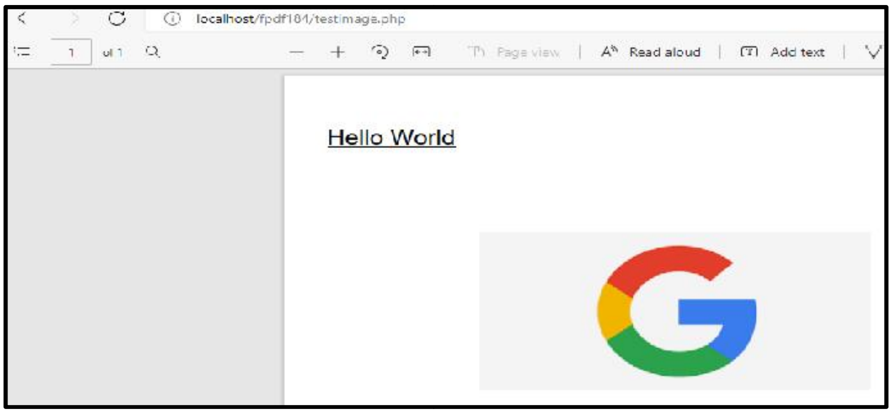
PHP helps to add images and graphics to the PDF pages using the inbuilt function such as image().

Syntax: image(file, x, y, w, h, type, link)

- o Path or URL of JPEG, GIF, PNG
- o Upper left corner
- o Upper left corner
- o Width of image
- o Height of image
- o Format JPEG, JPG, GIF and PNG
- o URL or identifier returned by addlink()

Example: Program to insert images and graphics in PHP (Figure 2).

```
<?php                //testimage.php
require('../fpdf184/fpdf.php');
$pdf = new FPDF();           // Instantiate and use the FPDF class
$pdf->AddPage();             //Add a new page
$pdf->SetFont('Arial', 'U', 18);          // Set the font for the text
$pdf->Cell(60,20,'Hello World');       // Prints a cell with given text
$pdf->image("google.png", 50, 50, 100, 0, '  ', "http://www.google.com");
            //Syntax: image(file, x, y, w, h, type, link)
$pdf->ln(20);
$pdf->output();    //return the generated output
?>
```



Hello World

Figure 2: Inserting Image in PHP

9.6 Navigation in PDF Pages Using PHP

It is possible to navigate across PDF pages as well. The following functions can be used:

- o addLink(): Creates an internal link and returns its identifier.
- o setLink(link, y, page)

The link identifier returned by AddLink(). The coordinate of target position; -1 indicates the current position. The default value is 0 (top of page). The number of target page; -1 indicates the current page. This is the default value.

Example: Program to navigate in PDF pages using PHP.

```
<?php                    //samplenavi.php
require("../fpdf184/fpdf.php");
$pdf = new FPDF();
$pdf->addPage();      //First page
$pdf->setFont("Times", 'B', 14);
$pdf->write(5, "For a link to the next page - Click");
$pdf->setFont('Times', 'U');
$pdf->setTextColor(0, 0, 255);
$linkToPage2 = $pdf->addLink();
$pdf->write(5,"Please click here", $linkToPage2);
            //write(h, txt, link) wher line height, String or text to print &Url or identifier returned by addlink()
$pdf->setFont('Times',"U");
$pdf->addPage();      //Second page
$pdf->setLink($linkToPage2);
$pdf->image("google.png", 50, 50, 100, 0, '', "http://www.google.com");
$pdf->ln(20);
$pdf->setTextColor(1);
$pdf->cell(0, 5, "Click the following link, or click on the image", 0, 1, 'L');
$pdf->setFont('', 'U');
$pdf->setTextColor(0,0,255);
```

```
    $pdf->write(5, "Social Networking", "http://www.facebook.com");
    $pdf->output();
    ?>
```

9.7 Creating a PDF with Header and Footer

FPDF is actually a class definition provided for your use and extension.There are two methods: header() and footer() that are used to add header and footer in a PDF. The AliasNbPages() method is used to track the overall page count in the PDF document before it is sent to the browser.The FPDF library also allows you to create a custom header and footer for your PDF files.

Example: Program for adding Footer.

```
<?php                //footer123.php
require("../fpdf184/fpdf.php");
class PDF extends FPDF     {
function Footer()  {
  // Go to 1.5 cm from bottom
  $this->SetY(-15);          //Function in Footer specifying the position
  $this->SetFont('Arial','I',8);                    // Select Arial italic 8
  // Print current and total page numbers
  $this->Cell(0,10,'Page '.$this->PageNo().'/{nb}',0,0,'C');
}}
$pdf = new PDF();
$pdf->AliasNbPages();
$pdf->AddPage();
$pdf->Output();
?>
```

Example: The following code illustrates creating a PDF file with a header and footer.

```
<?php                          //headerfooter.php
  require ("../fpdf184/fpdf.php");
  class PDF extends FPDF{
    function Header()       {
      $this->SetFont("Arial", "B", 24);
      $this->Cell(80);
      $this->Cell(40, 10, "Heading");
      $this->Ln(25);       }
    function Footer()       {
      $this->SetY(-15);
      $this->SetFont("Arial", "U", 10);
      $this->Cell(0,10 "Page: ".$this->PageNo() . "/{nb}",0,0,"C");       }   }
  $pdf = new PDF();
```

```
    $pdf->AliasNbPages();
    $pdf->AddPage();
    $pdf->SetFont("Arial", "I", 12);
    $pdf->Output();
?>
```

The above example code generates a PDF file with a header and footer as specified.

Example: Program to create PDF with a header and footer (Figure 3).

```
<?php                              //ihf.php
require('fpdf.php');
class PDF extends FPDF     {
function Header(){
   $this->Image('logo.png',10,50,50);
    $this->SetFont('Arial','B',15);
    $this->Cell(80);
    $this->Cell(30,10,'Title',1,0,'C');
    $this->Ln(20);   }
function Footer()  {
    $this->SetY(-20);
    $this->SetFont('Arial','I',8);
    $this->Cell(0,10,'Page '.$this->PageNo().
             '/{nb}',0,0,'C');       }}
$pdf = new PDF();
$pdf->AliasNbPages();
$pdf->AddPage();                          //Page 1
$pdf->SetFont('Times','',12);
$pdf->Cell(30,10,'page 1',1,0,'C');
$pdf->AddPage();                          //Page 2
$pdf->SetFont('Times','',12);
$pdf->Cell(30,10,'page 2',1,0,'C');
$pdf->Output();
?>
```

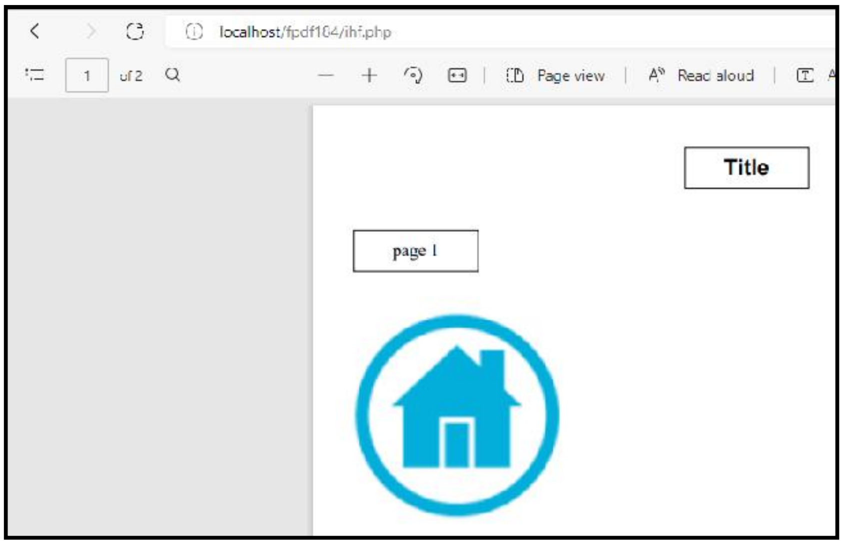


Figure 3: PDF with a header and footer

Summary

- PHP makes it very easy to do many things needed on a website, among which is to embed or create an image in a PDF file.
- PHP helps to add images and graphics to the PDF pages using the inbuilt function such as image().
- Adobe's Portable Document Format (PDF) provides a popular way to get a consistent look, both on screen and when printed, for documents.
- PHP has several libraries for generating PDF documents.
- The FPDF library is a set of PHP code you include in your scripts with the require function, so it doesn't require any server-side configuration or support, meaning you can use it even without support from your host.
- It is possible to navigate across PDF pages as well using functions like addlink() that helps to create an internal link and returns its identifier.

Keywords

FPDF:FPDF is a PHP class which allows to generate PDF files with pure PHP, that is to say without using the PDFlib library.

Cross platform standard:Using cross platform support, one can create a PDF file on a Unix workstation and you can open it on a Mac or PC and still see the document just like it was intended to be viewed.

AliasNbPages(): AliasNbPages() method is used to track the overall page count in the PDF document before it is sent to the browser.

PDF:PDF stands for Portable Document Format. As the name implies, it is a data format that can be used to describe documents. Adobe, the developers of PDF, market software to create, edit and visualize PDF files.

SelfAssessment

1. PDF stands for ______________.

A. Portable Document File

B. Portable Document Format
C. Portable Design Format
D. Portable Design File

2. ______________ is a data format that can be used to describe documents.
A. FPDF
B. TPDF
C. XPDF
D. PDF

3. Which of the following is/are true statement(s) for PDF?
A. PDF is commonly used as a format to exchange data, either complete pages that need to be printed or advertisements that needs to be included in a publication.
B. popular for soft proofing and reviewing content, because there are applications that allow you to make annotations on the PDF pages.
C. Adobe is the developer of PDF and has capability to market software to create, edit and visualize PDF files.
D. All of the above

4. Which of the following is NOT an advantage of PDF?
A. PDF files can be device independent
B. PDF files can contain multimedia elements
C. PDF does not support cross platform standard
D. PDF supports security

5. ______________ is a set of PHP code you include in your scripts with the require function, so it doesn't require any server-side configuration or support, meaning you can use it even without support from your host.
A. FPDF library
B. XPDF library
C. Workgroup library
D. Session library

6. The addpage(orientation, paper size, rotation) is used for adding page in a PDF using PHP. The orientation attribute here can take ______________ layouts.
A. portrait
B. landscape
C. both
D. none of the above

7. In the setDrawColor(r,g,b) function, the "rgb" stands for
A. Red, golden, black
B. Red, green, brown
C. Red, green, blue
D. Red, green, black

8. The AliasNbPages([string alias]) defines
A. an alias for the margin of PDF pages.
B. an alias for the total number of pages.
C. an alias for orientation of PDF pages.
D. an alias for the layout of PDF pages.

9. In the function, Settextcolor(int r , int g, int b), the minimum and maximum value taken by 'r', 'g' or 'b' can be
A. 0 and 255
B. 1 and 255
C. 0 and 125
D. 1 and 125

10. Which of the following is/are way(s) to alter the appearance of the text?
A. Bold
B. Underline
C. Italics
D. All of the above

11. ____________ function creates an internal link and returns its identifier.
A. setlink()
B. addLink()
C. internlink()
D. linker()

12. _______________ method is used to track the overall page count in the PDF document before it is sent to the browser.
A. AliasNbPages()
B. trackPages()
C. PageCounter()
D. trackCount()

13. PHP helps to add images and graphics to the PDF pages using the inbuilt function such as _______________.
A. imageSetting()
B. setImage()
C. imagePixel()
D. image()

14. In PHP, it is not possible to navigate across PDF pages.
A. True
B. False

15. The syntax, image(file, x, y, w, h, type, link) is used to insert image in a PDF. The type attribute can take _________ files.
A. JPG

B. GIF
C. PNG
D. All of the above

Answers for Self Assessment

l.	B	2.	D	3.	D	4.	C	5.	A
6.	C	7.	C	8.	B	9.	A	10.	D
11.	B	12.	A	13.	C	14.	B	15.	D

Review Questions

1. What is the meaning of Graphics? How it is used with PHP? Explain with example.
2. What is the mean of PDF? What are the PDF files and why these are used?
3. What are the uses of PDF documents and pages? How do these create?
4. Discuss about putting text in PDF.
5. What is the importance of font in PDF? How does it embed in it?
6. How do we use images and graphics in PDF?
7. Write a PHP program to place an image in several places on a page.
8. What is the meaning of navigation? How PDF provide navigation?
9. Write a short note on:
   (a) Headers and Footers in PHP
   (b) FPDF Library

Further Readings

The Joy of PHP: A Beginner's Guide by Alan Forbes.

PHP: The Complete Reference by Steven Holzner. 2017.

Learning PHP by Ramesh Bangia, 2012.

Learn PHP in One Day and Learn It Well by Jamie Chan.

Learning PHP, MySQL & JavaScript with j Query, CSS & HTML5by Robin Nixon, 2015.

Web Links

How to generate PDF file using PHP ? – GeeksforGeeks

Create PDF in PHP | Delft Stack

Generate PDFs with PHP - SitePoint

How to Create a PDF in PHP (linuxhint.com)

PHP Write Text Into an Existing PDF (fullstack-tutorials.com)

fpdf - How to insert Image in pdf using PHP (Generating a PDF)? - Stack Overflow

Insert an Image into a PDF in PHP | PDFTron SDK

How to Generate a PDF File in PHP (tutsplus.com)

Unit 10: Working with XML in PHP

CONTENTS

Objectives

After studying this unit, you will be able to:

- learn about working with XML in PHP and embedding XML in PHP.
- know about parsing XML with simple XML and parsing XML with XSLT.
- learn about generating XML in PHP.

Introduction

XML is a markup language that looks a lot like HTML. The Extensible Markup Language (XML) is a markup language and file format for storing, transmitting, and reconstructing arbitrary data. An XML document is plain text and contains tags delimited by < and >. It defines a set of rules for encoding documents in a format that is both human-readable and machine-readable.

10.1 XML Advantages

XML is a good choice for data transfer using text-based documents supporting Unicode.

The beauty of XML tags is that it permits lay readers to understand the data and its meaning. XML is written in Unicode so its symbols and special characters are included as part of the data. No more need to scour the data to find the single symbol that threw an error.

XML is a platform, network, and system independent

The biggest dilemma any company faces is adhering to internal data rules and standards, external industry standards, plus their own customers' requirements. Being text-based, XML can be read by any coding language, on any platform, or on any system. There is no longer a long drawn-out process to ensure compatibility, because data can be transferred quickly and seamlessly.

Users define their own tags in XML

These days, companies have their own data requirements and standards. The developers don't have wait for emerging new standards, with XML tags they can proceed based upon known requirements.

Searching data in XML is easy

Not only are XML tags written in an easy-to-read format, but utilizing an XML parser permits users to search for a particular tag. The days of plodding through endless code are over!.

XML provides easy communication of the hierarchy of data elements using the tree structure.

It can be saved in XML or text format, in any database: The SQL, Oracle, or Documentum. With XML, data can be saved in any database and modified for any requirements, context, or syntax.

User interface can rearranged for cosmetic reasons or better usability, without needing to change the underlying data structure

Frequently data specialists are asked to rearrange the interface of the data. Without XML, this means the underlying data structure must also be changed. This is costly in terms of time and manpower, and it increases the probability of errors. When the data structure does not have to be touched, costs are minimized.

Developers have greater flexibility as they design and develop websites or user display formats

Developers can easily revamp a website, or change the way data is displayed creating a design-driven site with changing the supporting data structures. XML continues to grow in popularity as the language of choice for transmitting data between different platforms, devices or data targets. It is a platform independent language which can simplify complex data transfers like attributes for catalog data or multiple table structures supporting your website. The web services such as SOAP and REST use XML format to exchange information. XML documents can be used to store configuration settings of an application.It allows you to create your own custom tags which make it more flexible.

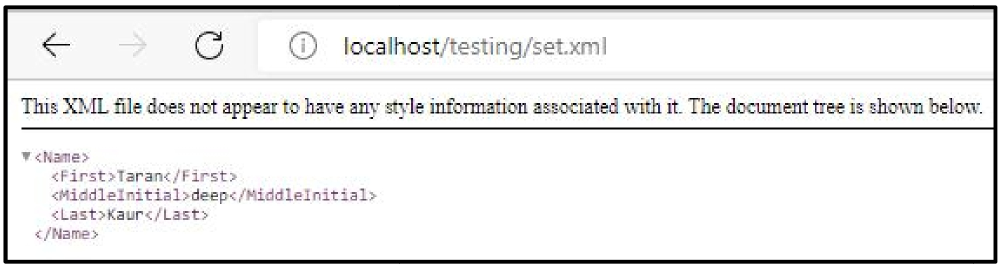
10.2 How is XML Defined?

XML is not a programming language like C++ or Java, but a markup specification language.A browser or other application must read the XML document in order to make it do something. The development of XML documents begins with the identification and definition of the data elements that will be displayed or exchanged. The data elements are defined using tags that indicate what a data element represents.

For example, a person's name might contain three tags:

```
<?xml version="1.0" ?>      //set.xml
<Name>
<First>Taran</First>
<MiddleInitial>deep</MiddleInitial>
<Last>Kaur</Last>
</Name>
```

The developers create a schema that contains the tag name, their data formats, and the relationships to one another.



This XML file does not appear to have any style information associated with it. The document tree is shown below.

```
▼<Name>
   <First>Taran</First>
   <MiddleInitial>deep</MiddleInitial>
   <Last>Kaur</Last>
 </Name>
```

10.3 XML vs HTML

There are two big differences between XML and HTML:

XML doesn't define a specific set of tags you must use.XML is extremely picky about document structure.XML gives you a lot more freedom than HTML. HTML has a certain set of tags: the <a></a> tags surround a link, the <p> starts paragraph and so on. An XML document, however, can use any tags you want. Put <rating></rating> tags around a movie rating, <height></height> tags around someone's height. Thus, XML gives you option to device your own tags.

HTML was designed to display data with focus on how data looks while XML was designed to be a software and hardware independent tool used to transport and store data, with focus on what data is.

- HTML is a markup language itself while XML provides a framework for defining markup languages.
- HTML is a presentation language while XML is neither a programming language nor a presentation language.
- HTML is case insensitive while XML is case sensitive.
- HTML is used for designing a web-page to be rendered on the client side while XML is used basically to transport data between the application and the database.
- HTML has its own predefined tags while what makes XML flexible is that custom tags can be defined and the tags are invented by the author of the XML document.

XML is very strict when it comes to document structure.HTML lets you play fast and loose with some opening and closing tags. But this is not the case with XML.HTML is not strict if the user does not use the closing tags but XML makes it mandatory for the user the close each tag that has been used.HTML does not preserve white space while XML does.HTML is about displaying data, hence static but XML is about carrying information, hence dynamic.

Notes:HTML list that's not valid XML:

```
<ul>
<li>Hello
<li>How are U
<li>World is beautiful
</ul>
```

This is not a valid XML document because there are no closing </li> tags to match up with the three opening <li> tags. Every opened tag in an XML document must be closed.

10.4 Embedding XML in PHP

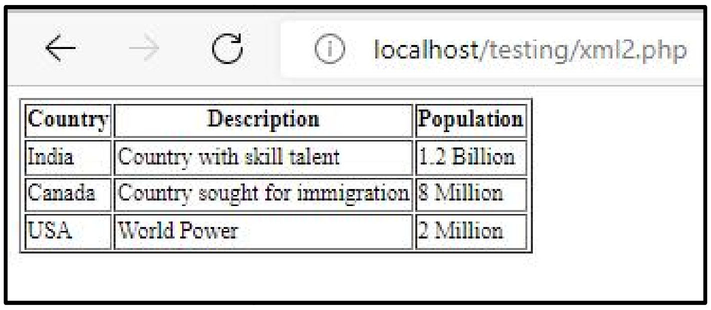
It is possible to embed XML in PHP. The following example illustrates the concept of embedding XML in PHP.

Example 1:

```
<?xml version="1.0" encoding="utf-8" ?>                    //xml2.xml
<wines>
<record>
<title>India</title>
<desc>Country with skill talent</desc>
<population>1.2 Billion</population>
</record>
<record>
<title>Canada</title>
<desc>Country sought for immigration</desc>
<population>8 Million</population>
</record>
<record>
<title>USA</title>
<desc>World Power</desc>
<population>2 Million</population>
</record>
</wines>
```

```
<?php                              //xml2.php
$wines = simplexml_load_file('xml2.xml');
echo "<table border='2'>";
echo "<tr>";
echo "<th>Country</th>";
echo "<th>Description</th>";
echo "<th>Population</th>";
echo "</tr>";
foreach ($wines->record as $records)
{
echo "<tr>";
echo "<td>$records->title</td>";
echo "<td>$records->desc</td>";
echo "<td>$records->population</td>";
echo "</tr>";
}
echo "</table>";
?>
```

Output:Figure 1shows the output of the example above.



localhost/testing/xml2.php

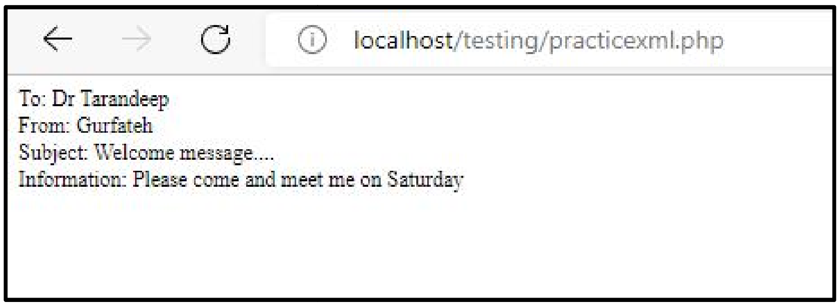
Country	Description	Population
India	Country with skill talent	1.2 Billion
Canada	Country sought for immigration	8 Million
USA	World Power	2 Million

Figure 1: Output for Example 1

Example 2:

```
<?xml version="1.0" encoding="UTF-8"?>      //practicexml.xml
<note>    //root node
<to>Dr Tarandeep</to>    //fields in root node
<from>Gurfateh</from>
<heading>Welcome message....</heading>
<body>Please come and meet me on Saturday</body>
</note>
<?php                              //practicexml.php
$xml=simplexml_load_file("practicexml.xml");          //parsing
echo "To:        ".$xml->to . "<br>";
echo "From:      ".$xml->from . "<br>";
echo "Subject:   ".$xml->heading . "<br>";
echo "Information: ".$xml->body;
?>
```

Output: Figure 2 shows the output of the example above.



localhost/testing/practicexml.php

```
To: Dr Tarandeep
From: Gurfateh
Subject: Welcome message....
Information: Please come and meet me on Saturday
```

Figure 2: Output for Example 2

Example 3:

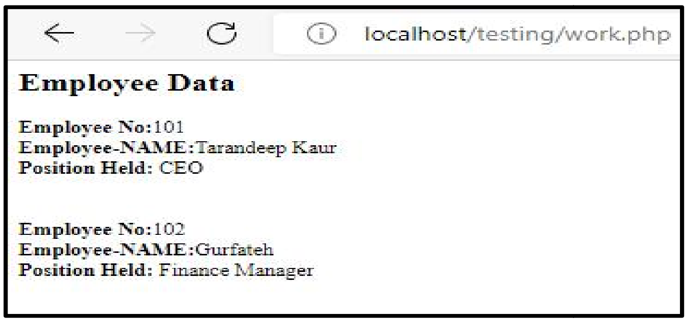
Write XML program to create employees XML file by employee id, employee name, designation, phone number and address.Write a PHP program to retrieve the employees' details from employees XML file.

Way 1:

```
<?xml version="1.0" encoding="utf-8"?>                    //work.xml
<employees>       //root node
<record emp_no = "101">   //child1
```

```
<name>Tarandeep Kaur</name>
<position>CEO</position>
</record>
<record emp_no = "102">                //child2
<name>Gurfateh</name>
<position>Finance Manager</position>
</record>
</employees>

<?php                          //work.php
$xml=simplexml_load_file('work.xml');
echo '<h2> Employee Data</h2>';
$list123 = $xml->record;
for ($i = 0; $i<count($list123); $i++)
{
echo '<b>Employee No:</b>' . $list123[$i]->attributes()->emp_no . '<br>';
echo '<b>Employee-NAME:</b>' . $list123[$i]->name . '<br>';
echo '<b>Position Held: </b>' . $list123[$i]->position . '<br><br><br>';
}
?>
```

Output: Figure 3shows the output of the example above.



localhost/testing/work.php

```
Employee Data

Employee No:101
Employee-NAME:Tarandeep Kaur
Position Held: CEO


Employee No:102
Employee-NAME:Gurfateh
Position Held: Finance Manager
```

Figure 3: Output for Example 3

Example 4:

Write XML program to create employees XML file by employee id, employee name, designation, phone number and address.Write a PHP program to retrieve the employees' details from employees XML file.

Way 2:

```
<?xml version="1.0" encoding="utf-8"?>                //practicexml2.xml
<employees>                    //root node or element
<record emp_id = "101">                          //child1
<ename>Dr Taran</ename>
<designation>Assistant Professor</designation>
        <phone>98767876</phone>
```

```
            <address>Punjab</address>
</record>
<record emp_id = "102">                              //child2
<ename>Gurfateh</ename>
<designation>Assistant Professor</designation>
            <phone>9786672</phone>
            <address>Haryana</address>
</record>
</employees>
```

In the above code:

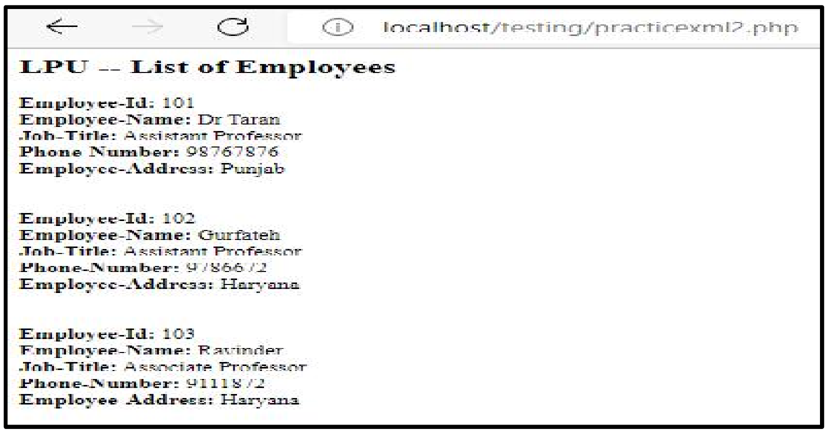
“<?xml version="1.0" encoding="utf-8"?>” specifies the xml version to be used and encoding

“<employees is the root element.

“<record…>…</record>” are the child elements of administration and sales respectively.

```
<?php                            //practicexml2.php
$xml = simplexml_load_file('practicexml2.xml');
echo '<h2> LPU -- List of Employees </h2>';
$list = $xml->record;
for ($i = 0; $i< count($list); $i++) {
echo '<b>Employee-Id:</b> ' . $list[$i]->attributes()->emp_id . '<br>';
echo '<b>Employee-Name:</b> ' . $list[$i]->ename . '<br>’;
echo '<b>Job-Title:</b> ' . $list[$i]->designation . '<br>’;
echo '<b>Phone-Number:</b> ' . $list[$i]->phone . '<br>’;
echo '<b>Employee-Address:</b> ' . $list[$i]->address . '<br><br><br>';
}
?>
```

Output: Figure 4shows the output of the example above.



```
localhost/testing/practicexml2.php

LPU -- List of Employees

Employee-Id: 101
Employee-Name: Dr Taran
Job-Title: Assistant Professor
Phone Number: 98767876
Employee-Address: Punjab


Employee-Id: 102
Employee-Name: Gurfateh
Job-Title: Assistant Professor
Phone-Number: 9786672
Employee-Address: Haryana


Employee-Id: 103
Employee-Name: Ravinder
Job-Title: Associate Professor
Phone-Number: 9111872
Employee Address: Haryana
```

Figure 4: Output for Example 4

10.5 XML Parsers

An XML parser is a program that translates the XML document into an XML Document Object Model (DOM) Object.The XML DOM Object can then be manipulated using JavaScript, Python, and PHP etc.The keyword CDATA which is the acronym for (Unparsed) Character Data.It is used to ignore special characters such as "<,>" when parsing an XML document.

Parsing XML in PHPUsing Document Object Model (DOM)

DOM is the acronym for Document Object Model.It is a cross platform and language neutral standard that defines how to access and manipulate data in

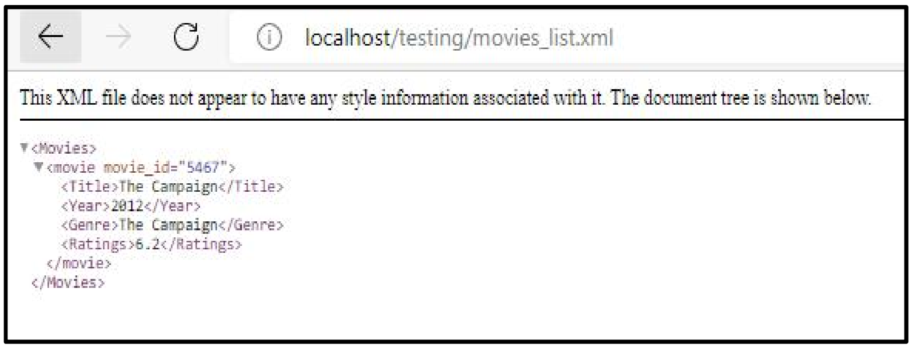
- HTML
- XHTML
- XML

DOM XML is used to access and manipulate XML documents. It views the XML document as a tree-structure.

Example:

```
<?php    //createxml.php
$dom = new DOMDocument();
            $dom->encoding = 'utf-8';
            $dom->xmlVersion = '1.0';
            $dom->formatOutput = true;
$xml_file_name = 'movies_list.xml';
            $root = $dom->createElement('Movies');
            $movie_node = $dom->createElement('movie');
            $attr_movie_id = new DOMAttr('movie_id', '5467');
            $movie_node->setAttributeNode($attr_movie_id);
$child_node_title = $dom->createElement('Title', 'The Campaign');
            $movie_node->appendChild($child_node_title);
            $child_node_year = $dom->createElement('Year', 2012);
            $movie_node->appendChild($child_node_year);
$child_node_genre = $dom->createElement('Genre', 'The Campaign');
            $movie_node->appendChild($child_node_genre);
            $child_node_ratings = $dom->createElement('Ratings', 6.2);
            $movie_node->appendChild($child_node_ratings);
            $root->appendChild($movie_node);
            $dom->appendChild($root);
$dom->save($xml_file_name);
echo "$xml_file_name has been successfully created";
?>
```

Output: Figure 5shows the output for the example above.



This XML file does not appear to have any style information associated with it. The document tree is shown below.

```
▼<Movies>
  ▼<movie movie_id="5467">
      <Title>The Campaign</Title>
      <Year>2012</Year>
      <Genre>The Campaign</Genre>
      <Ratings>6.2</Ratings>
    </movie>
  </Movies>
```

Figure 5: Output for Parsing XML in PHP Using Document Object Model (DOM)

Parsing XML With XSLT

Extensible Style sheet Language Transformations (XSLT) is a language for transforming XML documents into different XML, HTMLor any other format.Many websites offer several formats of their content—HTML, printable HTML, and WML (Wireless Markup Language) are common.The easiest way to present these multiple views of the same information is to maintain one form of the content in XML and use XSLT to produce the HTML, printable HTML, and WML. The PHP's XSLT extension uses the libxslt C library to provide XSLT support. The three documents are involved in an XSLT transformation: -

- The original XML document,
- The XSLT document containing transformation rules,
- And the resulting document.

1. To do an XSLT transformation in PHP, you create an XSLT processor, give it some input to transform, and then destroy the processor. You can create a processor by creating a new XsltProcessorobject:

   $processor = new XsltProcessor;

2. Then, parse the XML and XSL files into DOM objects:

   $xml = new DomDocument;

   $xml->load($filename);

   $xsl= new DomDocument;

   $xsl->load($filename);

3. Attach the XML rules to the object:

   $processor->importStyleSheet($xsl);

4. Process a file with the transformToDoc(), transformToUri(), or transformToXml() methods:

   $result = $processor->transformToXml($xml);

10.6 Generating XML in PHP- simplexml_load_string() Function

PHP simplexml_load_string() Function: We can convert an XML string into an object, then output the keys and elements of the object. In order to attach the XML rules to the object, use:

$processor->importStyleSheet($xsl);

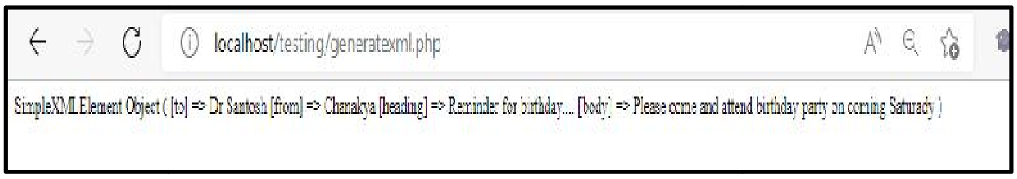
Then, process a file with the transformToDoc(), transformToUri(), or transformToXml() methods:

$result = $processor->transformToXml($xml);

We can convert an XML string into an object, then output the keys and elements of the object as illustrated below:

Example:

```
<?php                          //generatexml.php
$note=<<<XML
<note>
<to>Dr Santosh</to>
<from>Chanakya</from>
<heading>Reminder for birthday....</heading>
<body>Please come and attend birthday party on coming Saturady</body>
</note>
XML;
$xml=simplexml_load_string($note);
print_r($xml);
?>
```



localhost/testing/generatexml.php

```
SimpleXMLElement Object ( [to] => Dr Santosh [from] => Chanakya [heading] => Reminder for birthday.... [body] => Please come and attend birthday party on coming Saturady )
```

Example:

Creating an XML Document and PHP with one PHP Program:

```
<?php                //books123.php- Creating a separate XML Document
$simplexml= new SimpleXMLElement('<?xml version="1.0"?><books/>');
$book1= $simplexml->addChild('book');
$book1->addChild("Booktitle", “Punjab Rules");
$book1->addChild("PublicationDate", 2007);
$book2= $simplexml->addChild('book');
$book2->addChild("Booktitle", “India Rules");
$book2->addChild("PublicationDate", 2009);
$book3= $simplexml->addChild('book');
$book3->addChild("Booktitle", “United kingdom Rules");
$book3->addChild("PublicationDate", 2012);
file_put_contents('books.xml', $simplexml->asXML());
echo "Sucessfully created books.xml file";
?>
```

Summary

- XML is not a programming language like C++ or Java, but a markup specification language.

- XML is a markup language that looks a lot like HTML. XML is extremely picky about document structure. XML gives you a lot more freedom than HTML.
- HTML lets you play fast and loose with some opening and closing tags. But this is not the case with XML. HTML is not strict if the user does not use the closing tags but XML makes it mandatory for the user the close each tag that has been used.
- HTML does not preserve white space while XML does. HTML is about displaying data, hence static but XML is about carrying information, hence dynamic.
- HTML was designed to display data with focus on how data looks while XML was designed to be a software and hardware independent tool used to transport and store data, with focus on what data is.
- The keyword CDATA which is the acronym for (Unparsed) Character Data.It is used to ignore special characters such as "<,>" when parsing an XML document.
- The PHP's XSLT extension uses the libxslt C library to provide XSLT support.

Keywords

XML:Extensible Markup Language (XML) is a markup language and file format for storing, transmitting, and reconstructing arbitrary data.

XML Document: An XML document is plain text and contains tags delimited by < and >. It defines a set of rules for encoding documents in a format that is both human-readable and machine-readable.

XML Parser:An XML parser is a program that translates the XML document into an XML Document Object Model (DOM) Object.The XML DOM Object can then be manipulated using JavaScript, Python, and PHP etc.

Document Object Model (DOM):DOM specifies interfaces which may be used to manage XML or HTML documents.

Extensible Style sheet Language Transformations (XSLT):Extensible Style sheet Language Transformations (XSLT) is a language for transforming XML documents into different XML, HTML,or any other format.

SelfAssessment

1. XML stands for

A. Extensible Markup Language

B. Extensible Makeup Language

C. Extendable Markup Language

D. Extendable Makeup Language

2. _______________ is a markup language and file format for storing, transmitting, and reconstructing arbitrary data.

A. Orbit

B. XML

C. R programming

D. HTTP

3. Which of the following statement is/are true for an XML document?

A. An XML document is plain text.

B. Contains tags delimited by < and >.
C. Defines a set of rules for encoding documents in a format that is both human-readable and machine-readable.
D. All the above.

4. ___________ is a markup language itself while ___________ provides a framework for defining markup languages.
A. HTML, XML
B. XML, HTML
C. HTTP, XML
D. XML, HTTP

5. HTML is case insensitive while XML is case sensitive.
A. True
B. False

6. ___________ includes both a transformation language (XSLT) and a formatting language.
A. Extendable Style sheet Language
B. Extensible Style sheet Language
C. Extensible Stylus Setting Language
D. Extendible Stylus Setting Language

7. XML was designed to be a software and hardware independent tool used to transport and store data, with focus on what data is.
A. True
B. False

8. Which of the following statement is/are true for XML?
A. XML is a programming language like C++ or Java.
B. XML is a markup specification language.
C. XML is case sensitive.
D. All the above.

9. The users can define their own tags in XML.
A. True
B. False

10. XML provides easy communication of the hierarchy of data elements using the ________ structure.
A. linked list
B. queue
C. tree
D. graph

11. The term DOM stands for
A. Distant Object Model

B. Document Object Model
C. Document Orbit Model
D. Development Object Model

12. Which of the following statement is/are true for DOM?
A. It is a cross platform and language neutral standard that defines how to access and manipulate data in HTML, XHTML or XML.
B. DOM XML is used to access and manipulate XML documents.
C. It views the XML document as a tree-structure.
D. All the above.

13. _____________ is a language for transforming XML documents into different XML, HTML, or any other format.
A. XSMT
B. XMTL
C. XSLT
D. XMIT

14. ____________ function is used for converting an XML string into an object, then output the keys and elements of the object.
A. simplexml_convert_string()
B. simplexml_loadstring()
C. simplexml_output_string()
D. simplexml_set_string()

15. An XML parser is a program that translates the XML document into an XML DOM object.
A. True
B. False

Answers for Self Assessment

l.	A	2.	B	3.	D	4.	A	5.	A
6.	B	7.	A	8.	B	9.	A	10.	C
11.	B	12.	D	13.	C	14.	B	15.	A

Review Questions

1. What is XML? What are the main development goals of XML?
2. Write the features of XML.
3. How do we generate XML document in PHP? Explain with example.
4. How do we parse XML in PHP? Which parsers are used in PHP?
5. How do we create a parser? Explain with example.
6. How do we transform XML with XSLT?

7. Discuss simplexml_load_string() function with example.
8. Is it possible to embed XML in PHP? If yes, how?
9. Compare XML and HTML.
10. Write a short note on:
    a) Parsing XML in PHP using Document Object Model (DOM)
    b) Parsing XML With XSLT

Further Readings

The Joy of PHP: A Beginner's Guide by Alan Forbes.

PHP: The Complete Reference by Steven Holzner. 2017.

Learning PHP by Ramesh Bangia, 2012.

Learn PHP in One Day and Learn It Well by Jamie Chan.

XML and PHP by Vikas Vaswani, 2002

Web Links

PHP & XML (tutorialspoint.com)

PHP XML Tutorial: Create, Parse, Read with Example (guru99.com)

XML vs HTML - Head to Head Comparison (stechies.com)

HTML vs XML - GeeksforGeeks

Embedding PHP in Web Pages - Programming PHP [Book] (oreilly.com)

Embedding PHP in XML - Stack Overflow

Dr. Tarandeep Kaur, Lovely Professional University

Unit 11: File Upload and File Permissions in PHP

CONTENTS

Objectives

After studying this unit, you will be able to:

- know about the file handling and files access options in PHP.
- demonstrate the file uploading and file permissions in PHP.
- understand different methods of appending data in a file.
- know the process of deleting a file and use of unset() function.

Introduction

The file handling is an important part of any web application.You often need to open and process a file for different tasks.PHP allows you to include file so that a page content can be reused many times.It is very helpful to include files when you want to apply the same HTML or PHP code to multiple pages of a website. There are two ways to include file in PHP:

- include
- Require

Both include and require are identical to each other, except failure.

o **include** only generates a warning, i.e., E_WARNING, and continue the execution of the script.

o **require** generates a fatal error, i.e., E_COMPILE_ERROR, and stop the execution of the script.

Table 1 lists the difference between include() and require() function.

11.1 include() Function

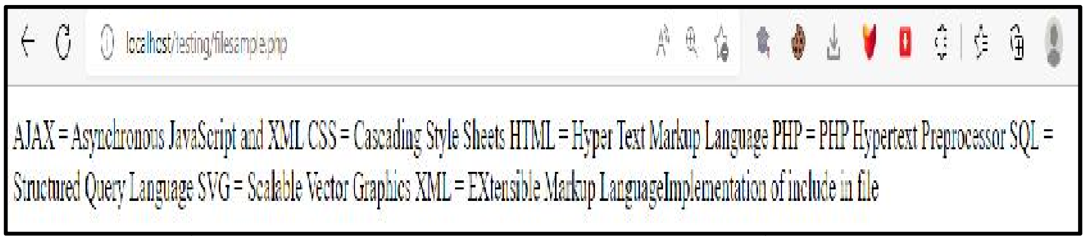
PHP include is used to include a file based on the given path. You may use a relative or absolute path of the file.There are two syntaxes available for include:

```
include 'filename';

or

include ('filename');
```

Example:

```
<?php                          //filesample.php
        include("webscript.txt");
        echo "Implementation of include in file"
?>
```

The output for the above example is:



localhost/testing/filesample.php

```
AJAX = Asynchronous JavaScript and XML CSS = Cascading Style Sheets HTML = Hyper Text Markup Language PHP = PHP Hypertext Preprocessor SQL =
Structured Query Language SVG = Scalable Vector Graphics XML = EXtensible Markup LanguageImplementation of include in file
```

Table 1: Include() v/s Require()

include()	require()
Does not stop the execution of the script even if any error occurs.	Stop the execution of the script when an error occurs.
Does not give a no fatal error, will only produce a warning (E_WARNING) and the script will continue to execute.	Gives a fatal error, (E_COMPILE_ERROR) along with the warning.
Mostly used when the file is not required, and the application should continue to execute its process when the file is not found.	Mostly used when the file is mandatory for the application.

11.2 PHP Accessing Files

PHP has several functions for creating, reading, uploading, and editing files.

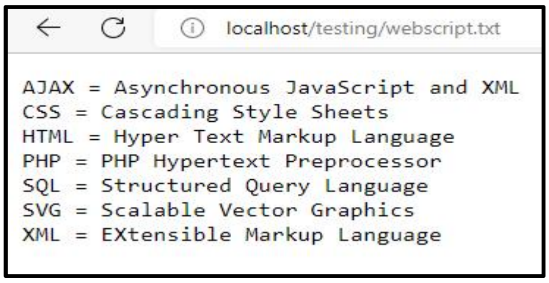
PHP readfile() Function

The readfile() function reads a file and writes it to the output buffer.It returns the number of bytes read on success. The readfile() function is useful if all you want to do is open up a file and read its contents.

Example:

```
<?php                  //webscript.txt
echo readfile("webscript.txt");
?>
```

The output for the above example is:


localhost/testing/webscript.txt

```
AJAX = Asynchronous JavaScript and XML
CSS = Cascading Style Sheets
HTML = Hyper Text Markup Language
PHP = PHP Hypertext Preprocessor
SQL = Structured Query Language
SVG = Scalable Vector Graphics
XML = EXtensible Markup Language
```

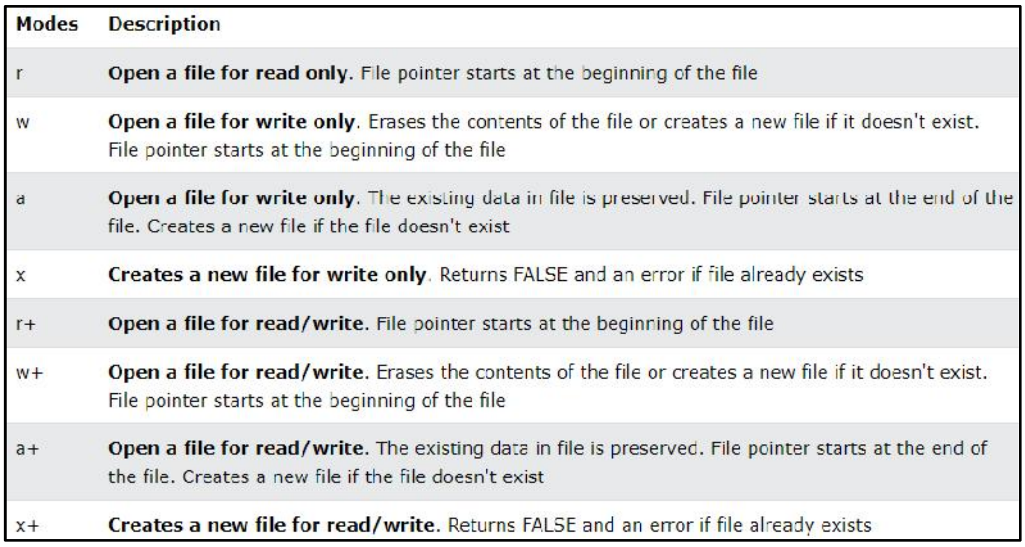
PHP File Open- fopen()

A better method to open files is with the fopen() function. This function gives you more options than the readfile() function. The syntax for the function is:

fopen("<filename>", "<mode>")

The first parameter of fopen() contains the name of the file to be opened and the second parameter specifies in which mode the file should be opened. The file may be opened in one of the following modes (Table 2):

Table 2: Different File Modes



Modes	Description
r	**Open a file for read only.** File pointer starts at the beginning of the file
w	**Open a file for write only.** Erases the contents of the file or creates a new file if it doesn't exist. File pointer starts at the beginning of the file
a	**Open a file for write only.** The existing data in file is preserved. File pointer starts at the end of the file. Creates a new file if the file doesn't exist
x	**Creates a new file for write only.** Returns FALSE and an error if file already exists
r+	**Open a file for read/write.** File pointer starts at the beginning of the file
w+	**Open a file for read/write.** Erases the contents of the file or creates a new file if it doesn't exist. File pointer starts at the beginning of the file
a+	**Open a file for read/write.** The existing data in file is preserved. File pointer starts at the end of the file. Creates a new file if the file doesn't exist
x+	**Creates a new file for read/write.** Returns FALSE and an error if file already exists

PHP File Read- fread()

The fread() function reads from an open file.The first parameter of fread() contains the name of the file to read from and the second parameter specifies the maximum number of bytes to read.

Syntax:

fread($myfile,filesize("webscript.txt"));

PHP Close File- fclose()

The fclose() function is used to close an open file. It requires the name of the file (or a variable that holds the filename) we want to close.

Syntax:

fclose(<filename> or <variable holding the file>);

PHP File Open/Read/Close

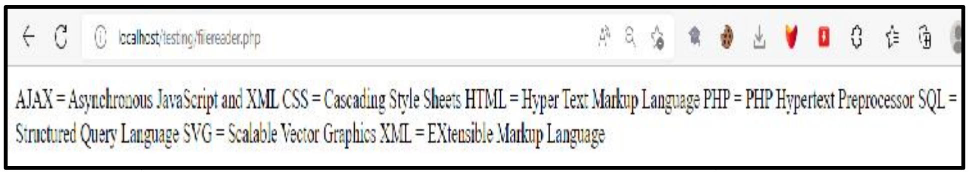
Example 1: The following example also generates a message if the fopen() function is unable to open the specified file:

```
<?php    //filereader.php
$myfile = fopen("webscript.txt", "r") or die("Unable to open file!");
echo fread($myfile,filesize("webscript.txt"));
fclose($myfile);
?>
```

Example 2:

```
<?php                //fileclose.php
$myfile = fopen("webscript.txt", "r");
// some code to be executed....
fclose($myfile);
?>
```

The output for the above example is:



localhost/testing/filereader.php

```
AJAX = Asynchronous JavaScript and XML CSS = Cascading Style Sheets HTML = Hyper Text Markup Language PHP = PHP Hypertext Preprocessor SQL =
Structured Query Language SVG = Scalable Vector Graphics XML = EXtensible Markup Language
```

PHP File Write- fwrite()

Example: Write a file ('w')

```
<?php                          //filewriter.php
$myfile = fopen("webscript.txt", "w") or die("Unable to open file!");
echo fwrite($myfile, "Hello World to PHP Learning");
// Syntax:  fwrite(file, string, length) where string to be written; length is optional
fclose($myfile);
?>
```

PHP Read Single Line- fgets()

The fgets() function is used to read a single line from a file.

Example:<?php   //fgets.php

```
$myfile = fopen("webscript.txt", "r") or die("Unable to open file!");
echo fgets($myfile);
fclose($myfile);
```

```
?>
```

PHP Check End-Of-File- feof()

The feof() function checks if the "end-of-file" (EOF) has been reached. It is useful for looping through data of unknown length.

Example: The example below reads the "webscript.txt" file line by line, until end-of-file is reached:

```
<?php                //endof.php
$myfile = fopen("webscript.txt", "r") or die("Unable to open file!");
// Output one line until end-of-file
while(!feof($myfile)) {
  echo fgets($myfile) . "<br>";
}
fclose($myfile);
?>
```

PHP Read Single Character- fgetc()

The fgetc() function is used to read a single character from a file.

Example: The example below reads the "webscript.txt" file character by character, until end-of-file is reached:

```
<?php                //readsingle.php
$myfile = fopen("webscript.txt", "r") or die("Unable to open file!");
// Output one character until end-of-file
while(!feof($myfile))
{
echo fgetc($myfile);
}
fclose($myfile);
?>
```

11.3 File Uploads in PHP

With PHP, it is easy to upload files to the server. There are certain steps involved in the process of uploading file in PHP.

1. Configure the "php.ini" File
2. First, ensure that PHP is configured to allow file uploads.
3. In your "php.ini" file, search for the file_uploads directive, and set it to On:

```
file_uploads= On
```

Create The HTML Form: Next, create an HTML form that allow users to choose the image file they want to upload.Some rules to follow for the HTML form above:

- Make sure that the form uses method="post"

- The form also needs the following attribute: enctype="multipart/form-data". It specifies which content-type to use when submitting the form
- Without the requirements above, the file upload will not work.

Example:<!DOCTYPE html>   //fupload.html

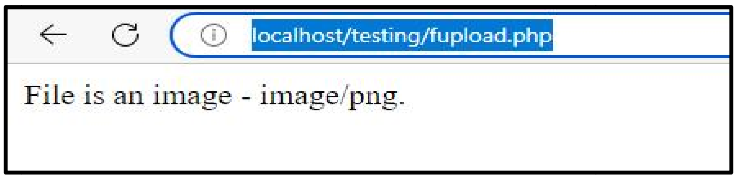
```
<html>
<body>
<form action="fupload.php" method="post" enctype="multipart/form-data">
 Select image to upload:
<input type="file" name="google" id="gp">
<input type="submit" value="Upload Image" name="submit">
</form>
</body>
</html>
```

In the above example, there are other things to notice:

- The type="file" attribute of the <input> tag shows the input field as a file-select control, with a "Browse" button next to the input control
- The form above sends data to a file called "upload.php", which we will create next.
- Create the upload file, PHP script.

Example: The "fupload.php" file contains the code for uploading a file.

```
<?php
$target_dir = "uploads/";
$target_file = $target_dir .basename($_FILES["google"]["name"]);
$uploadOk = 1;
$imageFileType = strtolower(pathinfo($target_file,PATHINFO_EXTENSION));
// Check if image file is a actual image or fake image
if(isset($_POST["submit"])) {
 $check = getimagesize($_FILES["google"]["tmp_name"]);
if($check !== false) {
   echo "File is an image - " . $check["mime"] . ".";
   $uploadOk = 1;  }
else {
   echo "File is not an image.";
   $uploadOk = 0;  }}
?>
```



```
File is an image - image/png.
```

The above PHP script description is explained below:

o $target_dir = "uploads/" - specifies the directory where the file is going to be placed

o $target_file specifies the path of the file to be uploaded

o $uploadOk=1 is not used yet (will be used later)

o $imageFileType holds the file extension of the file (in lower case)

11.4 File Permissions

- fopen()
- fread()
- fwrite() and fputs() functions are used to write data into file.
- Files append

Fopen(), fread() and fwrite() functions have already been explained above. The other functions are:

- **fputs()**: The fputs() function is an alias of the fwrite() function. The fputs() function in PHP is an inbuilt function which is used to write to an open file. The fputs() function stops at the end of the file or when it reaches the specified length passed as a parameter, whichever comes first.The file, string and the length which has to be written are sent as parameters to the fputs() function and it returns the number of bytes written on success, or FALSE on failure. The syntax is:

```
fputs(file, string)
```

Example 1:

```
<?php                //fileput.php
// Opening a file
$myfile = fopen("data.txt", "w");
// writing content to a file using fputs
echo fputs($myfile, "World is Beautiful!");
// closing the file
fclose($myfile);
?>
```

Example 2:

```
<?php                //fileput2.php
$fp = fopen("data.txt", "w"); // Opening a file
echo fputs($fp, "World is Changing", 13);
//writing content to a file with a specified string length using fputs
fclose($fp);         //closing the file

$fp=fopen("data.txt", "r"); //opening the same file to read its contents
echo fread($fp, filesize("data.txt"));
// closing the file
fclose($fp);
```

```
?>
```

- **Appending into File:**You can append data into file by using a or a+ mode in fopen() function (Table 3). The fwrite() function is used to write and append data into file.

Table 3: Difference between a and a+

a	a+
Append to a file. Writing operations append data at the end of the file. The file is created if it does not exist.	Open a file for reading and appending. All writing operations are performed at the end of the file, protecting the previous content to be overwritten. You can reposition the internal pointer to anywhere in the file for reading, but writing operations will move it back to the end of file. The file is created if it does not exist.

Example:

```
<?php        //fileapp.php
$fp = fopen('webscript.txt', 'a'); //opens file in append mode
fwrite($fp, 'We are working with Files');
fwrite($fp, 'appending data');
fclose($fp);
echo "File appended successfully";
?>
<?php              //fileapp2.php
$fp = fopen('webscript.txt', 'a+');     //opens file in append mode
fwrite($fp, 'Hello World');
echo fread($fp,filesize("webscript.txt")) or die('unable to read');
fclose($fp);
?>
```

- **Deletion of a File using Unlink() Function:**We can delete any file using unlink() function. The deleting of a file means completely erase a file from a directory so that the file is no longer exist. A function, unlink() function accepts only one argument, the file name. It is similar to UNIX C unlink() function.The function generates E_WARNING level error if file is not deleted. It returns TRUE if file is deleted successfully otherwise FALSE. The syntax is:

```
bool unlink (string $filename)
```

where $filename represents the name of the file to be deleted.

Example:

```
<?php                //fdelete.php
$res=unlink('data.txt');
if($res){
echo "Successfully Deleted";   }
else      {
echo "Sorry! no such file exsists";   }
?>
```

11.5 Unset() Function

The unset() function is an inbuilt function in PHP which is used to remove the content from the file by emptying it. It means that the function clears the content of a file rather then deleting it. The unset() function not only clears the file content but also is used to unset a variable, thereby making it empty. The syntax is:

unset($variable)

Example 1:

```
<?php           //funset.php
        $v1 = "Keep Smiling";
        // No change would be reflected outside
        function unset_value()
        {
                unset($v1);
        }
        unset_value();
        echo $v1;
?>
```

Example 2:

```
<?php           //funset2.php
        $v1= "Good Day";
        //Change would be reflected outside the function
        function unset_Example()
        {
                unset($GLOBALS['v1']);
        }
        unset_Example();
        echo $v1;
?>
```

11.6 Delete an Array Element based on Key Using Unset() Function

The unset() function is used to remove element from the array. It is used to destroy any other variable and same way use to delete any element of an array. The unset() command takes the array key as input and removed that element from the array. After removal the associated key and value does not change. The syntax is:

unset($variable)

Example:

```
<?php           //funset3.php
$ar = array('H', 'E', 'L', 'L', 'O');
```

```
echo "Before Deletion <br/>";
// Display the aray element
print_r($ar);
unset($ar[1]);
echo "<br/>After deletion <br/>";
print_r($ar);
?>
```

11.7 Unlink() v/s Unset() Function

Table 4 lists the differences between unlink and unset functions.

Table 4: Unlink() vs Unset() Function

unlink() Function	**unset() Function**
Used to delete a file within a directory completely on successful execution.	Used to make a specific file empty by removing its content.
There are two parameter filename and the other one is context.	There is only one parameter variable.
Returns True on success and false on failure.	Does not return any value.
Function for file system handling.	Function for variable management.

Summary

- The file handling is an important part of any web application. PHP allows you to include file so that a page content can be reused many times.
- It is very helpful to include files when you want to apply the same HTML or PHP code to multiple pages of a website.
- There are two ways to include file in PHP, include() and require().
- PHP has several functions for creating, reading, uploading, and editing files.
- With PHP, it is easy to upload files to the server.
- We can delete any file in PHP using unlink() function. The deleting of a file means completely erase a file from a directory so that the file is no longer exist.
- The unset() function is an inbuilt function in PHP which is used to remove the content from the file by emptying it.

Keywords

Include(): include only generates a warning, i.e., E_WARNING, and continue the execution of the script.

Require(): require generates a fatal error, i.e., E_COMPILE_ERROR, and stop the execution of the script.

PHP readfile() Function: The readfile() function reads a file and writes it to the output buffer.

PHP fread(): The fread() function reads from an open file.

PHP fopen(): A better method to open files is with the fopen() function. This function gives you more options than the readfile() function.

PHP fclose(): The fclose() function is used to close an open file. It requires the name of the file (or a variable that holds the filename) we want to close.

PHP feof(): The feof() function checks if the "end-of-file" (EOF) has been reached. It is useful for looping through data of unknown length.

Fputs(): The fputs() function in PHP is an inbuilt function which is used to write to an open file. The fputs() function stops at the end of the file or when it reaches the specified length passed as a parameter, whichever comes first.

Unset() Function: The unset() function is an inbuilt function in PHP which is used to remove the content from the file by emptying it.

Self Assessment

1. Which of the following is/are ways to include file in PHP?

A. include

B. requirement

C. inclusion

D. None of the above

2. __________ generates a fatal error, i.e., E_COMPILE_ERROR, and stop the execution of the script.

A. Include

B. Require

C. feof

D. fputs

3. The use of ________ does not stop the execution of the script even if any error occurs.

A. include

B. require

C. feof

D. fputs

4. ______ function reads a file and writes it to the output buffer.

A. fopen()

B. fclose()

C. readfile()

D. fread()

5. Which of the following statement is/are true for fopen()?

A. A better method to open files.

B. This function gives you more options than the readfile() function.

C. The first parameter of fopen() contains the name of the file to be opened and the second parameter specifies in which mode the file should be opened.

D. All the above.

6. _____________ function is used to close an open file.

A. fclose()

B. fopen()

C. fclosure()

D. fopening()

7. The function used to read a single character from a file is _________.

A. fgetf()

B. fgetr()

C. fgetc()

D. sgetc()

8. Which of the following statement is TRUE for require?

A. Stop the execution of the script when an error occurs.

B. Gives a fatal error, (E_COMPILE_ERROR) along with the warning.

C. Mostly used when the file is mandatory for the application.

D. All the above.

9. The ___________ function is used to read a single line from a file.

A. fgets()

B. fgetc()

C. fgetr()

D. fgetf()

10. The _________ function is an inbuilt function in PHP which is used to remove the content from the file by emptying it.

A. set()

B. link()

C. unset()

D. unlink()

11. ____________ function is used to write content of the string into file.

A. fwriter()

B. fwrite()

C. fcontent()

D. fstring()

12. When a file is opened in write mode, all the existing data in the file is erased and new data can be written to the file using the fwrite() function.

A. True

B. False

13. The unset() command takes the ____________ as input and removes that element from the array.

A. array key

B. array pointer

C. array element
D. array bound

14. Which of the following statement is/are TRUE for fputs() function?
A. It is an alias of the fwrite() function.
B. An inbuilt function which is used to write to an open file.
C. Returns the number of bytes written on success.
D. All the above.

15. Which of the following statement(s) aptly describes unlink() function?
A. Accepts only one argument, the file name.
B. Similar to UNIX C unlink() function.
C. Can be used to delete any file, completely erasing it from a directory so that the file no longer exists.
D. All the above.

Answers for Self Assessment

1.	A	2.	B	3.	A	4.	C	5.	D
6.	B	7.	C	8.	D	9.	A	10.	B
11.	B	12.	A	13.	A	14.	D	15.	D

Review Questions

1. Indicate the use of unlink() function?
2. Compare unlink() and unset() function?
3. How the files are uploaded in PHP?
4. Explain the functions for different file permissions?
5. How is file handling carried out in PHP?
6. How the file appending can be carried out?
7. Differentiate a and a+ append modes?
8. What are the different file modes in PHP?
9. Compare the following:
   (a) Include() vs Require()
   (b) Unlink() vs unset()
10. Explain with examples:
    (a) fread()
    (b) fopen()
    (c) fwrite()

Further Readings

The Joy of PHP: A Beginner's Guide by Alan Forbes.

PHP: The Complete Reference by Steven Holzner. 2017.

Learning PHP by Ramesh Bangia, 2012.

Learn PHP in One Day and Learn It Well by Jamie Chan.

XML and PHP by Vikas Vaswani, 2002.

Web Links

PHP File Handling | Most Commonly Used PHP File Handling (educba.com)

PHP File() Handling & Functions (guru99.com)

PHP | Basics of File Handling - GeeksforGeeks

PHP File - javatpoint

File Handling in PHP with Examples [2 Steps] (fosstechnix.com)

PHP File Handling (w3schools.com)

Unit 12: PHP Forms

CONTENTS

Objectives

After studying this unit, you will be able to:

- Understand working with forms in PHP.
- Learn to create HTML form in PHP.
- Differentiate between GET and POST methods.
- Learn about validation in forms.

Introduction

A form is a document containing black fields that the user can fill the data or user can select the data.Casually, the data will store in the data base. Using the forms, the data can be collected from the user and can be sent directly to the server through PHP scripting.There are two methods for sending data to the server,

- GET
- POST

The PHP super global $_GET and $_POST are used to collect form-data.

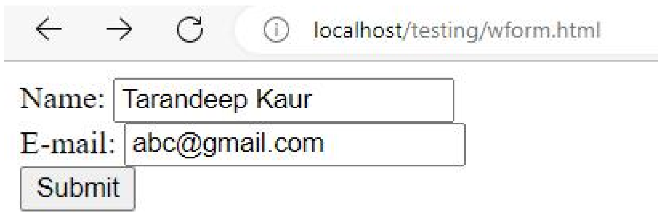
Example:

A simple HTML form looks like:

```
<html>                    //wform.html
<body>
<form action="wform.php" method="post">
Name: <input type="text" name="name"><br>
E-mail: <input type="text" name="email"><br>
<input type="submit">
</form>
```

```
</body>
</html>
```

When the user fills out the form above and clicks the submit button, the form data is sent for processing to a PHP file named “wform.php" (Figure 1 and Figure 2). The form data is sent with the HTTP POST method.

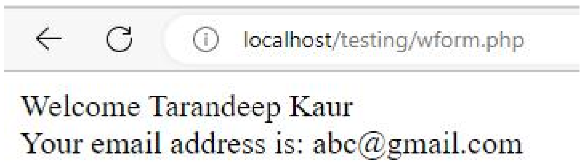

localhost/testing/wform.html

Name: Tarandeep Kaur
E-mail: abc@gmail.com

Figure 1: Output for wform.html

Example:

```
<html>              //wform.php
<body>
Welcome <?php echo $_POST["name"]; ?><br>
Your email address is: <?php echo $_POST["email"]; ?>
</body>
</html>
```


localhost/testing/wform.php

Welcome Tarandeep Kaur
Your email address is: abc@gmail.com

Figure 2: Output for wform.php

The same result could also be achieved using the HTTP GET method:

Example:

```
<html>              //wform_get.html
<body>
<form action="wform_get.php" method="get">
Name: <input type="text" name="name"><br>
E-mail: <input type="text" name="email"><br>
<input type="submit">
</form>
</body>
</html>
```

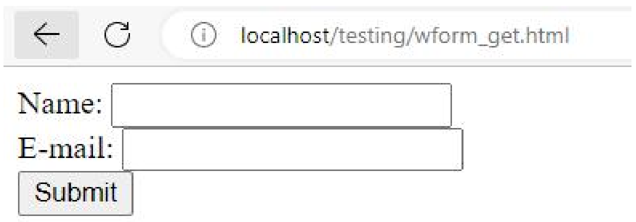
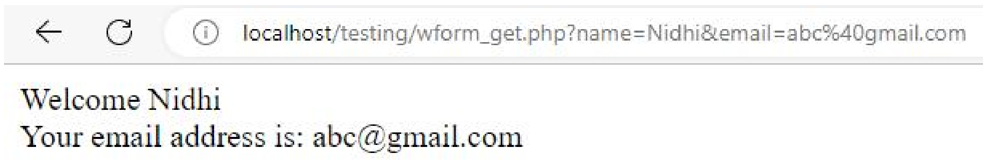


Figure 3: Output for wform_get.html

Example:

```
<html>              //wform_get.php
<body>
Welcome <?php echo $_GET["name"]; ?><br>
Your email address is: <?php echo $_GET["email"]; ?>
</body>
</html>
```



localhost/testing/wform_get.php?name=Nidhi&email=abc%40gmail.com

```
Welcome Nidhi
Your email address is: abc@gmail.com
```

Figure 4: Output for wform_get.php

12.1 Get v/s Post Method

Table 1lists the difference between the GET and POST method.

GET	POST
Harmless	Data will be re-submitted (the browser should alert the user that the data are about to be re-submitted)
Can be bookmarked	Cannot be bookmarked
Can be cached	Not cached
urlencoded	Use multipart encoding for binary data
Parameters remain in browser history	Parameters are not saved in browser history
Only ASCII characters allowed	No restrictions. Binary data is also allowed
GET is less secure compared to POST because data sent is part of the URL Never use GET when sending passwords or other sensitive information.	POST is a little safer than GET because the parameters are not stored in browser history or in web server logs

Data is visible to everyone in the URL	Data is not displayed in the URL
GET method adds the data to the URL	No restrictions

Table 1: GET vs POST Method

When to use GET?

- Information sent from a form with the GET method is visible to everyone (all variable names and values are displayed in the URL).
- GET also has limits on the amount of information to send. The limitation is about 2000 characters. However, because the variables are displayed in the URL, it is possible to bookmark the page. This can be useful in some cases.
- GET may be used for sending non-sensitive data.

Notes:GET should NEVER be used for sending passwords or other sensitive information!

When to use POST?

- Information sent from a form with the POST method is invisible to others (all names/values are embedded within the body of the HTTP request) and has no limits on the amount of information to send.
- Moreover, POST supports advanced functionality such as support for multi-part binary input while uploading files to server.
- However, because the variables are not displayed in the URL, it is not possible to bookmark the page.

Notes: Developers prefer POST for sending form data.

12.2 <u>Working with Forms in PHP and Database</u>

1. How to Insert Form Data into Database using PHP: We are going to store data in database which is submitted through HTML form.

- First, we create an HTML form that need to take user input from keyboard.
- HTML form is a document which stores information of a user on a web server using interactive controls. An HTML form contains different kind of information such as username, password, contact number, email id etc.The elements that are used in an HTML form are check box, input box, and radio buttons, submit buttons etc. With the help of these elements, the information of a user is submitted on the web server. The form tag is used to create an HTML form.
- Syntax: <form> Form Elements... </form>
- To pass the values to next page, we use the page name with the following syntax. We can use either GET or POST method to send data to the server.

```
<form action=other_page.php method= POST/GET>
Form Elements...
</form>
```

2. Database Connection and Command Execution

- The collection of related data is called a database.
- XAMPP stands for cross-platform, Apache, MySQL, PHP, and Perl. It is among the simple light-weight local servers for website development.
- In PHP, we can connect to database using localhost XAMPP web server.

3. How to Get the Form Data

- We are going to collect the form data submitted through HTML form.
- PHP $_POST method is a PHP super global variable which is used to collect data after submitting the HTML form.

Example:

```
//formsample.html
<form name=f1 action="form.php" method="POST">
Name<input type="text" name="Name"><br>
Regno<input type="text" name="Roll"><br>
<input type="Submit" value="Add Values">
</form>
```

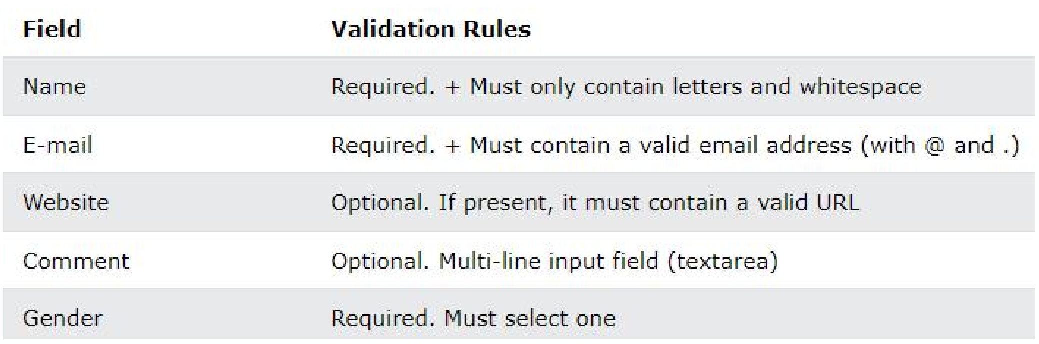
Example:

```
<!DOCTYPE html>                    //form.php
<html><head>
<title>Insert Page page</title>
</head><body><center>
<?php
// Taking all 2 values from the form data(input)
$names = $_POST["Name"];
$rolls = $_POST["Roll"];
$conn = mysqli_connect("localhost", "root", "", "dnew");
// Check connection
if(!$conn)
        die("cannot connect."<br>");
        echo "connected <br>";         //echo "<br>";
//Performing insert query execution
$sql = "INSERT INTO student1(Name,Roll) VALUE('$names','$rolls')";
if(mysqli_query($conn, $sql)) {
echo "<h3>data stored in a database successfully.";
        echo nl2br("\n$names\n $rolls\n ");      }
else {      echo "ERROR: Hush! Sorry $sql. ". mysqli_error($conn);        }
mysqli_close($conn); // Close connection
?>
</center></body></html>
```

12.3 Form Validation

We can work with validation of various fields in a form. For example, HTML form contains various input fields: required and optional text fields, radio buttons, and a submit button.The validation rules for the form are as follows (Table 2):

Table 2: Form Validation Rules



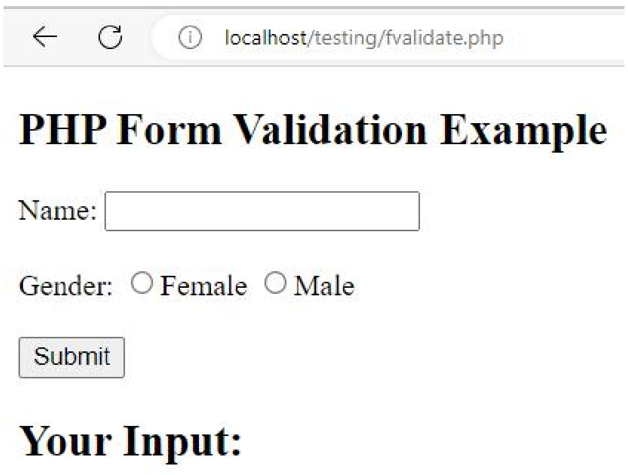
Field	Validation Rules
Name	Required. + Must only contain letters and whitespace
E-mail	Required. + Must contain a valid email address (with @ and .)
Website	Optional. If present, it must contain a valid URL
Comment	Optional. Multi-line input field (textarea)
Gender	Required. Must select one

Figure 5shows the output for the fvalidate.php program.

Example:

```
<!DOCTYPE HTML>                    //fvalidate.php- Single code on 3 slides
<html><head></head><body>
<?php    // define variables and set to empty values
$name = $gender = "";
if ($_SERVER["REQUEST_METHOD"] == "POST") {
  $name = test_input($_POST["name"]);
$gender = test_input($_POST["gender"]);       }
function test_input($data) {
  $data = trim($data);
  $data = stripslashes($data);
  $data = htmlspecialchars($data);
  return $data;    }
?>


<h2>PHP Form Validation Example</h2>
<form method="post" action="<?php echo htmlspecialchars($_SERVER["PHP_SELF"]);?>">
Name: <input type="text" name="name">     //Text Fields
<br><br>
Gender:                        //Radio Buttons
<input type="radio" name="gender" value="female">Female
<input type="radio" name="gender" value="male">Male
<br><br>
<input type="submit" name="submit" value="Submit">
</form>
//The $_SERVER["PHP_SELF"] is a super global variable that returns the //filename of
```

```
the currently executing script.

//The htmlspecialchars() function converts special characters to HTML //entities. This means that it will replace HTML characters like < and > with //&lt; and &gt;. This prevents attackers from exploiting the code by injecting //HTML or Javascript code (Cross-site Scripting attacks) in forms.

<?php
echo "<h2>Your Input:</h2>";
echo $name;
echo "<br>";
echo $gender;
?>
</body>
</html>
```


localhost/testing/fvalidate.php

PHP Form Validation Example

Name:

Gender: ○ Female ○ Male

Submit

Your Input:

Figure 5: Output for fvalidate.php

Notice that at the start of the script, we check whether the form has been submitted using $_SERVER["REQUEST_METHOD"]. If the REQUEST_METHOD is POST, then the form has been submitted- and it should be validated. If it has not been submitted, skip the validation, and display a blank form.

12.4 Form-Database Validation Techniques

The required field will check whether the field is filled or not in the proper way.

What is Validation?

Validation means check the input submitted by the user. There are two types of validation are available in PHP. They are as follows:

- Client-Side Validation: Validation is performed on the client machine web browsers.
- Server-SideValidation: After submitted by data, the data has sent to a server and perform validation checks in server machine.

Example: Program to check whether a Valid or Invalid Username and Password.

- connect.php

- index.php
- user.php

Example:

```
<?php                  //connect.php
// Create connection
$con=mysqli_connect("localhost","root","","dnew");
// Check connection
if (mysqli_connect_errno())
  {
  echo "Failed to connect to MySQL: " .mysqli_connect_error();
  }
?>
```

Example:

```
<html><head>   //index.php
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
<title>Login Form</title></head><body>
<form method="post" action="user.php" name="contactForm">
<table border="1" >
<tr>
<td><label for="username">User Name</label></td>
<td><input type="text" name="username" id="username"></td></tr>
<tr>
<td><label for="userpassword">Password</label></td>
<td><input type="password" name="userpassword" id="userpassword"></td></tr>
<tr>
<td><input type="submit" value="Submit"/>
<td><input type="reset" value="Reset"/></tr>
</table>
</form></body></html>
```

```
<?php                  //user.php
//Grab User submitted information
$username= $_POST["username"];
$userpassword1 = $_POST["userpassword"];
include('connect.php');                    //Connect to the database
//Fetching user record from database
$result=mysqli_query($con,"select username, userpassword from users where
username='$username'");
```

```
//checking if record selected
if($result === FALSE) {
die('no record fetched'.mysql_error());
}
$row =mysqli_fetch_array($result);
if($userpassword1===$row['userpassword']) {
echo "valid user";        }
else {
echo "invalid user";        }
mysqli_close($con);
?>
```

Summary

- A form is a document containing black fields that the user can fill the data or user can select the data.
- Usingthe forms, the data can be collected from the user and can be sent directly to the server through PHP scripting.
- The <form> element has two important attributes:action and method.
- The action specifies the URL that processes the form submission while the method specifies the HTTP method for submitting the form. The most used form methods are POST and GET.
- The form method is case-insensitive. It means that you can use either post or POST. If you don't specify the method attribute, the form element will use the get method by default.
- Unlike the POST method, the GET method appends the form data in the URL that processes the form.
- The informationsent from a form with the POST method is invisible to others (all names/values are embedded within the body of the HTTP request) and has no limits on the amount of information to send.

Keywords

Form: A form is a document containing black fields, that the user can fill the data or user can select the data.

HTTP POST method:If a form uses the POST method, the web browser will include the form data in the HTTP request's body. After submitting the form, you can access the form data via the associative array $_POST in PHP.

HTTP GET method:The get request is the default form request. The data passed through get request is visible on the URL browser, so it is not secured. You can send limited amount of data through get request.

Validation:Validation means check the input submitted by the user. There are two types of validation are available in PHP. They areClient-side and Server-side validation.

Client-side Validation:Validation performed on the client machine web browsers is client-side validation.

Server-side Validation:After submitted by data, the data has sent to a server and perform validation checks in server machine.

Self Assessment

1. __________ is a PHP super global that can be used to collect the form data.

A. $_GETCH

B. $_POST

C. $_GETTING

D. $_GETFL

2. Which of the following statement is TRUE for POST method?

A. Data will be re-submitted (the browser should alert the user that the data are about to be re-submitted).

B. Uses multipart encoding for binary data.

C. Parameters are not saved in browser history.

D. All the above.

3. Which of the following statement reflects, POST is better method than GET?

A. GET is more secure compared to POST.

B. Data is sent as a part of the URL in GET.

C. POST can be used for sending passwords or other sensitive information.

D. None of the above.

4. In ___________ method, data is not displayed in the URL.

A. POST

B. GET

C. GETCH

D. POSTCH

5. A ____________ is a document containing black fields, that the user can fill the data or user can select the data.

A. Form

B. Format

C. Formal

D. Session

6. ________ method can be used for sending data to the server.

A. POSTF

B. GET

C. GETF

D. POSTCH

7. In ____________ method, the parameters remain in the browser history.

A. GETF

B. POSTF

C. POSTCH

D. GET

8. The __________ method involves using the ASCII characters.

A. GETCH

B. GET

C. POST

D. POSTF

9. Which of the following statement is FALSE for GET method?

A. GET has no limitation. It allows using binary data.

B. Variables are displayed in the URL; it is possible to bookmark the page.

C. GET may be used for sending non-sensitive data.

D. All the above.

10. _____________ is a super global variable that returns the filename of the currently executing script.

A. $_SERVER["PHP_SELFLESS"]

B. $_SERVER["PHP_SELF"]

C. $_SERVERFUL["PHP_SELF"]

D. $_SERVERLESS["PHP_SELF"]

11. Which of the following statement is/are TRUE for htmlspecial chars ()?

A. Converts special characters to HTML entities.

B. Replaces HTML characters like < and > with &lt; and &gt;.

C. Prevents attackers from exploiting the code by injecting HTML or Javascript code (Cross-site Scripting attacks) in forms.

D. All the above.

12. For form validation, you need to check whether the form has been submitted using $_SERVER["REQUEST_METHOD"].

A. True

B. False

13. Which of the following statement is/are TRUE for PHP form validation?

A. It is the backend validation method.

B. It is also called as server-side validation.

C. It prevents from entering invalid data into the input field.

D. All the above.

14. Which of the following correspond to the rules for creating form validation?

A. All the Input Fields must be required.

B. Email Address must be valid format containing @ symbol.

C. The password must contain a combination of one uppercase & lowercase letter, number, special characters & minimum characters length 8. Even It will not accept any white spaces.

D. All the above.

15. Using the forms, the data can be collected from the user and can be sent directly to the server through PHP scripting. The <form> element has two important attributes: action and method.

A. True

B. False

Answers for Self Assessment

1.	B	2.	D	3.	C	4.	A	5.	A
6.	B	7.	D	8.	B	9.	A	10.	B
11.	D	12.	A	13.	D	14.	D	15.	A

Review Questions

1. Compare GET and POST method?
2. What situations involve using GET method in Forms?
3. Elaborate GET vs POST method.
4. In which situations, POST method in Forms can be used?
5. What steps are involved in inserting form data into a database?
6. What is form validation? Discuss with example.
7. Discuss with example getting form data in PHP.
8. Explain different form validation rules.
9. What arethe various form-database validation techniques? Discuss with example.
10. Write a short note on:
    a. Form validation
    b. GET method
    c. POST method

Further Readings

- The Joy of PHP: A Beginner's Guide by Alan Forbes.
- PHP: The Complete Reference by Steven Holzner. 2017.
- Learning PHP by Ramesh Bangia, 2012.
- Learn PHP in One Day and Learn It Well by Jamie Chan.
- XML and PHP by Vikas Vaswani, 2002.

Web Links

- PHP - Form Introduction (tutorialspoint.com)
- PHP: Dealing with Forms - Manual
- PHP Form Handling (w3schools.com)
- PHP Form | Learn Two Main Important PHP Form Methods (educba.com)
- PHP Complete Form Example (w3schools.com)

Unit 13: PHP State Management

CONTENTS

Objectives

After studying this unit, you will be able to:

- Explorethe concept of the PHP state management.
- Understandsession creation, session accessing, and session destroy in PHP.
- Learnabout working with cookies in PHP.

Introduction

A session is a way to store information (in variables) to be used across multiple pages.When you work with an application, you open it, do some changes, and then you close it. This is much like a session.The computer knows who you are. It knows when you start the application and when you end. But on the internet, there is one problem: the web server does not know who you are or what you do, because the HTTP address doesn't maintain state. The session variables solve this problem by storing user information to be used across multiple pages (e.g. username, favorite color, etc). By default, the session variables last until the user closes the browser.So, the session variables hold information about one single user, and are available to all pages in one application.

Session Function

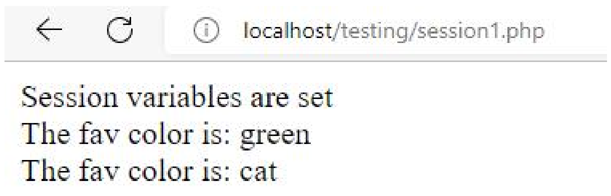
PHP session is used to store data on a server rather than the computer of the user. The Session Identifiers (SID) is a unique number which is used to identify every user in a session-based environment. SID is used to link the user with his information on the server like posts, emails etc. A session is started with the session_start() function. The session variables are set with the PHP global variable: $_SESSION.It creates a session or resumes the current one based on a session identifier passed via a GET or POST request or passed via a cookie.

Session Function- session_start(): The session_start() creates a session or resumes the current one based on a session identifier passed via a GET or POST request, or passed via a cookie (Figure 1).

Example:

```
<?php                    //session1.php
session_start();
```

```
?>
<!DOCTYPE html>
<html><body>
<?php
$_SESSION["favcolor"]="green";
$_SESSION["favanimal"]="cat";
echo "Session variables are set". "</br>";
echo "The fav color is: " . $_SESSION["favcolor"] . "</br>";
echo "The fav color is: " . $_SESSION["favanimal"];
?>
</body></html>
```



```
Session variables are set
The fav color is: green
The fav color is: cat
```

Figure 1: Output for session1.php

13.1 Steps Involved in PHP sessions

- Starting a PHP Session
- Storing Session Data
- Accessing Session Data
- Destroying Session data:
  - Destroying certain session data
  - Destroying complete session

Starting a PHP Session

The firststep is to start up a session. After a session is started, session variables can be created to store information. The PHP session_start() function is used to begin a new session. It also creates a new session ID for the user.

Example:

```
<?php
session_start();
?>
```

Storing Session Data

The session data in key-value pairs using the $_SESSION[] superglobalarray.The stored data can be accessed during lifetime of a session.

Example:

Below is the PHP code to store a session with two session variables Rollnumber and Name.

```
<?php                //session2.php
```

```
session_start();
  $_SESSION["Fname"] = "Tarandeep";
$_SESSION["Lname"] = "Kaur";
echo $_SESSION["Lname"];
  ?>
```

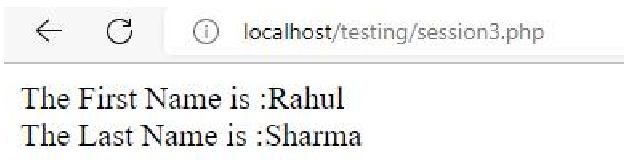
Accessing Session Data

The data stored in sessions can be easily accessed by firstly calling session_start() and then by passing the corresponding key to the $_SESSION associative array (Figure 2).

Example:

The PHP code to access a session data with two session variables Fname and Lname is shown below:

```
<?php                //session3.php
session_start();
echo 'The First Name is :' . $_SESSION["Fname"] . '<br>';
echo 'The Last Name is  :' . $_SESSION["Lname"] . '<br>';
?>
```



```
localhost/testing/session3.php

The First Name is :Rahul
The Last Name is :Sharma
```

Figure 2: Output for session3.php

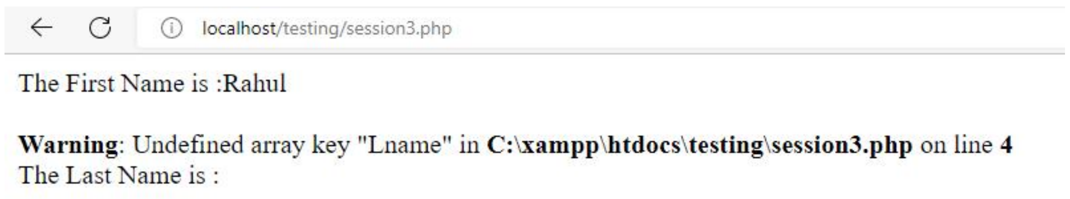
Destroying Session Data- unset()

Todelete only a certain session data, the unset feature can be used with the corresponding session variable in the $_SESSION associative array (Figure 3).

Example:

PHP code to unset only the “Lname ” session variable from the associative session array. Session is unset. Now, runsession3.php again and the error occurs.

```
<?php
session_start();
  if(isset($_SESSION["Fname"]))
{
   unset($_SESSION["Lname"]);
}
?>
```



```
localhost/testing/session3.php

The First Name is :Rahul

Warning: Undefined array key "Lname" in C:\xampp\htdocs\testing\session3.php on line 4
The Last Name is :
```

Figure 3: Output for session3.php

Destroying Session- session_destroy()

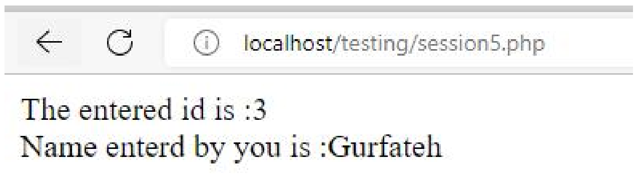
The session_destroy() function is used to completely destroy a session (Figure 4, Figure 5 and Figure 6). The session_destroy() function does not require any argument.

Example:

```
<?php
session_start();
session_destroy();
 ?>
```

Example of Session Creation, Destroying

```
<?php                //session5.php
session_start();
$_SESSION["id"] = “3";
$_SESSION["Name"] = "Gurfateh";
echo 'The entered id is :' . $_SESSION["id"] . '<br>';
echo 'Name enterd by you is :' . $_SESSION["Name"] . '<br>';
?>
```

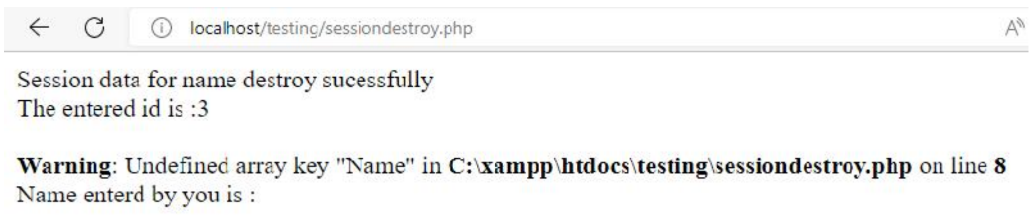


```
localhost/testing/session5.php

The entered id is :3
Name enterd by you is :Gurfateh
```

Figure 4: Output for session5.php

Example:

```
<?php                //sessiondestroy.php
session_start();
if(isset($_SESSION["id"])){
unset($_SESSION["Name"]);
}
echo "Session data for name destroy sucessfully<br>";
echo 'The entered id is :' . $_SESSION["id"] . '<br>';
echo 'Name enterd by you is :' . $_SESSION["Name"] . '<br>';
?>
```

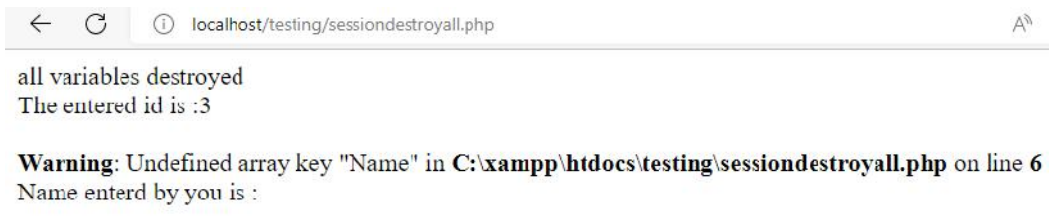


```
localhost/testing/sessiondestroy.php

Session data for name destroy sucessfully
The entered id is :3

Warning: Undefined array key "Name" in C:\xampp\htdocs\testing\sessiondestroy.php on line 8
Name enterd by you is :
```

Figure 5: Output for sessiondestroy.php

Example:

```
<?php                //sessiondestroyall.php
session_start();
session_destroy();
echo "all variables destroyed<br>";
echo 'The entered id is :' . $_SESSION["id"] . '<br>';
echo 'Name enterd by you is :' . $_SESSION["Name"] . '<br>';
?>
```



```
localhost/testing/sessiondestroyall.php

all variables destroyed
The entered id is :3

Warning: Undefined array key "Name" in C:\xampp\htdocs\testing\sessiondestroyall.php on line 6
Name enterd by you is :
```

Figure 6: Output for sessiondestroyall.php

13.2 session_unset() vs session_destroy()

Table 1lists the difference between session_unset() and session_destroy().

Table 1: session_unset() vs session_destroy()

session_destroy()	session_unset()
It destroys all the data associated with the current session. It does not unset any of the global variables associated with the session or unset the session cookie.	It deletes only the variables from session and session still exists. Only data is truncated.
bool session_destroy(void)	bool session_unset(void)

13.3 Cookie in PHP

PHP cookie is a small piece of information which is stored at client browser.It is used to recognize the user.

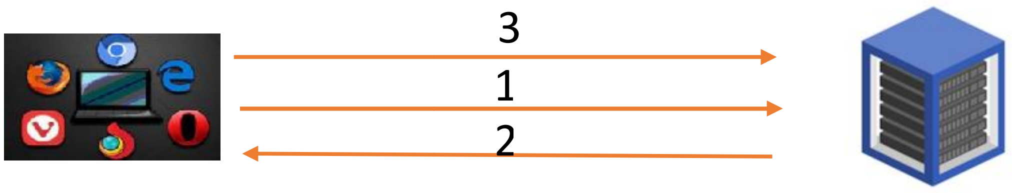


Figure 7: Cookies in PHP

*Figure 7*depicts the cookies in PHP where 1 is the request send through client browser and in step 2 server is sending the response along with the cookie. In the 3rd steps the client again the send the request along with the cookie.

The cookie is created at server side and saved to client browser.Each time when client sends request to the server, cookie is embedded with request. Such way, cookie can be received at the server side.The cookie in PHP is a small file with a maximum size of 4KB that the web server stores on the client computer. They are typically used to keep track of information such as a username that the site can retrieve to personalize the page when the user visits the website next time. A cookie can only be read from the domain that it has been issued from. The cookies are usually set in an HTTP header but JavaScript can also set a cookie directly on a browser.

Setting Cookies- setcookie() Function

To set a cookie in PHP, the setcookie() function is used. The setcookie() function needs to be called prior to any output generated by the script otherwise the cookie will not be set (Figure 8).

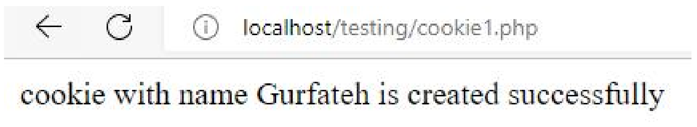
Syntax: setcookie (name, value, expire, path, domain, security);

Parameters used in setcookie(): setcookie() function requires six arguments in general which are:

- Name: It is used to set the name of the cookie.
- Value: It is used to set the value of the cookie.
- Expire: It is used to set the expiry timestamp of the cookie after which the cookie can't be accessed.
- Path: It is used to specify the path on the server for which the cookie will be available.
- Domain: It is used to specify the domain for which the cookie is available.
  - Security: It is used to indicate that the cookie should be sent only if a secure HTTPS connection exists.

Example:

```
<?php                //cookie1.php
$cname = "Gurfateh";
$cvalue = "14";
setcookie($cname, $cvalue, time() + 3 * 24 * 60 * 60);
echo "cookie with name " .$cname ." is created successfully";
?>
```



```
cookie with name Gurfateh is created successfully
```

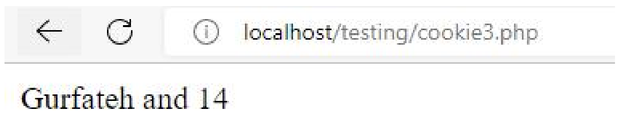
Figure 8: Output for cookie1.php

Setting Cookies- isset() Function

This function is used to check whether a cookie is set or not. It is always advisable to check whether a cookie is set or not before accessing its value. Therefore, to check whether a cookie is set or not, the PHP isset() function is used (Figure 9 and Figure 10).

Example:

```
<?php                          //cookie3.php
$cname="Gurfateh";
$croll="14";
setcookie("cname", "croll", time() + 3* 24 * 60 * 60);
echo $cname .  "   and  "  . $croll;?>
```



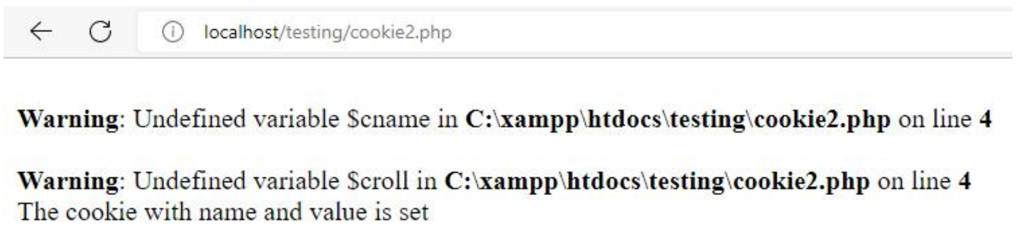
localhost/testing/cookie3.php

```
Gurfateh and 14
```

Figure 9: Output for cookie3.php

Example:

```
<?php                              //cookie2.php
if (isset($_COOKIE["cname"]))
{
echo "The cookie with name  " . $cname. " and value ".$croll. " is set<br>";
}
else
{
echo "No cookie is set reload the page to check the cookie details<br>";
}
?>
```



localhost/testing/cookie2.php

```
Warning: Undefined variable $cname in C:\xampp\htdocs\testing\cookie2.php on line 4

Warning: Undefined variable $croll in C:\xampp\htdocs\testing\cookie2.php on line 4
The cookie with name and value is set
```

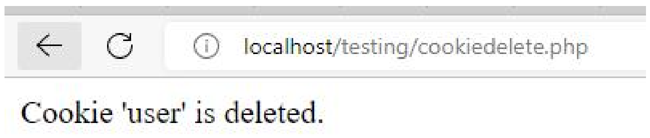
Figure 10: Output for cookie2.php

Deleting a Cookie

You can delete a cookie by calling setcookie() function with the cookie name and any value (Figure 11).

Example:

```
<!DOCTYPE html>                      //cookiedelete.php
<?php
// set the expiration date to one hour ago
setcookie("user", "", time() - 3600);
?>
<html>
<body>
<?php
echo "Cookie 'user' is deleted.";
?>
</body>
</html>
```



```
Cookie 'user' is deleted.
```

Figure 11: Output for cookiedelete.php

Summary

- A session is a way to store information (in variables) to be used across multiple pages.When you work with an application, you open it, do some changes, and then you close it. This is much like a session.
- PHP session is used to store data on a server rather than the computer of the user.A session is started with the session_start() function.
- Session Identifiers (SID) is a unique number which is used to identify every user in a session-based environment. SID is used to link the user with his information on the server like posts, emails etc.
- The first step is to start up a session. After a session is started, session variables can be created to store information. The PHP session_start() function is used to begin a new session. It also creates a new session ID for the user.
- The data stored in sessions can be easily accessed by firstly calling session_start() and then by passing the corresponding key to the $_SESSION associative array.
- In PHP, a cookie is a small piece of information which is stored at client browser.It is used to recognize the user.
- A cookie is created at server side and saved to client browser.Each time when client sends request to the server, cookie is embedded with request. Such way, cookie can be received at the server side.

Keywords

Session: A session is a way to store information (in variables) to be used across multiple pages. W

Session Identifier: The Session Identifiers (SID) is a unique number which is used to identify every user in a session-based environment.

$_SESSION: The session variables are set with the PHP global variable: $_SESSION.It creates a session or resumes the current one based on a session identifier passed via a GET or POST request or passed via a cookie.

Setcookie():The setcookie() function needs to be called prior to any output generated by the script otherwise the cookie will not be set.

isset() Function*:*This function is used to check whether a cookie is set or not: It is always advisable to check whether a cookie is set or not before accessing its value.

Unset() Function:To delete only a certain session data, the unset feature can be used with the corresponding session variable in the $_SESSION associative array.

Cookie: In PHP, a cookie is a small piece of information which is stored at client browser.It is used to recognize the user.

Self Assessment

1. Which of the following statement is TRUE for the session variables?
A. Session variables store the user information to be used across multiple pages (e.g. username, favoritecolor, etc).
B. Session variables last until the user closes the browser.
C. Session variables hold information about one single user, and are available to all pages in one application.
D. All the above.

2. The full form for SIDs is
A. Session Identifiers
B. Service Identifiers
C. Scope Identifiers
D. Set Identifiers

3. A ___________ is a way to store information (in variables) to be used across multiple pages.
A. Scope
B. Setter
C. Session
D. Spare

4. ______________ is a unique number which is used to identify every user in a session-based environment.
A. Service Identifiers
B. Session Identifiers
C. Scope Identifiers
D. Set Identifiers

5. _________________ is a PHP global variable that is used for setting the session variable.
A. $_SETSESSION
B. $_SESSION
C. $_SESSION_GLOBAL
D. $_SESSIONSETUP

6. The session_start() function creates a session or resumes the current one based on a session identifier passed via a ________ request, or passed via a cookie.
A. GET
B. GETCHS
C. GETCH
D. GETTS

7. The session_destroy() function is used to completely destroy a session.
A. True
B. False

8. Which of the following statement is correct for session_unset()?
A. It deletes only the variables from session and session still exists.
B. Only data is truncated.
C. The syntax for the unset() is bool session_unset(void).
D. All the above.

9. The function used to destroy all the data associated with the current session is _________.
A. session_unset()
B. session_current()
C. session_destroy()
D. currentsession_destroy()

10. The cookie is created at server side and saved to client browser. Each time when client sends request to the server, cookie is embedded with request.
A. True
B. False

11. A cookie in PHP is a small file with a maximum size of 4KB that the __________ stores on the client computer.
A. Web browser
B. Web server
C. Network
D. cacher

12. Which of the following is not a parameter in setcookie() function?
A. Name
B. Value
C. Timer
D. Path

13. ___________ function is used to check whether a cookie is set or not?
A. isset()
B. unset()
C. cookie_set()
D. cookieset_check()

14. You cannot delete a cookie by calling setcookie() function with the cookie name and any value.

A. True

B. False

15. ___________ parameter is used to set the expiry timestamp of the cookie after which the cookie can't be accessed.

A. Timer

B. Session

C. Timestamping

D. Expiry

Answers forSelf Assessment

1. D	2. A	3. C	4. B	5. B
6. A	7. A	8. D	9. C	10. A
11. B	12. C	13. A	14. B	15. D

Review Questions

1. What are the different steps involved in PHP sessions?
2. Discuss the accessing and destroying of the session data.
3. How deletion of cookie can be done?
4. Discuss the session_start() and session_destroy() functions.
5. What are cookies? How cookies can be set and deleted?
6. Explain with example the setting of cookies?
7. What are sessions in PHP? Discuss the steps in session management in PHP?
8. Discuss about the PHP state management.
9. Explain the significance of $_SESSION[]superglobal array?
10. Explain the following in context with cookies:
    a. isset() Function
    b. Setcookie() Function

Further Readings

- The Joy of PHP: A Beginner's Guide by Alan Forbes.
- PHP: The Complete Reference by Steven Holzner. 2017.
- Learning PHP by Ramesh Bangia, 2012.
- Learn PHP in One Day and Learn It Well by Jamie Chan.
- XML and PHP by Vikas Vaswani, 2002.

Web Links

- PHP | Sessions - GeeksforGeeks
- PHP Session - javatpoint

- PHP - Sessions (tutorialspoint.com)
- Session in PHP: Creating, Destroying, and Working With Session in PHP (simplilearn.com)
- PHP - Cookies (tutorialspoint.com)
- https://youtu.be/ngZpU7Q0j2k
- https://youtu.be/opoI4C8xoN8
- https://youtu.be/jort8_4U-88
- https://youtu.be/Ly34Z8DFvdE

Unit 14: Advanced PHP Concepts

CONTENTS

Objectives

After studying this unit, you will be able to:

- Explore the usage of PHP to access database.
- Performing database operations using mysqli.
- Know about the pre-defined variables in PHP and their working.
- Learn the object-oriented concepts in PHP.

Introduction

There are two ways to access databases from PHP:

1. Use a database-specific extension.
2. Use the database-independent PEAR DB library.

MySQL extension's function names, parameters, error handling, and so on are completely different from those of the other database extensions. If you want to move your database from MySQL to PostgreSQL, it will involve significant changes to your code.The PEAR DB, on the other hand, hides the database-specific functions from you, moving between database systems.

14.1 PHP and MySQL Database

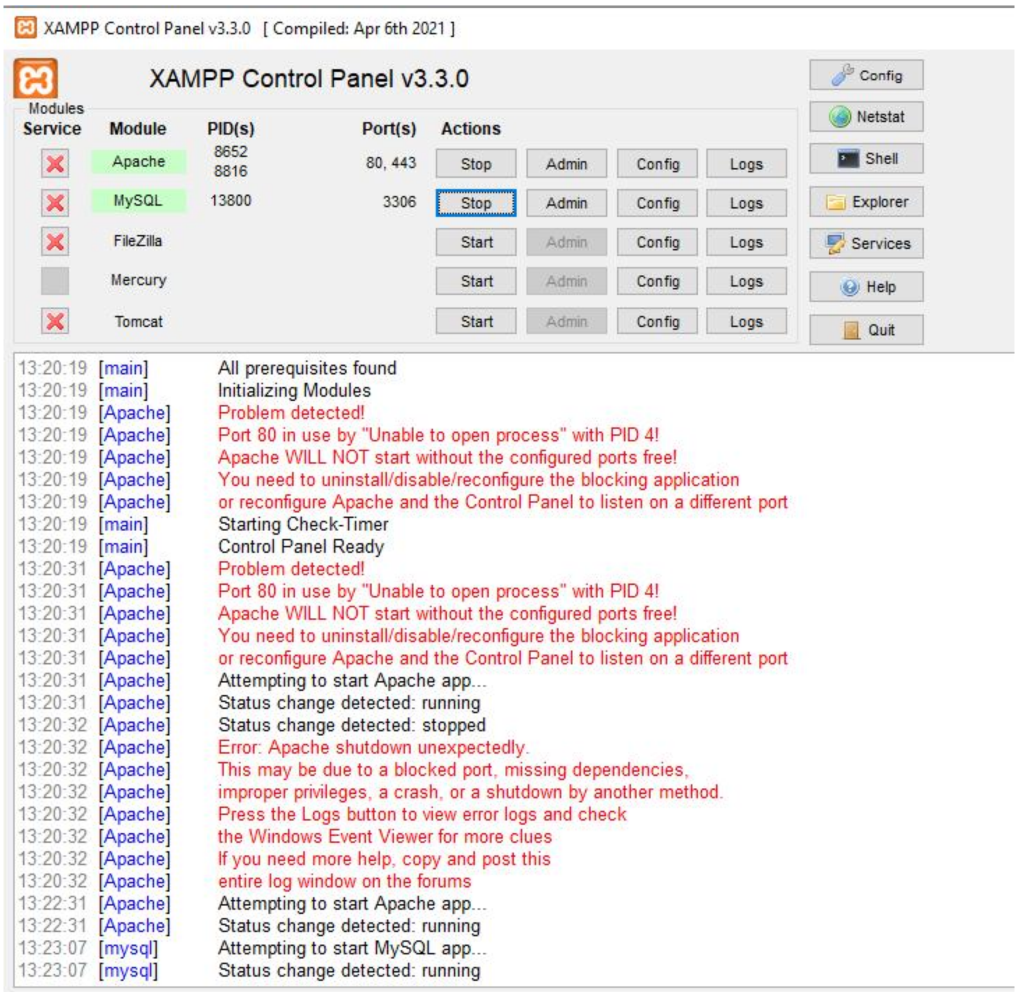
With PHP, you can connect to and manipulate databases.MySQL is the most popular database system used with PHP.MySQL is a database system used on the web. MySQL is a database system that runs on a server. MySQL is ideal for both small and large applications. It is very fast, reliable, and easy to use. It uses standard SQL and compiles on several platforms. MySQL is free to download and use. It has been developed, distributed, and supported by Oracle Corporation. MySQL is named after co-founder Monty Widenius's daughter: My.

The data in a MySQL database are stored in tables. PHP combined with MySQL are cross-platform (you can develop in Windows and serve on a Unix platform). PHP 5 and later can work with a MySQL database using: MySQLi extension (the "i" stands for improved).

Create a Connection to a MySQL Database

- Start XAMPP (Figure 1)
- Start MySQL
- Open browser> type localhost/phpMyAdmin

localhost / 127.0.0.1 | phpMyAdmin 5.1.1



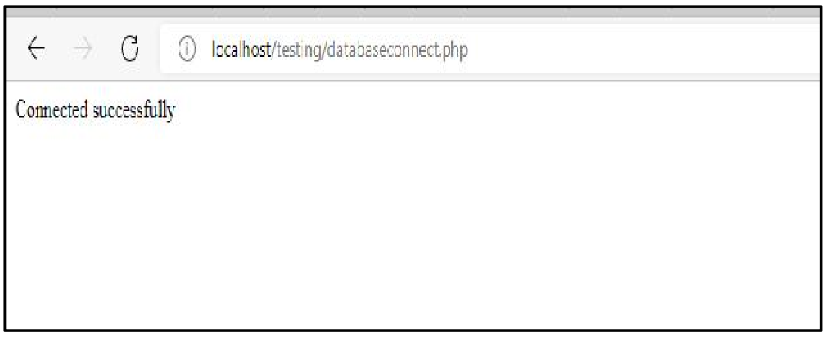
```
13:20:19  [main]      All prerequisites found
13:20:19  [main]      Initializing Modules
13:20:19  [Apache]    Problem detected!
13:20:19  [Apache]    Port 80 in use by "Unable to open process" with PID 4!
13:20:19  [Apache]    Apache WILL NOT start without the configured ports free!
13:20:19  [Apache]    You need to uninstall/disable/reconfigure the blocking application
13:20:19  [Apache]    or reconfigure Apache and the Control Panel to listen on a different port
13:20:19  [main]      Starting Check-Timer
13:20:19  [main]      Control Panel Ready
13:20:31  [Apache]    Problem detected!
13:20:31  [Apache]    Port 80 in use by "Unable to open process" with PID 4!
13:20:31  [Apache]    Apache WILL NOT start without the configured ports free!
13:20:31  [Apache]    You need to uninstall/disable/reconfigure the blocking application
13:20:31  [Apache]    or reconfigure Apache and the Control Panel to listen on a different port
13:20:31  [Apache]    Attempting to start Apache app...
13:20:31  [Apache]    Status change detected: running
13:20:32  [Apache]    Status change detected: stopped
13:20:32  [Apache]    Error: Apache shutdown unexpectedly.
13:20:32  [Apache]    This may be due to a blocked port, missing dependencies,
13:20:32  [Apache]    improper privileges, a crash, or a shutdown by another method.
13:20:32  [Apache]    Press the Logs button to view error logs and check
13:20:32  [Apache]    the Windows Event Viewer for more clues
13:20:32  [Apache]    If you need more help, copy and post this
13:20:32  [Apache]    entire log window on the forums
13:22:31  [Apache]    Attempting to start Apache app...
13:22:31  [Apache]    Status change detected: running
13:23:07  [mysql]     Attempting to start MySQL app...
13:23:07  [mysql]     Status change detected: running
```

Figure 1: XAMPP Control Panel

14.2 Connect to MySQL

The following code creates a connection to a MySQL Database through Code-"databaseconnect.php" as listed below (Figure 2):

```
<?php
$servername = "localhost";
$username = "root";
//$password = "password";
// Create connection
//$conn = new mysqli($servername, $username, $password);
$conn = new mysqli($servername, $username, "" );
// Check connection
if ($conn->connect_error) {
  die("Connection failed: " . $conn->connect_error);
}
echo "Connected successfully";
?>
```



localhost/testing/databaseconnect.php

```
Connected successfully
```

Figure 2: MySQL Database Connected Successfully

14.3 Create Database

The following code illustrates the creation of a database “Dnew” to a MySQL Database through code- “databaseCreate.php” (Figure 3).

```
<?php
//Connecting to database and then creating a database
$c1=mysqli_connect("localhost","root","");
if(!$c1)
die("not connected");
echo "connected";
$q="Create database Dnew";
if(mysqli_query($c1,$q))            {
echo "Database created";    }
else                 {
echo "database not created";          }
mysqli_close($c1);
?>
```

Notes: Check in the interface if database is created

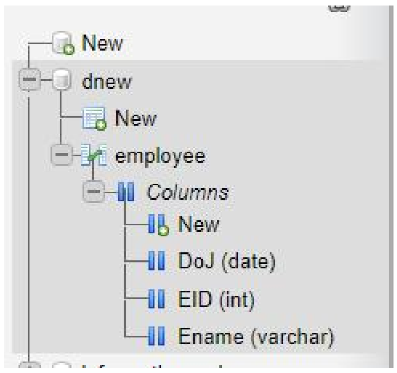


Figure 3: New Database Created "dnew"

14.4 Creating a Table in a Database

There are 2 ways to create table, either directly through browser-based MySQL admin or through code.

1. Through PHPmyadmin:
   - Click on the New Option
   - Enter the table name and columns, specifying the column names and data types (Figure 4).
2. Through Code: You can create a code file in PHP as listed below (Figure 5):

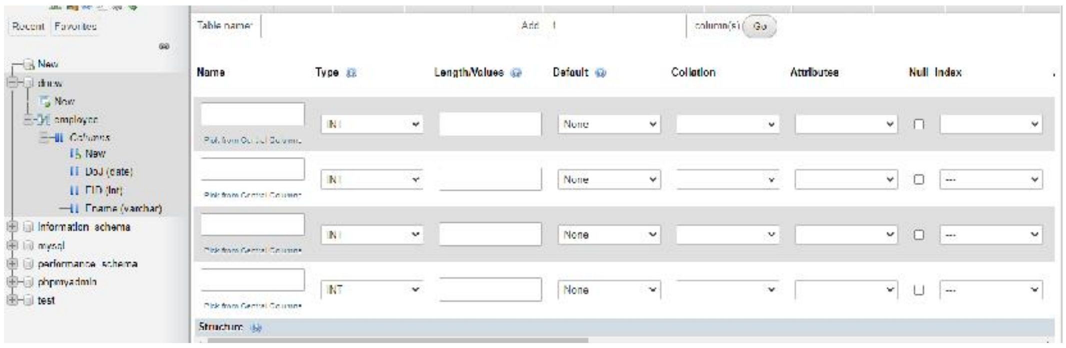
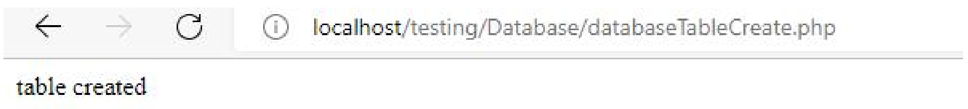


Figure 4: Table Creation in "dnew" Database

```
<?php
$con=mysqli_connect("localhost","root","","Dnew");
$q1="Create table student1(Name varchar(20),Roll integer(4))";
if(mysqli_query($con,$q1))
{
echo “table created";
}
else
{echo "not created";
}
mysqli_close($con);
?>
```



```
localhost/testing/Database/databaseTableCreate.php
table created
```

Figure 5: Table Created

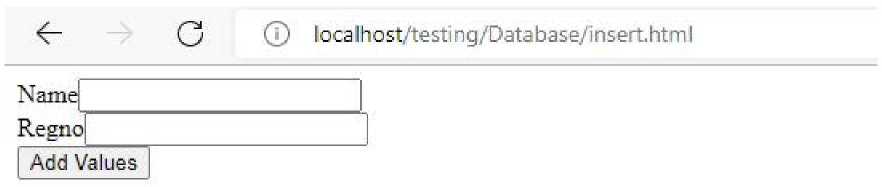
14.5 Inserting Data in a Database

Under this, a user input can be retrieved and then the record can be inserted to the table specified.

1. Design a HTML form to get the user input: Save it as insert.html (Figure 6).
2. Design the PHP page to insert the user input.

Example: <form name=f1 action="insertn.php" method="POST">

```
Name<input type="text" name="Name"><br>
Regno<input type="text" name="Roll"><br>
<input type="Submit" value="Add Values">
</form>
```



```
localhost/testing/Database/insert.html
Name
Regno
Add Values
```

Figure 6: Form Creation to Get User Input and Insertion in the Database

Example:<?php

```
        $n=$_POST["Name"];
        $r=$_POST["Roll"];
$c=mysqli_connect("localhost","root","","dnew"); //database name and connection
        if(!$c)
        die("cannot connect"."<br>");
        echo "connected";
        echo "<br>";
//Insert record to the table student1 through the HTML form
        $s="insert into student1(Name,Roll)value('$n','$r')";
        if(mysqli_query($c,$s))     {
        echo "Record inserted";     }
        else     {
        echo"Problem in command";          }
        mysqli_close($c);
?>
```

14.6 Updating Records in the Database

Example: <?php

```
$c=mysqli_connect("localhost","root","","dnew");
if(!$c)
die("cannot connect"."<br>");
echo "connected";
echo "<br>";
$s="update student1 set Name='Gurfateh' where Roll=123";
if(mysqli_query($c,$s))
{
echo "Record updated";
}
else{echo"Problem in command";}
mysqli_close($c);
?>
```

Example: Deleting the Data in Database

```
<?php
$c=mysqli_connect("localhost","root","",“dnew");
if(!$c)
die("cannot connect"."<br>");
echo "connected";
echo "<br>";
$s="delete from student1 where Roll=123";
if(mysqli_query($c,$s))
{
echo "Record Deleted";
}
else{echo"Problem in command";}
mysqli_close($c);
?>
```

Lab Exercise: Illustrate the selection of data from a database.

14.7 Pre-defined Variables

PHP provides many predefined variables to any script which it runs. PHP provides an additional set of predefined arrays containing variables from the web server, the environment, and user input. The new arrays are called superglobals as listed below:

- $GLOBALS: Contains a reference to every variable which is currently available within the global scope of the script. The keys of this array are the names of the global variables.
- $_GET: An associative array of variables passed to the current script via the HTTP GET method.
- $_POST: An associative array of variables passed to the current script via the HTTP POST met.
- $_FILES: An associative array of items uploaded to the current script via the HTTP POST method.
- $_REQUEST: An associative array consisting of the contents of $_GET, $_POST, and $_COOKIE.
- $_COOKIE: An associative array of variables passed to the current script via HTTP cookies.
- $_SESSION: An associative array containing session variables available to the current script.
- $_PHP_SELF: A string containing PHP script file name in which it is called.
- $php_errormsg: $php_errormsg is a variable containing the text of the last error message generated by PHP.
- $_SERVER: This is an array containing information such as headers, paths, and script locations. The entries in this array are created by the web server. There is no guarantee that every web server will provide any of these.

14.8 Pre-defined Variables- $_SERVER

Table 1lists the different variables of super global array $_SERVER. Figure 7 depicts the output for using $_SERVER variable through serverdetail.php.

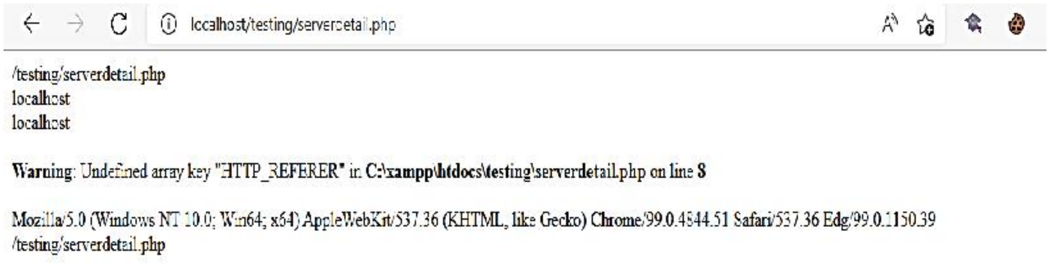
Table 1: $_SERVER Variable

Variable	Description
$_SERVER['PHP_SELF']	The filename of the currently executing script, relative to the document root
$_SERVER['argv']	Array of arguments passed to the script. When the script is run on the command line, this gives C-style access to the command line parameters. When called via the GET method, this will contain the query string.
$_SERVER['argc']	Contains the number of command line parameters passed to the script if run on the command line.
$_SERVER['SERVER_ADDR']	The IP address of the server under which the current script is executing.
$_SERVER['SERVER_NAME']	The name of the server host under which the current script is executing. If the script is running on a virtual host, this will be the value defined for that virtual host.
$_SERVER['SERVER_SOFTWARE']	Server identification string, given in the headers when responding to requests.

$_SERVER['REQUEST_METHOD']	Which request method was used to access the page; i.e. 'GET', 'HEAD', 'POST', 'PUT'.

Example: Program to illustrate the use of pre-defined variables.

```
<?php                //serverdetail.php
echo $_SERVER['PHP_SELF'];
echo "<br>";
echo $_SERVER['SERVER_NAME'];
echo "<br>";
echo $_SERVER['HTTP_HOST'];
echo "<br>";
echo $_SERVER['HTTP_REFERER'];
echo "<br>";
echo $_SERVER['HTTP_USER_AGENT'];
echo "<br>";
echo $_SERVER['SCRIPT_NAME'];
?>
```



```
/testing/serverdetail.php
localhost
localhost

Warning: Undefined array key "HTTP_REFERER" in C:\xampp\htdocs\testing\serverdetail.php on line 8

Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/99.0.4844.51 Safari/537.36 Edg/99.0.1150.39
/testing/serverdetail.php
```

Figure 7: Output for Using $_SERVER Variable

14.9 Object-oriented Concepts

We can imagine our universe made of different objects like sun, earth, moon etc. Similarly, we can imagine our car made of different objects like wheel, steering, gear etc. Similarly, there is object-oriented programming concepts which assume everything as an object and implement a software using different objects.

14.10 Important Terms Related to Object-Oriented Concepts

- Class− this is a programmer-defined data type, which includes local functions as well as local data. You can think of a class as a template for making many instances of the same kind (or class) of object.
- Object− an individual instance of the data structure defined by a class. You define a class once and then make many objects that belong to it. Objects are also known as instance.

- Member variable– these are the variables defined inside a class. This data will be invisible to the outside of the class and can be accessed via member functions. These variables are called attribute of the object once an object is created.
- Member function– these are the function defined inside a class and are used to access object data.
- Inheritance– When a class is defined by inheriting existing function of a parent class then it is called inheritance. Here child class will inherit all or few member functions and variables of a parent class.
- Parent Class– A class that is inherited from by another class. This is also called a base class or super class.
- Child Class– A class that inherits from another class. This is also called a subclass or derived class.
- Polymorphism– this is an object-oriented concept where same function can be used for different purposes. For example, function name will remain same, but it takes different number of arguments and can-do different task.
- Overloading– A type of polymorphism in which some or all of operators have different implementations depending on the types of their arguments. Similarly, functions can also be overloaded with different implementation.
- Data Abstraction– any representation of data in which the implementation details are hidden (abstracted).
- Encapsulation– Refers to a conceptwhere we encapsulate all the data and member functions together to form an object.
- Constructor– Refers to a special type of functionwhich will be called automatically whenever there is an object formation from a class.
- Destructor– Refersto aspecial type of function which will be called automatically whenever an object is deleted or goes out of scope.

14.11 Defining PHP Classes

The general form for defining a new class in PHP is as follows:

Example:
```
<?php
class phpClass
{
        var $var1;
        var $var2 = "constant string";
        function myfunc ($arg1, $arg2)
        { }
}
?>
```

Here is the description of each line: The special form class, followed by the name of the class that you want to define.A set of braces enclosing any number of variable declarations and function definitions. The variable declarations start with the special form var, which is followed by a conventional $ variable name; they may also have an initial assignment to a constant value. The function definitions look much like standalone PHP functions but are local to the class and will be used to set and access object data.

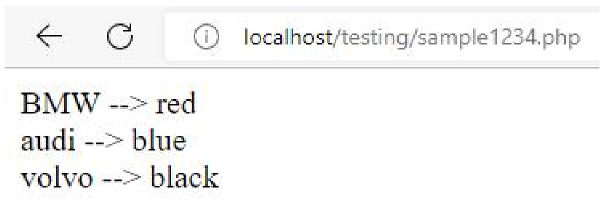
14.12 Creating Objects and Calling Methods in PHP

Once you defined your class, then you can create as many objects as you like of that class type. Following is an example of how to create object using new operator:

```
$physics = new Books;
$maths = new Books;
$chemistry = new Books;
```

Here,we have created three objects and these objects are independent of each other and they will have their existence separately. The example for creating objects and calling methods is shown below. Figure 8 depicts the output for the code, “sample1234.php”.

```
Example: <?php                          //sample1234.php
class cars {
 //Properties
 var $name;
 var $color;
 //Methods
 function set_name($name) {
  $this->name = $name;
 }
 function get_name() {
  return $this->name;
 }
  function set_color($color) {
  $this->color = $color;
 }
 function get_color() {
  return $this->color;
 }
}
$BMW = new cars();
$audi = new cars();
$volvo = new cars();
$BMW->set_name('BMW ');
$BMW->set_color('red');
$audi->set_name('audi ');
$audi->set_color('blue');
$volvo->set_name('volvo ');
$volvo->set_color('black');
echo $BMW->get_name();
echo " --> ";
echo $BMW->get_color();
echo "<br>";
echo $audi->get_name();
echo " --> ";
echo $audi->get_color();
echo "<br>";
echo $volvo->get_name();
echo " --> ";
echo $volvo->get_color();
?>
```



```
BMW --> red
audi --> blue
volvo --> black
```

Figure 8: Output for sample1234.php

Summary

- MySQL is the most popular database system used with PHP. MySQL is a database system used on the web that runs on a server.
- MySQL uses standard SQL and compiles on a number of platforms. MySQL is free to download and use. It is developed, distributed, and supported by Oracle Corporation.
- PHP combined with MySQL are cross-platform (you can develop in Windows and serve on a Unix platform). PHP 5 and later can work with a MySQL database using: MySQLi extension (the "i" stands for improved).
- PHP provides many predefined variables to any script which it runs. PHP provides an additional set of predefined arrays containing variables from the web server the environment, and user input.
- In object-oriented programming, everything as an object and implement a software using different objects.
- Once you defined your class, then you can create as many objects as you like of that class type.

Keywords

Database:A database is a collection of information that is organized so that it can easily be accessed, managed, and updated. In one view, databases can be classified according to types of content: bibliographic, full-text, numeric, and images.

MySQL:MySQL is the most popular database system used with PHP. MySQL is a database system used on the web that runs on a server.

Objects:An individual instance of the data structure defined by a class. You define a class once and then make many objects that belong to it. Objects are also known as instance.

Class:This is a programmer-defined data type, which includes local functions as well as local data. You can think of a class as a template for making many instances of the same kind (or class) of object.

Inheritance:When a class is defined by inheriting existing function of a parent class then it is called inheritance. Here child class will inherit all or few member functions and variables of a parent class.

Polymorphism:This is an object-oriented concept where same function can be used for different purposes. For example, function name will remain same, but it takes different number of arguments and can-do different task.

Self Assessment

1. MySQL database is of type
A. Access Jet Database Engine
B. Object-relational DBMS
C. General Public
D. Single platform database

2. ______________database is one that is congruent with the data defined in object classes and subclasses.
A. Convergent
B. Object-oriented programming
C. Class
D. Insecure

3. MySQL is the most popular database system used with PHP. Which of the following statement(s) is/are true for MySQL?
A. MySQL is a database system that runs on a server
B. MySQL uses standard SQL
C. MySQL compiles on several platforms
D. All the above

4. The MySQL extension's function names, parameters, error handling, and so on are completely different from those of the other database extensions.
A. True
B. False

5. In MySQLi extension, the 'i' is abbreviation that stands for
A. Impact
B. Included
C. Injected
D. Improved

6. Which of the following statement(s) is/are true?
A. XAMPP stands for cross-platform, Apache, MySQL, PHP, and Perl.
B. XAMPP is among the simple light-weight local servers for website development.
C. In PHP, we can connect to database using localhost XAMPP web server.
D. All the above.

7. PHP provides an additional set of predefined arrays containing variables from the web server the environment, and user input. The new arrays are called as_____________.

A. Varglobal
B. Subglobals
C. Superglobals
D. Preset globals

8. ________ is an associative array of items uploaded to the current script via the HTTP POST method.

A. $_DELETE
B. $_SETTING
C. $_FILES
D. $_GET

9. ____________ is an associative array consisting of the contents of $_GET, $_POST, and $_COOKIE.

A. $_REQUEST
B. $_SET
C. $_DELETE
D. $_SETTING

10. Which of the following statement is TRUE for the $_SERVER?

A. This is an array containing information such as headers, paths, and script locations.
B. The entries in this array are created by the web server.
C. There is no guarantee that every web server will provide any of these.
D. All the above.

11. ___________ is a programmer-defined data type, which includes local functions as well as local data.

A. Class
B. Pointer
C. Degree
D. Inheritance

12. An object is also called as a(n) ________.

A. Class
B. Instance
C. Extraction
D. Domain

13. Which of the following statement aptly indicates the member variables?

A. Member variables are defined inside a class.

B. The data will be invisible to the outside of the class and can be accessed via member functions.
C. Member variables are also called attribute of the object once an object is created.
D. All the above.

14. Polymorphism is an object-oriented concept where same function can be used for different purposes.
A. True
B. False

15. Any representation of data in which the implementation details are hidden represents ____________.
A. Data extraction
B. Data exclusion
C. Data abstraction
D. Data surfacing

Self Assessment

1.	B	2.	B	3.	D	4.	A	5.	D
6.	D	7.	C	8.	C	9.	A	10.	D
11.	A	12.	B	13.	D	14.	A	15.	C

Review Questions

1. How PHP can be used to access database?
2. How to insert form data into database using PHP?
3. How do we access a database using PHP? Write the steps of database connectivity withPHP?
4. How can the objects be created?
5. Discuss creating class and calling methods with examples.
6. Write a short note on:
   a. Objects
   b. Pre-defined Variables
7. Explain object-oriented programming concepts.
8. List the different variables of super global array $_SERVER.
9. How are the PHP classes defined?
10. Discuss the usage of PHP and MySQL Database.

Further Readings

- The Joy of PHP: A Beginner's Guide by Alan Forbes.
- PHP: The Complete Reference by Steven Holzner. 2017.
- Learning PHP by Ramesh Bangia, 2012.

- Learn PHP in One Day and Learn It Well by Jamie Chan.
- XML and PHP by Vikas Vaswani, 2002.

Web Links

- PHP MySQL Connect to database (w3schools.com)
- How to Connect Database to PHP? | Learn to Connect Database to PHP (educba.com)
- Accessing SQL Server Databases from PHP - TechNet Articles - United States (English) - TechNet Wiki (microsoft.com)
- PHP Predefined Variables (tutorialspoint.com)
- PHP: Predefined Variables - Manual
- PHP - Predefined Variables (tutorialspoint.com)
- Object Oriented Concepts in PHP | PHP tutorial by Wideskills
- PHP OOP Intro (w3schools.com)
- https://youtu.be/Uy5pN1c9KUU
- https://youtu.be/2gwXRo-FKW0
- https://youtu.be/nN2mIbspaFI

LOVELY PROFESSIONAL UNIVERSITY
Jalandhar-Delhi G.T. Road (NH-1)
Phagwara, Punjab (India)-144411
For Enquiry: +91-1824-521360
Fax.: +91-1824-506111
Email: odl@lpu.co.in

ISBN 978-81-19334-31-5
9 788119 334315