Web Technologies

DECAP472

Edited by
Dr. Pawan Kumar

LOVELY
PROFESSIONAL
UNIVERSITY

LOVELY
PROFESSIONAL
UNIVERSITY

Web Technologies

Edited By:
Dr. Pawan Kumar

CONTENT

Unit 01: Introduction to HTML5

CONTENTS

Objectives

After studying this unit, you will be able to:

- Discuss introduction regarding HTML5
- Explain document development environment
- Known how MacOS environment setup done
- Describe GitHub and browser Sync.

Introduction

HTML 5 is a updation of the Hypertext Markup Language (HTML), which is a typical programming language for displaying dynamic content on Web pages.

HTML5 was developed to remove the compatibility issue which was affecting the standard HTML4. One of the major differences between HTML5 and its older versions was proprietary plugins and APIs (for this web page built was tested in one browser as it may not run on another browser). It allows providing an interactive interface for loading all types of elements easily. For example, In HTML5 we need not install Flash plugins it will run by default in the browser.

1.1 HTML AND HTML5

HTML abbreviation consists of two words that have a high significance

Hypertext: Text (it often has embedded images, videos etc.) that is organized in order to link related items

Markup: A style guide for formatting whatsoever to be printed in hardcopy or soft copy format

Language: A language that a computer system understands and uses to execute commands.

It determines the basic structure of web pages; only basic HTML is not enough to make the webpage look dynamic and interactive.

On the other hand, HTML5 supports multimedia on mobile devices. New syntactic features were introduced to support this, such as video, audio and canvas tags. HTML5 also introduces new features which can really change the way users interact with documents including:

- New parsing rules for boosting flexibility
- Updated attributes
- Exclusion of outdated or redundant attributes
- Drag and drop capabilities from one HTML5 document to another
- Offline editing
- Messaging enhancements
- Detailed rules for parsing
- MIME and protocol handler registration
- A common standard for storing data in SQL databases (Web SQL)

Example: HTML5 was implemented by the new working group of the World WidHe TWMeLb5 Consortium (W3C) in 2007, as this group published the first public draft of HTML 5 in January 2008.

1.2 First HTML5 Document Development Environment Setup

HTM5 language has a custom syntax for HTML language which is well-suited with HTML 4 and XHTML1 documents published on the Web, but that has not enough compatibility with some of the hidden SGML features of HTML4.

HTML5 is not based on any particular rule or syntax as XHTML in which we use lower case tag names, quotes our attributes, all attribute that have a value and to close all empty elements.

HTML5 is more robust, flexible and comes with lot of supporting features:

- Uppercase tag names.
- Quotes are optional for attributes.
- Attribute values are optional.
- Closing empty elements are optional.

The DOCTYPE

It is the older version of HTML with longer DOCTYPEs and the HTML language was based on the SGML and mandatory to have a reference to a DTD.

Syntax used in HTML5 by authors to specify DOCTYPE and it is case sensitive.

```
<!DOCTYPEhtml>
```

Character Encoding

HTML5 Developers can use simple syntax for Character Encoding below-

```
<metacharset="UTF-8">
```

Script Tag <script>

```
<scripttype="text/javascript"src="xyz.js"></script>
```

It is a common tag practiced to adda attribute having value “text/javascript” below is the syntax –

HTML 5 has made the tag simpler to use removing additional information syntax as below –

```
<scriptsrc="xyz.js"></script>
```

LINK tag

```
<linkrel="stylesheet"type="text/css"href="xyz.css">
```

In HTML LINK tag is written as-

HTML 5 has made the tag simpler to use removing additional information syntax as below –

```
<linkrel="stylesheet"href="stylefile.css">
```

1.3 HTML5 Elements

HTML5 tags are marked with starting tag and an ending tag. Tags are delimited by the use of angle brackets with the tag name placed in between the tags.

The difference between both the tags is that in the closing tag slash is used before the tag name.

Following is the example of an HTML5 element –

```
<p>...</p>
```

HTMl5 tags are mostly case insensitive and can be written in all cases either uppercase, lower or mixed case preferably lower case is recommended as a conventional method.

Utmost the elements in HTML5 have some content like <p>…</p> contains in paragraph. Elements like br, hr, link, meta etc. are forbidden from containing any content at all and are known as void elements

Some of the elements are shown below

:

Tag	Description
<!--...-->	Specifiesacomment
<!DOCTYPE>	Specifiesthedocumenttype
<a>	Specifiesananchor
<abbr>	Specifiesanabbreviation
<address>	Specifiesanaddresselement
<applet>	Deprecated:Specifiesanapplet

1.4 HTML5 Attributes

Elements may contain attributes that are used to set various properties of an element.

Some attributes are defined globally and can be used on any element, while others are defined for specific elements only. All attributes have a name and a value and look like as shown below in the example.

Following is the example of an HTML5 attribute that illustrates how to mark up a div element with

an attribute named class using a value of "example" −

<div class="example">...</div>

Attributes may only be specified within start tags and must never be used in end tags.

HTML5 attributes are case insensitive and may be written in all uppercase or mixed cases, although the most common convention is to stick with lowercase.

Attribute	Options	Function
accesskey	UserDefined	Specifiesakeyboardshortcuttoaccessanelement.
align	right,left,center	Horizontallyalignstags
background	URL	Placesanbackgroundimagebehindanelement
bgcolor	numeric,hexidecimal,RGBvalues	Placesabackgroundcolorbehindanelement
class	UserDefined	Classifies an element for use with Cascading StyleSheets.
contenteditable	true,false	Specifiesiftheusercanedittheelement'scontentornot.
contextmenu	Menuid	Specifiesthecontextmenuforanelement.
data-XXXX	UserDefined	Customattributes.AuthorsofaHTMLdocumentcan definetheirownattributes.Muststartwith "data-".

1.5 First HTML5 Document Development Environment Setup

HTML and CSS are languages that are just interpreted by any web browser. They don't act like other conventional programming languages which are compiled into machine code or interpreted in a virtual machine. HTML and CSS can be written in anything - even as basic as notepad. These are grouped as "text editors" and can be powerful and help your workflow if you master one. The most important thing is you find an environment which is comfortable for you.

How to setup HTML/CSS environment

The main ones in order to benefit workflow are Sublime Text and Notepad++. Both have great plugins and themes available to it, but Notepad++ is only available on Windows, whereas Sublime Text is cross-platform. Once you choose an environment, it's as simple as adding to the beginning. You can then type anything and when opened with a web browser you will see a very basic browse

Following are the Points:

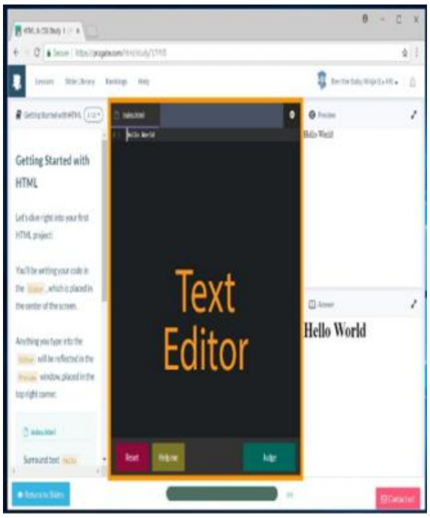
- The Text Editor.

- Browser Setup.
- Your Working Folder.
- Creating an HTML File.

Generally, it's called local development when you create and test a website using only your own computer. The setup required is called local environment setup. After you have set up your local development environment, you can work on creating your own website with HTML & CSS anytime and anywhere (even without an internet connection).

Text Editor

The two most essential tools for developing a website are a text editor and a browser. We'll start by choosing and setting up a text editor. A text editor (or just "editor") is like the window in the center of every lesson where you write your code. You'll need an editor to do any kind of programming, not just HTML & CSS.

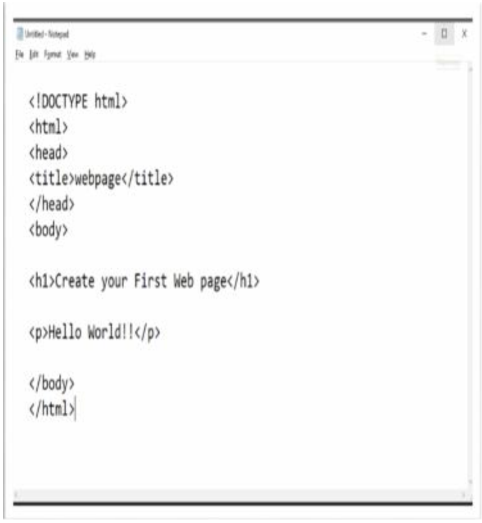


Getting Started with HTML

Let's dive right into your first HTML project!

You'll be writing your code in the Editor, which is placed in the center of the screen.

Anything you type into the Editor will be reflected in the Preview window placed in the top right corner.

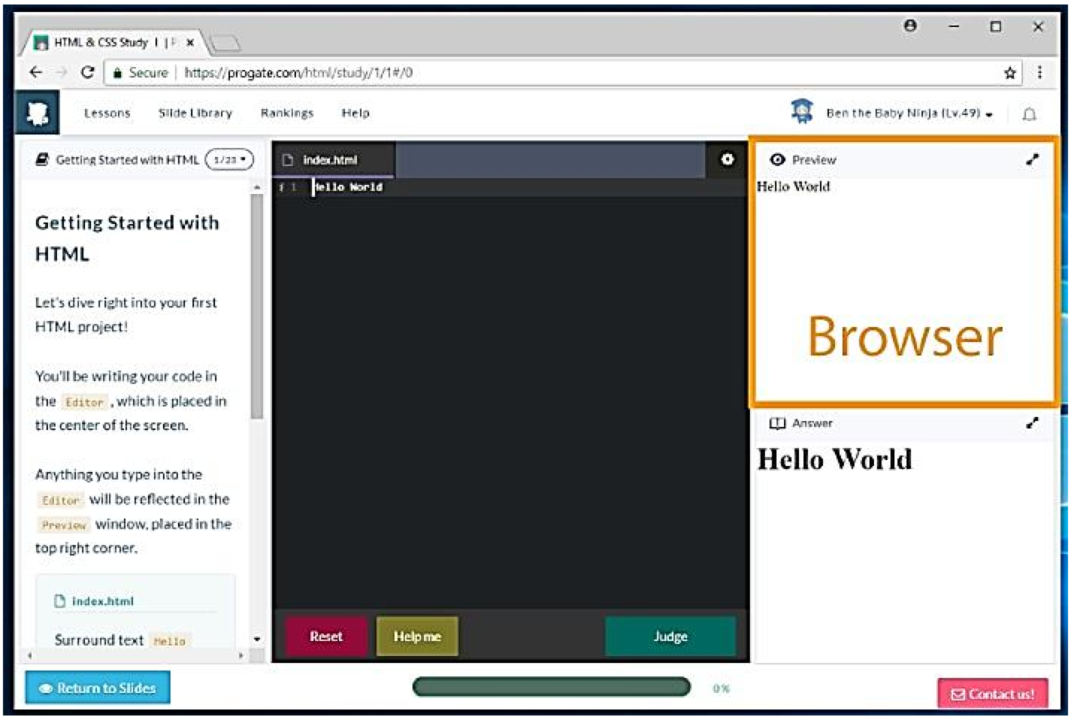


```
<!DOCTYPE html>
<html>
<head>
<title>webpage</title>
</head>
<body>

<h1>Create your First Web page</h1>

<p>Hello World!!</p>

</body>
</html>
```

An HTML editor is a program for editing HTML, the markup of a web page. Although the HTML markup in a web page can be controlled with any text editor, specialized HTML editors can offer convenience and added functionality.

Browser Setup

Browsers are for viewing and displaying websites, like the one you're using to view this article (most likely). There is a window on the right where the result of your HTML & CSS code is displayed so you can see if it's correct.



Getting Started with HTML

Let's dive right into your first HTML project!

You'll be writing your code in the `Editor`, which is placed in the center of the screen.

Anything you type into the `Editor` will be reflected in the `Preview` window, placed in the top right corner.

index.html

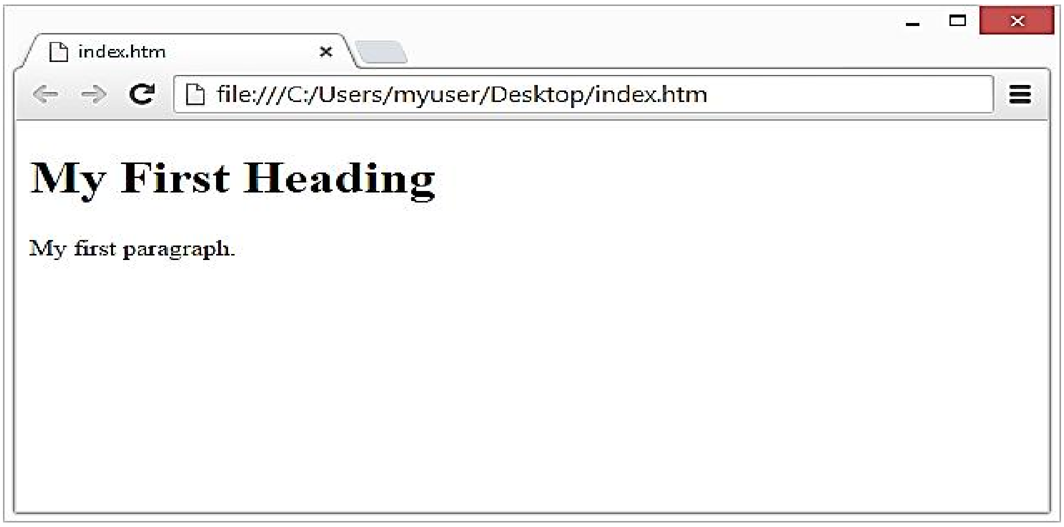
Surround text `Hello`

```
Hello World
```

Hello World

Hello World

The Text Display in Browser will appear like in diagram



My First Heading

My first paragraph.

OUR FIRST HTML DOCUMENT

```
<html>

<head>

<title>Welcome to HTML5</title>

</head>

<body>

<h1> Hello Welcome to HTML</h1>
```

```
<h2> Hello Welcome to HTML</h2>
<h3> Hello Welcome to HTML</h3>
<p> This is a Paragraph</p>
</body>
</html>
```

Output:

Hello Welcome to HTML

Hello Welcome to HTML

Hello Welcome to HTML

This is a Paragraph

Create own HTML5 File

To create a HTML5 document we have to first make a folder and name that particular folder. In that folder we have to put our notepad file of HTML5 with extension .html or .htm. Following steps to be preformed after mamking the folder

- "Normal text" surrounded by bracketed tags that tell browsers how to display web pages
- Pages end with ".htm" or ".html"
- HTML Editor – A word processor that has been specialized to make the writing of HTML documents more effortless.

There are many different programs that you can use to create web documents. HTML Editors enable users to create documents quickly and easily by pushing a few buttons. Instead ofentering all of the HTML codes by hand. HTML Editors are excellent tools for experienced web developers; however; it is important that you learn and understand the HTML language so that you can edit code and fix "bugs" in your pages.

For this Course, we will focus on using the standard Microsoft Windows text editors, NotePad. These programs will generate the HTML Source Code for you.

Starting Notepad

NotePad is the standard text editor that comes with the microsoft windows operating system. To start NotePad in windows 9x or XP follow the steps bellow:

Click on the "Start" button located on your Windows task bar.

Click on "Programs" and then click on the directory menu labeled "Accessories".

Locate the shortcut "NotePad" and click the shortcut once.

Now we can use the tags to implement a HTML5 document. The HEAD of your document point to above window part. The TITLE of your document appears in the very top line of the user's browser. If the user chooses to "Bookmark" your page or save as a "Favorite"; it is the TITLE that is added to the list.

The text in your TITLE should be as descriptive as possible because this is what many search engines, on the internet, use for indexing your site. Document properties are controlled by attributes of the BODY element. For example, there are color settings for the background color of the page, the document's text and different states of links.

Colors are set using "RGB" color codes, which are, represented as hexadecimal values. Each 2-digit section of the code represents the amount, in sequence, of red, green or blue that forms the color. For example, a RGB value with 00 as the first two digits has no red in the color.

Previewing HTML5 Document

Once you have created your basic starting document and set your document properties it is a good idea to save your file.

To save a file, in NotePad, follow these steps:

- Locate and click on the menu called "File".
- Select the option under File Menu labeled "Save As".
- In the "File Name" text box, type in the entire name of your file (including the extensionname .html).

1.6 HTML documents in TextEdit on Mac

- In the TextEdit app on your Mac, choose File > New, then choose Format > Make Plain Text.
- Enter the HTML code.
- Choose File > Save, type a name followed by the extension .html (for example, enter index.html), then click Save.
- When prompted about the extension to use, click "Use .html".

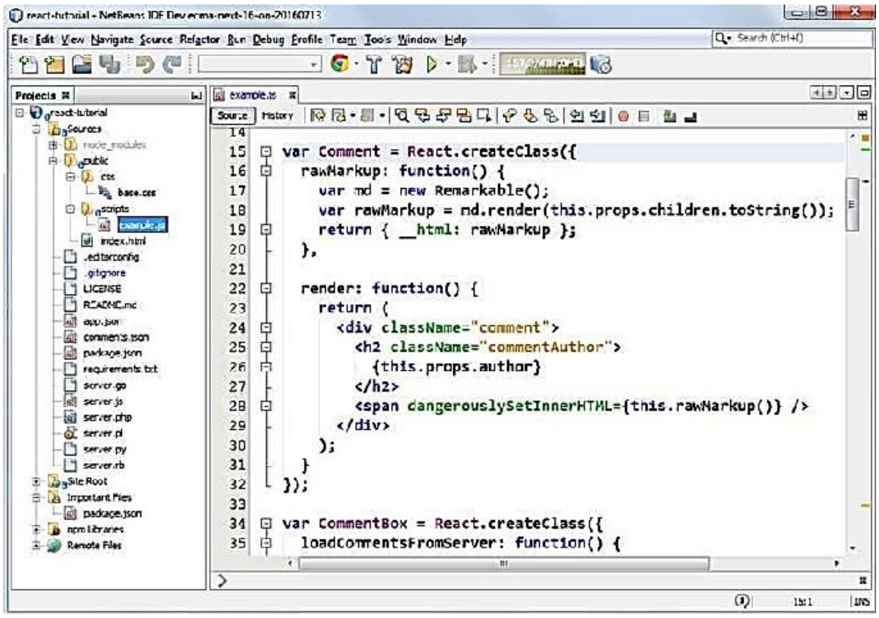
View as HTML Document

In the TextEdit app on your Mac, choose TextEdit > Preferences, then click Open and Save.Select

- "Display HTML files as HTML code instead of formatted text".
- set preferences that affect how HTML files are saved in TextEdit.
- In the TextEdit app on your Mac, choose TextEdit > Preferences, then click Open and Save.
- Below HTML Saving Options, choose a document type, a style setting for CSS and an encoding.
- Select "Preserve white space" to include code that preserves blank areas in documents.
- If you open an HTML file and don't see the code, TextEdit is displaying the file the sameway a browser would (as formatted text).

NetBeans

NetBeans is an open source IDE is used for developing software applications by developers. It can read and edit texts in HTML. PHP, C++ and a few more. It provides an intuitive user interface for macOS 10.14 users and comes with customized key-bindings. Interestingly, it supports Git and management of source codes.



```
var Comment = React.createClass({
  rawMarkup: function() {
    var md = new Remarkable();
    var rawMarkup = md.render(this.props.children.toString());
    return { __html: rawMarkup };
  },

  render: function() {
    return (
      <div className="comment">
        <h2 className="commentAuthor">
          {this.props.author}
        </h2>
        <span dangerouslySetInnerHTML={this.rawMarkup()} />
      </div>
    );
  }
});

var CommentBox = React.createClass({
  loadCommentsFromServer: function() {
```

Sublime Text 2

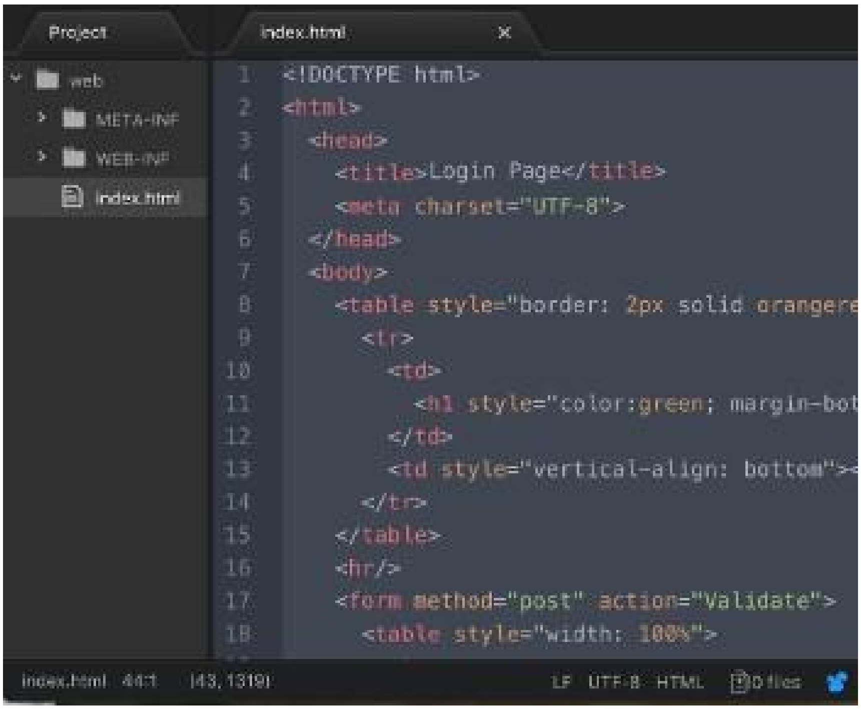
Sublime Text 2 is the minimalistic favorite of developers, because it can easily be completely customized to one’s own needs by using a JSON file. Moreover, there are several extensions to “stretch” the editor with. For example, a must-have extension is Package Control, which you can use to easily find and install extensions and plug-ins from right within the editor. To describe the full extent of features of Sublime Text 2 is rather difficult, as it can do, through its plug-ins and extensions, basically anything you need it to do.

Brackets

Brackets is a modern open source editor with several extremely interesting features. For example, when used in combination with Adobe Creative Cloud Extract (=preview) it can read design data such as colors, types, histories etc. directly from a PSD file and transform it into minimalistic, correct CSS code

Atoms

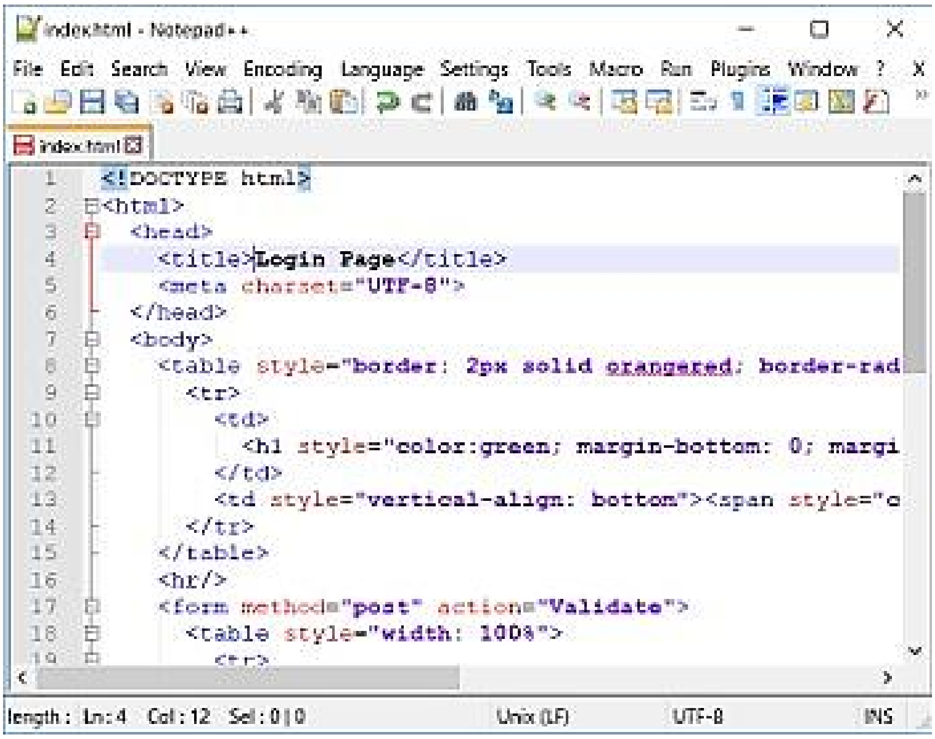
Atom is an open-source and free to use text and code editor with a rich set of plug-ins for different functions. It can be downloaded from Atom homepage, and it runs on Microsoft Windows, macOS, and Linux distributions. Atom highlights code written in many languages, including HTML, Java, Python, and others. Sample HTML code displayed in the Atom editor is shown in Figure



```
<!DOCTYPE html>
<html>
  <head>
    <title>Login Page</title>
    <meta charset="UTF-8">
  </head>
  <body>
    <table style="border: 2px solid orangere
      <tr>
        <td>
          <h1 style="color:green; margin-bot
        </td>
        <td style="vertical-align: bottom"><
      </tr>
    </table>
    <hr/>
    <form method="post" action="Validate">
      <table style="width: 100%">
```

Notepad++

Notepad++ is a free-to-use editor that supports several programming languages. It runs on Microsoft Windows and doesn't have packages for other operating systems. You can download Notepad++ from Notepad++ home.


index.html - Notepad++
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ? X
index.html
length : Ln : 4 Col : 12 Sel : 0 | 0
Unix (LF)
UTF-8
INS

```
<!DOCTYPE html>
<html>
  <head>
    <title>Login Page</title>
    <meta charset="UTF-8">
  </head>
  <body>
    <table style="border: 2px solid orangered; border-rad
      <tr>
        <td>
          <h1 style="color:green; margin-bottom: 0; margi
        </td>
        <td style="vertical-align: bottom"><span style="c
      </tr>
    </table>
    <hr/>
    <form method="post" action="Validate">
      <table style="width: 100%">
        <tr>
```

GitHub

To be very crisp about what exactly is GitHub, it is a file or code-sharing service to collaborate with different people.

GitHub is a highly used software that is typically used for version control.

It is helpful when more than just one person is working on a project. Say for example, a software developer team wants to build a website and everyone has to update their codes simultaneously while working on the project. In this case, Github helps them to build a centralized repository where everyone can upload, edit, and manage the code files.

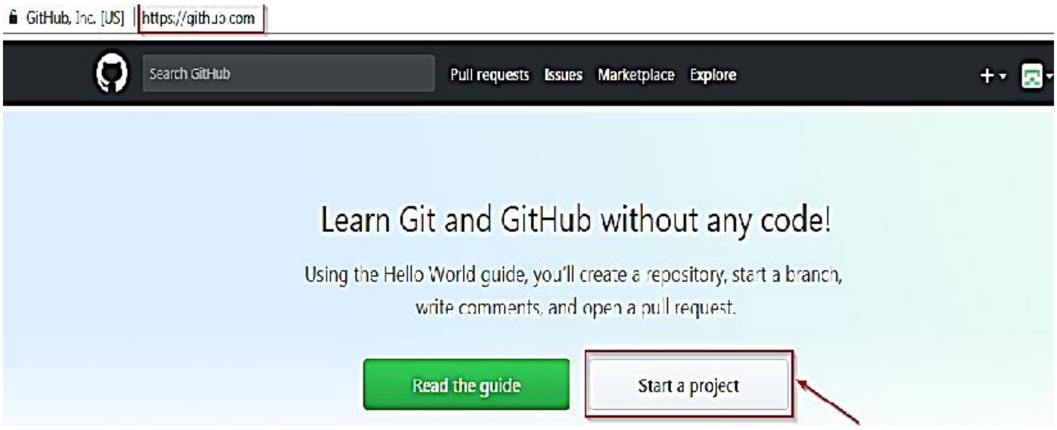
GitHub Repository

A repository is a storage space where your project lives. It can be local to a folder on your computer, or it can be a storage space on GitHub or another online host. You can keep code files, text files, images or any kind of a file in a repository.

You need a GitHub repository when you have done some changes and are ready to be uploaded. This GitHub repository acts as your remote repository.

Start a new project

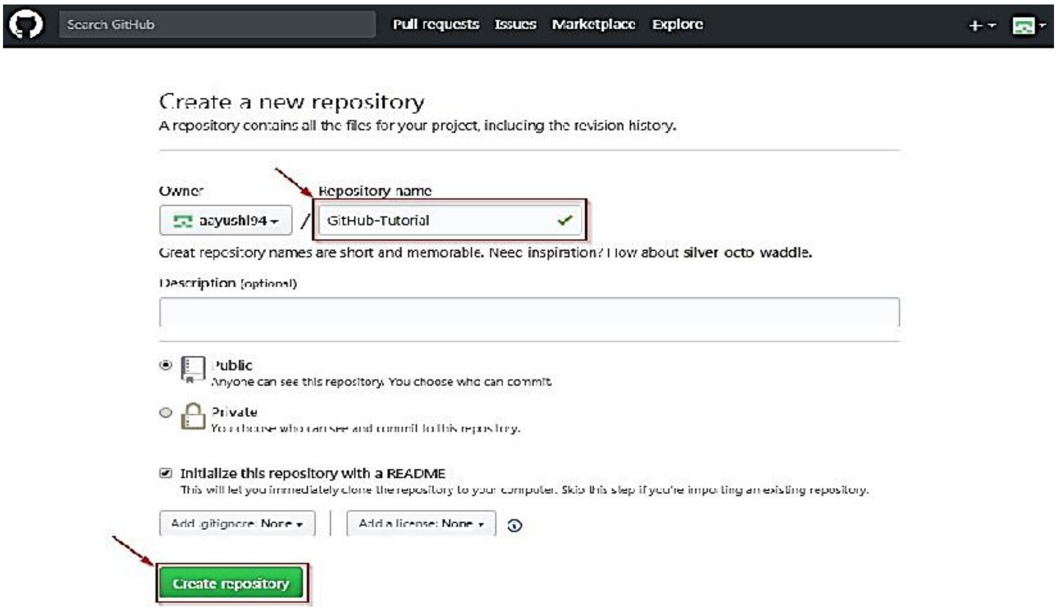
- Go to the link: https://github.com/ . Fill the sign up form and click on “Sign up for Github”.
- Click on “Start a new project”.


GitHub, Inc. [US]
https://github.com
Search GitHub
Pull requests
Issues
Marketplace
Explore
Learn Git and GitHub without any code!
Using the Hello World guide, you'll create a repository, start a branch, write comments, and open a pull request.
Read the guide
Start a project

- Refer to the below screenshot to get a better understanding.

Enter any repository name and click on “Create Repository”. You can also give a description to your

repository (optional).



Search GitHub | Pull requests | Issues | Marketplace | Explore

Create a new repository

A repository contains all the files for your project, including the revision history.

Owner: aayushi94 / Repository name: GitHub-Tutorial ✓

Great repository names are short and memorable. Need inspiration? How about silver octo waddle.

Description (optional)

● Public
Anyone can see this repository. You choose who can commit.

○ Private
You choose who can see and commit to this repository.

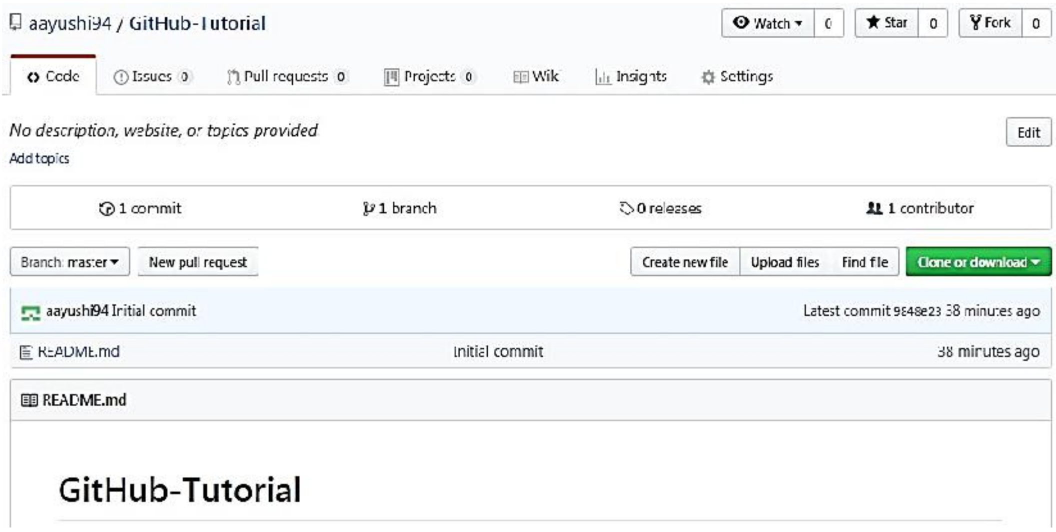
☑ Initialize this repository with a README
This will let you immediately clone the repository to your computer. Skip this step if you're importing an existing repository.

Add .gitignore: None ▾ | Add a license: None ▾

Create repository

Now, if you noticed by default a GitHub repository is public which means that anyone can view the contents of this repository whereas in a private repository, you can choose who can view the content.

Also, private repository is a paid version. Also, if you refer the above screenshot, initialize the repository with a README file. This file contains the description of the file and once you check this box, this will be the first file inside your repository



aayushi94 / GitHub-Tutorial

Watch ▾ 0 | ★ Star 0 | Fork 0

Code | Issues 0 | Pull requests 0 | Projects 0 | Wiki | Insights | Settings

No description, website, or topics provided | Edit

Add topics

1 commit | 1 branch | 0 releases | 1 contributor

Branch: master ▾ | New pull request | Create new file | Upload files | Find file | Clone or download ▾

aayushi94 Initial commit — Latest commit 9848e23 38 minutes ago

README.md — Initial commit — 38 minutes ago

README.md

GitHub-Tutorial

Summary

- Hypertext refers to the way in which Web pages (HTML documents) are linked together. Thus, the link available on a webpage is called Hypertext.
- HTML makes it possible to organize and format documents, similarly to Microsoft Word.
- Web technology is a standard that allows you to develop web applications with the help of pre- defined sets of classes, objects, methods, and properties available in a markup language, style sheet language, or programming language.
- A link relation is a relation between two files. HTML5 allow you to link your HTML document with another document or file.

- Elements are the building blocks of an HTML document. An element instructs the web browser how to display the HTML document on the user's screen.
- Root elements represent the main or the starting element that should be present in all HTML document.
- The DOM is a cross-platform and language-independent interface that allows programs and scripts to dynamically access and update the content, structure, and style of HTML or XML documents.

Keywords

Hypertext: Text (it often has embedded images, videos etc.) that is organized in order to link related items

Markup: A style guide for formatting whatsoever to be printed in hardcopy or soft copy format

HTML 5 is a updation of the Hypertext Markup Language (HTML), which is a typical programming language for displaying dynamic content on Web pages

DOCTYPEs It is the older version of HTML with longer DOCTYPEs

Class Classifies an element for use with Cascading Style Sheets

Data-XXXX Custom attributes. Authors of a HTML document can define their own attributes. Must start with "data".

NetBeans is an open-source IDE is used for developing software applications by developers.

GitHub To be very crisp about what exactly is GitHub, it is a file or code-sharing service to collaborate with different people

SelfAssessment

1. What are core components of Web applications?

A. UI (Front End (DOM, Framework))
B. Request Layer (Web API)
C. Back End (Database, Logic)
D. All of the above

2. makes every addressable item in a web application an Object that can be manipulated for color, transparency, position, sound and behaviors.

A. Document Object Model
B. Document adjust Model
C. Document Object
D. All of the above

3.In properties of DOM which object is at top of hierarchy:

A. Window Object
B. Document Object
C. Form Object
D. Link Object

4. when an HTML document is loaded into a window, it become a?

A. window Object
B. Document Object
C. Form Object
D. Link Object

5.which object is represented by link tags?

A. window Object
B. Document Object
C. Form Object
D. Link Object

6.What is full form of MVC?

A.Main View Controller

B.Mode View Controller

C.Model View Controller

D.None of the above

7.In MVC What handles user input and interactions?

A.Model

B.View

C.Control

D.All of the above

8.In MVC Where changes to the page are rendered and displayed

A.Model

B.View

C.Control

D.All of the above

9. is a programming concept where an ideal, or "virtual", representation of a UI is kept in memory and synced with the "real" DOM by a library such as REACTDOM.

A.The virtual DOM(VDOM)

B.Actual Dom

C.Dom

D.All of the above

10.ReactJS uses to increase performance?

A.The virtual DOM (VDOM

B.Actual Dom

C.Dom

D.All of the above

11. React renders HTML to the web page by using a function called

A. ReactDOM render ().
B. DOM render ().
C. ReactDOM().
D. All of the above

12. The ReactDOM.render() function takes arguments

A. HTML code
B. HTML element
C. Both of above
D. None of the above

13. JSX stand for

A. JavaScript XML
B. javascript
C. XML
D. None of a above

14. are independent and reusable bits of code.

A. Components
B. Properties
C. States
D. Classes

15. which component includes extends react. Component statement

A. Function

B. Class

C. Both of the above

D. none of the above

Answers for SelfAssessment

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

Review Questions:

1. What is the difference between HTML and HTML5?
2. What is <!DOCTYPE>? What are the different types of <!DOCTYPE> that are available?
3. What are the advantages of using HTML5?

4. How to create a link that will connect to another web browser page when clicked in HTML5? Explain with example.
5. What were some of the key goals and motivations for the HTML5 specification?
6. Can a <section> contain <article> elements? Can an <article> contain <section> elements? Provide usage examples

Further Readings

HTML 5 Black Book (Covers CSS3, JavaScript, XML, XHTML, AJAX, PHP, jQuery) 2Ed.

Web Enabled Commercial Application Development Using HTML, JavaScript, DHTML and PHP (4th Revised Edition)

Web Links

https://www.tutorialspoint.com/index.htm www.webopedia.com

www.web-source.net

Unit 02:Features of HTML5

CONTENTS

Objectives

After studying this unit, you will be able to:

- DiscusstheHistoryofHTML5
- How to use HTML Tag
- Describe the structure of HTML
- Explain Content Model
- Discuss SVG

Introduction

HTML5 is a markup language that was founded in 2004 by the Web Hypertext Application Technology WorkingGroup (WHATWG) whose members include Apple, Mozilla Foundation and Opera Software. Then in Oct 2006 W3C(World Wide Web Consortium) decided to stop their work on XHTML and start collaborating with "WHATWG" todevelopHTMLasatechnology.Afterthat,thefirstversionofHTML5waspublishedin2008thatwas writtenbyIanHickson,Google,ian@hixie.ch.Butit'snotcompletedandchangesarestillremainingbecauseaccordingtoWHATWGexperts "HTML5 is a continually evolving technology that will never end". After the first draft of HTML5, it's time tomakecompatiblebrowsersthatsupportHTML5features.AndthenMozillatookthefirststepandintroduced'Firefox3' which allows users to view HTML5 in the browser, but Safari, Google Chrome, and IE were far behind in supportof HTML5 features in their browsers. HTML5 was gaining popularity day by day. Then in April 2010, Steve Jobsdeclared that flash will never be allowed on Apple's smart devices. He said that "Flash was designed for PCs using amouse, not for touch screens using fingers". This statement is enough to change the mind of many companies andthat'sthereasontheybeganthedevelopmentofHTML5.

2.1 History of HTML5

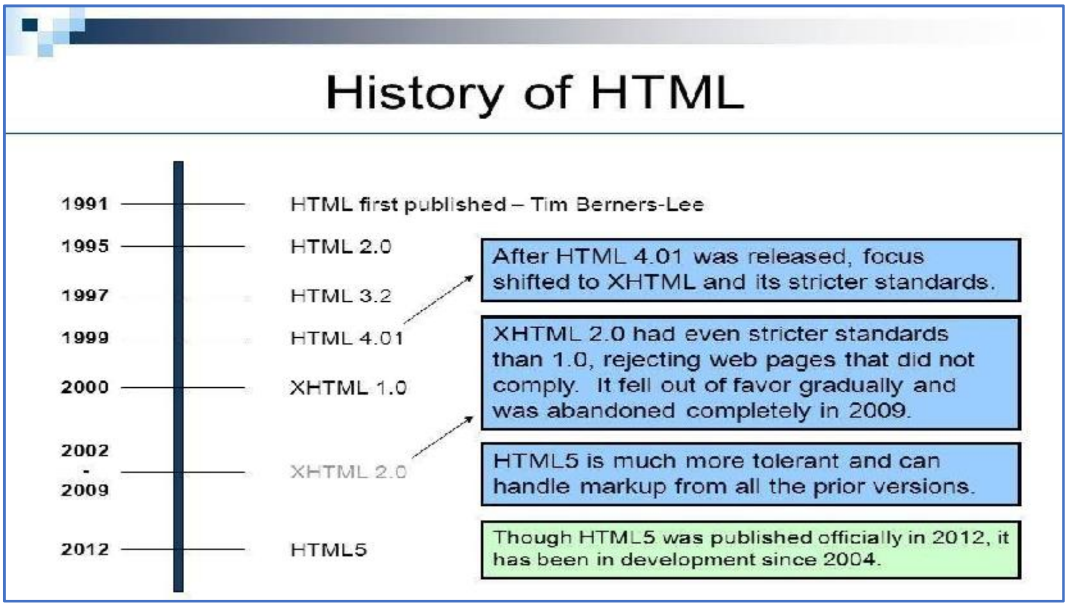
HTML1.0wasreleasedin1993withtheintentionofsharinginformationthatcanbereadableandaccessibleviaweb browsers.Butnotmanyofthedeveloperswereinvolvedincreatingwebsites.So,thelanguagewasalso notgrowing.

HTML 2.0, published in 1995, contains all the features of HTML 1.0 along with a few additional features, whichremainedthestandardmarkuplanguagefordesigningandcreatingwebsitesuntilJanuary1997andrefinedvariouscorefeaturesofHTML

HTML 3.0, where Dave Raggett introduced a fresh paper or draft on HTML. It included improved new features ofHTML,givingmorepowerfulcharacteristicsforwebmastersindesigningwebpages.ButthesepowerfulfeaturesofthenewHTML sloweddownthebrowserinapplyingfurtherimprovements.

HTML 4.01, which is widely used and was a successful version of HTML before HTML 5.0, which is currentlyreleased and used worldwide. HTML 5 can be said for an extended version of HTML 4.01, which was published intheyear2012.

HTMLwascreatedbySirTimBerners-Leeinlate1991butwasnotreleasedofficially,publishedin1995asHTML 2.0.HTML4.01waspublishedinlate1999andwasamajorversionofHTML.



History of HTML

1991 HTML first published – Tim Berners-Lee

1995 HTML 2.0

1997 HTML 3.2

1999 HTML 4.01

2000 XHTML 1.0

2002 - 2009 XHTML 2.0

2012 HTML5

After HTML 4.01 was released, focus shifted to XHTML and its stricter standards.

XHTML 2.0 had even stricter standards than 1.0, rejecting web pages that did not comply. It fell out of favor gradually and was abandoned completely in 2009.

HTML5 is much more tolerant and can handle markup from all the prior versions.

Though HTML5 was published officially in 2012, it has been in development since 2004.

2.2 Exploring New Features of HTML 5

1) Ithasintroducednewmultimediafeatureswhichsupportaudioandvideocontrolsbyusing <audio>and<video>tags.
2) Therearenewgraphicselementsincludingvectorgraphicsandtags.
3) Enrichsemanticcontentbyincluding<header><footer>,<article>,<section>and<figure> areadded.
4) DragandDrop-Theusercangrabanobjectanddragit furtherdroppingittoanewlocation.
5) Geo-locationservices-Ithelpstolocatethegeographicallocationofaclient.
6) Webstoragefacilitywhichprovideswebapplicationmethodstostoredataona webbrowser.
7) UsesSQLdatabasetostoredataoffline.
8) Allowsdrawingvariousshapesliketriangles,rectangles,circles,etc.

9) Capableof handlingincorrectsyntax.
10) EasyDOCTYPEdeclarationi.e.,<!doctypehtml>
11) Easycharacterencodingi.e.,<metacharset=″UTF-8″>

2.1.1 Difference Between HTML and HTML5

Features	HTML	HTML5
Definition	A hypertextmarkuplanguage(HTML)is theprimarylanguagefordevelopingwebpages.	HTML5isanewversion of HTMLwithnew functionalities with markuplanguagewithInternettechnologies
Multimedia support	LanguageinHTMLdoesnothavesupportfor videoandaudio.	HTML5supports bothvideoandaudio.
Storage	TheHTMLbrowserusescachememoryastemporarystorage.	HTML5hasthestorageoptionslikeapplicationcache,SQL database,andwebstorage.
Browsercompatibility	HTML is compatible with almost all browsersbecause it has been present for a long time, andthebrowserhasmade modificationstosupportallthefeatures.	InHTML5,wehavemanynewtags,elements, andsometagsthathave beenremoved/modified,soonlysomebrowsersarefullycompatible withHTML5.
Graphics support	InHTML,vectorgraphicsarepossiblewithtoolsLikeSilverlight,Adobe Flash, VML,etc.	InHTML5,vectorgraphicsaresupported bydefault.
Threading	In HTML, the browser interface andJavaScriptruninthesamethread.	The HTML5 has the JavaScript WebWorkerAPI,whichallowsthe browserinterfacetoruninmultiplethreads.
Doctype	Doctypedeclarationinhtmlistoolong <!DOCTYPEHTMLPUBLIC"-//W3C //DTDHTML4.01//EN" "http://www.w3.org/TR/html4/strict.dtd">	The DOCTYPEdeclarationinhtml5isverysimple"<!DOCTYPEhtml>
CharacterEncoding	Character encodinginHTMListoolong. <!DOCTYPEHTMLPUBLIC"-//W3C// DTDHTML4.0Transitional//EN">	Characterencodingdeclarationissimple<metacharset="UTF-8">
Multimedia support	Audio and video are not the part ofHTML4.	Audioandvideoareessentialpartsof HTML5,like:<Audio>,<Video>.

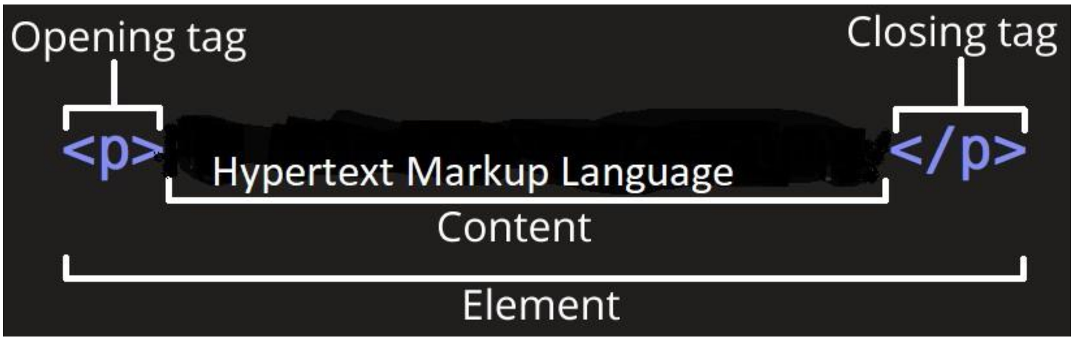
DidyouKnow? : HTML5 provides you with more features with respect to the HTML such as audio, video with the helpoftags,dragand dropfeature,Geolocation,browsersupport,etc.

2.3 AnatomyofanHTMLTag

HTMLconsistsof aseriesofelements, which you use toenclose,or wrap,differentparts ofthe contenttomake itappear acertain way, or act acertain way. The enclosingtagscan make aword or image hyperlink to somewhere elsecanitalicizewords,canmakethefontbiggerorsmaller,andsoon.Forexample,**HypertextMarkupLanguage**

Ifwewantedthelinetostandbyitself,wecouldspecifythatit isaparagraphbyenclosingit inparagraphtags:

<p>HypertextMarkupLanguage</p>



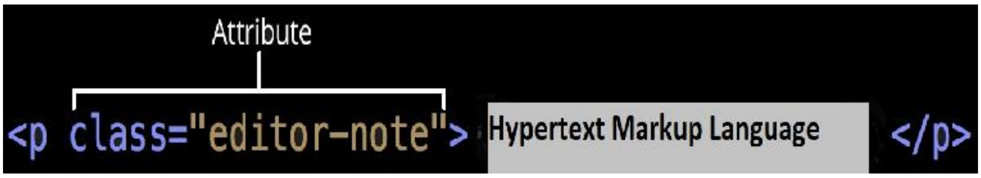
Theopeningtag: Thisconsistsof the nameof the element(in this case, p), wrapped inopening andclosing **angle brackets**. This states where the element begins or starts to take effect in this case where theparagraphbegins.

The closing tag: This is the same as the opening tag, except that it includes a *forward slash* before the elementname.Thisstateswheretheelementends—inthiscasewheretheparagraphends.Failingtoaddaclosingtag isoneof thestandardbeginnererrors andcanlead to strange results.

Thecontent: Thisisthecontentoftheelement,whichinthiscase,isjusttext.

Theelement: Theopeningtag,theclosingtag,andthecontenttogethercomprisetheelement.

Elementscanalsohaveattributes thatlooklikethe following:



Attributescontainextrainformationabouttheelementthatyoudon'twanttoappear intheactual content.Here, class istheattribute *name* and editor-note aretheattribute *value*.The class attributeallowsyoutogivetheelement a non-unique identifier that can be used to target it (and any other elements with the same class value) withstyleinformationandotherthings.

An attributeshouldalwayshave the following:

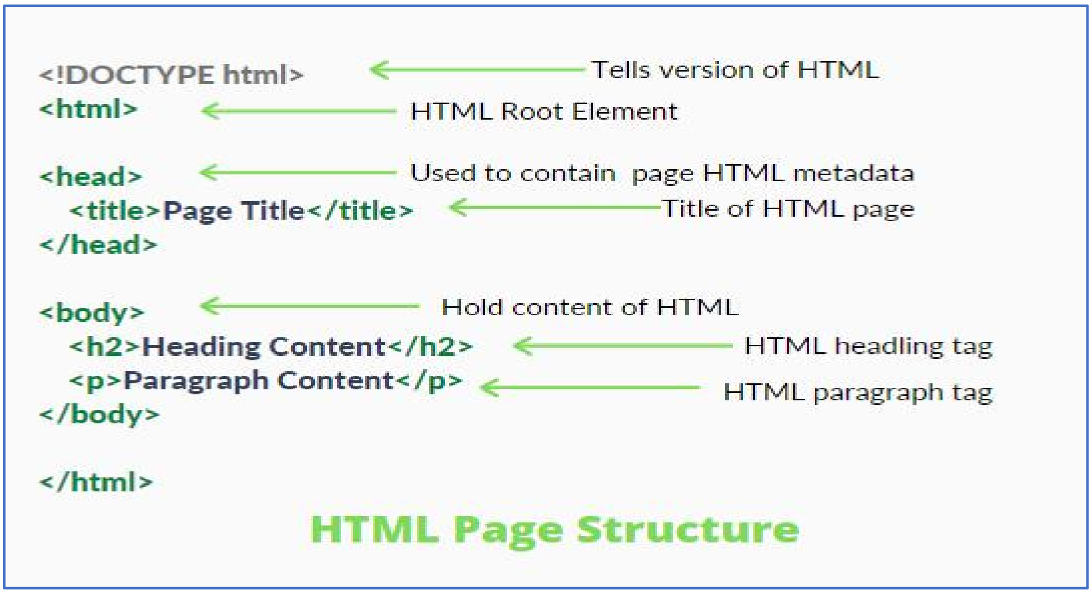
1. A space between it and the element name (or the previous attribute, if the element already has one or moreattributes).
2. Theattributenameisfollowedbyanequalsign.
3. Theattributevalueiswrappedbyopeningandclosingquotationmarks.

Note: Simpleattributevaluesthatdon'tcontainASCIIwhitespace.

2.4 HTMLDocumentStructure

The **<HTML>**is a markup language that is used by the browser to manipulate text, images, and other content todisplayitintherequiredformat.
Tags in HTML: Tags are one of the most importantparts ofan HTML Document. HTML uses some predefinedtagswhichtellthebrowseraboutcontentdisplayproperty,thatishowtodisplayaparticularly givencontent

In itssimplestform,thefollowingisanHTMLdocument:



```
<!DOCTYPE html>
<html>

<head>
  <title>Page Title</title>
</head>

<body>
  <h2>Heading Content</h2>
  <p>Paragraph Content</p>
</body>

</html>
```

HTML Page Structure

TheDOCTYPE

- TheDOCTYPEtellsthewebbrowserwhichversionofHTMLthepageiswrittenin.Inthisclass,wewillbeusing'XHTMLTransitional',whichallowsusalittleflexibility.

The<html>Element

The<html>element tellsthebrowserthat the pagewill be formatted in HTML and, optionally,whichworldlanguagethepagecontentisin.

The<head>and<body>Elements

- The<head>elementsurroundsallthespecial"behindthescenes"elementsofawebdocument.Mostoftheseelementsdonotgetdisplayeddirectlyonthewebpage.
- The <body> element surrounds all the actual content (text, images, videos, links, etc.) thatwill bedisplayedonourwebpage.

The<meta>Element

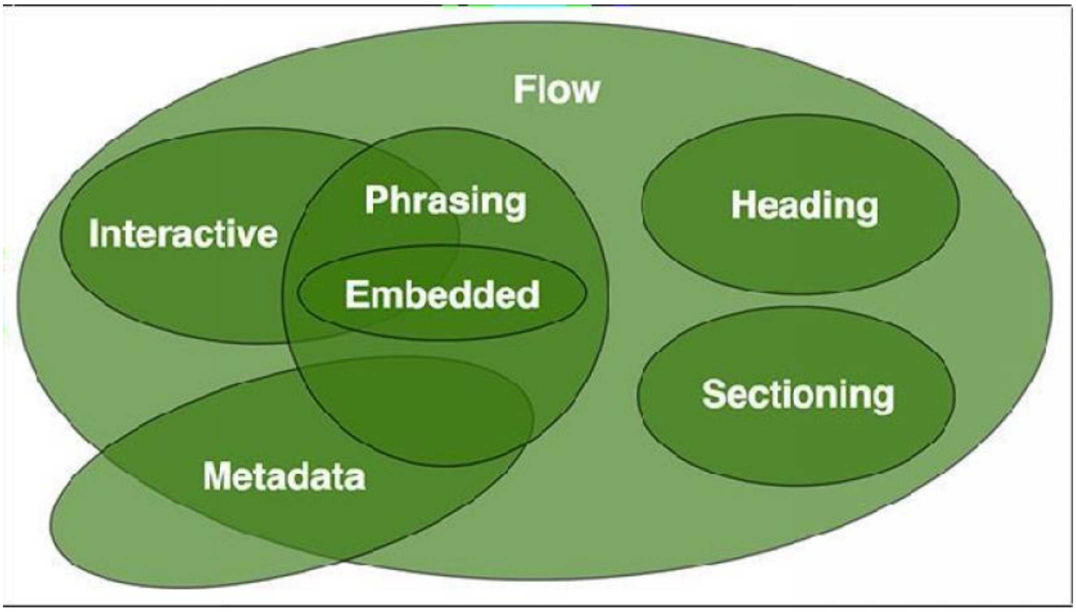
- Immediatelyafterthe<head>line,weplacethis<meta>element:
- ThislinedeclaresthatthedocumentisencodedintheUTF-8(Unicode)characterset.
- Therecanbemultiple<meta>linesonthesamewebpage.The<meta>elementisoftenusedtoprovideadditionalinformationsuchaspagekeywords,apagedescription,andtheauthor(s)ofawebdocument.

The<title>Element

- The<title>elementdefineswhattextwillshowinthewebbrowser'stitlebar:

2.5 HTML ContentModel

The contentModelreferstothesetofrulesthatdefinewhattypeofcontenteachelementisallowedto have.Mostly,thistranslates intowhatotherelementsareallowedtobenestedinsidewhich other elements.PriortothemodernHTMLspecification,HTMLelementswereeitherblock-levelorinlineelements.



ModernHTMLspecificationssplitthesetwocontentmodelsintosevenmodels.

Metadata: Metadatacontentisresponsibleforsettingupthepresentation(look)orbehaviortothe restoftheHTMLpage.ItcanalsosetuptherelationshipoftheHTMLdocumentwithother documents.

```
<!doctypehtml>
<htmllang="en">
<head>
<metaname="application-name"content="HTML5forbeginners">
</head>
<body>
</body>
</html>
```

Flowcontent: Most contents of HTML documents arein this type. These contents influence other contentsto flow. Sectioning content represents a section in the current document. Each sectioning content potentiallyhasaheadingcontentand footer.

Heading Content: Heading Content is used to provide different heading levels in HTML documents.Thesecontentsareusedtocreateheadlinesforatext.

Forexample,ifyouwanttodisplayanarticleonyourHTMLpage,thearticleiswritteninnormalpl aintext,but itstitle is written in bold and large text. In this case, you can use the heading elements to make the title distinct fromthe remainingtext.Aheadingcontentisdefinedas-h1,h2,h3,h4,h5,h6,group.

Phrasing Content: Phrasing content is the text that you see in the document and in the HTML elementsthat markup texts at the intra-paragraph level. Runs of phrasing content make up paragraphs. Phrasingcontentreferstothosesmallpiecesoftextthataresurroundedbyothertexts.For example,links.Alinkis often surrounded by texts. The element that contains phrasing content should contain either text orembeddedcontent.Elements thatcontainthistypeofcontentareinline-levelandmusthaveanendtag.

Embedded content: Embedded content embeds resources from other sources or adds content from othermark-uplanguages.Forexample,imagevideos,etc.

TheHTMLelementsthatcancontainembeddedcontentare-

audio,canvas,embed,iframe,img,math,object,svg,video

InteractiveContent:Thecontents on the webpage that can interact withusers are interactivecontent.Forexample,links,buttons,etc.Interactivecontentsareseeninsidethefor m.

2.6 HTMLCharacterEntity

- HTMLsymbolslikemathematicaloperators,arrows,technicalsymbols,andshapes,arenotp resentonanormal keyboard.

- ReservedcharactersinHTMLmustbereplacedwithcharacterentities.
- SomecharactersarereservedinHTML.
- Ifyouusethelessthan(<)orgreaterthan(>)signsinyourtext,thebrowsermightmixthemwithtags .

Characterentitiesareusedtodisplayreservedcharactersin HTML.0.character entitylookslikethis:

&entity_name;

OR

&#entity_number;

Todisplayalessthansign(<)wemustwrite:&lt;or&#60

Advantageofusinganentityname: Anentitynameiseasytoremember.

Adisadvantageofusinganentityname: Browsersmaynotsupportallentitynames,butthesupportforenti tynumbersisgood.

2.7 HMTL LINKS

Links are found in nearly all web pages. Links allow users to click their way from page to page.HTML links are hyperlinks.You can click on a link and jump to another document.When you move the mouse over a link, the mouse arrow will turn into a little hand.

The HTML <a> tag defines a hyperlink. It has the following syntax:

<a href="url">link text</a>

The most important attribute of the <a> element is the href attribute, which indicates the link's destination.The link text is the part that will be visible to the reader.Clicking on the link text, will send the reader to the specified URL address.

This example shows how to create a link to W3Schools.com.

<a href="https://www.w3schools.com/">Visit W3Schools.com!</a>

By default, the linked page will be displayed in the current browser window. To change this, you must specify another target for the link. The target attribute specifies where to open the linked document.

The target attribute can have one of the following values:

_self - Default. Opens the document in the same window/tab as it was clicked

_blank - Opens the document in a new window or tab

_parent - Opens the document in the parent frame

_top - Opens the document in the full body of the window

2.8 Canvas

The HTML5 canvas element can be used to draw graphics on the webpage via JavaScript. The canvas was originally introduced by Apple for the Mac OS dashboard widgets and to power graphics in the Safari web browser. Later it was adopted by Firefox, Google Chrome, and Opera. Now the canvas is a part of the new HTML5 specification for next-generation web technologies. By default, the <canvas> element has 300px of width and 150px of height without any border and content. However, custom width and height can be defined using the CSS height and width property whereas the border can be applied using the CSS border property.

Understanding Canvas Coordinates

The canvas is a two-dimensional rectangular area. The coordinates of the top-left corner of the canvas are (0, 0) which is known as origin, and the coordinates of the bottom-right corner are (canvas width, canvas height). Here's a simple demonstration of canvas default coordinate system.

HTML5 Canvas

The HTML5 Canvas is an Immediate Mode bit-mapped area of the screen that can be manipulated with JavaScript and CSS. The *HTML5 Canvas* is an *Immediate Mode* bit-mapped area of the screen that can be manipulated with JavaScript and CSS.

Immediate Mode

Immediate Mode refers to the way the canvas renders pixels on the screen. The HTML5 Canvas completely redraws the bitmapped screen on every frame using Canvas API calls from JavaScript. As a programmer, your job is to set-up the screen display before each frame is rendered.

Flash, Silverlight, and DOM <div> manipulation techniques use Retained Mode. In Retained Mode a list of individual display objects is stored and manipulated.

HTML5 Canvas Properties

Canvas Has Three Properties:

- width
- height
- id

Width and height read/write which means you can resize the Canvas on the fly

HTML5 Canvas Methods

getContext() : You need the context to draw anything on the Canvas.

toDataUrl() : Outputs the bitmapped data of the Canvas to a string (can be used to create a screenshot)

CSS can be used in conjunction with Canvas object itself. However, individual drawings on the Canvas cannot be manipulated with CSS

Example: you can scale the Canvas using CSS

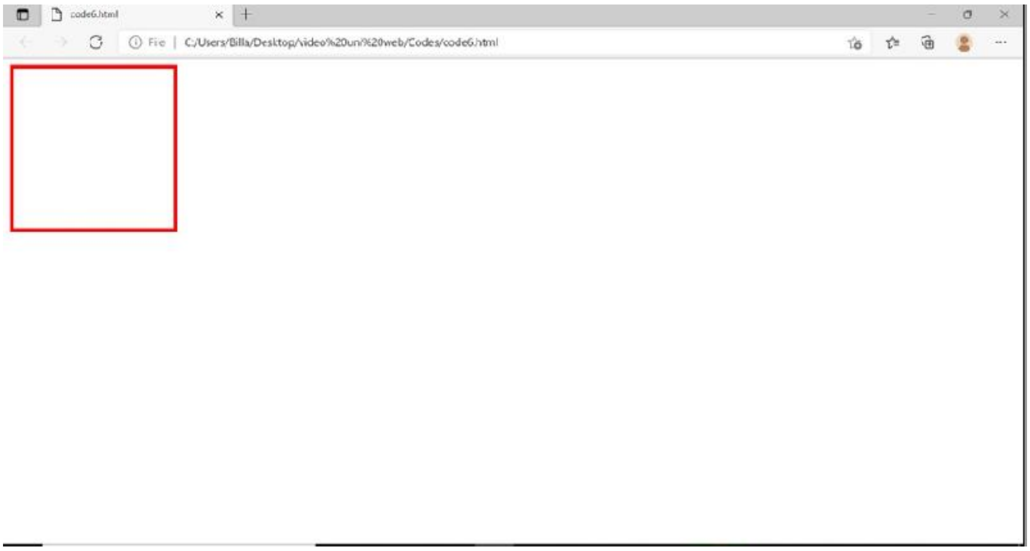
style="width: 400px; height:400px"

Does not resize but instead scales (like setting width ad height for a Flash embed)

Example:

```
<!DOCTYPE html>
<html>
<body>
<canvas id = "Ani" height = "200" width = "200"
        style = "border:5px solid red">
</canvas>
</body>
</html>
```

Output:



2.9 SVG

SVG stands for Scalable Vector Graphics. SVG is used to define graphics for the Web.SVG is a W3C recommendation

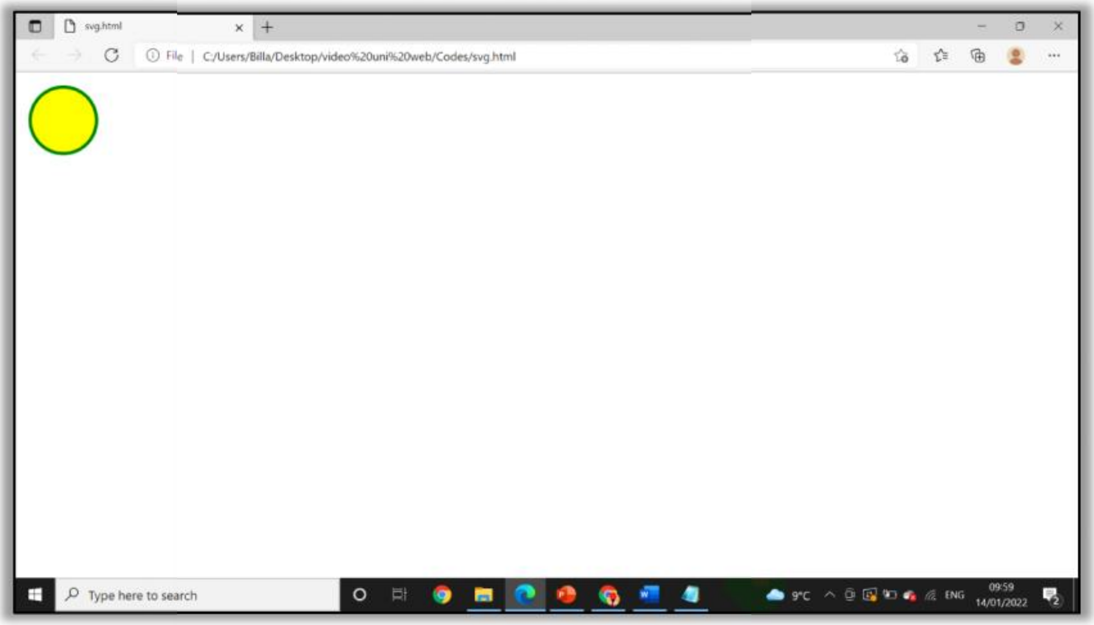
The HTML <svg> Element

The HTML <svg> element is a container for SVG graphics.SVG has several methods for drawing paths, boxes, circles, text, and graphic images.

Example:

```
<!DOCTYPE html>
<html>
<body>
<svg width="100" height="100">
<circle cx="50" cy="50" r="40" stroke="green" stroke-width="4" fill="yellow" />
</svg>
```

```
</body>
</html>
```


svg.html
File | C:/Users/Billa/Desktop/video%20uni%20web/Codes/svg.html
Type here to search
9°C
ENG
09:59
14/01/2022

Summary

- HTML is a markup language that is used for creating attractive web pages with the help of styling, and which looks in a nice format on a web browser.
- HTML tags are like keywords define that how a web browser will format and display the content.
- If You have used an open tag <tag>, then you must use a close tag </tag>.
- HTML attributes are special words that provide additional information about the elements or attributes that are the modifier of the HTML element.
- Each element or tag can have attributes, which define the behavior of that element.
- <canvas> gives you an easy and powerful way to draw graphics using JavaScript. It can be used to draw graphs, make photo compositions, or do simple (and not so simple) animations.
- SVG stands for Scalable Vector Graphics and it is a language for describing 2D graphics and graphical applications in XML and the XML is then rendered by an SVG viewer.
- SVG is mostly useful for vector type diagrams like Pie charts, Two-dimensional graphs in an X, Y coordinate system, etc.

Keywords

- **DOCTYPE:**tellsthewebbrowserwhichversionofHTMLthepageiswrittenin.Inthis class,wewillbeusing‘XHTMLTransitional’,whichallowsusalittleflexibility.
- The<head>elementsurroundsallthespecial“behindthescenes”elementsofawebdocument.Mostoftheseelementsdonotgetdisplayeddirectlyonthewebpage.
- The <body> element surrounds all the actual content (text, images, videos, links, etc.) thatwill bedisplayedonourwebpage.
- e0⁼is displayed on the browser in bold format and the size of the text depends on the number of headings.⁼

- Yp> **P Tag defines a paragraph=**

Self Assessment

1. Which of the following tag is used to define options in a drop-down selection list?
A. <select>
B. <list>
C. <dropdown>
D. <option>

2. HTML tags are enclosed in-
A. # and #
B. { and }
C. ! and ?
D. < and >
3. Which of the following tag is used to add rows in the table?

A. <td> and </td>

B. <th> and </th>

C. <tr> and </tr>

D. None of the above

4. The <hr> tag in HTML is used for -

A. new line

B. vertical ruler

C. new paragraph

D. horizontal ruler

5. Which of the following attribute is used to provide a unique name to an element?

A. class

B. id

C. type

D. None of the above

6. Which of the following HTML tag is used to display the text with scrolling effect?

A. <marquee>

B. <scroll>

C. <div>

D. None of the above

7. Which of the following HTML tag is the special formatting tag?

A. <p>

B. <b>

C. <pre>

D. None of the above

8. How to insert a background image in HTML?

A. <body background = "img.png">

B. <img background = "img.png">

C. <bg-image = "img.png">

D. None of the above

9. Which of the following is the correct way to create a list using the lowercase letters?
A. <ol alpha = "a" >

B. <ol type = "a">
C. <ol letter = "a">
D. None of the above

10. Which of the following HTML attribute is used to define inline styles?

A. style

B. type

C. class

D. None of the above

11. Which of the following is the paragraph tag in HTML?

A. <p>

B. <b>

C. <pre>

D. None of the above

12. An HTML program is saved by using the ____ extension.
A. .ht

B. .html

C. .hml

D. None of the above

13. A program in HTML can be rendered and read by –
A. Web browser
B. Server
C. Interpreter
D. None of the above

14. What are the types of unordered or bulleted list in HTML?
A. disc, square, triangle
B. polygon, triangle, circle
C. disc, circle, square
D. All of the above

15. Which of the following is the correct way to create a list using the lowercase letters?

A. <ol alpha = "a" >

B. <ol type = "a">

C. <ol letter = "a">

D. None of the above

Answers for Self Assessment

1.	D	2.	D	3.	C	4.	D	5.	B
6.	A	7.	C	8.	A	9.	B	10.	A

11.	A	12.	B	13.	A	14.	C	15.	B

Review Questions

1. What are some of the key new features in HTML5?
2. What are the different new form element types in HTML 5?
3. Explain the layout of HTML?
4. What were some of the key goals and motivations for the HTML5 specification?
5. How to create a hyperlink in HTML
6. What is the canvas element in HTML5?
7. What’s the difference between the <svg> and <canvas> elements?
8. Give a simple implementation of the <video> tag to embed a videostored at http://www.example.com/amazing_video.mp4. Give the video width of 640 pixels by 360 pixels.

Further Readings

- HTML5 Black Book (Covers CSS3, JavaScript, XML, XHTML, AJAX, PHP, j Query) 2Ed.
- Web Enabled Commercial Application Development Using HTML, Java Script, DHTML and PHP (4th Revised Edition)
- Trevor Burnham. *Async JavaScript*. The Pragmatic Bookshelf, Raleigh, NC and Dallas, TX, 2012.

Unit 03: Introduction to CSS3

CONTENTS

Objectives

After studying this unit, you will be able to:

- Introduction to CSS3
- Discuss the selectors present in the syntax of a CSS file
- How to work CSS3
- Learn about how to insert CSS ai an HTML document.

Introduction

CSS is used to control the style of a web document in a simple and easy way.

CSS is the acronym for "Cascading Style Sheet".

Cascading Style Sheets fondly referred to as CSS, is a simple design language intended to simplify the process of making web pages presentable.

CSS is a MUST for students and working professionals to become great Software Engineers especially when they are working in Web Development Domain.

3.1 Benefits of CSS3

CSS or cascading sheet is a text-based coding language that specifies the website formats and the way of communicating with web browsers. The language allows web developers to regulate various style elements and functionalities, like layout, color, fonts, and therefore the formatting and display of HTML documents.

The main goal (as a method sheet language) was to separate document content from document presentation, which incorporates style elements, like color, layout, and fonts. CSS handles the design and feel a part of an internet page. Using CSS, you will control the color of the text, the design of fonts, the spacing between paragraphs, how columns are sized and laid out, etc.

- CSS saves time − You can write CSS once and then reuse the same sheet in multiple HTML pages. You can define a style for each HTML element and apply it to as many Web pages as you want.
- Easy maintenance − To make a global change, simply change the style, and all elements in all the web pages will be updated automatically.
- Global web standards − Now HTML attributes are being deprecated and it is being recommended to use CSS. So, it's a good idea to start using CSS in all the HTML pages to make them compatible with future browsers.
- Platform Independence − The Script offers consistent platform independence and can support the latest browsers as well.

Whylearn CSS?

Create a Stunning Website - CSS handles the look and feel part of a web page. Using CSS, you can control the color of the text, the style of fonts, the spacing between paragraphs, how columns are sized and laid out, what background images or colors are used, layout designs, variations in display for different devices and screen sizes as well as a variety of other effects.

Become a web designer - If you want to start a career as a professional web designer, HTML and CSS designing is a must skill.

Control web - CSS is easy to learn and understand but it provides powerful control over the presentation of an HTML document. Most commonly, CSS is combined with the markup languages HTML or XHTML.

Learn other languages - Once you understand the basics of HTML and CSS then other related technologies like JavaScript, PHP, or angular have become easier to understand.

Applications of CSS

CSS saves time - You can write CSS once and then reuse the same sheet in multiple HTML pages. You can define a style for each HTML element and apply it to as many Web pages as you want.

Pages load faster - If you are using CSS, you do not need to write HTML tag attributes every time. Just write one CSS rule of a tag and apply it to all the occurrences of that tag. So less code means faster download times.

Easy maintenance - To make a global change, simply change the style, and all elements in all the web pages will be updated automatically.

Superior styles to HTML - CSS has a much wider array of attributes than HTML, so you can give a far better look to your HTML page in comparison to HTML attributes.

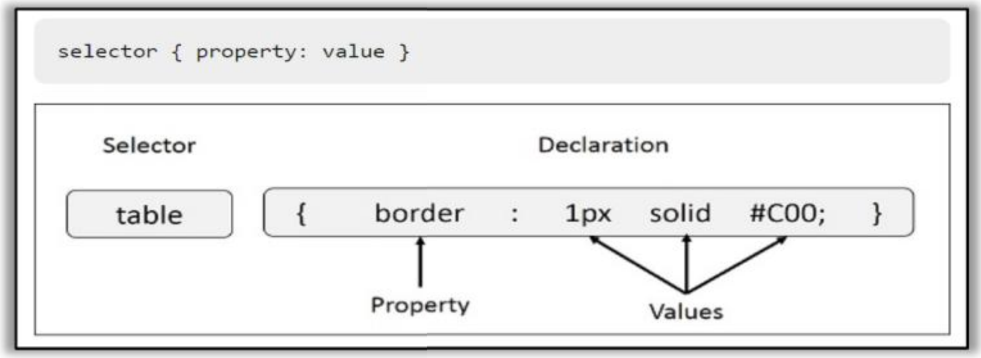
Multiple Device Compatibility - Style sheets allow content to be optimized for more than one type of device. By using the same HTML document, different versions of a website can be presented for handheld devices such as PDAs and cell phones or for printing.

Global web standards - Now HTML attributes are being deprecated and it is being recommended to use CSS. So, it's a good idea to start using CSS in all the HTML pages to make them compatible with future browsers.

3.2 Understanding the syntax of CSS

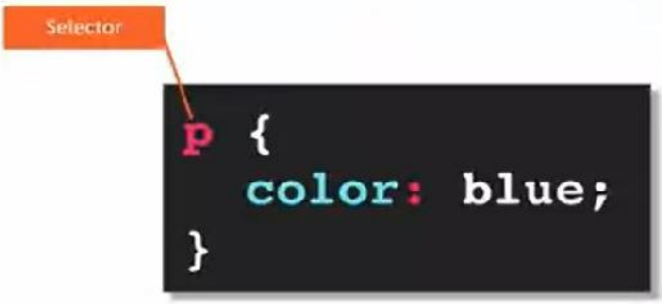
The syntax can be defined as a rule that defines the structure or the order of the statements used in a programming language. It also specifies how words and symbols are put together to form statements and expressions.CSS also uses syntax to apply CSS rules in an HTML document.

A CSS comprises style rules that are interpreted by the browser and then applied to the corresponding elements in your document. A style rule is made of three parts- Selector, property, and Value.



```
selector { property: value }
```

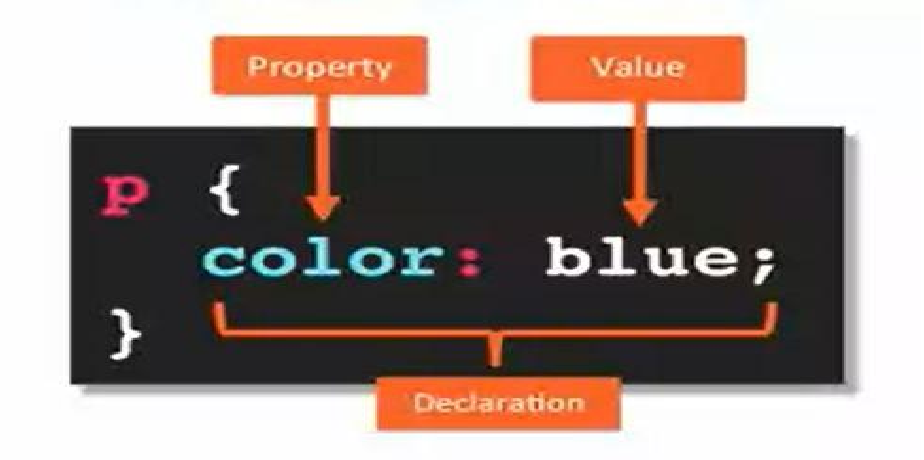
Selector − A selector is an HTML tag at which a style will be applied. This could be any tag like <h1> or <table> etc.



```
p {
  color: blue;
}
```

Property − A property is a type of attribute of an HTML tag. Put simply, all the HTML attributes are converted into CSS properties. They could be color, border etc.

Value − Values are assigned to properties. For example, color property can have value either red or #F1F1F1 etc.



```
p {
  color: blue;
}
```

Example,

- You can define a table border as follows −
- table{ border :1px solid #C00; }
- Here table is a selector and border is a property and given value 1px solid #C00 is the value of that property.

3.3 Define selectors in various simple ways

The following code to understand the selectors.

The Type Selectors

```
h1 {
  color: #36CFFF;
}
```

The Universal Selectors

Rather than selecting elements of a specific type, the universal selector quite simply matches the name of any element type −

```
* {
  color: #000000;
}
```

This rule renders the content of every element in our document in black.

The Descendant Selectors

Suppose you want to apply a style rule to a particular element only when it lies inside a particular element. As given in the following example, the style rule will apply to the <em> element only when it lies inside <ul> tag.

```
ulem {
  color: #000000;
}
```

The Class Selectors

You can define style rules based on the class attribute of the elements. All the elements having that class will be formatted according to the defined rule.

```
.black {
  color: #000000;
}
```

This rule renders the content in black for every element with a class attribute set to black in our document.

Example

You can make it a bit more particular. For example −

```
h1.black {
  color: #000000;
}
```

This rule renders the content in black for only <h1> elements with class attribute set to black.

You an apply more than one class selector to given element. Consider the following example −

```
<p class = "center bold">
  This para will be styled by the classes center and bold.
</p>
```

3.4 Inserting CSS in an HTML Document

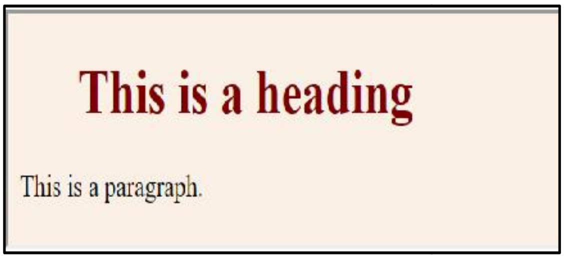
There are four ways to associate styles with your HTML document.

The most commonly used methods are

- Inline CSS and
- External CSS.
  - Embedded CSS - The <style> Element

You can put your CSS rules into an HTML document using the <style> element. This tag is placed inside the <head>...</head> tags. Rules defined using this syntax will be applied to all the elements available in the document.

```
<!DOCTYPE html>
<html><head>
<style type = "text/css" media = "all">
      body {
        background-color: linen;
      }
      h1 { color: maroon;
        margin-left: 40px;  }
</style>
</head>
<body>
<h1>This is a heading</h1>
<p>This is a paragraph.</p>
</body>
</html>
```



This is a heading

This is a paragraph.

Attributes

Attributes associated with <style> elements are –

Attribute	Value	Description
type	text/css	Specifies the style sheet language as a content-type (MIME type). This is required attribute.
media	Screentvprojection Handheldprint braille auralall	Specifies the device the document will be displayed on. The default value is *all*.

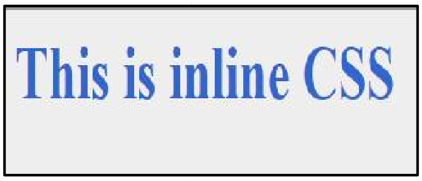
Inline CSS - The style Attribute

- You can use the style attribute of any HTML element to define style rules. These rules will be applied to that element only. Here is the generic syntax −
- <element style = "...style rules....">

Attribute	Value	Description
style	style rules	The value of the *style* attribute is a combination of style declarations separated by semicolons (;).

Example

```
<html>
<head>
</head>
<body>
<h1 style = "color:#36C;">
This is inline CSS
</h1>
</body>
</html>
```


This is inline CSS

External CSS - The <link> Element

- The <link> element can be used to include an external stylesheet file in your HTML document.
- An external style sheet is a separate text file with .CSS extension. You define all the Style rules within this text file and then you can include this file in any HTML document using <link> element.
- Here is the generic syntax of including external CSS file −
- <head>
- <link type = "text/css" href = "..." media = "..." />
- </head>

Attributes

Attribute	Value	Description
type	text CSS	Specifies the style sheet language as a content-type (MIME type). This attribute is required.
href	URL	Specifies the style sheet file having Style rules. This attribute is required.

media	Screentvprojection Handheldprint Brailleauralall	Specifies the device the document will be displayed on. The default value is *all*. This is an optional attribute

Example,

Consider a simple style sheet file with a name mystyle.css having the following rules −

```
h1, h2, h3 {
 color: #36C;
 font-weight: normal;
 letter-spacing: .4em;
margin-bottom: 1em;
text-transform: lowercase;
}
```

Now you can include this file mystyle.css in any HTML document as follows −

```
<head>
<link type = "text/css" href = "mystyle.css" media = " all" />
</head>
```

Imported CSS - @import Rule

- @Import is used to import an external stylesheet in a manner similar to the <link> element. Here is the generic syntax of @import rule.
- <head>
- @import "URL";
- </head>
- Here URL is the URL of the style sheet file having style rules. You can use another syntax as well −
  - <head>
  - @importurl("URL");

</head>

CSS Colors

Colors in CSS can be specified by the following methods:

Hexadecimal colors

Hexadecimal colors with transparency

RGB colors

RGBA colors

HSL colors

HSLA colors

Predefined/Cross-browser color names

With the current color keyword

- **Hexadecimal Colors**

A hexadecimal color is specified with: #RRGGBB, where the RR (red), GG (green) and BB (blue) hexadecimal integers specify the components of the color. All values must be between 00 and FF.

For example, the #0000ff value is rendered as blue, because the blue component is set to its highest value (ff) and the others are set to 00.

Example,

different HEX colors:

```
#p1 {background-color: #ff0000;}   /* red */
#p2 {background-color: #00ff00;}   /* green */
#p3 {background-color: #0000ff;}   /* blue */
```

- RGB Colors

An RGB color value is specified with the rgb() function, which has the following syntax:

rgb(red, green, blue)

Each parameter (red, green, and blue) defines the intensity of the color and can be an integer between 0 and 255 or a percentage value (from 0% to 100%).

- For example, the rgb(0,0,255) value is rendered as blue, because the blue parameter is set to its highest value (255) and the others are set to 0.
- Also, the following values define equal color: rgb(0,0,255) and rgb(0%,0%,100%)
- **RGBA Colors**

RGBA color values are an extension of RGB color values with an alpha channel - which specifies the opacity of the object.

An RGBA color is specified with the rgba() function, which has the following syntax:

rgba(red, green, blue, alpha)

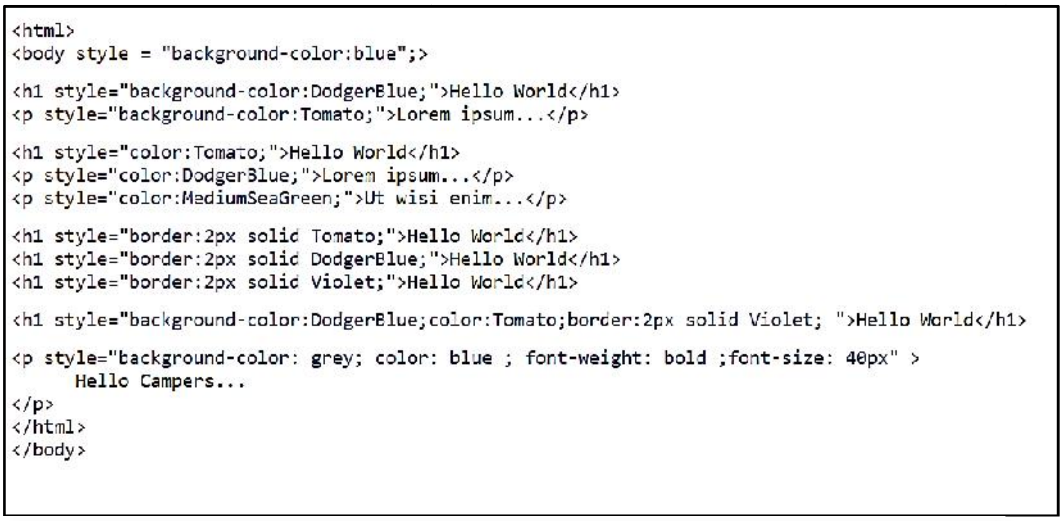
- HSL Colors

HSL stands for hue, saturation, and lightness - and represents a cylindrical-coordinate representation of colors.

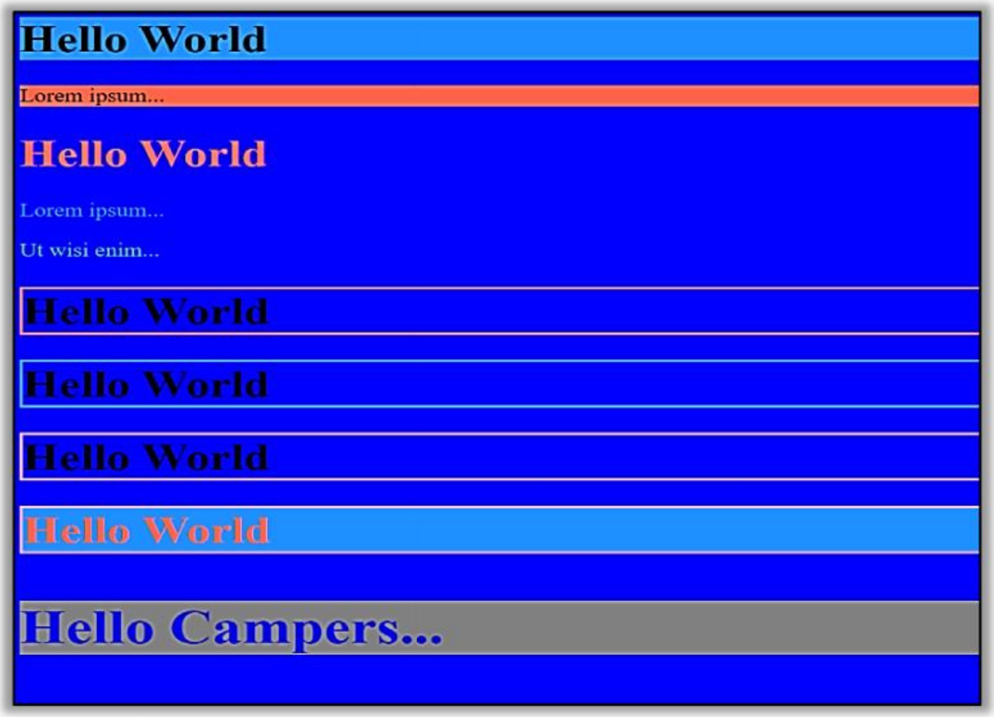
An HSL color value is specified with the hsl() function, which has the following syntax:

hsl(hue, saturation, lightness)

Example



```
<html>
<body style = "background-color:blue";>

<h1 style="background-color:DodgerBlue;">Hello World</h1>
<p style="background-color:Tomato;">Lorem ipsum...</p>

<h1 style="color:Tomato;">Hello World</h1>
<p style="color:DodgerBlue;">Lorem ipsum...</p>
<p style="color:MediumSeaGreen;">Ut wisi enim...</p>

<h1 style="border:2px solid Tomato;">Hello World</h1>
<h1 style="border:2px solid DodgerBlue;">Hello World</h1>
<h1 style="border:2px solid Violet;">Hello World</h1>

<h1 style="background-color:DodgerBlue;color:Tomato;border:2px solid Violet; ">Hello World</h1>

<p style="background-color: grey; color: blue ; font-weight: bold ;font-size: 40px" >
      Hello Campers...
</p>
</html>
</body>
```



Hello World

Lorem ipsum...

Hello World

Lorem ipsum...

Ut wisi enim...

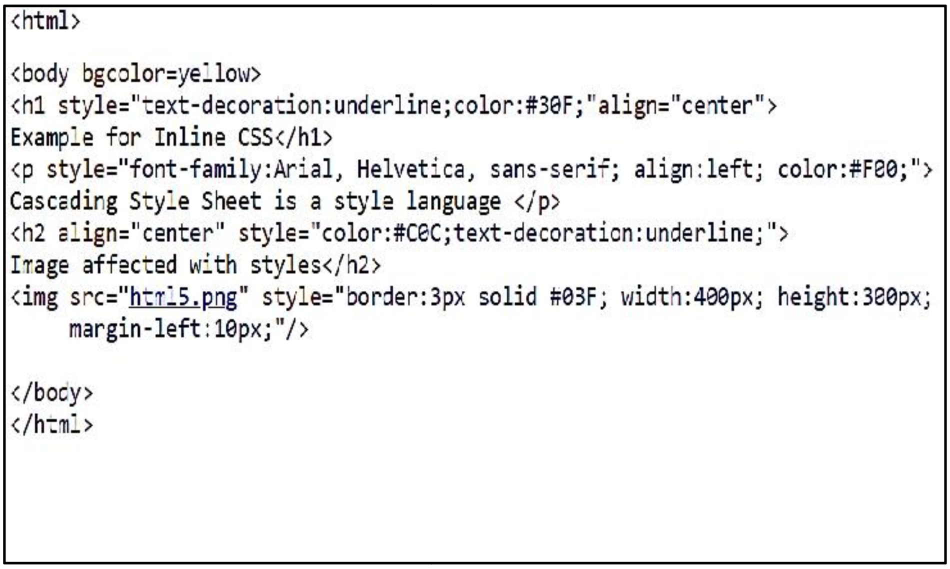
Hello World

Hello World

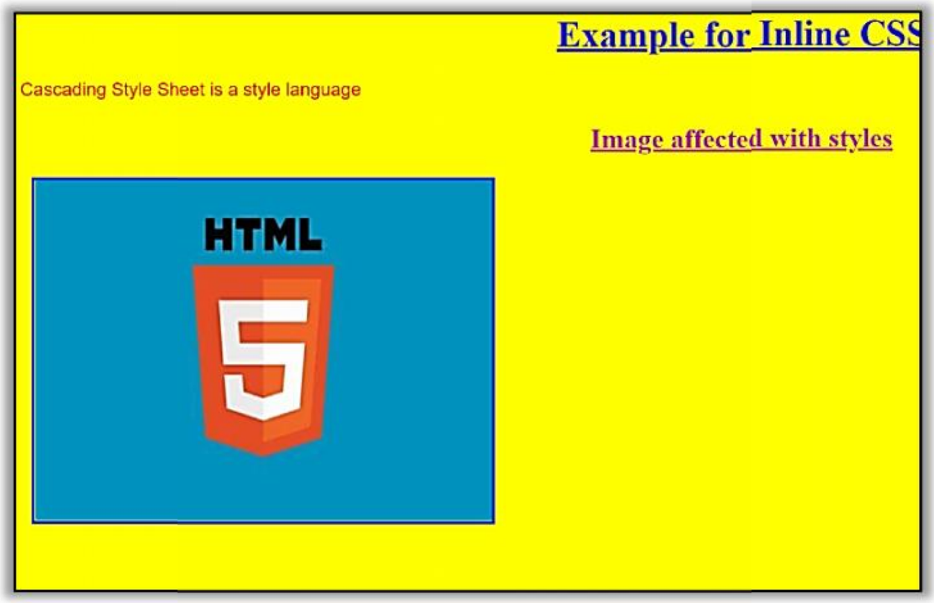
Hello World

Hello World

Hello Campers...



```
<html>

<body bgcolor=yellow>
<h1 style="text-decoration:underline;color:#30F;"align="center">
Example for Inline CSS</h1>
<p style="font-family:Arial, Helvetica, sans-serif; align:left; color:#F00;">
Cascading Style Sheet is a style language </p>
<h2 align="center" style="color:#C0C;text-decoration:underline;">
Image affected with styles</h2>
<img src="html5.png" style="border:3px solid #03F; width:400px; height:300px;
     margin-left:10px;"/>

</body>
</html>
```



Example for Inline CSS

Cascading Style Sheet is a style language

Image affected with styles

Summary

- **Cascading Style Sheets** (**CSS**) is a stylesheet language used to describe the presentation of a document written in HTML.
- You declare a CSS class by using a dot (.) followed by the class name. You make the class name yourself. After the class name, and then enter the properties/values that you want to assign to your class.
- It supports more colors and a wide range of color definitions.
- It allows you to change the style of the same HTML file in which you are working.
- Cascading Style Sheets fondly referred to as CSS, is a simple design language intended to simplify the process of making web pages presentable
- Allows Multiple backgrounds on a web page.
- Its display shadow with the text.
- Provides an easy and quick to add a style sheet to a web page. You do not need to create the whole pager or edit a new element in the head of your document.
- Displays the Web page only after the entire style sheet is loaded.

Keywords

CSS: stands for Cascading Style Sheets.

Selectors:CSS selectors are used to "find" (or select) the HTML elements.

Style:element contains style information for a document or part of a document.

Href: Specifies the location of the linked document.

@Import is used to import an external stylesheet in a manner similar to the <link> element. Here is the generic syntax of the @import rule.

Self Assessment

1. Which of the following property is used as the shorthand property for the padding properties?

A. padding-left

B. padding-right
C. padding
D. All of the above

2. The CSS property used to make the text bold is –
A. font-weight : bold
B. weight: bold
C. font: bold
D. style: bold

3. The CSS property used to specify the transparency of an element is -
A. Opacity1
B. filter
C. visibility
D. overlay

4. What is a CSS selector?
A. A CSS selector is the CSS class name
B. A CSS selector is the set of properties that are going to be applied on HTML elements
C. A CSS selector is name of CSS file.
D. A CSS selector is the first part of a CSS Rule. It may an HTML element or pattern of elements.

5. A CSS_____ are used to selecting HTML elements based on their element name, id, attributes, etc
A. Selectors
B. Syntax
C. Value
D. None of these

6. A CSS ___________consists of a selector, property, and its value.
A. Syntax rule
B. Plus rule
C. Declare rule
D. None of these

7. CSS stands for -
A. Cascade style sheets
B. Color and style sheets
C. Cascading style sheets
D. None of the above

8. Which of the following is the correct syntax for referring the external style sheet?
A. <style src = example.css>
B. <style src = "example.css" >
C. <stylesheet> example.css </stylesheet>
D. <link rel="stylesheet" type="text/css" href="example.css">

9. The property in CSS used to change the background color of an element is -
A. bgcolor
B. color
C. background-color
D. All of the above

10. The property in CSS used to change the text color of an element is –
A. bgcolor

B. color

C. background-color

D. All of the above

11. The CSS property used to control the element's font-size is -

A. text-style

B. text-size

C. font-size

D. None of the above

12. The HTML attribute used to define the inline styles is -

A. style

B. styles

C. class

D. None of the above

13. The HTML attribute used to define the internal stylesheet is -

A. <style>

B. style

C. <link>

D. <script>

14. Which of the following CSS property is used to set the background image of an element?

A. background-attachment

B. background-image

C. background-color

D. None of the above

15. Which of the following is the correct syntax to make the background-color of all paragraph elements to yellow?

A. p {background-color : yellow;}

B. p {background-color : #yellow;}

C. all {background-color : yellow;}

D. all p {background-color : #yellow;}

Answers for Self Assessment

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

Review Questions

1. What are the different ways you could integrate CSS into your HTML page?
2. What is the difference between the usage of an ID and a Class?
3. Elaborate the function of inline CSS with an example?

4. What are the different units used in CSS?
5. Name all the modules which are used in the current version of CSS?
6. What are CSS Selectors?
7. How to align images vertically in a division that spans vertically on the whole webpage?

Further Readings

- HTML5 Black Book (Covers CSS3, JavaScript, XML, XHTML, AJAX, PHP, jQuery) 2Ed.
- Web Enabled Commercial Application Development Using HTML, JavaScript, DHTML and PHP (4th Revised Edition)
- Trevor Burnham. *Async JavaScript*. The Pragmatic Bookshelf, Raleigh, NC and Dallas, TX, 2012.

Unit 04: CSS Class and ID

CONTENTS

Objectives

After studying this unit, you will be able to:

- Explain the Anatomy of a CSS Rule
- Discuss Class and ID Selector
- Learn about Pseudo class selector
- Learning about the background and Colors in CSS.

Introduction

In this chapter, we will understand the basic ingredients of CSS are the selectors, properties, and values. A CSS document can contain multiple selectors and a selector can have multiple properties which in turn can have a value consisting of one or several elements, we also learned about pseudo-classes, A Pseudo class in CSS is used to define the special state of an element. It can be combined with a CSS selector to add an effect to existing elements based on their states. Anatomy of CSSRule. CSS3 allows you to set the color of the background on the basis of various color specifications. It also enables you to set the transparency level of a color that has to be displayed in the background of an element.

4.1 Anatomy of a CSS3

A CSS comprises style rules that are interpreted by the browser and then applied to the corresponding elements in your document.

The CSS structure basics, from rulesets to values and units:

- Rules or rulesets
- Selectors
- Declarations
- Properties

- Values and units

A CSS rule or ruleset is a group of one or more CSS declarations - known as a declaration block, wrapped together by a CSS selector along with an opening and closing curly braces.

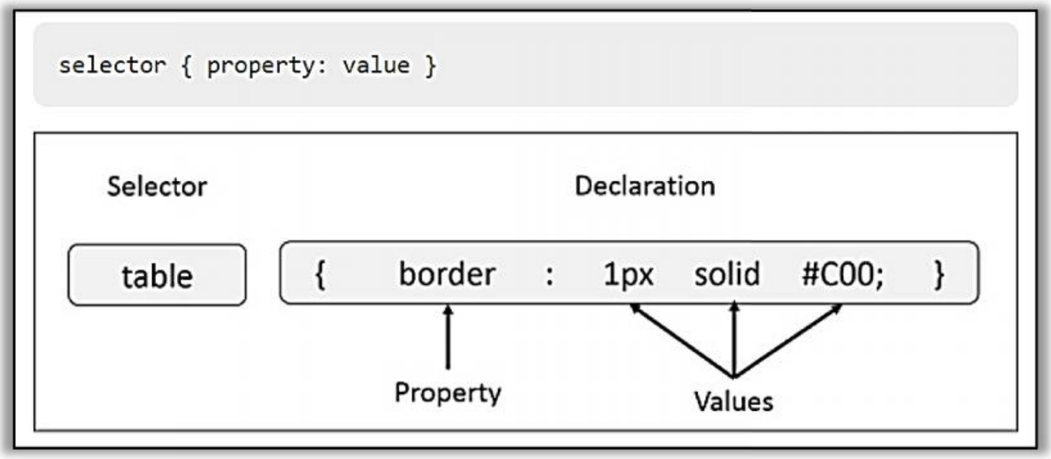
A CSS selector defines, identifies and targets the HTML elements to set the style of declarations applied. There are four categories.

- Type, class, and ID selectors
- Attribute selectors
- Pseudo-classes and pseudo-elements
- Combinators

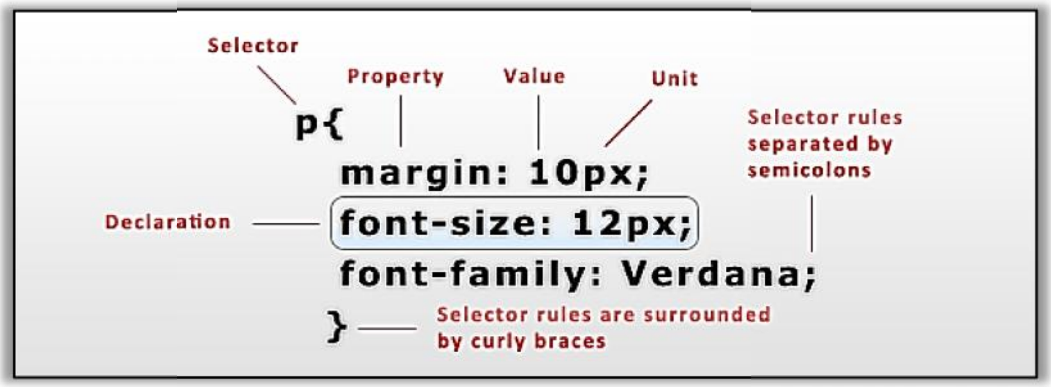
A CSS declaration is a property and value pair, with both entities separated by a colon punctuation mark and ended by a semicolon.

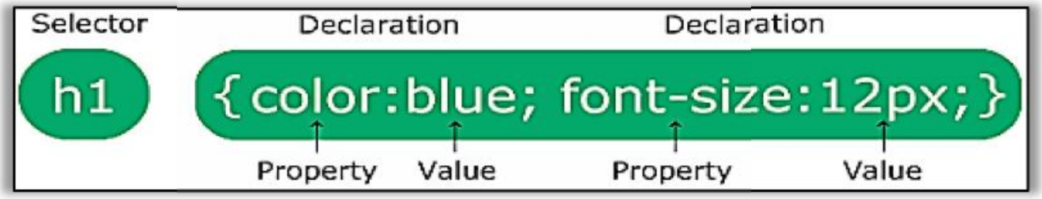
A CSS property is basically responsible for the exact style that you want to apply to the target element.

The value and units are separated by a colon punctuation mark while the unit is adjacent to the value.



```
selector { property: value }
```



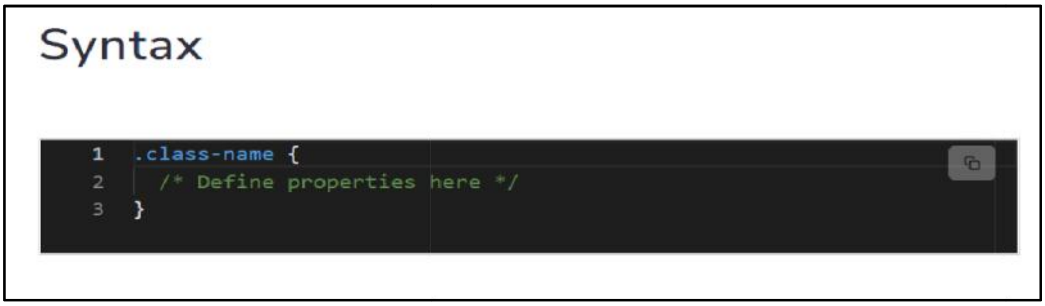


4.2 Class and ID Selector

In CSS, class and ID selectors are used to identify various HTML elements.

The main benefit of setting class or ID is that you can present the same HTML element differently, depending on its class or ID.

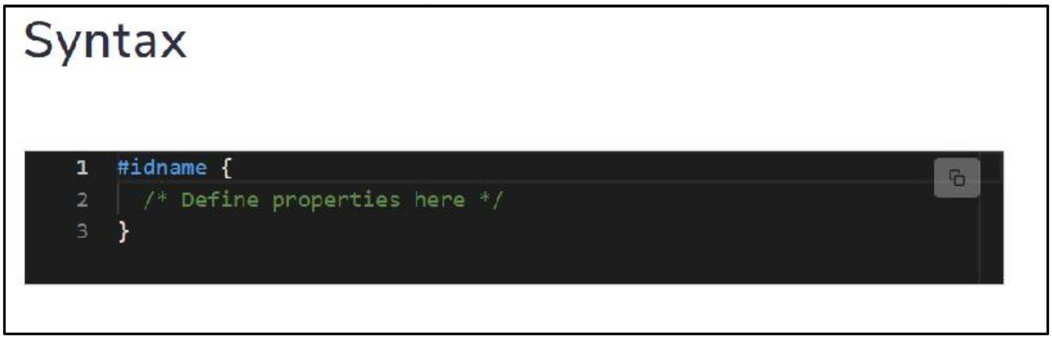
Class Selector The class selector selects elements with a specific class attribute. It matches all the HTML elements based on the contents of their class attribute. The. symbol, along with the class name, is used to select the desired class.



Syntax

```
.class-name {
  /* Define properties here */
}
```

ID selector The ID selector matches an element based on the value of its id attribute. In order for the element to be selected, its ID attribute must exactly match the value given in the selector. The # symbol and the id of the HTML element name are used to select the desired element.

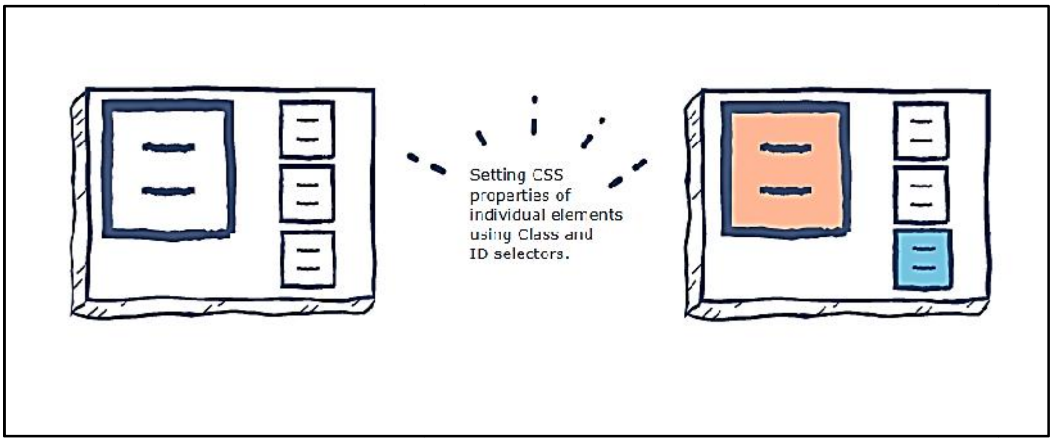


Syntax

```
#idname {
  /* Define properties here */
}
```

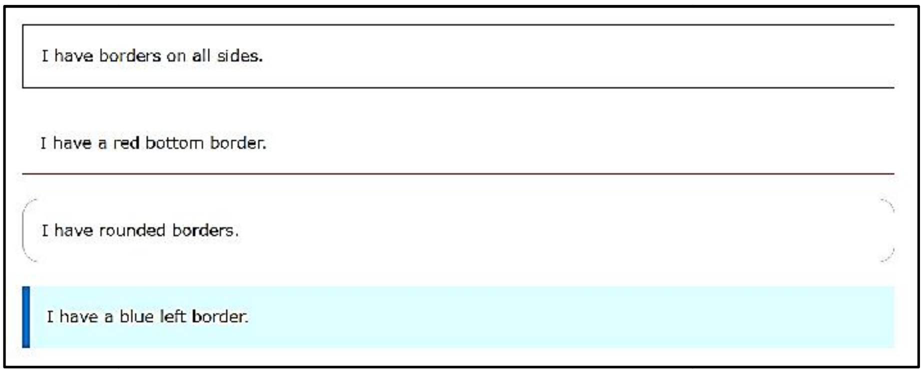
Difference Between Class sector and ID Selector

- The difference between an ID and a class is that an ID is only used to identify one single element in our HTML. IDs are only used when one element on the page should have a particular style applied to it. However, a class can be used to identify more than one HTML element.


Setting CSS properties of individual elements using Class and ID selectors.

4.3 CSS Borders

The CSS border properties allow you to specify the style, width, and color of an element's border.



I have borders on all sides.

I have a red bottom border.

I have rounded borders.

I have a blue left border.

The border-style property specifies what kind of border to display

- **The following values are allowed:**

  dotted - Defines a dotted border

  dashed - Defines a dashed border

  solid - Defines a solid border

  double - Defines a double border

  groove - Defines a 3D grooved border. The effect depends on the border-color value

Demonstration of the different border styles

p.dotted {border-style: dotted;}

p.dashed {border-style: dashed;}

p.solid {border-style: solid;}

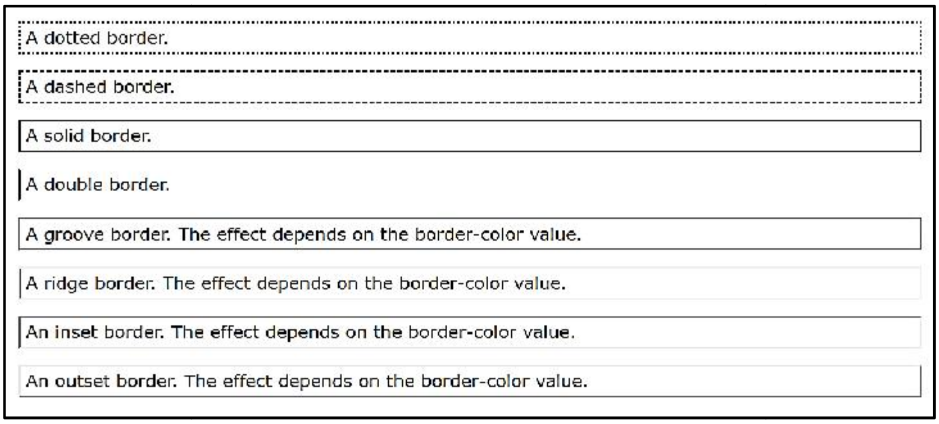
p.double {border-style: double;}

p.groove {border-style: groove;}

p.ridge {border-style: ridge;}

p.inset {border-style: inset;}

p.outset {border-style: outset;}



A dotted border.

A dashed border.

A solid border.

A double border.

A groove border. The effect depends on the border-color value.

A ridge border. The effect depends on the border-color value.

An inset border. The effect depends on the border-color value.

An outset border. The effect depends on the border-color value.

4.4 Pseudo-Class Selectors

A pseudo-class is used to define a special state of an element.

For example, it can be used to:

- Style an element when a user mouses over it
- Style visited and unvisited links differently

- Style an element when it gets focus

```
Selector: pseudo-class
{
Property: value;
}
```

Important points

While defining pseudo-classes in a <style>...</style> block, following points should be noted −

- a:hover MUST come after a:link and a:visited in the CSS definition in order to be effective.
- a:active MUST come after a:hover in the CSS definition in order to be effective.
- Pseudo-class names are not case-sensitive.
- Pseudo-class are different from CSS classes but they can be combined.

Links can be displayed in different ways:

```
/* unvisited link */
a:link {
  color: #FF0000;
}
/* visited link */
a:visited {
  color: #00FF00;
}
/* mouse over link */
a:hover {
  color: #FF00FF;
}
/* selected link */
a:active {
  color: #0000FF;
}
```

Pseudo-classes and HTML Classes

- Pseudo-classes can be combined with HTML classes:
- When you hover over the link in the example, it will change color:
- Example

```
a.highlight:hover
{
color: #ff0000;
}
```

Hover on <div>

- An example of using the :hover pseudo-class on a <div> element:
- Example

```
div:hover
```

```
{
  background-color: blue;
}
```

Simple Tooltip Hover

Hover over a <div> element to show a <p> element (like a tooltip):

Example

```
p {
  display: none;
  background-color: yellow;
  padding: 20px;
}
div:hover p {
  display: block;
}
```

What are Pseudo-Elements?

A CSS pseudo-element is used to style specified parts of an element.

For example, it can be used to:

Style the first letter, or line, of an element

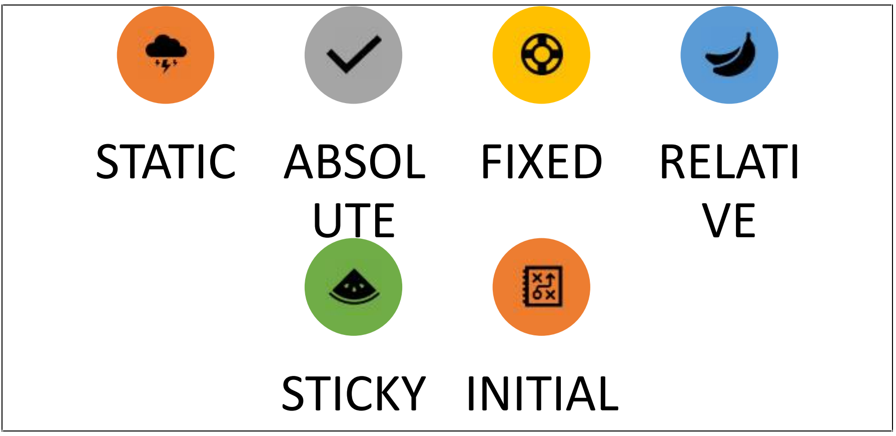
Insert content before, or after, the content of an element

The syntax of pseudo-elements:

```
Selector::pseudo-element {
property: value;
}
```

4.5 Style Placement in CSS3

The position property specifies the type of positioning method used for an element (static, relative, absolute, fixed, or sticky).



Static Default value. Elements render in order, as they appear in the document flow.

Absolute: The element is positioned relative to its first positioned (not static) ancestor element.

Fixed: The element is positioned relative to the browser window

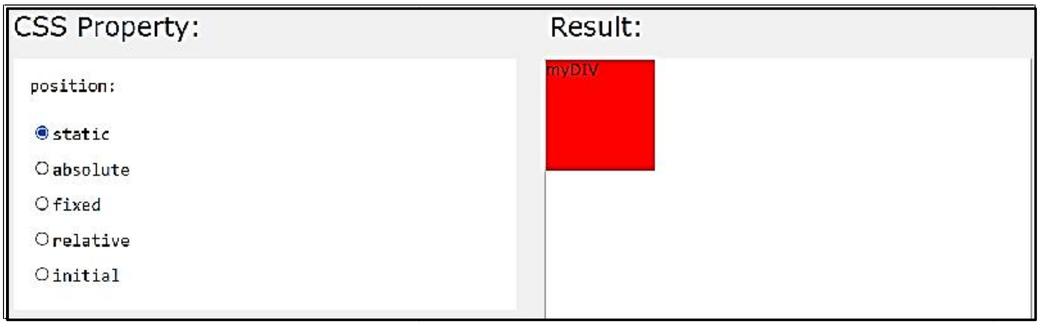
Relative: The element is positioned relative to its normal position, so "left:20px" adds 20 pixels to the element's LEFT position.

Sticky: The element is positioned based on the user's scroll position.

Example

```
h2.pos_left
{
  position: relative;
  left: -20px;
}
h2.pos_right
{
  position: relative;
  left: 20px;
}
```

- **Static**A static positioned element is always positioned according to the normal flow of the page. HTML elements are positioned static by default. Statically positioned elements are not affected by the top, bottom, left, right, and z-index properties.



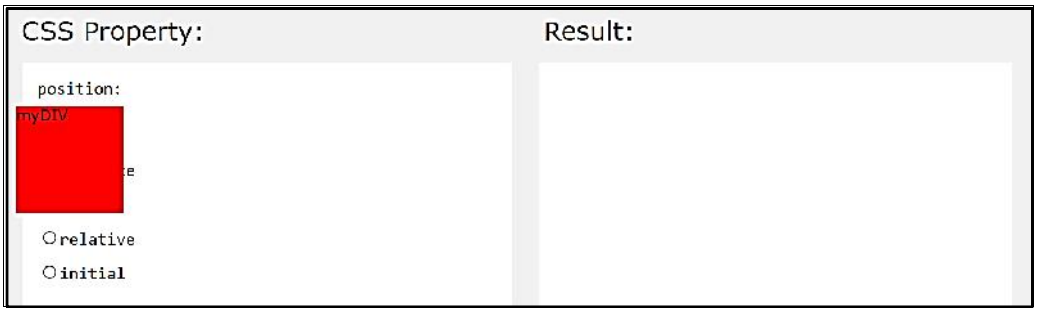
CSS Property:

Result:

position:

● static
○ absolute
○ fixed
○ relative
○ initial

- **Absolute**An absolutely positioned element is positioned relative to the first parent element that has a position other than static. If no such element is found, it will be positioned on a page relative to the 'top-left' corner of the browser window. The box's offsets further can be specified using one or more of the properties tops, right, bottom, and left.



CSS Property:

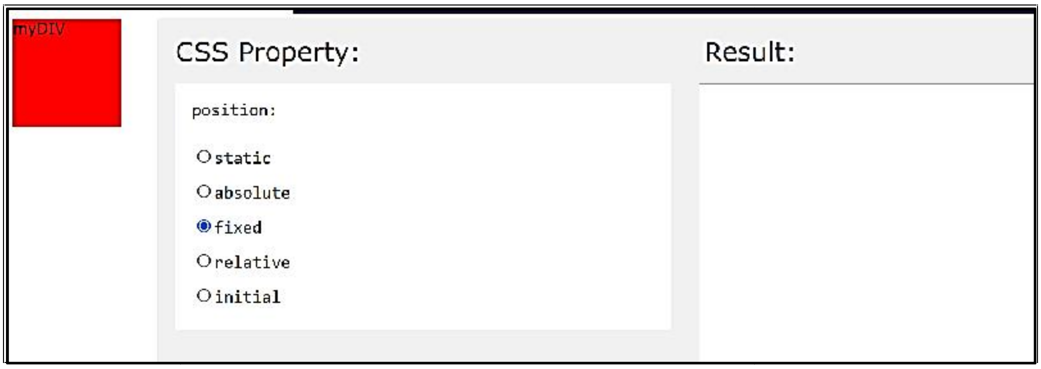
Result:

position:

○ relative
○ initial

- **Fixed positioning** is a subcategory of absolute positioning.

  The only difference is, a fixed positioned element is fixed with respect to the browser's viewport and does not move when scrolled



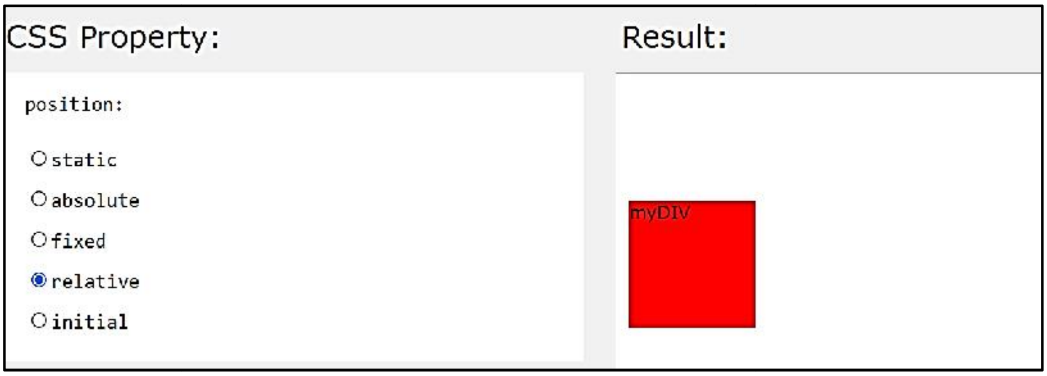
CSS Property:

position:

○ static
○ absolute
● fixed
○ relative
○ initial

Result:

- **Relative**A relative positioned element is positioned relative to its normal position.
- In the relative positioning scheme the element's box position is calculated according to the normal flow. Then the box is shifted from this normal position according to the properties — top or bottom and/or left or right.

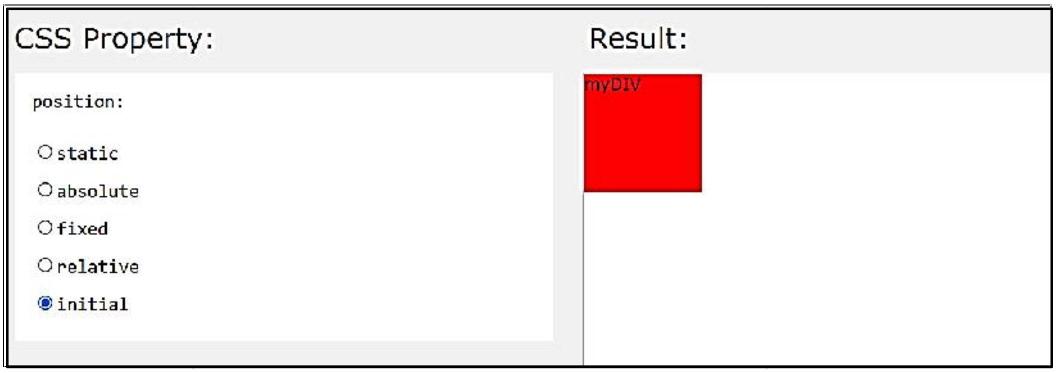

CSS Property:

position:

○ static
○ absolute
○ fixed
● relative
○ initial

Result:

- **Initial**


CSS Property:

position:

○ static
○ absolute
○ fixed
○ relative
● initial

Result:

4.6 Conflict Resolution

Font Selection is Important

Choosing the right font has a huge impact on how the readers experience a website.

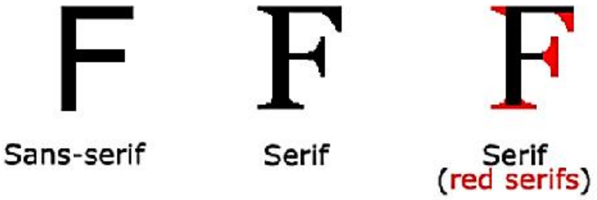
The right font can create a strong identity for your brand.

Using a font that is easy to read is important. The font adds value to your text. It is also important to choose the correct color and text size for the font.

Generic Font Families

In CSS there are five generic font families:

- Serif fonts have a small stroke at the edges of each letter. They create a sense of formality and elegance.
- Sans-serif fonts have clean lines (no small strokes attached). They create a modern and minimalistic look.
- Monospace fonts - here all the letters have the same fixed width. They create a mechanical look.
- Cursive fonts imitate human handwriting.
- Fantasy fonts are decorative/playful fonts.



The CSS font-family Property

In CSS, we use the font-family property to specify the font of a text.

Example

```
p1 {
  font-family: "Times New Roman", Times, serif;
}
.p2 {
  font-family: Arial, Helvetica, sans-serif;
}
.p3 {
  font-family: "Lucida Console", "Courier New", monospace;
}
```

CSS Google Fonts

- Google Fonts
- If you do not want to use any of the standard fonts in HTML, you can use Google Fonts.
- Google Fonts are free to use and have more than 1000 fonts to choose from.

How To Use Google Fonts

```
<head>
<link rel="stylesheet" href="https://fonts.googleapis.com/css?family=Sofia">
<style>
body {
  font-family: "Sofia", sans-serif;
}
</style>
</head>
```

Summary

- CSS is a set of rules which consist of a selector and adeclaration.
- A selector is used to select the element and apply styles too, and a declaration isthe combination of a property and a value for that element.
- Selectors are the conditions of a CSS ruleset. Theselectors come from theircapability to be combined, allowing you to create very specific conditions, applying styles only to the exact element.
- IDs and classes, which are attributes of an HTML element, that allowsbeing more specificwhen styling elements.
- Sometimes you may want to apply a style to more than one element, then you can use the group selectors.
- Dynamic pseudo-classes allow you to style an element.
- Borders are graphical elements that apply the edge of an element.
- We can use color values and functions to access more than 16 million colors, theyare RGB and HSL.

Keywords

CSS property is basically responsible for the exact style that you want to apply to the target element.

Class Selector The class selector selects elements with a specific class attribute. It matches all the HTML elements based on the contents of their class attribute.

ID selector The ID selector matches an element based on the value of its id attribute.

CSS border properties allow you to specify the style, width, and color of an element's border

Static Default value. Elements render in order, as they appear in the document flow.

Absolute: The element is positioned relative to its first positioned (not static) ancestor element.

Fixed: The element is positioned relative to the browser window

Relative: The element is positioned relative to its normal position, so "left:20px" adds 20 pixels to the element's LEFT position.

Sticky: The element is positioned based on the user's scroll position.

Self Assessment

1. A____ is an HTML tag at which a style will be applied. This could be any tag like <h1> or <table> etc.

A. Selector
B. Property
C. Value
D. None of these

2. A ______is a type of attribute of HTML tag in CSS rule syntax

A. Selector
B. Property
C. Value

D. None of these

3. ______are assigned to properties. For example, color property can havether red or #F1F1F1 etc

A. Selector

B. Property

C. Value

D. None of these

4. The _______specifies whether the browser should control the appearance of the adjacent borders that touch each other or whether each cell should maintain its style

A. border-collapse

B. border-spacing

C. Both of these

D. None of these

5. The _______specifies the width that should appear between table cells.

A. border-collapse

B. border-spacing

C. Both of these

D. None of these

6. Which of the following is not a value of the font-variant property in CSS?

A. normal

B. small-caps

C. large-caps

D. inherit

7. Which of the following CSS property is used to specify whether the table cells share the common or separate border?

A. border-collapse

B. border-radius

C. border-spacing

D. None of the above

8. The CSS property used to make the rounded borders, or rounded corners around an element is -

A. border-collapse

B. border-radius

C. border-spacing

D. None of the above

9. Which of the following tag is used to embed css in html page?

A. <css>

B. <!DOCTYPE html>

C. <script>

D. <style>

10. Which of the following CSS selectors are used to specify a group of elements?

A. tag

B. id

C. class

D. both class and tag

11. A ______is used to define a special state of an element.

A. Pseudo-class

B. Class

C. Rule class

D. None of these

12. Pseudo-classes are used to :

A. Style an element when a user mouses over it

B. Style visited and unvisited links differently

C. Style an element when it gets focus

D. All of the above

13. Pseudo-classes can be combined with _____classes:

A. HTML

B. CDS

C. GSS

D. None of these

14. The _______property specifies the type of positioning method used for an element.

A. Placing

B. Position

C. Accurate

D. None of these

15. An ______positioned element is positioned relative to the first parent element that has a position other than static.

A. Absolutely

B. Relatively

C. Static

D. All of the above

Answers for Self Assessment

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

Review Questions

1. Define the use of the ruleset.
2. Can we include the multiple CSS3 files in one HTML document? Explain?
3. What is the difference between the usage of an ID and a Class?
4. Elaborate the function of inline CSS with an example?
5. What are pseudo-elements in CSS? Explain with some code?
6. How would you define the Pseudo-classes in CSS3?
7. How many ways can a CSS be integrated as a web page?
8. Define the property used for image scroll controlling?
9. What are the differences between relative and absolute in CSS? Explain?

Further Readings

- HTML5BlackBook(CoversCSS3,JavaScript,XML,XHTML,AJAX,PHP,jQuery)2Ed.
- WebEnabledCommercialApplicationDevelopmentUsingHTML,JavaScript,DHTMLandPHP(4thRevisedEdition)
- Trevor Burnham. *Async JavaScript*. The Pragmatic Bookshelf, Raleigh, NC and Dallas, TX, 2012.

Unit 05: Advanced Topics of CSS3

CONTENTS

Objectives

After studying this unit, you will be able to:

- Discuss the Styling Text
- Explain The Box Model
- Learn about the Background Property
- Discuss Positioning elements

Introduction

In this chapter,you learn about Styling Text, The Box Model, Outline Property, and Positioning elements. The box model and various aspects related to it. Such as margin, padding, width, and border. Next, we will learn about different backgrounds, colors, and gradient effects on your web pages. You learn about different background properties, such as background-position and background-repeat, and how to set multiple background values on a single web page. Next will discuss positioning elements, InCSS helps you to place your HTML element. You can put any HTML element at whatever location you like. You can require whether you want the element positioned relative to its natural position on the page or absolute based on its parent element.

5.1 Styling Text

The font represents the style and size of the text that is displayed in a web browser. Apart from communicating a visual request to the content, fonts are also used to help users discriminate between different types of information.

The CSS properties used to style text generally tumble into two categories.

- **Font styles**: Properties that affect a text's font, e.g., which font gets applied, its size, and whether it's bold, italic, etc.

- **Text layout styles**: Properties that affect the spacing and other layout features of the text, allowing manipulation of, for example, the space between lines and letters, and how the text is aligned within the content box.

CSS provides three values for setting the font styles of text: normal, oblique, and italic. The normal style displays the text in upright and straight letters, oblique displays the text in slanting or learning letters, and italic displays the text in italic letters.

```
<style>
P{
Font-size:20px;
}
Span.normal {
Font-style:normal;
{
Span.italic{
Font-style:italic;
}
Span.oblique{
Font-style:oblique;
}
</style>
```

The text-align property is used to control how text is aligned within its holding content box. The following possible values are used with text-align property.

- Centre-Changes the text in the middle of the containing element.
- Justify-Fits the text in the containing element.
- Left-Align the text on the left side of the containing element.
- Right-Align the text on the right side of the containing element.

5.2 Box Model

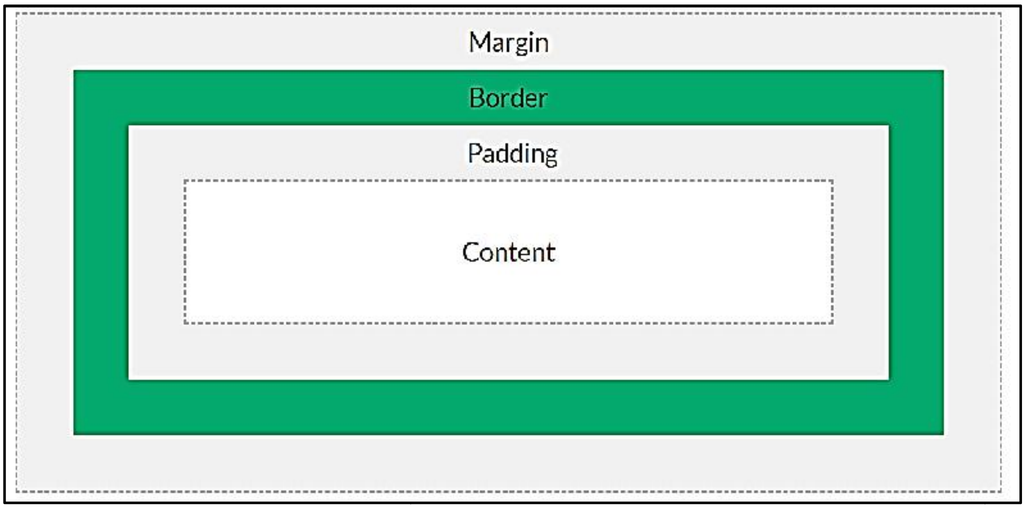
CSS treats an HTML document as an ordered tree of elements, where each element can have one or more child elements arranged in a well-organized way. The topmost element of this tree is called as the root element or the parent element. These elements display their content at a specific position, which is defined by using CSS properties.CSS to convert the data of HTML elements into the form of rectangular boxes, by using these outlines is called the box model to set the design of HTML documents. The box model describes how HTML elements are displayed as boxes.

The CSS box model is essentially a box that wraps around every HTML element. It consists of: margins, borders, padding, and the actual content.

The CSS box model is essentially a box that wraps around every HTML element. It consists of:

- **Margins**: In CSS margin area specifies the area around the border area.
- **Borders:** In CSS Border area specifies the area around the padding area
- **Padding:** In CSS padding area specifies the area around the content area.
- **Content**: In CSS content area specifies the displays the content of a document, such as text and images. This is bounded by a rectangle, which is called the content edge.

Note Content area always appears inside the padding area.



A CSS box model is a compartment that includes numerous assets, such as edge, border, padding, and material. It is used to develop the design and structure of a web page. It can be used as a set of tools to mark the layout of different components. According to the CSS box model, the web browser supplies each element as a square prism.

The various other features of the box model are in the following sections:

- The padding properties
- The border properties
- The margin properties
- The width height properties

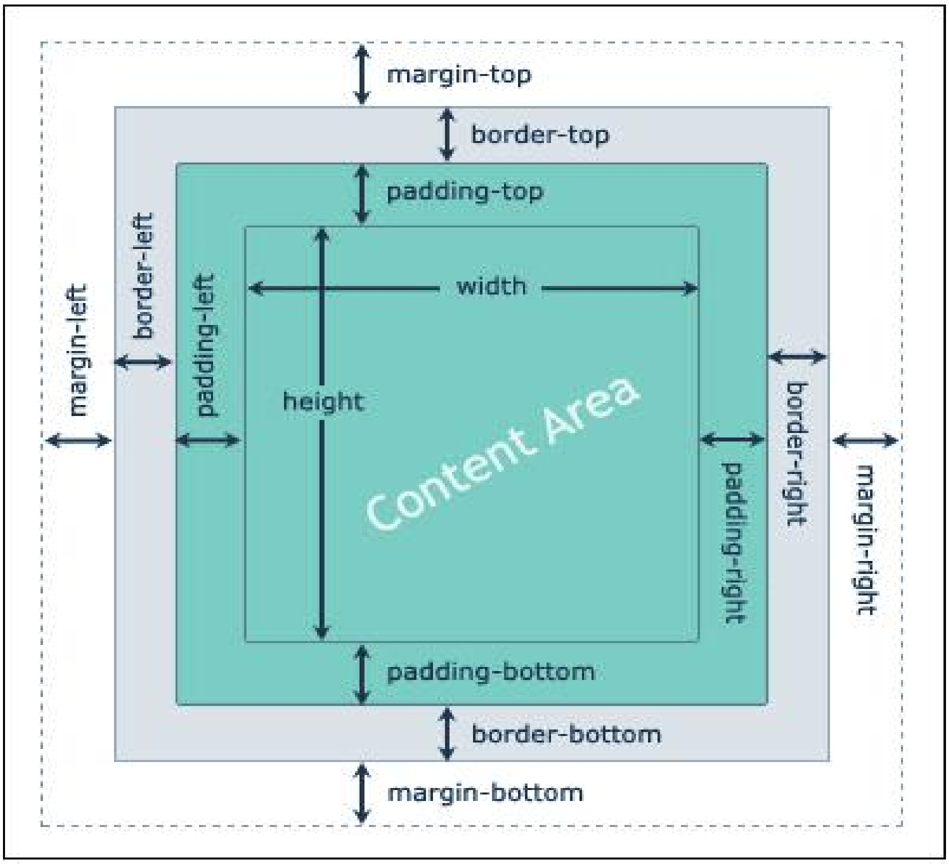
Padding Properties- is describing the distance between the border of an element and the content within it. The padding property is affected by the background color of an element; the value of the padding cannot be negative.

The padding property is used to change all the properties such as the padding-top, padding-bottom, padding-right, padding-left.

Border Properties- it specifies the space between the padding and converts it into a box model.It defines the width, colour, and style of the border area of the box.

Margin Properties- the blank area around the border of an element is called the margin. It is used to create extra space around an element. It is totally transparent and does not contain any background colour. margin is also used to fix the spacing around different elements. The margin property is used to set all the sides of an element, such as top, right, bottom, and left.

Width-height properties-specifies the width and height of the content area, padding area, or border area of a box.



Example,

```
  div {
    width: 300px;
    border: 15px solid green;
    padding: 50px;
    margin: 20px;
}
```

- In order to set the width and height of an element correctly in all browsers, you need to know how the box model works.
- Important: When you set the width and height properties of an element with CSS, you just set the width and height of the content area. To calculate the full size of an element, you must also add padding, borders, and margins.

Example:

- This <div> element will have a total width of 350px:

```
div {
width: 320px;
  padding: 10px;
  border: 5px solid gray;
 margin: 0;
```

How to calculate:

- 320px (width)

- + 20px (left + right padding)
- + 10px (left + right border)
- + 0px (left + right margin)
- = 350px

How to calculate the width of an element

The total width of an element should be calculated like this:

Total element width = width + left padding + right padding + left border + right border + left margin + right margin

The total height of an element should be calculated like this:

Total element height = height + top padding + bottom padding + top border + bottom border + top margin + bottom margin.

CSS Outline

- An outline is a line drawn outside the element's border.
- An outline is a line that is drawn around elements, OUTSIDE the borders, to make the element "stand out**".**

5.3 The Background Property

The background of a web page is the area on which the content of the web page, such as text, tables, borders, and images, is displayed. A web page should have a background that expresses the axiom of the web page. For example, while constructing a web page for an organization, the background can have a logo that represents the organization.

The CSS background property is used to define the background effects on the element. The following CSS background properties affect the HTML elements:

1. background-color
2. background-image
3. background-repeat
4. background-attachment
5. background-position

The background-colorproperty is used to set the color of the background area on which an element is displayed. It can be applied to any element. The background color property takes any of the following three values.

- Color name
- Hexadecimal equivalence of the color
- RGB color

Example,

```
h1
{  background-color:# FFFFFF;}
```

The Background-imageproperty is used to set an image in the background of an element. It is parallel to the background quality of the body element of HTML.

Example,

```
Body
{background-image: url(abc.jpg”)
```

The background-repeatpropertyallows you to piece the background images along the x-axis and y-axis of an element. The background-repeat property can take either of the following values:

- Repeat-x
- Repeat-y
- Repeat
- No-repeat

Example,

{

<body style="background: url('abc.jpg'); background-repeat:repeat-y,">

The background-attachmentproperty is used to fix or scroll the background image along with the text and other content displayed on it. The two values take this property fixed or scroll. Fixed means the background image does not move and scroll means the image scroll along with the text written over it.

Example,

{

Body {background-image:url('abc.jpg'); background -attachement:scroll;}

}

Example,

{

Body {background-image:url('abc.jpg'); background -position: right top;}

}

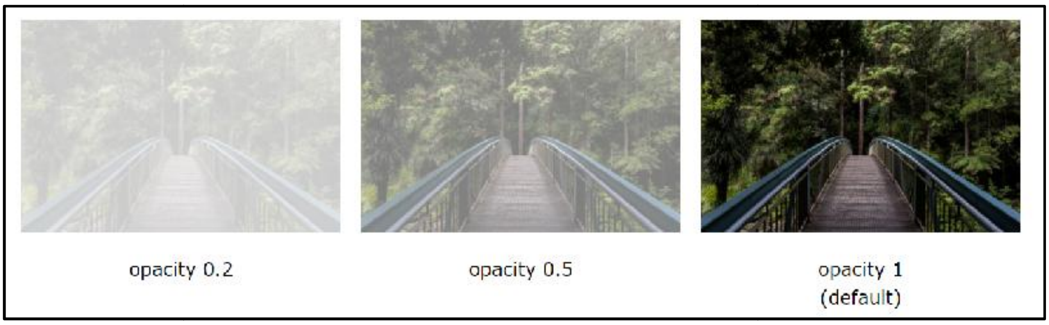
5.4 Exploring the Color Property

The RGB format uses three elementary colors, Red, Green, and blue, to specify the color of an element. CSS3 adds a new level, Alpha(A), the level of opacity of this RGB format.

CSS Opacity / Transparency

The opacity property is used to produce a transparency effect in an HTML element. When you use the opacity property for an HTML element, it is also applied to its child elements.

- The opacity property specifies the opacity/transparency of an element.
- The opacity property can take a value from 0.0 - 1.0. The lower value, the more transparent**:**



Transparent Hover Effect

- The opacity property is often used together with the: hover selector to change the opacity on mouse-over.

5.5 Positioning Elements

The **position** property in CSS talks about the technique of positioning for an element or an HTML entity.

The CSS position property is used to set the position for an element. it is also used to place an element behind another and is also useful for the scripted animation effect.

Understand CSS Layout - float and clear property

The CSS float property specifies how an element should float.

The CSS clear property specifies what elements can float beside the cleared element and on which side.

Float Left

Float Right

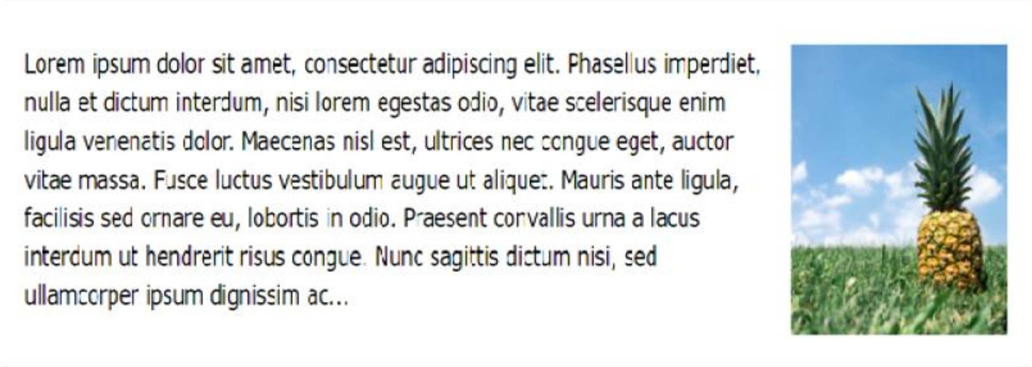
Float Properties

- The float property is used for positioning and formatting content e.g. let an image float left to the text in a container.
- The float property can have one of the following values:
- left - The element floats to the left of its container
- right - The element floats to the right of its container
- None - The element does not float (will be displayed just where it occurs in the text). This is default
- Inherit - The element inherits the float value of its parent.

Note: In its simplest use, the float property can be used to wrap text around images.

Example of float right

```
img
{
  float: right;
}
```

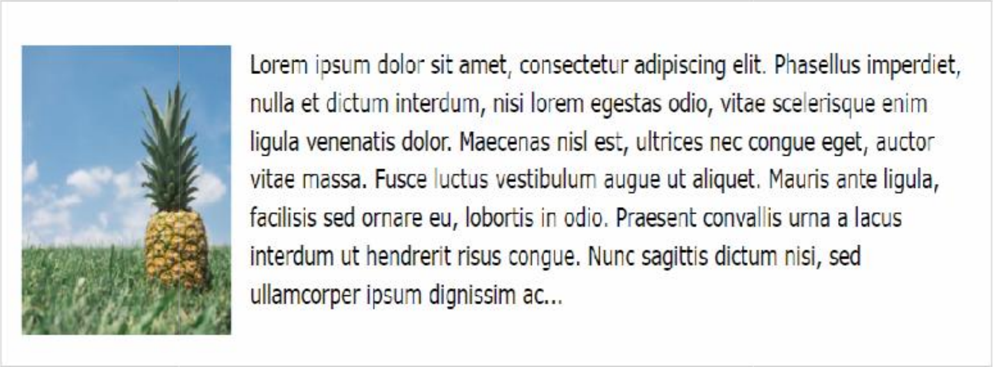


Lorem ipsum dolor sit amet, consectetur adipiscing elit. Phasellus imperdiet, nulla et dictum interdum, nisi lorem egestas odio, vitae scelerisque enim ligula venenatis dolor. Maecenas nisl est, ultrices nec congue eget, auctor vitae massa. Fusce luctus vestibulum augue ut aliquet. Mauris ante ligula, facilisis sed ornare eu, lobortis in odio. Praesent convallis urna a lacus interdum ut hendrerit risus congue. Nunc sagittis dictum nisi, sed ullamcorper ipsum dignissim ac...

Example of Float: left:

```
img {
```

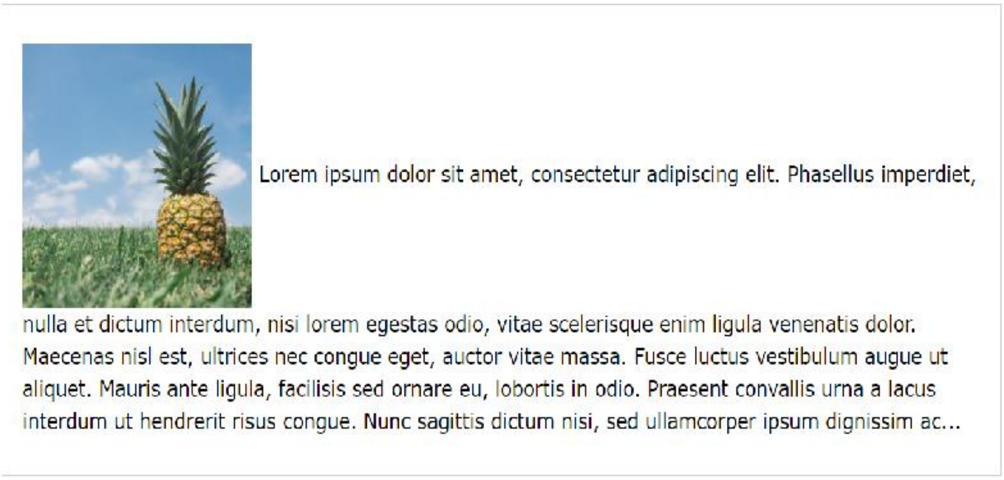
```
float: left;
}
```



Lorem ipsum dolor sit amet, consectetur adipiscing elit. Phasellus imperdiet, nulla et dictum interdum, nisi lorem egestas odio, vitae scelerisque enim ligula venenatis dolor. Maecenas nisl est, ultrices nec congue eget, auctor vitae massa. Fusce luctus vestibulum augue ut aliquet. Mauris ante ligula, facilisis sed ornare eu, lobortis in odio. Praesent convallis urna a lacus interdum ut hendrerit risus congue. Nunc sagittis dictum nisi, sed ullamcorper ipsum dignissim ac...

Example of No float

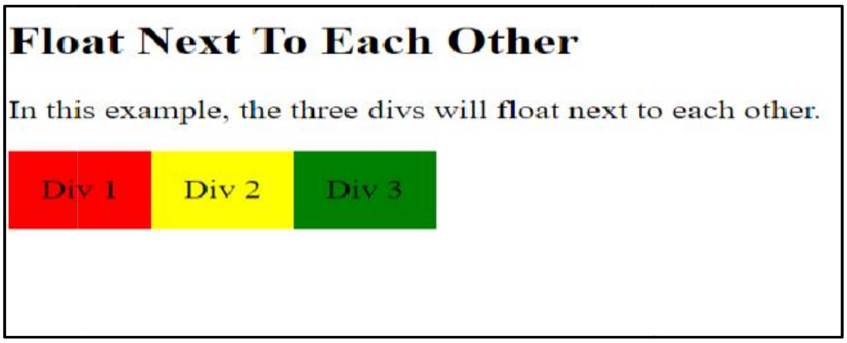
- In the following example the image will be displayed just where it occurs in the text (float: none ;):



Lorem ipsum dolor sit amet, consectetur adipiscing elit. Phasellus imperdiet, nulla et dictum interdum, nisi lorem egestas odio, vitae scelerisque enim ligula venenatis dolor. Maecenas nisl est, ultrices nec congue eget, auctor vitae massa. Fusce luctus vestibulum augue ut aliquet. Mauris ante ligula, facilisis sed ornare eu, lobortis in odio. Praesent convallis urna a lacus interdum ut hendrerit risus congue. Nunc sagittis dictum nisi, sed ullamcorper ipsum dignissim ac...

Example - Float Next To Each Other

- Normally div elements will be displayed on top of each other. However, if we use float: left we can let elements float next to each other.



Float Next To Each Other

In this example, the three divs will float next to each other.

The Clear Property

- When we use the float property, and we want the next element below (not on right or left), we will have to use the clear property.

- The clear property specifies what should happen with the element that is next to a floating element

Clear property can have one of the following values:

- None - The element is not pushed below left or right floated elements. This is default
- left - The element is pushed below left floated elements
- right - The element is pushed below right floated elements
- both - The element is pushed below both left and right floated elements
- inherit - The element inherits the clear value from its parent

When clearing floats, you should match the clear to the float: If an element is floated to the left, then you should clear to the left. Your floated element will continue to float, but the cleared element will appear below it on the web page.

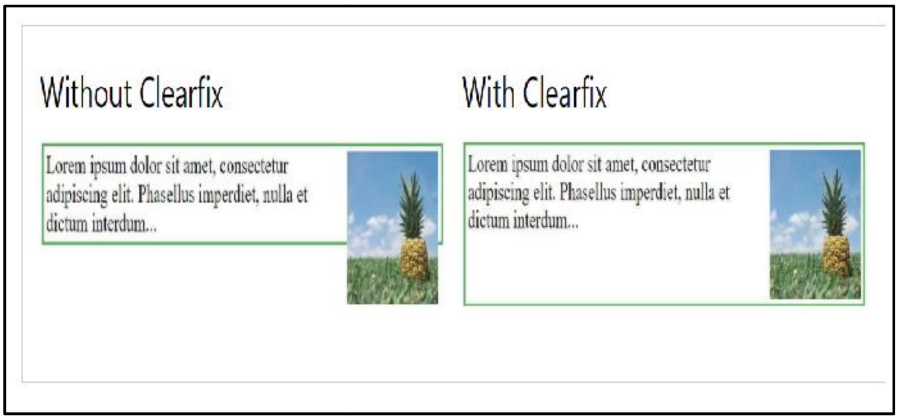
Example,

This example clears the float to the left. Here, it means that the <div2> element is pushed below the left floated <div1> element:

```
div1
{
 float: left;
}
div2
{
 clear: left;
}
```

The Clearfix Hack

- If a floated element is taller than the containing element, it will "overflow" outside of its container. We can then add a clearfix hack to solve this problem:



Without Clearfix

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Phasellus imperdiet, nulla et dictum interdum...

With Clearfix

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Phasellus imperdiet, nulla et dictum interdum...

Relative Positioning

Relative Positioning changes the position of the HTML element relative to where it normally appears. So “left30” 30 pixels to the element LEFT position.

You can use two values top and left along with the position property to move an HTML element anywhere in the HTML document.

Move left

Move right

Move top

Move bottom

```
<html>
<head>
</head>
<body>

<div style = “position:relative; left 50px; top:3px; background-color red”>

This is a CSS3 document.

</div>
</body>
</html>
```

Note: You can use *bottom* or *right* values as well in the same way as *top* and *left.*

Absolute Positioning

An element with position: absolute is positioned at the specified coordinates relative to your screen top-left
corner.
You can use two values top and left along with the position property to move an HTML element anywhere in the HTML document.

Move left

Move right

Move top

Move bottom

```
<html>
<head>
</head>
<body>

<div style = “position:relative; left 50px; top:3px; background-color red”>

This is a CSS3 document.

</div>
</body>
</html>
```

Summary

- Font’s properties help the person who reads to easily understand and identify important terms and information in a document.
- Box model is used when talking about design and layout. The CSS box model is essentially a box that wraps around every HTML element.
- Background properties at the same place in a style sheet, can be used to specify the values for the background color, background image, background-repeat, background-attachment, background position, and background-size properties.
- The opacity property is used to produce a transparency effect in an HTML element.
- Internet browser does not support the opacity property.
- Css3 enables to set ofa gradient background for an element. Gradient background is a variation of colors,which are arranged from lightest to darkest or vice versa.
- Background color property can take the value in three ways: as the direct color name, in hexadecimal color format, or in RGB color format.

- Color properties in CSS enables you to add different types of colors to our web page using the different specification. these specifications are RGB,RGBA,HSL,HSLA and opacity.
- Positioning allows you to take elements out of normal document flow and make them behave differently.
- Absolute values are the fixed value that is used to specify the font size on a web page.
- Relative value is how to set the font size of the text using relative values.

Keywords

Font sizeis used to change the size of the text.

Absolute Value Refers to the absolute size of the font.

Font-style property is used to specify the style of the font.

Text-align sets the horizontal alignment of the text.

Padding property is set the values for all four directions in a box layout model.

Border is specified between the padding and content in the box model.

URL () value allows you to provide the file path to any image, and it will show the background for that element

Self Assessment

1 Which of the following property adds padding to the top of an element?

A. height
B. padding-height
C. top
D. padding-top

2.A CSS _________is a compartment that includes numerous assets, such as edge, border, padding and material.

A. box model
B. table
C. Bootstrap
D. None of these

3.Which of the following property defines the style for the bottom border of an element?

A. border-bottom-style
B. border-collapse
C. border-style-bottom
D. none of the mentioned

4.Which of the following property defines the style for the Top border of an element?

A. border-top -style

B. border-collapse
C. border-style-top
D. none of the mentioned

5. Which of the following property defines the style for the Right border of an element?

A. border-right-style
B. border-left -style
C. Margin-right
D. none of the mentioned

6. Which of the following property defines the style for the left border of an element?

A. border-right-style
B. border-left -style
C. Margin-left
D. none of the mentioned

7. What clears the area around content?

A. Padding
B. Border
C. Margin
D. None of these

8.A ________ is that goes around the padding and content

A. Padding
B. Border
C. Margin
D. None of these

9.Clears an area outside the border

A. Padding
B. Border
C. Margin
D. None of these

10.Which of the following visibility property value is described by The element is not visible, but the layout of surrounding elements is not affected?

A. visible
B. hidden

C. collapse

D. none of the mentioned

11. Which of the following property is used to control the behavior of floating elements?

A. format

B. clean

C. clear

D. remove

12. The ______ property is used for positioning and formatting content

A. Float

B. clean

C. clear

D. remove

13. The CSS______ property specifies what elements can float beside the cleared element and on which side

A. format

B. clean

C. clear

D. remove

14. The default value of float is

A. None

B. Right

C. Left

D. Top

15. Which float property ensures the element floats to the right of its container

A. None

B. Right

C. Left

D. Top

Answers for Self Assessment

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

Review Questions

1. How do assign absolute values using the font-size property?
2. Explain font-size property using percentage values?

3. What is the main property of the Box Model in CSS Explain with an example?
4. What are the two properties that influence the dimensions of the content in the CSS Box Model?
5. **Tell us about the property used for image scroll controlling**
6. How to use the margin property of a box Explain with an example?
7. How to set the top margin property of an element? Negative values can be allowed in this property?
8. Display a margin of 10cm from the right and 10 am from the top?
9. Tell us about column-span and column fill properties?

Further Readings

- HTML5BlackBook(CoversCSS3,JavaScript,XML,XHTML,AJAX,PHP,jQuery)2Ed.
- WebEnabledCommercialApplicationDevelopmentUsingHTML,JavaScript,DHTMLandPHP(4thRevisedEdition)
- Trevor Burnham. *Async JavaScript*. The Pragmatic Bookshelf, Raleigh, NC and Dallas, TX, 2012.

Unit 06: Bootstrap

CONTENTS

Objectives

After studying this unit, you will be able to:

- Discuss introduction regarding Bootstrap
- Explain function of Bootstrap
- Web design, Grid System, Header Section
- Bootstrap Model and bootstrap Carousel

Introduction

The most popular HTML, CSS, and JavaScript framework for creating responsive and mobile-friendly websites is called Bootstrap. Utilization and download are both completely free. It is a front-end framework used to make web development simpler and quicker. It offers design templates for typography, forms, buttons, tables, navigation, modals, image carousels, and many other things that are based on HTML and CSS.

Bootstrap includes HTML and CSS based design templates for typography, forms, buttons, tables, navigation, modals, image carousels and many other, as well as optional JavaScript plugins.

Example: Bootstrap is a free front-end framework for faster and easier web development. It also give you the ability to easily create responsive designs.

Why to Learn Bootstrap

Mobile first approach

Bootstrap 3, framework consists of Mobile first styles throughout the entire library instead them of in separate files.

Browser Support

It is supported by all popular browsers.

Easy to Get Started

With just the knowledge of HTML and CSS, anyone can get started with Bootstrap. Also, the Bootstrap official site has good documentation.

Responsive design

Bootstrap's responsive CSS adjusts to Desktops, Tablets and Mobiles. More about the responsive design is in the chapter Bootstrap Responsive Design. Provides a clean and uniform solution for building an interface for developers.It contains beautiful and functional built-in components which are easy to customize.It also provides web-based customization.And best of all it is open-source.

6.1 Applications of Bootstrap

- **Scaffolding**: Bootstrap provides a basic structure with Grid System, link styles, and background.
- **CSS**: Bootstrap comes with the feature of global CSS settings, fundamental HTML elements styled and enhanced with extensible classes, and an advanced grid system.
- **Components**: Bootstrap contains over a dozen reusable components built to provide iconography, dropdowns, navigation, alerts, pop-overs, and much more.

6.2 Where to Get Bootstrap?

Boot is an open source software and it can be downloaded form getbootstrap.com

If you want to download and host Bootstrap yourself, go to getbootstrap.com, and follow the instructions there.

OR:

Include Bootstrap from a CDN

- If you don't want to download and host Bootstrap yourself, you can include it from a CDN (Content Delivery Network).
- MaxCDN provides CDN support for Bootstrap's CSS and JavaScript. You must also include jQuery.

6.3 Bootstrap CDN

You must include the following Bootstrap’s CSS, JavaScript, and jQuery from MaxCDN into your web page.

```
        <!—Latest compiled and minified Bootstrap CSS -->
<link rel="stylesheet"href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.
css">

        <!-- Latest compiled Bootstrap JavaScript -->
<script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/js/bootstrap.min.js"></script>

        <!-- latest jQuery library -->
<script src="https://code.jquery.com/jquery-latest.js"></script>
```

Advantage of Using Bootstrap CDN

- Many users already have downloaded Bootstrap from MaxCDN when visiting another site.

- As a result, it will be loaded from cache when they visit your site, which leads to faster loading time.
- Also, most CDN's will make sure that once a user requests a file from it, it will be served from the server closest to them, which also leads to faster loading time.

Create Web Page with Bootstrap

Add the HTML5 doctype

Bootstrap uses HTML elements and CSS properties that require the HTML5 doctype.Always include the HTML5 doctype at the beginning of the page, along with the lang attribute and the correct character set:

Bootstrap uses HTML elements and CSS properties that require the HTML5 doctype.Always include the HTML5 doctype at the beginning of the page, along with the lang attribute and the correct character set:

```
<!DOCTYPE html>

<html lang="en">
  <head>
    <meta charset="utf-8">
  </head>

</html>
```

Create Web Page with Bootstrap

Bootstrap 3 is designed to be responsive to mobile devices. Mobile-first styles are part of the core framework.

Bootstrap is mobile-first

To ensure proper rendering and touch zooming, add the following <meta> tag inside the <head> element:

<meta name="viewport" content="width=device-width, initial-scale=1">

Bootstrap is mobile-first

The width=device-width part sets the width of the page to follow the screen-width of the device (which will vary depending on the device).

The initial-scale=1 part sets the initial zoom level when the page is first loaded by the browser.

Containers

Bootstrap also requires a containing element to wrap site contents.

There are two container classes to choose from:

The .container class provides a responsive **fixed width container.**

The .container-fluid class provides a **full width container**, spanning the entire width of the viewport.

Example: Containers are not nestable (you cannot put a container inside another container).

Bootstrap Grids

Bootstrap’s grid system allows up to 12 columns across the page.

If you do not want to use all 12 columns individually, you can group the columns together to create wider columns:

```
<div class="col-md-12">Span 12 columns</div>

<div class="col-md-6">Span 6</div><div class="col-md-6">Span 6</div>

<div class="col-md-4">Span 4</div><div class="col-md-8">Span 8</div>

<div class="col-md-4">Span 4</div><div class="col-md-4">Span 4</div><div class="col-md-4">Span 4</div>
```

Bootstrap's grid system is responsive, and the columns will re-arrange automatically depending on the screen size.

Grid Classes

The Bootstrap grid system has four classes:

- xs (for phones)
- sm (for tablets)
- md (for desktops)
- lg (for larger desktops)

The classes above can be combined to create more dynamic and flexible layouts.

```
<!DOCTYPE html>

<html lang="en">

<head><meta http-equiv="Content-Type" content="text/html; charset=windows-1252">

<title>Any title</title>

</head>

<body>

//write code

</body>

</html>
```

Add the HTML 5 doctype:

Bootstrap uses HTML elements and CSS properties, so you have to add the HTML 5 doctype at the beginning of the page with lang attribute and correct character set.

- Bootstrap 3 is designed to be responsive to mobile devices.
- Mobile-first styles are part of the core framework of Bootstrap. You have to add the following <meta> tag inside the <head> element for proper rendering and touch zooming:

Note: The "width=device-width" part is used to set the width of the page to follow the screen-width of the device (vary according to the devices).

The initial-scale=1 part is used to set the initial zoom level when the page is first loaded by the browser.

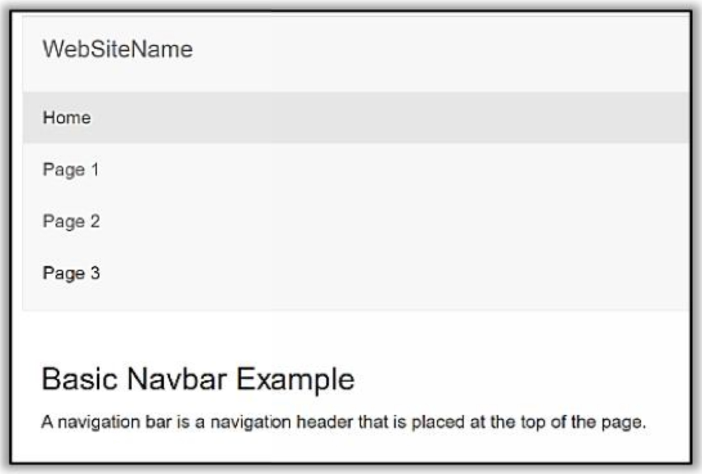
Bootstrap is mobile friendly

Bootstrap 3 is designed to be responsive to mobile devices.Mobile-first styles are part of the core framework of Bootstrap.You have to add the following <meta> tag inside the <head> element for proper rendering and touch zooming:

The "width=device-width" part is used to set the width of the page to follow the screen-width of the device (vary according to the devices).The initial-scale=1 part is used to set the initial zoom level when the page is first loaded by the browser.

Bootstrap Navigation bar

- A navigation bar is like a navigation header that is placed at the top of the page. You can collapse or extend it according to the screen size.



WebSiteName

Home

Page 1

Page 2

Page 3

Basic Navbar Example

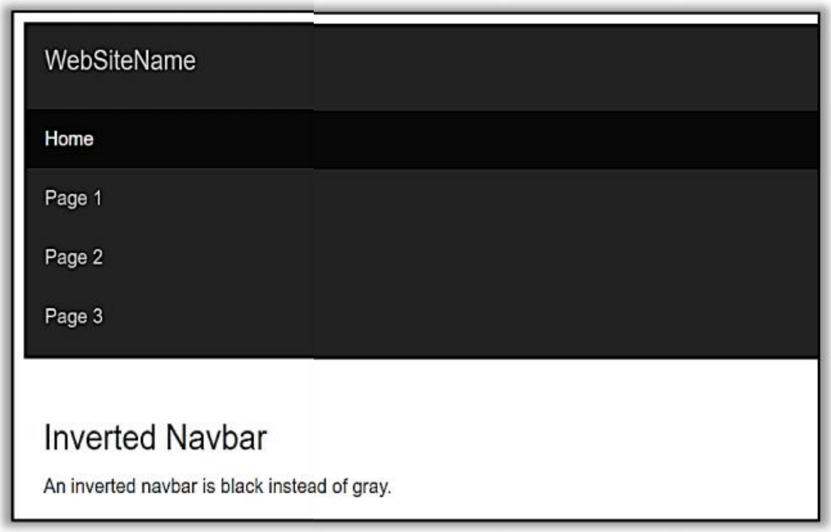
A navigation bar is a navigation header that is placed at the top of the page.

Creating a standard navigation bar at the top of the page with with:

<NAV CLASS="NAVBAR NAVBAR-DEFAULT">.

Bootstrap Inverted Navigation Bar

- Inverted navigation bar provides an alternative black navbar. It can be used to style the default navigation bar by changing. navbar**-default class** into. navbar**-inverse class**.



WebSiteName

Home

Page 1

Page 2

Page 3

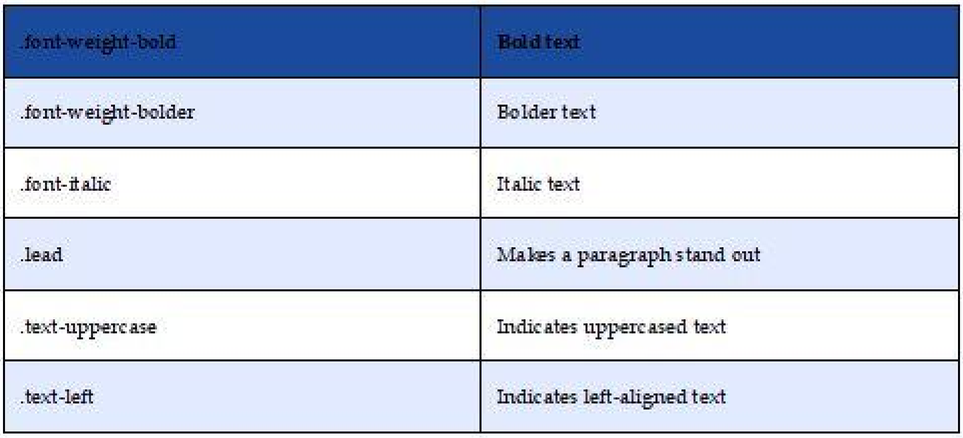
Inverted Navbar

An inverted navbar is black instead of gray.

6.4 Web Design,Grid System,Header Section

Typography

Bootstrap's global default font-size is 14px, with a line-height of 1.428.This is applied to the <body> element and all paragraphs (<p>).In addition, all <p> elements have a bottom margin that equals half their computed line-height (10px by default).



.font-weight-bold	Bold text
.font-weight-bolder	Bolder text
.font-italic	Italic text
.lead	Makes a paragraph stand out
.text-uppercase	Indicates uppercased text
.text-left	Indicates left-aligned text

Background Colors

The classes for background colors are

- bg-primary,
- .bg-success,
- .bg-info, .bg-warning,
- .bg-danger,
- .bg-secondary,
- .bg-dark and
- .bg-light.

The classes for text colors are:

- .text-muted, .text-primary,
- .text-success, .text-info,
- .text-warning, .text-danger,
- .text-secondary, .text-white, .text-dark, .text-body (default body color/often black) and .text-light

6.5 Bootstrap Forms

In Bootstrap, there are three types of form layouts:

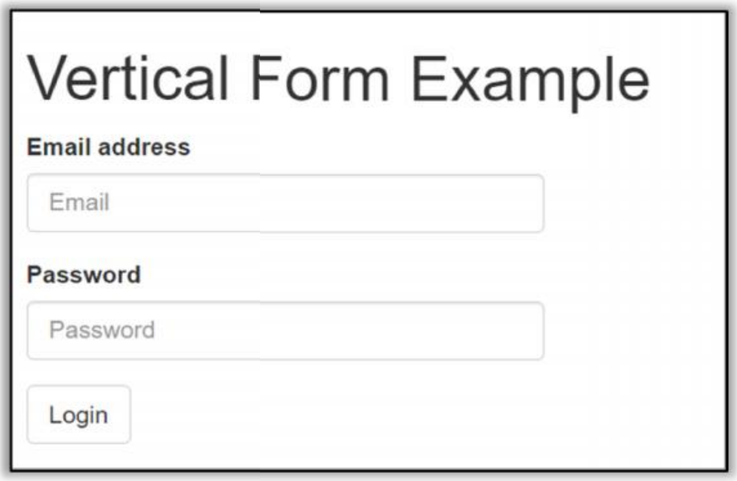
- Vertical form (this is default)
- Horizontal form
- Inline form

There are three standard rules for these 3 form layouts:

Always use <form role="form"> (helps improve accessibility for people using screen readers)

Wrap labels and form controls in <div class="form-group"> (needed for optimum spacing)

Add class .form-control to all textual <input>, <textarea>, and <select> elements



Vertical Form Example

Bootstrap Inline Form

In an inline form, all of the elements are inline, left-aligned, and the labels are alongside.

Note: This only applies to forms within viewports that are at least 768px wide!

Additional rule for an inline form:

Add class .form-inline to the <form> element

A horizontal form means that the labels are aligned next to the input field (horizontal) on large and medium screens. On small screens (767px and below), it will transform to a vertical form (labels are placed on top of each input).

Horizontal form

Additional rules for a horizontal form:

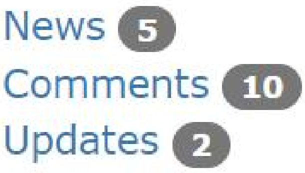
Add class. form-horizontal to the <form> element

Add class. control-label to all <label> elements

Tip: Use Bootstrap's predefined grid classes to align labels and groups of form controls in a horizontal layout.

Bootstrap Badges and Labels

Badges are numerical indicators of how many items are associated with a link:



The numbers (5, 10, and 2) are the badges.Use the .badge class within <span> elements to create badges:Badges can also be used inside other elements, such as buttons:

Labels

Labels are used to provide additional information about something:

Use the. label class, followed by one of the six contextual classes. label-default,. label-primary,. label-success, .label-info, .label-warning or .label-danger, within a <span> element to create a label:

6.6 Bootstrap Modals and Bootstrap Carousel

A carousel in Bootstrap 4 can be defined as a slideshow that is used for the purpose to slideshow through a series of content to show the audience the content and to enhance the website. A carousel can also be called as slideshow or image slider.

It can be regarded as one of the best ways of displaying the large number of contents inside a small space on a web page.

A carousel can be created in four main ways, which are

- Using slides only
- Using controls
- With the indicators
- With the captions

This carousel only contains slides. The class carousel-inner is used to add different slides to the carousel.In this, two little arrows will be added to the slides which can help a user to manually switch to any slide in the carousel.

.carousel-control-prev class -It inserts a left or previous button to the carousel, that will give access to the user to go back between the slides.

.carousel-control-next class - It inserts a right or next button to the carousel, that will give access to the user to go forward between the slides.

.carousel-control-prev-icon class - It can be used along with the .carousel-control-prev in order to create a previous button.

.carousel-control-next-icon class - It can be used along with the .carousel-control-next to create a "next" button

Indicators

The .carousel-indicators is the class that can be used to add indicators for the carousel. The indicators in this class are used to indicate how many slides are in the carousel, and which slide the user is currently viewing.

Captions

A user can also add captions to their slides in Bootstrap 4. A caption can be easily used to convey to the audience of the website that what exactly is happening in the carousel. For the purpose of adding captions, a user has to add elements inside <div class="carousel-caption"> within each <div class="carousel-item"> in order to create a caption for every slide.

6.7 Bootstrap 4 Modal

The Modal component is a dialog box/popup window that is displayed on top of the current page

Creating a model

<!-- The Modal -->

```
<div class="modal" id="myModal">
<div class="modal-dialog">
<div class="modal-content">
<!-- Modal Header -->
<div class="modal-header">
<h4 class="modal-title">Modal Heading</h4>
<button type="button" class="close" data-dismiss="modal">&times;</button>
</div>
<!-- Modal body -->
<div class="modal-body">
    Modal body..
</div>
<!-- Modal footer -->
<div class="modal-footer">
<button type="button" class="btnbtn-danger" data-dismiss="modal">Close</button>
</div>
```

Grid

Furthermore, Bootstrap provides a grid system, which creates a page layout through a series of rows and columns that can place the content. There is no Grid system in CSS.

Designing Classes

Also, Bootstrap contains already designed classes. The programmer can use them in his program to add styling to the elements without writing code from the beginning. On the other hand, in CSS, the programmer should write the code from scratch.

Considering the usage, the Bootstrap helps to create responsive designs that are more presentable while CSS helps to create presentable webpages. Bootstrap and CSS help to build presentable user interfaces. The basic difference between Bootstrap and CSS is that the Bootstrap is a front-end framework while CSS is a style sheet language. They help to add background and borders, text effects, 2D, 3D animations etc. to web pages.

Bootstrap Auto-layout Columns

Creating equal width columns for all devices (x-small, small, medium, large, x-large, and xx-large) through simply using the class .col, without specifying any column number.

```
<div class="container">
<!--Row with two equal columns-->
<div class="row">
<div class="col">Column one</div>
<div class="col">Column two</div></div>
<!--Row with three equal columns-->
<div class="row">
<div class="col">Column one</div>
<div class="col">Column two</div>
<div class="col">Column three</div>
</div>
```

</div>

Summary

- Bootstrap is a free front-end framework for faster and easier web development. It also gives you the ability to easily create responsive designs.
- Content delivery network (CDN) is a system of distributed servers (network) that deliver webpages and other Web content to a user based on the geographic locations of the user, the origin of the webpage and a content delivery server.
- A caption can be easily used to convey to the audience of the website that what exactly is happening in the carousel
- Bootstrap provides a grid system, which creates a page layout through a series of rows and columns that can place the content. There is no Grid system in CSS.
- Bootstrap's global default font-size is 14px, with a line-height of 1.428.This is applied to the <body> element and all paragraphs (<p>).
- Bootstrap 3 is designed to be responsive to mobile devices.Mobile-first styles are part of the core framework of Bootstrap.You have to add the following <meta> tag inside the <head> element for proper rendering and touch zooming:

Keywords

Bootstrap: Bootstrap is a free front-end framework for faster and easier web development. It also give you the ability to easily create responsive designs.

Carousel -A carousel in Bootstrap 4 can be defined as a slideshow that is used for the purpose to slideshow through a series of content to show the audience the content and to enhance the website

CSS- Bootstrap comes with the feature of global CSS settings, fundamental HTML elements styled and enhanced with extensible classes, and an advanced grid system

Grids- Bootstrap provides a grid system, which creates a page layout through a series of rows and columns that can place the content.

Containers- Containers are not nestable (you cannot put a container inside another container).

Bootstrap Inline: In an inline form, all of the elements are inline, left-aligned, and the labels are alongside

Navigation Bar- A navigation bar is like a navigation header that is placed at the top of the page. You can collapse or extend it according to the screen size

Inverted Navigation Bar: Inverted navigation bar provides an alternative black navbar. It can be used to style the default navigation bar by changing. navbar**-default class** into. navbar**-inverse class**.

Self Assessment

1. Which of the following class in Bootstrap is used to provide a responsive fixed width container?

A. .container-fixed
B. .container-fluid
C. .container
D. All of the above

2. Is Bootstrap3 mobile-first?

A. True
B. False

C. Can't say

D. May be

3. How many columns are allowed in a bootstrap grid system?

A. 2

B. 12

C. 3

D. 5

4. What are advantages of Bootstrap

A. Easy to use

B. Responsive

C. Mobile first

D. All of the above

5. What is full form of CDN

A. Content Direct Network

B. Content Delivery Network

C. Content Delivery Node

D. Create Delivery Network

6. Which of the following class in bootstrap is used to create a big box for calling extra attention?

A. .box

B. .container

C. .container-fluid

D. .jumbotron

7. The correct syntax of creating a standard navigation bar is -

A. <nav class="navigationbar navbar">

B. <nav class="navbar navbar-default">

C. <nav class="nav navbar">

D. <nav class="navbar default">

8. Which class in Bootstrap Grid System used for phones:

A. Xs

B. Sm

C. Md

D. Lg

9. Which class in Bootstrap Grid System used for tablets:

A. Xs

B. Sm

C. Md

D. Lg

10. Which class in Bootstrap Grid System used for desktops:

A. Xs

B. Sm

C. Md

D. Lg

11. Which class in Bootstrap Grid System used for larger desktops:

A. Xs

B. Sm

C. Md

D. Lg

12. Which is default form layout in Bootstrap

A. Vertical form

B. Horizontal form

C. Inline form

D. None of these

13. _______ are used to provide additional information about something

A. Labels

B. Badges

C. Headings

D. None of these

14. A __________can also be called as slideshow or image slider

A. Carousel

B. Labels

C. Badges

D. Headings

15. A carousel can be created using :

A. Using slides only

B. Using controls

C. With the indicators

D. All of the above

Answers for Self Assessment

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

Review Questions

1. Explain why Bootstrap is preferred for website development?
2. What is Bootstrap Grid System? Explain with an example?
3. Explain the typography and links in Bootstrap
4. What is media object in Bootstrap and what are their types

5. Explain the uses of carousel plugin in Bootstrap.
6. What are the bootstrap media objects?
7. What are Bootstrap panels? Explain how to create a Bootstrap panel with a heading
8. What are Bootstrap alerts?
9. What is navigation Bar?
10. What is Inverted Navigation Bar?

Further Readings

HTMl, JavaScript , DHTML and PHP By. Ivan Bayross.

Beginning JavaScript 2nd Edition By. Paul Wilton

Html 5 Black Book, Covers Css 3, Javascript, Xml, Xhtml, Ajax, PhpAndJquery, Second Edition

Web Links

https://www.javascripttutorial.net/javascript-function/

www.webopedia.com

www.web-source.net

Unit 07:Introduction to JavaScript

CONTENTS

Objectives

After studying this unit, you will be able to:

- Introduction to JavaScript
- Adjusting Development Environment for JavaScript Development
- Where to place JavaScript Code

Introduction

A lightweight programming language ("scripting language")used to make web pages interactiveinsert dynamic text into HTML (ex: user name), react to events (ex: page load user click), get information about a user's computer (ex: browser type), perform calculations on user's computer (ex: form validation).

JavaScript was first known as LiveScript, but Netscape changed its name to JavaScript, possibly because of the excitement being generated by Java. JavaScript made its first appearance in Netscape 2.0 in 1995 with the name LiveScript. The general-purpose core of the language has been embedded in Netscape, Internet Explorer, and other web browsers.

Applications of Javascript Programming

- **Manipulating HTML Pages** - Javascript helps in manipulating HTML page on the fly.
- Adding and deleting any HTML tag very easily using javascript and modify your HTML to change its look and feel based on different devices and requirements.
- **User Notifications** - You can use Javascript to raise dynamic pop-ups on the webpages to give different types of notifications to your website visitors

- **Back-end Data Loading** - Javascript provides Ajax library which helps in loading back-end data while you are doing some other processing. This really gives an amazing experience to your website visitors.
- **Presentations** - JavaScript also provides the facility of creating presentations which gives website look and feel. JavaScript provides RevealJS and BespokeJS libraries to build a web-based slide presentations.
- Client Side validation - This is really important to verify any user input before submitting it to the server and Javascript plays an important role in validting those inputs at front-end itself.

Example: To "make web pages alive," JavaScript was initially developed. Scripts are what the programmes in this language are known as. They can be directly written in the HTML of a web page and executed when the page loads. Plain text scripts are delivered and run. They can run without any additional setup or compilation.

7.1 Why to Learn JavaScript

Javascript is everywhere, it comes installed on every modern web browser and so to learn Javascript you really do not need any special environment setup. For example Chrome, Mozilla Firefox , Safari and every browser you know as of today, supports Javascript. Javascript helps you create really beautiful and crazy fast websites.You can develop your website with a console like look and feel and give your users the best Graphical User Experience.

Great thing about Javascript is that you will find tons of frameworks and Libraries already developed which can be used directly in your software development to reduce your time to market.

7.2 Applications of JavaScript Programming

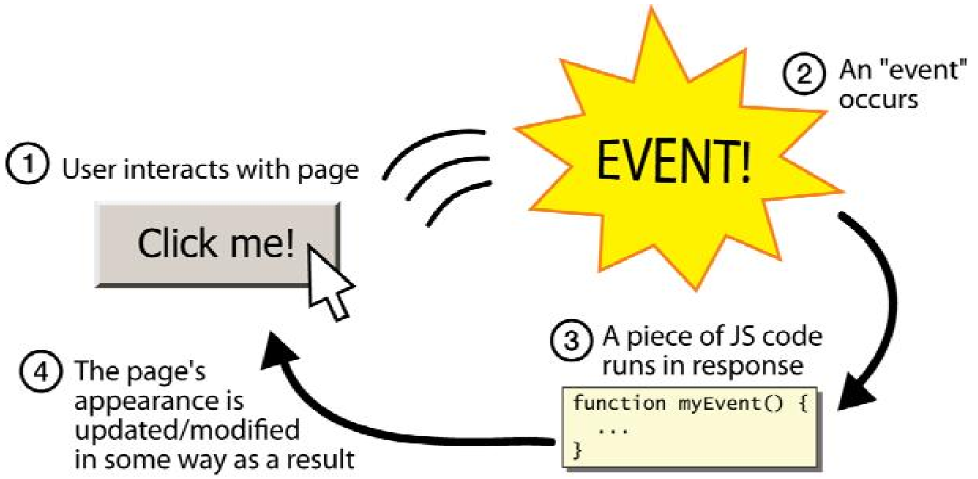
- Manipulating HTML Pages - Javascript helps in manipulating HTML page on the fly.
- This helps in adding and deleting any HTML tag very easily using javascript and modify your HTML to change its look and feel based on different devices and requirements.
- User Notifications - You can use Javascript to raise dynamic pop-ups on the webpages to give different types of notifications to your website visitors.
- Back-end Data Loading - Javascript provides Ajax library which helps in loading back-end data while you are doing some other processing. This really gives an amazing experience to your website visitors.
- Presentations - JavaScript also provides the facility of creating presentations which gives website look and feel. JavaScript provides RevealJS and BespokeJS libraries to build a web-based slide presentation.
- Client side validation - This is really important to verify any user input before submitting it to the server and Javascript plays an important role in validting those inputs at front-end itself.

7.3 Linking JavaScript File

JavaScript file can be linked by script tag that must be placed in HTML page's head. Code is stored in a separate .js file.JS code can be placed directly in the HTML file's body or head (like CSS), but this is bad style (should separate content, presentation, and behavior).

```
<script src="filename" type="text/javascript"></script>
```

Event-driven Programming


① User interacts with page
Click me!
② An "event" occurs
EVENT!
③ A piece of JS code runs in response
function myEvent() {
...
}
④ The page's appearance is updated/modified in some way as a result

Split breaks apart a string into an array using a delimiter. It can also be used with regular expressions (seen later). Join merges an array into a single string, placing a delimiter between them

JavaScript Frameworks and Libraries

There are many useful Javascript frameworks and libraries available:

- Angular
- React
- jQuery
- Vue.js
- Ext.js

JavaScript can be implemented using JavaScript statements that are placed within the <script>... </script> HTML tags in a web page.You can place the <script> tags, containing your JavaScript, anywhere within your web page, but it is normally recommended that you should keep it within the <head> tags. The <script> tag alerts the browser program to start interpreting all the text between these tags as a script

JavaScript syntax

A simple syntax of your JavaScript will appear as follows.

```
<script ...>
  JavaScript code
</script>
```

Script tag

The script tag takes two important attributes −

> **Language − This attribute specifies what scripting language you are using. Typically, its value will be javaScript. Although recent versions of HTML (and XHTML, its successor) have phased out the use of this attribute.**
>
> **Type − This attribute is what is now recommended to indicate the scripting language in use and its value should be set to "text/javaScript".**

Script Tag

```
<script language = "javaScript" type = "text/javaScript">
  JavaScript code
</script>
```

First JavaScript Code

Let us take a sample example to print out "Hello World"

```
. <html>
<body>
<script language = "javascript" type = "text/javascript">
document.write("Hello World!")
</script>
</body>
</html>
```

7.4 Advantages of JavaScript

- Speed. Client-side JavaScript is very fast because it can be run immediately within the client-side browser. Unless outside resources are required, JavaScript is unhindered by network calls to a backend server.
- Simplicity. JavaScript is relatively simple to learn and implement.
- Popularity. JavaScript is used everywhere on the web.
- Interoperability. JavaScript plays nicely with other languages and can be used in a huge variety of applications.
- Server Load. Being client-side reduces the demand on the website server.
- Gives the ability to create rich interfaces.

JavaScript offers procedural programming features

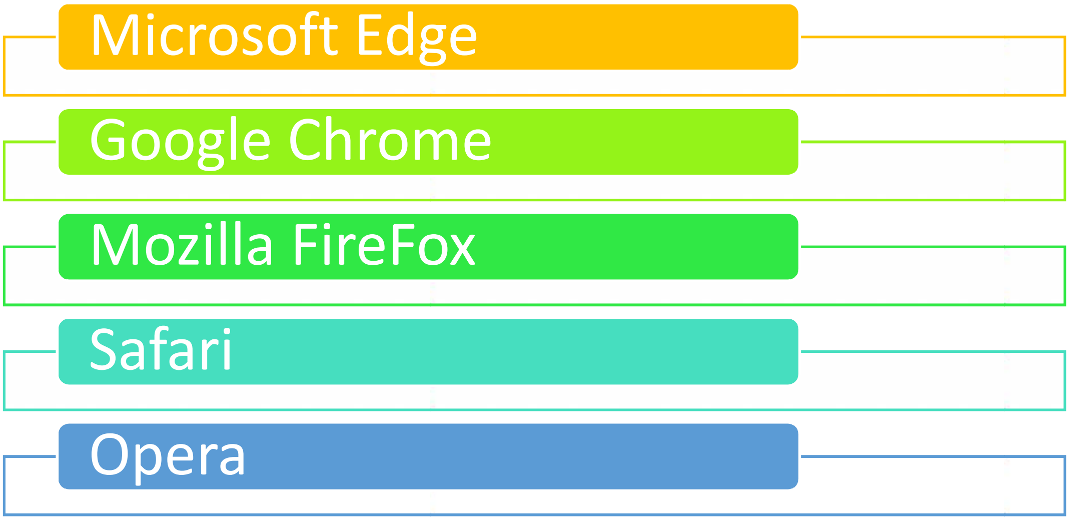
Even though the language is simple to discover, it offers all the procedure-based features that make it a well-liked and influential programming language. With JavaScript, if you have options to generate branches, initiate conditional checking, loops and much more, which will make your website much more exciting t to use.

Setup JavaScript Development Environment

To use JavaScript in developing a web application you must have at least two things, a browser, and an editor to write the JavaScript code.

Browser

You can also install the following browser as per your preference:



Microsoft Edge

Google Chrome

Mozilla FireFox

Safari

Opera

IDEs for JavaScript Application Development

You can write JavaScript code using a simple editor like Notepad. However, you can install any open-sourced or licensed IDE (Integrated Development Environment) to get the advantage of IntelliSense support for JavaScript and syntax error/warning highlighter for rapid development.

JavaScript Editors:

The followings are some of the well-known JavaScript editors:

- Visual Studio Code (Free, cross-platform)
- Eclipse (Free, cross-platform)
- Atom (Free, cross-platform)
- Notepad++ (Free, Windows)
- Code Lobster (Free, cross-platform)
- WebStorm (Paid, cross-platform)

Where can I download JavaScript?

- You don't have to get or download JavaScript.
- JavaScript is already running in your browser on your computer, on your tablet, and on your smart-phone.
- JavaScript is free to use for everyone.

7.5 JavaScript importance

JavaScript is one of the 3 languages all web developers must learn:

1. HTML to define the content of web pages
2. CSS to specify the layout of web pages
3. JavaScript to program the behavior of web pages

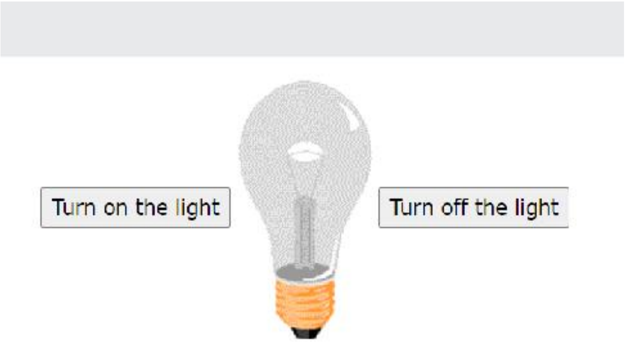
JavaScript Can Change HTML Content

- One of many JavaScript HTML methods is getElementById().

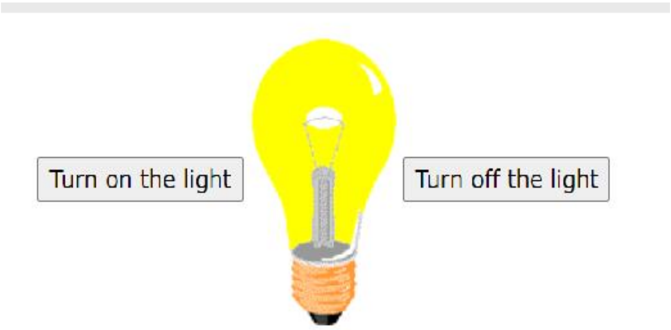
- The example below "finds" an HTML element (with id="demo"), and changes the element content (innerHTML) to "Hello JavaScript":
- Example
- document.getElementById("demo").innerHTML = "Hello JavaScript";

JavaScript Can Change HTML
Attribute Values

JavaScript changes the value of the src (source) attribute of an <img> tag



JavaScript changes the value of the src (source) attribute of an <img> tag



Changing the style of an HTML element, is a variant of changing an HTML attribute:

Example

document.getElementById("demo").style.fontSize = "35px";

Hiding HTML elements can be done by changing the display style:

Example

document.getElementById("demo").style.display = "none";

7.6 The <script> Tag

In HTML, JavaScript code is inserted between <script> and </script> tags.

Example

```
<script>
document.getElementById("demo").innerHTML = "My First JavaScript";
</script>
```

JavaScript Functions and Events

A JavaScript function is a block of JavaScript code, that can be executed when "called" for.

For example, a function can be called when an event occurs, like when the user clicks a button.

JavaScript in <head> or <body>

- You can place any number of scripts in an HTML document.

- Scripts can be placed in the <body>, or in the <head> section of an HTML page, or in both.
- In following example (next slide), a JavaScript function is placed in the <head> section of an HTML page.
- The function is invoked (called) when a button is clicked:

```
<!DOCTYPE html>
<html>
<head>
<script>
function myFunction() {
  document.getElementById("demo").innerHTML = "Paragraph changed.";
}
</script>
</head>
<body><h2>Demo JavaScript in Head</h2>

<p id="demo">A Paragraph</p>
<button type="button" onclick="myFunction()">Try it</button>
</body>
</html>
```

JavaScript in <body>

- In this example(next slide), a JavaScript function is placed in the <body> section of an HTML page.
- The function is invoked (called) when a button is clicked:

```
<!DOCTYPE html>
<html>
<body>

<h2>Demo JavaScript in Body</h2>

<p id="demo">A Paragraph</p>

<button type="button" onclick="myFunction()">Try it</button>

<script>
function myFunction() {
  document.getElementById("demo").innerHTML = "Paragraph changed.";
}
</script>

</body>
</html>
```

7.7 External JavaScript

- External scripts are practical when the same code is used in many different web pages.
- JavaScript files have the file extension .js.
- To use an external script, put the name of the script file in the src (source) attribute of a <script> tag:

Example:

```
<script src="myScript.js"></script>
```

You can place an external script reference in <head> or <body> as you like.The script will behave as if it was located exactly where the <script> tag is located.

External JavaScript Advantages

Placing scripts in external files has some advantages:

- It separates HTML and code
- It makes HTML and JavaScript easier to read and maintain
- Cached JavaScript files can speed up page loads

JavaScript Output

- JavaScript Display Possibilities
- JavaScript can "display" data in different ways:
  - Writing into an HTML element, using innerHTML.
  - Writing into the HTML output using document.write().
  - Writing into an alert box, using window.alert().
  - Writing into the browser console, using console.log().

Using innerHTML

- To access an HTML element, JavaScript can use the document.getElementById(id) method.
- The id attribute defines the HTML element. The innerHTML property defines the HTML content

Using document.write()

```
<!DOCTYPE html>
<html>
<body>

<h1>My First Web Page</h1>
<p>My first paragraph.</p>

<script>
document.write(5 + 6);
</script>

</body>
</html>
```

Using window.alert()

```
<!DOCTYPE html>
<html>
<body>

<h1>My First Web Page</h1>
<p>My first paragraph.</p>

<script>
window.alert(5 + 6);
</script>

</body>
</html>
```

JavaScript Comments

- JavaScript comments can be used to explain JavaScript code, and to make it more readable.

- JavaScript comments can also be used to prevent execution, when testing alternative code.

Single Line Comments

- Single line comments start with //.
- Any text between // and the end of the line will be ignored by JavaScript (will not be executed).

Multi-line Comments

- Multi-line comments start with /* and end with */.
- Any text between /* and */ will be ignored by JavaScript.

Summary

- A lightweight programming language ("scripting language")used to make web pages interactiveinsert dynamic text into HTML.
- Scripts are what the programme 's in this language are known as. They can be directly written in the HTML of a web page and executed when the page loads.
- A JavaScript function is a block of JavaScript code, that can be executed when "called" for
- Placing scripts in external files has some advantages: It separates HTML and code
- Split breaks apart a string into an array using a delimiter. It can also be used with regular expressions (seen later). Join merges an array into a single string, placing a delimiter between them
- JavaScript comments can be used to explain JavaScript code, and to make it more readable.
- External scripts are practical when the same code is used in many different web pages.JavaScript files have the file extension .js.
- One of many JavaScript HTML methods is getElementById().

Keywords

Javascript: Bootstrap is a free front-end framework for faster and easier web development. It also give you the ability to easily create responsive designs.

JavaScript Developer- To use JavaScript in developing a web application you must have at least two things, a browser, and an editor to write the JavaScript code.

External scripts- The are practical when the same code is used in many different web pages.JavaScript files have the file extension .js.

Inner HTML- To access an HTML element, JavaScript can use the document.getElementById(id) method.

JavaScript Function-A JavaScript function is a block of JavaScript code, that can be executed when "called" for

IDEs for JavaScript- JavaScript code using a simple editor like Notepad. However, you can install any open-sourced or licensed IDE (Integrated Development Environment)

Self Assessment

1. JavaScript is an important________ scripting language and widely used in dynamic websites.

A. client-side

B. Server side

C. Both of above

D. None of above

2. It is a technique used in web development in which scripts run on the client's browser

A. client-side scripting

B. Server side

C. Both of above

D. None of above

3. It is a technique that uses scripts on the webserver to produce a response that is customized for each client's request.

A. client-side scripting

B. Server side Scripting

C. Both of above

D. None of above

4. In which element we put javascript?

A. <javascript>

B. <scripting>

C. <script>

D. None of These

5. How we write "Hello World" in an alert box?

A. alertBox("Hello World");

B. msgBox("Hello World");

C. msg("Hello World");

D. alert("Hello World");

6. What are advantages of JavaScript

A. Speed

B. Interoperability

C. Simplicity

D. All of the above

7. Which type of JavaScript language is _____?

A. Object-oriented

B. Object-based

C. Functional programming

D. All of the above

8. Which symbol is used separate JavaScript statements?

A. Comma (,)

B. Colon (:)

C. Hyphen (_)

D. Semicolon (;)

9. JavaScript ignores?

A. newlines

B. tabs

C. spaces

D. All of the above

10. Which JavaScript method is used to access an HTML element by id?

A. getElementById()

B. getElement(id)

C. getElementById(id)

D. elementById(id)

11. Which property is used to define the HTML content to an HTML element with a specific id?

A. innerText

B. innerContent

C. elementText

D. innerHTML

12. Which JavaScript method is used to write HTML output?

A. document.write()

B. document.output()

C. console.log()

D. document.writeHTML()

13. Which JavaScript method is used to write on browser's console?

A. console.write()

B. console.output()

C. console.log()

D. console.writeHTML()

14. How many keywords are there in JavaScript to declare variables or constants?

A. 1

B. 2

C. 3

D. 4

15. What is the main difference between var and let keywords in JavaScript?

A. var defines a variable while let defines a constant

B. var defined function scoped variable while let define block scoped variable

C. The value of a variable declared with var can be changed while the value of a variable declared with let cannot be changed

D. All of the above

Answers for Self Assessment

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

Review Questions

1. Explain JavaScript? Explain its application in HTMl?

2. Which IDE can be used implementing JavaScript?
3. What is equality in JavaScript
4. What's the difference between Host objects and Native objects?
5. How to compare two objects in JavaScript?
6. How JavaScript can be used in innerHTML?
7. Explain Prototype Inheritance in JavaScript?
8. What is the new keyword in JavaScript?
9. Is JavaScript a pass-by-reference or pass-by-value language?
10. Why Script Tag is Used in JavaScript?

Further Readings

HTMl, JavaScript , DHTML and PHP By. Ivan Bayross.

Beginning JavaScript 2nd Edition By. Paul Wilton

Html 5 Black Book, Covers Css 3, Javascript, Xml, Xhtml, Ajax, PhpAndJquery, Second Edition

Web Links

https://www.javascripttutorial.net/javascript-function/

www.webopedia.com

www.web-source.net

Unit 08: Variables and Functions

CONTENTS

Objectives

After studying this unit, you will be able to:

- Discuss introduction regarding Variables
- Explain Function and Scope
- What are JavaScript Types and Common Language Constructs
- Handling of default value in JavaScript

Introduction

To workwith JavaScript as other programming languages, JavaScript has variables for the storage of some value. Variables can be assumed of as named containers. You can place data into these containers and then refer to the data simply by naming the container.

Before we use a variable in a JavaScript program, it must be declared with the var keyword as follows.

```
<script type = "text/javascript">
<!--
   var price;
   var name;
 //-->
</script>
```

You can also declare multiple variables with the same var keyword as follows −

```
<script type = "text/javascript">
```

```
<!--
    var price,name;
  //-->
</script>
```

Storingor placing a value in a variable is called variable initialization.Variable initialization at the time of variable creation or when you need that variable.

For instance, we might create a variable named price and assign the value 2000 to it later. For another variable, you can assign a value at the time of initialization as follows.

```
<script type = "text/javascript">
<!--
    var name = "hardisk";
    var price;
price = 2000;
  //-->
</script>
```

Example: The var keyword is used only for declaration or initialization, only once variable name is declared, and variable name cannot be same in document.

JavaScript is an untyped language which means JavaScript variable can hold a value of any data type. Unlike many other languages, you don't need to tell JavaScript during variable declaration what type of value the variable will hold. The value type of a variable can change during the execution of a program and JavaScript takes care of it automatically.

JavaScript Variable Scope

The scope of a variable is the section of a program in which it is defined. JavaScript variables have only two scopes.

Global Variables − A global variable has global scope which means it can be defined anywhere in your JavaScript code.

Local Variables − A local variable will be visible only within a function where it is defined. Function parameters are always local to that function.

Within the body of a function, a local variable takes priority over a global variable with the same name. If we have declared a local variable or function parameter with the same name as a global variable, we effectively abstract the global variable.For example.

```
<html>
<body onload = vscope();>
<script type = "text/javascript">
<!--
        var myVar = "global";     // Declare a global variable
        function vscope( ) {
          var myVar = "local";    // Declare a local variable
document.write(myVar);
        }
      //-->
```

```
</script>
</body>
</html>
```

This produces the following result −

```
local
```

JavaScript Variable Names

While naming our variables in JavaScript, following rules must be kept in mind.

wemust not use any typeof JavaScript reserved keywords as a variable name. These keywords are mentioned in the table below. For example, break or boolean variable names are not valid.

JavaScript variable names should not start with a numeral (0-9). They must begin with a letter or an underscore character. For example, 12test is an invalid variable name but _12test is a valid one.

JavaScript variable names are case-sensitive. For example, Name and name are two different variables.

JavaScript Reserved Words

8.1 Keywords used in JavaScript

abstract	else	instanceof	switch
boolean	enum	int	synchronized
break	export	interface	this
byte	extends	long	throw
case	false	native	throws
catch	final	new	transient
char	finally	null	true
class	float	package	try
const	for	private	typeof
continue	function	protected	var
debugger	goto	public	void
default	if	return	volatile

delete	implements	short	while
do	import	static	with
double	in	super	

8.2 Functions and Scope

Function is a set of reusable code which may be called anywhere in your program. This removes the need of writing the identical code multiple times. It allows programmers in writing modular codes. Functions permit a programmer to divide a massive software into some small and manageable functions. Like every other superior programming language, JavaScript additionally helps all of the functions essential to jot down modular code with the usage of functions. Some more functions like alert() and write() are also used. We are using functions; again and again, however, they were written in core JavaScript only once. JavaScript lets us to write our own functions as well. The section shows how to write our known JavaScript function.

Before using a function, we need to define it. The most common way to define a function in JavaScript is by using the function keyword, followed by a unique function name, a list of parameters (that might be empty), and a statement block surrounded by curly braces.

Basic syntax of function:

```
<scripttype="text/javascript">
<!--
functionfunctionname(parameter-list)
{
  statements
}
//-->
</script>
```

For example. It defines a function called Helloworld that takes no parameters:

```
<scripttype="text/javascript">
<!--
functionstart()
{
   alert("HelloWorld");
}
//-->
</script>
```

8.3 Calling a Function

To invoke a function we would simply need to write the name of that function as shown in the following code.

```
<html>
```

```
<head>
<scripttype="text/javasc
ript">function test()
{
  document.write("Hello World!");
}
</script>
</head>

<body>
<p>Clicktocallthefunction</p>
<form>
<inputtype="button"onclick="test()"value="Hello World">
</form>

<p>Use some different text and try again...</p>
</body>
</html>
```

```
Output:
Clicktocallthefunction
```

8.4 Function Parameters

As we have seen in the above section functions without parameters. But there is an ability to pass different parameters while calling a function. These passed parameters can be captured inside the function and any manipulation can be done over those parameters. A function can take multiple parameters separated by a comma.

For example we have modified our test function here. Now it takes two parameters.

```
<html>
<head>
<scripttype="text/javascript">fu
nctiontest(name,age)
{
      document.write(name+"is"+age+"yearsold.");
}
</script>
```

```
</head>

<body>
<p>Clicktocallthefunction</p>
<form>
<inputtype="button"onclick="test('Aseem',7)"value="Hello world">
</form>

<p>Usedifferentparametersinsidethefunctionandthentry...</p>
</body>
</html>
```

Output

Click to call the function

Use different parameters inside the function and then try...

8.5 The return Statement

A JavaScript function can have an optional return statement. This is required if you want to return a value from a function. This statement should be the last statement in a function.

For example, you can pass two numbers in a function, and then you can expect the function to return their multiplication in your calling program.

```
<html>
<head>
<script
type="text/javascript">functioncon
catenate(first,last)
{
    varfull;

    full=first+last;re
    turnfull;
}
functionsecondFunction()
{
    varresult;
    result=concatenate('Zara','Ali');document.write(re
        sult);
 }
 </script>
 </head>
 <body>
 <p>Clickthefollowingbuttontocallthefunction</p>
 <form>
 <inputtype="button"onclick="secondFunction()"value="CallFunction">
 </form>

 <p>Usedifferentparametersinsidethefunctionandthentry...</p>
 </body>
 </html>
```

Click the following button to call the function

Call Function

Use different parameters inside the function and then try...

8.6 JavaScript Types

JavaScript provides different data types to hold different types of values. There are two types of data types in JavaScript.

- Primitive data type
- Non-primitive (reference) data type

JavaScript is a dynamic type language; means there is no need to specify type of the variable because it is dynamically used by JavaScript engine. Only the need is to use var here to specify the data type. It can hold any type of values such as numbers, strings etc. For example:

var a=20;//holding number

var b="Aseem Khana";//holding string

JavaScript primitive data types

There are five types of primitive data types in JavaScript. They are as follows:

Data Type	Description
String	represents sequence of characters e.g. "aseem"
Number	represents numeric values e.g. 50
Boolean	represents boolean value either false or true

Undefined	represents undefined value
Null	represents null i.e. no value at all

JavaScript non-primitive data types

The non-primitive data types are as follows:

Data Type	Description
Object	represents instance through which we can access members
Array	represents group of similar values
RegExp	represents regular expression

8.7 JavaScript Default Parameters

Default function parameters are the parameters in JavaScript that provide a default value while declaring the JavaScript function. This is useful, as we can provide default values to the parameters, which can be used if a value is not provided when the function is called.

So default function parameters are used to initialize the named parameter with a default value if no value or undefined is passed during the function call.

For example, if we define a function in which we provide two variables, it divides the first parameter with the second parameter and returns the result of the division operator. Now if the user provides only one parameter, then the function call will fail. For avoiding this, you can provide a default value to the second parameter as 1, which will divide the single parameter passed during the function call with the default value 1, rather than giving an error.

By default, if a function parameter is not provided with a value during the function call, undefined is set in it.

JavaScript Default Parameters Syntax:

```
function function_name([param1[ = defaultValue1][, ..., paramS[ = defaultValueS ]]])
{
  statements
}
```

One important point to note that if you have multiple parameters in a function, say 2, and you provide a default value to the first one and leave the second parameter without a default value, then during the call, if that function is called with a single argument, it will get assigned to first parameter(one with default parameter value) and the second will get the undefined value, this is because parameter assignment happens from left to right. So you should not have parameters without default values after default parameters.

```
function printValue(a=2, b) {
  console.log("a = " + a + " and b = " + b);
}

printValue();    // Logs: a = 2 and b = undefined
printValue(8);    // Logs: a = 8 and b = undefined
printValue(8, 5);    // Logs: a = 8 and b = 5
```

The above, when we provided no argument while calling the function then, a used its default value while b got undefined. So provided one argument while calling the function, and that argument got assigned to the first parameter which is a, as argument assignment to parameters happens in order, hence b again got undefined.

Default values for parameters and calling it without arguments are explained in below example.

```
// default function parameters

function add(a=10, b=20)
{
        return a+b;
}

console.log(" Sum is : " + add());   // No argument
console.log(" Sum is : " + add(1));   // with one argument
console.log(" Sum is : " + add(5,6));   // with both argument
```

Output

```
Sum is : 30
Sum is : 21
Sum is : 11
```

Summary

- JavaScript supports a number of types of data, Such as numbers, text, and Booleans.
- Variables are JavaScripts's way of storing data, such as numbers and text, in memory so that they can be used again and again in your code.
- Variable names must not include certain illegal characters, like % and &, or be reserved word, like with.
- Use a function keyword to declare a function.
- FunctionName() is used to call a function.
- All functions implicitly return undefined if they don't explicitly return a value.
- the return statement to return a value from a function explicitly can be used.
- The arguments variable is an array-like object inside a function, representing function arguments.
- The function hoisting allows you to call a function before declaring it.

Keywords

Variables: Variables can be assumed as named containers. You can place data into these containers and then refer to the data simply by naming the container.

Functions: A function is a set of reusable codesthat may be called anywhere in your program. This removes the need of writing the identical code multiple times.

JavaScript Datatype: JavaScript is a dynamic type language; means there is no need to specify type of the variable because it is dynamically used by JavaScript engine.

Number: Number variables are just that—numbers. JavaScript does not require numbers to be declared as integers, floating-point (decimal) numbers, or any other number type.

Strings: String variables are variables that represent a string of text. The string may contain letters, words, spaces, numbers, symbols, or most anything you like

Parameters: Parameters are used to allow a function to import one or more values from somewhere outside the function

Global Variables: They are defined outside any functions; they can be changed anywhere in the script—inside or outside of functions

Local Variables:A local variable can be used only within the function in which it is declared. It does not exist outside that function, unless you pass it along to another function by using a parameter.

Self Assessment

1. The const keyword is used to define a ______.

A. Function scopes variable
B. Block scoped variable
C. Constant
D. Constant with no initial value

2. Which is the correct syntax to declare a constant in JavaScript?

A. const constant_name;
B. constant_name const;
C. constant_name const = value;
D. const constant_name = value;

3. What will be the value of VALUE?

```
<script>
    const VALUE = 10;
    VALUE = 20;
</script>
```

A. 10
B. 20
C. ValueError
D. TypeError

4. What is the default value of an uninitialized variable?

A. 0
B. undefined
C. null
D. NaN

5. Can be redeclare a variable that is declared with var keyword?

A. Yes
B. No
C. Both A and B
D. None of above

6. ___________are containers for storing data
A. Variables
B. Arrays
C. Functions
D. All of the above
7. JavaScript arrays are written with _____.
A. round brackets ()
B. curly brackets {}
C. double quotes ""
D. square brackets []

8. JavaScript objects are written with _____.

A. round brackets ()

B. curly brackets {}

C. double quotes ""

D. square brackets []

9. Which JavaScript operator is used to determine the type of a variable?

A. typeof

B. sizeOf

C. typOf

D. sizeof

10. Which keyword is used to define a JavaScript function?
A. module
B. fun
C. func
D. function

11. Can we use a function as a variable value?
A. True
B. False

12. In JavaScript a variable contains one value while an object may contain ___.
A. One value
B. Two values
C. Three values
D. Many values

13. Which is the correct syntax to access an object property in JavaScript?
A. objectName:propertyName
B. propertyName
C. objectName["propertyName"
D. Both B. and C.

14. Which property is used to get the length of a string in JavaScript?
A. strlen
B. len
C. length
D. Lent

15. Which character is used to break up a code line within a text string in JavaScript?

A. Single quote (')

B. Single backslash (\)

C. Double quote (")

D. Tipple single quote (''')

Answers for Self Assessment

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

Review Questions

1. Write a JavaScript code block using arrays and generate the current date in words, this should include the day, the month and the year. The output should be as follows, Saturday, October 09,2021.
2. What if I put a function into my script but decide not to call it in the script? Will it matter?
3. What happens if I decide to remove a function from my script later?
4. What is the best way to determine when to use a function and when to just code what I want right into the script?
5. Write a JavaScript function which accepts an argument and returns the type

   Note : There are six possible values that typeof returns: object, boolean, function, number, string, and undefined.

Further Readings

HTMl, JavaScript , DHTML and PHP By. Ivan Bayross.

Beginning JavaScript 2nd Edition By. Paul Wilton

Html 5 Black Book, Covers Css 3, Javascript, Xml, Xhtml, Ajax, Php And Jquery, Second Edition

Web Links

https://www.javascripttutorial.net/javascript-function/

www.webopedia.com

www.web-source.net

Aseem Khanna, Lovely Professional University

Unit 09: Introduction to JavaScript Using Objects and Functions

CONTENTS

Objectives

After studying this unit, you will be able to:

- Discuss introduction regarding Objects
- Explain Objects Using 'new Object()'
- What are Literals and the 'this' keyword
- Passing variables and Value vs. by Reference

Introduction

JavaScript is an Object-Oriented Programming (OOP) language In JavaScript, an object is a standalone entity, with properties and type. Compare it with a cup, for example. A cup is an object, with properties. A cup has a color, a design, weight, a material it is made of, etc. In the same way, JavaScript objects can have properties, which define their characteristics. As a programming language it must provide the following features to become anObject-Oriented Language:

Encapsulation: the capability to store related information, whether data or methods, together in an object.

Aggregation: the capability to store one object inside another object.

Inheritance: the capability of a class to rely upon another class (or the number of classes) for some of its properties and methods.

Polymorphism: the capability to write one function or method that works in a variety of different ways.

Objects are composed of attributes. If an attribute contains a function, it isconsidered to be a method of the object, otherwise, the attribute is considered aproperty

In JavaScript, objects are king. If you understand objects, you understand JavaScript. In JavaScript, almost "everything" is an object. Booleans can be objects (if defined with the new keyword), Numbers can be objects (if defined with the new keyword), Strings can be objects (if defined with the new keyword), Dates are always objects, Maths are always objects, Regular expressions are always objects.

JavaScript Primitives

A primitive value is a value that has no properties or methods and a primitive data type is a data that has a primitive value.
JavaScript defines 5 types of primitive data types:

- string
- number
- boolean
- null
- undefined

Primitive values are immutable (they are hardcoded and therefore cannot be changed). For e.g. If x = 3.14, you can change the value of x. But you cannot change the value of 3.14.

Object Properties

Object properties can be any of the three primitive data types, or any of the abstract data types, such as another object. Object properties are typically variables that are used internally in the object's methods but can also be globally visible variables that are used throughout the page.

```
objectName.objectProperty=propertyValue;
```

This code gets the document title using the "title" property of the document object.

```
varstr=document.title;
```

Objects written as name value pairs are similar to:

- Associative arrays in PHP
- Dictionaries in Python
- Hash tables in C
- Hash maps in Java
- Hashes in Ruby and Perl

Object Method

Methods are actions that can be performed on objects. Object properties can be both primitive values, other objects, and functions. An object method is an object property containing a function definition.

There is a small difference between a function and a method – at a function is a standalone unit of statements and a method is attached to an object and can be referenced by the this keyword

Methods are useful for everything from displaying the contents of the object to the screen to performing complex mathematical operations on a group of local properties and parameters.

For example: The following is a simple example to show how to use the

write() method of the document object to write any content on the document.

```
document.write("Hello World");
```

9.1 Creating Objects Using 'new Object()' Syntax

The new operator is used to create an instance of an object. For creation of an object, the new operator is followed by the constructor method.

Some examples are here that shows a constructor methods as Object(), Array(), and Date(). These constructors are built-in JavaScript functions.

The syntax for adding a property to an object is:

```
var employee = new Object();

var books = new Array("Delhi", "Mumbai", "Kolkata");

var day = new Date("August 13, 1984");
```

The Object() Constructor

A constructor is a function that creates and initializes an object. JavaScript offers a special constructor function called Object() to build the object and the return value of the Object() constructor is assigned to a variable.

The variable contains a reference to the new object. The properties assigned to the object are not variables and are not defined with the var keyword.

Example to demonstrate how object is created

```
<html>

<head>

<title>Objects Defined by User</title>

<script type="text/javascript">

var test = new Object();      // Create the object

test.subject = "DBMS"; // Assigning  properties to the object

test.teacher = "Aseem Khanna";

</script>

</head>

<body>

<script type="text/javascript">

document.write("Test name is : " + test.subject + "<br>"); document.write("Test teacher is : " + test.teacher + "<br>");

</script>

</body>

</html>
```

Output:

```
Test nameis: DBMS
Test teacheris:Aseem Khanna
```

Defining Methods for an objects

```
<html>

<head>

<title> Objects Defined by User </title>

<script type="text/javascript">

// Definition of function that will work as a method

 function addtest(marks)

{

this.test = marks;

}
```

```
function student(sname, course )
{
this.sname = sname;
this.course = course;
this.addtest = addtest; // Assign that method as property.
}
</script>
</head>
<body>
  <scripttype="text/javascript">
         varmyclass=new class("BCA","Ashraf");myclass.addtest(sst);
         document.write("Student Name is : " + myclass.sname +
         "<br>");document.write("Student
         Courseis:"+myclass.course+"<br>");document.write("Student test
         courseis:"+myclass.test+"<br>");
  </script>
  </body>
  </html>
```

OUTPUT:

Student Name is: Aseem Khanna

Student Course is:BCA

Student Test Course is: DBMS

A constructor is a function that creates and initializes an object. JavaScript offers a special constructor function called Object() to build the object and the return value of the Object() constructor is assigned to a variable.

The 'with' Keyword

'with' Keyword represents as shorthand for referencing an Objects properties or methods.It specifies as an argument by which the default object for the duration of the block which follows. Without naming an object it can be used for that object properties and methods.

Syntax

```
  with(object){
          propertiesusedwithouttheobjectnameanddot
  }
```

Example showing 'With' Keyword

```
<html>
```

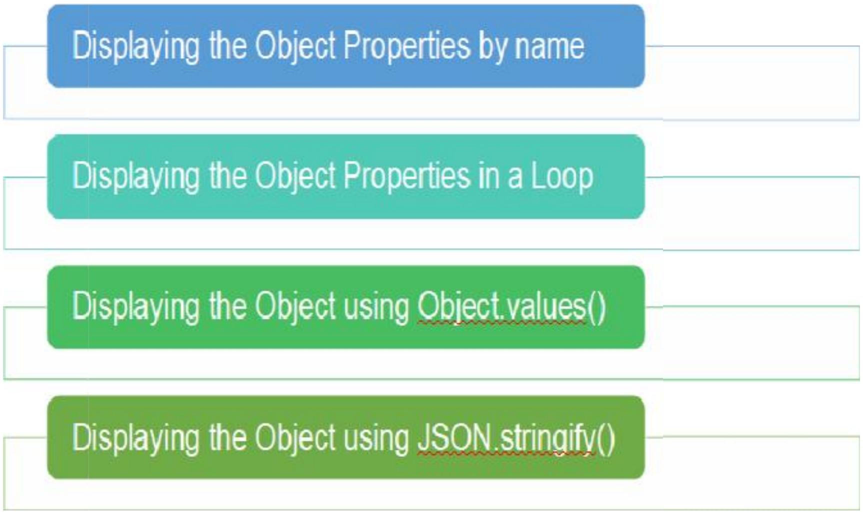
```
<head>
<title>User-defined objects</title>
<script type="text/javascript">
// Definition of a function which will work as a method function addPrice(amount){
with(this){
price = amount;
}
}
function Car(model, manufacturer)
{
this.model = model;
this.manufacturer  = manufacturer;.this.price = 0;
this.addPrice = addPrice; // Assign that method as property.
}
</script>
</head>
<body>
<script type="text/javascript">
var mycar = new car("Thar", "Mahindra"); mycar.addPrice(1200000);
document.write("Car model is : " + mycar.model + "<br>");
document.write("Car Manufacturer  is : " + mycar.manufacturer + "<br>");
document.write("Car Price is : " + mycar.price + "<br>");
</script>
</body>
</html>
```

Output:

```
Car modelis:Thar
Car Manufactureris: Mahindra
Car Price is: 1200000
```

9.2 To Display JavaScript Objects

Some common solutions to display JavaScript objects are:



Displaying the Object Properties by name

Displaying the Object Properties in a Loop

Displaying the Object using Object.values()

Displaying the Object using JSON.stringify()

Example in which we can display the properties of object

```
const person = {
  name: "John",
  age: 30,
  city: "New York"
};
document.getElementById("demo").innerHTML =
person.name + "," + person.age + "," + person.city;
const person = {
  name: "John",
  age: 30,
  city: "New York"
};
let txt = "";
for (let x in person) {
txt += person[x] + " ";
};
document.getElementById("demo").innerHTML = txt
```

9.2.1 Using Object.values()

Any JavaScript object can be converted to an array using Object.values():

```
const person = {
  name: "John",
  age: 30,
  city: "New York"
};
const myArray = Object.values(person);
```

JavaScript for Loops

Loops can execute a block of code a number of times. JavaScript Loops are handy, if you want to run the same code over and over again, each time with a different value. JavaScript support different kind of loops

for - loops through a block of code a number of times

for/in - loops through the properties of an object

for/of - loops through the values of an iterable object

while - loops through a block of code while a specified condition is true

do/while - also loops through a block of code while a specified condition is true

9.2.1 The For Loop

The for loop has the following syntax:

```
for (statement 1; statement 2; statement 3) {
  // code block to be executed
}
```

Statement 1 is executed (one time) before the execution of the code block.

Statement 2 defines the condition for executing the code block.

Statement 3 is executed (every time) after the code block has been executed.

```
for (let i= 0;i< 5;i++)
 {
 text+= "Thenumberis " + i + "<br>";
}
```

From the example above, you can read:

Statement 1 sets a variable before the loop starts (let i = 0).

Statement 2 defines the condition for the loop to run (i must be less than 5).

Statement 3 increases a value (i++) each time the code block in the loop has been executed.

JavaScript While Loop

Loops can execute a block of code as long as a specified condition is true.

Syntax

```
while (condition)
 {
  // code block to be executed
}
```

In the following example, the code in the loop will run, over and over again, as long as a variable (i) is less than 10:

Example

```
while (i< 10)
{
  text += "The number is " + i;
i++;
}
```

The Do While Loop

The do while loop is a variant of the while loop. This loop will execute the code block once, before checking if the condition is true, then it will repeat the loop as long as the condition is true.

Syntax

```
do {
  // code block to be executed
}
while (condition);
do {
  text += "The number is " + i;
i++;
}
while (i< 10);
```

JavaScript Break and Continue

The break statement "jumps out" of a loop. The continue statement "jumps over" one iteration in the loop.

```
<html>
<body>
<script type="text/javascript">
<!--
var grade='A';
document.write("Entering switch block<br />"); switch (grade)
{
case 'A': document.write("Good job<br />");
break;
case 'B': document.write("Pretty good<br />"); break;
case 'C': document.write("Passed<br />"); break;
case 'D': document.write("Not so good<br />"); break;
case 'F': document.write("Failed<br />"); break;
default: document.write("Unknown grade<br />")
}
document.write("Exiting switch block");
//-->
</script>


<p>Set the variable to different value and then try...</p>
</body>
```

```
</html>
```

Output:

```
Entering switch block good job
Exiting switch block
Set the variable to different value and then try...
```

The For In Loop

The JavaScript for in statement loops through the properties of an Object:

Syntax

```
for (key in object) {
  // code block to be executed
}
const person = {fname:"John", lname:"Doe", age:25};
let text = "";
for (let x in person) {
  text += person[x];
}
```

Explanation of Example:

- The for in loop iterates over a person object
- Each iteration returns a key (x)
- The key is used to access the value of the key
- The value of the key is person[x]

JavaScript For Of

The JavaScript for of statement loops through the values of an iterable object. It lets you loop over iterable data structures such as Arrays, Strings, Maps, NodeLists, and more

```
for (variable of iterable)
 {
  // code block to be executed
}
```

Example Explanation

- variable - For every iteration the value of the next property is assigned to the variable. Variable can be declared with const, let, or var.
- iterable - An object that has iterable properties.

9.3 JavaScript Sets

A JavaScript Set is a collection of unique values. Each value can only occur once in a Set.You can create a JavaScript Set by:

- Passing an Array to new Set()
- Create a new Set and use add() to add values
- Create a new Set and use add() to add variables

Method	Description
new Set()	Creates a new Set
add()	Adds a new element to the Set
delete()	Removes an element from a Set
has()	Returns true if a value exists in the Set
forEach()	Invokes a callback for each element in the Set
values()	Returns an iterator with all the values in a Set

The new Set() Method

Pass an Array to the new Set() constructor:

Example

```
// Create a Set
const letters = new Set(["a","b","c"]);
```

Create a Set and add values:

```
// Create a Set
const letters = new Set();
// Add Values to the Set
letters.add("a");
letters.add("b");
letters.add("c");
```

9.4 JavaScript Function

JavaScript functions are defined with the function keyword.You can use a function declaration or a function expression.

```
function functionName(parameters)
 {

  // code to be executed

}
```

Declared functions are not executed immediately. They are "saved for later use", and will be executed later, when they are invoked (called upon).

Example

```
function myFunction(a, b)
 {
  return a * b;
}
```

Semicolons are used to separate executable JavaScript statements.Since a function declaration is not an executable statement, it is not common to end it with a semicolon.

9.5 Function Expression

A JavaScript function can also be defined using an expression. A function expression can be stored in a variable:

Example

```
const x = function (a, b) {return a * b};
```

After a function expression has been stored in a variable, the variable can be used as a function

The Function() Constructor

As you have seen in the previous examples, JavaScript functions are defined with the function keyword.Functions can also be defined with a built-in JavaScript function constructor called Function().

Example

```
const myFunction = new Function("a", "b", "return a * b");

let x = myFunction(4, 3);
```

'This' Keyword

In an object method, this refers to the object. Alone, this refers to the global object. In a function, this refers to the global object. In a function, in strict mode, this is undefined. In an event, this refers to the element that received the event. Methods like call(), apply(), and bind() can refer this to any object. Method can be reuse. With the call() method, you can write a method that can be used on different objects. In JavaScript all functions are object methods. If a function is not a method of a JavaScript object, it is a function of the global object

The example below creates an object with 3 properties, firstName, lastName, fullName.

```
const myObject = {
firstName:"John",
lastName: "Doe",
fullName: function ()
{
   return this.firstName + " " + this.lastName;
  }
}
// This will return "John Doe":
myObject.fullName();
```

In the example above, this refers to the person object.

The JavaScript call() Method

The call() method is a predefined JavaScript method.It can be used to invoke (call) a method with an owner object as an argument (parameter).

```
const person = {
  fullName: function() {
    return this.firstName + " " + this.lastName;
  }
}
const person1 = {
  firstName:"John",
  lastName: "Doe"
}
const person2 = {
  firstName:"Mary",
  lastName: "Doe"
}

// This will return "John Doe":
person.fullName.call(person1);
```

JavaScript if Else and Else if

Conditional statements are used to perform different actions based on different conditions. In JavaScript we have the following conditional statements: Use if to specify a block of code to be executed, if a specified condition is true. Use else to specify a block of code to be executed, if the same condition is false. Use else if to specify a new condition to test, if the first condition is false. Use switch to specify many alternative blocks of code to be executed.

Use the if statement to specify a block of JavaScript code to be executed if a condition is true.

Syntax

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

The Else Statement

Use the else statement to specify a block of code to be executed if the condition is false.

```
if (condition)
{
  // block of code to be executed if the condition is true
}
else
{
  // block of code to be executed if the condition is false
}
```

If the hour is less than 18, create a "Good day" greeting, otherwise "Good evening":

```
if (hour < 18)
 {
  greeting = "Good day";
}
else
{
  greeting = "Good evening";
}
```

The result of greeting will be:

Good day

The Else if Statement

Use the else if statement to specify a new condition if the first condition is false.

Syntax

```
if (condition1)
 {
  // block of code to be executed if condition1 is true
}
else if (condition2)
 {
  // block of code to be executed if the condition1 is false and condition2 is true
}
else {
  // block of code to be executed if the condition1 is false and condition2 is false
}
```

Example

If time is less than 10:00, create a "Good morning" greeting, if not, but time is less than 20:00, create a "Good day" greeting, otherwise a "Good evening":

```
if (time < 10)
{
  greeting = "Good morning";
}
 else if (time < 20)
{
  greeting = "Good day";
}
Else
 {
  greeting = "Good evening";
}
```

The result of greeting will be:

Good morning

Summary

- Objects are composed of attributes. If an attribute contains a function, it isconsidered to be a method of the object, otherwise, the attribute is considered aproperty
- Methods are actions that can be performed on objects. Object properties can be both primitive values, other objects, and functions.
- A constructor is a function that creates and initializes an object. JavaScript offers a special constructor function called Object( ).
- 'with' Keyword represents as shorthand for referencing an Objects properties or methods.
- Loops can execute a block of code as long as a specified condition is true.
- A JavaScript Set is a collection of unique values. Each value can only occur once in a Set
- The JavaScript for of statement loops through the values of an iterable object. It lets you loop over iterable data structures such as Arrays, Strings, Maps, NodeLists, and more

Keywords

Object: Objects are composed of attributes. If an attribute contains a function.

Object properties: They can be any of the three primitive data types, or any of the abstract data types, such as another object.

Else IF: Use the else if statement to specify a new condition if the first condition is false.

JavaScript: The call() method is a predefined JavaScript method.It can be used to invoke (call) a method with an owner object as an argument (parameter).

Function() Constructor: JavaScript functions are defined with the function keyword.Functions can also be defined with a built-in JavaScript function constructor called Function().

JavaScript Set: A JavaScript Set is a collection of unique values. Each value can only occur once in a Set.

'This' Keyword: this refers to the global object. In a function, this refers to the global object. In a function, in strict mode, this is undefined. In an event, this refers to the element that received the event.

Self Assessment

1. In JavaScript, an ________is a standalone entity, with properties and type.

A. Object
B. Arrays
C. Output
D. Input

2. What is definition of an undefined value in JavaScript?
A. Variable used in the code doesn't exist
B. Variable is not assigned to any value
C. Property doesn't exist

D. All of the above

3. What are the types of Pop-up boxes available in JavaScript?
A. Alert
B. Prompt
C. Confirm
D. All of the above

4. What are the two basic groups of datatypes in JavaScript?
A. A Primitive
B. Reference types.
C. All of the above
D. None of the above

5. Loops are implemented using.?
A. While Block
B. For Block
C. Do While Block
D. All the above

6. Because the ________ loop checks the condition/expression before the block is executed, the control structure is often also known as a pre-test loop
A. While loop
B. Not While loop
C. Do While loop
D. All of the above

7. Because ______loops check the condition after the block is executed, the control structure is often also known as a post-test loop
A. While loop
B. Not While loop
C. Do While loop
D. All of the above

8. The ________ statement "jumps out" of a loop
A. Break
B. Continue
C. While
D. Do while

9. The ______statement "jumps over" one iteration in the loop.
A. Break
B. Continue
C. While
D. Do while

10. Using _______we can add new element to set.
A. add()
B. new Set()
C. delete()
D. values()

11. In an object method, __________ refers to the object.
A. this
B. that

C. set
D. array

12. Use the _______statement to specify a new condition if the first condition is false.
A. else if
B. else
C. if
D. all of the above

13. Which of the following are the errors in JavaScript?
A. Load time errors
B. Run time errors:
C. Logical Errors:
D. All of the above

14. Find output of below JavaScript addition codedocument.write ("1 plus 1 is " + 1 + 1);
A. 2
B. 1 plus 1 is 2
C. 1 plus 1 is 11
D. 1 plus 1 is 1 + 1

15. Javascript is ideal to?

A. make computations in HTML simpler
B. minimize storage requirements on the web server
C. increase the download time for the client
D. none of the mentioned

Answers for Self Assessment

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

Review Questions

1. Display current date, Month, year and day of the week of the Machine.
2. Display current day of week(using ifelse() condition) to print “Hello Everyone Today is Friday, Starting of a weekend”
3. Write a JavaScript program that accept two integers and display the larger.
4. Write a JavaScript conditional statement to find the largest of five numbers. Display an alert box to show the result
5. Write a JavaScript program to find the armstrong numbers of 3 digits?

Note: An Armstrong number of three digits is an integer such that the sum of the cubes of its digits is equal to the number itself. For example, 371 is an Armstrong number since 3**3 + 7**3 + 1**3 = 371.

6. Write a JavaScript program to compute the greatest common divisor (GCD) of two positive integers.
7. Using a for loop output the elements in reverse order.

Further Readings

HTMl, JavaScript , DHTML and PHP By. Ivan Bayross .

Beginning JavaScript 2nd Edition By. Paul Wilton

Html 5 Black Book, Covers Css 3, Javascript, Xml, Xhtml, Ajax, PhpAndJquery, Second Edition

Web Links

https://www.javascripttutorial.net/javascript-function/

www.webopedia.com

www.web-source.net

Aseem Khanna, Lovely Professional University

Unit 10: Using JavaScript to Build Web Applications

CONTENTS

Objectives

After studying this unit, you will be able to:

- Discuss introduction regarding DOM
- Explain Concept of Handling Events
- What is this 'event' Argument
- Basics of HTTP and Ajax
- Processing JSON
- Mobile Nav Menu Automatic Collapse

Introduction

A Document object represents the whole HTML document that is displayed in that window. The Document object has various properties that refer to other objects which allow access to and modification of document content.

When html document is loaded in the browser, it becomes a document object. It is the root element that represents the html document. It has properties and methods. By the help of document object, we can add dynamic content to our web page.

HTML is used to structure the web pages and Javascript is used to add behavior to our web pages. When an HTML file is loaded into the browser, the Javascript cannot understand the HTML document directly.

Window object: Top of the hierarchy. It is the outmost element of the object hierarchy.

Document object: Each HTML document that gets loaded into a window becomes a document object. The document contains the contents of the page.

Form object: Everything enclosed in the <form>...</form> tags sets the form object.

Form control elements: The form object contains all the elements defined for that object such as text fields, buttons, radio buttons, and checkboxes.

10.1 Why DOM is Required?

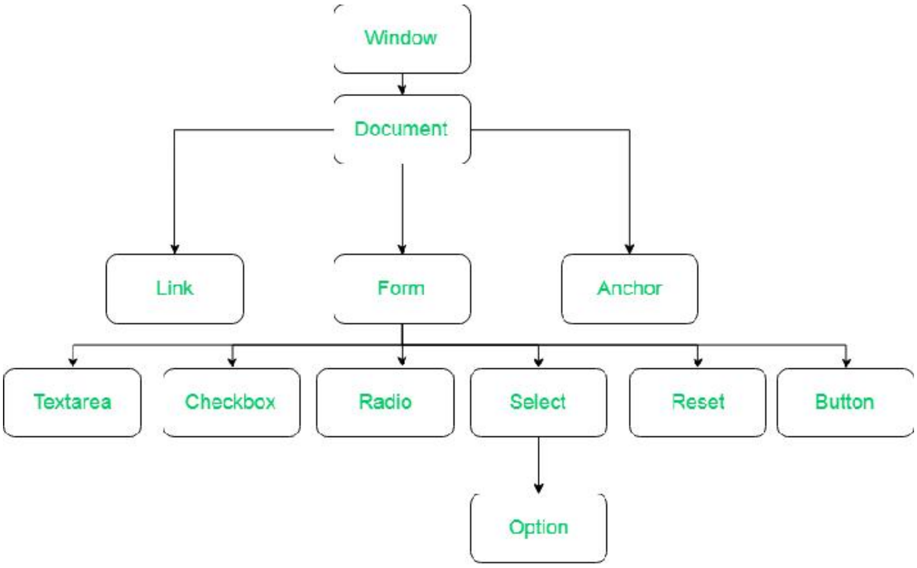
Javascript interprets DOM easily i.e javascript cannot understand the tags(<h1>H</h1>) in HTML document but can understand object h1 in DOM. Now, Javascript can access each of the objects (h1, p, etc) by using different functions.

We most often see DOM in the context of JavaScript - this makes sense as JS is the most often used programming language of the web. The ECMAScript spec (a.k.a. the JS spec) defines no formal I/O, so it's necessary that the environment your code runs in provide some kind of interface to actually do anything.

Notes: DOM isn't exclusive to JavaScript and web browsers - there are DOM parsers for pretty much every language under the sun.

What is Object Model?

Documents are modeled using objects and the model includes not only the structure of a document but also the behavior of a document and the objects of which it is composed of like tag elements with attributes in HTML.



Properties of DOM

Window Object: Window Object is always at top of the hierarchy.

Document Object: When an HTML document is loaded into a window, it becomes a document object.

Form Object: It is represented by form tags.

Link Object: It is represented by link tags.

Anchor Object: It is represented by a HREF tags.

Form Control Elements: Form can have many control elements such as text fields, buttons, radio buttons, and checkboxes, etc.

10.2 JavaScript HTML DOM Events

HTML DOM allows JavaScript to react to HTML events:

A JavaScript can be executed when an event occurs, like when a user clicks on an HTML element.

To execute code when a user clicks on an element, add JavaScript code to an HTML event attribute

onclick=JavaScript

HTML DOM allows JavaScript to react to HTML events:

A JavaScript can be executed when an event occurs, like when a user clicks on an HTML element.

To execute code when a user clicks on an element, add JavaScript code to an HTML event attribute

onclick=JavaScript

- When a user clicks the mouse
- When a web page has loaded
- When an image has been loaded
- When the mouse moves over an element
- When an input field is changed
- When an HTML form is submitted
- When a user strokes a key

JavaScript HTML DOM - Changing CSS

The HTML DOM allows JavaScript to change the style of HTML elements.

To change the style of an HTML element, use this syntax:

document.getElementById(id).style.property = new style

Example shows assignment of <p> Tag

```
<html>
<body>
<p id="p2">Hello World!</p>
<script>
document.getElementById("p2").style.color = "blue";
</script>
</body>
</html>
```

10.3 The 'event' Argument

The HTML DOM allows you to execute code when an event occurs.Events are generated by the browser when "things happen" to HTML elements like when we click on an element it will load the page and the required fields are changed.

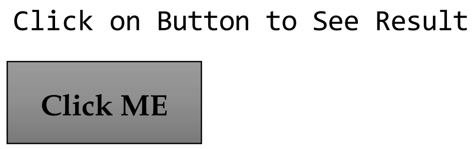
The change in the state of an object is known as an Event. In html, there are various events which represents that some activity is performed by the user or by the browser. When JavaScript code is included in HTML, js react over these events and allow the execution. This process of reacting over the events is called Event Handling. Thus, js handles the HTML events via Event Handlers.

onclick Event Type

This is the most frequently used event type which occurs when a user clicks the left button of his mouse. You can put your validation, warning etc., against this event type.

```
<html>

<head>

<scripttype="text/javascript">

<!--

function          clickMe()

    {document.write("Good
   Morning")
}
//-->
</script>
</head>
<body>
<p> Click on Button to See Result</p>
<inputtype="button"onclick="ClickMe()"value="Click ME"/>
</body>
</html>
```

Output:



```
Click on Button to See Result
```

onmouseover and onmouseout

These two type of events will help you to creategood effects with images or even with text as well. The onmouseover event triggers when you bring your mouse over any element and the onmouseout triggers when you move your mouse out from that element.

```
<html>
<head>
<script type="text/javascript">
<!--
function over() {
document.write ("Mouse Over Event ");
}
function out() {
document.write ("Mouse Out Event");
}
//-->
</script>
</head>
<body>
<p>Bring your mouse inside the required area:</p>
<div onmouseover="over()" onmouseout="out()">
<h2>You have entered in the Area</h2>
</div>
</body>
</html>
```

Output:

```
Bringyourmouseinsidethe required area:

You have entered in the Area
```

There are more events show below in the table

Event Performed	Event Handler	Description
click	onclick	When mouse click on an element
mouseover	onmouseover	When the cursor of the mouse comes over the element
mouseout	onmouseout	When the cursor of the mouse leaves an element
mousedown	onmousedown	When the mouse button is pressed over the element
mouseup	onmouseup	When the mouse button is released over the element
mousemove	onmousemove	When the mouse movement takes place.
Keydown & Keyup	Onkeydown & onkeyup	When the user press and then release the key
focus	onfocus	When the user focuses on an element
submit	Onsubmit	When the user submits the form
blur	Onblur	When the focus is away from a form element
change	onchange	When the user modifies or changes the value of a form element
load	onload	When the browser finishes the loading of the page
unload	onunload	When the visitor leaves the current webpage, the browser unloads it
resize	onresize	When the visitor resizes the window of the browser

JavaScript Cookies

A cookie is an amount of information that persists between a server-side and a client-side. A web browser stores this information at the time of browsing. A cookie contains the information as a string generally in the form of a name-value pair separated by semi-colons. It maintains the state of a user and remembers the user's information among all the web pages.

When a user sends a request to the server, then each of that request is treated as a new request sent by the different user. So, to recognize the old user, we need to add the cookie with the response from the server. Browser at the client-side. Now, whenever a user sends a request to the server, the cookie is added with that request automatically. Due to the cookie, the server recognizes the users.

Creating a cookie in JavaScript

In JavaScript, we can create, read, update and delete a cookie by using document.cookie property.

The following syntax is used to create a cookie:

document.cookie="name=value";

JavaScript can create, read, and delete cookies with the document.cookie property.

With JavaScript, a cookie can be created like this:

document.cookie = "username=John Doe";

You can also add an expiry date (in UTC time). By default, the cookie is deleted when the browser is closed:

document.cookie = "username=John Doe; expires=Thu, 18 Dec 2013 12:00:00 UTC";

Read a Cookie with JavaScript

With JavaScript, cookies can be read like this:

let x = document.cookie;

document.cookie will return all cookies in one string much like: cookie1=value; cookie2=value; cookie3=value;

The Cookie String

The document.cookie property looks like a normal text string. But it is not.

Even if you write a whole cookie string to document.cookie, when you read it out again, you can only see the name-value pair of it.

If you set a new cookie, older cookies are not overwritten. The new cookie is added to document.cookie, so if you read document.cookie again you will get something like:

cookie1 = value; cookie2 = value;

Advantages of Cookies

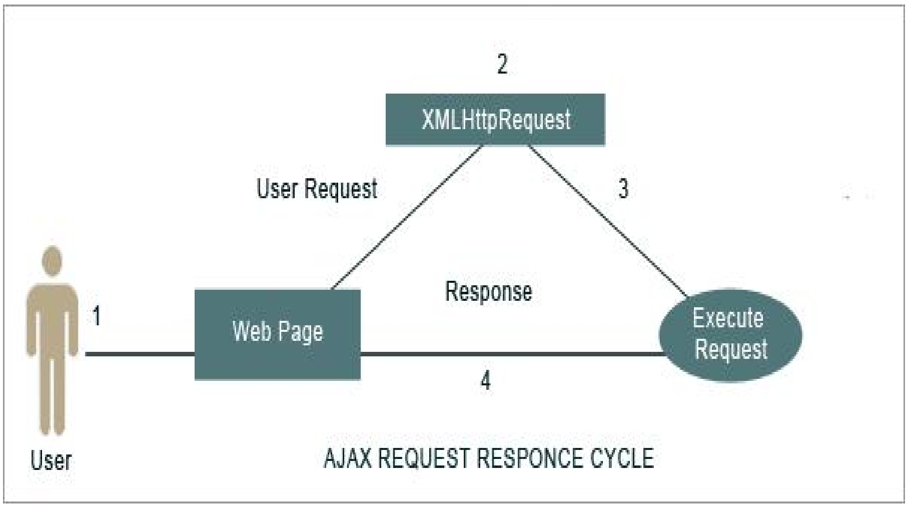
- Cookies do not require any server resources since they are stored on the client.
- Cookies are easy to implement.
- You can configure cookies to expire when the browser session ends (session cookies) or they can exist for a specified length of time on the client computer (persistent cookies).

Disadvantages of Cookies

- Users can delete a cookies.
- Users browser can refuse cookies, so your code has to anticipate that possibility.
- Cookies exist as plain text on the client machine and they may pose a possible security risk as anyone can open and tamper with cookies.

HTTP Basics Ajax Basics

Ajax (also AJAX "Asynchronous JavaScript and XML") is a set of web development techniques that uses various web technologies on the client-side to create asynchronous web applications. With Ajax, web applications can send and retrieve data from a server asynchronously (in the background) without interfering with the display and behavior of the existing page



By decoupling the data interchange layer from the presentation layer, Ajax allows web pages and, by extension, web applications, to change content dynamically without the need to reload the entire page. In practice, modern implementations commonly utilize JSON instead of XML.

10.4 HTTP Basics Ajax Basics

AJAX = Asynchronous JavaScript And XML.AJAX is not a programming language.

AJAX just uses a combination of: A browser built-in XMLHttpRequest object (to request data from a web server)JavaScript and HTML DOM (to display or use the data). AJAX allows web pages to be updated asynchronously by exchanging data with a web server behind the scenes. This means that it is possible to update parts of a web page, without reloading the whole page.

Ajax Technology

AJAX cannot work independently. It is used in combination with other technologies to create interactive webpages.JavaScript is a Loosely typed scripting language.JavaScript function is called when an event occurs in a page.Glue for the whole AJAX operation. It is also used with DOM and API for accessing and manipulating structured documents. It represents the structure of XML and HTML documents& CSS. It allows for a clear separation of the presentation style from the content and may be changed by JavaScript

10.5 Processing JSON

JSON stands for JavaScript Object Notation. JSON is extremely trivial data-interchange format for data exchange between server and client which is quick and easy to parse and generate.

Like XML, JSON is also a text-based format means it’s easy to write and easy to understand for both humans and computers, but unlike XML, JSON data structures occupy less bandwidth than their XML versions. JSON is based on two basic structures

```
<html lang="en">
<head>
<meta charset="utf-8">
<title>Parse JSON Object in JavaScript</title>
</head>
<body>
<script>
  var json = `{
    "book": {
        "name": "Build your world with C++",
        "author": "Aseem Khanna",
        "year": 2013,
        "genre": "Programming Language",
        "bestseller": true
      }
   }`;

   // Converting JSON object to JS object
   var obj = JSON.parse(json);
   console.log(obj);
</script>
<p><strong>Note:</strong> Please check out the browser console by pressing the f12 key on the keyboard. </p>
</body>
</html>
```

Parsing JSON in Javascript

Functions are not allowed in JSON.If you need to include a function, write it as a string.

we can convert it back into a function later:

Example Shows JSON Parsing

```
<html>
<body>
<h2>Convert a string into a function.</h2>
<p id="demo"></p>
<script>
const text = '{"name":"", "age":"function() {return 35;}", "city":"Jalandhar"}';
const obj = JSON.parse(text);
obj.age = eval("(" + obj.age + ")");
document.getElementById("demo").innerHTML = obj.name + ", " + obj.age();
</script>
</body>
</html>
```

OUTPUT:

Convert a string into a function.

Aseem, 35, Jalandhar

Summary

- Javascript interprets DOM easily i.e javascript cannot understand the tags(<h1>H</h1>) in HTML document but can understand object h1 in DOM.

- Documents are modeled using objects and the model includes not only the structure of a document but also the behavior of a document and the objects of which it is composed of like tag elements with attributes in HTML.
- Form can have many control elements such as text fields, buttons, radio buttons, and checkboxes, etc.
- A JavaScript can be executed when an event occurs, like when a user clicks on an HTML element.
- The HTML DOM allows JavaScript to change the style of HTML elements.
- The HTML DOM allows you to execute code when an event occurs.Events are generated by the browser when "things happen" to HTML elements like when we click on an element it will load the page and the required fields are changed.
- A cookie contains the information as a string generally in the form of a name-value pair separated by semi-colons.
- The document.cookie property looks like a normal text string

Keywords

JavaScript:HTML is used to structure the web pages and Javascript is used to add behavior to our web pages.

Form control elements: The form object contains all the elements defined for that object such as text fields, buttons, radio buttons, and checkboxes.

DOM:DOM easily i.e javascript cannot understand the tags(<h1>H</h1>) in HTML document but can understand object h1 in DOM.

Event: HTML DOM allows you to execute code when an event occurs.

onclick Event: This is the most frequently used event type which occurs when a user clicks the left button of his mouse.

Onmouseover: onmouseover event triggers when you bring your mouse over any element and the onmouseout triggers when you move your mouse out from that element

Cookie:A cookie is an amount of information that persists between a server-side and a client-side.

Cookie String: document.cookie property looks like a normal text string.

Ajax: Ajax (also AJAX "Asynchronous JavaScript and XML") is a set of web development techniques that uses various web technologies on the client-side to create asynchronous web applications

JSON:JSON stands for JavaScript Object Notation. JSON is extremely trivial data-interchange format for data exchange between server and client which is quick and easy to parse and generate.

Self Assessment

1. The W3C ____________is a platform and language-neutral interface that allows programs and scripts to dynamically access and update the content, structure, and style of a document.

A. Document Object Model
B. Document various Model
C. Docking Object Model
D. Document Object variables

2. What is the purpose of the Attr object in the HTML DOM?

A. Used to focus on a particular part of the HTML page
B. HTML Attribute
C. Used to arrange elements
D. CSS attribute

3. What is the work of the form control elements in the HTML DOM?
A. User Interface elements
B. All the possible elements
C. Debugging elements
D. Collecting elements

4. Which of the following are Examples of HTML events
A. When a user clicks the mouse
B. When a web page has loaded
C. When an image has been loaded
D. All of the above

5. The HTML DOM allows JavaScript to change the style of HTML elements.
A. True
B. False

6. ____________is the process of ensuring that user input is clean, correct, and useful.
A. Data validation
B. Data fragmentation
C. Data variability
D. Data analysis

7. Which event handler is used to initiate when mouse click on an element?
A. onclick
B. onmouseover
C. onmouseout
D. onmousedown

8. Which event handler is used to initiate when the cursor of the mouse comes over the element?
A. onclick
B. onmouseover
C. onmouseout
D. onmousedown

9. Which event handler is used to initiate when the cursor of the mouse leaves an element?
A. onclick
B. onmouseover
C. onmouseout
D. onmousedown

10. Which event handler in form events is used to initiate when the user focuses on an element?
A. onfocus
B. onsubmit
C. onblur

D. onchange

11. Which event handler in form events is used to initiate when the focus is away from a form element?

A. onfocus

B. onsubmit

C. onblur

D. onchange

12. With AJAX one can :

A. Update a web page without reloading the page

B. Request data from a server - after the page has loaded

C. Receive data from a server - after the page has loaded

D. All of the above

13. What is full form of AJAX:

A. Asynchronous Java And XML.

B. Asynchronous JavaScript And XML.

C. Asynchronous JavaScript And HTML.

D. Synchronous JavaScript And XML.

14. Typical validation tasks are:

A. has the user filled in all required fields?

B. has the user entered a valid date?

C. has the user entered text in a numeric field?

D. All of the above

15. Which constraint validation specifies that the input element should be disabled

A. Disabled

B. pattern

C. required

D. type

16. Which constraint validation Specifies that the input field requires an element

A. Disabled

B. pattern

C. required

D. type

Answers for Self Assessment

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

Review Questions

1. What's the difference between window vs document?
2. What are the methods to get one or more elements from the DOM?
3. What is main function of DOM?
4. What is method involved in HTML DOM?
5. What are the functionalities performed by onload() and onUpload()?
6. What are HTML DOM Event?
7. What are JavaScript Cookies?
8. What is Ajax?
9. What isCookie String?
10. What are the JSON Techniques?
11. How parsing is done in JSON? Explain with Example?

Further Readings

HTMl, JavaScript , DHTML and PHP By. Ivan Bayross .

Beginning JavaScript 2nd Edition By. Paul Wilton

Html 5 Black Book, Covers Css 3, Javascript, Xml, Xhtml, Ajax, Php And Jquery, Second Edition

Web Links

https://www.javascripttutorial.net/javascript-function/

www.webopedia.com

www.web-source.net

Unit 11: Basics of React

CONTENTS

Objectives

After studying this unit, you will be able to:

- Discuss React's Future its obstacles and Roadblocks
- How the Files functional programming works
- Declaring variables and Arrow Functions
- Es6 objects and arrays, Promise classes

Introduction

React (also known as React.js or ReactJS) is a free and open-source front-end JavaScript library for building user interfaces based on UI components. It is maintained by Meta (formerly Facebook) and a community of individual developers and companies.

React can be used as a base in the development of single-page or mobile applications. However, React is only concerned with state management and rendering that state to the DOM, so creating React applications usually requires the use of additional libraries for routing, as well as certain client-side functionality.

Notes: Declarative-React adheres to the declarative programming paradigm. Developers design views for each state of an application, and React updates and renders components when data changes. This is in contrast with imperative programming.

Components of React

At the heart of all React applications are components. A component is a self-contained module that renders some output. We can write interface elements like a button or an input field as a React component. Components what is React? 1 are compostable. A component might include one or more other components in its output. Broadly speaking, to write React apps we write React components that correspond to various interface elements. We then organize these components inside higher-level components which define the structure of our application. For example, take a form. A form might consist of many interface elements, like input fields, labels, or buttons. Each element inside the form can be written as a React component. We'd then write a higher-level component, the form component itself. The form component would specify the structure of the form and include each of these interface elements inside of it.

Virtual DOM

Another notable feature is the use of a virtual Document Object Model, or virtual DOM. React creates an in-memory data-structure cache, computes the resulting differences, and then updates the browser's displayed DOM efficiently. This process is called reconciliation. This allows the programmer to write code as if the entire page is rendered on each change, while the React libraries only render subcomponents that actually change.

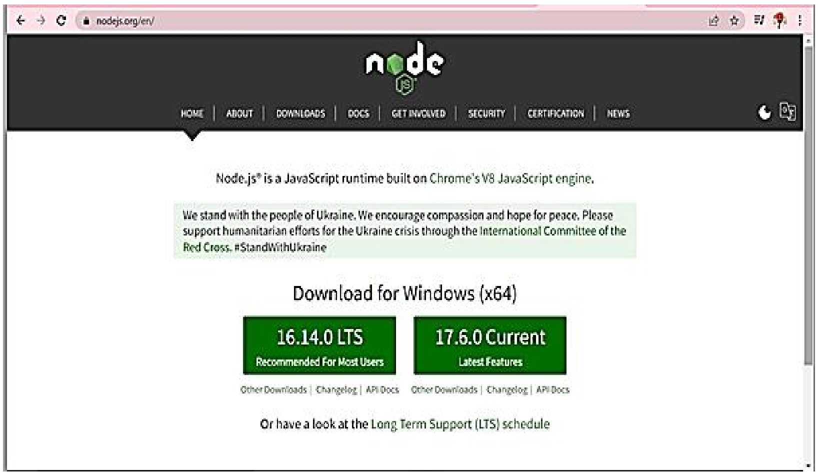
11.1 Redact's Future, Installation of React

Installation of React need some pre-Installation of software which are discussed step by step

- Install Node JS and NPM
- Install Visual Studio Code
- Install React from Terminal
- npm install -g create-react-app

Install Node JS and NPM

- Go to https://nodejs.org/en/
- Download the latest version of NodeJS LTS with Administrator privileges.



Node.js® is a JavaScript runtime built on Chrome's V8 JavaScript engine.

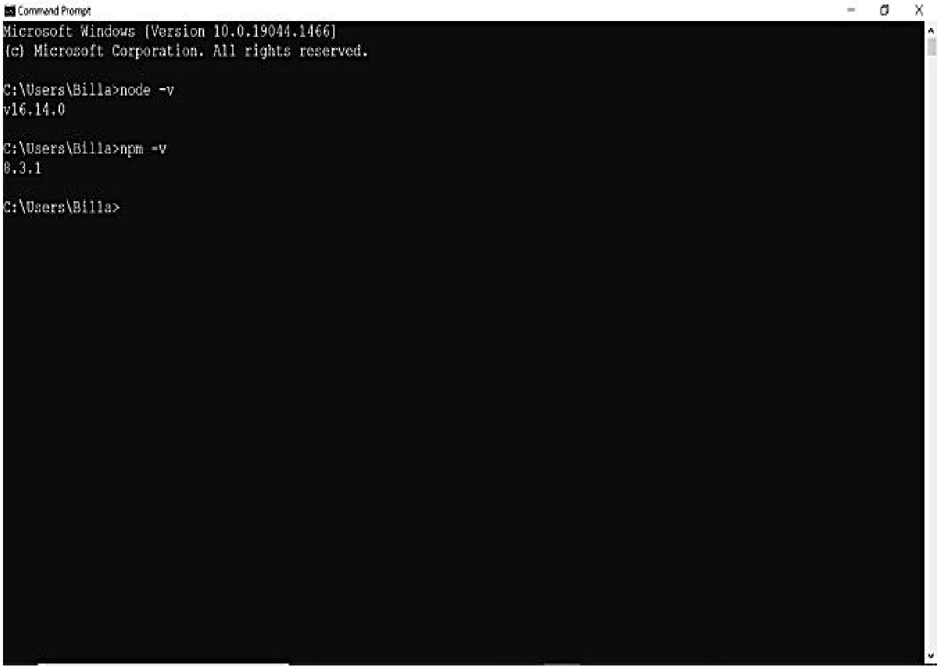
We stand with the people of Ukraine. We encourage compassion and hope for peace. Please support humanitarian efforts for the Ukraine crisis through the International Committee of the Red Cross. #StandWithUkraine

Download for Windows (x64)

16.14.0 LTS
Recommended For Most Users
17.6.0 Current
Latest Features
Other Downloads | Changelog | API Docs
Other Downloads | Changelog | API Docs

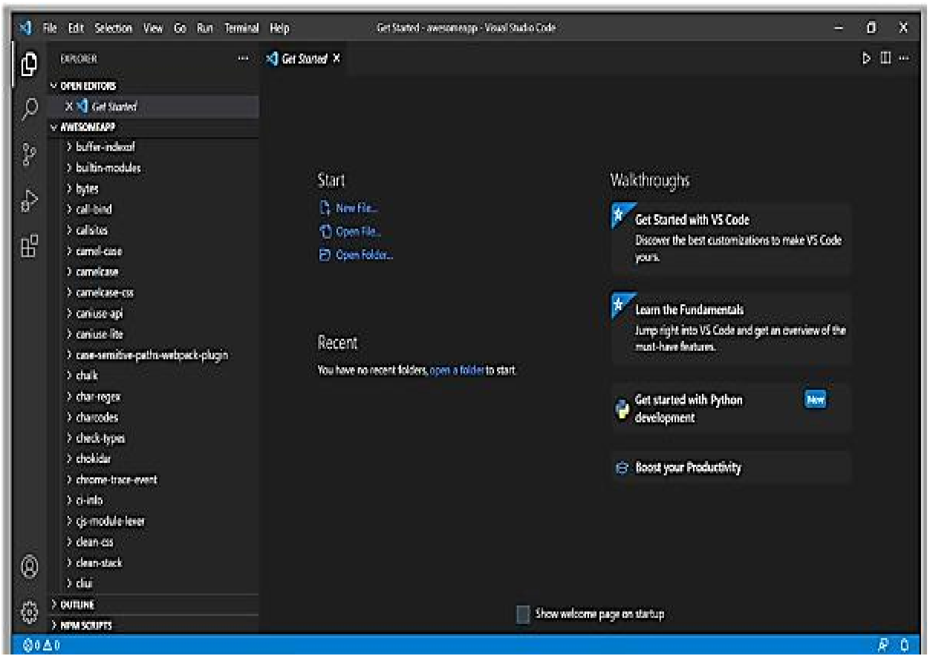
Or have a look at the Long Term Support (LTS) schedule

Npm installation* npm -v



```
Microsoft Windows [Version 10.0.19044.1466]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Billa>node -v
v16.14.0

C:\Users\Billa>npm -v
8.3.1

C:\Users\Billa>
```

Install Visual Studio Code

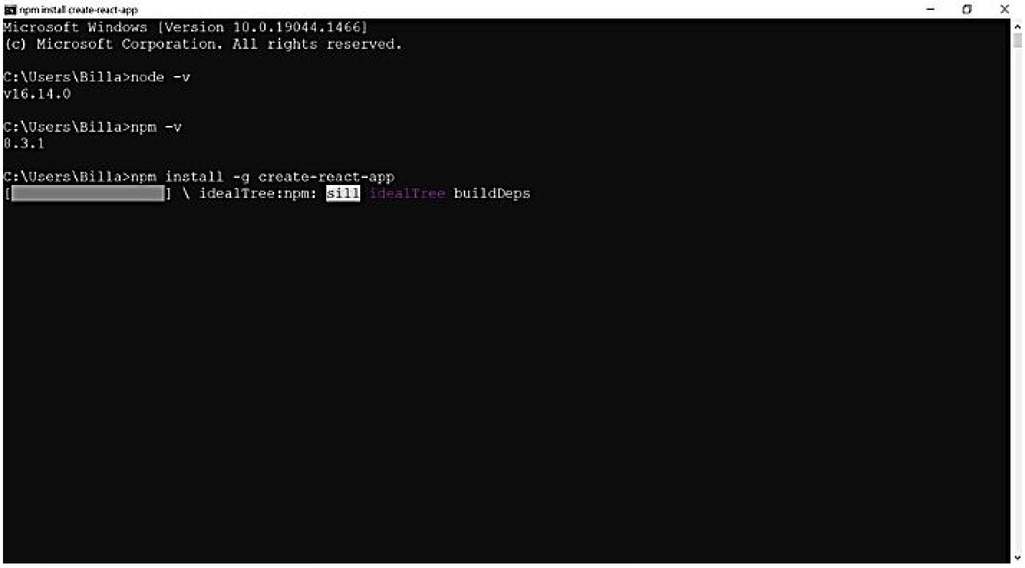


You have no recent folders, open a folder to start.

Discover the best customizations to make VS Code yours.

Jump right into VS Code and get an overview of the must-have features.

npm install -g create-react-app



```
Microsoft Windows [Version 10.0.19044.1466]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Billa>node -v
v16.14.0

C:\Users\Billa>npm -v
8.3.1

C:\Users\Billa>npm install -g create-react-app
[████████████████████] \ idealTree:npm: sill idealTree buildDeps
```

npm is the package manager for the Node JavaScript platform. It puts modules in place so that node can find them, and manages dependency conflicts intelligently. It is extremely configurable to support a wide variety of use cases. Most commonly, it is used to publish, discover, install, and develop node programs.

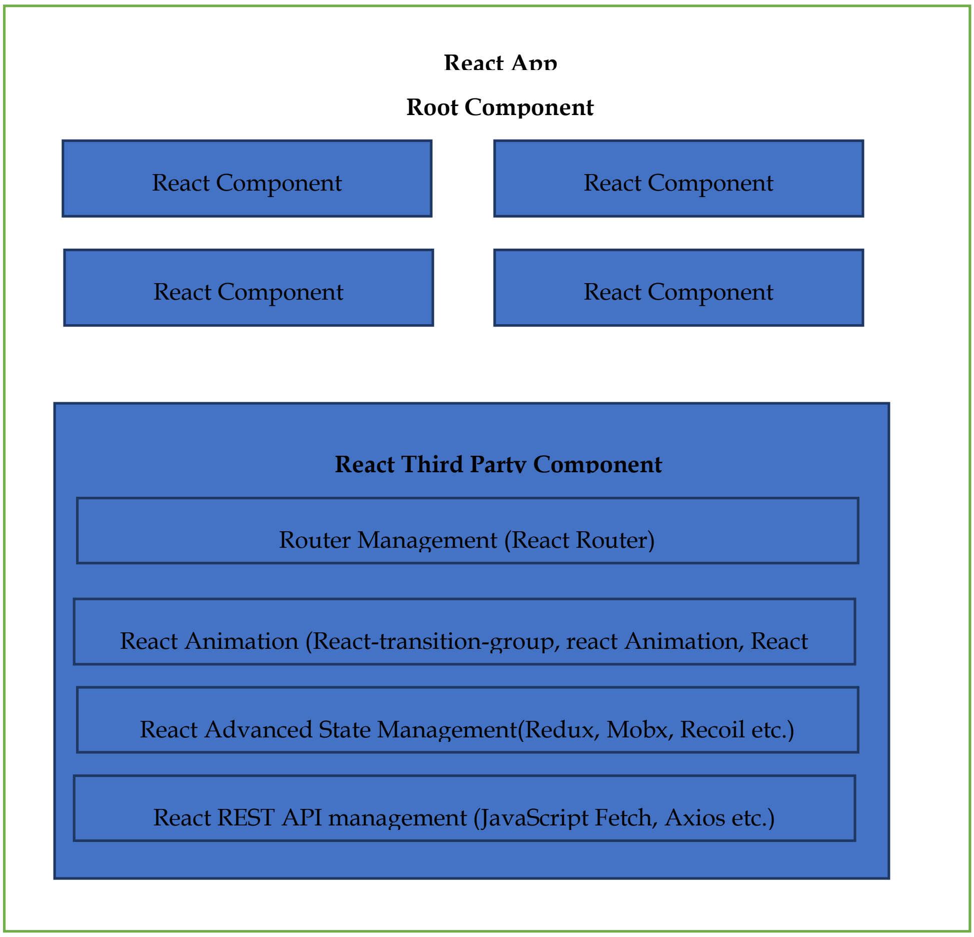
Notes:MVC (Model-View-Controller) is a pattern in software design commonly used to implement user interfaces, data, and controlling logic. It emphasizes a separation between the software's business logic and display. This "separation of concerns" provides for a better division of labor and improved maintenance.

11.2 Working with Files Functional Programming

React (also known as **React.js** or **ReactJS**) is a free and open-source front-end JavaScript library for building user interfaces based on UI components. It is maintained by Meta (formerly Facebook) and a community of individual developers and companies.

The main objective of ReactJS is to develop User Interfaces (UI) that improves the speed of the apps. It uses virtual DOM (JavaScript object), which improves the performance of the app. The JavaScript virtual DOM is faster than the regular DOM. We can use ReactJS on the client and server-side as well as with other frameworks. It uses component and data patterns that improve readability and helps to maintain larger apps.

Architecture of React Application


React App
Root Component
React Component
React Component
React Component
React Component
React Third Party Component
Router Management (React Router)
React Animation (React-transition-group, react Animation, React
React Advanced State Management(Redux, Mobx, Recoil etc.)
React REST API management (JavaScript Fetch, Axios etc.)

- React app starts with a single root component.
- Root component is build using one or more component.
- Each component can be nested with other component to any level.
- Composition is one of the core concepts of React library. So, each component is build by composing smaller components instead of inheriting one component from another component.
- Most of the components are user interface components.
- React app can include third party component for specific purpose such as routing, animation, state management, etc.

Using Create React App tool

Open a terminal and go to your workspace.

```
> cd /go/to/your/workspace
```

Next, create a new React application using Create React App tool.

```
>Create-react-app expense-manager
```

It will a create new folder expense-manager with startup template code.

Next, go to expense-manager folder and install the necessary library

```
cd expense-manager npm install
```

The npm install will install the necessary library under node_modules folder. Next, start the application.

```
npm start
```

Compiled successfully!

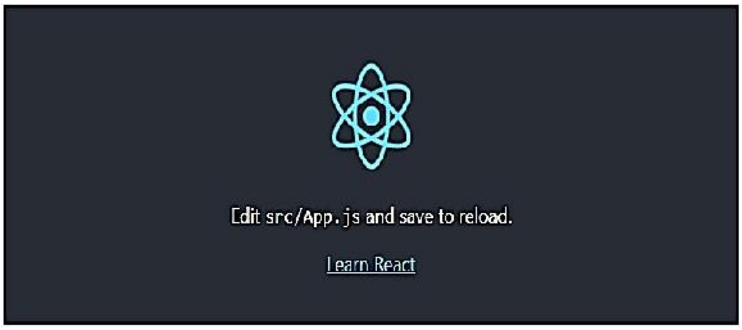
You can now view react-cra-web-app in the browser.

Local: http://localhost:3000

On Your Network: http://192.168.56.1:3000

Note that the development build is not optimized. To create a production build, use npm run build.

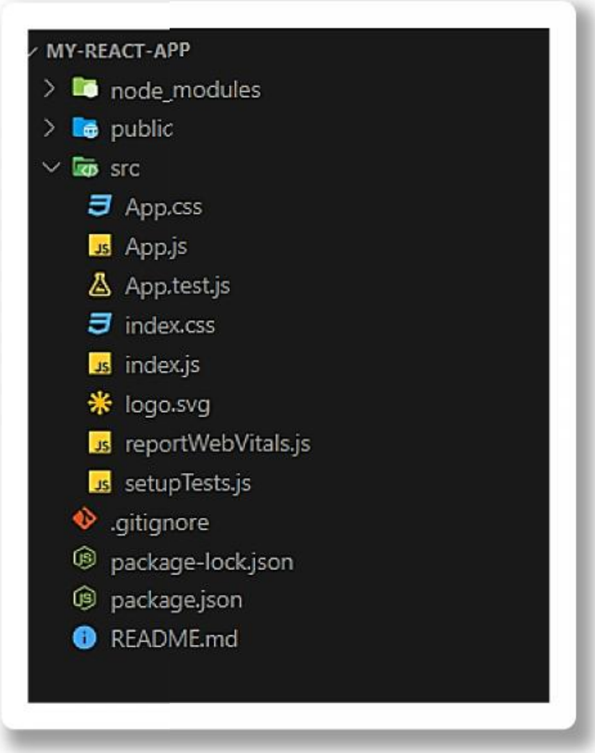
Next, open the browser and enter http://localhost:3000 in the address bar and press enter. The development web server will serve our webpage as shown below.



Edit src/App.js and save to reload.

Learn React

Structure of React Folder

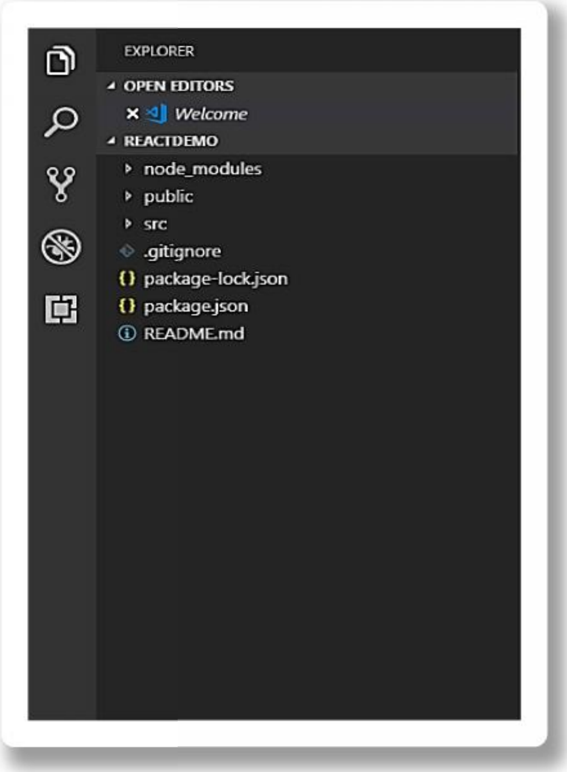


```
MY-REACT-APP
 > node_modules
 > public
 v src
     App.css
     App.js
     App.test.js
     index.css
     index.js
     logo.svg
     reportWebVitals.js
     setupTests.js
   .gitignore
   package-lock.json
   package.json
   README.md
```

- After creating a React application, we will notice that in Terminal, it provides quick tips or commands that will help in executing the React application. For example :-
- npm start - This command will start the development server.
- npm run build - Bundles the app into static files for production.
- npm test - starts the test runner.
- npm run object - Removes this tool and copies build dependencies, configuration files, and scripts into the app directory.

Folder Structure of React

The React application automatically creates required folders, as shown below.



EXPLORER
- OPEN EDITORS
  - × Welcome
- REACTDEMO
  - node_modules
  - public
  - src
  - .gitignore
  - package-lock.json
  - package.json
  - README.md

Gitignore

Thegitignore file tells Git which files to ignore when committing your project to the GitHub repository. gitignore is located in the root directory of your repo. / will ignore directories with the name. Ignored files are usually build artifacts and machine generated files that can be derived from your repository source or should otherwise not be committed.

Package. json file

The package. json file is the heart of any Node project. It records important metadata about a project which is required before publishing to NPM, and also defines functional attributes of a project that npm uses to install dependencies, run scripts, and identify the entry point to our package.

Package-lock.json

Package-lock.json contain exact dependency tree to be installed in /node_modules. It helps while a team is working on private apps to ensure that they are working on the same version of dependencies and sub-dependencies. It also maintains a history of changes done in package.json so, that at any point of time, when required previous changes can be looked back in the package-lock.json file.

node_modules

This folder contains all dependencies and sub-dependencies specified in package.json used by React app. It contains more than 800 subfolders; this folder is automatically added in the .gitignore file

Public–

This folder contains files which don't require additional processing by web pack. The index.html file is considered as an entry point for the web application. Here, the favicon is a header icon and manifest.xml file contains configuration when your application is used for Android app

src –

This folder is the heart of React application as it contains JavaScript which needs to be processed by webpack. In this folder, there is a main component App.js, its related styles (App.css), test suite (App.test.js). index.js, and its style (index.css); which provide an entry point into the App. Lastly, it contains registerServiceWorker.js which takes care of caching and updating files for the end user. It helps in offline capability and faster page loading after the first visit.

11.3 ES6

ES6 stands for ECMAScript 6.ECMAScript was created to standardize JavaScript, and ES6 is the 6th version of ECMAScript, it was published in 2015, and is also known as ECMAScript 2015.ECMAScript (European Computer Manufacturers Association Script) is a scripting language based on JavaScript. Invented by Brendan Eich at Netscape, ECMAScript made its first appearance in the Navigator 2.0 browser.

ES6 or ECMAScript 6 is a scripting language specification which is standardized by ECMAScript International. This specification governs some languages such as JavaScript, Action Script, and Jscript. ECMAScript is generally used for client-side scripting, and it is also used for writing server applications and services by using Node.js. ES6 allows you to write the code in such a way that makes your code more modern and readable. By using ES6 features, we write less and do more, so the term 'Write less, do more' suits ES6. ES6 introduces you many great features such as scope variable, arrow functions, template strings, class destructions, modules, etc

React es6 Classes

Classes are an essential part of object-oriented programming (OOP). Classes are used to define the blueprint for real-world object modeling and organize the code into reusable and logical parts. Before ES6, it was hard to create a class in JavaScript. But in ES6, we can create the class by using the class keyword. We can include classes in our code either by class expression or by using a class declaration.

A class is a type of function, but instead of using the keyword function to initiate it, we use the keyword class, and the properties are assigned inside a constructor () method.

Example:

A simple class constructor:

```
Class Car
{
Constructor (name) {
This. brand = name;
  }
}
```

11.4 ES6 Object and Arrays

An object is an instance which contains a set of key value pairs. Unlike primitive data types, objects can represent multiple or complex values and can change over their life time. The values can be scalar values or functions or even array of other objects.

The syntactic variations for defining an object are discussed further.

```
var identifier = {
  Key1: value, Key2: function () {
    //functions
  },
  Key3: [“content1”,” content2”]
}
```

The contents of an object are called **properties** (or members), and properties consist of a **name** (or key) and **value**. Property names must be strings or symbols, and values can be any type (including other objects).

Creating an object in es6

```
Class Car {
Constructor (name) {
This. brand = name;
  }
}
Constmy car = new Car("Ford");
```

ES6 Arrays

An array declaration allocates sequential memory blocks and is static. This means that an array once initialized cannot be resized. Each memory block represents an array element. Array elements are identified by a unique integer called as the subscript/index of the element. Arrays too, like variables, should be declared before they are used. Array initialization refers to populating the array elements. Array element values can be updated or modified but cannot be deleted

Declaring and Initializing Arrays

To declare and initialize an array in JavaScript use the following syntax −

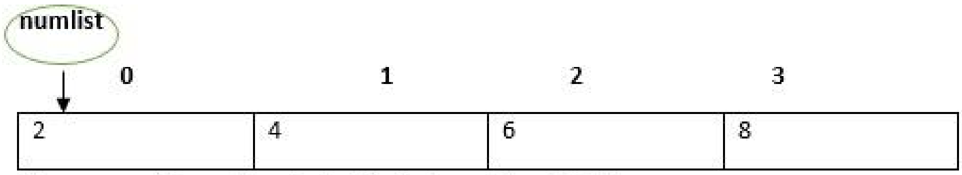
```
var array_name; //declaration
array_name = [val1, val2, valn...]   //initialization
```

OR

```
var array_name = [val1,val2…valn]
```

Note − the pair of [] is called the dimension of the array.

For example, a declaration like: var numlist = [2,4,6,8] will create an array as shown in the following figure.



The array pointer refers to the first element by default.

Accessing Array Elements

The array name followed by the subscript is used to refer to an array element.

Following is the syntax for the same.

```
array_name[subscript]
```

Example: Simple Array

```
var set;
Set = ["1","2","3","4"]
console.log (set [0]);
console.log (set [1]);
```

The following output is displayed on successful execution of the above code.

```
1
2
```

Arrow Function

Arrow functions are introduced in ES6 which provides you a more accurate way to write the functions in JavaScript. They allow us to write smaller function syntax. Arrow functions make your code more readable and structured. They are also known as anonymous functions (the functions without a name and not bound with an identifier). They don't return any value and can declare without the function keyword. Arrow functions cannot be used as the constructors.

Arrow functions allow us to write shorter function syntax:

Before:

Hello = function ()

{

Return "Hello World!";

}

After the use of arrow function

Hello = () =>

{

Return "Hello World!";

}

It gets shorter! If the function has only one statement, and the statement returns a value, you can remove the brackets and the return keyword.

Arrow Functions Return Value by Default:

Hello = () => "Hello World!";

If you have parameters, you pass them inside the parentheses:

Arrow Function with Parameters:

Hello = (val) => "Hello" + val;

Advantages of Arow Function

- It reduces code size.
- The return statement is optional for a single line function.
- Lexically bind the context.
- Functional braces are optional for a single-line statement.

THIS Function

The handling of this is also different in arrow functions compared to regular functions. In short, with arrow functions there are no binding of this. In regular functions the this keyword represented the object that called the function, which could be the window, the document, a button or whatever. With arrow functions, this keyword always represents the object that defined the arrow function.

React es6 variables

Before ES6 there were only one way of defining your variables: with the var keyword. If you did not define them, they would be assigned to the global object. Unless you were in strict mode, then you would get an error if your variables were undefined.

Now, with ES6, there are three ways of defining your variables: var, let, and const.

var

var x = 5.6;

If you use var outside of a function, it belongs to the global scope. Use var inside of a function, it belongs to that function. Var inside of a block, i.e.a for loop, the variable is still available outside of that block.

Let

Let x = 5.6;

Let is the block scoped version of var, and is limited to the block (or expression) where it is defined. If you use let inside of a block, i.e.a for loop, the variable is only available inside of that loop. let has a block scope.

Const

Const x = 5.6;

Const is a variable that once it has been created, its value can never change.const has a block scope.

11.5 Modules of es6

JavaScript modules allow you to break up your code into separate files. This makes it easier to maintain the code-base.ES Modules rely on the import and export statements. A module organizes a related set of JavaScript code. A module can contain variables and functions. A module is nothing more than a chunk of JavaScript code written in a file. By default, variables and functions of a module are not available for use.

You can export a function or variable from any file.

Let us create a file named person.js, and fill it with the things we want to export.

There are two types of exports:

You can export a function or variable from any file. Suppose to create a file named person.js, and fill it with the things we want to export. There are two types of exports:

- **Named:** Named exports are distinguished by their names. There can be several named exports in a module. A module can export selected components using the syntax given below −

  Syntax:

```
//using multiple export keyword
Export component1
Export component2
...
...
ExportcomponentN
```

- **Default export**: Modules that need to export only a single value can use default exports. There can be only one default export per module.

Syntax

```
Export default component_name
```

Summary

- React (also known as React.js or ReactJS) is a free and open-source front-end JavaScript library for building user interfaces based on UI components.
- Virtual Document Object Model, or virtual DOM. React creates an in-memory data-structure cache, computes the resulting differences, and then updates the browser's displayed DOM efficiently.
- ES6 or ECMAScript 6 is a scripting language specification which is standardized by ECMAScript International.
- Gitignore file tells Git which files to ignore when committing your project to the GitHub repository.

- Classes are an essential part of object-oriented programming (OOP). Classes are used to define the blueprint for real-world object modeling and organize the code into reusable and logical parts

Keywords

- **React:**React can be used as a base in the development of single-page or mobile applications
- **Const:** const is a variable that once it has been created, its value can never change.const has a block scope.
- **Let:**let is the block scoped version of var, and is limited to the block (or expression) where it is defined.
- **ES6 Arrays:**An array declaration allocates sequential memory blocks and is static. This means that an array once initialized cannot be resized.
- **Arrow Function**: Arrow functions are introduced in ES6 which provides you a more accurate way to write the functions in JavaScript.
- **Modules of es6**: JavaScript modules allow you to break up your code into separate files. This makes it easier to maintain the code-base.
- **Virtual DOM:**Virtual Document Object Model or virtual DOM. React creates an in-memory data-structure cache, computes the resulting differences, and then updates the browser's displayed DOM efficiently.

Self Assessment

1. Which of the following are the advantages of React.js?

A. React.js can increase the application's performance with Virtual DOM.
B. React.js is easy to integrate with other frameworks such as Angular, BackboneJS since it is only a view library.
C. React.js can render both on client and server side.
D. All of the above

2. Which of the following is not a disadvantage of React.js?

A. React.js has only a view layer. We have put your code for Ajax requests, events and so on.
B. The library of React.js is pretty large.
C. The JSX in React.js makes code easy to read and write.
D. The learning curve can be steep in React.js.

3. Which of the following command is used to install create-react-app?

A. npm install -g create-react-app
B. npm create-react-app
C. npm install -f react-app
D. install -g create-react-app

4. What of the following is used in React.js to increase performance?

A. Original DOM
B. Virtual DOM
C. Both A and B.
D. None of the above.

5. A class is a type of function, but instead of using the keyword function to initiate it, which keyword do we use?

A. Constructor

B. Class

C. Object

D. DataObject

6. Which of the following acts as the input of a class-based component?

A. Class

B. Factory

C. Render

D. Props

7. Which of the following keyword is used to create a class inheritance?

A. Create

B. Inherits

C. Extends

D. This

8. Which command is used to change directory?

A. Cd

B. Mkdir

C. Rm

D. Cm

9. Which command is used to make directory?

A. Cd

B. Mkdir

C. Rm

D. Cm

10. What are the two ways to handle data in React?

A. State & Props

B. Services & Components

C. State & Services

D. State & Component

11. This folder is the heart of React application as it contains JavaScript which needs to be processed by webpack

A. Src

B. Public

C. Node_modules

D. package-lock.json

12. This folder contains all dependencies and sub-dependencies specified in package.json used by React app. It contains more than 800 subfolders, this folder is automatically added in the .gitignore file.

A. Src

B. Public

C. Node_modules

D. package-lock.json

13. ____________ contain exact dependency tree to be installed in /node_modules.

A. Src

B. Public

C. Node_modules

D. package-lock.json

14. What does ES6 stand for?

A. ECMAScript 6

B. ECMA 6

C. ECMAJavaScript 6

D. EJavaScript 6

15. Which of the following function is used to change the state of the React.js component?

A. this.setState

B. this.setChangeState

C. this.State{}

D. None of the above.

Answers for Self Assessment

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

Review Questions

1. What is the virtual DOM?
2. What is the difference between the ES6 and ES5 standards?
3. What is an arrow function and how is it used in React?
4. What are the components in React?
5. What are the differences between class and functional components?
6. What are the advantages of using react?
7. What is the virtual DOM? How does react use the virtual DOM to render the UI?
8. What are the lifecycle methods going to be deprecated in React v16?
9. What are modules explain its types?
10. What objects are in react? Explain?

Further Readings

- HTMl, JavaScript , DHTML and PHP By. Ivan Bayross.
- Beginning JavaScript 2nd Edition By. Paul Wilton
- Html 5 Black Book, Covers Css 3, Javascript, Xml, Xhtml, Ajax, PhpAndJquery, Second

Edition

Web Links

- https://www.javascripttutorial.net/javascript-function/
- www.webopedia.com
- www.web-source.net

Unit 12:Pure React

CONTENTS

Objectives

After studying this unit, you will be able to:

- Discuss DOM and Its Features
- Discussed Framework and WebAPI
- Declaring React elements
- What is Rander Function and JSX

Introduction

A function is said to be pure if the return value is determined by its input values only and the return value is always the same for the same input values.

A React component is said to be pure if it renders the same output for the same state and props. For React pure class components, React provides the PureComponent base class. Class components that extend the React.PureComponent classes are treated as pure components.

Create React Application:

Step 1: Create a React application using the following command:

```
npx create-react-app pure-react
```

Step 2: After creating your project folder i.e. pure-react, move to it using the following command:

```
cd pure-react
```

Pure Components

Pure components have some performance improvements and render optimizations because React implements they shouldComponentUpdate () method for them with a shallow comparison for props and state.

Pure functional component in React

Functional components are very useful in React, especially when we want to isolate state management from the React component. That's why they are often called stateless components.

However, functional components cannot leverage the performance improvements and render optimizations that come with React.PureComponent since they are not classes by definition.

Optimizing a functional component, so that React can treat it as a pure component shouldn't necessarily require that the component be converted to a class component. To create a pure functional component in React, React provides a React.memo () API. Using the React.memo() API, the React functional component can be wrapped as follows to get React Pure Functional Component.

Notes: React. Memo () is a higher-order component that takes a React component as its first argument and returns a pure React component

12.1 Document Object Model

The Document Object Model (DOM) is a platform and language-neutral interface that allows programs and scripts to dynamically access and update the content, structure, and style of a document

The document object represents the whole html document. When html document is loaded in the browser, it becomes a document object. It is the root element that represents the html document. It has properties and methods. By the help of document object, we can add dynamic content to our web page.

Requirement of DOM

HTML is used to structure the web pages and JavaScript is used to add behavior to our web pages. When an HTML file is loaded into the browser, the JavaScript cannot understand the HTML document directly. So, a corresponding document is created (DOM). DOM is basically the representation of the same HTML document but in a different format with the use of objects.

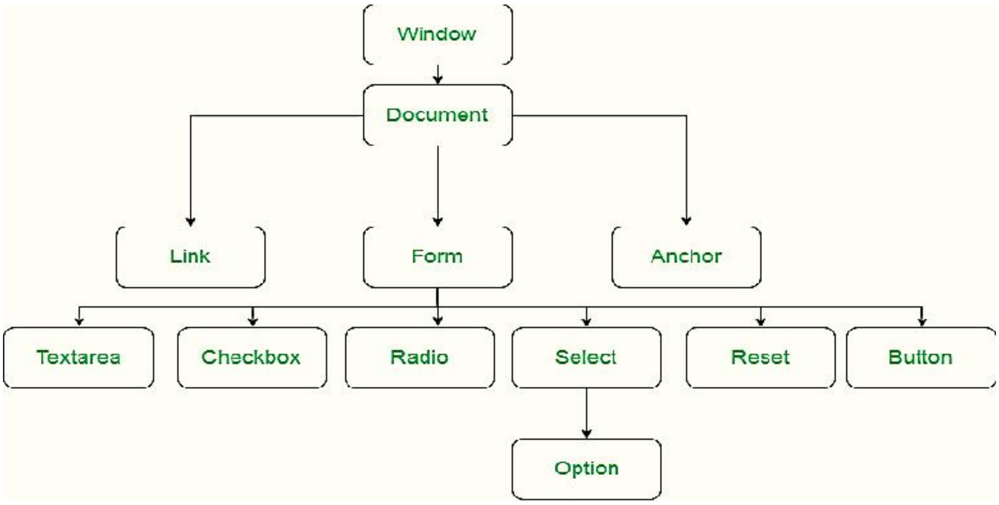
JavaScript interprets DOM easily i.e.JavaScript cannot understand the tags(<h1>H</h1>) in HTML document but can understand object h1 in DOM. Now, JavaScript can access each of the objects (h1, p, etc) by using different functions.

Object Model

Documents are modeled using objects, and the model includes not only the structure of a document but also the behavior of a document and the objects of which it is composed of like tag elements with attributes in HTML.

Properties of DOM

Let's see the properties of the document object that can be accessed and modified by the document object.



Properties of DOM

Window Object: Window Object is always at top of the hierarchy.

Document object: When an HTML document is loaded into a window, it becomes a document object.

Form Object: It is represented by form tags.

Link Object: It is represented by link tags.

Anchor Object: It is represented by a href tags.

Form Control Elements: Form can have many control elements such as text fields, buttons, radio buttons, and checkboxes, etc.

12.2 Framework and Web API

Software Framework designed to reduce overhead in web development. We have two types of framework Architecture

- Model-View-Controller (MVC)
- Push vs. Pull Based

Most MVC Frameworks user push-based architecture “action based” (Django, Ruby on Rails, Symfony, Stripes). Pull-based or “component based” (Lift, Angular2, React)

Three Tier Organizations

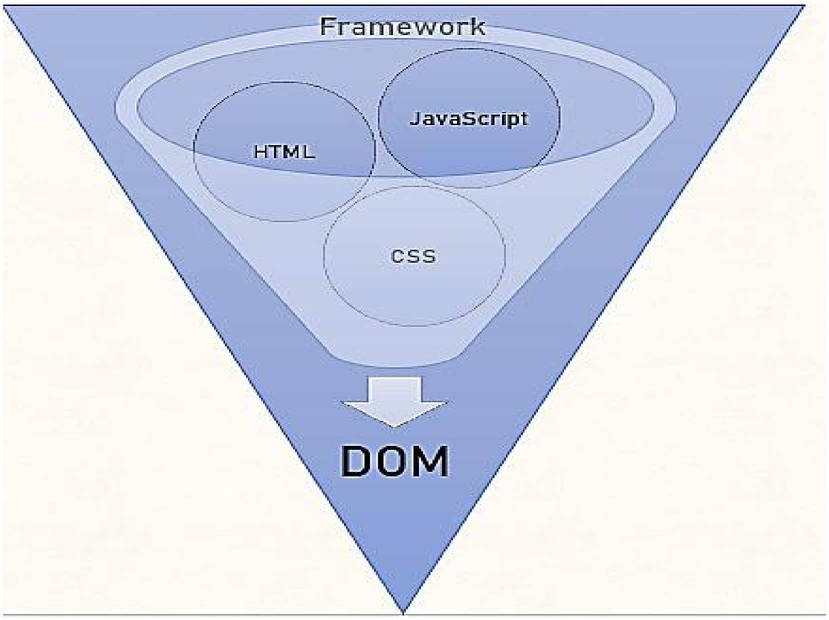
Client (Usually the browser running HTML/JavaScript/CSS)

Application (Running the Business Logic)

Database (Data Storage)

React has three type of server at Server Side: Django, Ruby on Rails

Client Side: Angular, React, Vue



Backend

All of the awesome that runs your application. Web API involved in running the app. Connection layer between the frontend and backend. They are connected through API calls (POST, GET, PUT, etc.). Transmit Content from the Backend to the Frontend commonly in JSON Blobs. Service Architecture that drives everything (Where all the logic is)

WebAPI

API stands for Application Programming Interface. It is an application programming interface for the Web. Browser API can also extend functionality of a web browser. The Server API can extend the functionality of a web server.

The intermediate layer between front end and back-end systems and it must have an APIswhich will be consumed by third-party service. We also have to recognize how consumable the API (Signature, content negotiation) is

12.3 Virtual DOM

The virtual DOM (VDOM) is a programming concept where an ideal, or "virtual", representation of a UI is kept in memory and synced with the "real" DOM by a library such as React DOM. This process is called reconciliation.

React uses Virtual DOM exists which is like a lightweight copy of the actual DOM(a virtual representation of the DOM). So for every object that exists in the original DOM, there is an object for that in React Virtual DOM.

It is exactly the same, but it does not have the power to directly change the layout of the document. Manipulating DOM is slow, but manipulating Virtual DOM is fast as nothing gets drawn on the screen.

So each time there is a change in the state of our application, virtual DOM gets updated first instead of the real DOM. You may still wonder, "Aren't we doing the same thing again and doubling our work? How can this be faster?" Read below to understand how things will be faster using virtual DOM.

Virtual DOM makes things faster

When anything new is added to the application, a virtual DOM is created and it is represented as a tree. Each element in the application is a node in this tree. So, whenever there is a change in state of any element, a new Virtual DOM tree is created.

This new Virtual DOM tree is then compared with the previous Virtual DOM tree and makes a note of the changes. After this, it finds the best possible ways to make these changes to the real DOM. Now only the updated elements will get rendered on the page again.

Virtual DOM helpful in React

In react, everything is treated as a component be it a functional component or class component. A component can contain a state. Each time we change something in our JSX file or let's put it in simple terms, whenever the state of any component is changed react updates it's Virtual DOM tree.

Though it may sound that it is ineffective but the cost is not much significant as updating the virtual DOM doesn't take much time. React maintains two Virtual DOM at each time, one contains the updated Virtual DOM and one which is just the pre-update version of this updated Virtual DOM.

It just means that the changes to the real DOM are sent in batches instead of sending any update for a single change in the state of a component.

We have seen that the re-rendering of the UI is the most expensive part and React manages to do this most efficiently by ensuring that the Real DOM receives batch updates to re-render the UI. This entire process of transforming changes to the real DOM is called Reconciliation

Notes: React Element are the smallest building blocks of React apps. An element specifies what should be there in our UI. An Element is a plain object describing what we want to appear in terms of the DOM nodes.

12.4 React Element

Elements are the smallest building blocks of React apps. An element specifies what should be there in our UI. An Element is a plain object describing what we want to appear in terms of the DOM nodes.

Creating a React Element is Cheap compared to DOM elements. An Element can be Created by using JSX or React without JSX

For Example:

```
Const element = <h1>Hello, world</h1>;
```

Object of type h1 and assigned it to a variable called as element. This element should be rendered into the Browser DOM, and for that we need a container.

In React we have two elements

Reacts Element - It is a simple object that describes a DOM node and its attributes or properties you can say. It is an immutable description object and you cannot apply any methods on it.

Reacts Component - It is a function or class that accepts an input and returns a React element.

12.5 Render Function

React's goal is in many ways to render HTML in a web page. React renders HTML to the web page by using a function called React DOM.render (). The React DOM.render () function takes two arguments which has HTML code and an HTML element. The purpose of the function is to display the specified HTML code inside the specified HTML element.

There is another folder in the root directory of your React project, named "public". In this folder, there is an index.html file.

You'll notice a single <div> in the body of this file. This is where our React application will be rendered.

Example to Display a paragraph inside an element with the id of "root":

```
React DOM.render(<p>Hello</p>, document.GetElementById('root'));
```

The result is displayed in the <div id="root"> element:

```
<body>
<div id="root"></div>
</body>
```

12.6 JSX

JSX stands for JavaScript XML. It allows us to write HTML in Reactand makes it easier to write and add HTML.It is easy to make a template using JSX in React, but it's not an easy template language instead it comes with the complete power of JavaScript.It is faster than normal JavaScript because it performs optimizations while translating to regular JavaScript. Rather than separating the markup and logic in separated files, React uses components for this purpose.

Coding in JSX

JSX allows us to write HTML elements in JavaScript and place them in the DOM without any create Element() and/or appendChild () methods. It converts HTML tags into react elements. It is not required to use JSX, but JSX makes it easier to write React applications.

Here are two examples. The first uses JSX and the second do not:

```
Const myelement = <h1>I Love JSX!</h1>;

ReactDOM.render(myelement,document.GetElementById('root'));

Const myelement = React.createElement('h1', {}, 'I do not use JSX!');

ReactDOM.render (myelement, document.GetElementById('root'));
```

Expression in JSX

With JSX you can write expressions inside curly braces { }.The expression can be a React variable, or property, or any other valid JavaScript expression. JSX will execute the expression and return the result

Example:

Execute the expression 5 + 5:

```
Const myelement = <h1>React is {5 + 5} times better with JSX</h1>;
```

Reacts Components

Components are like functions that return HTML elements. They are independent and reusable bits of code. They serve the same purpose as JavaScript functions, but work in isolation and return HTML. React components come in two types, Class components and Function components.

When creating a React component, the component's name MUST start with an upper-case letter.A class component must include the extends React.Component statement is creating an inheritance to react. They give your component access to react.Component's functionsalso requires a render () method, this method returns HTML.

Create a Class component called Car

```
Class Car extends React. Component
 {
render () {
Return<h2>Hi, I am a Car!</h2>;
  }
}
```

Function Component

A Function component also returns HTML, and behaves much the same way as a Class component, but Function components can be written using much less code, are easier to understand.

Create a Function component called Car

```
FunctionCar ()
{
Return<h2>Hi, I am a Car!</h2>;
}
```

Rendering Components

React application has a component called Car, which returns an <h2>element. To use this component in your application, use similar syntax as normal HTML:

```
<Car />
```

Characteristics of React

Flexible

React is remarkably flexible. Once you have learned it, you can use it on a vast variety of platforms to build quality user interfaces. React is a library, NOT a framework. Its library approach has allowed React to evolve into such a remarkable tool.

React was created with a single focus: to create components for web applications. A React component can be anything in your web application like a Button, Text, Label, or Grid.

It has a great developer experience when we start coding in it. Rapid development and React's small API combined creates a fantastic developer experience.

React's API is very simple to learn. It has very few concepts to learn.

Facebook Support Resources

React is heavily used in the Facebook app, website, and in Instagram. That's why Facebook is deeply committed to it. They use over 50k React components in their production environment. The top four React contributors on GitHub are full-time Facebook employees. the React team maintains a blog that consistently gives you details for each release and because of the deep commitment by Facebook to

React in production, when breaking change occur in React, Facebook consistently provides Codemod that automates the change.React's popularity has grown steadily. It has a massive active community and its GitHub Repository has over 164k Stars. It is one of the Top 5 Repositories on GitHub.

Performance

The React team realized that JavaScript is fast, but updating the DOM makes it slow. React minimizes DOM changes. And it has figured out the most efficient and intelligent way to update DOM.

Before React, most frameworks and libraries would update the DOM unintelligently to reflect a new state. This resulted in changes to a significant portion of the page.

Ease of Testing

React's design is very user friendly for testing.Traditional UI browser testing is a hassle to setup. On the other hand, you require very little or no configuration for testing in React.Traditional UI browser requires browsers for testing, but you can test React components quickly and easily using the node command-line. Browser testing is slow. But command-line testing is fast, and you can run a considerable amount of test suites at a time.Traditional UI browser testing is often time consuming and challenging to maintain. React test can be written quickly using tools like Jest & Enzyme.

React is an excellent tool with which to create interactive applications for mobile, web, and other platforms.It popularity and usage are increasing day by day for good reason. As a developer, coding in React makes you better at JavaScript, a language that holds nearly 90% of the web development share today.

Summary

- React component is said to be pure if it renders the same output for the same state and props.
- Pure components have some performance improvements and render optimizations.
- Functional components are very useful in React, especially when we want to isolate state management from the React component.
- Components are like functions that return HTML elements
- React uses Virtual DOM exists which is like a lightweight copy of the actual DOM(a virtual representation of the DOM).
- Web API involved in running the app. Connection layer between the frontend and backend.
- A Function component also returns HTML, and behaves much the same way as a Class component, but Function components can be written using much less code, are easier to understand

Keywords

- **DOM:**Document Object Model (DOM) is a platform and language-neutral interface that allows programs and scripts to dynamically access and update the content, structure, and style of a document
- **Pure React:** A function is said to be pure if the return value is determined by its input values only and the return value is always the same for the same input values
- **Object Model:**Documents are modeled using objects, and the model includes not only the structure of a document but also the behavior of a document and the objects of which it is composed of like tag elements with attributes in HTML.
- **JSX:**JSX stands for JavaScript XML. It allows us to write HTML in React and makes it easier to write and add HTML.
- **Render Function**:React renders HTML to the web page by using a function called ReactDOM.render ().
- **Virtual DOM**: React uses Virtual DOM exists which is like a lightweight copy of the actual DOM (a virtual representation of the DOM).
- **Framework**: Software Framework designed to reduce overhead in web development
- **Pure components:**It has some performance improvements and render optimizations because React implements the shouldComponentUpdate() method for them with a shallow comparison for props and state.

Self Assessment

1. What are core components of Web applications?

A. UI (Front End (DOM, Framework))
B. Request Layer (Web API)
C. Back End (Database, Logic)
D. All of the above

2. _______makes every addressable item in a web application an Object that can be manipulated for color, transparency, position, sound and behaviors.

A. Document Object Model
B. Document adjust Model
C. Document Object
D. All of the above

3. In properties of DOM which object is at top of hierarchy:

A. Window Object
B. Document object
C. Form Object
D. Link Object

4. When an HTML document is loaded into a window, it becomes a______

A. Window Object
B. Document object
C. Form Object
D. Link Object

5. Which object is represented by link tags?
A. Window Object
B. Document object
C. Form Object
D. Link Object

6. What is full form of MVC
A. Main View Controller
B. Mode View Controller
C. Model View Controller
D. None of the above

7. In MVC what handles user input and interactions?
A. Model
B. View
C. Control
D. All of the above

8. In MVC Where changes to the page are rendered and displayed
A. Model
B. View
C. Control
D. All of the above

9. _________is a programming concept where an ideal, or “virtual”, representation of a UI is kept in memory and synced with the “real” DOM by a library such as ReactDOM
A. The virtual DOM (VDOM)
B. Actual Dom
C. Dom
D. None of the above

10. ReactJS uses _____ to increase performance?
A. The virtual DOM (VDOM)
B. Actual Dom
C. Dom
D. None of the above

11. React renders HTML to the web page by using a function called ___
A. ReactDOM.render().
B. DOM.render().
C. ReactDOM().
D. All of the above

12. The ReactDOM.render() function takes arguments:
A. HTML code
B. HTML element.
C. Both of above
D. None of the above

13. JSX stands for _________

A. JavaScript XML.

B. JavaScript.

C. XML.

D. None of above.

14. ________________are independent and reusable bits of code.

A. Components

B. Properties

C. States

D. Classes

15. Which component includes extends react.component statement

A. Function

B. Class

C. Both of the above

D. None of the above

Answers for Self Assessment

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

Review Questions

1. What is the virtual DOM?
2. What is difference between Pure React and React?
3. What is JSX and how it works?
4. What are the differences between functional and class components?
5. What is the virtual DOM? How does react use the virtual DOM to render the UI?
6. What are the differences between controlled and uncontrolled components?
7. Explain about types of side effects in React component
8. What are the rendering characteristics explain?
9. What are WebAPI? Explain in detail?
10. What is framework and how it is used in React?

Further Readings

- HTMl, JavaScript , DHTML and PHP By. Ivan Bayross.
- Beginning JavaScript 2nd Edition By. Paul Wilton
- Html 5 Black Book, Covers Css 3, Javascript, Xml, Xhtml, Ajax, PhpAndJquery, Second

Edition

Web Links

- https://www.javascripttutorial.net/javascript-function/
- www.webopedia.com
- www.web-source.net

Unit 13: Prop State and the Component Tree

CONTENTS

Objectives

After studying this unit, you will be able to:

- Discuss Prop, State and the Components
- Discussed Property validation
- React stage-management working
- What is state within the component Tree

Introduction

Components are like functions that return HTML elements. They are independent and reusable bits of code. They serve the same purpose as JavaScript functions, but work in isolation and return HTML.Components come in two types, Class components and Function components.Components make the task of building UIs much easier. As UI broken down into multiple individual pieces called components and work on them independently and merge them all in a parent component which will be your final UI.

Components in React basically return a piece of JSX code by which it tells what should be rendered on the screen.

React has two types of components:

Functional Components: They are simply JavaScript functions. Function component can be created We can create a functional component in React by writing a JavaScript function. These functions may or may not receive data as parameters.

Example shows a valid functional component in React:

Constgood component=()=>

{

Return<h1>Hello World</h1>;

}

Class Components: The class components are a little more complex than the functional components. The functional components are not aware of the other components in your program whereas the class components can work with each other. We can pass data from one class component to other class components.

Example shows a valid class-based component in React:

Classgood component extends React.Component

{

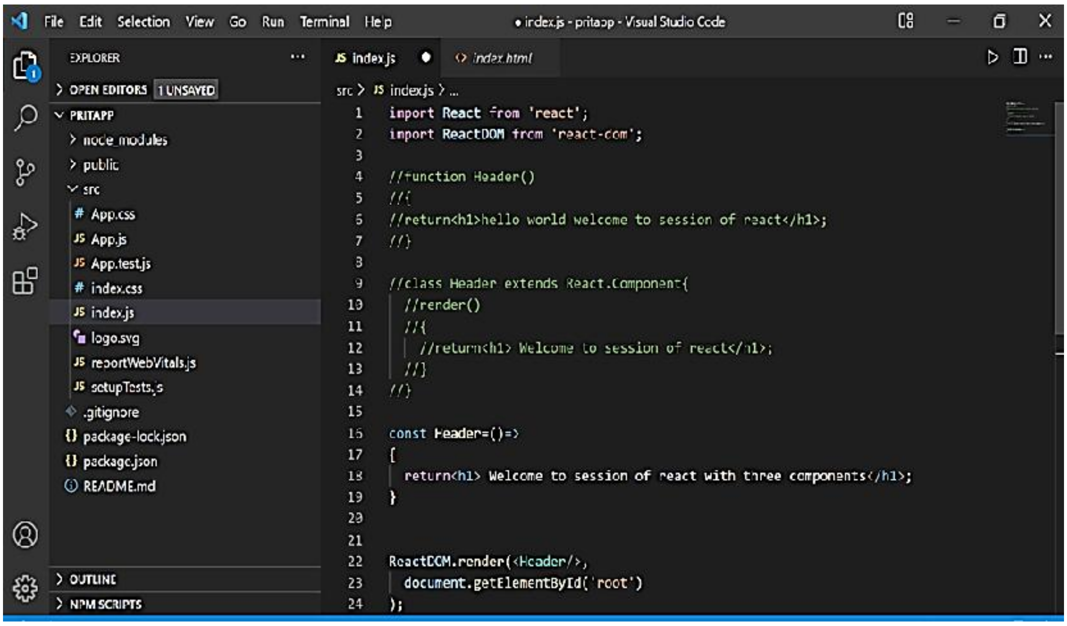
Render ()

{

Return<h1>Hello World</h1>;

}

}



```
import React from 'react';
import ReactDOM from 'react-dom';

//function Header()
//{
//return<h1>hello world welcome to session of react</h1>;
//}

//class Header extends React.Component{
  //render()
  //{
    //return<h1> Welcome to session of react</h1>;
  //}
//}

const Header=()=>
{
  return<h1> Welcome to session of react with three components</h1>;
}


ReactDOM.render(<Header/>,
  document.getElementById('root')
);
```

Notes: Props are immutable so we cannot modify the props from inside the component. Inside the components, we can add attributes called props.

13.1 Props &Property Validation

Props stand for "Properties." They are read-only components. It is an object which stores the value of attributes of a tag and work similar to the HTML attributes.It gives a way to pass data from one component to other components. It is similar to function arguments. Props are passed to the component in the same way as arguments passed in a function.

Props are immutable so we cannot modify the props from inside the component. Inside the components, we can add attributes called props. These attributes are available in the component as this.Props and can be used to render dynamic data in our render method.

When you need immutable data in the component, you have to add props to reactDom.render() method in the main.js file of your ReactJS project and used it inside the component in which you need.

Props can be used to pass any kind of data such as:

String

Array

Integer

Boolean

Object

Function

To send props into a component, use the same syntax as HTML attributes:

Example

Add a "brand" attribute to the Car element:

```
Const myelement = <Car brand="Ford" />;
```

- The component receives the argument as a props object:
- Example
    - Use the brand attribute in the component:

```
Function Car(props) {
Return<h2>I am a { props.brand }!</h2>;
}
```

Props are also how you pass data from one component to another, as parameters.

Example: Send the "brand" property from the Garage component to the Car component:

```
Function Car (props) {
 Return <h2>I am a {props. brand }!</h2>;
}
Function Garage () {
 Return (
<>
<h1>Who lives in my garage? </h1>
<Car brand="Ford" />
</>
 );
}
ReactDOM.render(<Garage />, document.getElementById('root'));
```

13.2 Property Validation

Properties validation is a useful way to force the correct usage of the components. This will help during development to avoid future bugs and problems, once the app becomes larger. It also makes the code more readable, since we can see how each component should be used.

Need of Validating Props in React JS

Props are used to passing the read-only attributes to react components. For the proper functioning of components and to avoid future bugs and glitches it is necessary that props are passed correctly. Hence, it is required to use props validation for improving react component's performance.

React JS has an inbuilt feature for validating props data type to make sure that values passed through props are valid. React components have a property called prop Types which is used to setup data type validation.

The syntax to use prop Types is shown below.

```
Class Component extends React. Component {
Render () {}
}
Component.propTypes = {/* definition goes here*/};
```

Validators

PropTypes.symbol	This means the prop should be a symbol.
PropTypes.instanceOf	This means the prop should be an instance of a particular JavaScript class.
PropTypes.isRequired	This means the prop should be provided.
PropTypes.oneOf()	This means the props should be one of several types of specified values.
PropTypes.element	This means the props must be an element.

Use of Prop Types in React

Prop Types are a mechanism to ensure that components use the correct data type and pass the right data, and that components use the right type of props, and that receiving components receive the right type of props.

What Do You Need to Know About React Web Development

Netflix is a very good React website example; as well as Salesforce, Yahoo Mail, Flipboard, and many React examples. Some of these were early adopters while others joined in late. What's common to all of them is that, at some point, they took a decision to switch from a previous tech stack. Then, we need to ask, what makes this JS library stand out?

It is one of the most popular open-source JavaScript libraries today that prides itself upon enabling developers to build interactive user interfaces. Thus, there are popular websites built with React in abundance. It was originally developed as an internal technology for use within Facebook but became open-source in 2013.

13.3 State Management

State

State represents the value of dynamic properties of a React component at a given instance. React provides a dynamic data store for each component. The internal data represents the state of a React component and can be accessed using this. state member variable of the component. Whenever the state of the component is changed, the component will re-render itself by calling the render () method along with the new state.

A simple example to better understand the state management is to analyze a real-time clock component. The clock component primary job is to show the date and time of a location at the given instance. As the current time will change every second, the clock component should maintain the current date and time in it's state. As the state of the clock component changes every second, the clock's render () method will be called every second and the render() method show the current time using its current state.

State Management

React components have a built-in state object. The state is encapsulated data where you store assets that are persistent between component renderings.

The state is just a fancy term for a JavaScript data structure. If a user changes state by interacting with your application, the UI may look completely different afterwards, because it's represented by this new state rather than the old state.

Need of State Management

React applications are built using components and they manage their state internally and it works well for applications with few components, but when the application grows bigger, the complexity of managing states shared across components becomes difficult.

Here is a simple example of an e-commerce application, in which the status of multiple components will change when purchasing a product.

Add that product to the shopping list

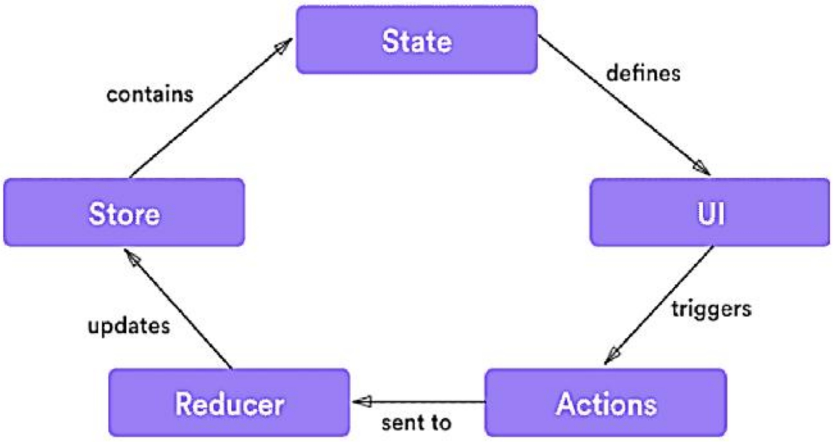
Add product to customer history

Trigger count of purchased products

If developers do not have scalability in mind, then it is really hard to find out what is happening when something goes wrong. This is why you need state management in your application.

Redux

Redux was created to resolve this particular issue. it provides a central store that holds all states of your application. Each component can access the stored state without sending it from one component to another.



Three Building Parts

Action

Store

Reducers

Actions are payloads of information that send data from your application to your store. Actions are sent using store.Dispatch ().Actions are created via an action creator.

Reducers in Redux

Reducers specify how the application's state changes in response to actions sent to the store. The store holds the application state. You can access stored state, update the state, and register or unregister listeners via helper methods.Redux gives you code organization and debugging superpowers. This makes it easier to build more maintainable code, and much easier to track down the root cause when something goes wrong.

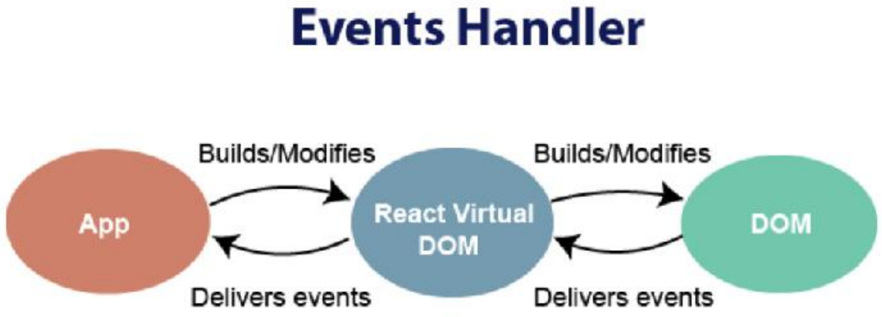
React Hook

These are functions that hook you into React state and features from function components. Hooks don't work inside classes and it allows you to use React features without writing a class.

Hooks are backwards-compatible, which means it doesn't keep any breaking changes. React provides some built-in Hooks like useState, UseEffect and useReducer etc. You can also make custom hooks.

13.3.1 React Events

An event is an action that could be triggered as a result of the user action or system generated event. For example, a mouse click, loading of a web page, pressing a key, window resizes, and other interactions are called events. React has its own event handling system which is very similar to handling events on DOM elements. The react event handling system is known as Synthetic Events. The synthetic event is a cross-browser wrapper of the browser's native event.



Handling events with react have some differences from handling events on DOM. These are:

- React events are named as camel Case instead of lowercase.
- With JSX, a function is passed as the event handler instead of a string

Event Declaration in Plain HTML:

```
<button on click="showMessage ()">
   Hello JavaTpoint
</button>
```

Event declaration in React:

```
<button onClick={showMessage}>
   Hello JavaTpoint
</button>
```

Events can be added in React. They can be written in camel Case syntax

- OnClick instead of onclick.
- React event handlers are written inside curly braces:
- onClick={shoot} instead of onClick="shoot()"

Put the shoot function inside the Football component:

```
FunctionFootball ()
 {
Const shoot = () => {
alert ("Great Shot!");
 }
 Return (
 <button onClick= {shoot}>Take the shot!</button>
   );
 }
 ReactDOM.render(<Football />, document.getElementById('root'));
```

Passing Arguments in React

- To pass an argument to an event handler, use an arrow function.
- Send "Goal!" as a parameter to the shoot function, using arrow function:

```
FunctionFootball ()
{
Const shoot = (a) =>
 {
Alert (a);
 }
Return
(
<button on Click= {() =>shoot ("Goal!")}>Take the shot!</button>
 );
}
ReactDOM.render(<Football />, document.getElementById('root'));
```

13.4 React Conditional Rendering

In React, you can conditionally render components and we have several ways to do by using if statement and using some JavaScript operator to decide which component to render

If else statement

An if…else statement will execute the actions contained in the if block when the condition is satisfied. Otherwise, it will execute the actions contained in the else block.

In JSX, you are able to use JavaScript code with markup to render dynamic values within your application. JSX uses curly braces ({ and }) to signify expressions that need to be interpreted prior to rendering. The caveat, however, is that there is a limit to what can be done within such braces.

```
Import React from 'react';
Import ReactDOM from 'react-dom/client';


Functionhello () {
        Return<h1>Hello!</h1>;
}
```

```
Functionworld () {
        Return<h1>there!</h1>;
}


function day(props) {
Constisday = props.isday;
 if (isday) {
Return<There/>;
 }
Return<Hello/>;
}
```

Switch Statement

You can conditionally return different markup from a component based on set conditions using an if…else statement. The same could be achieved with a switch statement where you can specify the markup for various conditions.

Ternary Operator

The conditional (ternary) operator is the only JavaScript operator that takes three operands. This operator is frequently used as a shortcut for the if statement

13.5 React Lists

Lists are used to display data in an ordered format and mainly used to display menus on websites. In React, Lists can be created in a similar way as we create lists in JavaScript.In React, you will render lists with some type of loop.The JavaScript map() array method is generally the preferred method.

We can render lists with some type of loop.The JavaScript map () array method is generally the preferred method.

```
FunctionCar (props)
 {
Return<li>I am a { props.brand }</li>;}
FunctionGarage () {const cars = ['Ford', 'BMW', 'Audi'];
Return (
<><h1>who lives in my garage? </h1>
<ul>
    {cars.map ((car) =><Car brand={car} />)}
</ul>
</>
 );
}
```

React Forms

Forms are an integral part of any modern web application. It allows the users to interact with the application as well as gather information from the users. Forms can perform many tasks that depend on the nature of your business requirements and logic such as authentication of the user, adding user, searching, filtering, booking, ordering, etc. A form can contain text fields, buttons, checkbox, radio button, etc.

Creating Forms

React offers a stateful, reactive approach to build a form. The component rather than the DOM usually handles the React form. In React, the form is usually implemented by using controlled components.

There are mainly two types of form input in React

Uncontrolled Component- The uncontrolled input is similar to the traditional HTML form inputs.

The DOM itself handles the form data. The HTML elements maintain their own state that will be updated when the input value changes. To write an uncontrolled component, you need to use a ref to get form values from the DOM. In other words, there is no need to write an event handler for every state update. We can use a ref to access the input field value of the form from the DOM.

Notes: To write an uncontrolled component, you need to use a ref to get form values from the DOM.

Controlled Component-Controlled components have functions that govern the data passing into them on every on-Change event, rather than grabbing the data only once, e.g., when you click a submit button. This data is then saved to state and updated with set State () method. This makes component have better control over the form elements and data. A controlled component takes its current value through props and notifies the changes through callbacks like an on-Change event.

A parent component "controls" these changes by handling the callback and managing its own state and then passing the new values as props to the controlled component. It is also called as a "dumb component."

Adding forms in React

```
FunctionMy Form () {
  return (
<Form>
<Label>Enter your name:
<input type="text" />
</label>
</form>
  )
}
ReactDOM.render(<MyForm />, document.getElementById('root'));
```

Handling Forms

Handling forms is about how you handle the data when it changes value or gets submitted.In HTML, form data is usually handled by the DOM.In React, form data is usually handled by the components.

13.6 React Lists

Lists are used to display data in an ordered format and mainly used to display menus on websites. In React, Lists can be created in a similar way as we create lists in JavaScript. In React, you will render lists with some type of loop.The JavaScript map() array method is generally the preferred method.

Example:

```
Function Car (props) {
  Return <li>I am a { props. brand }</li>;}
Function Garage() { const cars = ['Ford', 'BMW', 'Audi'];
```

```
  Return (
<><h1>who lives in my garage? </h1>
<ul>
      {cars.map((car) =><Car brand={car} />)}
</ul>
</>
  );
}
```

13.7 React Forms

Forms are an integral part of any modern web application. It allows the users to interact with the application as well as gather information from the users. Forms can perform many tasks that depend on the nature of your business requirements and logic such as authentication of the user, adding user, searching, filtering, booking, ordering, etc. A form can contain text fields, buttons, checkbox, radio button, etc.

Creating Form

React offers a stateful, reactive approach to build a form. The component rather than the DOM usually handles the React form. In React, the form is usually implemented by using controlled components.

There are mainly two types of form input in React.

- Uncontrolled Component
- Controlled Component

Uncontrolled Component- The uncontrolled input is similar to the traditional HTML form inputs.

The DOM itself handles the form data. The HTML elements maintain their own state that will be updated when the input value changes. To write an uncontrolled component, you need to use a ref to get form values from the DOM. In other words, there is no need to write an event handler for every state update. You can use a ref to access the input field value of the form from the DOM.

Controlled Component- In HTML, form elements typically maintain their own state and update it according to the user input. In the controlled component, the input form element is handled by the component rather than the DOM.The mutable state is kept in the state property and will be updated only with set State () method. Controlled components have functions that govern the data passing into them on every on Change event, rather than grabbing the data only once, e.g., when you click a submit button. This data is then saved to state and updated with set State () method. This makes component have better control over the form elements and data.

A controlled component takes its current value through props and notifies the changes through callbacks like an on Change event. A parent component "controls" this changes by handling the callback and managing its own state and then passing the new values as props to the controlled component. It is also called as a "dumb component."

Example of Forms

```
FunctionMy Form () {
Return (
<Form>
<Label>Enter your name:
<input type="text" />
</label>
</form>
```

```
 )
}
ReactDOM.render(<MyForm />, document.getElementById('root'));
```

13.8 Handling Forms

Handling forms is about how you handle the data when it changes value or gets submitted.In HTML, form data is usually handled by the DOM.In React, form data is usually handled by the components.

When the data is handled by the components, all the data is stored in the component state.You can control changes by adding event handlers in the onChange attribute.We can use the useState Hook to keep track of each inputs value and provide a "single source of truth" for the entire application.

Summary

- Components are like functions that return HTML elements. They are independent and reusable bits of code.
- Properties validation is a useful way to force the correct usage of the components.
- Handling events with react have some differences from handling events on DOM
- Forms can perform many tasks that depend on the nature of your business requirements and logic such as authentication of the user, adding user, searching, filtering, booking, and ordering.
- React offers a stateful, reactive approach to build a form. The component rather than the DOM usually handles the React form.
- React components have a built-in state object. The state is encapsulated data where you store assets that are persistent between component renderings.
- Handling forms is about how you handle the data when it changes value or gets submitted.

Keywords

- **Component:**Components in React basically return a piece of JSX code by which it tells what should be rendered on the screen.
- **Functional Components:** They are simply JavaScript functions. Function component can be created
- **Class Components:** The class components are a little more complex than the functional components.
- **Props:** They are read-only components. It is an object which stores the value of attributes of a tag and work similar to the HTML attributes.
- **React Forms:** Forms are an integral part of any modern web application. It allows the users to interact with the application as well as gather information from the users.
- **Handling Forms:**Handling forms is about how you handle the data when it changes value or gets submitted.
- **Uncontrolled Components:** The uncontrolled input is similar to the traditional HTML form inputs.
- **Handling Forms:** Handling forms is about how you handle the data when it changes value or gets submitted.

Self Assessment

1. Which component required extending react. component …….
A. Class component
B. Function component
C. Both a and b
D. None of these

2. Which component requires rendering ……….

A. Class component

B. Function component

C. Both a and b

D. None of these

3. Which of the following acts as the input of a class-based component?

A. Class

B. Props

C. Factory

D. None of the mentioned

4. React.js is written in which of the following language?
A. C
B. C++
C. JavaScript
D. Java

5. How many numbers of elements a valid react component can return?
A. 1
B. 2
C. 4
D. 5

6.____are arguments passed into React components
A. Class
B. Props
C. Factory
D. None of the mentioned

7. How many ways of defining your variables in ES6?

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

8. What is a state in React?
A. A permanent storage.
B. Internal storage of the component.
C. External storage of the component.
D. None of the above

9. What are the two ways to handle data in React?

A. State & Props

B. Services & Components

C. State & Services

D. State & Component

10. Props can be used to pass
A. String
B. Array
C. Integer
D. All of above

11. ______are a new feature added in React v16.8. It allows to use all React features without writing class components.

A. Hooks

B. Props

C. State

D. All of above

12. __________ is a Javascript framework built using Typescript, while Reactjs is a Javascript library and built using JSX.

A. Angular

B. Pro

C. Milt

D. All of the above

13. The conditional operator is the only JavaScript operator that takes three operands.

A. Ternary

B. Primary

C. Tertiary

D. All of the above

14. ________are used to display data in an ordered format and mainly used to display menus on websites

A. Lists

B. Props

C. Component

D. None of above

15. ______are an integral part of any modern web application. It allows the users to interact with the application as well as gather information from the users.

A. Lists

B. Forms

C. Component

D. None of above

Answers for Self Assessment

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

Review Questions

1. What is the Component? Explain its Types?
2. Explain how lists work in React
3. How do you create forms in React? Explain with example?
4. How do you implement state in React? How do you update the state of a component?
5. How can you embed two or more components into one?
6. What do you mean by State Management? Explain its components?
7. How to Handle Forms? Explain the creation of forms?
8. What are the differences between state and props?
9. What are the differences between class and functional components?
10. Explain the lifecycle methods of components.

Further Readings

- HTMl, JavaScript , DHTML and PHP By. Ivan Bayross.
- Beginning JavaScript 2nd Edition By. Paul Wilton
- Html 5 Black Book, Covers Css 3, Javascript, Xml, Xhtml, Ajax, PhpAndJquery, Second Edition

Web Links

- https://www.javascripttutorial.net/javascript-function/
- www.webopedia.com
- www.web-source.net

Unit 14: React Router and Server

CONTENTS

Objectives

After studying this unit, you will be able to:

- Discuss Prop, State and the Components
- Discussed Property validation
- React stage-management working
- What is state within the component Tree

Introduction

Routing is a process of binding a web URL to a specific resource in the web application. It is the binding of a URL to a component in React. React does not support routing natively as it is basically an user interface library. React community provides many third-partycomponents to handle routing in the React application. Let us learn React Router, a top choice routing library for React application. Components in React basically return a piece of JSX code by which it tells what should be rendered on the screen. React has two types of components:

14.1 Router

Routing is a process in which a user is directed to different pages based on their action or request. ReactJS Router is mainly used for developing Single Page Web Applications. React Router is used to define multiple routes in the application. When a user types a specific URL into the browser and if this URL path matches any 'route' inside the router file, the user will be redirected to that particular route. React Router is a standard library system built on top of the React and used to create routing in the React application using React Router Package.

It provides the synchronous URL on the browser with data that will be displayed on the web page. It maintains the standard structure and behavior of the application and mainly used for developing single page web applications.

Need of React Router

React Router plays an important role to display multiple views in a single page application.Without React Router, it is not possible to display multiple views in React applications. Most of the social media websites like Facebook, Instagram uses React Router for rendering multiple views.

Router Installation

React contains three different packages for routing.

- **react-router**: It provides the core routing components and functions for the React Router applications.
- **react-router-native**: It is used for mobile applications.
- react-router-dom: It is used for web applications design.

It is not possible to install react-router directly in your application. To use react routing, first, you need to install react-router-dom modules in your application. The below command is used to install react router dom.$ npm install react-router-dom --save

Components of React Router

The BrowserRouter, Route, Switch and Link are all components of the React-Router.These components are divided into three categories.The first category is routers, for example <BrowserRouter>. The second category is route matchers, such as <Route> and <Switch> and the last category is navigation, such as <Link>, and <Redirect>

<BrowserRouter>

BrowerRouter is a router implementation that has the ability to incorporate routing in react. It uses the HTML5 History API which include pushState, replaceState and the popState event to keep our UI in sync with the URL.It is the parent component that is used to store all other components and it uses regular URL paths.

<Route>

Route is the conditional component that renders a component based on the URL defined or the URL it is pointing to. In other words, it is a component that renders some UI when its path matches the current URL.

<Link>

Link component is used to create links to different routes and implements navigation around the application. Links accepts the to prop, which signifies where we want the link to navigate our user to.

<Switch>

The switch component is used to render only the first route that matches the location rather than rendering all matching routes.

Notes: Link in React is similar to anchor tag in html. It sets the target url along with reference text.

React Router and Client-Side Routing

React Router is an API for React applications. Most current code is written with React Router 3, although version 4 has been released. React Router uses dynamic routing.

When we say dynamic routing, we mean routing that takes place as your app is rendering, not in a configuration or convention outside of a running app. That means almost everything is a component in React Router.

14.2 Use of React Router

React Router, and dynamic, client-side routing, allows us to build a single-page web application with navigation without the page refreshing as the user navigates. React Router uses component structure to call components, which display the appropriate information.

By preventing a page refresh, and using Router or Link, which is explained in more depth below, the flash of a white screen or blank page is prevented.This is one increasingly common way of having a more seamless user experience. React router also allows the user to utilize browser functionality like the back button and the refresh page while maintaining the correct view of the application.

To send props into a component, use the same syntax as HTML attributes:

Example:

Add a "brand" attribute to the Car element:

```
const myelement = <Car brand="Ford" />;
```

- The component receives the argument as a props object:
- Example
  - Use the brand attribute in the component:

```
function Car(props) {
  return <h2>I am a { props.brand }!</h2>;
}
```

Props are also how you pass data from one component to another, as parameters.

Example

Send the "brand" property from the Garage component to the Car component:

```
function Car(props) {
  return <h2>I am a { props.brand }!</h2>;
}
function Garage() {
  return (
<>
<h1>Who lives in my garage?</h1>
<Car brand="Ford" />
</>
  );
}
ReactDOM.render(<Garage />, document.getElementById('root'));
```

14.3 Installation of React Router

React Router plays an important role to display multiple views in a single page application.Without React Router, it is not possible to display multiple views in React applications. Most of the social media websites like Facebook, Instagram uses React Router for rendering multiple views.

React contains three different packages for routing.

react-router: It provides the core routing components and functions for the React Router applications.

react-router-native: It is used for mobile applications.

react-router-dom: It is used for web applications design.

It is not possible to install react-router directly in your application. To use react routing, first, you need to install react-router-dom modules in your application. The below command is used to install react router dom.$ npm install react-router-dom --save

Components of React Router

The BrowserRouter, Route, Switch and Link are all components of the React-Router. These components are divided into three categories.The first category is routers, for example <BrowserRouter>. The second category is route matchers, such as <Route> and <Switch> and the last category is navigation, such as <Link>, and <Redirect>

<BrowserRouter>:

BrowerRouter is a router implementation that has the ability to incorporate routing in react.It uses the HTML5 History API which include pushState, replaceState and the popState event to keep our UI in sync with the URL.It is the parent component that is used to store all other components and it uses regular URL paths.

<Route>:

Route is the conditional component that renders a component based on the URL defined or the URL it is pointing to.In other words, it is a component that renders some UI when its path matches the current URL.

<Link>:

Link component is used to create links to different routes and implements navigation around the application. Links accepts the to prop, which signifies where we want the link to navigate our user to.

<Switch>:

The switch component is used to render only the first route that matches the location rather than rendering all matching routes.

React Router and Client-Side Routing

React Router is an API for React applications. Most current code is written with React Router 3, although version 4 has been released. React Router uses dynamic routing. When we say dynamic routing, we mean routing that takes place as your app is rendering, not in a configuration or convention outside of a running app. That means almost everything is a component in React Router.

Why use React Router?

React Router, and dynamic, client-side routing, allows us to build a single-page web application with navigation without the page refreshing as the user navigates. React Router uses component structure to call components, which display the appropriate information. By preventing a page refresh, and using Router or Link, which is explained in more depth below, the flash of a white screen or blank page is prevented.

This is one increasingly common way of having a more seamless user experience. React router also allows the user to utilize browser functionality like the back button and the refresh page while maintaining the correct view of the application.

What Happens When You Need to Navigate TWO Routing System?

An API is any place where a piece of code talks to another piece of code, but we often use it to mean somebody's external resource that gives me values, or our own internal database resource(s).If you are using a frontend and a backend, and you are potentially writing in multiple languages that don't necessarily have the same routing conventions, don't worry! The backend functions just as an API, and the user really doesn't interact with it at all. The routes that used to manage the user experience and the routes that used to manage queries to the database are not the same.

Add React Router

To add React Router in your application, run this in the terminal from the root directory of the application.

npmi -D react-router-dom

Folder Structure

Within the src folder, we'll create a folder named pages with several files:

- src\pages\:
- Layout.js
- Home.js
- Blogs.js
- Contact.js
- NoPage.js
- Each file will contain a very basic React component.

Basic Usage

Now use Router in our index.js file.

Example:

- Use React Router to route to pages based on URL:
- import ReactDOM from "react-dom";
- import { BrowserRouter, Routes, Route } from "react-router-dom";
- Now use Router in our index.js file.

Example

- import Layout from "./pages/Layout";
- import Home from "./pages/Home";
- import Blogs from "./pages/Blogs";
- import Contact from "./pages/Contact";
- importNoPagefrom "./pages/NoPage";

Pros

Routing between components is fast as the amount of data that renders is less. The rest of the data is rendered by the DOM, and even when there's tons of HTML and CSS to render, the DOM handles that part in the blink of an eye. Using lazy loading, any delay in rendering HTML is compensated for better user experience, animations and transitions can be easily implemented when switching between different components.It gives a real sense of a single-page application in action. No separate pages are rendered, and the current page doesn't refresh to load a new view

Cons

The initial loading time is considerably large as all the routes, components, and HTML have to be loaded at once when the application first mounts. The whole website or web app needs to be loaded on the first request.There is unnecessary data download time for unusable views that cannot be anticipated on the first render of the application.It generally requires an external library, which means more code and more dependency on external packages, unlike routing on the server-side.Client-side routing and rendering convert JavaScript to HTML, making search engine crawling less optimized.

14.4 What is react-router-dom ?

React Router DOM is annpm package that enables you to implement dynamic routing in a web app. It allows you to display pages and allow users to navigate them.It is a fully-featured client and server-side routing library for React. React Router Dom is used to build single-page applications i.e. Applications that have many pages or components but the page is never refreshed instead the content is dynamically fetched based on the URL. This process is called Routing and it is made possible with the help of React Router Dom.

Notes: To write an uncontrolled component, you need to use a ref to get form values from the DOM.

14.5 Router Parameters

React (also known as React.js or ReactJS) is a free and open-source front-end JavaScript library for building user interfaces based on UI components. It is maintained by Meta (formerly Facebook) and a community of individual developers and companies.

React Java Library

React can be used as a base in the development of single-page or mobile applications. However, react is only concerned with state management and rendering that state to the DOM. So creating React applications usually requires the use of additional libraries for routing, as well as certain client-side functionality.

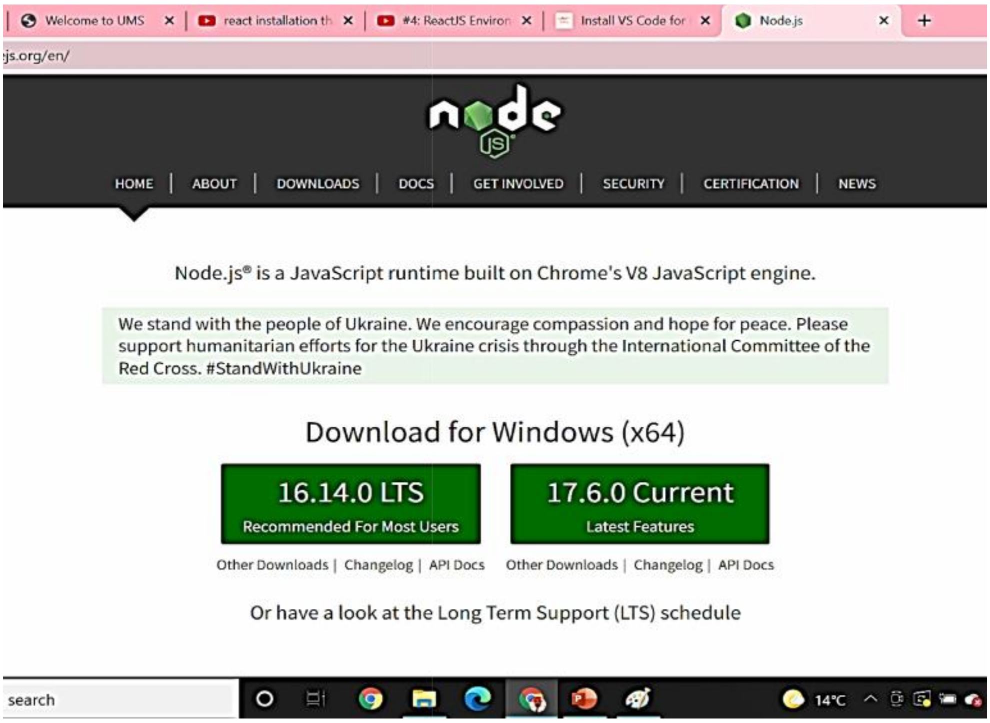
Components of React

React code is made of entities called components. These components are reusable and must be formed in the SRC folder following the Pascal Case as its naming conversion.Components can be rendered to a particular element in the DOM using the React DOM library.

Virtual DOM

Another notable feature is the use of a virtual Document Object Model, or virtual DOM. React creates an in-memory data-structure cache, computes the resulting differences, and then updates the browser's displayed DOM efficiently. This process is called reconciliation.

Install NodeJS and NPM.



Node.js® is a JavaScript runtime built on Chrome's V8 JavaScript engine.

We stand with the people of Ukraine. We encourage compassion and hope for peace. Please support humanitarian efforts for the Ukraine crisis through the International Committee of the Red Cross. #StandWithUkraine

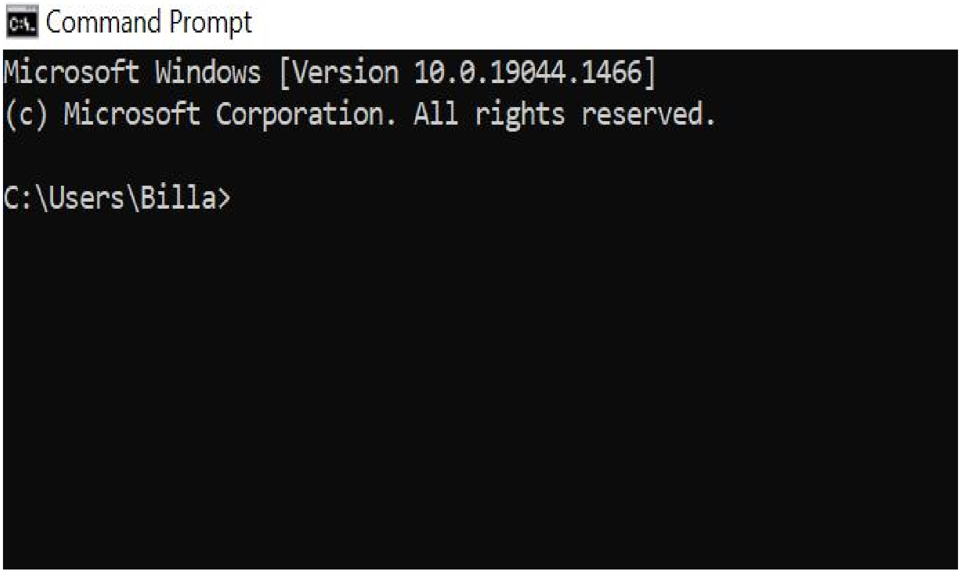
Download for Windows (x64)

16.14.0 LTS
Recommended For Most Users

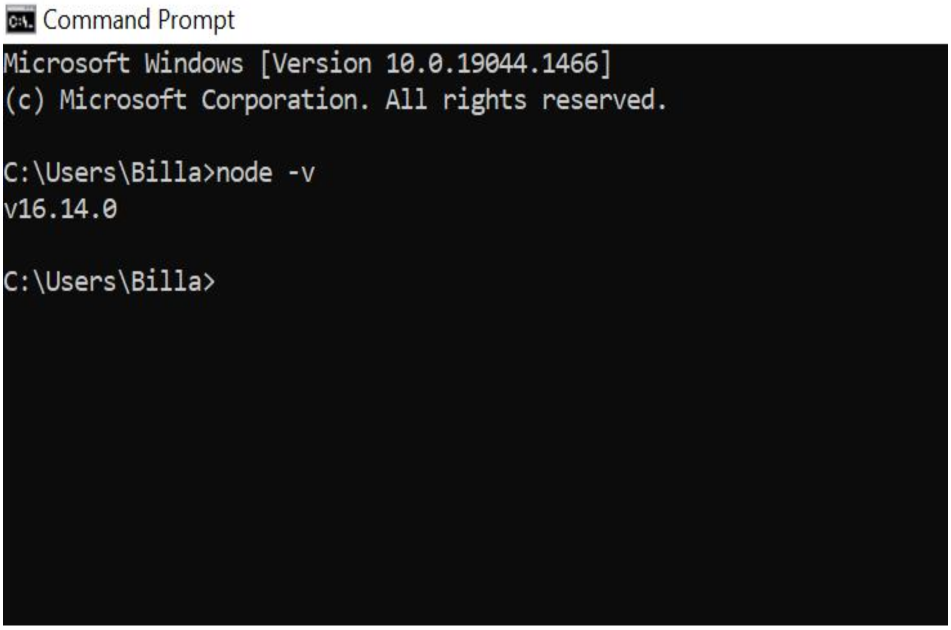
17.6.0 Current
Latest Features

Other Downloads | Changelog | API Docs   Other Downloads | Changelog | API Docs

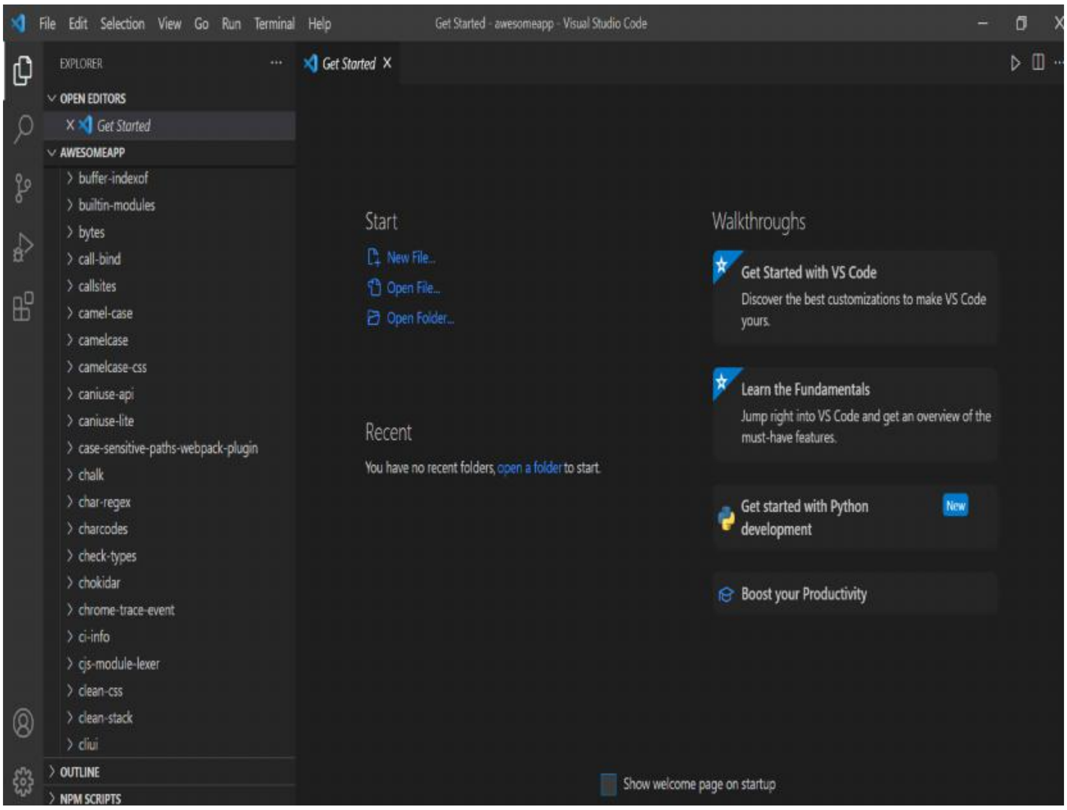
Or have a look at the Long Term Support (LTS) schedule


Command Prompt

```
Microsoft Windows [Version 10.0.19044.1466]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Billa>
```



```
Command Prompt
Microsoft Windows [Version 10.0.19044.1466]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Billa>node -v
v16.14.0

C:\Users\Billa>
```



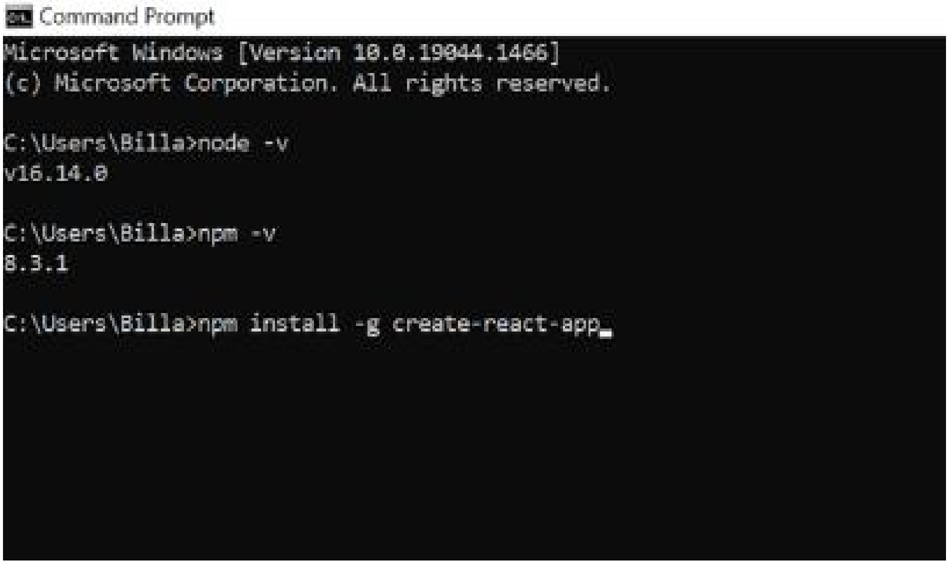
You have no recent folders, open a folder to start.

Discover the best customizations to make VS Code yours.

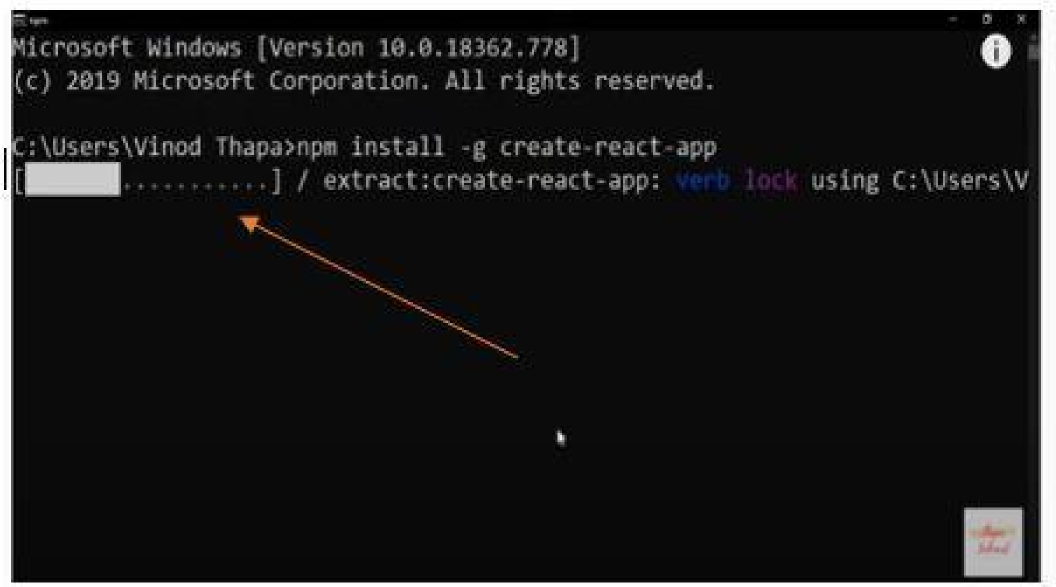
Jump right into VS Code and get an overview of the must-have features.

Install react from terminal

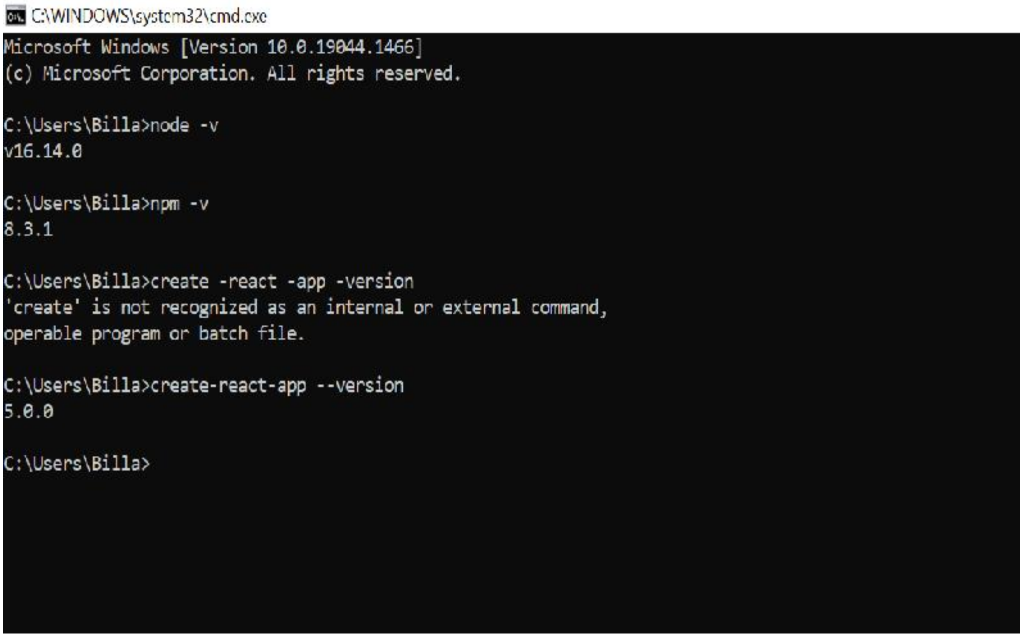
npm install -g create-react-app



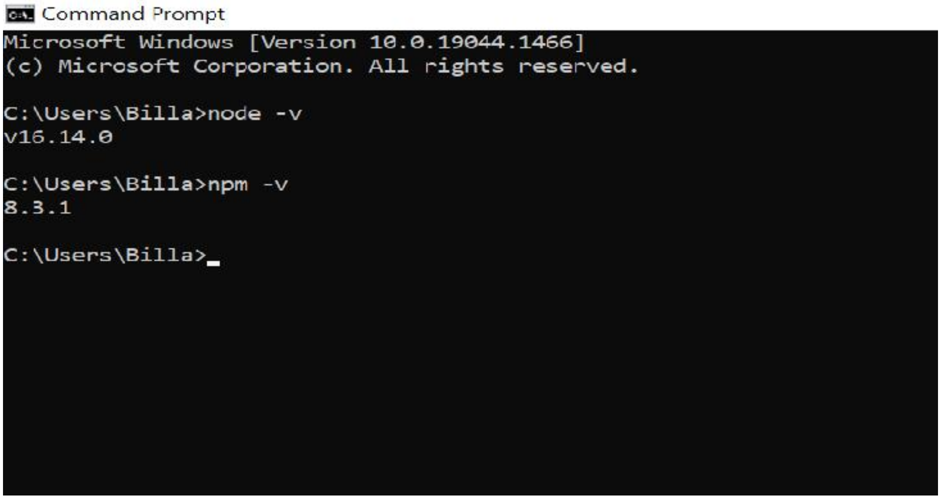
```
Microsoft Windows [Version 10.0.19044.1466]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Billa>node -v
v16.14.0

C:\Users\Billa>npm -v
8.3.1

C:\Users\Billa>npm install -g create-react-app_
```



```
Microsoft Windows [Version 10.0.18362.778]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\Users\Vinod Thapa>npm install -g create-react-app
[██████...........] / extract:create-react-app: verb lock using C:\Users\V
```

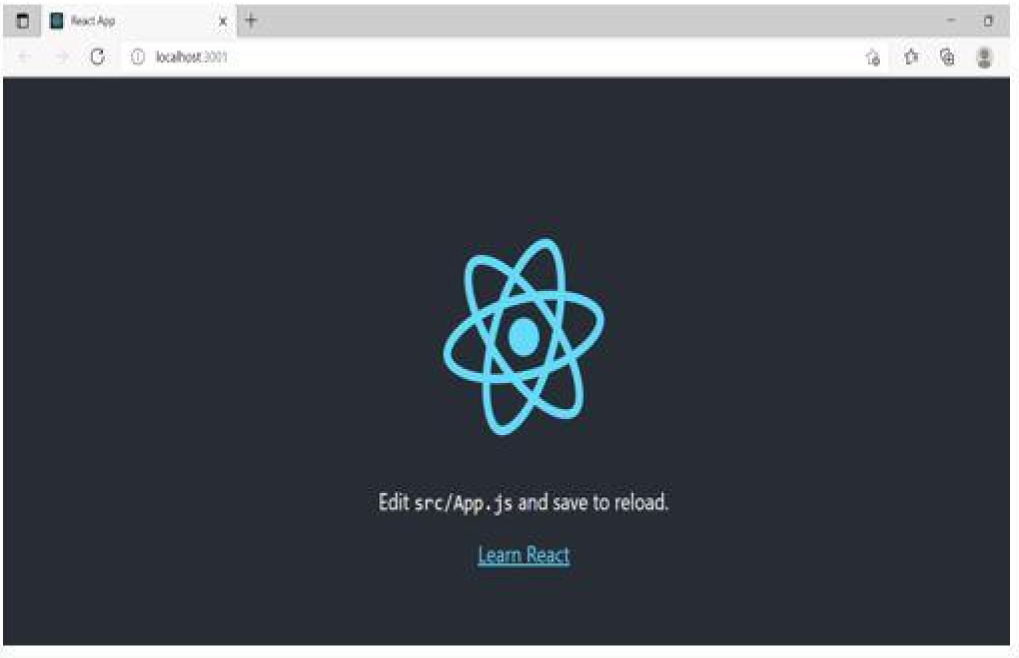


```
Microsoft Windows [Version 10.0.19044.1466]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Billa>node -v
v16.14.0

C:\Users\Billa>npm -v
8.3.1

C:\Users\Billa>create -react -app -version
'create' is not recognized as an internal or external command,
operable program or batch file.

C:\Users\Billa>create-react-app --version
5.0.0

C:\Users\Billa>
```



Command Prompt

```
Microsoft Windows [Version 10.0.19044.1466]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Billa>node -v
v16.14.0

C:\Users\Billa>npm -v
8.3.1

C:\Users\Billa>_
```



Edit src/App.js and save to reload.

Learn React

What is npm for?

npm is the package manager for the Node JavaScript platform. It puts modules in place so that node can find them, and manages dependency conflicts intelligently.It is extremely configurable to support a wide variety of use cases. Most commonly, it is used to publish, discover, install, and develop node programs.

What is Gitignore?

The .gitignore file tells Git which files to ignore when committing your project to the GitHub repository. gitignore is located in the root directory of your repo. / will ignore directories with the name.Ignored files are usually build artifacts and machine generated files that can be derived from your repository source or should otherwise not be committed

package. json file

The package. json file is the heart of any Node project. It records important metadata about a project which is required before publishing to NPM. Also defines functional attributes of a project that npm uses to install dependencies, run scripts, and identify the entry point to our package.

What is a package lock json file?

In version 5, npm introduced the package-lock. ... json file is to keep track of the exact version of every package that is installed so that a product is 100% reproducible in the same way even if packages are updated by their maintainers. This solves a very specific problem that package.

What is ES6?

ES6 stands for ECMAScript 6.ECMAScript was created to standardize JavaScript, and ES6 is the 6th version of ECMAScript, it was published in 2015, and is also known as ECMAScript 2015.ECMAScript (European Computer Manufacturers Association Script) is a scripting language based on JavaScript. Invented by Brendan Eich at Netscape, ECMAScript made its first appearance in the Navigator 2.0 browser.ES6 allows you to write the code in such a way that makes your code more modern and readable. By using ES6 features, we write less and do more, so the term 'Write less, do more' suits ES6. ES6 introduces you many great features such as scope variable, arrow functions, template strings, class destructions, modules, etc.

React E6 Classes

Classes are an essential part of object-oriented programming (OOP). Classes are used to define the blueprint for real-world object modeling and organize the code into reusable and logical parts.Before ES6, it was hard to create a class in JavaScript. But in ES6, we can create the class by using the class keyword. We can include classes in our code either by class expression or by using a class declaration.

Notes: A class is a type of function, but instead of using the keyword function to initiate it, we use the keyword class, and the properties are assigned inside a constructor() method.

Arrow functions are introduced in ES6, which provides you a more accurate way to write the functions in JavaScript. They allow us to write smaller function syntax. Arrow functions make your code more readable and structured.Arrow functions are anonymous functions (the functions without a name and not bound with an identifier). They don't return any value and can declare without the function keyword. Arrow functions cannot be used as the constructors.

React E6 Arrow Function

Arrow functions allow us to write shorter function syntax:

Before:

```
hello = function()
{
  return "Hello World!";
}
hello = () =>
 {
  return "Hello World!";
}
```

It gets shorter! If the function has only one statement, and the statement returns a value, you can remove the brackets and the return keyword.

Arrow Functions Return Value by Default:

```
hello = () => "Hello World!";
```

Export

You can export a function or variable from any file.Let us create a file named person.js, and fill it with the things we want to export.There are two types of exports:

Named and Default.

Import for Name

You can import modules into a file in two ways, based on if they are named exports or default exports.Import named exports from the file person.js:

```
import { name, age } from "./person.js";
```

14.6 The Render Function

The ReactDOM.render() function takes two arguments:

HTML code and an.its element and the purpose of the function is to display the specified HTML code inside the specified HTML element.There is another folder in the root directory of your React project, named "public". In this folder, there is an index.html file.You'll notice a single <div> in the body of this file. This is where our React application will be rendered.There is another folder in the root directory of your React project, named "public". In this folder, there is an index.html file.You'll notice a single <div> in the body of this file. This is where our React application will be rendered.

Display a paragraph inside an element with the id of "root":

```
ReactDOM.render(<p>Hello</p>,

document.getElementById('root'));
```

The result is displayed in the <div id="root"> element:

```
<body>

<div id="root"></div>

</body>
```

Keywords

Router: Routing is a process in which a user is directed to different pages based on their action or request. ReactJS Router is mainly used for developing Single Page Web Applications

Browser Router: BrowerRouter is a router implementation that has the ability to incorporate routing in react.

React Router: React Router is an API for React applications. Most current code is written with React Router 3, although version 4 has been released. React Router uses dynamic routing

React Java Library:React can be used as a base in the development of single-page or mobile applications.

LINK: Link component is used to create links to different routes and implements navigation around the application. Links accepts the to prop, which signifies where we want the link to navigate our user to.

Gitinore: The.gitignore file tells Git which files to ignore when committing your project to the GitHub repository.

Export: You can export a function or variable from any file.Let us create a file named person.js, and fill it with the things we want to export.

Summary

- Routing is a process in which a user is directed to different pages based on their action or request. ReactJS Router is mainly used for developing Single Page Web Applications
- React Router plays an important role to display multiple views in a single page application. Without React Router, it is not possible to display multiple views in React applications

- Gi ignore file tells Git which files to ignore when committing your project to the GitHub repository.
- npm is the package manager for the Node JavaScript platform. It puts modules in place so that node can find them, and manages dependency conflicts intelligently
- The name of a JavaScript library is ReactDOM. This library includes a number of React-specific methods that all deal with the DOM in some way.
- ECMAScript (or ES) is a standard for scripting languages such as JavaScript, ActionScript, and JScript developed by the European Computer Manufacturers Association. It was originally intended to standardize JavaScript, which is the most widely used ECMAScript implementation.

Self Assessment

1. _______is a process in which a user is directed to different pages based on their action or request.

A. Routing
B. Stating
C. Propes
D. All of the above

2. ______is used to define multiple routes in the application.

A. Props
B. States
C. React Router
D. All of the above

3. To use react routing which package is required….

A. React Router Package
B. React package
C. State package
D. All of above

4. ______plays an important role to display multiple views in a single page application.

A. Props
B. States
C. React Router
D. All of the above

5. Which provides the core routing components and functions for the React Routerapplications.

A. react-router:
B. react-router-native:
C. react-router-dom:
D. all of the above

6. Which is used for mobile applications.

A. react-router:

B. react-router-native:

C. react-router-dom:

D. all of the above

7. Which is used for web applications design.

A. react-router:

B. react-router-native:

C. react-router-dom:

D. all of the above

8. _________is a router implementation that has the ability to incorporate routing in react.

A. BrowerRouter

B. Switch

C. Link

D. All of the above

9. ________is the conditional component that renders a component based on the URL defined or the URL it is pointing to.

A. BrowerRouter

B. Switch

C. Route

D. All of the above

10. __________component is used to create links to different routes and implements navigation around the application.

A. Brower Router

B. Switch

C. Link

D. All of the above

11. The _________component is used to render only the first route that matches the location rather than rendering all matching routes.

A. BrowerRouter

B. Switch

C. Link

D. All of the above

12. ________is annpm package that enables you to implement dynamic routing in a web app.

A. React Router DOM

B. Dom

C. Router

D. None of the above

13. _______is a standard library for routing in React

A. React Router

B. React states

C. React props

D. All of the above

14. Which of the following are the advantages of React.js?

A. React.js can increase the application's performance with Virtual DOM.
B. React.js is easy to integrate with other frameworks such as Angular, BackboneJS since it is only a view library.
C. React.js can render both on client and server side.
D. All of the above

15. What of the following is used in React.js to increase performance?
A. Original DOM
B. Virtual DOM
C. Both A and B.
D. None of the above.

Answers for Self Assessment

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

Review Questions

1. How we render the routre? Explain its Types?
2. Explain npm in React?
3. How we can install react? Explain its steps?
4. How we implement Router in React?
5. What is Gitinore?
6. What are Routers Parameters?
7. What is react-router-dom? Explain?
8. What Happens When You Need to Navigate TWO Routing System?
9. How we can add react router?
10. Explain the need of React Router?

Further Readings

HTMl, JavaScript , DHTML and PHP By. Ivan Bayross.

Beginning JavaScript 2nd Edition By. Paul Wilton

Html 5 Black Book, Covers Css 3, Javascript, Xml, Xhtml, Ajax, PhpAndJquery, Second Edition

Web Links

https://www.javascripttutorial.net/javascript-function/

www.webopedia.com

www.web-source.net

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