Day 1 07-03-2021

MEAN Stack : Mongo Db/ MySQL Express Module Angular Node JS

MERN Stack : Mongo Db/MySQL Express Module React JS Node JS

MEVN Stack : Mongo Db/MySQL Express Module Vue Node JS

Phase 1

Agile : video Session

Git : Video Session

HTML/HTML5, Css/CSS3, JavaScript using ES5 and Bootstrap

Phase 2

ES6 Using TypeScript

Angular 8/9/10

Phase 3 :

Node JS

Core Module

http module

express module rest service.

Mongo DB

Phase 1:

Online shopping application

Login Module A Person

Customer Module B Person

Account Module C Person

Version Control System : Version Control system that records changes in file or folder or projects.

Version control system provide remote repository. Repository is folder or directory which hold files, folder or projects.

Centralized version control

SVN : In SVN local machine or developer machine directory connected to remote or server machine.

Git : Distributed Version Control system.

Git is a distributed sub version control tool use to source code managements.

Git is open source command are create using Linux Kernel.

Git commands

1. git –version

To make git local repository

1. git init : This command is use to create local repository
2. ls : This command is use to check all file and folder in current directory
3. ls –a : This command is use to check hidden folder and files.
4. **Git status :**  This command is use to find the last status of git command or is use to check the local repository.
5. **Git add filename:** This file is use to add in git staging area. It is a simple area where file present in git directory, that store information about what will go to next commit.

This command use to add file from file system to staging area.

1. **git rm --cached a1.txt :** This command is use to remove file from staging are to file system.
2. **git add . :** all file or folder present in that location**.**
3. **Git commit –m “message” :** This command is use to pass the file from staging are to Local repository
4. **Git remote add origin URL :**  This command is use to connect local repository with remote repository.
5. **Git push –u origin HEAD :**  This command is use to push the data from local repository to remote repository.

**Branch :** Branch is known as virtual pointer which hold more than one commit details.

Default branch is master or main

1. Git branch : This command use to display default as well as user branch details.

Default branch in local and remote may be main/master

Raj

Ravi

1. **Git clone url :** This command use to create local repository with existing file available in remote repository.

**Task1\_AK\_LOGIN**

**Task2\_RV\_APPLICATION**

**Tas3\_RJ\_CUSTOMER**

1. **Git branch branchName : This command is use to create the branch**
2. **Git branch : this command is use to check all branch available locally.**
3. **Git checkout branchName : This command is use to move from one branch to another branch.**
4. **git branch -D ravi :** This command is use to delete the branch
5. **git checkout -b ravi** : This command is use to create the branch and switch the branch

**Day 2 07-03-2021**

**HTML**

https://[www.google.com](http://www.google.com)

req (http/https)-----🡪

Client Server

🡨--res(http/htts)----------

Http : Hyper text transfer protocol / secure

www: world wide web

google : domain / server

com : commercial

<https://www.google.com> 🡪 URL

Uniform resource locator

HTML

CSS

JS

HTML 🡪 it is use to display the content on browser.

CSS 🡪 It is use to apply formatting style or presentation for contents.

JS 🡪 It is use to do action on contents.

HTML : Hyper Text Mark up language : It is use to create static as well as dynamic web page.

**Web Page :** Display the contents in different format like text, bold, italics, form, audio, video or clips etc.

SGML

XML

Web Application : Combination of more than one web page.

Static web page : display the content as it is on browser.

Dynamic web page : When user interact with any content event(action) occurs.

HTML Version 1, 2, 3, 4, and 5

HTML provide tags or elements which help to create the web pages.

HTML is case insensitive (means we can write the tag may be upper case or lower case).

HTML Tags

Syntax

<tagName> opening tag

</tagName> closing tag

<tagName/> self closing

1. html
2. head
3. body
4. title tag
5. paragraph tag : This tag is use to display the content on browser. <p></p>
6. break <br/> : This tag is use to break the content in new line.
7. Heading tag : This tag is use to write the heading for the paragraph tags

H1 to h6

H1 largest

H6 smallest

1. Attribute : Attribute is use to describe the properties of a tags.

Syntax

<tagName name1=”value” name2=’value’ name3=value></tagName>

Attribute must in opening tag in the form of key-value pairs.

Where value may be single or double without quote. But if value more than one world then it must be in single or double quote.

1. Font tag : This tag is use to change color, size and style (face).

<font color=’red’></font>

<font size =4></font>

<font face=’arial’><font>

<font color =”green” size=5 face=’monotype corsiva’></font>

1. b : to make bold
2. I : italic tags
3. U : under line
4. Hyperlink : This tag is use to connect one page to another page or one bookmark for the contents within same page.
5. External hyper link

<a href=”PathOfweb”>Text</a>

1. Internal hyper link or book mark.

<a href=”#uniqueName”></a>

<a name=”uniqueName”></a>

1. Image tag : This tag is use to add the image to web page.

<img src=”ImageName.formatOfImage” />

1. List tag :
   1. Unorder List UL : Unorder List and Li: List Item : Order doesn’t matter.
   2. Order List OL : Order List and LI : List Item

Order matter.

* 1. Definition List : DL : Definition List , DT : Definition Term and DD : Definition Description

Table Tag :

Employee Details

Id Name Salary

1 Raj 12000

2 Ravi 14000

3 Ramesh 16000

<table>

<tr>

<th>Id</th>

<th>Name</th>

<th>Salary</th>

</tr>

<tr>

<td>1</td>

<td>Ravi</td>

<td>12000</td>

</tr>

</table>

Table 🡪

Tr 🡪 table row

Th 🡪 table heading

Td🡪 table data

thead

tbody

tfoot

**Forms :** This tag is use to create the forms like Login Page, Application Page, Feedback form, Product Order etc.

**Login page**

Form tag

Before HTML5

**<input type="text/password/radio/checkbox/button/file/submit/reset"/>**

If button type submit, When we click on submit button it always check action present in form tag then it will redirect to target page.

By default method for form is get consider.

If method is Get information send through URL using technique URL Re-writing technique.

Like URL?key=value&key=value&key=value

In Get body empty.

If method is post, then the information send through body part.

It is very difficult to capture the data.

Performance wise get is faster than post.

In Get we can pass only 255 character data.

Post method is high secure but slow compare to get.

More than 255 character then we have to use the post method.

Get/Post/Put/Delete and more

HTML5 version also support only get and post.

Day 3 13-03-2021 :

VS : Visual Studio : .net

VSCode : micro soft and open source.

Angular and React or Vue JS

Brackets

Atoms

**Eclipse : Java**

MyEclipse

Notepad++

HTML is not a structure. All tags are optional.

Like HTML, HEAD or Body.

HTML 4.0 version

<!doctype HTML public =”url……………………..dtd”>

Document type definition.

Inside DTD file they written rules

What is root tag name, that tag contains how many child tag like head and body.

Inside head tag must title, meta tags, style, script etc.

Inside body it must contains more than p, h1, b, div, span etc .

HTML 5. 0

<!doctype HTML > Giving the instruction to browser we are going to write HTML5 features in web page again this tag is optional.

In HTML 5 they introduce more tags to achieve dynamic task.

<file:///C:/Users/91990/Desktop/Phase%201%20SimpliLearn%20Online%20Training/GitDemo/HTML5%20Programs/home.html?fname=Raj&lname=Kumar&gender=Male&hobbies=Reading&hobbies=Watching+TV&city=Bangalore&address=Nanindin+Layout+%0D%0A>++++++++

**CSS :**

Without CSS if we want to apply any formatting style we have to depending upon the tags as well as complexity more.

Actual contents and formatting style both are combine together. So to achieve separation on concern become very difficult.

**Cascading style sheet**. Css provide set property and values which help to do good look and feel (presentation logic for web page).

CSS files

1. Inline CSS
2. Internal CSS or embedded CSS
3. External CSS

**Inline CSS**

Syntax

<tagName style=”property:value;property:value;property:value;”>

</tagName>

tagName : p, h1 to h6, div, span, table, tr, form, input tag

**Internal or Embedded CSS**

Syntax

<style type=”text/css”> opening tag

Selector {property:value;property:value}

</style> closing tag

This tag we have to write in between head tag of HTML web page.

Types of selectors

1. **Universal selector : \* {property:value}**

 \* {color: red;}

1. **Specific selector : tagName {property:value}**

p{color: red;}

1. **Multi specific selector : tagName,tagName{property:value;property:value}**

 h1,h2{color: seagreen;}

1. **Class selector (local class selector ) : tagName.className{property:value}**

 p.firstPColor{color: red;}

 p.secondPColor{color: springgreen;}

1. Class selector (global class selector)

.className{property:value}

     .fontClass{font-size: 24pt;

     font-family: 'monotype corsiva';}

1. Id selector : #idName{property:value}

  #footer{background-color: yellow;}

Class selector : group of tags. More than one tag can have same class name.

Id selector : id must be Unique. Two tag have same type of tags, have same name attribute, same class name but id must be unique.

1. Child selector

tagName childtagName {property:value}

div p {color: wheat;}

3 External CSS

filename.css (external class file)

write CSS rules (no Style tags)

Then every we page using link tag to connect the external CSS file.

Day 4 14-03-2021 :

**Box Model :**

Every HTML tags like p, div, h1 to h6, span, form, table etc

Internally follow box model concept.

Every HTML tags by default contains padding, border and margin.

Padding and margin we can’t see but border we can see.

**JavaScript :**

**JavaScript was object based interpreter scripting language till ES5.**

**Object based Vs Object Oriented Programming**

**Compiler Vs Interpreter**

**Programming Vs Scripting**

**Java / C++ / Python**

**ES5 JavaScript**

**ECMA : European Computer Manufacture Association**

**ECMA is a concept.**

**JavaScript is a one of the implementation of ECMA (ES5, ES6….ES8 etc)**

**ES10**

**Concept**

**Using JavaScript we can do coding on web page.**

**Req(http/https)-🡪**

**Client Server**

**🡨-res(http/htts)---- HTML/HTML5**

**CSS/CSS3**

**JS**

**JS was/is use to do validation on client side.**

**ES5**

**Java**

**Asp.net**

**Php**

**Python**

**Node JS**

**After Node JS JavaScript use on client side as well as Server side coding.**

**MEAN Stack :**

**Java Full Stack.**

**jQuery, React JS is a library**

**Angular is Framework**

**Node is a run time environment for JavaScript library or framework.**

**JavaScript :**

**Syntax**

**<script type=”text/JavaScript”> opening tag**

**</script> closing tag**

**This tag we can write in between head tag or body tag of web page.**

**Variable and data types**

**In JavaScript to declare the variable we have to use the keyword as var.**

**Syntax**

**var variableName;**

**var a; // by default value of a is undefined**

**var n=10; //n become number data types.**

**var m=10.10; //m become number data types.**

**var name=”Ravi” //string type consider**

**var res = true; //boolean type consider**

**var obj = new Date(); // object reference consider.**

**Index.html**

**<html>**

**<head>**

**</head>**

**<body>**

**<p>Welcome</p>**

**</body>**

**</html>**

**DOM Hierarchy created in browser memory.**

**Html**

**head body**

**p—textNode(Welcome)**

**DOM Programmatically : DOM is a API(Application Programming interfaces ) which provided by different language like JavaScript, Java, Python, asp.net, php.**

**API : may using function, property, classes, modules we can read, write and update HTML code dynamically.**

**Operator :**

**Arithmetic Operator : +, -, \*, /, %**

**Conditional Operator : >, >=, <, <=, ==, !=**

**Assignment operator : =**

**Logical operator : &&, ||, !**

**Increment and decrement Operator : ++, --**

**Triple equal operator : ===**

**Type of operator : typeof**

**Short hand operator : +=, -=, \*=, /=,**

**Ternary operator : condition ? ture:false;**

**If statements**

**var a = 10;**

**var dd = new Date();**

**primitive type and reference data types.**

**Primitive type is use to hold only single value or it hold only value.**

**Reference type it can hold value as well reference data types.**

**dd.getMonth();**

**dd.getYear();**

**If statement**

1. **Simple if**
2. **If else**
3. **If else if**
4. **Nested if**

**Switch statement : the user or programmer want to execute set of code or block base upon their requirement then we can use switch statements.**

**Looping :**

**It is use to execute set statement again and again till the conditions becomes false.**

**While loop**

**Do while loop**

**For loop**

**functions : function is use to write set of instruction to perform specific task.**

**2 types of functions**

1. **Pre-defined global functions**
2. **User-defined functions.**

**Pre-defined functions**

1. **Alert(“msg”): This function is use to write pop up message.**
2. **Prompt(): This function is use to receive the value through keyboards.**
3. **parseInt() : converting string to integer (without decimal number)**
4. **parseFloat() : converting string to float (with decimal point number)**
5. **eval(): converting string to number (it is a combination of parseInt() and parseFloat())**
6. **confirm(): This function want to take the confirmation from the user to do the task. It contains two button with ok and cancel. If ok then return else false.**

**Do {**

**alert()**

**prompt()**

**switch() {**

**case 1 Addition**

**case 2 Substation**

**}**

**Do you want to continue confirm**

**}while()**

**Day 5 : 20-03-2021**

**1st way Normal function Syntax**

**function fuctionName(parameterList) {**

**}**

1. **function no passing parameter and no return type.**
2. **Function passing parameter but not return type.**
3. **Function passing parameter and return value.**

**Event : event is a interaction between user and components (html tags (DOM elements)).**

**Event provide bridge between html code and Java script logic.**

**Event is known delegation model.**

**Type of events.**

**All event in JavaScript start with pre-fix on followed by event name**

**onClick**

**onDblclick button**

**onMouseOver**

**onMouseOut image**

**onKeyUp**

**onKeyDown text field**

**onBlur**

**onFocus text fields**

**onChange dropdown**

**onSubmit validation**

**onLoad**

**onUnload body tags**

**etc**

**DOM : Document Object Model**

**Index.html**

**Html**

**Head body**

**Meta p,h1 to h6**

**Title table, tr, td etc**

**Script**

**Link**

DOM API (Document Object model Application Programming interface).

JavaScript, Java, python, C#

**DOM API**

1. document.getElementById(“idName”).value

**Form Validation :**

**Using 2 ways**

1. **using JavaScript**
2. **HTML5 features**

**Expression style function**

**Callback function : passing the function itself or body of function to another as a parameter is known as callback functions.**

**Arrow function**

**User-defined object**

**ES5 style function or prototype style.**

**Object 🡪 any real world entity.**

**Properties or state -🡪 have -🡪 variables/ fields**

**Person**

**Behavior --🡪 do/does -🡪 functions/ methods.**

**Place**

**Bank**

**Car**

**Computer**

**We can add run type property as well as functionality to user-defined objects.**

<https://www.surveymonkey.com/r/YKY5HMX?session_id=1769462755&type=GLVC&trainer=akash%20kale&course=&mega_category=&id=&mega_id>=

<https://www.surveymonkey.com/r/YKY5HMX?session_id=1769462755&type=GLVC&trainer=akash%20kale&course=&mega_category=&id=&mega_id>=