**Phase 1**

**Day 1**

**03-07-2021**

**Full Stack**

**MEAN Stack**

**MERN Stack**

**MEAN Stack**

**Mongo DB / MySQL**

**Express Module** using JavaScript we can create web application as well as REST full web service.

**Angular Framework** / React JS

**Node JS**

**MEAN Stack**

**Front end Phase 1 and Phase 2**

Phase 1

SDLC Agile

Git

HTML/CSS/Basic JavaScript

Using ES5 style DOM, event, function

Using ES6 style

Babel

Webpack

Overview Of Node JS

Phase 2

TypeScript

Angular Framework 10/11/12 etc

**Phase 3 backend**

Phase 3

Node JS

Module fs, util, url, http module

Express , REST API.

Mongo DB : Database

Mongodb as well as mongoose module.

**Phase 4**

Testing : jasmine, mocha, chai etc

Docker

AWS Overview etc

Visual Studio Code

**VSCode**

**Phase 1**

**Git**

**Version Control System :** Version control system that records or history changes of files or code inside the application.

3 line commit s1,

3 line commit s2,

3 line commit s3

Local version control :RCS : Revision Control system.

Centralized version control : SVN :

c1 c2 c3

Server (Repository ) folder.

Distributed Version Control

c1 c2 c3

Local Rep Local Rep Local rep

Server (Repository ) folder.

Github, bitbucket, AWZ, Azure, Google cloud etc.

We are going to learn Distributed Version control system

Git is type of Distributed version control system.

It is open source.

**Git commands**

**git --version**

to make local folder as local repository we have to run the command as

git init : This command is use to create the local repository. (only one time).

git status : This command is use to display last command status.

git add filename : This command is use to add the file from local folder to git staging area.

git add . : all files and folder present in current directory.

git commit –m “Message” : This command is use to send the file from staging area to local repository.

git config --global user.email "akash300383@gmail.com"

git config --global user.name "Akash"

if we do any changes

git add .

git status

git commit –m “message”

Github is one of the remote repository

Connecting local repository with remote repository

git remote add origin URL

git push –u origin HEAD

to check remote repository URL

git remote show origin

to push the data from local repository to remote repository we have to run the command as

connect the virtual lab

create folder

create file with data

git init

git add .

git commit –m “message”

**git branch :** git branch is like a pointer. git branch is use to keep or hold more than one commit details.

Default branch may **main** or **master**.

---🡪c--🡪c--🡪c A branch

---🡪commit---🡪commit---🡪commit-🡪

--🡪C-🡪C B---Branch

To create user-defined branch

git branch branchName

to check all branch details as well as current branch details.

git branch

To switch from one branch to another branch

**git checkout branchName**

**Phase 1**

**Day 2**

**04-07-2021**

**git branch : view all branch**

**git branch branchName : created new branch**

**git checkout branchName : switch to branch**

**git checkout –b branchName : create as well as switch to new branch**

manager main.txt

coding….

Push this code to remote repository

Create folder

Create files (main.txt)

Git init

Git add .

Git commit –m “done”

Git remote add origin URL

Git push –u origin HEAD

Git clone URL : first time to load the new repository

Git pull : it add new changes in existing repository

C1---🡪C2---🡪c3(HEAD)

**Raj**

**git clone URL**

create new file

create branch

git checkout –b Raj\_Login

git add .

git commit –m “message”

git push –u origin HEAD

inform to manager or remote repository developer after merge then branch code in main or master branch

clean up activity

git checkout main

git branch –D Raj\_login

git pull ( pull from remote repository)

**Ravi**

**Git clone URL**

If we want to do any changes even to write . or space. Do in user-defined branch.

**Push user-defined branch to remote repository**

**Pull in main/master branch from remote repository**

**demo.txt**

Demo.java

int a;

1st person code

int a;

2nd person code

**UI Technologies**

**HTML**

https://[www.googel.com](http://www.googel.com) URL

http : it is protocol. Set of rules which help to communicate more than one machine.

Www :world wide web

Google : domain

Com : commercial

Req(http/https)----🡪

Client Server

🡨--res(http/https)----

HTML/HTML5

Using HTML1, html2 and html3 we can create static web pages. We can create dynamic web page using other programming language like jsp, asp, php or JavaScript.

From HTML4 and HTML5 we can create dynamic web page alone without depending other technologies.

If user interact with web page some changes happen on web page.

HTML is provide lot of pre-defined tags.

Html is not a case sensitive.

Tags syntax

<tagName> opening tag

</tagName> closing tag

1. Html
2. Head
3. Body

What is html

Few html tags

Hyperlink

Add image

List tag

Table

Forms table

P

Heading h1 to h6

Attribute : attribute is known as properties of tags.

Attribute we have to use opening tag in the form of key-value pairs. Value may be single or double or without quote. We can write more than one attribute for tags.

<tagName key1=”value” key2=’value’ key3=value>

</tagName>

**Html 4**

**JSF : Java Server Faces**

Xhtml

<!doctype HTML PUBLIC =”URL……..dtd”/>

Document type definition

This file contains the rules for html page

What is the root tag name,

What is child tag tagName

Number of type tags 0 or 1 or many. Etc

**Html 5**

**<!doctype HTML>** giving the instruction to browser we are using html5 features in that page. (optional).

**Hyperlink**

Using hyperlink we can connect one page to another page or connect to content in same page(like bookmark).

External hyperlink

<a href=”PagePath.html”>Text</a>

a anchor

href hyper reference

Internal hyperlink

**Phase 1**

**Day 3**

**10-07-2021**

RxJS : Reactive JavaScript

**List tags**

**Unorder List**

**Order list**

**Table tags**

Employee Details

**Id Name Salary**

100 Ravi 12000

101 Ramesh 14000

Table

Tr table row

Th table heading

tr

td table data

**table**

**thead**

**tr**

**th**

**tbody**

**tr**

**td**

**tr**

**td**

**tfoot**

**tr**

**td**

**forms**

in HTML using form tag we can create GUI application. The end user can fill details like application form, login page, feedback page.

**Login page**

UserName

**HTML4**

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

**HTML5**

<input type=”number/email/date/time/url/range/radiobox”/>

<file:///C:/Users/91990/Desktop/July%202021%20MEAN%20Stack%20Batch/Phase%201%20Material/Notes/MEAN%20Stack%20July%202021%20Batch/HTML%20Programs/home.html?user=Raj>

file:///C:/Users/91990/Desktop/July%202021%20MEAN%20Stack%20Batch/Phase%201%20Material/Notes/MEAN%20Stack%20July%202021%20Batch/HTML%20Programs/home.html?user=Ramesh&pass=123345

by default form tag method consider as a **GET**

If method is GET. Data send through URL using query param concept.

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

**Method = post**

If method is post data will send through body part of request.

<file:///C:/Users/91990/Desktop/July%202021%20MEAN%20Stack%20Batch/Phase%201%20Material/Notes/MEAN%20Stack%20July%202021%20Batch/HTML%20Programs/home.html>

**HTML :** it is use to display the contents on browser.

**CSS : Cascading style sheet.**

CSS provide lot of pre-defined properties which help apply good look and feel for web page.

Using CSS we can achieve separation of concern.

HTML CSS

Actual content Formatting style

Types of CSS files

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

**Inline CSS syntax**

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

</tagName>

**Internal css or embedded CSS**

Syntax

<style type=”text/css”>

selector {property:value;property:value;}

</style>

Types of selector

1. Universal selector : \*
2. Specific selector : tagName :p, h1, h2, b etc
3. Multi specific selector : p,b,h1{property:value}
4. Local class selector : **tagName.className{property:value;}**
5. Global class selector

.className{property:value}

1. #idName{property:value}
2. parentTAgName childTagName{property:value}

class selector : groups of tag of same type or different types.

to make unique between two tags we use id attribute.

Id selector and class selector

<p class=”p1” id=”a1”>First para</p>

<p class=”p2” id=”a2”>Second Para</p>

<p class=”p1” id=”a3”>Third para</p>

<p class=”p2” id=”a4”>Fourth para</p>

<p></p>

Div is like a container which can contains more than other tags like p, div, h1 to h6 etc. This refer the part of the web page.

P 🡪 it contains contents or information.

**External CSS**

**styles.css**

css rules

**Box Model**

Every DOM (Document Object Model). Every html tag is known as dom elements.

Every DOM element internally follow BOX Model.

Div

P

B

H1

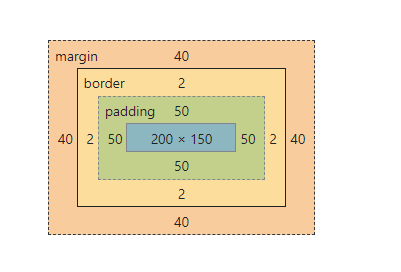
**BOX Model**

Margin

Border

Padding

Contents



border-style: solid;

            border-width: 2px;

            border-color: red;

            width: 200px;

            height: 150px;

            padding: 50px;

            margin: 40px;

            display: inline-block;

**Box Model :**

**FlexBox Model : Flex box model use one – dimension (row or columns).**

**FlexBox Architecture**

**Display : flex : It is consider as flex-box model or container**

**Inside that DOM tags is known as flex-items.**

**We can display Flex-item 🡪 Row wise or columns side.**

**Grid Box Model 🡪 Two dimension (row and column)**

Grid Box Model defines the elements or arrange the element in rows and columns. Two dimension

Inside a grid box model if we use any dom elements. They are consider as grid item. We can place grid item into specific location using line numbers, names or grid target area.

It is like a Tables

<div>

<img src=”imageName.”>

<span>Desc</span>

<input type=”button” value=”Add to Card”/>

</div>

<div>

<img src=”imageName.”>

<span>Desc</span>

<input type=”button” value=”Add to Card”/>

</div>

<div>

<img src=”imageName.”>

<span>Desc</span>

<input type=”button” value=”Add to Card”/>

</div>

<div>

<img src=”imageName.”>

<span>Desc</span>

<input type=”button” value=”Add to Card”/>

</div>

Visibility : hidden This property help to remove the element from DOM hierarchy

Display : None : it doesn’t display on browser it but contains memory space in DOM hierarchy.

float :left/right

float : left Vs display : inline-block

div

div

**JavaScript**

JavaScript is use to do programming on web page.

JavaScript was object based interpreter scripting language.

Object based Or Proto type vs Object Oriented (C++,Java, Python etc)

**class, object, inheritance, encapsulation, polymorphism and abstraction.**

Interpreter Vs compiler

Compiler convert whole code at time. But interpreter convert line by line.

Scripting Vs Programming

**JavaScript Using ES5**

ECMA (European Computer Manufacture Association)

ECMA is a concept.

The short cut of ECMA is ES

ES5

ES6

ES7

ES8

JavaScript is one of the implementation of ECMA Script.

**Syntax of JavaScript**

<script type=”text/JavaScript”>

</script>

This script tag we have to write in between head tag or body tag or without any tags.

**.html**

**Variable :** variable is a name which hold value.

To declare the variable in JavaScript (ES5) we can use **var** keyword.

Syntax

**var** variableName;

**Operator :**

**Arithmetic Operator :** +, -, \*, /, % (remainder)

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

**Assignment operator :** =

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

Pre – increment and post increment

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

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

**typeof**

**Phase 1 : Day 5**

**17-07-2021**

**If statements :**

**Simple if**

**if(condition){**

**}**

**If else**

**if(condition) {**

**}else {**

**}**

**Nested if : if within another if**

**if(condition) {**

**if(condition) {**

**}else {**

**}**

**}else {**

**}**

**If ladder or if else if**

**if(condition1) {**

**}else if(condition2) {**

**}else if(condition3) {**

**}else {**

**}**

**Looping : looping is use to execute the code again and again till the condition become false.**

**Initialization : start and end**

**Condition : true**

**Increment and decrement : increment or decrement**

**While loop**

**Syntax**

**var i=1,n=10;**

**while(i<=n) {**

**do the task**

**i++ or n--;**

**}**

**Do while loop**

**var i=0,i<=10;**

**do {**

**}while(i<=n);**

**For loop : fixed iteration.**

**1 2 4**

**for(initialization;condition;increment/decrement) {**

**body of the loop 3**

**}**

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

**2 types**

1. **Pre-defined function**
2. **alert(“msg”) : pop up message or alert box.**
3. **Prompt: this function is use to take the value through keyboards.**
4. **parseInt() : it is use to convert string to number(without decimal).**
5. **parseFloat() : it is use to convert string to number (with decimal).**
6. **eval(): it is use to convert string to number ( the number with or without decimal).**
7. **Confirm() : This function contains 2 button ok and cancel. If user click ok button it return true and if user click cancel button it return false.**

1. **User-defined function**

**In JavaScript we can declare the function different ways.**

1. **Normal function syntax**

**Syntax**

**function functionName(parameterList) {**

**function body;**

**}**

1. **Function no passing parameter and no return type.**
2. **Function passing parameter and no return type.**
3. **Function passing parameter and return value.**
4. **No passing parameter but return type.**

**Event : Event provide the interaction between user and html component(html tags).**

**Event provide the bridge between html code and JavaScript logic.**

**Type of event. In JavaScript all event start with pre-fix on followed by event name.**

**<select>**

**<option>Bangalore</option>**

**<option>Mumbai</option>**

**<option>Delhi</option>**

**</select>**

**Example**

**onClick**

**onDblclick : button**

**onMouseOver**

**onMouseOut : image**

**onKeyUp : search box of google**

**onKeyDown : chatting : she /he typing**

**onChange : drop down**

**onSubmit : form validation**

**onFocus : after enter in text field**

**onBlur : exit from text field**

**onLoad : web page or application loaded body tag**

**onUnload : when application close or re start.**

**etc**

**DOM : Document Object Model**

**index.html**

**<html>**

**<head>**

**<title>This is title message</title>**

**</head>**

**<body>**

**<h1>This is heading </h1>**

**<p>Welcome to HTML Web Page</p>**

**</body>**

**</html>**

**When we open this web page on browser. Internally browser created DOM Hierarchy.**

**Html**

**Head body**

**Title h1 p**

**TextNode textNode textNode**

**This is title message Welcome to My Web Page.**

**DOM API : Document Object Model Application programming interface.**

**DOM : in html all tags are knows as DOM elements.**

**All programming language like Java, Python, C#, JavaScript provided pre-defined function/ methods or classes which help to read, write and update DOM element dynamically or programmatically.**

**DOM API using JavaScript**

**Read the text field value**

**1st way**

**document.formName.textFieldName.value**

**2nd way**

**Form Validation**

**JavaScript using ES6**