**MEAN Stack Training**

**Day 1**

**11-02-2022**

Mongo DB / MySQL Express JS Angular Node JS

Mongo DB / MySQL Express JS React JS Node JS

Mongo DB / MySQL Express JS Vue JS Node JS

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

Uniform Resource Locator

http or s : protocol : hyper text transfer protocol secure

www : world wide web

google : domain

com : commercial

req(https/htt)------🡪

Client Server

🡨----res(http/https)------ html/html5

Html -🡪 display the content on browser

Css/css3

Css 🡪 it is use to display the content in proper format or presentation logic.

JS

JS -🡪 it is use to do programming on web page.

IDE :

VS Code :

HTML : Hyper text Mark up language

Html is use to create the web page.

Web page mainly use to display the content on browser in different format.

Html provide lot of pre-defined tags or elements.

Syntax

<tagName> opening tag

</tagName> closing tag

Html is not a case sensitive.

Html tags

1. Html
2. Head
3. Body

<html>

<head>

<title>This is my web page</title>

</head>

<body>

<p>Welcome to My Web Page</p>

</body>

</html>

.html or .html

1. Title tag
2. Paragraph tag p
3. Break tag br
4. Heading tag 6 heading
   1. H1 to h6

H1 is largest and h6 smallest.

attribute : attribute is known a properties of a tags.

Every tag contains one or more attribute in the form of key-value or name-value pairs. Attribute we have to use in opening tag. Value can be single or double or without quote.

Syntax

<tagName name1=”value1”> </tagName>

<tagName name1=”value1”> </tagName>

**Font tag** : This tag is use to change the color, size and style (face) of contents.

Hyper link : This tag is use to connect one web page to another web page or it is use to create book mark.

1. External hyper link
2. Internal hyper link or book mark

External hyper link

<a href=”pageName.html”>Text</a>

a anchor tag

href : hyper reference.

Add the images

<img src=”imageName.jpeg/gif/” />

List tags

It is use to display the item in proper format.

Unorder list

Order list

Table

**Id Name Age**

100 Ravi 21

101 Ramesh 22

table

tr 🡪 table row

th 🡪 table heading

tr🡪 table row

td 🡪 table data

Forms tag

Login Page

<form>

</form>

Html textfield syntax

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

Before submit form

<file:///C:/Users/91990/Desktop/Real%20Variable%20MEAN%20Stack/2022%20-%20MEAN%20Stack%20Training%20Real%20Variable%20Client/Programs/HTML%20Programs/login.html>

After submit form

<file:///C:/Users/91990/Desktop/Real%20Variable%20MEAN%20Stack/2022%20-%20MEAN%20Stack%20Training%20Real%20Variable%20Client/Programs/HTML%20Programs/home.html?user=Ravi&pass=12345>

<file:///C:/Users/91990/Desktop/Real%20Variable%20MEAN%20Stack/2022%20-%20MEAN%20Stack%20Training%20Real%20Variable%20Client/Programs/HTML%20Programs/home.html?user=Raj&user=Deep&pass=1234>

By default every html form send data through

url using re-writing technique if method is get.

URL?key1=value1&key2=value2&key3=value3

If method is get it is not secure.

If you want to send the data through body part we can use post method.

Post method slower than get method.

Using get we can send maximum 255 character data.

Html 4

<!doctype html url =”url………dtd”/>

Document type definition

Root tag

Number of child tags

Optional tags.

Html5

<!doctype HTML/>

Online Shopping

index.html

6 pages

Banner and images

login page

Username TextField

Password PasswordField

Submit Reset

Home Page

Link1 Link2 Link3 Link4

**Day 2**

**14-02-2022**

CSS : Cascading Style sheet : CSS provide set of properties and values which help to apply good look and feel for web page. With html it is not possible or may be code become more complex.

With help of CSS we can achieve separation of concern.

Types of CSS

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

Inline CSS syntax

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

</tagName>

difference between tag and attribute.

Attribute is properties of tags.

<tag key1=”value1”> : root tag as well as complex tag

<child1>

<child2>Hi</child2> : simple tag

<child3>Hello</child3>

<child4>How r you</child4>

</child1>

</tag>

Div tag : it is known as container tag. Which is use to add more than one tag ie p, h1 to h6 as well as another div. Div tag is use to specify the section of web page.

Internal css or embedded css

Syntax

<style type=”text/css”>

</style>

We have to write this style tag in between head tag.

selector {property:value;property:value;}

types of selector

1. Universal selector : \* {property:value}
2. Specific selector : tagName {property:value}

p{color:red}

h1{color:green}

1. Multi specific selector : tagName,tagName,tagName{property : value}
2. Local Class selector : tagName.className {property:value}
3. Global class selector : .className {property:value}
4. Id selector : #idName{property:value}
5. Child selector : outerTag innerTag {property : value}

**Class selector Vs Id Selector**

Class : it is to refer group of tag of same type or different types.

Id: id is use to name tag unique. May be tag have same name or different name.

<p class=”abc” id=”p1”>First Para</p>

<p class=”xyz” id=”p2”>Second Para</p>

<p class=”abc” id=”p3”>Third Para</p>

<p class=”xyz” id=”p4”>Fourth Para</p>

External CSS

CSS Box Model

Border property

Every tag in html internally follow box model



**Day 3**

**15-02-2022**

**JavaScript : JavaScript** was object based interpreter scripting language.

Object based or prototype base Vs Object oriented

OOPs

Object, class, Inheritance, Polymorphism, Encapsulation and Abstraction.

Old Version of JavaScript provide lot of pre-defined or user-defined object.

Interpreter Vs Compiler : both translator which convert from one format to another format.

Interpreter it will convert the code line by line. Compiler convert whole code at time.

Programming Vs scripting : running on browser and we are not converting in any other format.

JavaScript using ES5 feature.

ECMA Script : EMCA is a concept. (European Computer Manufacture Association).

JavaScript is a one of the implementation of ECMA.

Using JavaScript we can do programming on web page.

Syntax for script tag

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

document.write(“Welcome to JS);

</script> closing tag

We can write more than one script tag between head tag as well as body tag.

JavaScript is case sensitive. In JavaScript not mandatory every statement end with semicolon.

Variable : variable is a name which can hold value and value can change during the execution of a program. We can declare the variable using var keyword.

var a;

var name;

Data types : Data type is a type of data which tells what type of data it can hold.

var a; undefined

var b=10; number type consider

var c=20.20; number type consider

var name = “Ravi” string type consider

var res = true; boolean type consider

var obj = new Date(); object type consider

Operator :

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

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

Logical operator : &&, ||, !

Assignment operator =

Increment and decrement operator ++, --

++ it increment the value by one – it decrement the value by 1

typeof operator

ternary operator : condition ? true:false;

ternary operator is shortcut of if statement.

Conditional statements

If statement

If(condition) {

Set of logic execute

}

If else

If(condition) {

True block

}else {

False block

}

If else if or if ladder

If(condition) {

}else if(condition) {

}else if(condition) {

}else {

}

Switch statement : it is use to execute the set of statement depends upon the user choice.

Looping : looping is use to execute the set of statement again and again till the condition become false.

While loop

Do while loop

For loop (classical for loop)

1st 2nd 4th

for(initialization; condition ; increment / decrement ) {

body of the loop; 3rd

}

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

Divided into two types

1. Pre-defined function
2. alert(“Msg”) : it is use to display the pop up message.
3. prompt() : it is use to take the value through keyboards.
4. parseInt(): it is use to convert string to integer
5. parseFloat(): it is use to convert string to float
6. eval() it is use to convert string to number.
7. Confirm() : it contains two button ok and cancel. If user click on ok button it return true else it return false.

Do {

Alert 1: add, 2:sub etc

Prompt() take the value

Conversation convert the value

switch(){

case 1 Addition

case 2 Subtraction

case 3: Multiplication

case 4: Division

}

Confirm do you want to continue

}while()

1. User-defined function

Normal function syntax

function functionName(parameterList) {

body of the function;

}

1. function no pass parameter as well as no return type.
2. Function passing parameter but no return type.
3. Function passing parameter and return value
4. No passing parameter but return value.

**Day 4**

**16-02-2022**

Event : Event is a interaction between user and component (html tag or dom).

Or Event is a delegation model or event provide bridge between html and JavaScript.

JavaScript provided lot of pre-defined event and all events start with pre-fix on followed by event name.

Types of events

onClick

onDblclick

onMouseOver

onMouseOut

onKeyUp

onKeyDown

onChange

onFocus

onBlur

onChange

onLoad

onUnload

etc

DOM : Document Object Model :

DOM Hierarchy

Index.html

**<html>**

**<head>**

**<title>Title Message</title>**

**</head>**

**<body>**

**<p>Welcome to My Web page</p>**

**</body>**

**</html>**

**Html -🡪 root tag**

**head body**

**title p**

**TextNode: Title Message**

**NextNode : Welcome to My Web Page**

**DOM API (Document Object Model Application Programming interface).**

**Lot of programming provided DOM API like Java, Python, C# and JavaScript which help read, write and update DOM dynamically.**

**document.getElementsByName(“name”)**

**Form Validation**

**JavaScript is use to do validation ie all fields required, min length, max length, valid email id, valid phone number or credit-card.**

**Using JavaScript we can do validation on client side.**

1. **We can do validation using JavaScript**
2. **We can do validation using HTML5 features**
3. **We can do validation using Angular Framework**
4. **We can do validation using React JS**

**object :**

**object : any real world entity is known as a object.**

**two types of object.**

1. **Pre-defined object.**
2. **User-defined object.**

**Every object hold two things.**

1. **Properties or state or fields or variables.**

**Have --🡪**

1. **Behaviour or function or methods.**

**Do/does-🡪**

**Pre-defined objects.**

**Basic pre-defined objects.**

**Syntax to create the reference of pre-defined or user-defined.**

**var refereceName = new ObjectName();**

**refereneName.fieldName;**

**referenceName.functionName();**

**String object.**

**Literal style**

**Creating memory using new keyword.**

**Date**

**array : ES5 Methods**

**literal style**

**object style using new keyword.**

**splice(index,numberOfElementDelete,n1,n2,n2)**

**1st parameter : index position**

**2nd parameter number of elements to delete**

**3rd parameter number, 4th parameter and 5th parameter till infinite parameter to add.**

**Day 5**

**17-02-2022**

**JavaScript using ES6 features**

**Till ES5 to declare the variable we were using var keyword.**

**From ES6 onwards we can use let and const keywords to declare the variable.**

**var : var is a keyword to declare the variable. Using var we can re-declare same variable once again with same value or different value.**

**using var keyword we can do global declaration.**

**using let keyword we can’t re-declaration.**

**Using let keyword we can do local variable declaration or block scope declaration.**

**Types of function :**

1. **Normal function declaration**
2. **Expression style function**

**let/var functionName = function() {**

**}**

1. **Arrow style function**

**let/var functionName = ()=> {**

**}**

**Using normal style or expression style we have to return the value using return keyword.**

**Arrow function return value without return keyword.**

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

**Types of loop**

**While loop**

**Do while loop**

**For loop**

**forEach() function**

**for in loop**

**for of loop**

**for in loop it give index position**

**syntax**

**for(let/var variableName in arrayName) {**

**}**

**For of loop it give value directly**

**syntax**

**for(let/var variableName of arrayName) {**

**}**

**let/var obj = new ObjectName();**

**obj.propertyName; to get the value**

**obj.propetyName=value; set the value**

**obj.funtionName();**

**User-defined object**

**Object : any real word entity.**

**Example**

**Properties or state – have**

**Person**

**Behaviour -- do,does**

**Place**

**Animal**

**Bank**

**Customer**

**Order**

**In JavaScript we can create user defined object using**

1. **Literal style**
2. **Function style : using ES5**
3. **Class style : using ES6**

**let objetReferenceName = {key:value,functionName:function() {**

**}}**

Day 6

18-02-2022

Creating user-defined object in function style

Create user-defined object

Write one or more property

And one or more behaviour

1st behaviour to change property

2nd behaviour to display property

Using function style

Label Text Field

Add Name onClick = addName()

Get the value of text field using getElementById() or any other ways.

Those values store in array variables.

Display those names one by one using loop

In same page.



21-02-2022

JSON : Java Script Object Notation :

Json is use to share the data between two technologies.

JSON created using object literal style.

JSON is a pre-defined object which provide set of methods which help to covert string to json or json to object to string.

JSON.parse() : this method is use to covert string to json.

JSON.stringify() : This method is use to convert object/json to string.

Promise : Promise is a pre-defined object provided by JavaScript which help to handle asynchronous event of data.

**Synchronous**

**Asynchronous**

**Synchronous Statement: Every statement depends one by one**

**document.write(“Hi”);**

**document.write(“Hello”);**

**document.write(“How r you”);**

**Asynchronous Statement : every statement execute independently**

**document.write(“Hi”);**

**document.write(“Hello”);**

**document.write(“How r you”);**

**Synchronous function call**

fun1();

fun2()

fun3();

**Asynchronous Statement call**

fun1();

fun2()

fun3();

Synchronous call to server

1st req

2nd req

3r req

Client Server

Asynchronous call to server

1st req

2nd req

3r req

Client Server

JavaScript provided set of pre-defined function to do asynchronous task.

setInterval

setTimeout

clearInterval

scope object :

JavaScript provide two pre-defined scope object ie

sessionStorage and localStorage.

If you want to share the between two pages we take the help of scope object.

sessionStorage hold the value till session open ie once application close the data store in session storage get lost.

If you want to store the data permanently. localStorage store the data permanently.

sessionStorage.setItem(“key”,value);

localStorage.setItem(“key”,value);

sessionStorage.getItem(“key”)

localStorage.getItem(“key”)

sessionStorage.removeItem(“key”);

localStorage.removeItem(“key”);

Promise : : Promise is a pre-defined object provided by JavaScript which help to handle asynchronous event of data.

Promise can be resolve and rejected.

We will create user-defined promise

If function return promise to handle this data we have to take the help then and catch().

then() will execute if promise resolve

catch() will execute if promise rejected.

JavaScript provide pre-defined function

Ie fetch(): This function help to call backend technologies REST API develop in any language.

fetch method return type is promise object.

22-02-2022

let emp = {id:1,name:”Ravi”,age:21};

let employees = [

{id:1,name:”Ravi”,age:21},

{id:2,name:”Ramesh”,age:24},

{id:3,name:”Raj”,age:25}

]

let empString = JSON.stringify(employees);

let empJson = JSON.parse(empString);

fetch vs axios

Node JS :Node JS is run time environment for JavaScript library, Framework or Programs.

Before Node JS JavaScript only known as client side scripting language But after Node JS JavaScript is known as Client side as well as server side scripting language.

­

Before Node JS if we want to run any JavaScript program we have to take the help of HTML page.

Inside Html page we are writing JavaScript code(JavaScript code can be internally script or external js).

After node js we can run the JavaScript program through commands.

In Node JS we can’t use DOM (Document Object Model) we can’t use.

document.write(“Welcome to JS”)

console.log(“Welcome to Node JS”);

Typescript : Typescript is known super set of JavaScript which support all features of ES6.

Still browser can’t understand typescript file we have to convert typescript to JavaScript with help of transpiler.

To install the transpiler we require node js.

npm (node package manager) which help download external dependencies or modules depends upon the application requirements.

npm install –g moduleName

To enable typescript transpiler we have to take the help of npm command

**npm --version**

**npm install –g typescript**

tsc --version

first create sample.ts

console.log("Welcome to Typescript program");

then convert ts to js using command as

tsc sample.ts

After converted we can see sample.js

node sample.js

Typescript provide or support data types

In ES5 or ES6 JavaScript

var a =10;

a=”Ravi”;

Typescript

let variableName:datatype;

let variableName:datatype=value;

**tsc datatype.ts : it will convert ts to js (ES5)**

**tsc dataType.ts --target es6 : it will convert ts to js (ES6)**

**array in typescript**

**23-02-2022**

**In Typescript we can create array to store same type value as well as different types of values.**

**tsc filename.ts –-target es6**

**user-defined object in JavaScript as well as typescript with data types using 3 ways**

1. **Literal style**
2. **Class style**
3. **Function style**

**Different types of function in Typescript**

**In JavaScript only function name must be match not mandatory number of parameter as well as type of parameter must be match**

**rest parameter : if we write array variable as parameter then we have to pass array variable and array variable can contains zero or 1 or many values. But we have to pass mandatory.**

**Rest parameter is a type of array only but it can receive zero or 1 or many parameter.**

**To declare the rest parameter we have to declare the variable using …variableName**

**In one function we can use only one rest parameter and it must be last parameter.**

**Spread parameter or operator : spread parameter is use to pass the value for rest parameter using array concept.**

**Optional : to declare optional parameter we have use variableName? optional parameter declaration must from right to left. We can’t leave mandatory parameter between two optional parameter.**

**Optional parameter default values are undefined.**

**If we want meaningful value then we can use default initialization.**

**Oops concept using typescript**

**Typescript support access specifiers concept.**

**Access specifiers help to provide the visibility of variable, function and class.**

**private**

**public**

**Create Product class in ES6 style**

**Which contains pid,pname,and price as private property**

**Using constructor set the value for pid,name,and price.**

**Then create more than product object and display product details.**

**Inside a constructor we have to write constructor price must be > 100 if else than 100 set value as 100.**

**24-02-2022**

**In Typescript we can use access specifiers in parameter variable insider constructor to make variable as instance variable.**

**Encapsulation : Binding or wrapping data and code in a single unit is known as Encapsulation.**

**class className {**

**property**

**behaviour**

**}**

**Typescript class with setter and getter methods.**

**Polymorphism : One name many forms.**

**Two types**

**Compile time**

**Function overloading : function have same name but different parameter list(type of parameter list or number of parameter list must be different).**

**Run time**

**Function overriding**

**Function have same name and same signature(number of parameter list and type of parameter list must be same).**

**To achieve function override we need inheritance.**

**Using abstract we can achieve partial abstraction.**

**Abstraction : hiding the internal implementation without knowing background details.**

**Interface : interface is use to achieve 100% abstract which contains only abstract functions.**

**In typescript we use interface to provide specification with all incomplete functions.**

**The class which implements that interface must be provide the implementation or body for that functions.**

**As well as with help to interface we can create literal style type of objects with variables.**

**String template with back ticks**

**Typescript Program Assignment you have to do**

****