**31-03-2022**

**Day 1**

**Phase 4 : Front end technology**

Html, CSS, JavaScript, Overview of Bootstrap, Typescript, Angular Framework

Mongo DB

Self learning :

HTML, CSS, Mongo DB

**JavaScript, typescript and Angular Framework**

[**https://www.google.com**](https://www.google.com) **--🡪 URL**

http/https---------------🡪 req

Client Server

Res 🡨--------http/https html or html 5

Html is use to display the content on browser. Css or css3

With the help of css we can decide how to display the content :

JS (JavaScript)

Using JavaScript we can do programming on web page.

Hyper text mark up language which help to create the web page.

Using html we can create static web page as well as dynamic web page.

Cascading style sheet : Css provide set of properties and value which help of apply good look and feel for web page.

Front end technologies back end technologies

HTML or HTML5

CSS or CSS3 Java (JEE)

Bootstrap : open source CSS web framework Servlet, JSP and EJB

Spring framework

JavaScript : programming on web page Spring MVC and Spring boot

We can do validation, Asp.net

Dynamic programming on web page Php

And more Python

We want to read, write and update DOM

(Document Object model). Any html tag is known Node JS

As elements.

Jquery : jQuery is external library which help

To read, write and update DOM every easily.

Angular Framework using REST API

React JS Library

Before Node JS JavaScript is known as Client side scripting language but after node js JavaScript also known as Client side as well as server side scripting language.

**VS Code : (Visual Studio code )**

**Few basic html tag**

**P, all heading tags, h1 to h6**

**Br**

**Font tag**

**Img**

**Hyper link**

**List tag ul, OL and LI**

**Table tag**

**Form tag**

**Img tag**

**Hyper link**

**Div tag**

**Span tag**

**List tag :**

**Un Order list**

**Order List**

**Definition list**

**CSS : Cascading Style Sheet**

**Css provide lot of properties and values which help to apply good look and feel for web page. It complex through alone html.**

**Using css we can do separation of concern.**

**So actual content and formatting style we can write separately.**

**Css divided into 3 types**

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

**Inline CSS**

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

**Internal or embedded CSS**

**This tag we have to write in between head tag.**

**<style type=”text/css”>**

**Selector {property :value}**

**</style>**

**Types of selector**

1. **Universal selector : \* {property:value}**
2. **Specific selector : tagName {property:value}**
3. **Class selector (local class selector or global class selector)**
4. **Id selector**

**<p class=”abc” id=”a1” >First Para</p>**

**<p class=”xyz” id=”a2”>second Para</p>**

**<p class=”abc” id=”a3”>third Para</p>**

**<p class=”mno” id=”a4”>fourth Para</p>**

**<h1 class=”xyz” id=”a5”>first heading </h1>**

**<h1 class=”abc” id=”a6”>second heading </h1>**

**Name attribute, class attribute and id attribute**

**Class : group of tags of same type or different types.**

**Id attribute is use to make the unique ness between two tags when two tags have same name or different names.**

**Using CSS we have to create class selector base upon our requirements.**

**Before HTML5 we want to create the responsive web page.**

**Laptop, Desktop , mobile or other device.**

**CSS web framework it display all dom element base upon the device.**

**Bootstrap is open source CSS web framework which help to create responsive web application.**

**Container bootstrap classes**

1. **Container : leave some space left side as well as right side.**
2. **Container-fluid**

**Alert classes**

**Bootstrap Grid : Boot strap grid layout is use to arrange the component in row and column format.**

**Using this grid layout we can arrange the component base upon the device.**

**Device screen size.**

**XS extra small <576px**

**SM small**

**MD medium >=768px**

**LG large >=992px**

**XL extra large >=1200px**

**By default grid layout allow up to 12 column across the page.**

**We can divide our container in number of row and each row consider as 12 columns.**

**JavaScript :**

**Using ES5 old version**

**JavaScript was known as object base or proto type base interpreter scripting language.**

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

**Syntax to write the JavaScript code**

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

**<script type=”text/JavaScript”>**

**document.write(“Welcome to JavaScript”)**

**</script>**

**Declaring the variable**

**In JavaScript we can declare the variable using keyword var**

**Operator :**

**== ===**

**== : it check only value**

**=== : it check value as well as data types.**

**If statement**

**Switch statement**

**Looping**

**While loop**

**Do while loop**

**For loop**

**JavaScript function :**

**In JavaScript function are divided into two types**

1. **Pre-defined function (global function)**
2. **User-defined functions**
3. **alert(“Msg”);**

**alert(“Welcome to JS”);**

1. **prompt(“Enter the name”)**
2. **parseInt() : without decimal**
3. **parseFloat : with decimal**
4. **eval() : int and float**
5. **confirm() : This pre-defined display pop message . it contains two button ok and cancel. If we click ok it return true and if click cancel it return false.**

**do {**

**alert : 1 Add 2: Sub**

**prompt() take the choice through keyboard.**

**Conversation using eval, parseInt or parseFloat**

**switch() {**

**case 1: add operation**

**case 2: sub operation**

**default wrong choice**

**}**

**Confirm -🡪 do you want to continue**

**}while();**

**User-defined function**

1. **Normal function declaration syntax**

**function functionName() {**

**}**

1. **Function no passing parameter as well as not return type.**
2. **Function passing parameter and no return type.**
3. **Function no passing parameter and return type.**
4. **Function passing parameter and return type.**

**Events : Event provide bridge between html and JavaScript. Event is known as delegation model.**

**When user interact with any html tag different types of event generate.**

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

**Like**

**onClick : button**

**onDblclick : button**

**onMouseOver**

**onKeyUp**

**onKeyDown**

**onSubmit**

**onChange**

**onLoad**

**onUnload**

**onFocus**

**onBlur**

**etc**

**04-04-2022**

**Day 3**

**JavaScript is object based interpreter scripting language which provided lot of pre-defined object as well as we can create user-defined objects.**

**Pre-defined object**

**BOM : Browser Object Model**

**DOM : Document Object Model**

**JavaScript follow object hierarchy**

**Object --🡪 property**

**Behaviour**

**Object**

**Property**

**Behaviour**

**Object**

**Property**

**Behaviour**

**DOM is a property of window object**

**window.document.write();**

**Or**

**document.write();**

**DOM : document object model : in HTML Every tags is known as DOM elements. Like html, head, body, p, h1 etc.**

**DOM API : lot of programming language provided predefined function or methods which help to read, write and update dom element or html content dynamically.**

**Like Java, Python, C# as well as JavaScript etc.**

**Using ES6 Version**

**From ES6 (ECMA Script )**

**From ES6 onwards we can declare the variable using var, let and const keyword.**

**Using var keyword we can re-declare same variable once again with same value or different values.**

**Using let keyword we can’t do re-declaration.**

**var a=10; int a=10;**

**a=20; re-assign a=20;**

**var a =”Ravi”; re-declaration int a=30;**

**using var keyword it is consider as global scope.**

**Using let keyword we can declare local or block scope.**

**const keyword is use to declare the constant variable we can’t change the value.**

**types of function**

1. **Normal function**
2. **Expression style function**
3. **IIFE Function Immediate Invoke function expression.**

**(functionbody)(functionCall)**

**It will invoke then the function load. This type of function we can call only once when it loaded. Because this function doesn’t contains any name which also known as anonymous function.**

1. **Arrow style function**
2. **callback function : passing the function itself or function name or function body to another function as a parameter is known as callback function.**

**Basic JavaScript object**

**array :**

**05-04-2022**

**Creating user-defined object in JavaScript**

**3 ways**

1. **literal style : using ES5**
2. **function style : using ES6**
3. **class style : using ES6**

**JSON : JavaScript Object Notation :**

**Store the information in the form of key value pairs. Key must be in double quotation and it must be unique. Value can be any type of data types.**

**JavaScript provided pre defined object ie JSON which provided set of methods which help to covert string to object and vice-versa.**

**Typescript**

**Typescript is a super set of JavaScript. Which support all features of ES6.**

**Typescript support data types.**

**We have to create .ts extension file to do the coding for typescript.**

**We can’t include ts file in html page we have to convert ts to js using node js command.**

**To do that conversation we have to transpiler ie tsc.**

**We run JS file using Node JS**

**Before Node JS JavaScript is known as Client side scripting language.**

**But after node js JavaScript can call as Client side as well as server side scripting language.**

**Node js provide lot of modules like package in java which help to create server side coding using JavaScript like java, python or asp.net.**

**Express JS is like a spring framework.**

**MEAN Stack : Mongo Db / My SQL Express JS Angular Node JS**

**MERN Stack :Mongo Db/ My SQL Express JS React JS Node JS**

**Running simple js program using node js**

**First write typescript program**

**Save the file with name demo.ts**

**Now we have to convert ts to js**

**tsc demo.ts**

**if you get the error please install tsc module**

**npm install –g typescript**

**After install**

**Then convert using command as**

**tsc demo.ts**

**node node.js**

**generally when we develop the program we will write all code one file. we have to write the code in multiple files.**

**Those files are connect using import and export concept.**

**In typescript as well as angular we use concept as module. Module is like a package which help to divide code in different files base upon the functionality and we will connect those files using import and export.**

**Angular is a open source web framework developed by google. Which help to create SPA (Single Page application).**

**Source page target page**

**Index.html welcome.html**

**Hyperlink**

**Button**

**Submit button**

**Using JavaScript code**

**Once move from one page to another using the whole dom loaded once again.**

**Rather than loading whole page we want to load only part of web page. That thing we can do using single page application.**

**Using Angular as well as React JS we can create Single Page application.**

**Angular Framework base upon html, css, Typescript.**

**Google provided Angular CLI (Command line interface) which help to create the Angular project with configuration details.**

**To create the angular project we have to enable ng command (next generation).**

**npm install –g @angular/cli**

**First create angular project folder**

**Open the command prompt inside this folder.**

**To create the new project using angular cli**

**ng new project-name**

**Ex : ng new demo-app**

**Do you want routing -🡪 no**

**Styling -🡪 css**

**cd demo-app**

**open project in vs code (short cut way)**

**open the command prompt inside a project folder and code .**

**To run the project using commands**

**ng serve –o**

**it automatically compiled the program and after compiled successfully it will open in default browser with url as**

[**http://localhost:4200**](http://localhost:4200)

**in project**

**expand src**

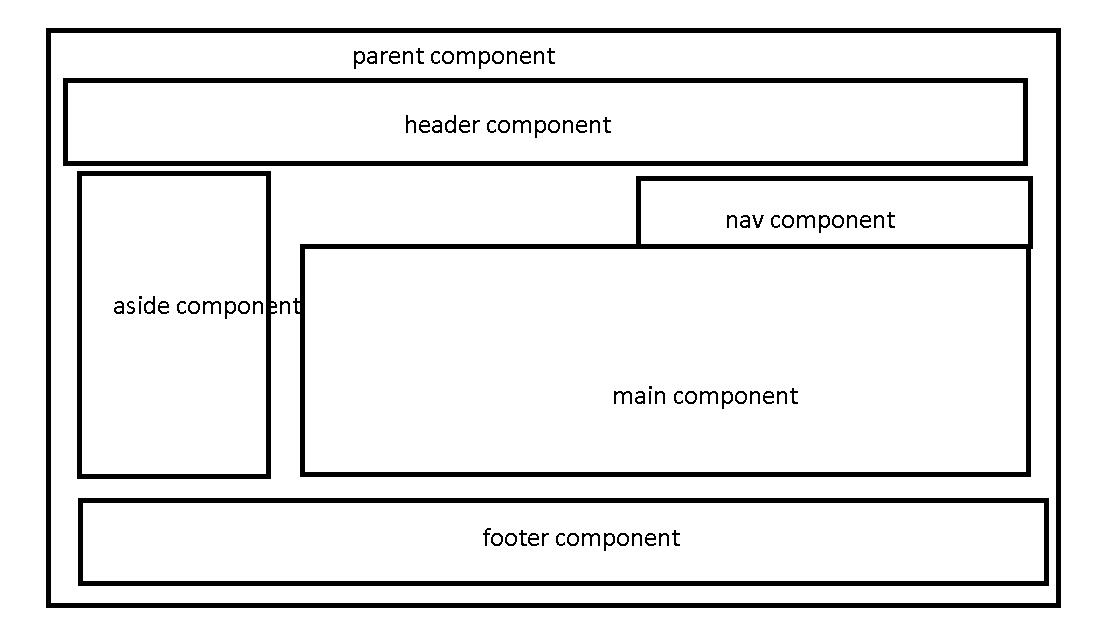
**expand app**

**app.component.html (template ) : In Angular html page is known as a template.**

**app.component.ts component file**

**app.component.css style sheet file**

**Angular use component. Component is use to control the view or part of view on web page.**

****

**In Angular to create the component. Angular use typescript class and provided decorator concept.**

**Decorator is like a annotation in java which provide extra information for class or property or functions.**

**Angular provide lot of pre-defined decorator. All decorator star with pre-fix @ followed name of the decorator.**

**In Angular using @Component decorator we are creating user-defined tags.**

**@Component**

**This decorator contains lot of attribute.**

**Selector : This attribute provided user-defined tags names.**

**templateUrl : This attribute is use to connect the html page. So wherever do use selector as user-defined tat name that place the content of html page get replace.**

**styleUls : This attribute use to connect the css file. it is like a link tag in normal html page.**

**app.module.ts**

**module is collection of more than one components.**

**@NgModule**

**This decorator is use to make typescript class as module.**

**Attribute inside a @NgModule**

**declaration : This attribute hold all component declaration details.**

**Import : This attribute hold more than module (pre-defined or user-defined details).**

**BrowserModule : This module is use to display the output in browsing area.**

**provider : This attribute hold angular service class details.**

**Bootstrap : in this section we can provide main component which load first.**

**Syntax to create the component using command prompt**

**ng generate component componentName**

**Or**

**ng g c componentName**

**Data binding**

**ng new data-binding**

**Data binding is use to share the data between component to view and view to component.**

**2 types**

1. **one way data binding**
2. **string interpolation : component to view**

**component ------------------------------🡪View (template)**

**{{variableName}}**

**{{8+10}}**

**{{display()}}**

**{{sayHello(“Ravi”)}}**

**ng g c string-interpolation**

**{{}}**

1. **property binding : component to view**

**Component ----------------------🡪View (template )**

**[]**

**<input type=”text” value=”lname”/> value itself is lname consider**

**<input type=”text” [value]=”lname”/> lname is a variable part of component and we will get the value of lname.**

**<p [innerText]=”lname”></p>**

**ng g c property-binding**

1. **event binding : view to component**

**view ------------------------🡪component**

**angular use same event provided by JavaScript. Only difference they remove on pre-fix and wrap with parenthesis.**

**JavaScript event Angular event**

**onClick (click)**

**onDblClick (dblclick)**

**onMouseOver (mouseover)**

**onSubmit (ngSubmit)**

**etc etc**

**ng g c event-binding**

**using event binding with string interpolation or property binding we can achieve two way data binding.**

**Passing the value from template to view.**

**Template reference : using template reference we can pass the value from view to component.**

**<input type=”text” #nameRef/>**

1. **two way data binding**

**two way binding : component to view and component**

**View 🡨-----------------------🡪Component**

**Two achieve two way data binding in angular we have to use ngModel**

**<input type=”text” [(ngModel)]=”variableName”/><br/>**

**ng g c two-way-data-binding**

**ngModel is a pre-defined attribute provide by angular to achieve two way data binding. This attribute is a part of FormsModule. So we have to import FormsModule in app.module.ts file in import section.**

**ng new angular-forms**

**07-04-2022**

**If we want to pass the value from template or view to component**

1. **we can use template reference.**

**<input type=”text” #nameRef/>**

**If we want to pass the group of values from template to component then we can use angular forms.**

**Angular provided two types of forms**

1. **template driven form**

**the flow of the application from template or view to component**

**it is easy to develop the application.**

**For simple for this type of form is good.**

**In this type of form we have to use ngForm and ngModel pre-defined attribute provide angular. These two attribute is a part of FormsModule. So you have to import FormsModule in app.module.ts file in import section.**

**In template driven form we have to create the form reference using ngForm attribute.**

**<form #loginRef=”ngForm”> This syntax is use to create the reference of form.**

**</form>**

1. **model driven form or reactive forms.**

**The flow of the application from component to view.**

**It complex to create the application**

**For complex this type of form is good.**

**We have to use fromGroup and formControlName pre-defined attribute provide angular. These two attribute part of ReactiveFormsModule. Template side.**

**Depending type of forms you have to import specific module in app.module.ts file inside import section.**

**Angular provided two api ie FormGroup, FormControl, FormArray as well as FormBuilder. Component side**

**FormControl is always use to bind particular form component like textfield, password field, radio button, checkbox, dropdown etc.**

**These all form control is a part of form group. FormGroup is a collection or group of more than one form control.**

**ng new angular-forms**

**so two component using command as**

**ng g c tdf-login-page**

**ng g c mdf-login-page**

**Angular Service :**

**If we write any business logic it may be simple or complex inside a component that logic become local to that component. So we can use that logic within that component or that component template.**

**Rather than writing business logic in every component you can make separate class and then write the business logic.**

**Angular service mainly help to provide common business logic.**

**Template Component**

**Tdf-login.html tdf-login.ts**

**Service class**

**Mdf-login.html mdf-login.ts**

**Business logic**

**Angular service mainly divided into two types**

1. **user-defined service** 
   1. **creating service object explicitly**
   2. **creating object using DI (Dependency Injection)**

**IOC : Inversion of control . IOC is a concept.**

**DI : Dependency Injection. DI is a implementation of IOC.**

**In Agular we can achieve DI using constructor base not setter base.**

**If you want to achieve DI for Service class we have to create normal class with decorator @Injectable**

**Then we have provide service class details in app.module.ts file provider attribute section.**

**In component class we have to take the help of constructor to pull the service class object.**

1. **pre-defined service**

**Angular provided pre-defined API it HttpClient which help to call backend technologies REST API ie Get, post, put and delete methods.**

**ng new structure-directive**

**ng new test-app-demo**

**ng new angular-service**