Phase 4 : Web Application

Total number of days 7

HTML

CSS

Basic JavaScript : Self learning

Adv JS

Typescript and Angular Framework : : live session

HTML :

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

hyper text transfer protocol

www :; word wide web

google : domain name

commercial application

---------🡪 req(http/https)---------------------------------🡪

Client Server

🡨---------res(http/https)---------------------

HTML :

HTML : Hypertext markup language : HTML is use to create the web page. HTML provided lot of pre-defined tags.

Syntax

<tagName> opening tag

</tagName> closing tag

VSCode

Html tags

Html

Head

Body

P

Heading tag h1 to h6

H1 is largest

H6 is smallest

Attribute : attribute is known as properties of tags.

We can use attribute in the opening tag in the form of key-value pairs.

<tagName name=”value”></tagName>

Font tag : This tag is use to change the color, style and size.

Hyper link : this tag is use connect one page to another page.

Syntax

<a href=”pageName.html”></a> a means anchor tag and href means hyper reference.

To add the images

<img src=”imageName”/>

List tags :

This tag is use to display list of items in property order.

Unorder list

Order list

Table tag

<file:///D:/Desktop/Java%20Full%20Stack%20may%202022%20Batch%202/Phase%204/Programs/HTML%20Programs/application.html>

<file:///D:/Desktop/Java%20Full%20Stack%20may%202022%20Batch%202/Phase%204/Programs/HTML%20Programs/application.html?fname=Raj&lname=Deep&gender=male&hobbies=Reading&hobbies=Playing&city=Bangalore&dob=2022-08-02&add=RT+Nagar+%0D%0A>++++++++

CSS : Cascading Style sheet :

CSS provided lot of pre-defined properties and values which help to apply formatting style for the web page.

With help of CSS we can achieve separation of concern means actual content and formatting style we can write separately.

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>

This is good if we are planning to apply formatting style of few tags.

Internal or Embedded CSS

<style type=”text/css”>

Selector {property:value;property:value;}

</style>

This tag we have to write in between head tag.

Types of selector

1. Universal selector : \* means all tags : \* {property:value;property:value;}
2. Specific selector : tagName p {property:value};
3. Local class selector : tagName.className{property:value}
4. Global class selector : .className{property:value}
5. Id selector #idName{property:value}
6. Child selector parentTagName childTagName{property:value}

Class selector Vs Id selector

Two tags which have same name or different name belong to same class. Class attribute groups of the tags.

But to make unique ness between two tag we have to use id. So id must be unique when tag have same name or different name.

<p class=”abc” id=”p1”>This is first para tag</p>

<p class=”xyz” id=”p2”>This is second para tag</p>

<h1 class=”abc” id=”p3”>This is h1 tag </h1>

<p class=”abc” id=”p4”>This is third para tag</p>

<h2 class=”abc” id=”p5”>This is h2 tag </h2>

<p class=”xyz” id=”p6”>This is fourth para tag</p>

CSS Box Model

Every HTML tags internally follow BOX Model

Day 2 23-08-2022

JavaScript : JavaScript was known as object based interpreter scripting language.

From ES6 onward JavaScript also known as object oriented interpreter scripting language.

ES : ECMA Script

Old Version JavaScript using ES5 version

New version JavaScript using ES6, ES7…. Version

JavaScript using ES5 version

Using JavaScript we can do programming on web page without server.

To write the JavaScript code we have to use tag ie

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

</script> closing tag

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

We write more than one script tag in between head and body tag.

Variable and data types

To declare a variable in JavaScript we use var keyword till ES5. From ES6 JS onward we can use let and const keyword.

JavaScript is known as loosely data types.

Syntax

var variableName;

var a; In JS default value for the variable is undefined.

Number type

String type

Boolean type

Object reference type

JS Operator

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

Logical operator && ||, !

Conditional Operator >, >=, <, <=, ==, ===

Assignment operator =

Increment and decrement ++ --

typeof

if statement and switch statement

if statement

if else

if else if

switch statement

looping : it is use to execute the statement again and again till the condition become false.

While loop

Do while loop

Initialization : start and end position

Condition : it must be true

Body of the loop

Increment or decrement

For loop

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

Functions are mainly divided into two types.

Pre-defined function or built in function

1. alert(“msg”); it is a type of pre defined function which help to display pop up message.
2. prompt(): it is use to take the value through keyboards.
3. parseInt() : it is use to convert string to integer.
4. parseFloat() : it is use to convert string to float
5. eval() : it is use to convert string to number it may be integer or float.
6. confirm() : it display pop message with two button. If user click ok it return true or return false.

User defined functions

Syntax to write user defined function

function functionName(parameterList) {

}

1. function no passing parameter and no return type
2. function passing parameter and no return type
3. function passing parameter and return type
4. no passing parameter and return type.

Normal syntax of function

Events : Events is a interaction between user and component ie button, textfield, radio button , checkbox, keyboard, mouse etc.

Types of events : in JavaScript all event start with pre-fix on followed by event name

onClick

onDblClick

onMouseOver

onMouseOut

onKeyUp

onKeyDown

onBlur

onFocus

onChange

onSubmit

onLoad

onUnload

etc

event provide the bridge between html and JS code.

DOM : Document Object Model :

When we run any html page in browser internally it will create the DOM hierarchy.

Index.html

Html

Head body

Meta p

Title h1

Script div

Style

DOM API : Document Object Model Application Programming interface.

Lot of programming language like Java, Python, JavaScript provided lot of pre-defined function which help to read, write and update HTML content dynamically.

Day 3 :

DOM Operation : Creating new tag, adding dynamic value and removing the tags.

JavaScript using ES6 as well as ES5

From ES6 onward 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 value.

But using let keyword we can’t do.

var a=10;

a=20; re-initialization

var a=30; // no error

let b=10;

b=20;

let b=30; // error re-declaration

using var we can declare global scope

but using let we can declare local or block scope.

Normal style function

Expression style function

Arrow style function

Callback function : passing the function name or function body to another function as parameter is known as callback function.

JavaScript objet :

JavaScript provided lot of pre-defined object as well as we can create user-defined object.

In JavaScript we can create user defined object 3 ways.

1. Literal style object creation : ES5 JavaScript concept
2. Using function style : ES5 JavaScript concept
3. Using class style : ES6 JavaScript concept.

Typescript : Typescript support all features of ES6 including it will support data types. JS doesn’t support. We can’t include typescript file in html page using script tag means browser still can’t understand typescript programs. So we have to convert ts to js with help of transpiler.

Typescript is a super set of JavaScript. But browser doesn’t support we have to convert into js.

To do this one we require node js.

Node js a run time environment for JavaScript program. Node js provided lot of pre-defined module with help of those modules we can create server side technologies with help of JavaScript.

25-8-2022

Npm (node package manager) using npm we can download the external node modules to develop the application depending upon the requirements.

Npm install –g typescript

or

npm install typescript --location=global

datatypes in typescript

let/var variableName:datatype;

type of function with passing parameter and return type

optional parameter and default initialization

user defined object creation

constructor

constructor with short cut initialization

modules : module is a like package in java. Which contains collection of function, classes, interfaces and variable which have same name but different purpose use.

With the help of modules we can write more than one function, class, interfaces or variable in different file and we can connect with help of keyword import and export.

Angular Framework :

Angular is a open source web framework provided by google company.

Angular Framework is base upon MVC architecture.

Angular JS : based upon html, css, JavaScript using ES5

Angular Framework : version 2 to 13 version

Base upon html, css, JavaScript and Typescript

Angular framework is use to create SPA (Single Page Application).

Index.html welcome.html

With help of angular application rather than loading the whole page we only part of part of the page base upon requirements.

Creating angular project manually very difficult. So Google provided Angular CLI (Command line interface) which help to create sample angular project and base upon requirement we can customize that project.

To create the angular project we have install or enable ng command (next generation).

Npm install –g @angular/cli

Or

Npm install @angular/cli --location=global

After installation ng command now we can create the new project using command as

ng new project-name

ng new welcome-app

routing -🡪 No

styling -🡪 CSS

cd welcome-app

To run the project run the command as

ng serve –o

After compiled this project 100% it will open in default browser with url as

<http://localhost:4200>

app.component.html -------------------🡪 template

app.component.css --🡪 styling

app.component.ts -🡪 typescript

Angular is known component base architecture framework. In Angular component control the part of view in web page.



With help of component we can do re-usability of that code.

With help of component we can create user defined tags.

To make normal typescript class as a component angular provided pre-defined decorator.

Decorator is a type of special function which add extra behavior to class, property or constructor etc.

Decorator is like a annotation in java.

@Component : This is a type decorator angular provide to make that normal Type Script class as a component.

This decorator contains lot of property

Selector : This property use to provider user defined tag.

templateUrl : This attribute provide the link to html page.

styleUrl: This attribute provide the link to css page like external style sheet.

app.module.ts

This class is knows a module class. In Angular module is known as collection of more than one components.

@NgModule

This module contains more than one attribute

1. Declaration : this section contains more than one component declaration.
2. Imports :This section contains all pre-define as well as user-defined module details.

BrowserModule : we are rendering our output on browser.

1. Provider : in this section we have to provide the details about angular service class.
2. Bootstrap : This attribute provide us parent component details which load as a first page in angular application.

WE can create more than one module and each module contains more than one components.

main.ts This file provide us main or parent module details.

Angular.json file : This file provide main file details.

 "index": "src/index.html",

            "main": "src/main.ts",

index.html

This file contains component selector as tag

Using angular cli we can create component

ng generate component compoentname

Or

ng g c componentname

ng new angular-data-binding

routing 🡪 no

styling 🡪 css

ng new angular-directives

routing 🡪 no

styling 🡪 css

26-08-2022

Data binding

Data binding provide the bridge between view or template to component and vice-versa. With help of data binding we can share the data between template to component and component to template.

2 types of data binding

1. One way data binding
2. String interpolation : Component -------------🡪View

{{}} : This symbol we have to use to achieve string interpolation. {{}} This syntax we have to use in template or html page.

{{variableName}}

{{display()}}

{{10+40}}

1. Property binding : component --------------🡪View

[] : this symbol we have to use to achieve property binding. [] this syntax we have to use in template or html page.

<input type=”text” value=”lname”/> we will value itself is lname.

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

<p [innerText]=”lname”></p>

1. Event binding : view ------------------------🡪Component

()

Angular use same event as JavaScript only difference all event start wrap inside () and remove on pre-fix

JavaScript Event Angular event

onClick (click)

onDlbClick (dblclick)

onMouseOver (mouseover)

onSubmit (ngSubmit)

etc

We can combine event binding with string interpolation or property binding with achieve two way data binding.

if we want to pass the value from template to component

template reference : using this we can pass the value form template to component.

<input type=”text” #nameRef/><br/>

1. Two way data binding : if we do any changes in component automatically update to template and vice-versa.

[()] To achieve two way data we have to use this symbol.

In Angular we can achieve two way data binding using ngModel attribute without depending upon event binding.

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

If we use ngModel pre-defined attribute in template page we will get the error. Because This attribute is part of FormsModule. So we have to import FormsModule in app.module.ts file in imports section.

Angular directive

Using angular directive we can add extra behavior for html page or dom elements or template.

Angular directive mainly divided into 3 types.

1. Component directive : component is a type of directive which help to create the user-defined tags.

@Component({

selector :”my-tag”

templateUrl:”./my-page.html”

})

export class MyCompoenent {

variable

function

}

1. Structure directive : using structure directive we can add and remove elements.

\*ngIf

\*ngFor

Using structure directive we can use if statement and looping in html page.

1. Attribute directive

Attribute directive is use to apply css styling

ngStyle and ngClass

ngStyle is like a style attribute in line css

ngClass is like a class selector in angular

ng new angular-forms

routing 🡪 no

styling 🡪 css

ng new angular-service

routing 🡪 no

styling 🡪 css

Syntax to create the model class

ng g class employee

29-08-2022

ng new angular-forms

using template reference we can pass the value of textfield, password field, radio button, checkbox etc from view to component. But if want to pass the more than one value from view to component we can use angular forms.

So

Angular provide totally 2 types forms

1. Template driven form
2. Model driven form or reactive form

Template Driven form : The flow of the form template to component. It is very easily to develop. The people from html and css they prefer this type of forms. In This form we have to use ngForm and ngModel pre-defined attribute in view side. These attribute are part of FormsModule. So we have to import this FormsModule in app.module.ts file.

In template driven form we have to create the reference of form

<form #loginRef=”ngForm”>

</form>

Model Driven form : the flow of the form component to view or template. It is complex to develop the people from typescript they can prefer this type of forms. In this form We are going to use two pre-defined API FormGroup and FormControl in component side and formGroup and formControlName pre-defined attribute on view or template side. formGroup and formControlName attribute part of ReactiveFormsModule so we have to import ReactiveFormsModule in app.module.ts file.

In Model Driven Form TextField, PasswordField, RadioButton, checkbox are connected with FromControle API. We have to bind or wrap more than one FormControl inside FormGroup. FormGroup is a combination of more than one FormControl.

Create two component in angular forms project

ng g c tdf-login-page

ng g c mdf-login-page

Angular Service : if we write any business logic it may be simple logic or complex logic in component that logic become local to that component. So we want logic or code can be access more than one component then we can use service class.

Template Component

Tdf-login.html tdf-login.ts

Login-sevice.ts

Mdf-login.html mdf-login.ts

So rather than writing business logic in every complex we have to write that code in service class and more than one service class create the object and access that service logic.

Angular service mainly divided into two types.

1. User-defined service
2. Creating object using new keyword
3. Creating object using DI (Dependency Injection)

IOC :inversion of control : it is a design pattern. Rather than creating and maintaining the life of the object allow and created by container. IOC is concept.

DI : Dependency injection: DI is a implantation of IOC.

Angular support DI Using constructor base only.

So if we want to achieve DI in angular we have follow few steps.

Create service class with decorator @Injectable.

And that service class details we have to provide in provider attribute in app.module.ts file.

And using constructor we have to pull the object inside a component.

1. Pre-defined or build in service : Angular provided pre-defined API ie HttpClient which get lot of pre-defined method like get,post, put and delete which help to load static json data from a json file or we can call rest api develop in any language like Spring boot.

First create the products.json file

ng g c product it is use to create the component

ng g class product it is use to create the model class

ng g s product it is use to create the service

We have to do the DI for HttpClient inside a Service class.

HttpClient predefined API part of HttpClientModule. So we have to import HttpClientModuel in app.module.ts in imports section.

HttpClient api all methods return type is Observable. So to load the data from Observable we have to use subscribe methods.