Total Day 11

Phase 4 : Day1 : 09-09-2022

Front end technology

Html, css, JavaScript, bootstrap, Typescript and Angular Framework

https://[www.google.com](http://www.google.com) ----------🡪 URL Uniform resource locator

http : protocol : set of rules which help to communicate more than one machine

s : secure

http : hypertext transfer protocol

www : world wide web

google : domain

com : commercial

req(http/https)-----------🡪Req

Client Server

🡨------res(http/https)

HTML

CSS

Bootstrap

JavaScript

ES5 and ES6

Typescript

Angular Framework

HTML : Hyper text mark up language which is use to create the web page. Web page can be static or dynamic.

1.0, 2.0, 3.0, 4.0, 5.0

HTML 5

Using html we can display the content on browser.

CSS : Cascading style sheet : with help of CSS we can apply lot of formatting style for web page or we can apply presentation logic for web page.

1.0, 2.0, 3.0

With the help of css we can apply formatting style for that contents.

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

JavaScript is object oriented interpreter scripting language from ES6 onwards.

ECMA

JavaScript was using to do validation.

1. Using JavaScript
2. Using HTML5 features

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

Bootstrap is a open source css web framework. Which provided lot of pre-defined classes which help to apply formatting style for web page.

Typescript : It is super set of JavaScript which support more ES6 or Oops features.

JavaScript doesn’t support data type concept. Typescript support data types.

Angular Framework :

HTML :

CSS : Cascading style sheet

Inline CSS

Internal or embedded css

External css

HTML is not a structure. We can write the program in html without tag but file must be .html or .htm etc.

Html 4

<!doctype HTML PUBLIC url=”…………………………dtd”/>

Document type definition

From html 5 they remove this rule and introduce more tags to make dynamic web page

Section, main, header, footer, audio, video etc.

<!DOCTYPE html> : we are giving the instruction to browser we are going to write

Html5 features.

JavaScript :

ES5

JavaScript is an object based interpreter scripting language.

Object base : old version JavaScript didn’t support class keyword it provided lot of pre-defined object as well as we can create user defined object.

Interpreter : it will check the code line by line.

Scripting language :

To run the JavaScript we have to use script tag. This tag we can write in between head tag or body tag of html page.

We can write script as internal or external.

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

</script> closing tag

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

Variable declaration : To declare the variable in JS till ES5 we use var keyword.

Syntax

var variableName;

data type : JavaScript is known as loosely type data types. Base upon value it consider that type of data type.

1. Undefined
2. Number
3. String
4. Boolean
5. Object reference

Operator :

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

Assignment operator : =

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

Increment and decrement operator : ++, --

Ternary operator : condition ? true:false

If statement

1. Normal if statement
2. If else
3. If else if
4. Nested if
5. Switch statement

Looping

while loop

do while loop

for loop

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

Function mainly divided into two types.

1. Pre-defined or global function
2. User-defined or custom function

Pre-defined functions

1. alert(“msg”); This function is use to display the message as pop up.
2. prompt() : This function is use to take the value through keyboards.
3. parseInt() : This function is use to convert string to integer
4. parseFloat() : This function use to convert string to float
5. eval() : this function use to convert string to number. It is a combination of int and float.
6. confirm() : this function display pop up message with 2 button. If user click ok it return true else return false.

do {

alert -🡪 1: Add 2 :sub

using prompt take the choice

switch statement

case 1: take the value of a and b and convert using eval, parseInt or parseFloat

do the sum

case 2 take the value of a and b and convert using eval, parseInt or parseFloat

do the sub

default wrong choice

using confirm ask the do you want to continue

}

Using alert display thank you

Phase 4 : Day2 : 10-09-2022

CSS : Cascading style sheet

Inline css : <tagName style=”property:value;”> </tagName>

Internal or embedded css

<style type=”text/css”>

Selector {property:value}

</style>

1. universal selector : \*
2. specific selector : tagName
3. global class selector .className
4. local class selector tagName.className

Bootstrap : Bootstrap is an open source responsive css web framework. Which provided lot of pre-defined classes base upon tags like p, div, span, button, form, tables etc.

Responsive provide a features to arrange the component or html tags base upon the device width and height.

Semantic UI

Angular : Angular material

After HTML5 we can make responsive web application with help of CSS also.

Bootstrap provided lot of predefined classes.

So we have to include bootstrap external file

1. offline
2. online

bootstrap 5.x

container and container-fluid

container and container-fluid is known as layout class which we can apply for div or p.

bootstrap grid layout

grid layout help us to arrange the html component or tags in row and columns formats.

By default every row in grid layout divided into 12 columns.

xs < 576px

sm ≥ 576px

md ≥ 768px

lg ≥ 992px

xl ≥ 1200px

xxl ≥ 1400px

12-09-2022

user-defined functions

In JavaScript we can write user-defined function lot of ways.

1. Normal function syntax

function functionName() {

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 type
4. Function no passing parameter but return type

Events : event is interaction between user and component(html tags). Event is a delegation model.

Event provide bridge between html and JavaScript code.

Type of events : In Js all event start with pre-fix on followed by event name.

onClick button

onDblClick button

onMouseOver image

onMouseOut image

onKeyUp textfield or passwordfield

onKeyDown textfield

onSubmit Validation

onChange drop down

onBlur texfield exit

onFocus textfield enter

onLoad body tag

onUnLoad etc body tag

DOM : document object model

index.html : when we run this page in browser internally it will create DOM hierarchy

html

head body

title div

meta Welcome textNode

script p

style My App textNode

DOM API (Document Object Model Application Programming Interface).

Lot of programming language like Java, Python, JavaScript, C# provided lot of api in the form of classes, function or methods which help read, write and update HTML content dynamically.

JavaScript object :

Object are divided into two types

Pre defined object

User defined object

Pre-defined object

JavaScript follow object hierarchy

BOM Hierarchy browser object model

DOM hierarchy document object model

object 🡪 property (fields/variable)

function

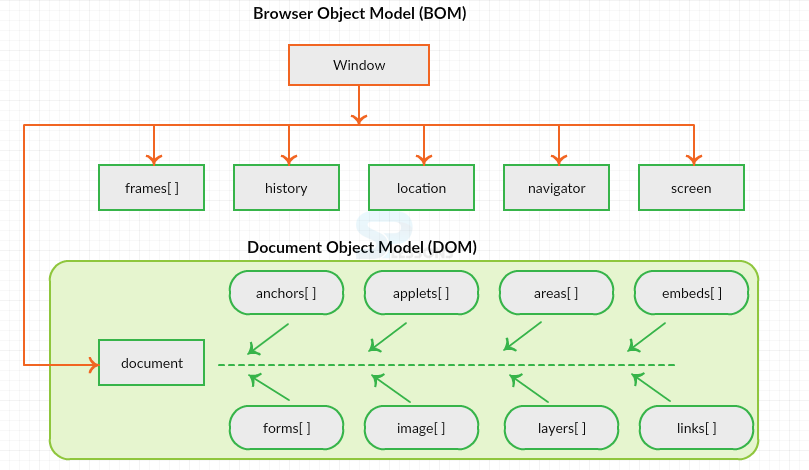
object -🡪 property (fields/variable)

function

object ---🡪 property

function

object



window.document.write(“Welcome”);

or

document.write(“Welcome”);

window.alert(“Welcome”)

or

alert(“Welcome”);



BOM : This hierarchy is use full when we want more default about browser or web page.

DOM : this hierarchy is use full we want to get the content of web page.

ES6 JavaScript

To declare the variable in JavaScript from ES6 onward we can use keyword as var, let and const.

Using var we can declare same variable once again with same value or different value.

Using let we can re-declare same variable once again with same value or different value.

var a=10; int a=10;

a=20; initialization a=20;

var a =30; it is possible int a=30; We get the error

let b =10;

b =20; //initialization

let b =30; // re-declaration error

using var we can declare global scope if we declare variable outside a function or if we declare inside a function then it is consider as function scope.

Using let we can declare block scope ie if block, for block etc.

const keyword is use to declare a constant variable like a final in java.

13-09-2022

1. Normal function : ES5
2. Expression function :ES5
3. Arrow function : Arrow function is replacement of expression function. Arrow function also known as anonymous function without function keyword. Arrow function is a ES6 features. Arrow function is short cut of normal or expression style function. Arrow function is equal to lambda expression in Java8.

Arrow function by default return output without return keyword. If we use simple line statement in arrow function no need to use return keyword as well as curly braces. But if we want to write more than one statement then we have to use curly braces as well as return keyword.

1. Callback function : passing the function or function body or function itself to another function as parameter is known as callback function.
2. IIFE (Immediate Invoice Function Expression )

Syntax

(functionbody)(functionCall)

This function is use to call only once while creation.

JavaScript Pre defined object

1. array : array is use to store same type as well as different type of values. In JavaScript array is known as dynamic memory creation. In JS array provided lot of pre-defined method which help to add, remove, update, iterate all element very easily.

Object scope : From html5 onward JavaScript provided two pre-defined object is sessionStorage and localStorage. These two object is known as storage object which help to share the data between One JS file to another js file.

sessionStorage.setItem(“key”,value)

sessionStorage.getItem(“key”)

sesssionStorage.removeItem(“key”)

localStorage.setItem(“key”,value)

localStorage.getItem(“key”)

localStorage.removeItem(“key”)

if we store the value in sessionStorage values are available till browser or application once if we close the application value get destroy automatically.

If we store the value in localStorage the value stored permanently. We have remove that value explicitly. If close the application after open application once again we can get the value.

Creating user defined object

object : any real world entity

till ES5 JS not supporting class keyword.

So to create the objet we are using literal style or function class.

In JavaScript we can create user defined object 3 ways

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

JSON : JavaScript object notation : JSON created based upon object literal style.

In JSON key must in double quote. In Literal style that mandatory.

If we give this object to java technology or .net technology they can’t understand.

So we have to convert JS object into jsno. JavaScript provided pre-defined object.

Ie JSON which contains pre-defined method which help to convert object to string or json and json to object.

employeeDetails = {“id”:100,”name”:”Ravi”,”age”:21,”result”:true,”add”:{“city”:”Bangalore”,”state”:”Kar”},”project”:[{},{}]};

16-09-2022

TypeScript

Primitives data types

Single Page Application

Angular decorator

Angular routing

Types of directives

Angular forms

Dependency Injection

TypeScript : TypeScript is known as super set of JavaScript. TypeScript is very stick in JavaScript.

TypeScript support all ES6 features and it is a type of object oriented programming.

Browser can’t understand TypeScript program or we can’t add typescript program directly in html page we have to convert ts to js. To convert ts to js we required transpiler. Ex : TypeScript(TS to JS) or babel(ES6 or JSX to JS5).

Node JS : Node JS is a run time environment for the JavaScript program.

Before Node JS JavaScript was known as Client side scripting language. To run the JavaScript all browser provided run time environment. With the help of CSS we are not able to develop server side technologies.

After node js we can say JavaScript also known as client as well as server side scripting language. Node JS provided lot of pre-defined module. With help of those modules we can develop server side programming language. Like storing data in file, security, connecting to database may be mysql or mongo db etc, creating rest api using Node JS.

Servlet, jsp or Spring boot

Asp.net

Php

Python

Node JS

In Node Js we can’t write DOM and BOM because we can run node js program using command prompt.

npm (node package manager) . This command by default available with node js which help to download external node js module. It is like a mvn

npm install typescript -g

or

npm install typescript --location=global

TypeScript features

1. Data types :

var variableName:datatype;

let variableName:datatype;

1. Array with data types.
2. Function with data types.

19-09-2022

Typescript modules :

While developing any project using typescript or angular generally we doesn’t write all the code in one file.

Module : module is a collection of function, variable, classes, interfaces etc which have same name but different purpose. So if we write set of function, classes in one file. Those function or classes visibility or accessibility within that file only. So we want to access in another file we have to use import and export keyword to connect the function or classes from one file to another file. Module is like a package.

In Typescript the file name itself consider as module (internal module).

Angular is an open source web framework provided by Google. Angular framework help to develop SPA (Single Page Application).

Index.html welcome.html

Hyperlink

Button

Submit

Using JavaScript code

Once we move from one page to another page once again the whole dom loaded in browser memory.

Frameset : these tags is use to include more than one web page as one web page.

Header.html every page contains there own DOM.

Footer.html every page contains there own DOM.

Nav.html

Main.html

Index.html

In Single Page application. We are creating only main or index page. And with help of component we are creating more than one block of the code.

Component is use to control the view or part of the view in web page.



With help of command we can create more than one component and every component work independently. Component always maintain relationship.

Parent - >child relationship

Sibling relationship

Angular JS : base upon html, css and JavaScript

1.0 to 1.x

Angular JS follow MVC architecture.

Angular Framework : Base upon html, css, javascript and typescript.

2, 4, to 14

Angular framework is component base architecture

MVC

Model

View

Controller is replaced by component.

Angular CLI (Command Line interface) which help to create the sample project.

In angular

npm install –g @angular/cli

or

npm install @angular/cli –location=global

ng version or ng –-version

Ng (next generation)

new new project-name

example

ng new welcome-app

routing 🡪 no

styling 🡪 CSS

cd welcome-app

To run the project we have to use the command as

ng serve

after project compiled 100%

then open the browser and write <http://localhost:4200>

expand src

app

app.component.html 🡪 template

app.component.css -🡪 external css file

app.component.ts -🡪

Angular use component concept.

Component : Component is use to control the view or part of view. Angular use normal typescript class and with help of decorator they made this class as component class.

Decorator : decorator is like a meta data or event we can decorator is annotation. Which help to make function or class or property is special type. Decorator provide extra information for that type.

In typescript we can create the decorator start with @ followed by decorator name.

Angular created lot of decorator

Like

@Component

@NgModule

@Injectable

@Input

@Output

@ViewChild

@Pipe

Using angular component we are creating user-defined tag.

@Component : this decorator contains lot of fields

1. Selector : name using this attribute or properties we are creating user defined tag. This attribute provide the user-defined tag name
2. Template-url : This attribute is use to connect to html page.
3. styesUrl : This attribute is use to connect the css file like external link tag.

app.module.ts

NgModule :

declaration : This section contains all component declaration.

imports : in this section we have to provide details about all pre-defined or user-defined module.

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

providers : in this section we have to provide angular service class details.

Bootstrap : in this section we have to provide main or parent component details.

main.ts

AccountModule LoginModule CustomerModule AppModule

Every module contains more than one component.

platformBrowserDynamic().bootstrapModule(AppModule)

  .catch(err => console.error(err));

This file contains parent module details.

Create new project

ng new angular-data-binding

Routing 🡪 no

Styling 🡪 css

In angular if we want html page means we have to create the component.

Using angular cli we can create the component

Ng generate component componentname

or

Ng g c componentname

Data binding

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

2 types

1. one way data binding
2. string interpolation : it is type of one way data binding. The flow of the application component to view.

Component ---------------------------🡪View

To achieve string interpolation we have to use operator as

{{}}

This {{}} we have to use in template.

Syntax

{{variableName}} {{name}}

{{expression}} {{6+7}}

{{functionClass()}} {{display()}}

ng g c string-interpolation

1. property binding : it is a type of one way data binding. The flow the application component to view.

Component -----------------------------🡪View

To achieve property binding we have to use the operator as

[]

<input type=”text”/> pure html

<input type=”text” value=”Raj”/> pure html and value is Raj

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

<input type=”text” [value]=”lname”/> now angular check the lname variable value inside component and that variable value display inside a textfield.

<p [innerText]=”lname”></p>

<div [innerText]=”lname”></div>

innerText is a pre-defined attribute for all html tags. Which help to achieve property binding.

ng g c property-binding

1. two way data binding

ng new types-of-directives

routing 🡪 no

styling 🡪 css

ng new angular-forms

routing 🡪 no

styling 🡪 css

20-09-2022

One way data binding

1. string interpolation
2. property binding
3. event binding : event is a type of one way data binding. The flow of the application view to component.

Operator to achieve event binding ()

Angular use same event provided by JavaScript only different they remove on prefix and all event names wrap with ().

JavaScript event Angular event

onClick (click)

onDblClick (dblclick)

onMouseOver (mouseover)

onSubmit (ngSubmit)

etc etc

ng g c event-binding

with help of string interpolation and property binding we can access the value of that variable.

Event binding with string interpolation or property binding we can achieve two way data binding.

If want to pass the value from template or view to component.

1. Using template reference : with help of template reference we can pass the value template to component

<input type=”text” #nameRef />

<input type=”button” value =”Click here” (click)=”passValue(nameRef)”/>

Two way data using ngModel attribute.

If we do any changes in component automatically it will update in view and vice-versa without depending upon any events.

To achieve two way data biding we have to use [()]

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

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

ng g c two-way-binding

Types of Directives : Using angular directives we are adding extra behavior or functionality to existing DOM.

3 types of directives

1. Component directive : component is a type of directive which help to create user defined tags with the help of selector attribute. which link with template using templateUrl. That template contains static as well as well as dynamic data (using data binding).

@Component({

Selector :”my-tag”,

templateUrl:”mypage.html”

})

class MyComponent {

}

1. Structure directive : using structure directive we can add or remove dom elements dynamically.

\*ngIf

\*ngFor

But with help of structure directive in angular we can use if and for loop in html or template page.

1. Attribute directive : using attribute directive we can add css effect for web page.

ngStyle style attribute inline css

ngClass class attribute in internal or external css

ng g c structure-directive

ng g c attribute-directive

if we want to display complex data like Employee, customer, order, product then have to create the model class.

ng g class Employee

Or

ng g interface Employee

Angular Forms

Using template reference we can pass the value from template or view to components.

This approach is good if we are planning to pass 1 or 2 value but not more.

So we want to pass group of value from template to component then we can take the help of Angular Forms.

2 types of forms

1. Template Driven Form

Flow of the application template to component.

Easy to develop and good for simple form

People or Developer from html and css prefer this type of form.

In This form we have to use ngModel and ngForm attribute. These attribute is part of FormsModule. So we have to import FormsModule in app.module.ts file.

1. Model Driven or Reactive Form

Flow of the application component to template

Complex to develop and good for complex form.

People or developer from TypeScript and Angular background prefer this type of form.

In this type of forms we have to use Angular API ie FormGroup, FormControl, FormArray etc in component side. Then in template side we have to use formGroup and formControlName attribute to bind the FormGroup and FormControl reference. formGroup and formControlName is a part of ReactiveFormsModule. So we have to import ReactiveFormsModule in app.moduel.ts file.

ng g c tdf-login-page

ng g c mdf-login-page

In Template Driven Form we have create the reference of form using ngForm attribute. loginRef is a reference name.

<form #loginRef = “ngForm”>

</form>

ng new angular-service

routing -🡪 no

styling -🡪 css

21-09-2022

According to model driven form text field, passwordfield, radiobutton, checkbox all html component bind with FormControl. FormGroup is a combination of more than one formcontrol.

Form Validation : without writing any details like usename, password, phone number, email send to server. Server need to verify before checking from a database.

Validation and Verification

Generally we do validation on client side using JavaScript or HTML5 features.

Angular validation or React JS validation.

Verification must be happen from server side.

Angular provided pre-defined attribute to do the validation.

valid : if rules satisfies on textfield, password field etc using required attribute then it is true else it is false.

invalid : it is reverse of valid.

touched

untouched

dirty

pristine

Validation using template driven form

Angular take the help of html5 to do validation using template driven form.

\d it must contains 1 digits

\D it must contains 1 alphabets

[abc] it must start with a or b or c

[a-z] it must start with a to z

[a-z]\* zero or many

[a-z]+ 1 or many

[a-z]? zero or 1

[a-z,0-9]+@[a-z,0-9]+.com

a to z or 0 to 1 mandatory before @

a to z or 0 to 1 mandatory after @

.com mandaotyr

Angular Service :

If we write any business logic in component that logic we can access with in that component or that component html page.

Html component service

Tdf-login component ts file

.html loginService

Mdf-login component ts file checkDetails

.html

Angular service class divided into two types

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

Angular support DI concept only constructor base not setter base.

Steps

Create the normal class with decorator @Injectable

We have to register for service class

1st option

In app.module.ts file in provider attribute we have to provide the details about service class

Inside component with help of constructor we have to achieve di.

1. Pre-defined service

Angular provided pre defined API is HttpClient which help to call Backend technologies service ie REST API which develop in any language.

So in user defined or in component ( generally we do in Service) we have to do the DI for HttpClient. HttpClient is a pre defined API part of HttpClientModule. So in app.module.ts file we have to import HttpClientModule.

After DI with help of HttpClient reference we can call

Get(), post(), put() and delete() method.

@Injectable()

class MyService {

info(public http:HttpClient) { // DI for HttpClient

}

}

http.get(url)

http.post(url)

http.put(url);

http.delete(url)

all these methods return type is Observable. Observable is a part of RxJS (Reactive JavaScript Programming). Observable is use to load the event of asynchronous data.

Synchronous

Asynchronous

Diff Synchronous Vs Asynchronous

Diff Promise Vs Observable

If method is return Observable we have take the help subscribe to load the data one by one from Observable.

We have to create sample json file which hold json data. And Node JS provided pre defined module which help to run static file as a json server.

products.json

{

"products":[

{"id":101,"pname":"Tv","price":55000},

{"id":102,"pname":"Computer","price":34000}

­]

}

Using npm command plz install json-server module

Ie

Npm install json-server --location=global

or

Npm install –g json-server

After installation successfully

Open the command prompt in the location where the json file present

json-server products.json

So json server provide url to get the data in json format

<http://localhost:3000/projects>

section end project

ng new angular-routing

routing 🡪 yes

style 🡪 css

phase end project

ng new angular-product-service

routing 🡪 yes

style 🡪 css

bootstrap