08-10-2020

Day 1

Phase 4

Html, CSS, JS using ES5 and ES6, bootstrap

Typescript and Angular Framework

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

req(http)------------🡪

Client server

🡨-----------res(http/https) html or html5

Html : hypertext mark up language which help to create the web page. Web page can be static or dynamic.

Html provided lot of pre-defined tags which help to create the web page.

Html is not a case sensitive.

<tagName> opening tag

</tagName> closing tag

Html, head, body, title, p, heading tag h1 to h6, font tag etc.

IDE : UI VS Code

Attribute : attribute is known as properties of a tag. We can use attribute in opening tag of html page in the form of key=value pairs.

In HTML4

<!doctype HTML public url =<http://www.sasdfasdsafasfasfddf.dtd>/>

Document type definition

Inside this file they written the rules.

1. Root tag name ie html
2. Which contains two child tag ie head and boy
3. Inside head we can write title, style, meta, script tag
4. Inside body we can write more than one p, div, span, etc.

In html 5 they remove this rules and they introduce new tags to make html dynamic

<!doctype HTML/>

Hyperlink : this tag is use to connect one page to another page.

<a href=”pageName.html”>Text</a>

Anchor

Href : hyper reference

List Tag :

Unorder list

Order List

Definition List

CSS : Cascading style sheet

CSS provided lot of pre-defined properties which help to apply formatting style for web page as well as 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

<tagName style=”property:value;”>Contents</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 : \* all tags
2. Specific selector : tagname
3. Class selector
   1. Local class selector tagName.className{property:value}
   2. Global class selector .className{property:value}
4. Id selector #idName{property:value}

<p class=”abc” id=”a1”>1st para</p>

<p id=”a2”>2nd para</p>

<div id=”a3”>1st div</div>

<p class=”abc” id=”a4”>3rd para</p>

<h1 class=”abc” id=”a5”>1st h1</h1>

<h1 id=”a6”>2nd h1</h1>

class : collection or groups of tags which have same name or different names.

Id : using id we have tag is unique which have same name or different name.

Bootstrap : it is an external open source CSS responsive web framework which provided lot of pre-defined css classes which upon the html tags.

09-10-2020

Day 2

HTML : it is use to display the content on browser.

CSS / Bootstrap : it is use to display the content on browser in proper format.

JavaScript : it provide programing features on browser without server.

JavaScript was object based interpreter scripting language till ES5.

From ES6 onwards JavaScript known as object oriented interpreter scripting language.

ES : ECMA : European Computer Manufacture Association

JavaScript is one of the implementation of ES.

Object based JS till ES5. In old version JS (Vanilla JS). JS provided lot of pre-defined object as well as we can create user defined without class keyword. Even we can say protocol base scripting language.

But from ES6 onward JS also supported class concept.

Interpreter : check the code line by line.

Syntax of script tag

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

</script> closing tag

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

We can write JavaScript ie internal JavaScript as well as externa JS.

In JS to declare a variable we use var keyword till ES5 JS. From ES6 onward to declare the variable we can use var, let and const keyword.

var variableName;

var a; default value of a is undefined.

Data types : In JavaScript variable behave that type of data types base upon the value which we assign or stored.

var b=10; number type consider

var c = 10.20; number type consider

var name = “Ravi”; string type consider

var result = true; Boolean type consider

var obj = new Date(); object type consider

Operators :

1. Arithmetic Operator : +, -, \*, /, %
2. Conditional operator : >, >=, <, <=, ==, ===, !=
3. Logical operator: &&, ||, !
4. Assignment operator : =
5. Increment and decrement ++, --
6. Ternary operator: condition ? true : false;
7. Type of operator or function : typeof

Condition statement

1. Simple if
2. If else
3. If else if
4. Switch statement

Looping :

While loop

Do while loop

For loop

functions

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

Function mainly divided into 2 types

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

Pre defined function

1. alert(“Msg”): it is use to display the 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
6. confirm() : it display pop message with two button if click ok it return true and if we click cancel it return false.

do {

alert 1: Add 2 :Sub

take your choice using prompt and convert using eval

switch

case 1: ask the value of a and b and display sum

case 2 ask the value of a and b display sub

do you want to continue using confirm

}while()

alert 🡪 thank you

user defined function

1. normal system for user defined function

function funtionName(parameterList) {

}

1. function no passing parameter and no return type
2. passing parameter but not return type.
3. Passing parameter and return the value
4. No passing parameter but return value

Event : event provide bridge between html and JS code.

Types of events

All event in JS start with pre-fix on followed by event name.

onClick

onDblclick

onMouseOver

onMouseOut

onKeyUp

onKeyDown

onChange

onSubmit

onLoad

onUnload

onFocus

onBlur

etc

JavaScript Object :

Object : object is any real world entity.

Person , Place, Bank, Customer etc.

In Js Object are divided into two types

1. Pre defined object
2. User defined object

Pre defined object hierarchy

BOM and DOM

Browser Object Model

Document Object Model

Object 🡪 properties

Behavior

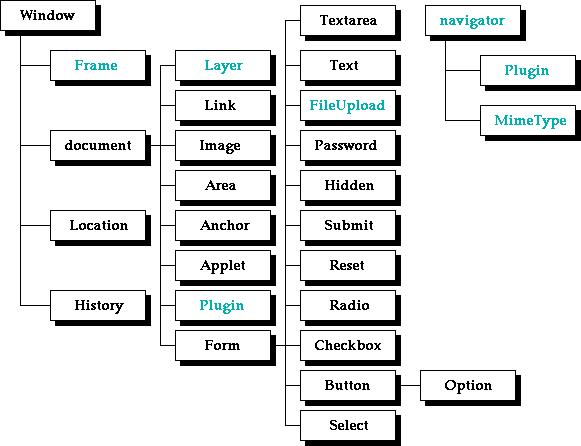
Object --🡪 properties

Behavior

Object --🡪 property

Behavior

Object



In BOM window is top most object which contains lot of property, behavior and another objects as property

window.alert(“”)

or

alert(“”)

window.confirm()

or

confirm()

document.write(“welcome”)

or

window.document.write(“Welcome”)

if we need browser details like history, location of pages, frames details then we have to work on BOM

if we need content of web page then we have to work on DOM.

index.html

DOM Hierarchy

Html

Head body

Title

textNode

Meta p

Script h1

Style

DOM API Document Object Model Application Programming interface. Lot of programming language like Java, Python, C# as well as JS provided lot of function or methods which help to read, write and update DOM element dynamically.

15-10-2020

Day 3

DOM Operation using External JS File

ES6 JavaScript

ES6 Features

1. From ES6 onwards we can declare the variable using let, const to declare the variable

If we declare variable using var keyword. It consider as global scope.

if we declare variable using let keyword. It consider as block scope.

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

But using let keyword we can’t re-declare same variable once again.

var a=10; int a=10;

a=20; a=20;

var a=30; int a=30; Error

let b=10;

b=20;

let b=30; //Error

for(var i=0;i<=10000;i++{

}

document.write(i);

for(let j=0;j<=10000;j++{

}

document.write(j); //Error

const k =1000;

k = 2000;

Array : array is consider as pre defined basic object which help to store more than one value of same or different types. Array provided lot of pre defined methods.

Types of functions

1. Normal function
2. Expression style function

var/let functionName = functionBody() {

}

1. Arrow function : arrow function also known as anonymous function. This function created using expression style function. In Arrow function function keyword replace by arrow. It is like a lambda expression in java.

In Arrow style function we can return the value without return keyword. In arrow style if we want to write multi line statement then we have to use curly braces and we have to return value using return keyword.

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

Object : any real world entity

Person

Employee

Customer

Bank

Car

We can describe the object in JS using 3 ways.

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

Typescript : Typescript is type of scripting language which help to create the web page. Typescript also known as super set of JavaScript. Typescript support data types. Browser doesn’t support doesn’t support typescript program directly. So we have to convert ts to js using transpiler.

Node JS : Node js is known as run time environment for JavaScript program ie may normal program or library or framework.

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

With help of node js we can create server side programming language ie web service, connecting database like mysql or oracle or mongo db, storing data in file, security program etc.

Node js provided npm (node package manager) which help to download external node js modules. Npm like a mvn in package (maven).

Node JS doesn’t provide BOM and DOM.

npm --version

npm install typescript –g

or

npm install typescript --location=global

1. Typescript support datatypes concept

var/let variableName:datatype;

1. array with datatypes
2. function with data types.
3. User defined object creation using class style with data types.
4. Typescript support interface
5. Modules : modules is a collection of classes, interfaces, variable and function which have same name but different purpose use. Module is like package in java.

Using import and export we will connect more than one file code (module code).

Angular JS

Base upon html, css and JS using ES5.

* 1. 1.x

Angular Framework 2.x to 14.x

Base upon html, css , js and TypeScript and Node JS

Both are open source web framework provided by google.

Angular JS based upon MVC architecture framework.

But Angular Framework controller is replaced by component. It is base upon component or component based architecture framework.

Angular Vs React JS

Both are use to create SPA (Single Page Application).

Index.html or index.jsp welcome.html

Hyper link

Submit button

Using JS code

Angular is a framework and react js is library.

Library doesn’t follow standard but framework follow standard.

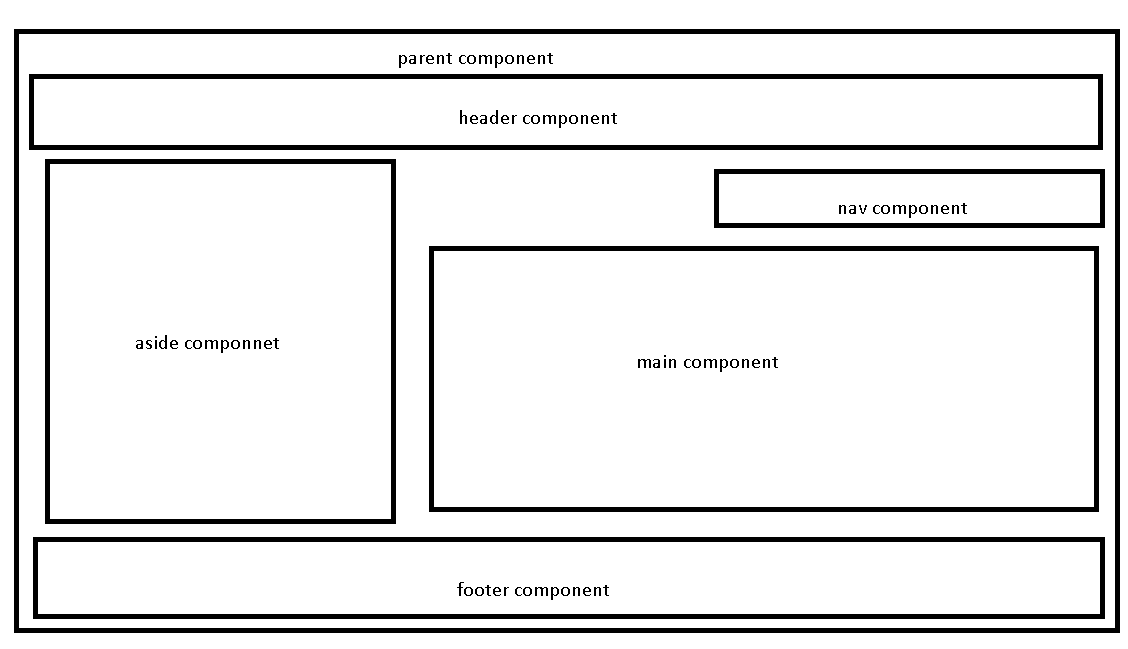
React JS is only View in MVC.

But in Angular follow whole MVC (model view component / controller).

Angular support class style component.

React JS support both style of component function and class style.

Component control the view or part of the view of web page and every component work independently.



Google provided angular cli (command line interface) which help to create the project

npm install @angular/cli –g

or

npm install @angular/cli --location=global

Angular

Ng :next generation for web page or web application

ng new demo-app

do you want routing file -🡪 no

styling --🡪 CSS

once project created using command prompt move inside a project folder.

cd demo-app

ng serve : This command is use to run the project.

After 100% project complied open the browser and hit the URL as

<http://localhost:4200>

open

src

app

app.component.html --🡪 template page

app.component.css -🡪 styling page

app.component.ts -🡪 component file (typescript file)

decorator : decorator like an annotation in Java. Which provide meta data information. Using decorator we can make class or property or function type of special function or class or property.

Angular created lot of pre-defined decorator. All decorator start with pre-fix @ followed by decorator name.

@Component

@NgModule

@Injectable

@Input

@Output

@Pipe

Etc

@Component is a type of decorator we can use on class to make normal typescript class a component class.

@Component decorator container lot of properties or attribute

Selector : “my-tag”, This attribute is use to create the user defined tags.

templateUrl : This attribute is use to connect to html page.

styleUlr: this attribute is use to connect css file like link tag.

29-10-2022

In Angular project

app.module.ts

@NgModule : decorator

Module is a collection more than one components.

Attribute

declaration : in this section we have to provide all component details.

import : in this section we have provide all pre-defined or user defined module details. Ie BrowserModule this module help to display the output in browsing area.

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

Bootstrap : in this section we will provide the parent component details.

main.ts

while developing enterprise application we will create more than one module.

LoginModule AccountModule EmployeeModule CustomerM

Every module contains more than one component and each component

provide one html page.

Main.ts file provide the parent module details.

Angular.json

"index": "src/index.html",

            "main": "src/main.ts",

We will create user defined component

Using angular CLI we can create the component

Command Line interface (ng)

ng generate component componentname

Or

ng g c componentname

Angular Data binding

ng new angular-data-binding

routing -🡪 no

styling 🡪 css

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

Two types of data binding

1. One way data binding
   1. String interpolation : the flow of the application component --🡪view

It is a type of one way data binding. To achieve this type of data binding we use {{}}. This symbol we have to use in template.

{{variableName}}

{{10+40}}

{{functionName()}}

ng g c string-interpolation

* 1. Property binding : the flow of the application component -🡪View

It is a type of one way data binding. To achieve this type of data binding we need to use []

<input type=”text”/>

<input type=”text” value=”Ravi”/>

<input type=”text” value=”lname”/>

<input type=”text” [value]=”lname”/> lname variable must be part of component. The value lname will display in textfield.

<p [innerText]=”lname”></p>

<div [innerText]=”lname”></div>

ng g c property-binding

* 1. Event binding : it is type of one way data binding. The flow the application view to component. To achieve event binding angular use ()

Angular use same event provided by JavaScript. They remove pre-fix on and all event name wrap with ().

JavaScript Event Angular event

onClick (click)

onDblClick (dblclick)

onMouseOver (mouseover)

onSubmit (ngSubmit)

etc etc

ng g c event-binding

using event binding and spring interpolation or property binding we can achieve two way data binding.

template reference :

syntax

<input type=”text” #referenceName/>

1. Two way data binding : angular support two data binding using ngModel. If we do any changes in component it will update automatically in view and vice-versa.

Two achieve two way data binding we have to use ngModel attribute with [()]

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

ngModel is a pre defined attribute part for FormsModule. So we have to import FormsModule in app.module.ts in import section.

ng g c two-data-binding

Directives and types of directives

ng new types-of-directives

routing 🡪no

styling 🡪 css

Directives : Directives help us to add extra behavior or functionality for existing dom or html tags.

3 types of directives

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

@Component({

selector :”my-tag”,

templateUrl:”./mypage.html”

})

export class MyComponent {

}

1. Structure directive : using structure directive we can add or remove dom elements from html page.

\*ngIf

\*ngFor

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

ng g c structure-directive

1. Attribute directive : using attributed directive we can add styling for html page.

ngStyle

ngClass

ng new angular-forms

routing 🡪no

styling 🡪 css

ng new angular-service

routing 🡪 no

styling 🡪no

30-10-2022

Using template reference we can pass the value from template to complete.

Angular forms : using Angular forms we can pass group of values in the form json from template to component.

2 types of forms

1. Template driven form or tdf

The flow of the application template to component.

It is easy to develop and good for simple form.

The people from html or css background they prefer this type of form.

To achieve template driven form we need to use ngForm and ngModel attribute. These attribute are part of FormsModule. So we need to import FormsModule in app.module.ts file.

ng g c login-tdf

in template driven from we have to create the reference of form using ngForm attribute.

<form #loginRef=”ngForm”>

</form>

1. Model driven or reactive forms or mdf

The flow of the application component to template.

It is complex to develop and good for complex form.

The people from typescript and angular background they prefer this type of form. To achieve model driven form we need to use FormGroup, FormControl, FormArray API in component side and formGroup and formControlName attribute in template side. These attribute are part of ReactiveFormsModule. So we need to import ReactiveFormsModule in app.module.ts file.

ng g c login-mdf

In model driven form we need to create the reference of FormGroup and FormControl.

According to model driven approach textfield, passwordfield, radiobutton etc is known as FormControl. FormGroup is collection of more than one formcontrol.

Angular Service : service layer is use to write the business logic.

If we write any simple or complex business logic in component that code become local to that component and we can access within that component or that component template page.

Template component service

Login-tdf.html login-tdf component

Checkuser checkUser()

Login-mdf.html login-mdf component

Checkuser

In Angular we can create two types of service

1. User defined service
2. Creating service class object using new keyword.
3. Creating service class object using DI (dependency injection)

Angular support only constructor base DI.

Create normal typescript class with decorator @Injectable.

Now we have provide the service class details in module level or component level with help of provider attribute.

In component with help of constructor we have pull the object ie DI.

ng new angular-service-read-jsondata

Routing 🡪 no

Styling 🡪 css

1. Pre defined or build in service

Angular provided pre defined API ie HttpClient. This API help use to read the data from json file or read backend technologies REST API.

This API is part of HttpClientModule. So while using HttpClient API we need to import HttpClientModule in app.module.ts file in import section.

Then in user defined service class we have to do the DI for HttpClient API.

Using HttpClient api reference we can use below method.

get(), post(), put() , patch() and delete()

all these methods return type is Observable. Observable is use to handle asynchronous event of data.

Promise

Observable

Observable is a part of RxJS. (reactive js).

To load the data from Observable we need to call subscribe(). Subscribe method take 3 callback parameter

1st : next : it is use to load the data one by one

2nd error : if any error generate while loading data then 2nd parameter get called.

3rd : complete : after loaded the data successfully third parameter get called.

ng g c product : it is use to create the component

ng g class product it is use to create the model Or normal typescript. This class is uses to map the json data.

ng g s product :this command is use to create service class.