Phase 1

Core java and data structure, git and agile

Phase 2

Jdbc, hibernate, servlet and jsp

Phase 3

Spring framework and spring boot, rest full web service and junit

Phase 4 : frontend technologies.

Html, css, Java using ES5 and ES6, Bootstrap

Typescript

Angular framework

Html, css and Basic JS : Self learning

Live session

Adv Js

Typescript

Angular Framework

Day 1 :

10/01/2022

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

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

Client Server

🡨---------------http/https(res) HTML / HTML5

CSS / CSS3

JS

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

Html 1, 2, 3, 4, and 5.0

Cascading style sheet : CSS provided lot of pre defined properties which help to apply formatting style for web page.

Separation of concern. With help of CSS actual concept and formatting style can be write in different file.

JavaScript is/was use to do validation on client side.

HTML :HTML provided lot of pre defined tags

<tagName> opening tag

</tagName> closing tag

Html

Head

Body

Title

P

Heading tags like h1 to h6

H1 largest

H6 smallest

VS Code (Visual Basic Code)

Attribute : attribute is known as properties of tags. For every html tags contains one or more than one attribute. Attribute we have to write in key-value or name-value pairs. Attribute we have to use in opening tags.

<tagName name1=”value1” name2=”value2”></tagName>

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

<a href=”pagename.html”>Text</a>

1, 2, 3, 4, 5

In HTML 4

<!DOCTYPE html url=<http://www.sssafafsafasfsdd.dtd>/> document type definition

This file contains the rules for html file ie root tag name ie html, which contains two tags head and body, inside a head we can write title, style, script, meta etc. inside body we can write more than one paragraph or any tags.

In html5 they remove this rules and added more tags to make html dynamic web page with help of tags.

<!DOCTYPE html >

Header

Footer

Section

Main

Audio

Video

Etc

List tags : These tags help use to display the content in proper format.

UnOrder List

Order List

Definition list

Day 2 :

10/02/2022

CSS : Cascading Style sheet :

CSS provided lot of pre defined attribute which help apply formatting style for web page.

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>

<p></p>

<h1></h1>

<div></div> : container tag which contains more than one tags.

Internal CSS or embedded CSS

Syntax this tag we have to use in head tag of html page.

<style type=”text/css”>

Selector {property:value}

</style>

Types of selector

1. Universal selector : \* {color:red} \* mean all tags, p, h1, div, table etc.
2. Specific selector : tagName {color:red}
3. Class selector :
   1. Local class selector tagName.className(property:value}
   2. Global class selector .className{property:value
4. Id selector : #idName{property:value}
5. Child selector : parentTagName childTagName{property:value}

Id selector and Class selector

Id must be unique for each tag. Don’t repeat same id value for tag it may be tag can be same or different.

Class : class is a collection or group of tags which have same name or different name.

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

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

<p id=”p3” class=”xyz”>This is third para</p>

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

<div id=”p5”>This is first div tag </div>

<div id=”p6” class=”xyz”> This second div tag </div>

<h1 id=”p7 ” class=”abc”>This is h1 first tag</h1>

<h1 id=”p8” class=”xyz”>This is h1 second tag</h1>

External CSS

JavaScript

JavaScript was known as Object based or prototype base interpreter scripting language till ES5 version JS.

But from ES6 onward we can say JavaScript is object oriented interpreter scripting language.

ES : ECMA (European Computer Manufacture Association)

ECMA is a concept. JS is a one of the implementation of ES or ECMA.

In Old Version JS till ES5 there was no class keyword. That JavaScript provided lot of pre-defined object as well as we can create user defined object but not class keyword or class concept. That JavaScript also known as Vanilla JS.

Complier convert whole code into another format at time

Interpreter check the code line by line

From ES6 onward we can use class keyword So from ES6 onwards we can say JS is object oriented interpreter scripting language.

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

document.write(“Welcome to JS”);

</script> closing tag

We have to write this code in between head or body tag.

Variable : In JS to declare the variable we use var keyword till JS5 from ES6 onward we can use let and const keyword.

var variableName;

Data types

JS is known as loosely type data type. Means in JS doesn’t support static data type declaration. Base upon value it behave the variable is that type of data types.

Operator :

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

If statement

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

Looping

while loop

do while loop

for loop

functions :

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

2 types

1. Pre defined function
2. User defined function

Pre defined function

1. alert : alert is use to display pop up message

alert(“msg”)

1. prompt() : This function is use to take the value through keyboards.
2. parseInt(); it help to convert string to integer,
3. parseFloat() it help to covert string to float
4. eval(): it help to convert string to number (int or float)
5. confirm() : it contains two button if user click ok it return true and if user click cancel it return false.

do {

using alert 1: Addition 2 : subtraction

plz enter your choice

using prompt take the choice

switch

case 1: ask the value a and b and display sum convert using eval

break

case 2: ask the value a and b and display sub convert using eval

break

default wrong option

using confirm : ask do you want to continue

}while() : if yes then continue

Using alert : Display Thank you

Day 3 :

10/08/2022

User defined function

1. Normal style syntax

function functionName(parameter) {

function body;

}

1. Function no passing parameter and no return type

function info() {

alert(“Welcome to user defined function”);

}

1. Function passing parameter but not return type
2. Function passing parameter and return type
3. Function no passing parameter but return value.

Events : event provide bridge between html and JavaScript code. Event is interact between user and html tags(html components).

Types of events

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

onClick : button single click

onDblClick : button double click

onMouseOver : when cursor move on image make image big

onMouseOut : when cursor come out from image original size.

onKeyUp : textfield, when release the key after types.

onKeyDown : chatting application she or he is typing

onChange : drop down

onSubmit : mainly we use while doing form validation

onFocus : when we enter in text field

onBlur : when we exist from text field.

DOM and BOM

Document object model

Browser object model

object : object is any real world entity. Person, Bank, Custom, Car, Account

Every object has two things

One properties and behavior

Properties is known as variable or fields.

Behavior is known as function or methods.

In JavaScript object mainly divided into 2 types.

Pre defined object

User defined object.

Pre defined object

JavaScript provided two pre defined object hierarchy ie BOM and DOM

Object 🡪 property

Behavior

Object ----🡪 property

Behavior

Object ---🡪 property

Behavior

Object ----🡪 property

Behavior

BOM hierarchy window is top most object.

Which contains set of properties and behavior as well as another object like history, location, navigator, document etc.

DOM is a property of BOM.

If we want to learn more about browser then we have to work on BOM hierarchy

If we want to learn content of web page then we have to work on DOM hierarchy

Index.html

Html

Head body

Title p Hello (TextNode)

Meta div welcome (TextNode)

Script

Style

When we run the web page on browser internally it will create DOM hierarchy.

DOM : document object model : DOM is api (application programming interface).

Lot of programming language like java, python, C# as well as JS provided lot of pre-defined function or methods or classes which help read, write and update html content dynamically.

Now we will create the new tag and dynamic values

ES6 JS features

From ES6 onwards to declare a variable we can use let and const keyword.

var a=10; int a=10;

a=20; a=20;

var a=30; int a=30; error in java

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

let b = 10;

b =20;

let b =30; error in JS from ES6 onwards.

Using var we can declare global scope but using let we can declare block scope ie can be if or loop etc.

Using const we can declare constant variable like final keyword in java.

const c=100;

c=200; error in JS

function style

1. Normal style function
2. Expression style function
3. Arrow style : arrow function also known as anonymous function. Then arrow function , function keyword replace by =>.
4. Callback function : passing the function name or function body to another function as a parameter is known as callback function.

Day 4 :

10/08/2022

Array : array is one of the pre defined object which help to store the collection of elements of any types and it provided lot of pre defined method which help to do some operation very easily on these elements.

Creating user defined object

In JS we can create user defined object 3 ways

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

JavaScript provided pre defined object ie JSON which help to convert string to json and vice versa.

According to JSON key must in double quote and value can be any types means number, string, boolean, array, complex object type.

JSON :JavaScript Object Notation.

JSON is use to share the data between one technologies to another technologies.

Like Java to Asp.net or php or python

Angular to Spring boot

Scope Object : HTML5 and JS provide pre defined object is localStorage and sessionStorage which help to share the data between one page to another page.

To store the value

localStorage.setItem(“key”,value)

sessionStorage.setItem(“key”,value)

To get the value

localStorage.getItem(“key”)

sessionStorage.getItem(“key”)

to remove the value

localStorage.removeItem(“key”)

sessionStorage.removeItem(“key”);

Synchronous and Asynchronous

Synchronous Statement execution

document.write(“1st statement”);

document.write(“2nd statement”);

document.write(“3rd statement”);

Asynchronous Statement execution

document.write(“1st statement”);

document.write(“2nd statement”);

document.write(“3rd statement”);

Synchronous function call

fun1()

fun2()

fun3();

Asynchronous function call

fun1()

fun2()

fun3();

Synchronous client call

1st

2nd

3rd

Client Server

Asynchronous client call

1st

2nd

3rd

Client Server

Day 5 :

10/15/2022

TypeScript :

JavaScript doesn’t support Data types concepts.

Typescript is a type of scripting language which is also known as super set of JavaScript which support data types.

But All browser doesn’t support typescript directly so we need to convert ts file into js file. To convert ts to js we required transpiler which help convert ts to js.

Node JS :

Node JS is a run time environment for JavaScript program or library or framework.

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.

Node JS provided lot of predefined modules which help to create the server side application, we can create REST API, we can connect database ie mongo db or mysql database etc.

Node JS provided npm (node package manager) it is like a mvn (maven) which help to do download external JavaScript modules.

From node js onward we can run JavaScript program using command prompt with help of node command.

In Node JS doesn’t provide BOM and DOM.

Npm install moduleName –g

Or

Npm install moduleName --location=global

npm install typescript –g

Or

npm install typescript --location=global

Ts support data types

let/var variableName:datatype;

let/var variableName:datatype = value;

we can use private or public access specifiers for constructor parameter variable to make the variable is a type of instance variable.

Typescript interface

Typescript modules : modules is like a package. Modules is a collection of variable, function, classes and interfaces which have same name but different purpose. In Typescript if we want function or classes etc in another file that file itself is known as modules and using import and export keyword we connect both file together.

Bootstrap : Bootstrap is open source web css framework which provided lot of pre-defined css classes with respective dom elements. Like div, p, h1 to h6, forms etc.

Using bootstrap we can create responsive web application base device.

Now a day we can create responsive web application using html5 features.

Day 6 :

10/16/2022

Gridlayout : bootstrap gridlayout help to arrange the component in row and columns.

By default each row divided into 12 columns.

Angular JS

Base upon html, css and JavaScript

Base up MVC architecture.

Angular Framework

Base upo html, css, js and typescript

Angular Framework : Angular is open source web framework provided by google company. In angular Framework controller is replaced by component.

Component is use to controller the view or part of the view page.

Using Angular we can create SPA (Single Page application).

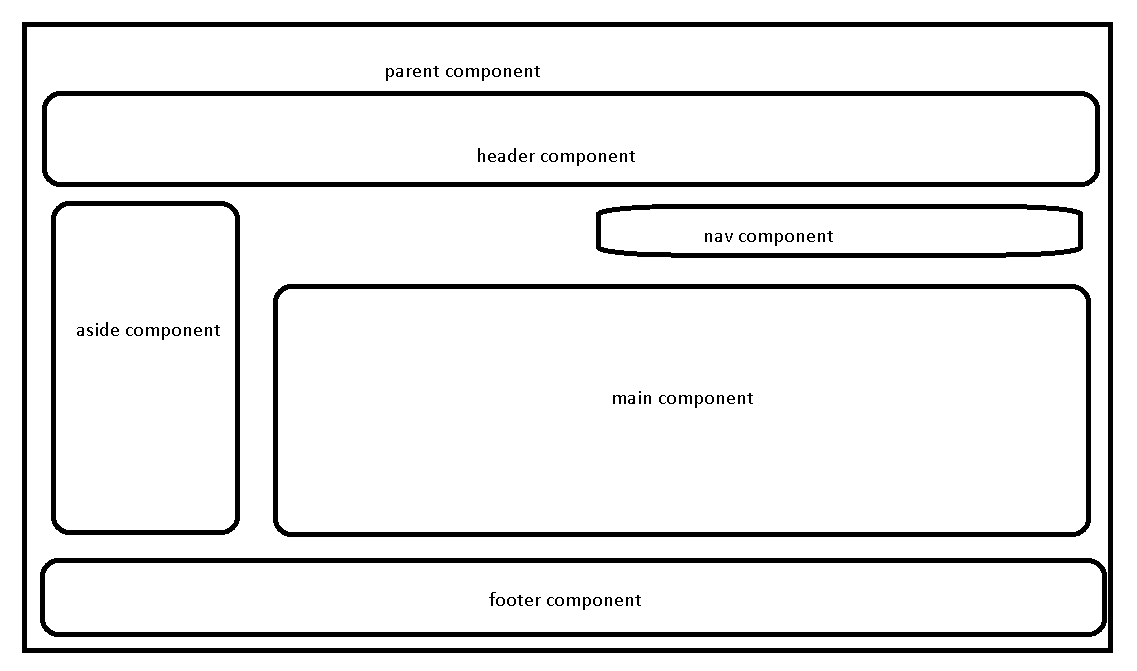
Angular and React Js

Index.html or index.jsp welcome.html

Hyper link

Submit button

Js



Angular cli (command line interface). Using angular CLI we can create the project.

npm install @angular/cli –g

Or

Npm install @angular/cli --location = global

ng version

create angular project folder

inside that folder create angular project

ng new project-name

ng new demo-app

do you need routing 🡪 N

styling --🡪CSS

after project created please move inside a project folder

cd demo-app

ng serve this command is use to run the angular project

after project compiled 100% open the browser and write url as <http://localhost:4200>

open the project in VS code

expand

src

app

app.component.html ---🡪 template file

app.component.css -🡪 css styling file

app.component.ts 🡪 component file

decorator : decorator is like a annotation is java which provide meta data information about function or classes or properties.

Angular created lot of pre-defined decorator

@Component

@NgModule

@Injectable

@Input

@Output

@Pipe etc

Using @Component decorator we can make normal class as component class in angular.

@Component decorator contains lot of pre-defined attribute

selector : This attribute provide tag name or user-defined tag.

templateUlr: this property help to connect the html page.

styleUrl : This property help to connect css file like link tag.

app.module.ts -🡪 module file

module is collection of component, service, pipe etc.

@NgModule

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

imports : in this section we have to import pre-defined or user defined modules.

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

bootstrap : in this section we provide parent component details that component html page consider as first page or landing page of the application.

We create lot of modules

Login module account module product module employee module

Every module contains more than one component and each component connect to html page.

From main.ts file we will get the details about parent modules.

Angular.json file open

From here we will get the details about main.ts file and index.html page.

ng new angular-data-binding

routing 🡪 no

styling 🡪 css

cd angular-data-binding

ng server –o :after compiled program 100% it will open automatically in default browser.

Angular cli provided command which help to create the new component

Inside a project folder open the command prompt

ng generate component componentname

or

ng g c componentname

Data binding : Data binding provide bridge between template to component. Using data binding we can share the data between template or html and component ie ts file.

2 types

1. One way data binding

Uni-directional it may be component to view or view to component.

1. String interpolation : it is a type of one way data binding. The flow of the application from component to view or template.

To achieve one way data binding we use {{}}

{{varialbeName}} {{fname}}

{{expression}} {{5+7}}

{{functionCall}} {{display()}}

{{}} we have to use in template or html side.

1. Property binding : it is a type of one data binding : the flow the application from component to view.

To achieve property binding we have to use []

<input type=”text”/>

<input type=”text” value=”Raj”/>

<input type=”text” value=”lname”/>

<input type=”text” [value]=”lname”/> it search lname variable inside a component and the value of lname it will display inside a text field.

<p [innerText]=”lname”></p>

ng g c property-binding

1. Event binding : event binding also type of one way data binding. The flow of the application template to component.

To achieve event data binding we use ()

Angular use same event provided by JavaScript only different on pre-fix remove and event name wrap inside a parenthesis .

JavaScript Events Angular Event

onClick (click)

onDblClick (dblclick)

onMouseOver (mouseover)

onKeyUp (keyup)

onSubmit (ngSubmit)

etc etc

ng g c event-binding

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

We want to pass the value from template to component.

Using template reference.

<input type=”text” #nameRef/>

1. Two data binding : using ngModel attribute

Bi directional

It automatically sync if we do any change in component or template or template.

29-10-2022

To achieve two way data binding angular provided pre-defined attribute ie ngModel. ngModel attribute is a part of FormsModule so we have to import FormsModule in app.module.ts file in import section.

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

ng g c two-data-binding

Angular Forms

Angular forms is use to share or pass the group of value in the forms json from template to component.

1. Template Driven form or TDF

The flow of the application in TDF is view to component.

It is easy to use.

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

In this type of form we have two pre-defined attribute ngForm and ngModel. These two attribute part of FormsModule. So we have to import FormsModule in app.module.ts file

In this type of form we write more code in template side

Using ngForm attribute we have to create the reference of form

<form #loginRef = “ngForm”>

</form>

1. Model Driven Form/ Reactive Form or MDF

The flow of the application in MDF is component to view.

It is complex.

The people form typescript and angular background they prefer this type of form.

In this type of forms we have to use set API component ie FormsGroup and FormControl,FormArray etc. In template side we have to use formGroup and formControlName attribute. These attribute part of ReactiveFormsModule. So we have to import in app.module.ts file in import section.

In this type of form we write more code in component side.

In model driven form textfield, passwordfield, radiobutton etc are bind with FormControl API. FormGroup is a collection of more than one FormControl.

ng new angular-forms

Routing 🡪 no

Styling 🡪 css

ng g c login-tdf

ng g c login-mdf

Angular Service :

If we write any simple or complex logic in component that code become local to that component as well as that template. So if need to globally access in more than one component then we can use angular service concept.

Template component service

Tdf checkUser()

Function

Mtd checkUser()

Angular service mainly divided into two types.

1. User defined service
   1. Creating object using new keyword.
   2. Creating object using DI concept.

We have to create the user defined class with decorator as @Injetable

We have to provide this class details in app.module.ts file in provider attribute.

(we can register in module level or component level or providerId attribute in @Injectable decorator).

Angular support constructor base DI. It doesn’t support setter base DI.

1. Built in or pre defined service

Angular provided pre defined API ie HttpClient. HttpClient part of HttpClientModule. So while using HttpClient we need to import HttpClientModule in app.module.ts.

In user defined service class we have to do DI for HttpClient API.

This API provided lot of pre-defined method ie get(), post(), put() and delete() which help load the data from static json file or backend technologies rest api.

HttpClient all method ie get(), post(), put() and delete() method return type is Observable.

Observable is part of RxJS(reactive js). Observable is use to load the event of data in asynchronous format. If we want to load the data from Observable we need to use subscribe method. Subscribe method take 3 parameter as callback

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

2nd error :if any error generate at the beginning or middle or last. Then 2nd parameter get called.

3rd completed : after loaded all data successfully then 3rd parameter called.

ng new angular-service-read-json-data

routing 🡪 no

styling 🡪 css

We have to product json data in products.json file

ng g c product : this command create product component

ng g class product : it is use to create the model. (this class is responsible to map the json data retrieve from json file or backend restfull web service).

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

30-10-2022

Angular providing structure directive

\*ngIf

\*ngFor

<div \*ngIf=”true”>It display these contents</div>

<div \*ngIf=”false”>It hide these contents</div>

<div \*ngIf=”f1”>It display these contents</div> f1 must be Boolean part of component

<div \*ngIf=”f2”>It hide these contents</div> f2 must be Boolean part of component

<div \*ngFor=”let obj of arrayVariableName”>

<p>{{obj}}</p>

</div>

Angular routing

Using angular routing we can navigate from one component to another component base upon path provided in routing file.

Ng new angular-routing

Routing -🡪 yes

Styling 🡪 css

ng g c aboutus

ng g c contactus

ng g c feedback

ng g c login

ng g c dashboard