**11 Days**

HTML/HTML5 and CSS/CSS3 --🡪 3 days

JavaScript using ES5 and ES6 --🡪 3 days

Overview of Node

TypeScript

Angular Framework 11 --🡪 5 days

**Day 1- 30-03-2021**

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

Uniform resource locator.

----🡪Req(https/https)--🡪

Client Server

🡨-- res(http/https)

HTML/HTML5

**Hyper Text Markup language** CSS/CSS3

**Cascading Style sheet**

JS

**JavaScript**

http : hyper text transfer protocol

protocol : set of rules which help to communicate more than once device.

S : secure

www: world wide web

google : domain or server name or app name.

com : commercial

HTML/HTML5--🡪 It is use to display the content on web page(browser).

CSS/CSS3 --🡪 : Look and feel for the content or presentation for the contents.

JS (JavaScript) -🡪 Actions on contents or event on contents or programming on web page.

**HTML** : Hyper text mark up language : HTML is use to create the web page.

Web page is use to display the contents in different format like text, bold, italics, form, table format, audio, video etc.

HTML is very simple language. HTML provide lot of pre-defined tags or elements which use to create the web page.

HTML is case insensitive.

HTML tags syntax

<tagName> opening tag

</tagName> closing tag

<tagName/> self-closing tags

HTML tags

1. Html
2. Head
3. Body

<html>

<head>

</head>

<body>

</body>

</html>

1. **Title tag :** This tag is use to display the message in title bar of web page.

This tag must be in between head tag.

1. Paragraph tag : This tag is use to display the message in browsing area.

<p> </p>

1. Break tag : <br> or <br/>
2. Heading tags : heading tag is use to write heading on paragraph or other tags.

6 types of heading tag

h1 to h6

H1 largest

H6 smallest

**Attribute :** Attribute is known as properties of tags.

Syntax of attribute

<tagName attributeName=”value” attributeName=’value’ attributeName=value>

</tagName>

Attribute we have to use in opening tag in the form key-value pairs. Where value may be single or double or without any quote.

Separation between 2 attribute are space.

P and heading tags.

align=’left/right/center’

**font tag :** This tag is use to change the size, color and style(face).

Hyper link :

1. External hyper link
2. Internal hyper link or bookmark(one page html app).

External hyper link

<a href=”pageName.html”>Text</a>

a🡪 anchor tag

href 🡪 hyper reference.

**Internal Hyper link**

<a href=”#a1”>Text</a>

**Add Images**

<img src=”imageName.jpeg/gif/png/” />

img : image

src: source

**list tags**

This tag is to display the items or contents one by one

1. Unorder list tag

<ul>

<li></li>

</ul>

ul : Unorder list

li : list item

1. Order list

<ol>

<li></li>

<li></li>

</ol>

Ol : Order list

Li : list item

1. Definition list

<dl>

<dt></dt>

<dd></dd>

</dl>

Dl : definition list

Dt : definition term

Dd : definition description

**Table Tag :**

Employee details

**Id Name Salary**

100 Ravi 12000

101 Ramesh 14000

103 Lokesh 16000

Table ->

Tr --🡪 table row

Th-🡪 table heading

Td -🡪 table data.

Form tag

Login page

Login Page

UserName TextField

Password PasswordField

Submit Reset

Form tag

<input type=”text/password/radio/checkbox/button/file/submit/reset”/>

Query Param concept : the information send through URL.

URL?name=value&name=value&name=value&name=value

In form tag we can use method attribute to mention the type of method.

By default all html form method is consider as **get**.

If method is get. Information send through URL using query param concept.

Get method is not a secure.

If we want to achieve security we have to use method = post

If method is post data send through body part of http request.

The performance wise get the faster than post.

Using get we can send maximum 255 character data through URL.

Application.html

**FirstName TextField**

**LastName TextField**

**Gender RadioButton Male RadioButton Female**

**Hobbies checkbox Reading checkbox playing**

**City Dropdown**

**Address textarea**

**UserName TextField**

**Password PasswordField**

**Create Reset**

<file:///C:/Users/91990/Desktop/Angular%20Training/UI%20Programs/HTML%20Programs/home.html?fname=Raj&lname=Deep&gg=male&hh=reading&hh=playing&city=Bangalore&add=Kar>

HTML is not a structure.

HTML 4.x

<!doctype HTML public uri=”………………………..dtd”

<html>

</html>

Document type definition : This file contains all tags rules details. Like root tagName html, contains number of child tags like head and body Then body must contains more than one p, h1 to h6.

HTML 5

<!doctype HTML> : Given the instruction to browser we are using html 5 features. (optional).

IDE : Integration Development Environment

Notepad ++

Eclipse

Netbean

Bracket

ATOM

VS Code (Visual Studio Code) :

Develop small web application which contains html5 features.

Online shopping.

Login page :

Dashboard page :

4 hyperlink

First page

Second page

Third page

Fourth page

**Day 2- 31-03-2021**

CSS : Cascading Style Sheet

Without CSS if we want to apply any formatting style we have to depends upon the other tags or other tags attribute.

Without CSS actual contents and formatting style both are mix together.

We can’t achieve separation of concern.

So with the help of CSS we can apply more formatting style which is not possible with the help of tags.

Using CSS we can achieve separation of concern (actual contents and formatting style separate).

Types of CSS

1. Inline CSS
2. Internal CSS or embedded CSS
3. External CSS

Inline CSS : Syntax

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

Body, p, h1 to h6,div, span, form etc.

Internal CSS or embedded CSS

Syntax

<style type=”text/css”>

Selector {property : value;property:value}

</style>

This tag we have to write in between head tag.

Different types of selector

1. Universal selector : \* {property:value}
2. Specific selector : tagName {property:value}
3. Multi specific selector : tagName,tagName,tagName {property:value}
4. Local class selector : tagName.className {property:value}
5. Global class selector .className {property:value;}
6. Id selector : #IdName {property:value}

Class selector : more than one can be belongs to same class or class is groups of tags.

Class can contains same type of tags or different type of tags.

Id selector : to make tag unique we have to use the Id for that tag.

p.abc {}

.xyz{}

<p class=”abc” id=”p1”></p>

<p class=”xyz” id=”p2”></p>

<p class=”abc” id=”p3”></p>

<h1 class=”xyz” id=”hh”></h1>

Using ID in JavaScript or jQuery we can read, write and update the content dynamically.

Using internal or embedded CSS the rules apply for only one web page.

External CSS file.

**Font family related property**

Every html tag follow Box Model concept.

Every tag surrounding contains padding, border and margin.

Padding and margin we can’t see but we can see border.

**Semantics tags**

**CSS3 Properties**

CSS3 transform property : This property allow you to translate, rotate, scale and skew element or html tags.

CSS3 transitions property : This property allow to change property values like size, color, background colour smoothly, over a given period of a time.

2 points

1. CSS property which we want to change
2. The duration for to get the effect.

Bootstrap : Bootstrap is a open source CSS framework. Bootstrap is use to create the responsive web application depending upon the device. Like Laptop, Computer, Mobile etc.

Using HTML5 Meta tag we can achieve responsive web application.

<meta name="viewport"

    content="width=device-width, initial-scale=1.0">

This tag responsible to read the device with and align the html contents.

Here <meta> meta-data (data about data)

Viewport is a attribute name. gives the instructor to browser how to control the page’s

dimension and scaling.

The initial scale 1.0 parts set the initial zoom level when the page is first loaded by the

browser.

Sytles.css

.fontClass {

font-size:20px;

font-family:

color:

}

Bootstrap provided lot pre-defined CSS classes depending upon the tags.

Container and container-fluid

Container : container is a pre-defined class provided by bootstrap which use fixed width depending upon the device.

Container-fluid : This type of class take the full with of the viewport or device.

<div>

</div>

­Button related classes

     <input type="button" value="click Here"/><br/>

     <input type="button" value="click Here" style="color:red;background-color: yellow;"/><br/>

     <input type="button" value="click here" class="btn"/><br/>

     <input type="button" value="click here" class="btn btn-success"/><br/>

     <input type="button" value="click here" class="btn btn-secondary"/><br/>

     <input type="button" value="click here" class="btn btn-primary"/><br/>

     <input type="button" value="click here" class="btn btn-danger"/><br/>

     <input type="button" value="click here" class="btn btn-info"/><br/>

     <input type="button" value="click here" class="btn btn-link"/><br/>

     <input type="button" value="click here" class="btn btn-dark"/><br/>

     <input type="button" value="click here" class="btn btn-success" style="color:blue;background-color: cornsilk;"/><br/>

**Grid layout :** Bootstrap Grid layout system use a series of container, rows and columns to layout and align the contents.

Grid layout each row divided into 12 columns.

Xs : extra small < 576px

Sm :small >=576px

Md : medium >=768px

Lg : large >=992px

XL extra large >=1200px

Day 4 : 02-04-2021

JavaScript

JavaScript was object based interpreter scripting language up to ES5.

Object based Vs Object oriented : Java, C++, C#

Class, object, Inheritance, Encapsulation, Polymorphism, Abstraction

Interpreter Vs compiler : both are translator which convert one format to another. Compiler convert whole code at time. Interpreter convert line by line.

ES :ECMA : European Computer Manufacture Association.

ES is a concept. The implementation of ES is JavaScript.

ES5

ES6, ES7 etc.

If we want to do programming on web page. We can use JavaScript.

If we want to write JavaScript we have to use the script tags.

Syntax

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

document.write(“Welcome to JavaScript”);

</script> closing tag

We can write more than one script tag in head part or body part of html code.

In JavaScript it is not mandatory every statement end with semicolon.

JavaScript is case sensitive.

JavaScript can’t understand any HTML tags. but if want to use the any html code inside a script tag we have to use inside a double quote.

**Variable and data types**

In JavaScript to declare the variable we use var keyword.

Syntax

var variableName;

dynamic data type or loosely data type concept.

var a; // default value for that variable is consider as undefined.

var n=10; //now n is consider as number data type

var m=10.10; //number type consider

var name=”Ravi Kumar”; //string type consider

var res = true; //Boolean type consider

var obj = new Date(); //object reference consider

**Operators :**

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

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

Logical operator : && , || !

Increment and decrement : ++ --

Assignment operator : =

Triple operator : ===

Ternary operator : condition?true:false;

Typeof :

If statement :

1. Simple if

If (condition) {

}

1. If else

If(condition) {

}else {

}

1. Nested if : if within another if

If(condition) {

If(condition) {

}else {

}

}else {

If(condition ) {

}else {

}

}

1. If else if

If(condition ) {

}else if(condition ) {

}else if(condition) {

}else {

}

switch statement

syntax

switch(variableName) {

case v1: block1;

break;

case v1: block2;

break;

default : wrongblock;

break;

}

switch, case, break and default are keywords.

variableName : it must be type of number(without decimal) , char or string values.

Loop : Looping is use to execute the set of statement again and again till the condition the conditions become false.

Initialization

Condition

Increment / decrement

While loop : entry loop

I=1,n=10;

while(i<=n) { depends upon the condition

Task

I++ or n--;

}

Do while loop : exit loop

I=1,n=10;

do {

task

i++ or n--

}while(i<=n);

For loop

Syntax

1 2 4 fixed iteration

for(initialization;condition;increment/decrement) {

body of the loop; 3

}

function and events

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

2 types

1. Pre-defined function
2. alert(“Msg”); This is use to display the message.
3. prompt(“Msg”) : This is use to take the value through keyboards.
4. parseInt() : convert string to integer
5. parseFloat() : convert string to float
6. eval() : convert string to number (int or float).
7. Confirm() : This function with alert message contains two button ok or cancel. If user click ok then it return true or return false.

Do

Alert 1: Add, 2 :sub

Using confirm

Switch

Case 1 Add

Eval, parseInt parseFloat

Case 2 Sub

Confirm : to continue or finish

while(variableName)

Alert : finish

1. User-defined function

Normal creation of user-defined functions

Syntax

function functionName(parameterList) {

}

Event : Interaction between user and component (HTML tags).

Or

Event provide the bridge between HTML code and JavaScript.

Type of events :

In JavaScript all events are start with pre-fix on followed by name of the events.

onClick

onDblClick

onMouseOver

onMouseOut

onKeyUp

onKeyDown

onChange

onBlur

onFocus

onSubmit

onLoad etc

DOM : Document Object model :

We can get the value of HTML forms component using name attribute or id attribute.

Name : more than one tag can contains same name like radio button, checkbox etc

To get the value using name attribute

var allNames = document.getElementsByName(“user”);

//firstName

var fname = allNames[0].value;

var lname = allNames[1].value;

FName : <input type=”text” name=”user”/>

LName : <input type=”text” name=”user”/>

[100,200,300];

Form Validation

1. Using JavaScript
2. Using HTML5 features

Git : Git is open source sub version control software or tool. Which help to records every update done one every team or person in file or projects.

Git repository : it is folder which contains more than one file or sub folder.

Local repository :

Syntax to create the local repository

**git init : This command is use to create the local repository**

**git status : : This command is use to check the status of local repository.**

**git add filename : This command is use to add the file to staging area.**

**git commit –m “Message” : This command is use to pass the file from staging area to local repository**

if we do any change in folder like added new file, updated file, delete the files.

**git add .**

**git commit –m “message for that change”**

**Remote repository: remote folder git hub, aws, azure etc.**

**Git hub : create the account the create the remote repository**

**To connect local repository to remote repository we have to execute the command as**

git remote add origin <https://github.com/Kaleakash/abcde.git>

now added now we have to push the data from local repository to remote repository using command as

git push –u origin HEAD

next time if we do any changes we have to fire 3 commands.

git add .

git commit –m “message for commit”

git push –u origin HEAD

**Day 5- 03-04-2021**

JavaScript Object Hierarchy

object : object is a real world entity.

Properties or State have variables / fields

object

behaviour do/does functions / methods

person

bank

animal

car

laptop

pen

In JavaScript object are divided into 2 types

1. Pre-defined objects
2. User-defined objects

BOM Hierarchy : Browser Object Model

DOM Hierarchy : Document Object Model

object 🡪 property / variables

Behaviour / functions

object

-🡪property

Behaviour

Object

* Property

Behaviour

Object



window is a top most object in BOM Hierarchy

BOM hierarchy object are useful when we want to get the details about the browser.

History, location, screen etc.

document is a top most object in DOM hierarchy

document object provide the details about the web page contents.

Using DOM Hierarchy we can read, write and update the HTML contents dynamically.

setTimeout()

setInterval()

clearInteval()

In JavaScript we can achieve two type of communication

1. Synchronous communication
2. Asynchronous communication

Statement wise

L1 Statement

L2 Statement

L3 Statement

If the program execute sequentially. L2 will execute after L1. L3 will execute after L2.

This concept is also known as synchronous statements.

Function call

dis1()

dis2();

dis3();

If the execution call execute sequentially. Dis2() start execution after dis1() finish. Dis3() start execution after dis2().

1st req

2nd req

3rd req

4th req

Client Server

2nd request execution start after 1st request acknowledgement.

3rd request execution start after 2nd request acknowledgement.

Asynchronous communication

N number of statement or function call or request execute independently.

JavaScript is a single thread scripting language.

Java is a multi threaded programming language.

setTimeout()

setInterval()

clearInteval()

These are pre-defined methods part of window object which will execute asynchronously.

setTime(1st,2nd)

1st parameter function name, 2nd parameter time to execute the code.

**In CSS**

font-size

background-color

JavaScript apply style using id

document.getElementById(“idName”).style.color

document.getElementById(“idName”).style.fontSize

document.getElementById(“idName”).style.backgroundColor

Basic Pre-defined objects or utilities objects.

String

Date

Array

Boolean

Math

**User-Defined object using ES5 style**

object : any real world entity

properties

Person

Behaviour

Bank

Animal

Car

Class : class is a blue print of object or template object.

But up to ES5 if we want to describe the object we have to take the help of function style syntax.

**ES6 JavaScript**

Etc

JSON

Etc

**Day 6 05-04-2021**

**ES6 Features**

1. **From ES6 onwards to declare the variable we can use var, let and const keywords.**

**var is use to declare global scope**

**let is use to declare local scope or block scope.**

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

**Using let we can’t re-declare same variable.**

**const is use to declare constant variable which we can’t change the value. const is use to declare the local scope or block scope.**

1. **Type of loop**

**while loop**

**do while loop**

**for loop**

**for in loop**

**using in loop we can get the index position.**

**for(let index in arrayName) {**

**}**

**for of loop**

**using of loop we can get the value**

**for(let val of arrayName) {**

**}**

**ES5 and ES6**

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

**callback function may be synchronous or Asynchronous.**

**Array using forEach() with callback concept.**

**Type of functions**

**Rest Operator or Parameter : it is use to receive 0, 1 or many parameter like a array but in array we have to pass mandatory empty array.**

**In function while declaring rest operator or parameter we have to use …variableName**

**Arrow function : arrow function is a type of anonymous function without name. It is a short cut of anonymous function.**

**Syntax**

**let/var functionName = ()=> code**

**OOPs concept using ES6 style.**

**object : any real world entity.**

**Properties or state have**

**Object**

**Behaviour do/does**

**Angular JS : class style**

**React JS : class style and function style.**

**constructor : it is a type of special function which help to create the memory.**

**Points**

1. **To write the constructor we have to use the function name as the constructor.**
2. **It will call automatically when we create the object.**
3. **In the life of the object if we want to perform any task only one time then we have to use the constructor.**
4. **If ES6 we can’t write more than one constructor within a same class.**

**Inheritance :**

**Inheritance is use to inherits the properties and behaviour of old class to new class.**

**To achieve inheritance we can use extends keywords in ES6.**

JSON : JavaScript Object Notation : JSON format data is use to shared the data between two technologies using web service concept.

Web Service : Giving the service for web application when both application running using different language like Java, .net, Python, PHP, Angular, React.

Java (req)

HDFC XML/JSON HSBC

.net

Java .net

Like map in Java.

JSON store data in the form of key value pairs.

According to json key must type of string and it must be unique.

Value may be number, string, Boolean, array type, complex object type.

Syntax

{“key1”,value1,”key2”,”value2”,”key3”:”value3”}

Key must be string and value may be any other data types.

Scope object

2 type of scope object provided

sessionStorage

localStorage

we can move from one page to another page using button, hyperlink or using javascript function. So we want to shared data between one page to another page we can take the help of scope or storage objects.

sessionStorage hold value still session get destroy. Once application close all details from session scope destroy.

localStorage hold value in machine permanently.

Both pre-defined objects provided pre-defined methods like

sessionStorage.setItem(“key”,value); value always type of string consider.

sessionStorage.getItem(“key”);

sessionStorage.removeItem(“key”);

**Day 7 06-04-2021**

ES5 and ES6 :

**Promise :** Promise is pre-defined object it may resolve and reject depending the conditions.

If we want to call backend service (REST Service ). We use promise Object in JavaScript.

Promise use Asynchronous callback function to get resolve (success) or reject (failure) details.

User – defined promise

First we have to create the Promise object.

var obj = new Promise((resolve,reject)=>{

})

obj.then().catch();

then get call if promise is resolve. Catch get call if promise rejected.

Backend service can be created using any technologies like Java, .net, php, python, node js.

**Web Service : RestFull Service API or REST API**

Front end technologies backend

Html/css/javascript ES5/ES6 Java or Spring boot

JavaScript REST API .net (asp.net)

Python

Angular php

React JS node js

Vue JS

Using Promise object we can call REST API develop in any language.

URL : Store/Retrieve / Update / Delete

XML / JSON

REST API or REST CALL : URL

1. Few URL are to store the records
2. Few URL are to retrieve the records
3. Few URL are update the records
4. Few URL are delete the records.

fetch : It is a pre-defined function provided by javascript which help to call backend service.

Return type of fetch function is promise object.

Promise using fetch load whole data at time.

Angular 11 : Angular is a open source framework provided by Google company.

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

JavaScript library

JavaScript Framework

jQuery library : jQuery is a type of external JavaScript library. Library contains lot of pre-defined function which internally connected to each other to do some task.

Without library in JavaScript if we want to do any DOM Operation like Read, Write, Update. We have to write the code from beginning or we can take the help of document.getElementById() like methods.

But with library we can do same task very easily.

Limitation of jQuery or any other library.

Library doesn’t follow standard. When the code become complex to maintain that code very difficult.

Framework : Framework follow standard rules and regulation. The implementation of all design pattern taken care by framework. Design pattern ( Best practise or solution of repeating problem).

Framework taken 70 to 80 task. Hardly we have to write 20 to 30% code to make final product. Framework also known as template.

Angular Framework

Multi page application

One.html two.html

Hyperlink

Button

Submit

When we navigate from one page to another page the whole dom get loaded.

Single Page application

In Single page application rather than loading whole page we load only part of the page.

Angular JS : HTML/CSS/JavaScript ES5 1.0,1.1 1.x

Angular Framework : Angular 2 to 11

: HTML/CSS/JavaScript ES5 and ES6 /TypeScript

Nod JS : Node JS is a run time environment for the JavaScript program or library or Framework.

Node Js contains lot of pre-defined modules (core module or external modules). With the help of those modules we can do server side programming language.

Modules like a package or namespace. Module is a collection of classes, functions, variable, interfaces etc.

Before Node JS JavaScript only for Client Side Scripting language.

But After node JS JavaScript can do on client side as well as server side scripting language.

Using Node JS we can do file handling, we can create web application, we can connect database like

MySQL or Mongo Db database.

We can execute the JavaScript program without taking help of html page ie using node commands.

In Node JS program we can’t use BOM and DOM

Means we can’t use window and document. But apart from window and document object other coding part of JavaScript we can achieve.

So to display welcome message through JavaScript. We have to use console (pre-defined global object).

TypeScript : TypeScript is a super set of JavaScript. TypeScript is one of the implantation of ES6.

TypeScript support all ES6 Features.

TypeScript support data types concept.

When write the code in TypeScript. The file extension must be .ts

Ts file can’t understand by Browser.

So we have to convert TS to JS

To convert TypeScript to JavaScript it require transpiler (compiler and interpreter).

Typescript transpiler (tsc) help to convert TS to JS.

Name of the transpiler itself is typescript.

To install tsc we have to take the help npm

npm : Node Package manager. This is by default enable with node software. Which help to download external modules.

Command to install any external module

npm install –g moduleName : globally

npm install moduleName : locally

npm install -g typescript

create the typescript file with extension .ts

to convert ts to js

tsc filename.ts

then

node filename.js

**DataTypes : number, string, boolean, any etc.**

let variableName:datatype;

let num:number;

to convert ts to JS (ES5)

tsc filename.ts

to convert ts to JS (ES6)

tsc filename.ts --target ES6

let, const and var

Array with datatypes.

Functions :

ES5

ES6

TypeScript

Number of parameter, type of parameter and return type must match base upon the datatype.

OOPs Concept.

Using constructor short cut initialization for instance property

Inheritance :

class Employee {

}

class Manager extends Employee {

}

**interface**

syntax

interface Abc {

id:number;

display():void;

}

Interface contains variable and incomplete method means method without body.

**Modules :** modules contains set of function, classes, interfaces, variable,

It is like a package in java.

We take the help import and export keywords to connect more than one file’s functions, classes, interfaces and variables as well as when two function, classes, interfaces have same name we can create the alias.

**Decorator :** It is a type of special function which add extra behaviour to class, function or property. It is also known as meta – data (data about data). Decorator like a annotation in Java.

All decorator start with @ followed by decorator name with set of property.

Angular is a open source framework provided by google company. Angular is use to create SPA (Single Page Application).

To create the project manually more complex.

Angular CLI (Command Line Interface)

To enable Angular CLI we have to install external module as

npm install –g @angular/cli

ng : next generation for HTML or Web Page

Angular

To create new angular project we have to write the command as

ng new project-name

ng new welcome-app

policy option it will ask yes/no

routing 🡪 no

style 🡪 css

after project created successfully move inside a project folder using cd command

cd project-name

to run the project we have to use the command as

ng serve

if again ask the policy given the option as yes/no

after compiled 100% successfully.

Open the browser and write the URL as

<http://localhost:4200>

Angular application running on server with default port number 4200.

app.component.html --🡪 template page

Template page parse (compiled ) by angular.

app.component.ts -🡪 component file (type of ts file)

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

Now to make normal class as a component angular provided decorator as

@Component

import { Component } from '@angular/core';

This code is use to point to the file where @Component decorator rules written.

@Component contains lot of properties

1. Selector : name for the component or user-defined tagname.

Using angular we are creating user-defined tags.

<p>

<h1>

<form>

1. templateUrl : This property is use to connect html page. This html content replace on tag where we use.
2. stylesUrl : This property is use to connect the external css file like link tag in html.

The ts, html file and css file connected to each other using @Component

<app-root></app-root>

app.component.html code get replace.

<app-root></app-root>

ng serve - -port 4230

app.module.ts

module : it is a collection of more than one component.

@NgModule: This decorator is use to make the class is as a module.

import { NgModule } from '@angular/core';

@NgModule

1. **declaration** : In this section we have to provide the details about all our component.
2. **import** : This property is use to import pre-defined or use-defined modules. BrowserMoudle this module help to display the output in browsing area.
3. providers : []
4. Bootstrap : []: In Bootstrap we can provide the main component or parent component will open first.

Angular 2 to 11 is use to create the web application , mobile, desktop etc.

main.ts

LoginModule Application Module Customer Module

CreateLoginComponet C1 C1

DisplayLoginComponent c2 C2

In Angular one html page is equal to one component.

In main ts file we get the details about main module which has load first

platformBrowserDynamic().bootstrapModule(AppModule)

  .catch(err => console.error(err));

index.html

contains the root component selector name as a tag

angular.json file provide the details about index and main file.

Create new project

ng new angular-data-binding

Data binding

Data binding is use to share the data or variable or property from component to template(view) and vice-versa.

Data binding provide the bridge between component to template (view).

2 types

1. One – way data binding
   1. String interpolation – Component ----🡪 View (Template )

{{variableName}}

{{10+20}}

{{sayHello()}}

* 1. Property binding – component -- > view (template)

<input type=”text” [value]=”variableName”/>

<div [innerText]=”variableName”></div>

<span [innerText]=”variableName”></span>

* 1. Event binding : view --- > Component

Angular using same event provided by JavaScript but with different syntax

In JavaScript all event start with pre-fix on followed by event name But in Angular on prefix removed and wrap the event name using parentheses.

onClick (click)

onDblclick (dblclick)

onSubmit (ngSubmit)

onMouseOver (mouseover)

To receive the value of textfield, passwordfield, radiobutton etc.

We have to use template reference.

Syntax

<input type=”text” #refName />

Calculator

1st Number TextField 10

2nd TextField 20

Result TextField Result

Sum Sub

**Using Event and String interpolation or property binding**

**Two way data binding**

View 🡨-----------------🡪 Component

If we do any changes in view automatically updated in component and vice – versa.

To achieve 2 way data binding angular provided pre-defined attribute ngModel

We have to using in textfield, passwordfield, radiobutton etc.

[(ngModel)]=”variableName”

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

If we are trying to achieve 2 way binding using ngModel attribute. We have to import FormsModule inside a app.module.ts file. ngModel attribute is a part of FormsModule.

Directive : add extra functionality or behaviour to DOM elements.

Type- of directives tags

Angular divided into 3 types.

1. Component directive

@Component({

selector :”MyTag”,

template/templateUrl:

styles/styleUrl:

})

export class AppDemo {

v:number=100;

}

Component is a type of directive which help to control the view or part of the view page.

1. Structural directive : using structure directive we can add or remove DOM elements.

\*ngIf

\*ngFor

1. Attribute directive : attribute directive help to apply styling for web page using events.

ngStyle and ngClass

First create the project

ng new angular-directive

after project created successfully

cd angular-directive

three component create

ng g c component-directive

ng g c structure-directive

ng g c attribute-directive

Creating the component using command prompt

ng generate component componentname

or

ng g c componentname

**Day 9 : 8-04-2020**

Attribute directive : Attribute directive is use to apply css for the template.

ngStyle and ngClass

Angular-Forms

In Angular we will create 2 types forms

1. Template Driven Form (TDF)
2. Model Driven Form or Reactive Forms (MDF)

Template Driven Forms :

1. It is easy if you know html code
2. The flow of this form is Template (View ) --🡪 Component
3. Good for simple forms.

In Template Driven form we have to create the reference of form using ngForm attribute.

Syntax

<form #loginRef=”ngForm”>

</form>

ngForm attribute is a part of FormsModule. So we have to import FormsModule in app.module.ts file.

If we want to pass the value through form reference we have to use ngModel attribute for every textfield, password, radio button ec.

**Model Driven Form :**

Model Driven Form : Flow from component to template. This type of form is good for complex forms. It is a good for those people working typescript.

According to model driven form textfield, radiobutton, checkbox, are known as formcontrol. Form control must be wrap in formgroup.

FormGroup 🡪which can contains more than one FormControl like textfield, passwordfield, radiobutton, checkbox, etc.

FormGroup also can contains another FormGroup that FormGroup contains more than one FormControl.

Login Page -🡪FormGroup

TextField FormControl

PasswordFiled FormControl

Address -🡪 FormGroup

City FormControl

State FormControl

PinCode FormControl

formGroup and formControlName are attribute part of ReactiveFormsModule. So we have to import this module in app.module.ts file.

**Angular Validation**

To do validation Angular provided pre-defined classes

ng-valid, ng-invalid, ng-touched, ng-untouched, ng-pristine, ng-dirty etc

ng-valid : it will become true if all condition in forms a satisfied.

ng-invalid : reverse of ng-valid.

By Default every html textfield, passwordfield,radio button etc. Default properties are enable.

Ng-untouched : didn’t touch the textfield. Revers of ng-untouched is ng-touched.

Ng-pristine : didn’t change any value. reverse of ng-pristine is ng-dirty .

Ng-valid : true

Profile component

Template Driven Form Model Driven Form

Profile Details

First Name

Last Name

Gender Radio Button

Hobbies Checkbox

City drop down box

SkillSet TextArea Validation

Submit Reset

Your Details is as

**Day 10- 09-04-2021**

Angular Service

Create new project

If we write any logic in component that code become local to that template.

If we need common logic for more than one component in place writing in every component , make separate class and inside that class we write more than one function. In Every component we will create the object of service class and call that methods.

Angular service mainly use to achieve separation on concern.

Angular service divided into two types

1. User-defined service
   1. Creating object of that class using new keyword.
   2. Creating object of that class using DI concept (Dependency Injection).
2. Pre-defined service HttpClient.

ng new angular-service-with-pipe

ng g c first

ng g c second

IOC : Inversion of control : it is a design pattern or programming. In place of creating object or resource explicitly allow to create by container(Angular container). If container create it will maintain the life of the object or resource. Pull from container whenever you want.

It is a concept.

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

3 types

1. Constructor base DI
2. Setter base DI
3. Interface DI

Angular support constructor base DI.

fake-component

ng g c fake

we have to make user define class with decorator @Injectable ie AngularService class.

We have to register service class in module level or component level using property provider

1. ngModule level provided the details about service class.

If we register service class in module level using provider attribute. Then angular create only one object and same object shared in more than one component. (singleton design pattern).

1. Component level provided the details about service class.

If we register service class in component level using provider attribute. Then angular create different object on component level. That object become local to that components.

http service (pre-defined service)

Angular Provide Pre-defined API HttpClient which help to call backend technologies (REST API – RestFull Web Service).

Java or Spring boot Angular

Asp.net React JS

Python Rest API Vue JS

PHP JSON

To do CRUD Operation (Insert, Delete, Update and Retrieve records).

In Angular using HttpClient we can call any REST API develop in any technologies.

If we want to use HttpClient service in angular we have to do DI for HttpClient in Angular service or component.

HttpClient API is a part of HttpClientModule. So we have to import this module in app.module.ts file

According to REST API

Get method : to Get the Resources like employee details, customer details, manager details.

http.get(“url”)

This method return type is Observable object.

1. Using Promise : return all data at time. Using then() and catch() we can resolve or reject promise.

Promise can’t cancel

1. Observable : Observable is a part rxjs (Reactive JS). Angular use third party plugin to call Rest API using HttpClient. So to load the data we have to call subscribe(). This method take 3 parameter as a callback

1st parameter : next : to load the data one by one

2nd parameter : error : if any error generated at the beginning or middle or at last.

3rd parameter : completed : after successfully loaded all data it call third parameter to give the confirmation.

1st parameter mandatory. 2nd and 3rd are optional parameter.

Observable can cancel using unsubscribe()

To map the backend json data we have to create model class or interface.

using json-server module part of node we can make static json file as a server.

Create employee component

Create employee service

Create employee model

And display in table format

URL for json service : [localhost:3000/employees](http://localhost:3000/employees)

http.post(“url”,jsonData).subscribe(result,error,completed());

pipe (filter) : pipe is use to do transformation of data on template.

**Pre-defined pipe**

uppercase

lowercase

currency

date

json

etc

{{variableName | pipeName }}

**Custom pipe**

create normal class and Implements Transformer interface and provide the body for transform

On that class write decorator as @Pipe with name for the pipe.

Custom pipe details we have to provide declaration section in app.module.ts file.

Angular with bootstrap

**Angular materialization**

ng new angular-material

Button styling,

Form related styling

Table related styling

Etc

We have to that modules in app.module.ts

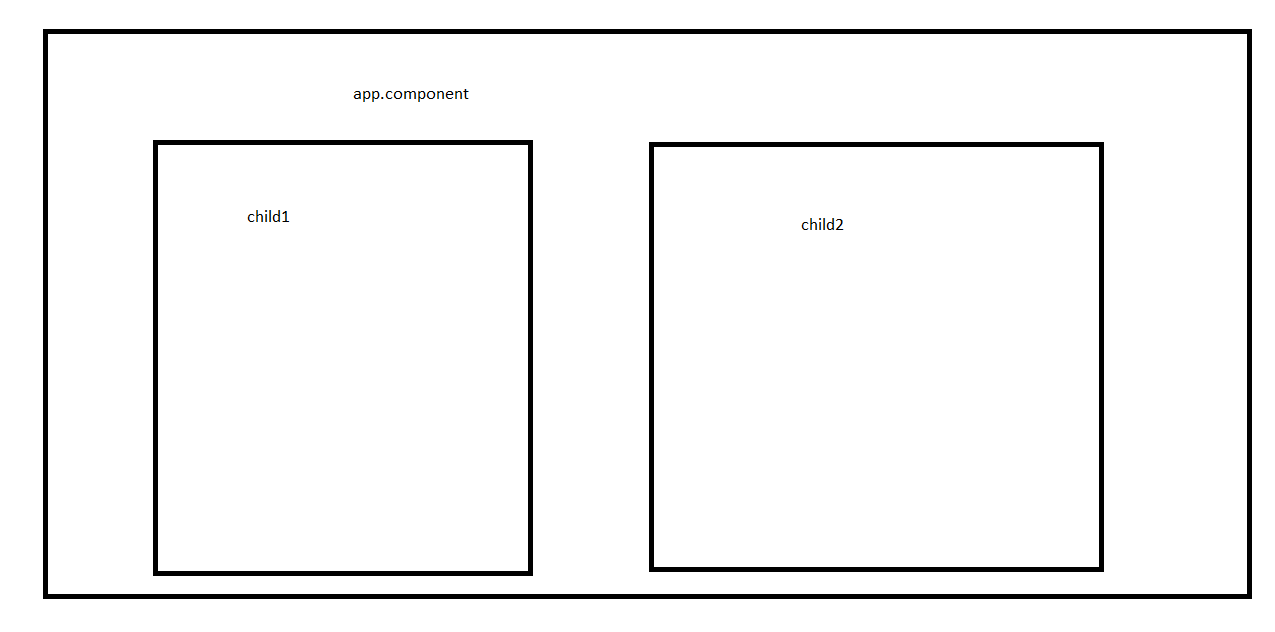
All modules start with prefix Mat-TableTagName

angular-component-communication

angular-routing : while creating routing option yes.

10-04-2021

Component communication



App.component : parent

Child1 : child for app component

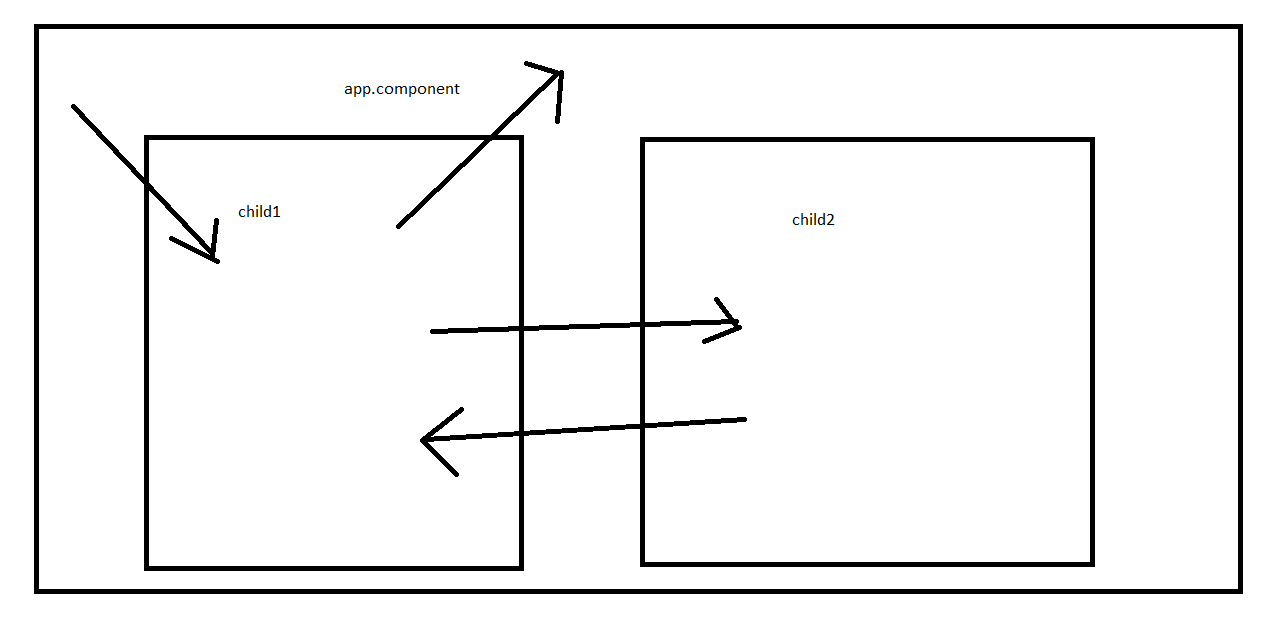
Child2 : child for app component

Child1 and child 2 are sibling

Every component variable or property can access within that component template.

Property may be simple (primitive property), array property, user defined class object or array of user-defined class objects.

We can access within that component or that’ component template.



ng new component-communication

then create two component

ng g c child1

ng g c child2

1. Parent – child -🡪 Using @Input decorator we can pass the value from parent to child component.
2. Child – parent 🡪
3. Using @Output decorator with the help of EventEmitter API part of Angular.(child component we have use the decorator). Child component pass the value to parent component using event.
4. Using @ViewChild decorator with help of inject (Parent component we have to use the decorator). Parent component access the property or behaviour of child component using child component reference.

Sibling :

1. Using localStorage or sessionStorage. (JavaScript and HTML5 Features).
2. Using common shared service. (One component set the value and another component get the value).
3. Using rxjs (third party API).

child3 and child4

creating service using command prompt

Angular Routing

SPA:

Angular routing is use to navigate from one template to another template using path provided by router module.

Aboutus ContactUs Login Success Dashboard

Sub1 Sub2 Sub3

Create the 3 component

ng g c aboutus

ng g c contactus

ng g c login

In Routing angular provide predefined tag <router-outlet></router-outlet>. It is a place holder which help to load the content of component’s template depending upon the path provided by routing file.

login.json

{“login”:[

{“user”,”Ravi”,”pass”:”123”},

{“user”,”Ramesh”,”pass”:”567”}

]

}

Check username and password from json file using json-server node module with the help of HttpClient get method and insider subscribe(next) if valid navigate to success page or error.

Auth Guard : Auth are interfaces which restrict the user not to move any page directly.

When client do the authentication from backend technologies. So the backend technologies generate one unique id sessionId is or JWT (Json web token).

In Angular side we have to store that id in sessionStorage till the user click on logout button.

Req

Client Server

Res + unique id (sessionId + JWT)

Life cycle methods

Interfaces.

Create

onInit interface, ngOnInit() method

afterContentInit ngAfterContentInit() method

afterViewInit ngAfterViewInit() method

Changes

doChange

onChanges

AfterContentChecked

AfterViewCheked

Destroy

onDestroy

Angular 1.x base upon MVC

Angular 2 to 11 : component base architecture.

MVC : Model View Controller

View -🡪 html / template

C -🡪 Controller replace by Component

Model -🡪 model (typescript and services)

Angular Developer

Iconic :

React JS

Reactive Native