BB-3(FULL STACK)

HTML PENDING

**CSS**

* Cascading style sheets
* Defines how to display HTML elements

**Uses of CSS**

* CSS saves lots of work
* Styles are generally saved in external .css files,external style sheets enable you to change the appearance and the layout of all pages in a web site , just by editing one single file.

**Styling rules**

* Style rule is composed of 2 parts —> 1. Selector 2. Declaration
* The **selector** indicates the element to which the rule needs to be applies
* The **declaration** determines the property value of selector
* **Property:** which specifies the characteristics such as color,font-family,position and is followed by a colon(:)
* **Value:** which expresses the specification of property such as red for color,arial for font family,12p for font size , and is followed by semicolon (;)

**P { color : red; }**

**Internal styling**

Written in head tag , used to style for multiple tags

It reduces the work in such a way that if we change the styling once it can be applicable for all that particular tags

Ex:

<head>

<style>

p{ /\* internal styling \*/

color:#66CCFF;

font-size:24px;

font-style:oblique;

font-weight: bold;

}

</style>

</head>

**Inline styling**

Styles separately for each tag

Ex:

<p style="color:#6666FF; /\*inline styling \*/

font-size:24px;

font-style:italic;

font-weight:lighter;">

CSS is the language we use to style an HTML document.

CSS describes how HTML elements should be displayed.

This tutorial will teach you CSS from basic to advanced.

</p>

**Id selector**

* Id selector is used to specify a style for single , unique elements.
* Id selector uses id attribute of the html elements and is defined as “#”
* The style rule below will be appeared to the element with id=”para1;”

Ex:

#para1 {

text-align: center;

color: red;

}

<p id=”para1”>Hllo world </p>

**Class Selector**

* Class selector is used to specify a style for a group of elements.unlike the id selector,the class selector is most often used on several elements
* This allows to to set a particula style for any html elements with the same class.
* Class selector uses html class attribut and defined as “ . “

Ex:

.center {

text-align: center;

color: red;

}

<h1 class=”center”> welcome baby </h1>

Here all the P elements with class=”center” will be aligned to center and color will be red

**External styling**

**files used —> about.html , styles.css**

* When we wants to apply the same styling for so many pages at that time we cant change manually but using inline or internal stylings
* Here we simply create one styles.css file and we will link that styling file to remaining pages for by using the below syntax : **<link rel="stylesheet" href="path of style.css"/>**

**Ex:**

**about.html**

<!DOCTYPE html>

<html>

<head>

<title>about page</title>

<link rel="stylesheet" href="css/styles.css"/>

</head>

<body>

<h1>

hiiii !!!!!!

</h1>

</body>

</html>

**Style.css**

h1{

color:#CC33CC;

}

**MARGIN AND PADDING**

**Similar concepts**

**Block elements**

**Inline elements**

**—--------------------------**

**Class**

**Id**

**—--------------------------**

**Margin**

**Padding**

**—--------------------------**

**Padding**

The CSS padding properties are used to generate space around an element's content, inside of any defined borders.

* padding-top
* padding-right
* padding-bottom
* padding-left

Ex:

**about.html**

<div class="custom"> <!-- padding -->

<img src="images/r.jpg">

</div>

**Styles.css**

.custom{

background-color: #9999CC;

width:800px;

height:800px;

padding-top:30px;

padding-left:30px;

}

**Margin**

The CSS margin properties are used to create space around elements, outside of any defined borders.

* margin-top
* margin-right
* margin-bottom
* margin-left

**About.html**

<span background-color:#CC6699">

<a href="home.html">home</a> <!-- margin tag -->

<a href="services.html">services</a>

<a href="careers.html">career</a>

<a href="contact.html">contact</a>

<a href="about.html">about</a>

</span>

**Styles.html**

a{

text-decoration:none;

color:#006600;

font-weight:bold;

font-size:18px;

margin-right:55px;

}

**Javascript (day4)**

* Js code can be kept in <head> or <body> of a HTML document
  + Javascript function should be defined in the <head>
  + This ensures that the function is loaded before it is needed
  + Javascript in the <body> will be executed as the page loads
* Javascript can be in separate js file , js code can be written in script tag
* <script src=”myjsfile.js”></script>
* Put this html whenever you would put the actual js code
* As external js file will let you use the same js on multip;e html pages
* An external .js file cannot itself contain a <script> tag

**VARIABLES:**

* Variables are declared with **var** keyword.
* No need to give the datatypes while declaring

Ex:

<script>

var a=10;

var b=”hello”

document.write(a);

document.write(a);

</script>

* Variable name should begin with letter or underscore.
* Variable names are casesensitive.
* Variables are untyped (can hold values of any type)
* The word var is optional (but its good to use it)
* Variables declared within the function are local to that function (accessible only within the function).
* // single line **comment**
* /\* —------ \*/ multi line comment

**OPERATORS,I**

Because most javascript code is borrowed from C (and is therefore like java) we wont spend much time on it.

* **Arithmetic Operator**
* **- \* / % ++ - -**
* **Comparison Operator**

**< <= > >= == !=**

* **Logical**

**&& || !**

* **Bitwise**

**& | ^ ~ << >> >>>**

* **Assignment**

**+= -= \*= /= %= <<= >>= >>>= &= ^= |=**

**OPERATORS,II**

* **String operator**

**+**

* **Conditional operator**

**Condition ? value if true : value if false**

* **Special equality test**

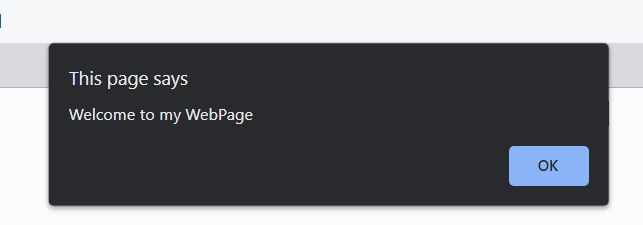
**== and != try to convert their operands to the same type before performing the test**

**=== and !== consider their operads *unequal* if they are of different type**

* **Additional operators to be discussed**

**new void typeof delete**

**FUNCTIONS**



Function hello(){

alert (“welcome to my page”)

}

hello();

**Function with parameters**

function myname(x){

document.write("<br>Welcome" +" " +x)

}

myname("Amala...")

or

document.write(`<br>Welcome ${x}`)

Both the above will get the same output as Welcome Amala…

**function add(x,y){**

**document.write(`<br>Addition of 2 numbers is ${x+y}`)**

**}**

**function mul(x,y){**

**document.write("<br>Multiplication of 2 numbers is "+x\*y)**

**}**

**function calc(){**

**add(2,3);**

**mul(3,4);**

**}**

**calc();**

**CONDITIONAL STATEMENTS**

**var u=30**

**var v=20**

**if(u>v){**

**document.write(`<br>A is greater tha B i.e., ${u}`)**

**}**

**else{**

**document.write(`<br>B is greater tha A i.e., ${v}`)**

**}**

**LOOPS**

**//loops**

**//for loop**

**var i;**

**for(i=0;i<=10;i++){**

**document.write(i+"<br>")**

**}**

**//while loop**

**var x=1**

**while(x<=5){**

**document.write(x+"<br>")**

**x++;**

**}**

**//do-while loop**

**var i=2;**

**document.write("do-while loop");**

**do{**

**document.write("<br>"+i+"<br>")**

**i++;**

**}while(i>4);**

**VALIDATOR FUNCTION EXAMPLE (form2.html)**

**Example1: successful registered/not with checkbox**

**<body>**

**<form name="first">**

**<input type="checkbox" name="terms">Read Terms and Conditions**

**<input type="button" value="submit" onclick="validator();">**

**</form>**

**<script>**

**function validator(){**

**if(document.first.terms.checked){**

**alert("Registered Sucessfully")**

**}**

**else{**

**alert("Please check the terms & conditions properly you stupid")**

**}**

**}**

**</script>**

**</body>**

**Example2:checking name and password**

**<body>**

**<form name="myform" onsubmit="return validateform()">**

**Name : <input type="text" name="name"><br><br>**

**Pass : <input type="password" name="password"><br><br>**

**<input type="submit" value="register">**

**</form>**

**<script>**

**function validateform(){**

**var name=document.myform.name.value;**

**var password=document.myform.password.value;**

**if(name==""){**

**alert("Fill the Name Properly")**

**return false;**

**}**

**if(password.length<8){**

**alert("Password should be at least 8 characters")**

**return false;**

**}**

**if(name!="" && password.length>8){**

**alert("registered successfully")**

**}**

**}**

**</script>**

**</body>**

**MOUSEOVER - MOUSEOUT**

**<input type="button" value="Mouse out" onmouseout="alert('Mouse out')">&nbsp;&nbsp;**

**<input type="button" value="handover" onmouseover="alert('Mouse over')">**

**ARRAY**

**Can be created in 4 ways**

1. **Array literal** : var colors=[“red” , “green” , “blue”]
2. **new Array()** : To create an empty array→ var colors=new Array();
   1. Can add elements to array later→ colors[0]=”red”;colors[1]=”green”;colors[2]=”blue”
3. **new Array(n)** : can be used with a single numerical argument to create as array od tha size → var colors=new Array(3);
4. **new Array(...)** : with two / more arguments to create an array containing those values.

**Bootstrap classes**

**DAY - 5**

**RESPONSIVE WEBPAGE**

**WHAT IS BOOTSTRAP**

**BOOTSTRAP HISTORY**

* Bootstrap was developed by Mark Otto and Jacob Thornton at Twitter,and released as an open source product in August 2011 on GitHub.
* Advantages of Bootstrap
  + Easy to use : basic knowledge of HTML and CSS can work with bootstrap.
  + Responsive Features : Bootstraps responsive adjusts to phones,tablets and desktops.
  + Mobile-first approach : in Bootstarp 3 , mobile-first styles are part of core framework.
  + Browser compatibility : Bootstrap is compatible with all modern browsers ( Chrome ,Firefox…… )

**D-6**

**New html elements in html5:**

<article> <figcaption> <progress>

<aside> <footer> <section>

<audio> <header> <source>

<canvas> <hgroup> <svg>

<datalist> <mark> <time>

<figure> <nav> <video>

These are just some of the new elements introduced in HTML5.

Features in HTML5:

→built in audio and video support (without plugins)

→ Enhanced form controls and attributes.

→ The Canvas ( a way to draw directly on a web page)

→ Drag and drop functionality.

→ Support for css3

→ more advanced features for developers , such as data storage and offline applications.

<form>is just another kind of HTML tag

◼ HTML forms are used to create (rather primitive) GUIs on Web pages

◼ Usually the purpose is to ask the user for information

◼ The information is then sent back to the server

◼ A form is an area that can contain form elements

◼ The syntax is: ...form elements...

◼ Form elements include: buttons, checkboxes, text fields, radio buttons, drop-down menus, etc

◼ Other kinds of HTML tags can be mixed in with the form elements

◼ A form usually contains a Submit button to send the information in he form elements to the server

◼ The form’s parameters tell JavaScript how to send the information to the server (there are two different ways it could be sent)

◼ Forms can be used for other things, such as a GUI for simple programs

**FORM TAG**

**INPUT TAG**

**Forms examples**

**<form>**

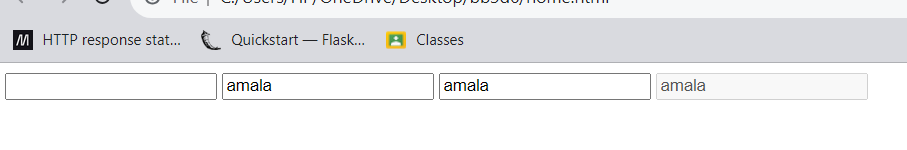
**<input type="text" name="fname">**

**<input type="text" name="fname" value="amala">**

**<input type="text" name="fname" value="amala" readonly>**

**<input type="text" name="fname" value="amala" readonly disabled>**

**</form>**

****

**Textarea**

**<textarea> </textarea>**

**<textarea rows="20" cols="10"> </textarea>**

**<textarea rows="10" cols="30" readonly>welcome dear.....</textarea>**

**<textarea rows="10" cols="30" readonly disabled>welcome dear.....</textarea>**

**</form>**

**</body>**

**Button**

* **Submit : sends data**
* **Reset : reset all form elements to inintail state**
* **Pushme : takes action as specified by javascript**

**FORM WITH BOOTSTRAP**

**<form>**

**<div class="form-group">**

**<label>username</label>**

**<input type="text" class="form-control">**

**</div>**

**<div class="form-group">**

**<label>password</label>**

**<input type="password" class="form-control">**

**</div>**

**</form>**

****

**<div style="width: 25%;">**

**<form>**

**<div class="form-group">**

**<label>SELECT STATE : </label>**

**<select class="form-control" multiple> <!--to select multiple options we used multiple-->**

**<option> Andhra pradesh </option>**

**<option> Telangana </option>**

**<option> KArnataka </option>**

**<option> Chennai </option>**

**</select>**

**</div><br>**

**</form>**

**</div><br>**

**<div style="width: 25%;">**

**<form>**

**<div class="form-group">**

**<label>SELECT STATE : </label>**

**<select class="form-control form-control-sm"> <!--small size-->**

**<option> Andhra pradesh </option>**

**<option> Telangana </option>**

**<option> KArnataka </option>**

**<option> Chennai </option>**

**</select>**

**</div><br>**

**</form>**

**</div><br>**

**<div style="width: 25%;">**

**<form>**

**<div class="form-group">**

**<label>SELECT STATE : </label>**

**<select class="form-control"> <!-- normal size -->**

**<option> Andhra pradesh </option>**

**<option> Telangana </option>**

**<option> KArnataka </option>**

**<option> Chennai </option>**

**</select>**

**</div><br>**

**</form>**

**</div><br>**

**<div style="width: 25%;">**

**<form>**

**<div class="form-group">**

**<label>SELECT STATE : </label>**

**<select class="form-control form-control-lg"> <!-- large size -->**

**<option> Andhra pradesh </option>**

**<option> Telangana </option>**

**<option> KArnataka </option>**

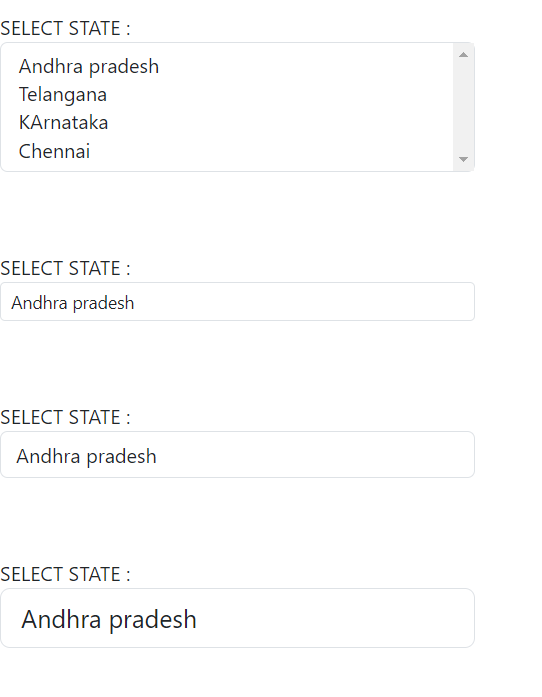
**<option> Chennai </option>**

**</select>**

**</div><br>**

**</form>**

**</div><br>**

****

**PROGRESS BAR**

**HTML & CSS**

**<div class="progress">**

**<div class="progress-bar" role="progressbar"**

**style="width: 25%;" aria-valuemax="100" aria-valuemin="0" aria-valuenow="25">**

**</div>**

**</div><br>**

**JS**

**<div class="progress">**

**<div class="progress-bar" role="progressbar"**

**style="width: 50%;" aria-valuemax="100" aria-valuemin="0" aria-valuenow="50">**

**</div>**

**</div><br>**

**ANGULAR**

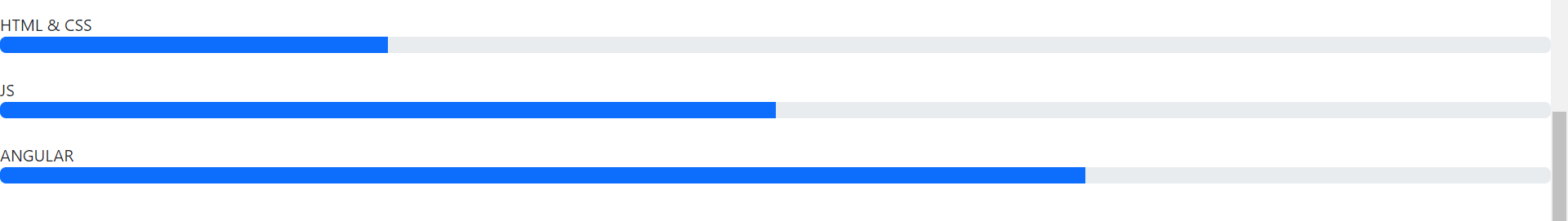
**<div class="progress">**

**<div class="progress-bar" role="progressbar"**

**style="width: 70%;" aria-valuemax="100" aria-valuemin="0" aria-valuenow="70">**

**</div>**

**</div><br>**

****

**JUMBOTRON**

**<div class="row" style="padding: 10px 60px;"> <!-- jumbotron -->**

**<div class="col-lg-12">**

**<div class="jumbotron">**

**<h1 class="display-4">hello, world</h1>**

**<p class="lead">**

**somelines of code**

**</p>**

**<p>somelines of code</p>**

**<p class="lead">**

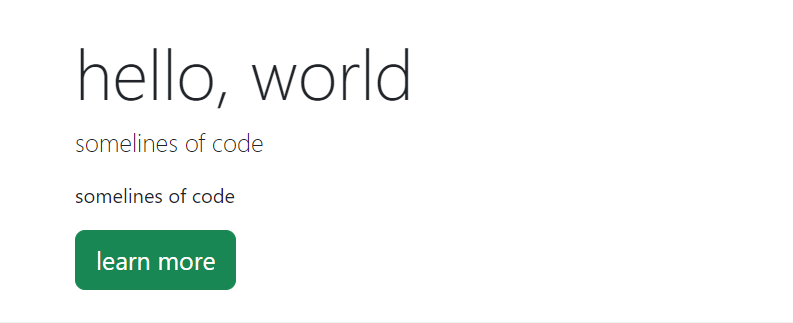
**<a class="btn btn-success btn-lg" href="#" role="button">learn more</a>**

**</p>**

**</div>**

**</div>**

**</div>**

****

**CAROUSEL**

**<div id="carousel" class="carousel slide"> <!--carousel -->**

**<div class="carousel-inner">**

**<div class="carousel-item active">**

**<img src="images/picture.jpg" class="d-block w-100">**

**</div>**

**<div class="carousel-item">**

**<img src="images/plant.jpg" class="d-block w-100">**

**</div>**

**<div class="carousel-item">**

**<img src="images/bbicon.jpg" class="d-block w-100">**

**</div>**

**</div>**

**<button class="carousel-control-prev"**

**type="button" data-bs-target="#carousel"**

**data-bs-slide="prev">**

**<span class="carousel-control-prev-icon"**

**aria-hidden="true"></span>**

**<span class="visually-hidden">Previous</span>**

**</button>**

**<button class="carousel-control-next"**

**type="button" data-bs-target="#carousel"**

**data-bs-slide="next">**

**<span class="carousel-control-next-icon"></span>**

**<span class="visually-hidden">Next</span>**

**</button>**

**</div>**

****

**Here we will be getting the arrows(left and right ) to slide the images in the webpage**

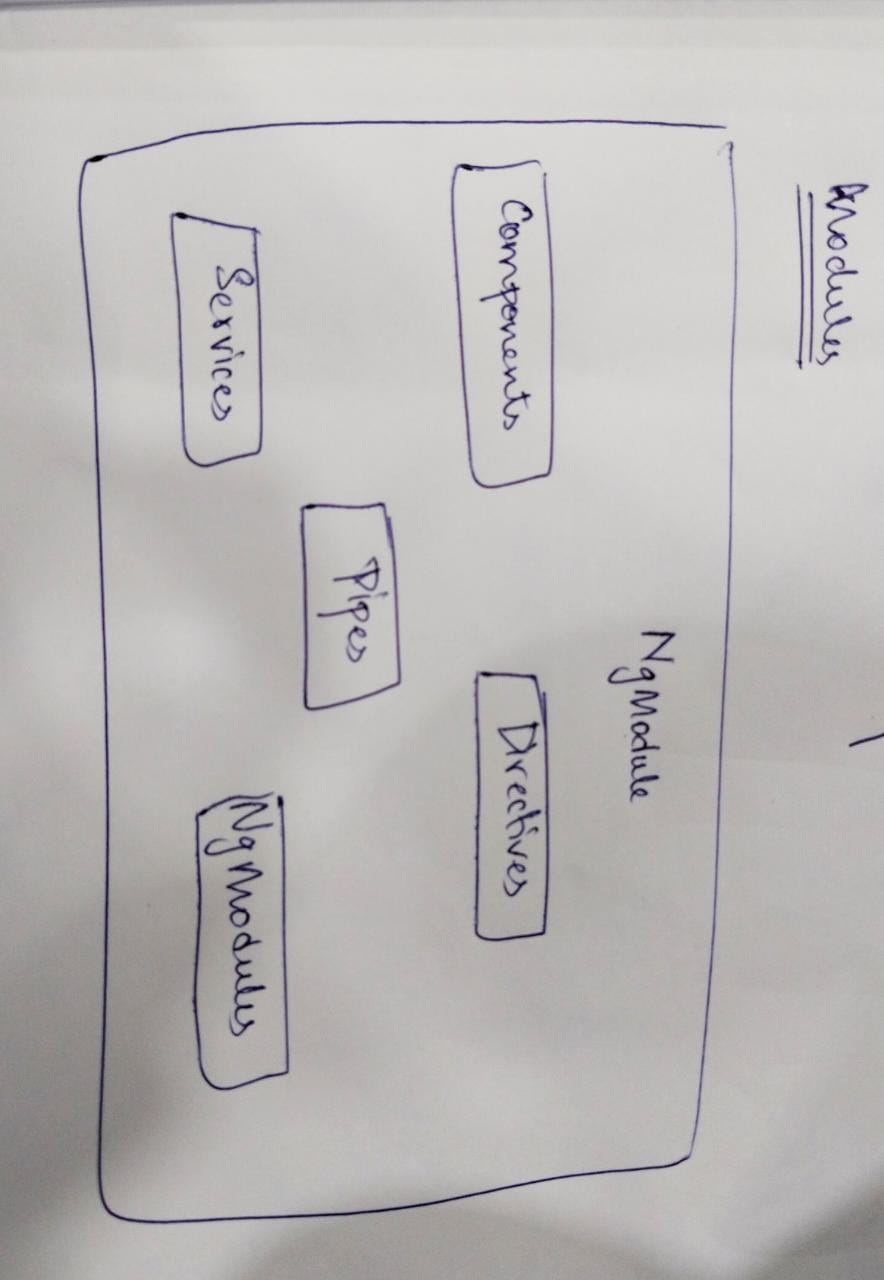
**ANGULAR 14-8-23**

**Introduction to angular**

* A frontend framework for building applications using web technologies like html,css and js
* Empowers developers to build applications for browsers,mobiles or desktop
* With angular we can develop mobile applications and desktop applications also not only for webpages.

**Building blocks of Angular**

* Modules
* Components
* Templates
* Metadata
* Data binding
* Directives
* Pipes
* Routing
* Forms
* Services
* Dependency injection



**To work with angular**

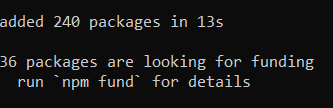
**Following should be installed**

* Node js –to compile code
* Angular cli – to create angular project

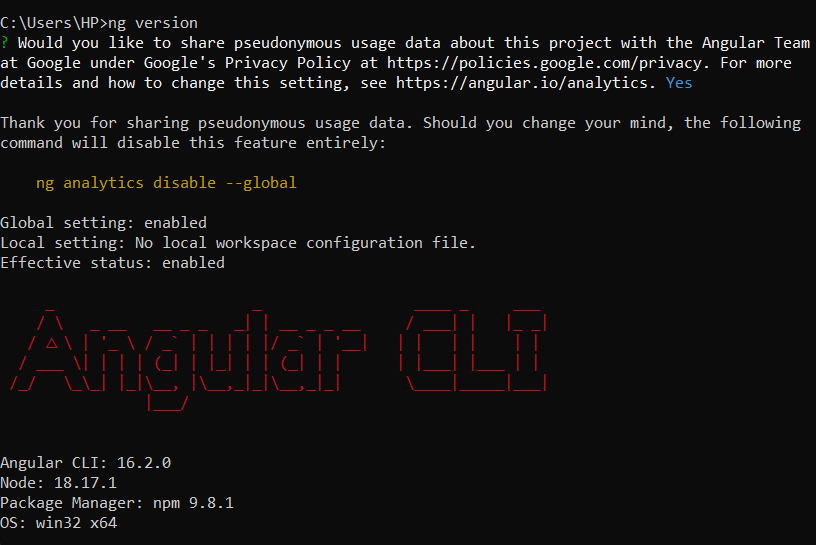
Command to install angularcli



Successful installation



To check the version of angular(ng version)



* Vs code –editor

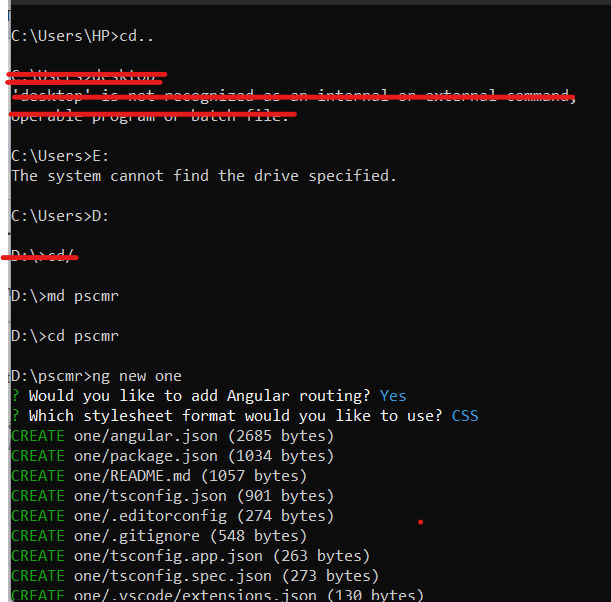
**To create an angular application**

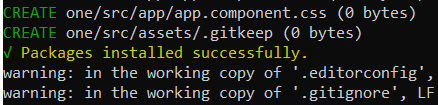
cd – change directory

cd.. – to come out directory

md – make directory

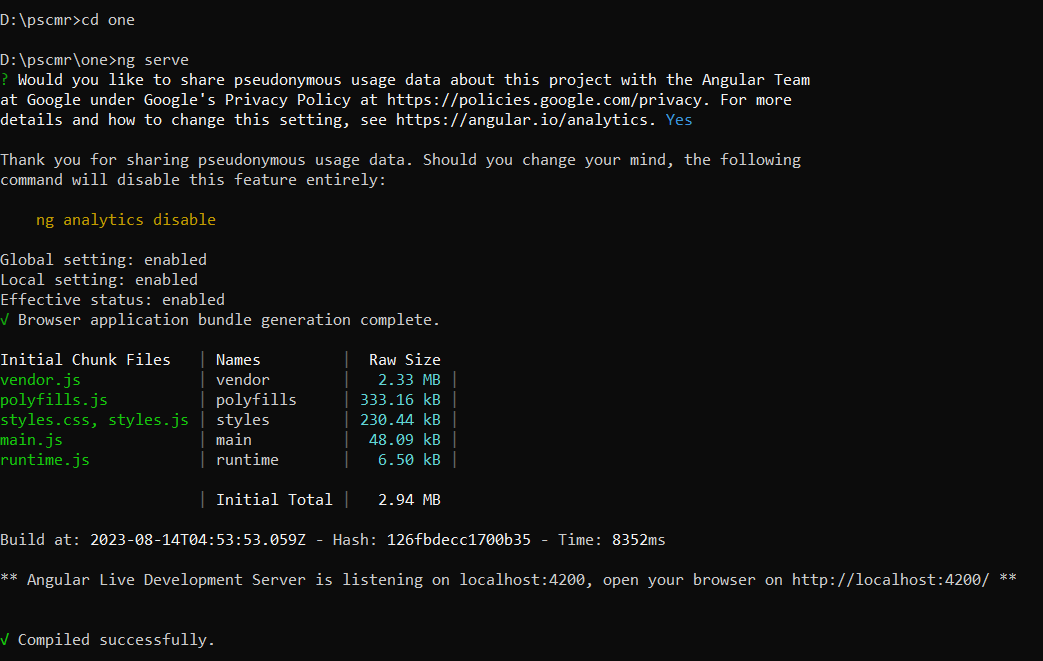
**ng new filename** – to create angular application

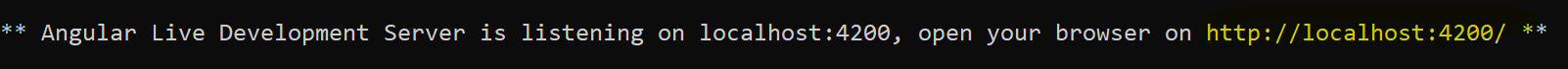




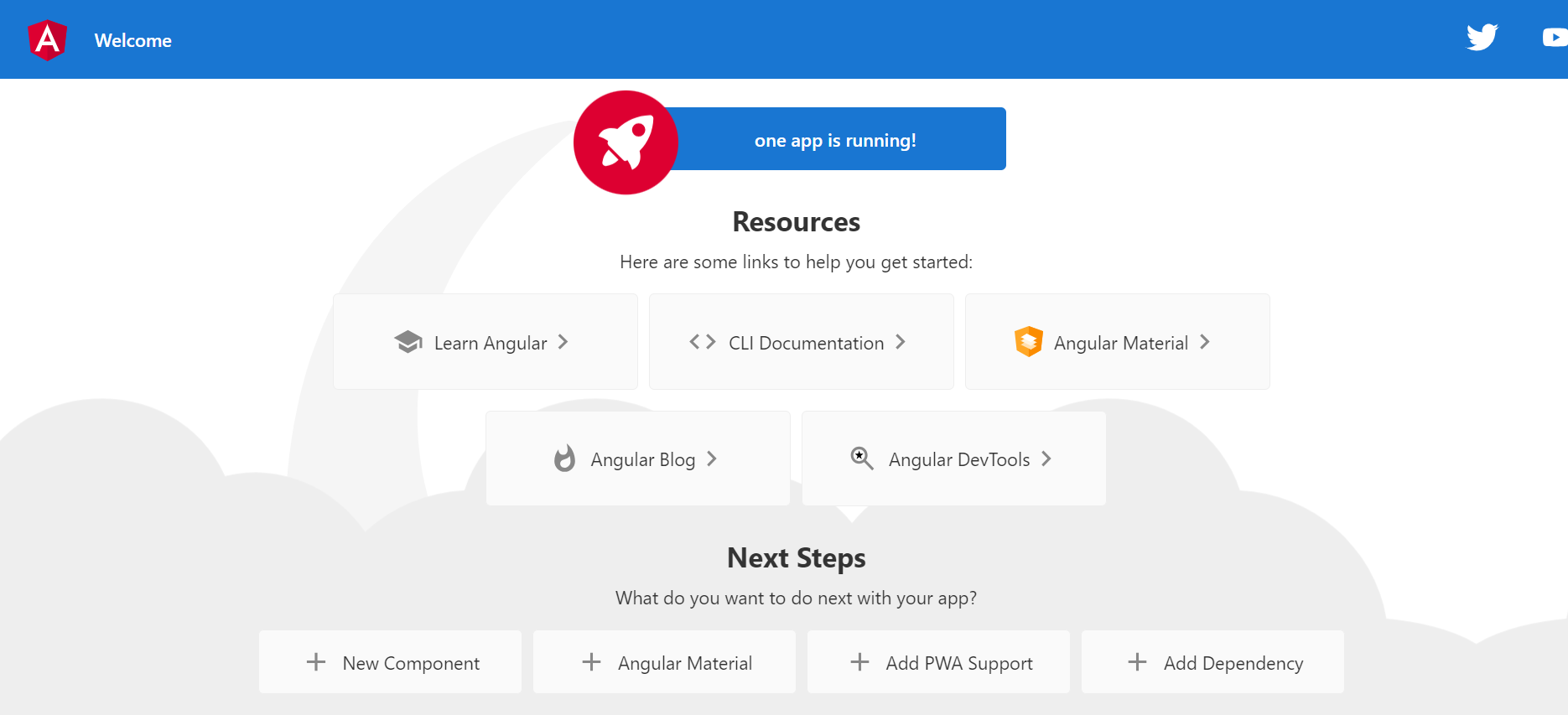
**To run the project (ng serve)**

Go to project directory and run ng serve

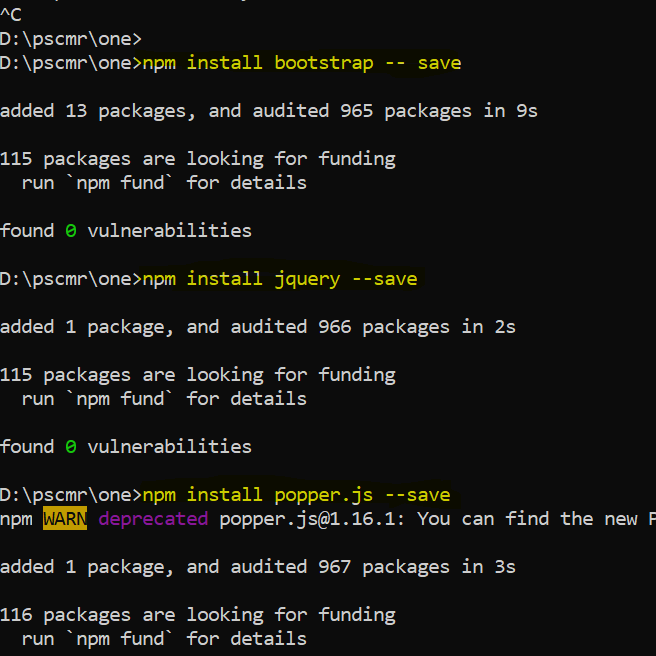
****

**Run the above yellow marked url in the browser (localhost:4200)**

**output:**

****

* **To work with internal styling refer to app.component.css**
* **To work with external styling refer to styles.css**
* **There are 2 ways to work with bootstrap**
  + By copying the links from bootstrap website
  + By installing the bootstrap
    - Some of the commands to install bootstrap are
    - npm install bootstrap -- save
    - npm install jquery --save
    - pm install popper.js --save

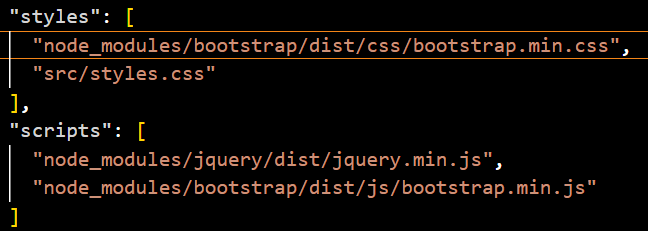


Here after installing the bootstrap we need to copy some styles and scripts in angular.jon file to work with bootstrap

After including those we need to serve again

I.e..,whenever we made any changes in the angular.json file we need to serve again

**These below codes should be placed in angular.json file under build**



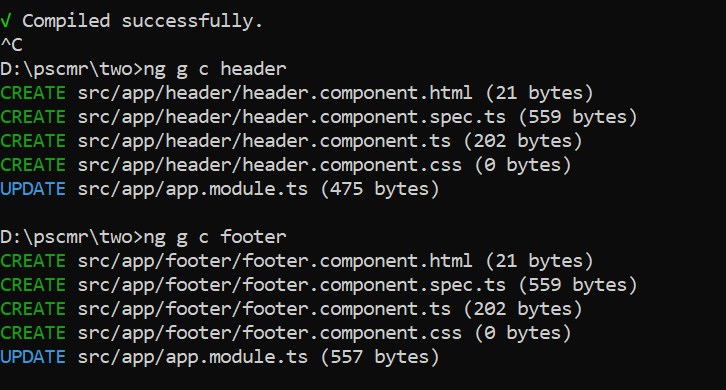
Steps to follow for every project

1. Ng new projectname
2. Cd projectname
3. Ng serve
4. Copy link i.e.., localhost:4200 in browser
5. Go to cmd again and install bootstrap by giving all the 3 commands.
6. Open folder in vs code
7. Go to app.component.html and make the changes as u wish
8. Now go to angular.json and insert the code snippets of bootstrap in style and script in build part
9. Run the ng serve again
10. Check the result in browser.

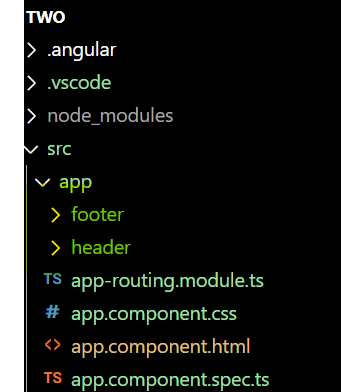
In Angular we will create separate files for header and footer and content.So to create those separate files for header and footer we need to create a components for header and footer

**To create a components**

* ng g c componentname
  + Ex: ng g c header
  + ng g c footer
  + ng serve

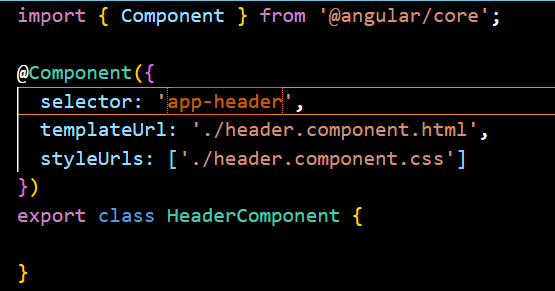


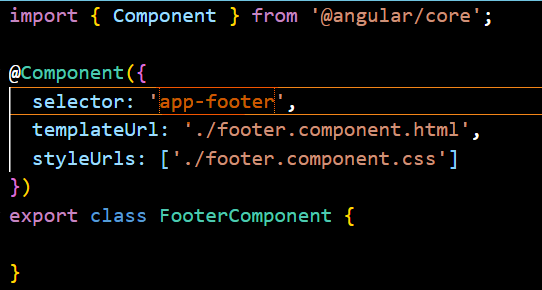
After the components got created those components will be visible in the **app in vs code**

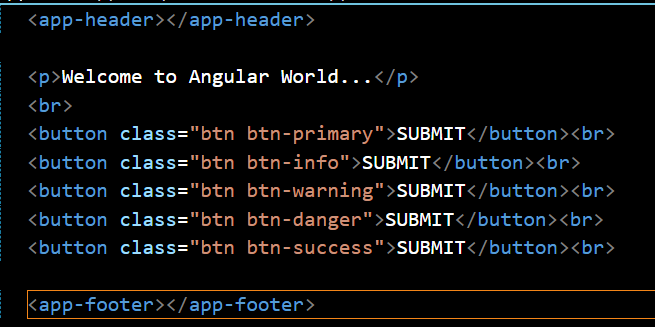


Now now we need to give those header and footer tags in app.component.html

Those tags will be available in header.component.ts and footer.component.ts





Now we need to give those tags in app.components.html

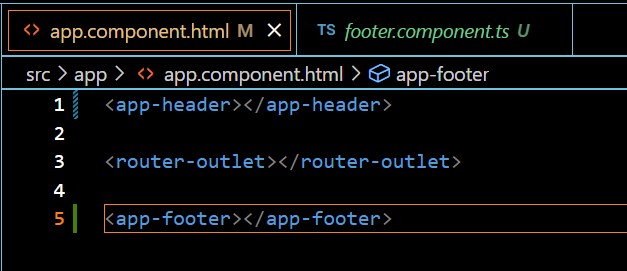
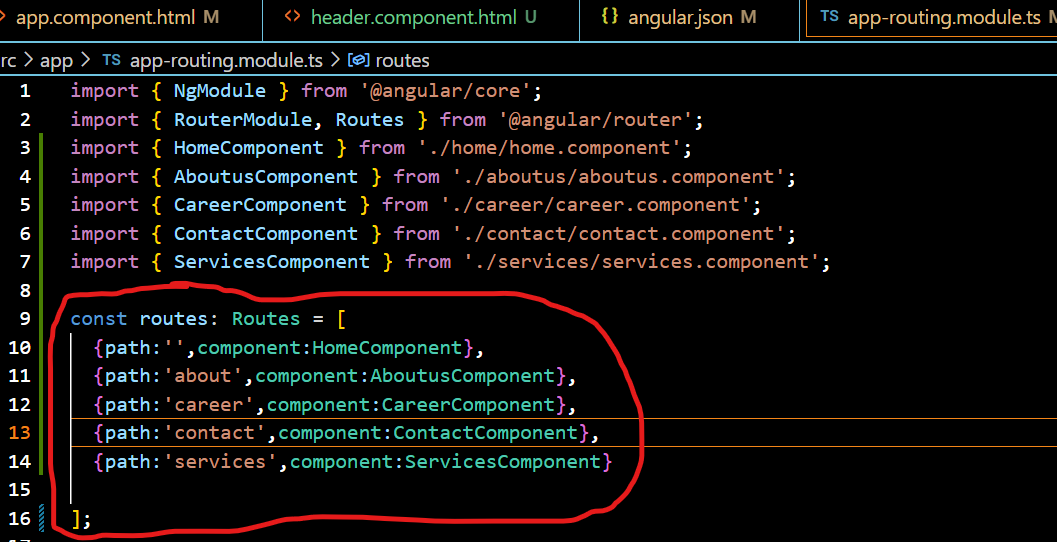
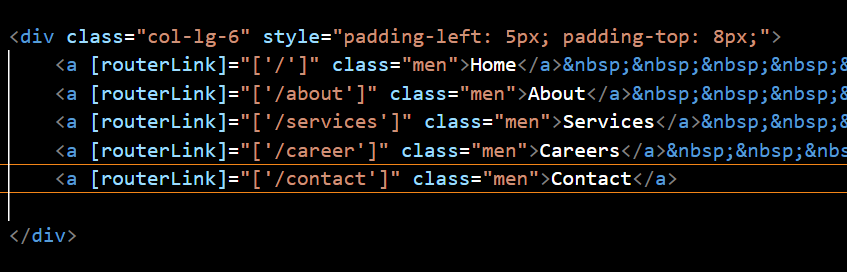
**Working with images in Angular**

* All the media files i.e.., images , audios, videos etc.. should be saved in assets file which is present in src

**Router-outlet**

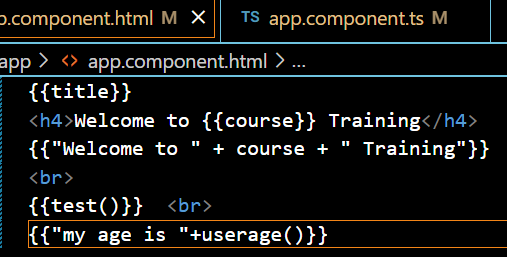
Router-outlet is used to create a single page application

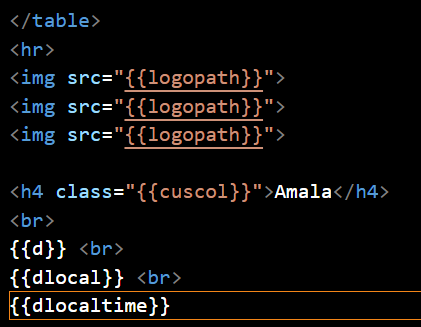
To define patjhs go to app-routing modules.ts

1. In app.component.html ,initially keep router-outlet
2. 
3. In app-modules.ts give the paths
4. 
5. 
6. This should be in header.component.html in palace of <a href >we need to give [routerLink]

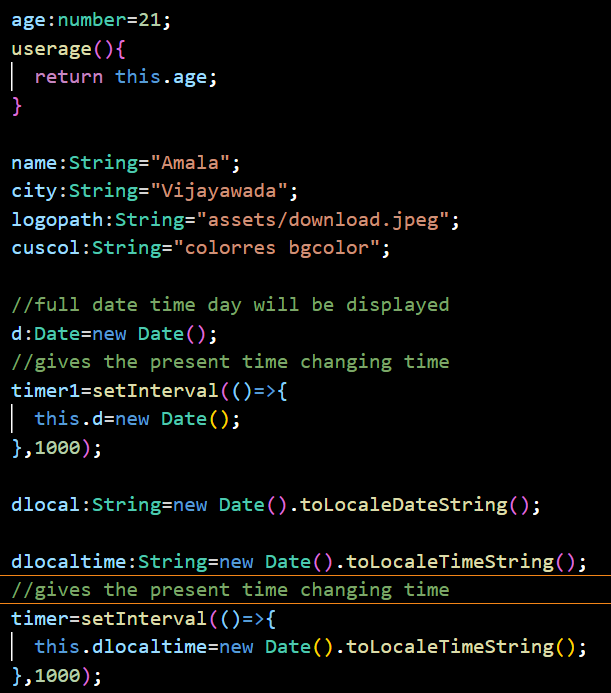
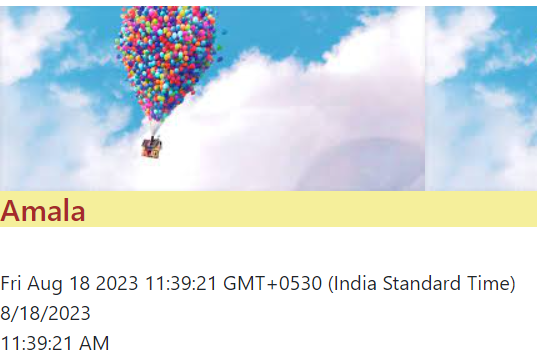
**WORKING WITH TS FILES(18/08/23)**

* To pass any data to view(.html) we need a model that model is .ts file i.e.., we are binding the data from model to view
* **DataBinding can be done in 2 ways**
  + **One way data binding** (binding data from model-view or view-model)
    - String Interpolation (allows to bind data from **ts file(mode) to html**
      * Syntax : {{expression}}
      * app.component.html



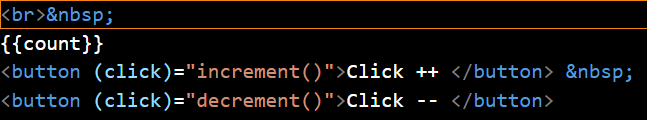
* 
* App.component.ts file



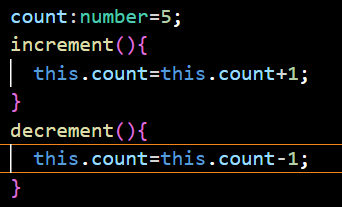
* 
* Output
* 

* + - Attribute binding
    - Class binding
    - Style binding
    - Event binding
      * It is one way binding
      * Allows to bind data from **html to model**(ts file)
      * Mouse events
        + Click

App.component.html

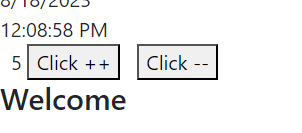


App.component.ts



**ngIf** - returns the expression only if it is true ,else if it is false doesn't return any expression (hide or remove element)





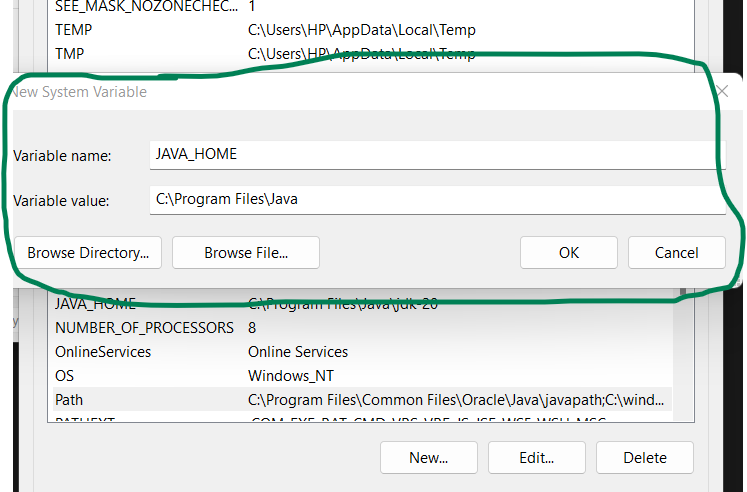
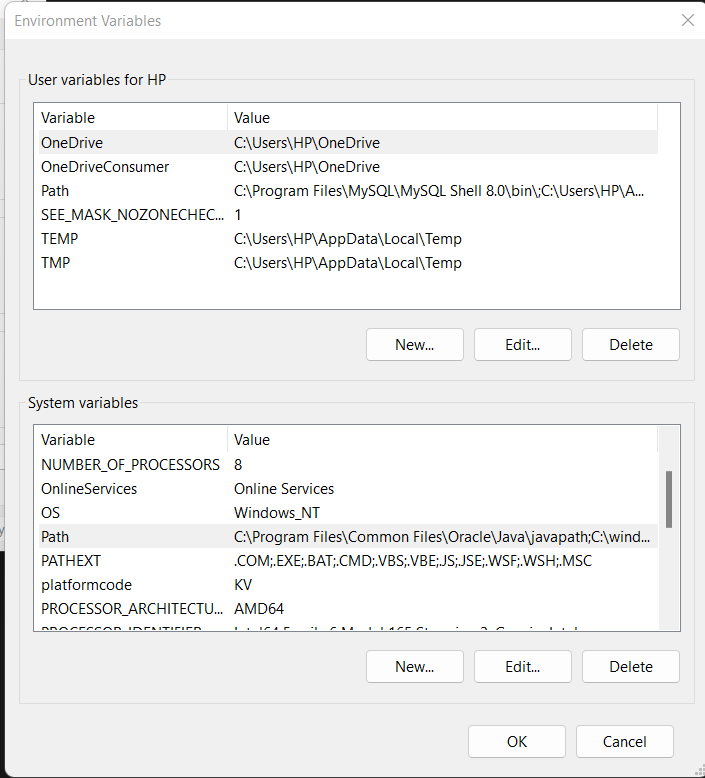
* + - * + double click
        + mouseover
        + mouseout
        + mouse move
      * Keyboard events
        + key down
        + key up
        + key press
      * Focus events
        + focus
        + blur
        + focusin
        + focusout

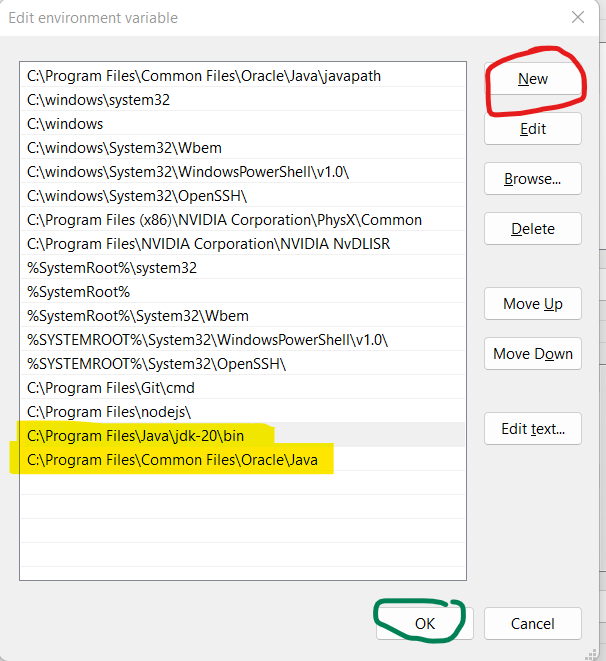
* + **Two way data binding** (1st tha data is binded from model to view and based of user changes the data will be again binded to model from view)

**JAVA(16-08-23)**

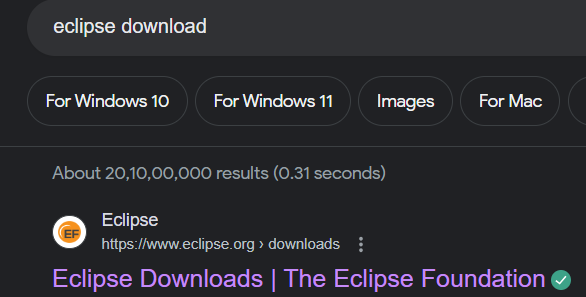
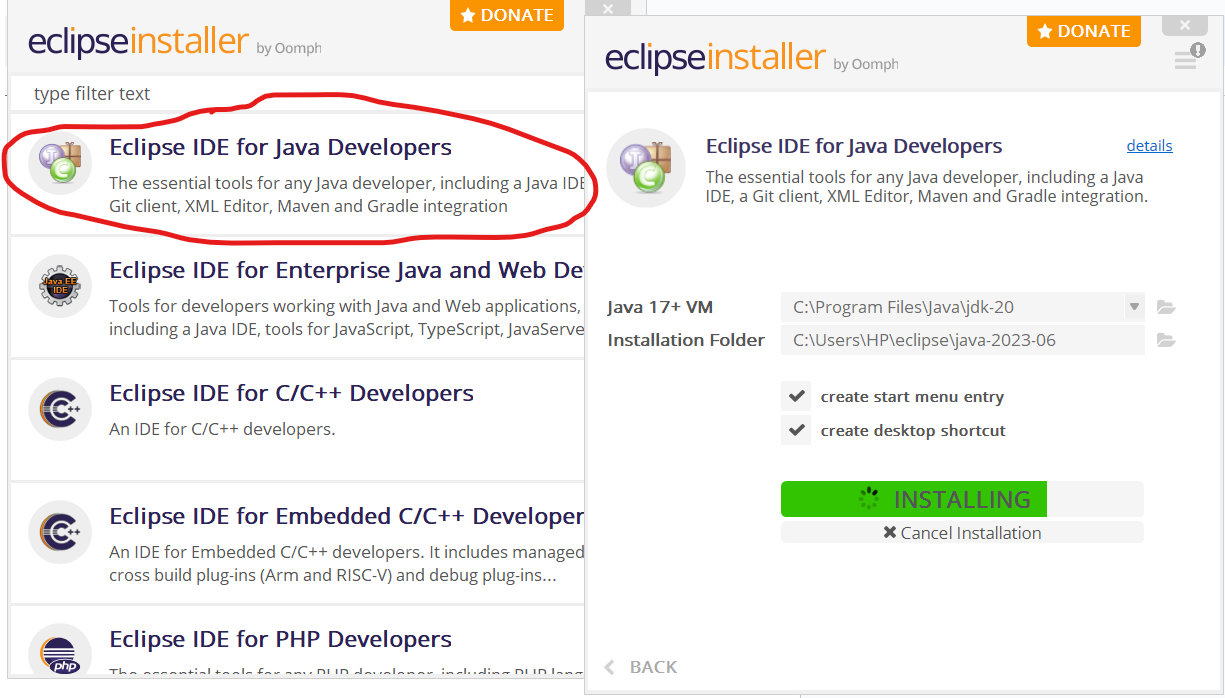
* The world wide Web has popularised the use of java,bcoz programs can be transparently downloaded with the web pages and executed in computer with a java capable browser.
* A java Application is a standalone java program that can be executed independently of any web browser.
* A java applet is a program designed to be executed under a java capable browser.

**Java installation**

* download java (browser)
* goto oracle website for java
* jdk20 -> windows -> x64 installer
* Environmental variable set up->environment variable button ->System variables ->new -> variable name-java\_home->variable value is the path of java which will be in C:/ file
* ->system variables -> path (double click)
* ->new -> copy the 2 paths as shown below
* 
* 



**IDE FOR JAVA**

* INSTALL ECLIPSE
* 
* After downloading -> click on ide for java developers
* And install
* 
* After installing launch it.

**Types of java**

Java provides 10 fundamental types:

* Integer : **byte,short**,**int** and **long**
* Floating points: f**loat , double**
* Characters : **cha**r
* **Boolean**
* **Void**
* **String**

**Variables**

* The variables are declared by specifying its type and name , and initialised at the point of declaration , or later with the help of assignment expression.

Ex:

int x;

double f=0.333;

char c=”c”;

string s=”amala”;

Int x=10;

**Literals**

* The integer value can be written in decimal ,hexadecimal,octal and long form

int x=34;

int y=0x3ef;

int z=0772;

Int m= 2433333344L

* The floating point values are if type **double** by default

double d=6.38;

float f=6.43F;

**EXPRESSIONS**

* Arithmetic
* Bit level
* Relational
* Logical
* String related

**Arithmetic expressions**

* Addition
* Subtraction
* Division
* Multiplication
* Modulus

**Relational operators**

* Equivalent (==)
* not equal (!=)
* Less than (< )
* Greater than (>)
* Less Than or equal (<=)
* Greater than or equal(>=)
* Relational expressions always returns a boolean value

**Bit level operators**

* And (&)
* or(|)
* not(~)
* Shift left (<<)
* Shift right with sign extension(>>)
* Shift right with zero extension(>>>)
* Char ,byte,short arguments are promoted to int before and the result is an int

**Logical operators**

* And (&&)
* or(||)
* not(!)
* The Logical operator can only be applied to boolean expressions and return boolean values.

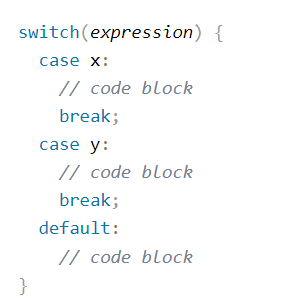
**String operators**

* Concatenation (+)
* If the expression begins with string and uses + operator then,the next argument is converted to string
* String cannot be compared with == and !=

**Switch statement**

* Java switch statement executes one statement form multiple conditions.
* It is like (if-else-if ladder)statement
* Switch statement works with byte ,short,int,long,enum types,string and some wrapper types like bye,short,long and int.
* Since java 7 we can use string in switch statements.

**syntax**



**Arrays**

* Array can be used to store a number of elements of the same type

int[] a;

float[] b;

String[] c;

* The declaration does not specify a size.however it can be inferred when initialised.

int[] a = {1,2,3,4,5,6};

float[] b = {1.2F,3.45F};

string[] c ={“java “,”is”,”great”};

* Components can be accessed with an integer index with values from 0 to length minus 1

a[2]=1000; // modify the third element of an array

* Every array has a member called length that can be used to get the length of an array

Int len=a.length; // get the size of an array