**1 . Directives:** The Angular directive helps us to manipulate the DOM. You can change the appearance, behavior, or layout of a DOM element using the Directives. The directives can be classified into three categories.

1.1. structural directives.

1.2. Component Directives.

1.3. Attributes directives.

**1.1.Component Directives:-** Usually, in main.ts file we write a class in typescript. When place the component directive over the class, angular treat that class as component.

Syntax:- @Component({

Selector:’ ‘;

Templates: ‘ ‘;

Styles:[ ]

})

 It contains the details on component processing, instantiated and usage at run time.

**1.2. Structural Directives**:- structural elements can change the DOM LayOUT by adding and removing the DOM elements. Structural directives start with a \* sign.

There are 3 structural directives.

A. ngFor.

B. ngSwith

C. ngIf

1.2.1,ngFor:- ngFor:- The \*ngFor directive is used to repeat a portion of HTML template once per each item from an iterable list (Collection).

Syntax:

<html-element \*ngFor=”let var-name of/in collection-obect”></html-element>

**Note**:-ngFor directive provides several variables. These variable values can be stored in local variable and used in templates. Variables are

* index: number: The zero-based index of the current element in the collection.
* count: number: The total no of items in the collection
* first: boolean: True when the item is the first item in the collection.
* last: boolean: Is set to True, when the item is the last item in the collection.
* even: boolean: True when the item has an even index in the collection.
* odd: boolean: is set to True when the item has an odd index in the collection.

1.2.2.ngIf:- The ngIf Directives is used to add or remove HTML elements based on an expression. The expression must return a boolean value. If the expression is false then the element is removed from DOM tree, else the element is inserted in layout.

Syntax:1

<html-element \*ngIf=”expression ”>

Content will be displayed , if expression it true, Otherwise this element is remove from DOM tree.

</html-element>

Syntax:2

<html-element \*ngIf=”expression then true-block else else-block ”>

</html-element>

<ng-template #true-block} …</ng-template>

<ng-template #else-block}…</ng-template>

1.2.3 ngSwitch:- it allows us to add or remove the DOM element based on predefined condition. It works in conjuction with ngSwitchCase and ngSwitchDefault directives.

Syntax:-

<container\_element [ngSwitch]="switch\_expression">

<inner\_element \*ngSwitchCase="match\_expresson\_1">...</inner\_element>

<inner\_element \*ngSwitchCase="match\_expresson\_2">...</inner\_element>

<inner\_element \*ngSwitchCase="match\_expresson\_3">...</inner\_element>

<inner\_element \*ngSwitchDefault>...</element>

</container\_element>

Note:-1. ngSwitch must be in outer-container.

2. ngSwitchcase must be in inner-container.

3.It displays all elements, if their case values is matched with swith-expression value. Otherwise it remove the element from layout.

Examples:

1.app.component.html

<h2 style="text-align:center;color:green;border: 2px solid red;"> Student Table </h2>

<hr>

<table [ngStyle]="{'width':'100vw','border':'2px solid orange'}" border="1">

    <tr>

        <th>name</th>

        <th>age</th>

        <th>marks</th>

    </tr>

    <tr \*ngFor="let i of xy" [ngStyle]="{'text-align':'center'}">

        <td>{{i.sname}}</td>

        <td>{{i.age}}</td>

        <td>{{i.marks}}</td>

    </tr>

</table>

2. app.component.ts

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

import { reduce } from 'rxjs';

@Component({

  selector: 'app-root',

  templateUrl: './app.component.html',

  styleUrls: ['./app.component.css']

})

export class AppComponent {

  x:stu\_data=new stu\_data('suku',41,[1,2,3]);

  y:stu\_data=new stu\_data('sv',39,[4,5,6]);

  z:stu\_data=new stu\_data('sumasuha',12,[10,12]);

  xy:stu\_data[]=[this.x,this.y,this.z];

  a:object={'color':'red','font-size':'25px'};

  b:string='two';

}

 class stu\_data{

  public sname:string;

  public age:number;

  public marks:number[];

  constructor(a:string,b:number,c:number[]){

    this.age=b;

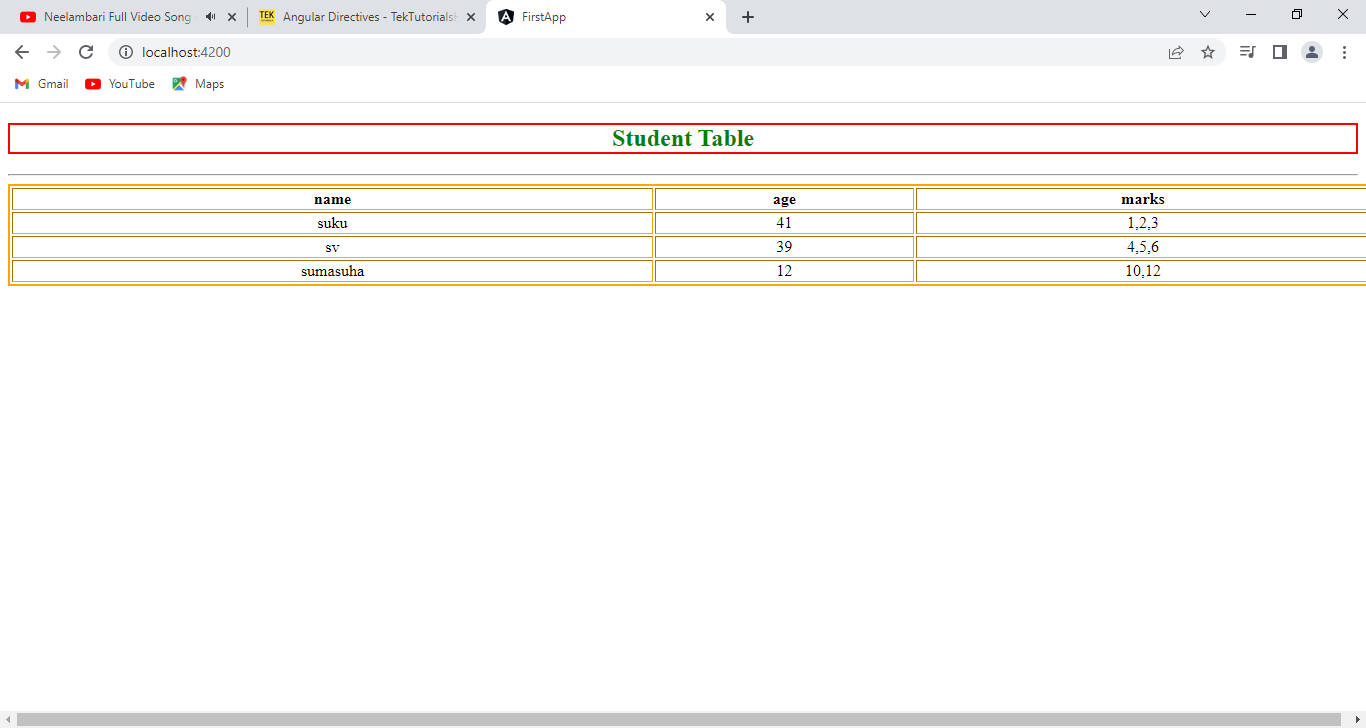
    this.marks=c;

    this.sname=a;

  }

  display(){console.log(this.age+this.sname+this.marks);}

}



Example:2

1.app.component.html

<h2 style="text-align:center;color:green;border: 2px solid red;"> Student Table </h2>

<hr>

<table [ngStyle]="{'width':'100vw','border':'2px solid orange'}" border="1">

    <tr>

        <th>name</th>

        <th>age</th>

        <th>marks</th>

    </tr>

    <tr \*ngFor="let i of xy" [ngStyle]="{'text-align':'center'}">

       <div \*ngIf="i.age>35 then one else two"></div>

        <ng-template #one>

            <td>{{i.sname}}</td>

            <td>{{i.age}}</td>

            <td>{{i.marks}}</td>

        </ng-template>

        <ng-template #two>

            <td>{{i.sname}}</td>

            <td>Minor</td>

            <td>Not-Studied</td>

        </ng-template>

    </tr>

</table>

2.app.component.ts

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

import { reduce } from 'rxjs';

@Component({

  selector: 'app-root',

  templateUrl: './app.component.html',

  styleUrls: ['./app.component.css']

})

export class AppComponent {

  x:stu\_data=new stu\_data('suku',41,[1,2,3]);

  y:stu\_data=new stu\_data('sv',39,[4,5,6]);

  z:stu\_data=new stu\_data('sumasuha',12,[10,12]);

  xy:stu\_data[]=[this.x,this.y,this.z];

  a:object={'color':'red','font-size':'25px'};

  b:string='two';

}

 class stu\_data{

  public sname:string;

  public age:number;

  public marks:number[];

  constructor(a:string,b:number,c:number[]){

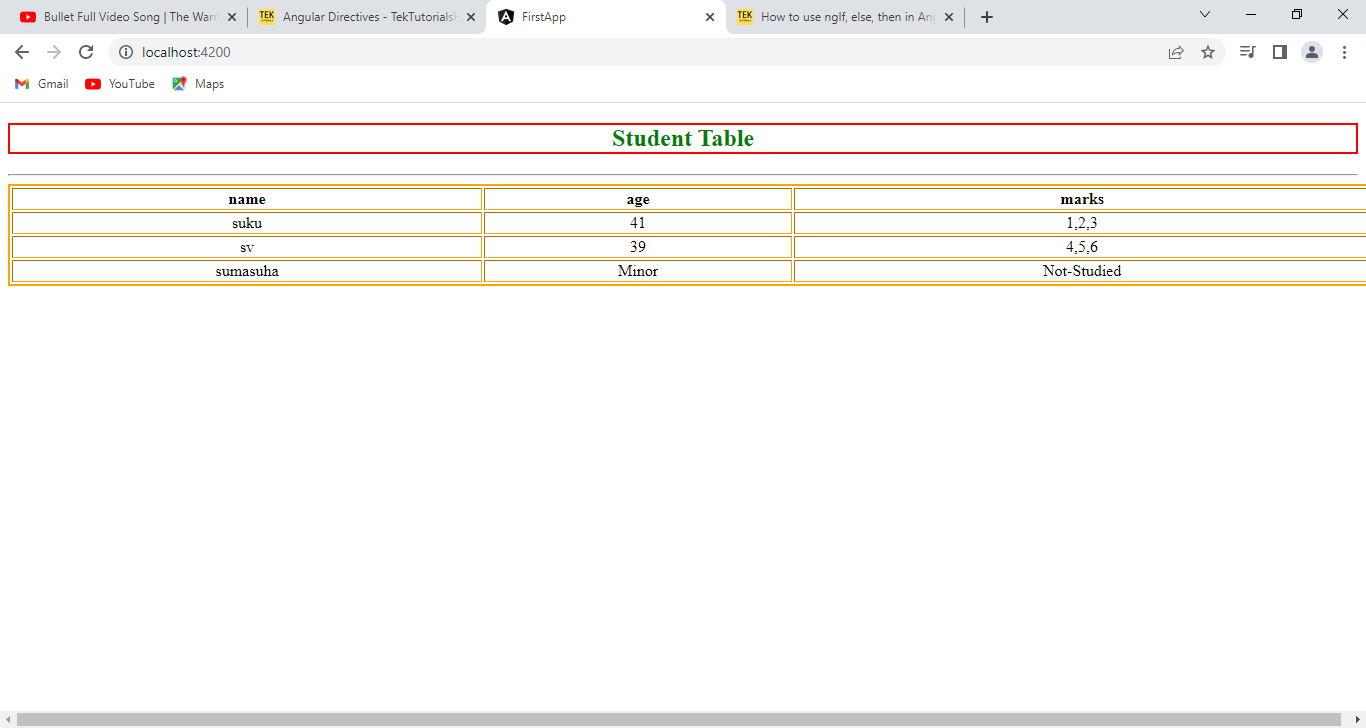
    this.age=b;

    this.marks=c;

    this.sname=a;

  }

}



Example:3

1.app.component.html

<h3 [ngSwitch]='y'>

    <p \*ngSwitchCase='1'>MONDAY</p>

    <p \*ngSwitchCase='2'>TUESDAY</p>

    <p \*ngSwitchCase='3'>WEDNESDAY</p>

    <p \*ngSwitchCase='4'>THURSDAY</p>

    <p \*ngSwitchCase='5'>FRIDAY</p>

    <p \*ngSwitchCase='6'>SATUDAY</p>

    <p \*ngSwitchCase='7'>SUNDAY</p>

</h3>

<h4>

    {{y}}

</h4>

2.app.component.ts

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

import { reduce } from 'rxjs';

@Component({

  selector: 'app-root',

  templateUrl: './app.component.html',

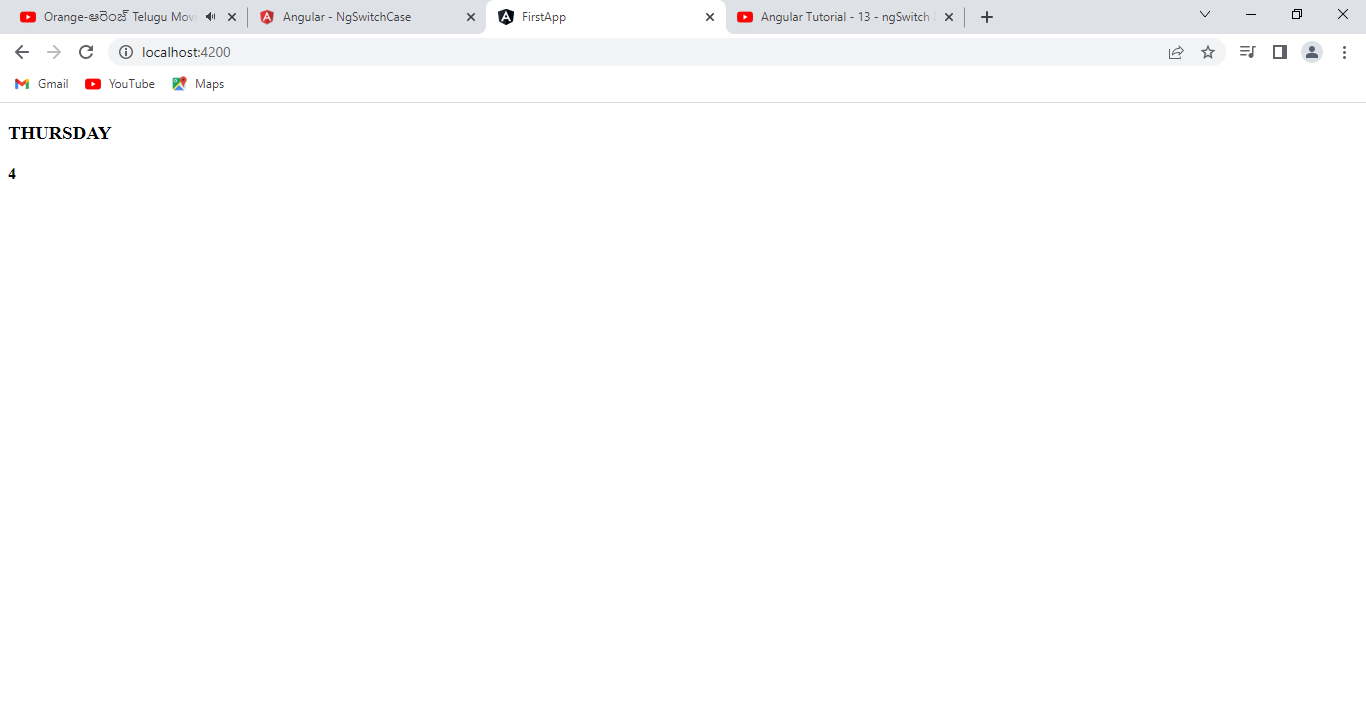
  styleUrls: ['./app.component.css']

})

export class AppComponent {

    y:number=Math.floor(Math.random()\*10);

}



**1.3.Attribute directives:** Attribute dierctives can change the appearance /behaviour of html elements.The Attribute Directives are

1.3.1. ngModule.

1.3.2. ngStyle.

1.3.3. ngClass.

1.3.1.ngClass:- The ngClass directive adds and removes CSS classes on an HTML element.

Syntax:

<html-element [ngClass]=” expression ”>..element</html-element>

Where expression can be in one of the following forms.

a.string🡪 It is type of property of component class.

b.Array🡪 It contains properties of component class.

c.object.--> { classname:boolean type property of component class,..etc}

1.3.2 ngStyle:- This directive is used to change the multiple style properties of html element.

Syntax:

<html-element [ngStyle]=”expression”>content </html-element>

Where expression can be in one of the following forms.

a.object.--> { ‘classname’:’ ’boolean type property of component class’,..etc}

Example:1

1.app.component.html

<h2 [ngClass]="a">This is sukumar</h2>

<h2 [ngClass]="b">This is srinivasa</h2>

<h2 [ngClass]="c">This is Govinda</h2>

<h2 [ngClass]="[a,d]">Narayana</h2>

<h2 [ngClass]="{one:x}">Vishnu</h2>

2.app.component.css

.one{

 background-color: red;

 margin:5px;

}

.two{

background-color: green;

margin:5px;

}

.three{

background-color: blue;

margin:5px;

}

.four{

    color:black;

    font-size: medium;

}

3.app.component.ts

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

import { reduce } from 'rxjs';

@Component({

  selector: 'app-root',

  templateUrl: './app.component.html',

  styleUrls: ['./app.component.css']

})

export class AppComponent {

    a:string='one';

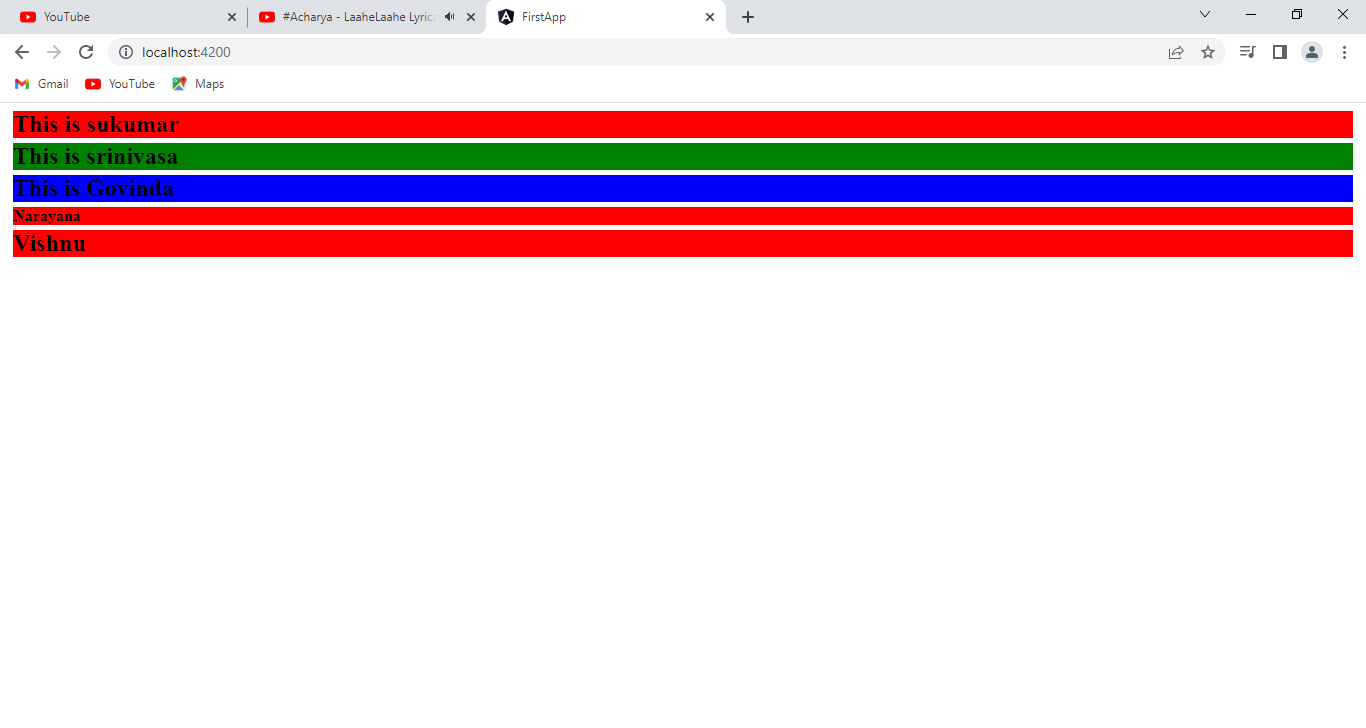
    b:string='two';

    c:string='three';

    d:string='four';

    x:boolean=true;

}



Example:2

1.app.component.html

<h1 [ngStyle]="{'color':'blue'}">This is first element</h1>

<h2 [ngStyle]='b'>This is second Element</h2>

2.app.component.ts

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

import { reduce } from 'rxjs';

@Component({

  selector: 'app-root',

  templateUrl: './app.component.html',

  styleUrls: ['./app.component.css']

})

export class AppComponent {

    a:any='color:red;border:1px solid green; width:50vw; margin:5px 0px;';

    b:object={'color':'red','border':'1px solid green', 'width':'50vw','margin':'5px 0px'};

}

