

11814605

Advitya

Q1>

Class binding

Class binding in Angular enables us to add capability to make use of our typescript code to enable, disable or add classes , there are two types of class binding

- Single class binding
- Multi class binding

Single class binding enables us to add a single class to our html tag ,if the value of the class passed is true then the class is applied to the tag else it's not

Eg:-

Html code

```
<div [class.red]>color</div>
```

Ts code

```
red=true
```

class is applied

Multi class binding enables us to add multiple classes to our html tag unlike single class binding

Eg:-

HTML code

```
<div [class]='theme'>color</div>
```

Ts code

```
color="red gradient blur"
```

all the classes are applied

Code

Html code

```
<button [class.colorx]="enableClr" [class.sizeX]="enableSiz" [class.hovx]="enableHov">
  single class binding
</button>

<br />
```

```
<button [class]="multiVar">Multi class binding</button>
```

Css

```
.colorx{  
  background-color: aqua;  
}  
  
.sizex{  
  height: 100px;  
  width: 200px;  
}  
  
.hovx:hover {  
  background-color: blueviolet;  
}
```

Ts code

```
import { Component, OnInit } from '@angular/core';  
  
@Component({  
  selector: 'app-ca',  
  templateUrl: './ca.component.html',  
  styleUrls: ['./ca.component.css'],  
})  
export class CaComponent {  
  enableClr = true;  
  enableHov = true;  
  enableSiz = true;  
  multiVar = 'colorx sizex hovx';  
}
```

output

question 1 11814605 Advitya G2

single class binding

Multi class binding

Q2>

Angular directives :

Angular directives are used to add other functionalities to HTML tags

Eg:-

NgIf – used for render html elements if condition is satisfied

Html

```
<div *ngIf="name == 'Advitya'">
  <h4>name is Advity</h4>
</div>
```

Ts code

```
name='Advitya'
```

NgFor – used for looping through

Html

```
<h1 *ngFor="let item of forLoop">{{ item }}</h1>
```

Ts code

```
forLoop = ["hello", "how", "are you"];
```

ngClass – used for adding classes to the html tag

HTML

```
<button ngClass="btn btn-danger">button</button>
```

ngSwitchCase – used for selecting from multiple conditions

HTML

```
<div [ngSwitch]="icrecream">
  <div *ngSwitchCase="'vanilla'">vanilla flavour on the way !</div>
  <div *ngSwitchCase="'chocolate'">chocolate flavour on the way !</div>
  <div *ngSwitchCase="'strawberry'">strawberry flavour on the way !</div>
  <div *ngSwitchCase="'mint'">mint flavour on the way !</div>
</div>
```

Ts code

```
import { Component, OnInit } from '@angular/core';

@Component({
  selector: 'app-switch',
  templateUrl: './switch.component.html',
})
export class SwitchComponent {
  icrecream = 'chocolate';
}
```

Output

question 2 1 11814605 Advitya G2

chocolate flavour on the way !

Q3>

Bootstrap forms

Bootstrap forms help us to make our forms look good, it also helps to make our forms responsive

Form group class

This class is used for grouping together different html tags together

```
<form>
  <div class="form-group">
    <label for="name">name</label>
    <input type="text" />
  </div>
  <div class="form-group">
    <label for="class">class</label>
    <input type="text" />
  </div>
</form>
```

Form controls class

Class is used for styling input,select and textarea

```
<form>
  <input class="form-control" type="text" />
</form>
```

Types of forms

Template driven forms

Template driven forms are more easy to make because we are choosing from a predefined template these forms also take less time to make but the customisations are limited

Reactive forms

These forms give more options to customize as compared to template forms but take more time to make as there is no template and we need to add each option our self

NgModel

Used to bind the view to model,by using NgModel we can validate the forms and get the values stored in the tag eg:- we can get the values in text box

Ts code

```
import { Component, OnInit } from '@angular/core';
import { NgForm } from '@angular/forms';

@Component({
  selector: 'app-formca',
  templateUrl: './form.component.html',
})
export class FormComponentCa {
  out = '';
  onSubmit(input: NgForm) {
    this.out = 'Registered';
  }
}
```

HTML code

```
<h2 class="jumbotron bg-info form-group">Registration page</h2>
<form #ngf="ngForm" (ngSubmit)="onSubmit(ngf)">
  <label> first name </label>
  <input
    class="form-control"
```

```

    required
    name="name"
    ngModel
    #first="ngModel"
    type="text"
  />
<div class="alert alert-danger" *ngIf="first.touched && !first.valid">
  Enter a name
</div>

<label> last name</label>
<input
  class="form-control"
  required
  name="last"
  ngModel
  #last="ngModel"
  type="text"
/>

<label> Password</label>
<input
  class="form-control"
  required
  name="pass"
  ngModel
  #pass="ngModel"
  type="password"
/>
<div class="alert alert-danger" *ngIf="pass.touched && !pass.valid">
  Enter a password
</div>

<label> rewrite Password</label>
<input
  class="form-control"
  required
  name="passr"
  ngModel
  #passr="ngModel"
  type="password"
/>

<div class="alert alert-danger" *ngIf="passr.touched && !passr.valid">
  Enter a password
</div>
<div class="alert alert-danger" *ngIf="pass.value != passr.value">
  password not matching

```

```

</div>

<button class="btn btn-success" type="submit">submit</button>
</form>

<span class="badge badge-success" style="height: 100px; width:500px; line-
height: 70px; font-size: 50px !important;">{{out}}</span>

```

Output

question 3 11814605 Advitya G2

Registration page

first name

last name

Password

rewrite Password

submit

➤ Final app.component.html code

```

<h1> question 1 11814605 Advitya G2</h1>
<app-ca></app-ca>

<h1> question 2 11814605 Advitya G2</h1>
<app-switch></app-switch>

<h1> question 3 11814605 Advitya G2</h1>
<app-formca></app-formca>

```