**DataBinding Quiz**

Chris has created a highlight directive as shown below. What color will be applied to the page if he enters yellow in the input field?

//highlight.directive.ts

import {Directive, ElementRef,Input} from '@angular/core';

@Directive({

selector:['highlight']

})

export class HighlightDirective{

constructor(el:ElementRef){

    el.nativeElement.style.backgroundColor='blue';

}

}

//app.component.ts

template:`

              <label>Provide color to be applied on the page</label>

              <input type='text' [(ngModel)]='colorName'>

              <div highlight>Hello </div>

          `



color red will be applied to the div element



color yellow will be applied to the div element



yellow color will not be applied as there is no binding of model colorName and highlight

Q2 of 3outlined\_flag

What is the correct statement to be placed at line number 3 to disable the button?

Line 1: @Component({

Line 2: selector: 'my-app',

Line 3: template: '<button ….>Click Here</button>'

Line 4: })

Line 5: export class AppComponent {

Line 6: exists: Boolean = true;

Line 7: }



(disabled)="exists"



disabled = {{ exists }}



[disabled] = exists



[disabled]= “exists”

Q3 of 3outlined\_flag

The below code snippet has some lines of code missing.

Line 1: @Component({

Line 2: template:`

Line 3: <div>

Line 4: <h1 [textContent]='"user is" +person.name'> </h1>

Line 5: <table>

Line 6: <tr>

Line 7: <td [attr.colspan]='1+1'>

Line 8: <button >Female</button>

Line 9: </td>

Line 10: <td [attr.colspan]='1+1'>

Line 11: <button>Male</button>

Line 12: </td>

Line 13: </tr>

Line 14: <tr>

Line 15: <td>{{person.rating}}</td> //line2

Line 16: <td>{{person.address}}</td>

Line 17: </tr>

Line 18: </table>

Line 19: </div>

Line 20: `

Line 21: })

export class AppComponent{

private female={

name:'Jenny Wesley' ,

gender:'F',

rating:4,

address:'Wesley states'

}

private male={

name:'Ross Green',

gender:'M',

address:'New York'

}

private person=this.female;

}

As per the requirement mentioned below please complete the lines.

Requirement:

1. set the background color yellow for the div element (line3)
2. hide rating if not present (line15)



line3: [style.backgroundColor]='yellow' line15: <td [hidden]='!person.rating'>



line3: [style.backgroundColor]=' "yellow" ' line15: <td hidden='!person.rating'>



line3: [style.background-color]=' "yellow" ' line15: <td [hidden]='!person.rating'>



line3: [style.background-color]=' "yellow" ' line15: <td hidden='!person.rating'>

**Quiz with Answers:**

Q1 of 3outlined\_flag

Chris has created a highlight directive as shown below. What color will be applied to the page if he enters yellow in the input field?

//highlight.directive.ts

import {Directive, ElementRef,Input} from '@angular/core';

@Directive({

selector:['highlight']

})

export class HighlightDirective{

constructor(el:ElementRef){

    el.nativeElement.style.backgroundColor='blue';

}

}

//app.component.ts

template:`

              <label>Provide color to be applied on the page</label>

              <input type='text' [(ngModel)]='colorName'>

              <div highlight>Hello </div>

          `



color red will be applied to the div element



color yellow will be applied to the div element



yellow color will not be applied as there is no binding of model colorName and highlight

**Option yellow color will not be applied as there is no binding of model colorName and highlight is Correct**

Q2 of 3outlined\_flag

What is the correct statement to be placed at line number 3 to disable the button?

Line 1: @Component({

Line 2: selector: 'my-app',

Line 3: template: '<button ….>Click Here</button>'

Line 4: })

Line 5: export class AppComponent {

Line 6: exists: Boolean = true;

Line 7: }



(disabled)="exists"



disabled = {{ exists }}



[disabled] = exists



[disabled]= “exists”

**Option [disabled]= “exists” is Correct**

Q3 of 3outlined\_flag

The below code snippet has some lines of code missing.

Line 1: @Component({

Line 2: template:`

Line 3: <div>

Line 4: <h1 [textContent]='"user is" +person.name'> </h1>

Line 5: <table>

Line 6: <tr>

Line 7: <td [attr.colspan]='1+1'>

Line 8: <button >Female</button>

Line 9: </td>

Line 10: <td [attr.colspan]='1+1'>

Line 11: <button>Male</button>

Line 12: </td>

Line 13: </tr>

Line 14: <tr>

Line 15: <td>{{person.rating}}</td> //line2

Line 16: <td>{{person.address}}</td>

Line 17: </tr>

Line 18: </table>

Line 19: </div>

Line 20: `

Line 21: })

export class AppComponent{

private female={

name:'Jenny Wesley' ,

gender:'F',

rating:4,

address:'Wesley states'

}

private male={

name:'Ross Green',

gender:'M',

address:'New York'

}

private person=this.female;

}

As per the requirement mentioned below please complete the lines.

Requirement:

1. set the background color yellow for the div element (line3)
2. hide rating if not present (line15)



line3: [style.backgroundColor]='yellow' line15: <td [hidden]='!person.rating'>



line3: [style.backgroundColor]=' "yellow" ' line15: <td hidden='!person.rating'>



line3: [style.background-color]=' "yellow" ' line15: <td [hidden]='!person.rating'>



line3: [style.background-color]=' "yellow" ' line15: <td hidden='!person.rating'>

**Option line3: [style.background-color]=' "yellow" ' line15: <td [hidden]='!person.rating'> is Correct**

**Quiz : Built-in Pipes**

Q1 of 3outlined\_flag

In a Library Management application, there is a requirement that all the books should have their price shown in the US-dollar symbol where 'books' is an array containing 'title', 'price', 'dateofpurchase' as attributes. Which of the following is the correct statement to display the price of each book?



<p \*ngFor = "let p of books"> {{ p.price | currency }} </p>



<p \*ngFor = "let p of books">{{ p.price | currency:"INR" }} </p>



<p \*ngFor = "let p of books"> {{ p.price | currency:"USD":'code' }} </p>



<p \*ngFor ="let p of books"> {{p.price | currency:"CAD":'symbol-narrow' }}</p>

Q2 of 3outlined\_flag

What will be the output for template expression with the pipe as given below ?

{{["a", "b", "c", "d", "e", "f"] | slice : 2:-1}}



c,d,e,f



d,e,f



c,d,e



Error: both negative and positive index cant be used together

Q3 of 3outlined\_flag

What will be the output for template expression with the pipe as given below ?

{{'1.25' | percent : '3.3-4'}}



001.250%



125.000%



012.500%



000.125%

**Builtin pipes Quiz Answers**

Q1 of 3outlined\_flag

In a Library Management application, there is a requirement that all the books should have their price shown in the US-dollar symbol where 'books' is an array containing 'title', 'price', 'dateofpurchase' as attributes. Which of the following is the correct statement to display the price of each book?



<p \*ngFor = "let p of books"> {{ p.price | currency }} </p>



<p \*ngFor = "let p of books">{{ p.price | currency:"INR" }} </p>



<p \*ngFor = "let p of books"> {{ p.price | currency:"USD":'code' }} </p>



<p \*ngFor ="let p of books"> {{p.price | currency:"CAD":'symbol-narrow' }}</p>

**Option <p \*ngFor = "let p of books"> {{ p.price | currency }} </p> is Correct**

Q2 of 3outlined\_flag

What will be the output for template expression with the pipe as given below ?

{{["a", "b", "c", "d", "e", "f"] | slice : 2:-1}}



c,d,e,f



d,e,f



c,d,e



Error: both negative and positive index cant be used together

**Option c,d,e is Correct**

Q3 of 3outlined\_flag

What will be the output for template expression with the pipe as given below ?

{{'1.25' | percent : '3.3-4'}}



001.250%



125.000%



012.500%



000.125%

**Option 125.000% is Correct**

# Quiz : Custom Pipes

Q1 of 3outlined\_flag

Find out the erroneous line in the below code snippet:

Line 1: import { PipeTransform} from '@angular/core';

Line 2: ……

Line 3: export class LengthPipe implements PipeTransform {

Line 4:  length(value: string): string{

Line 5: return "Length ="+value.length;

Line 6: }

Line 7:}



Line number 1



Line number 3



Line number 4



Line number 5

**Option Line number 4 is Correct**

Q2 of 3outlined\_flag

Sam has created a custom pipe called 'MyPipe' and he wants to use it within a module. In which of the following properties of module metadata, he should declare 'Mypipe'?



imports



providers



bootstrap



declarations

**Option declarations is Correct**

Q3 of 3outlined\_flag

Find out what is the error  in the below code snippet:

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

……

export class LengthPipe implements PipeTransform {

length(value: string): string{

return "Length ="+value.length;

}

}



length method should be an no args method



LengthPipe class should be using @Component decorator to make it working



The LengthPipe class has not implemented the transform() method of PipeTransform Interface



The length method should return an numeric value

**Option The LengthPipe class has not implemented the transform() method of PipeTransform Interface is Correct**

# Quiz-Directives

Q1 of 4outlined\_flag

In the below code snippet, what is the correct statement to be placed at line number 6 to display the numbers array as a list?

Line 1: import { Component } from '@angular/core';

Line 2: @Component({

Line 3:     selector: 'my-app',

Line 4:    template: `

Line 5:       <ul>

Line 6:        .........

Line 7:       </ul>

Line 8:   `

Line 9:  })

Line 10: export class AppComponent

Line 11: {

Line 12:     numbers: any[] = [1,2,3];

Line 13: }



<li \*ngFor = 'numbers'>{{ numbers }} </li>



<li \*ngFor = 'num of numbers'>{{ num }} </li>



<li \*ngFor = '#num of numbers'>{{ num }} </li>



<li \*ngFor = 'let num of numbers'>{{ num }} </li>

**Option <li \*ngFor = 'let num of numbers'>{{ num }} </li> is Correct**

Q2 of 4outlined\_flag

Which of the following are true with respect to directives? (Select two)

A component is also a directive

There will be no error in the code if you do not write the template property for a directive

There will be error in the code if you do not write the template property for a directive

You can leave the template property empty but mentioning template property is must for directives

**Option A component is also a directive is Correct**

**Option There will be no error in the code if you do not write the template property for a directive is Correct**

Q3 of 4outlined\_flag

Which of the following options are correct statements to apply the CSS class 'pink' to the paragraph element? (Select two)

<p [ngClass]='pink'> background is pink</p>

<p [ngClass]='{pink:true}'> background is pink</p>

<p [ngClass]='["pink"]'> background is pink</p>

<p ngClass ={pink:true}>background is pink</p>

**Option <p [ngClass]='{pink:true}'> background is pink</p> is Correct**

**Option <p [ngClass]='["pink"]'> background is pink</p> is Correct**

Q4 of 4outlined\_flag

In an application, an event when triggered should toggle the element by adding or deleting it from the DOM. When deleting, it should not hide the element in the DOM but has to completely delete it. Which concept in Angular is used to achieve this functionality?



Pipes



Interpolation



Structural Directives



Attribute Directives

**Option Structural Directives is Correct**

# Quiz : Nested Components

Q1 of 3outlined\_flag

Joe has created two components and his requirement is to share data between two components. How he can share data between two components?



He must use @Input and @Output in a component class



He can directly access one component’s data into another



He must use id value of each component to share data between them.



Data sharing between two components is not possible in Angular

**Option He must use @Input and @Output in a component class is Correct**

Q2 of 3outlined\_flag

What is the correct statement to be placed at line number 5 to load SecondComponent?

**first.component.ts**

Line 1: @Component({

Line 2:              selector:"first-component",

Line 3:              template: `

Line 4:               <h1> First Component </h1>

Line 5:               ....

Line 6:             `

Line 7: })

Line 8: export class FirstComponent { }

**second.component.ts**

@Component({

                Selector:"second-component",

                Template: "<h1> Second Component </h1>"

})

export class SecondComponent { }



<SecondComponent></SecondComponent>



<second-component></second-component>



SecondComponent.template



We cannot load SecondComponent

**Option <second-component></second-component> is Correct**

Q3 of 3outlined\_flag

Consider the given code snippets in respective files of ***display***and ***destinations***components.

<!-- display.component.html -->

<div class="col-md-4 offset-4">

<h2> Best Destinations </h2>

/\* Line 1 \*/

</div>

/\* destinations.component.ts \*/

destinations = [

{ destinationId: 10, destinationName: "Eiffel Tower", charges: 300 },

{ destinationId: 20, destinationName: "Taj Mahal", charges: 350 },

{ destinationId: 30, destinationName: "Statue of Unity", charges: 250 },

{ destinationId: 40, destinationName: "Great Wall of China", charges: 300 }

]

<!--destinations.component.html-->

<table border="1">

<thead>

<tr>

<th>Destination Name</th>

<th>Charges</th>

</tr>

</thead>

<tbody>

<tr \*ngFor="let destination of destinations">

<td>{{ destination.destinationName }}</td>

<td>{{ destination.charges }}</td>

</tr>

</tbody>

</table>

/\* Line 2 \*/

Which of the following options should be used to render the desired output?



Add <app-display></app-display> in destinations.component.html at Line 2



Add <app-destinations></app-destinations> in display.component.html at Line 1

**Option Add <app-destinations></app-destinations> in display.component.html at Line 1 is Correct**

**Forms Quiz**

Sam wants to create a model driven form for which he should use one of the pre-defined class in the root module. Which class is used for the same?



BrowserModule



ReactiveFormsModule



HttpClientModule



RouterModule

Q2 of 3outlined\_flag

What is the correct statement to bind required and maxlength validators to name field in a reactive form?



name: [' ', Validators.required, Validators.maxlength(10)]



name: [' ', [Validators.required, Validators.maxlength(10)]]



name: [Validators.required, Validators.maxlength(10)]



name: Validators.required, Validators.maxlength(10)

Q3 of 3outlined\_flag

Jack has to create a reactive form in Angular. When end user goes to any input field in the form, he is required to display a popup with some message, irrespective of the fact that user may or may not have typed anything in the field. How can Jack achieve this?



He can use untouched directive.



He can use touched directive.



He can use dirty directive.



He can use pristine directive.

# Quiz : Services

Q1 of 3outlined\_flag

Select the appropriate option to import the EmpService in the module class, so that it would be available to an entire application:



providers: [EmpService]



imports: [EmpService]



Inject: [EmpService]



declarations: [EmpService]

Q2 of 3outlined\_flag

Find the correct statement to be placed at Line number 1 to create a custom service?

Line 1:     .....

Line 2: export class MyService {

Line 3:      books: any[] = [{id:1,name:'TypeScript'},{id:2, name:'Angular'}];

Line 4:      getData(){

Line 5:         return books;

Line 6:      }

Line 7: }



@Service({name:'myService'})



@Injectable()



@inject()



@Injectable({name:'myService'})

Q3 of 3outlined\_flag

Pick the correct statements regarding services in Angular. (Select 3 options)

Services can be used for network connection tasks.

Services can be used for writing reusuable templates for components.

Services can be used for writing reusable operational tasks for components.

Angular will take care of injecting the mentioned service dependencies.

Q1 of 3outlined\_flag

Select the appropriate option to import the EmpService in the module class, so that it would be available to an entire application:



providers: [EmpService]



imports: [EmpService]



Inject: [EmpService]



declarations: [EmpService]

**Option providers: [EmpService] is Correct**

Q2 of 3outlined\_flag

Find the correct statement to be placed at Line number 1 to create a custom service?

Line 1:     .....

Line 2: export class MyService {

Line 3:      books: any[] = [{id:1,name:'TypeScript'},{id:2, name:'Angular'}];

Line 4:      getData(){

Line 5:         return books;

Line 6:      }

Line 7: }



@Service({name:'myService'})



@Injectable()



@inject()



@Injectable({name:'myService'})

**Option @Injectable() is Correct**

Q3 of 3outlined\_flag

Pick the correct statements regarding services in Angular. (Select 3 options)

Services can be used for network connection tasks.

Services can be used for writing reusuable templates for components.

Services can be used for writing reusable operational tasks for components.

Angular will take care of injecting the mentioned service dependencies.

**Option Services can be used for network connection tasks. is Correct**

**Option Services can be used for writing reusable operational tasks for components. is Correct**

**Option Angular will take care of injecting the mentioned service dependencies. is Correct**