

← le^x 2169 - Angular - Developing Single Page Application - Assessment



Q1 of 20

A developer wants to use the navbar and navbar-expand classes to create a navigation bar in one of the angular components.

Choose the correct code(s) that will help him achieve this. [Select any two]

- <nav [ngClass] = "['navbar', 'navbar-expand']" > ... </nav>
- <nav [ngClass] = "['navbar', 'navbar-expand']" > ... </nav>
- <nav [ngClass] = "{ 'navbar': true, 'navbar-expand': true }" > ... </nav>
 *
- <nav [ngClass] = "{ 'navbar': true, 'navbar-expand': true }" > ... </nav>

Q2 of 20

Fill in the blanks appropriately at line 1, to add a custom CSS class "myClass" for styling the div tag



Type here to search



DELL

← lex 2169 - Angular - Developing Single Page Application - Assessment

Q2 of 20

Fill in the blanks appropriately at line 1, to add a custom CSS class "myClass" for styling the div tag

```
@Component({  
  selector: 'my-app',  
  template: `<div _____ 1 _____ > How to apply styling using a class? </div>`  
})  
export class AppComponent {  
  myClass: string = 'myClass';  
}
```

(class) = "myClass"

[class] = "'myClass'" 

class = "{{myClass}}"

[class] = "myClass"

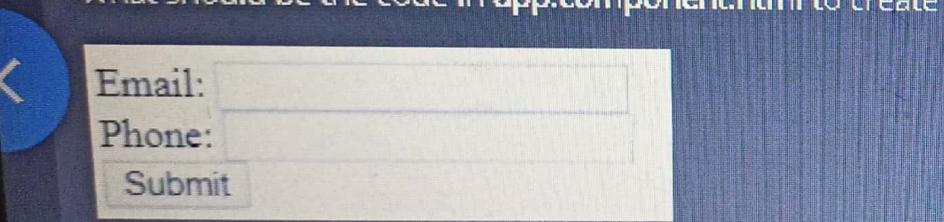
Act
Go to

← lek 2169 - Angular - Developing Single Page Application - Assessment

Consider the below code given in **app.component.ts** of an angular application:

```
email: string;  
phone: number;  
  
onSubmit() {  
    console.log(this.email, this.phone);  
}
```

What should be the code in **app.component.html** to create a form with two input fields i.e. **Email** and **Phone** as shown below:



Email:

Phone:

Submit

The data entered by the user should be displayed on the console on form submission?

Option A:

```
<form (ngSubmit)="onSubmit()">  
    Email:<input type="email" [(ngModel)]="email" name="txtEmail">  
    <br>  
    Phone:<input type="number" required [(ngModel)]="phone" name="txtPhone">  
    <br>  
    <button type="submit">Submit</button>  
</form>
```



Type here to search



```
<br>
<button type="submit">Submit</button>
</form>
```

Option B:

```
<form #frm="ngForm" (ngSubmit)="onSubmit()">
  Email:<input type="email" [(ngModel)]="email" name="txtEmail" #em="ngModel">
  <br>
  Phone:<input type="number" required [(ngModel)]="phone" name="txtPhone" #ph="ngModel">
  <br>
  <button type="submit">Submit</button>
</form>
```

Option C:

```
<form>
  Email:<input type="email" [(ngModel)]="email">
  <br>
  Phone:<input type="number" required [(ngModel)]="phone">
  <br>
  <button type="submit" (submit)="onSubmit()">Submit</button>
</form>
```

Only A

Type here to search



```
<br>
<button type="submit" (submit)="onSubmit()">Submit</button>
</form>
```

Only A

Only B

A and B

A, B and C



Q4 of 20

If the model has a property - `selected:boolean=true;` Which of the below code will display the output as "Bat"?

Option A:

```
<div *ngIf="selected && false; else bat">Bat</div>
<ng-template #bat>Man</ng-template>
```

Type here to search





2169 - Angular - Developing Single Page Application - Assessment

```
<div *ngIf="!selected; else bat">Bat</div>
<ng-template #bat>Man</ng-template>
```

Option C:

```
<div *ngIf="{{selected}}; else bat">Bat</div>
<ng-template #bat>Man</ng-template>
```

Option D:

```
<div *ngIf="selected; else bat">Man</div>
<ng-template #bat>Bat</ng-template>
```

[a] and [d]

[b] and [d]

only [c]

[a] only



Type here to search



```
boatArr=[  
    {boatId:100, boatName:'Boat1', boatRating:30},  
    {boatId:200, boatName:'Boat2', boatRating:32},  
    {boatId:300, boatName:'Boat3', boatRating:34},  
    {boatId:400, boatName:'Boat4', boatRating:37}  
]
```

The given array has to be used to render the boat details as shown below.

| | | |
|-----|-------|----|
| 100 | 30 | |
| 200 | 32 | |
| 300 | 34 | |
| 400 | Boat4 | 37 |

Identify the option(s) that will help in achieving the given result?

Option A:

```
<table>  
    <tr *ngFor="let boat of boatArr">  
        <td>{{boat.boatId}}</td>  
        <td *ngIf="boat.boatRating>35">{{boat.boatName}}</td>  
        <td>{{boat.boatRating}}</td>  
    </tr>
```



Type here to search



lex.infosysapps.com/en/viewer/quiz/lex_32689368853132337000?collectionId=lex_14474949806064925000&co

Apps Gmail YouTube GitHub Dashboard | Hacker... Programming tutor... Test Application Oracle Quer

← Lex 2169 - Angular - Developing Single Page Application - Assessment

```
<table>
  <tr *ngIf="boat.boatRating>35" *ngFor="let boat of boatArr">
    <td>{{boat.boatId}}</td>
    <td>{{boat.boatName}}</td>
    <td>{{boat.boatRating}}</td>
  </tr>
</table>
```

Option C:

```
<table>
  <tr *ngFor="let boat of boatArr">
    <td>{{boat.boatId}}</td>
    <td>{{boat.boatName}}</td>
    <td *ngIf="boat.boatRating>35">{{boat.boatRating}}</td>
  </tr>
</table>
```

both A and B

only A



Type here to search



Q6 of 20

Which component will be loaded in the browser when the user hits the URL with route path '/search'?

```
const routes: Routes = [
  {path: '', redirectTo: 'home', pathMatch: 'full'},
  {path: 'find', redirectTo: 'search'},
  {path: 'home', component: HomeComponent},
  {path: 'search', component: SearchComponent},
  {path: 'about', component: AboutComponent}
];
```

- HomeComponent
- SearchComponent
- AboutComponent
- FindComponent



Which of the following is the correct way to implement a custom validator in angular to validate email id having '@infy.com' as the domain?

Option A:

```
export class EmailValidator implements Validator {  
    validate(control: FormControl): { [key: string]: any } {  
        const emailRegexp = /^[\\w._]+@infy.com$/;  
        if (!emailRegexp.test(value)) {  
            return { "emailValid": true };  
        }  
        return null;  
    }  
}
```

Option B:

```
export class EmailValidator implements Validator {  
    validate(control: FormControl): { [key: string]: any } {  
        const emailRegexp = /^[\\w._]+@infy.com$/;  
        if (!emailRegexp.test(control)) {  
            return { "emailValid": false };  
        }  
        return null;  
    }  
}
```

Option C:

```
}
```

Option C:

```
export class EmailValidator implements Validator {
    validate(control: FormControl): { [key: string]: any } {
        const emailRegexp = /^[\\w._]+@[infy.com$/;
        if (!emailRegexp.test(control.value)) {
            return { "emailValid": false };
        }
        return null;
    }
}
```

Option D:

```
export class EmailValidator implements Validator {
    validate(control: FormControl): { [key: string]: any } {
        const emailRegexp = /^[\\w._]+@[infy.com$/;
        if (!emailRegexp.test(control.value)) {
            return { "emailValid": true };
        }
        return null;
    }
}
```



Type here to search





2169 - Angular - Developing Single Page Application - Assessment

```
export class EmailValidator implements Validator {
    validate(control: FormControl): { [key: string]: any } {
        const emailRegexp = /^[\\w._]+@[infy.com$/;
        if (!emailRegexp.test(control.value)) {
            return { "emailValid": true };
        }
        return null;
    }
}
```

- Option A
- Option B
- Option C
- Option D



```
fetchData() {  
  
    this.data = new Observable(observer => {  
        setTimeout(() => { observer.next(111); }, 100);  
        setTimeout(() => { observer.next(222); }, 200);  
        setTimeout(() => { observer.error(new Error('Something bad happened!')); }, 2000);  
        setTimeout(() => { observer.complete(); }, 2500);  
    });  
  
    let sub = this.data.subscribe((value) => this.myArray.push(value),  
        error => this.errors = true,  
        () => this.completed = true);  
  
}  
  
What will be output when fetchData() will be called?
```

- Values: 111 Errors: true Completed: true
- Values: 111 222 Errors: Completed: true
- Values: 111 222 Errors: true Completed:
- Values: 111 Errors: true Completed:



Type here to search



Infosys - Meeting Detail | How to Add a Label to HTML | Simple Form - Tryout - Viewer | 2169 - Angular - Developing S... | Home | Lex

lex.infosysapps.com/en/viewer/quiz/lex_32689368853132337000?collectionId=lex_14474949806064925000&collectionType=Course

Apps Gmail YouTube GitHub Dashboard | Hacker... Programming tutor... Test Application Oracle Queries - jav... Chegg Tenda Wireless Rout...

← lex 2169 - Angular - Developing Single Page Application - Assessment

Values: 111 Errors: true Completed:

Values: 111 Errors: true Completed:

Q9 of 20

Consider the given products array in the app.component.ts file:

```
let products = [{id:101, name: "Redmi Note 7s"}, {id:102, name: "Redmi Note 7 pro"}];
```

Which of the following is the correct way to redirect to the product-details component on click of the product name in the app component by passing id as a route parameter?

- <a *ngFor="let product of products" [routerLink]="/product-details", id> {{ product.name }}
- goToProductDetails(id) { this.router.navigate(['/product-details', id]); }
- goToProductDetails(id) { this.router.navigate([id]); }
- <a *ngFor="let product of products" [routerLink]="/product-details", product.id> {{ product.name }}

Activate Windows
Go to Settings to activate Windows

Type here to search

15:15 31-03-2021

Q10 of 20

Which of the below codes will display the time in the given format - "5:13 PM"
Note: `dateVal = new Date();`

- `{{ dateVal | date: 'fullTime' }}`
- `{{ dateVal | date: 'time' }}`
- `{{ dateVal | date: 'shortTime' }}`
- `{{ dateVal | date: 'Time' }}`

Q11 of 20

Observe the code given below:

```
/* style-binding comment */
```



2169 - Angular - Developing Single Page Application - Assessment

Observe the code given below:

```
/* style-binding.component.ts */
import { Component } from '@angular/core';

@Component({
  selector: 'style-binding',
  templateUrl: './style-binding.html'
})
export class StyleBindingComponent {
  isApplied=true;
}

/* style-binding.component.html */
<div>
  <button class="btn" [style.backgroundColor]="isApplied?'blue':'red">Style</button>
</div>
```

What will be the color of the **Style** button when the component loads in browser?

grey

blue

red

← lex 2169 - Angular - Developing Single Page Application - Assessment

Q12 of 20

Consider the below partially implemented code given in respective files of the app and fav-color components:

Note:- app is the parent component and fav-color is the child component.

```
/* app.component.ts */
@Component({
  selector: 'app',
  template: `Enter your color: <input type="text" />
    /* Line 1 */
})
export class AppComponent {
  /* Line 2 */
  favColor:string;
  selectedColor:string;
}

//fav-color.component.ts
@Component({
  selector: 'fav-color',
  template: `Your selected color is {{selectedColor}}`
})
export class FavColorComponent {
  /* Line 3 */
  selectedColor:string;
```

Type here to search



DELL

```
/* Line 3 */  
selectedColor:string;  
}  
}
```

The value of **favColor** property from the app component needs to be passed to the fav-color component. Fill in the blanks appropriately to achieve this.

- line 1 - <fav-color [selectedColor]="favColor"></fav-color> line 2 - No code required line 3 - @Input()
- line 1 - <fav-color [selectedColor]="favColor"></fav-color> line 2 - @Output() line 3 - @Input()
- line 1 - <fav-color [favColor]="selectedColor"></fav-color> line 2 - No code required line 3 - @Input()
- line 1 - <fav-color [favColor]="selectedColor"></fav-color> line 2 - @Output() line 3 - @Input()

Q13 of 20

The Child component with selector 'batman' has a field defined as below:

@Input() name:string;

The child component has a view with below code:

{'Hello '+name}{{}}



Simple Form - Tryout - Viewer F x 2169 - Angular - Developing Si x Home | Lex

lex.infosysapps.com/en/viewer/quiz/lex_32689368853132337000?collectionId=lex_14474949806064925000&collectionType=Course

Apps Gmail YouTube GitHub Dashboard | Hacker... Programming tutor... Test Application Oracle Queries - jav... Chegg Tenda

← lex 2169 - Angular - Developing Single Page Application - Assessment

Q13 of 20

The Child component with selector 'batman' has a field defined as below:
`@Input() name:string;`

The child component has a view with below code:
`{{Hello ' +name}}`

The parent component has a field defined as below:
`selectedHero:string = "Batman";`

What should be the code placed in parent component to display the output as 'Hello Batman'?

- `<batman [name]="selectedHero"></batman>`
- `<batman name="{{Batman}}></batman>`
- `<batman [name]="{{selectedHero}}></batman>`
- `<batman name="selectedHero"></batman>`

Q14 of 20



Type here to search



Q14 of 20

Consider the below code written in superman (child component) and app (parent component) components:

```
/* superman.component.ts */
@Output() engagePower:EventEmitter = new EventEmitter();
onClick() {
    this.engagePower.emit("Superman's power engaged!!");
}
```

```
/* superman.component.html */
<input (click)="onClick()">
```

```
/* app.component.ts */
onClick(val) { console.log(val); }
```

```
/* app.component.html */
/* Line 1 */
```

What should be the code placed at Line 1 to display "Superman's power engaged!!" in the console?

- <superman (engagePower)="onClick(\$event)"></superman>
-



2169 - Angular - Developing Single Page Application - Assessment

```
[{"heroId":666, "heroName":"Batman", "heroAge": 25},  
 {"heroId":667, "heroName":"Spiderman", "heroAge": 26},  
 {"heroId":669, "heroName":"Superman", "heroAge": 35},  
 ]
```

Which one of the below code, is a valid way to assign the 'superHeros' array to the Observable returned from the service?

[A]

```
export class AppComponent {  
  superHeros:SuperHero[];  
  constructor(public shService:SuperHeroService) {  
    shService.getSuperHeroDetail().subscribe(  
      (data)=>{  
        this.superHeros = data.forEach(_=>_.age<25);  
      }  
    );  
  }  
}
```

[B]

```
export class AppComponent {
```

Type here to search

[W]

```
export class AppComponent {  
  superHeros:SuperHero[];  
  constructor(public shService:SuperHeroService) {  
    shService.getSuperHeroDetail().subscribe(  
      (data)=>{  
        data.filter(_=>_.age<25);  
      }  
    );  
  }  
}
```

[Q]

```
export class AppComponent {  
  superHeros:SuperHero[];  
  constructor(public shService:SuperHeroService) {  
    shService.getSuperHeroDetail().subscribe(  
      (data)=>{  
        this.superHeros = data.filter(_=>_.age<25);  
      }  
    );  
  }  
}
```

Type here to search



```
shService.getSuperHeroDetail().subscribe(  
  (data)=>{  
    this.superHeros = data.filter(_=>_age<25);  
  }  
);  
}  
};
```

- Only option A
- Only option C
- Only options A and B
- Only options A and C

Q16 of 20

Consider the code below, an Angular application has a service which tells whether the current user is logged in or not.

```
class UserService {  
  isLoggedIn(): boolean { return false; }  
}
```

Type here to search



```
class UserService {  
    isLoggedIn(): boolean { return false; }  
}
```

The application has the below route guard:

```
@Injectable()  
class LoginGuard implements CanActivate {  
    constructor(private userService: UserService, private router: Router) {};  
    canActivate() {  
        if(this.userService.isLoggedIn()) return true;  
        else this.router.navigate(['/login']);  
    }  
}
```

The above guard is added to the routes as given below:

```
const routes: Routes = [  
    {path: '', redirectTo: 'login', pathMatch: 'full'},  
    {path: 'login', component: LoginComponent},  
    {path: 'tracks', component: ArtistTrackListComponent},  
    {path: 'albums', component: ArtistAlbumListComponent, canActivate: [LoginGuard]},  
];
```

The above guard is added to the routes as given below:

```
const routes: Routes = [
  {path: '', redirectTo: 'login', pathMatch: 'full'},
  {path: 'login', component: LoginComponent},
  {path: 'tracks', component: ArtistTrackListComponent},
  {path: 'albums', component: ArtistAlbumListComponent, canActivate:[LoginGuard]},
];
```

Which component will be loaded if a logged in user navigates to '<http://localhost:4200/albums>'?

- ArtistAlbumListComponent
- Nothing will be loaded
- ArtistTrackListComponent
- LoginComponent

Q17 of 20



DELL



2169 - Angular - Developing Single Page Application - Assessment

Q17 of 20

Consider the code written in the `product.component.spec.ts` file as given below:

```
import { ProductComponent } from './demo.component'
describe('ProductComponentTest', ()=> {
  let prodComp = new ProductComponent();
  it('Should create the ProductComponent', ()=>{
    const prod = prodComp;
    expect(prod).toBeTruthy();
  });
  it('Should have title as `product works!`', ()=>{
    const title = prodComp.title;
    expect(title).toEqual('product works!');
  });
});
```

Which of the below statement(s) is/are TRUE based on the above code?

- [A] The above code tests that the product component is created.
- [B] The above code tests that the title property on the component class has the value 'product works!'.

- Only option B
- Both options A and B
- Neither options A nor B
- Only option A

Q18 of 20

Consider the below code in the super-hero.component.ts file:

```
@Component({  
  selector: 'super-hero',  
  templateUrl: './super-hero.component.html',  
  styleUrls: ['./super-hero.component.css']  
})  
export class SuperHeroComponent {  
  //Some component code...  
}
```

Type here to search



Q18 of 20

Consider the below code in the `super-hero.component.ts` file:

```
@Component({  
  selector: 'super-hero',  
  templateUrl: './super-hero.component.html',  
  styleUrls: ['./super-hero.component.css']  
})  
export class SuperHeroComponent {  
  //Some component code...  
}
```

Which of the following is the correct way to load `SuperHeroComponent` in the app component?

- <app-component-super-hero></app-component-super-hero>
- <app-super-hero><app-super-hero>
- <super-hero></super-hero>
- <super-hero-component><super-hero-component>



2169 - Angular - Developing Single Page Application - Assessment

Q19 of 20

Identify what is wrong in the given implementation of a custom pipe?

```
@Pipe({name: 'lengthPipe'})  
export class LengthPipe implements PipeTransform {  
  length(value: string): string {  
    return "Length =" + value.length  
  }  
}
```

- The length method should be an no args method.
- The LengthPipe class should be using @Component decorator to make it work.
- The LengthPipe class has not implemented the transform() method of PipeTransform Interface.
- The length method should return an numeric value.



2169 - Angular - Developing Single Page Application - Assessment

THE LENGTH METHOD SHOULD RETURN AN INITIAL VALUE.

Q20 of 20

Consider the below code is given in app.component.ts file of an angular application:

duration:string = "3 Hours";

Which of the following is the correct way to display this variable in the associated template as a link.

- A: <a>{{duration}}
- B: <a>
- C: <a>
- D: <a>

All the options A, B, C and D

Options A, B and D only

Option A only

Options A and D only



Type here to search

