```
import React from "react";

export default class App extends React.Component {
 Constructor(props) {
  super(props);
  this.state = {count: 8};
 }

// increase method

 render() {
  return (
   < div >
Counter: {this.state.count} < /h1 >
Increase < /button >
< /div >
  );
 }
}
```

}

Select correct method to be used to increase the count value when button is clicked.

Options

- increase = function() {
 this.setState({count: +this.state.count + 1});
 }
- increase = () => {
 this.setState({count: this.state.count++});
 }
- increase = () => {
 this.setState({count: +this.state.count + 1});
 }
- increase() {
 this.setState({count: ++this.state.count});
 }

Mark for Review | Prev | Next

Q2. Consider The following code. Describe("My Second Test Suite", function () {

```
it("Handling child tabs and browser controls", function () {
 Cypress.on("uncaught:exception", (err, runnable) => {

  return false;
 });
cy.visit("https://chercher.tech/java/index-selenium-webdriver");
cy.get(":nth-child(17) > .alsosee > a")
 .invoke("removeAttr", "target")
 .click();
});
});
```

Options

- ○ In the given program, .invoke("removeAttr", "target")removes the target attribute and allows cypress to load the new link in the same window.
- ○ returning false here prevents Cypress from failing the test
- ◉ To handle the child tabs, jQuery methods should be used
- ○ All Of The Mentioned

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

@Component({
 selector: 'app-demo',
 template: `
 < div id="carouselExampleSlidesOnly" class="carousel slide" data-ride="carousel"  >
  < div class="carousel-inner" >
    ...
  <  /div  >
< /div >
 styleUrls: []
})
export class DemoComponent {

 images = [
  "https://picsum.photos/id/235/600/400",
  "https://picsum.photos/id/236/600/400",
  "https://picsum.photos/id/237/600/400",
  "https://picsum.photos/id/238/600/400"
 ]

}
```

```
})

export class DemoComponent {


 images = [

  "https://picsum.photos/id/235/600/400",

  "https://picsum.photos/id/236/600/400",

  "https://picsum.photos/id/237/600/400",

  "https://picsum.photos/id/238/600/400"

 ]


}
```

Find the correct code snippet from the options to fulfil the requirement.

Options

- (●) < div class="carousel-item" *ngFor="let image of images;let i = index" [ngClass]="{'active': i==0}" >
 < img class="d-block w-100" [src]="image" alt="slide i" >
 < /div >
- ( ) < div class="carousel-item" *ngFor="let image in images;let i = index" [ngClass]="{'active': i==0}" >
 < img class="d-block w-100" [src]="image" alt="slide i" >
 < /div >
- ( ) < div class="carousel-item" *ngFor="let image of images;let i = index" [ngClass]="{i==0?'active':''}" >
 < img class="d-block w-100" [src]="image" alt="slide i" >
 < /div >
- ( ) < div class="carousel-item" *ngFor="let image in images;let i = index" [ngClass]="{i==0?'active':''}" >
 < img class="d-block w-100" [src]="image" alt="slide i" >
 < /div >

Mark for Review | Prev | Next

Q4. Consider below HTML code snippet.

< div >

< section class="form" >

Welcome < span > User < /span > < /h1 >

< /section >

< /div >

Find an optimized jquery query to select span element faster.

Options

- ○ $("div .list:first-child section.form h1#head span");
- ○ $("h1#head span");
- ○ $($("h1#head"), "span");
- ◉ $("h1#head").find("span");

Mark for Review | Prev | Next

Q5. In favour of bootstrap supported browsers, below are the statements given by your team members. You have to pick the correct statement.

Options

- ○ Bootstrap supports the all browsers and platforms.
- ○ Bootstrap supports Internet Explorer 7, 8, 10, 11, Microsoft Edge.
- ◉ Internet Explorer 10+ is supported; IE9 and down is not. Please be aware that some CSS3 properties and HTML5 elements are not fully supported in IE10, or require prefixed properties for full functionality.
- ○ Proxy browsers (such as Opera Mini, Opera Mobile's Turbo mode, UC Browser Mini, Amazon Silk) are also supported by bootstrap.

Q6. spec in Jasmine represents a test case inside the test suite. How to define angular spec in Jasmine ?

Options

- We can define spec by calling the global Jasmine function "it" which takes a string and an annonymous function.
 Example
 describe("Suite Name", function() {
 it("should test spec", function() {
 expect(logic()).toEqual(expectedValue);
 });
 });
- We can define spec by calling the global Jasmine function "it" which takes a string and an arrow function.
 Example
 describe("Suite Name", function() {
 it("should test spec", ()= > {
 expect(logic()).toEqual(expectedValue);
 });
 });
- We can define spec by calling the global Jasmine function "it" which takes a string and an arrow function inside another arrow function which is second argument in describe function. Example
 describe("Suite Name", ()= > {
 it("should test spec", ()= > {
 expect(logic()).toEqual(expectedValue);
 });
 });
- (selected) All above