

# Evaluation Test

Total points 14/20

Instruction:

- There are 20 objective type questions.
- Each question weightage is shown by points.

**Email address \***

madhavipasupuleti506@gmail.com

**Which angular module is loaded by default? \***

1/1

- ☒ ng
- ☐ ng-app

## What is deep linking in AngularJS? \*

0/1

- ☐ Deep linking allows you to encode the state of application in the URL so that it can be bookmarked.
- ☐ Deep linking is a SEO based technique.
- ☒ Deep linking refers to linking various views to a central page.
- ☐ None of the above.

### Correct answer

- ☒ Deep linking allows you to encode the state of application in the URL so that it can be bookmarked.

## What are the advantage of AngularJS? \*

0/1

- ☒ Data Binding - AngularJS provides a powerful data binding mechanism to bind data to HTML elements by using scope.
- ☒ Customize & Extensible - AngularJS is customized and extensible as per you requirement. You can create your own custom components like directives, services etc.
- ☐ Code Reusability - AngularJS allows you to write code which can be reused. For example custom directive which you can reuse.
- ☒ Support – AngularJS is mature community to help you. It has widely support over the internet. Also, AngularJS is supported by Google which gives it an advantage.
- ☒ Compatibility - AngularJS is based on JavaScript which makes it easier to integrate with any other JavaScript library and runnable on browsers like IE, Opera, FF, Safari, Chrome etc.
- ☒ Testing - AngularJS is designed to be testable so that you can test your AngularJS app components as easy as possible. It has dependency injection at its core, which makes it easy to test.

### Correct answer

- ☒ Data Binding - AngularJS provides a powerful data binding mechanism to bind data to HTML elements by using scope.
- ☒ Customize & Extensible - AngularJS is customized and extensible as per you requirement. You can create your own custom components like directives, services etc.
- ☒ Code Reusability - AngularJS allows you to write code which can be reused. For example custom directive which you can reuse.
- ☒ Support – AngularJS is mature community to help you. It has widely support over the internet. Also, AngularJS is supported by Google which gives it an advantage.
- ☒ Compatibility - AngularJS is based on JavaScript which makes it easier to integrate with any other JavaScript library and runnable on browsers like IE, Opera, FF, Safari, Chrome etc.
- ☒ Testing - AngularJS is designed to be testable so that you can test your AngularJS app components as easy as possible. It has dependency injection at its core, which makes it easy to test.

service method is used to create a service whose purpose is to do some defined task. \*

1/1

- ☒ true
- ☐ false

Which attribute specifies a unique alphanumeric identifier to be associated with an element? \*

1/1

- ☐ class
- ☒ id
- ☐ article
- ☐ html
- ☐ Other: .....

Where should we implement the DOM manipulation in AngularJS? \*

1/1

- ☒ In the directives. DOM Manipulations should not exist in controllers, services or anywhere else but in directives.
- ☐ In the controllers. DOM Manipulations should not exist in directives, services or anywhere else but in directives.
- ☐ In the services. DOM Manipulations should not exist in directives, controllers or anywhere else but in services.

#### Feedback

<http://ng-learn.org/2014/01/Dom-Manipulations/>

Directives can be applied to which all element type? \*

0/1

- ☐ Element
- ☒ Attribute
- ☐ Comment
- ☐ All of the above

Correct answer

- ☒ All of the above

What would be the output of following code ? \*

1/1

```
function mul(x) {  
    return function(y) {  
        return {  
            result: x * y,  
            sum: function(z) {  
                return x * y + z;  
            }  
        };  
    };  
}  
console.log(mul(2)(3).result);  
console.log(mul(2)(3).sum(4));
```

- ☒ 6, 10
- ☐ undefined undefined
- ☐ Reference Error
- ☐ 10, 6

## How do you share data between controllers? \*

1/1

- ☒ Create an AngularJS service that will hold the data and inject it inside of the controllers. Using a service is the cleanest, fastest and easiest way to test.
- ☒ Using events
- ☒ Using \$parent, nextSibling, controllerAs, etc. to directly access the controllers
- ☒ Using the \$rootScope to add the data on

### Feedback

*Create an AngularJS service that will hold the data and inject it inside of the controllers.*

*Using a service is the cleanest, fastest and easiest way to test. However, there are couple of other ways to implement data sharing between controllers, like:*

- Using events*
- Using \$parent, nextSibling, controllerAs, etc. to directly access the controllers*
- Using the \$rootScope to add the data on (not a good practice)*

*The methods above are all correct, but are not the most efficient and easy to test.*

What would be the output of following code ? \*

1/1

```
(function(){
    var arrayNumb = [2, 8, 15, 16, 23, 42];
    Array.prototype.sort = function(a,b){
        return a - b;
    };
    arrayNumb.sort();
    console.log(arrayNumb);
})();

(function(){
    var numberArray = [2, 8, 15, 16, 23, 42];
    numberArray.sort(function(a,b){
        if(a == b){
            return 0;
        }else{
            return a < b ? -1 : 1;
        }
    });
    console.log(numberArray);
})();

(function(){
    var numberArray = [2, 8, 15, 16, 23, 42];
    numberArray.sort(function(a,b){
        return a-b;
    });
    console.log(numberArray);
})();
```

- ☒ [ 2, 8, 15, 16, 23, 42 ] [ 2, 8, 15, 16, 23, 42 ] [ 2, 8, 15, 16, 23, 42 ]
- ☐ undefined undefined undefined
- ☐ [42, 23, 16, 15, 8, 2] [42, 23, 16, 15, 8, 2] [42, 23, 16, 15, 8, 2]
- ☐ Reference Error

What would be the output of following code ? \*

0/1

```
function Person(name, age){
    this.name = name || "John";
    this.age = age || 24;
    this.displayName = function(){
        console.log(this.name);
    }
}

Person.name = "John";
Person.displayName = function(){
    console.log(this.name);
}

var person1 = new Person('John');
person1.displayName();
Person.displayName();
```

- ☐ John Person
- ☐ John John
- ☐ John undefined
- ☒ John John

Correct answer

- ☒ John Person

What angular function is used to manually start up an angular application? \*

1/1

- ☒ angular.bootstrap
- ☐ angular.element
- ☐ angular.copy
- ☐ None of the above



What would be the output of following code ? \*

1/1

```
(function() {  
    var fooAccount = {  
        name: 'John',  
        amount: 4000,  
        deductAmount: function(amount) {  
            this.amount -= amount;  
            return 'Total amount left in account: ' + this.amount;  
        }  
    };  
    var barAccount = {  
        name: 'John',  
        amount: 6000  
    };  
    var withdrawAmountBy = function(totalAmount) {  
        return fooAccount.deductAmount.bind(barAccount, totalAmount);  
    };  
    console.log(withdrawAmountBy(400)());  
    console.log(withdrawAmountBy(300)());  
})();
```

- ☒ Total amount left in account: 5600 Total amount left in account: 5300
- ☐ undefined undefined
- ☐ Total amount left in account: 3600 Total amount left in account: 3300
- ☐ Total amount left in account: 5600 Total amount left in account: 5600
- ☐ Other: .....

angular.module is primarily used to create application module. 0/1

\*

- ☐ true
- ☒ false

Correct answer

- ☒ true

Can an HTML page have multiple “ng-app” directive for bootstrapping multiple AngularJS application? \*

1/1

- ☐ Yes
- ☒ No

What would be the output of following code ? \*

0/1

```
(function(){  
    function sayHello(){  
        var name = "Hi John";  
        return  
        {  
            fullName: name  
        }  
    }  
    console.log(sayHello().fullName);  
})();
```

- ☐ Hi John
- ☐ Uncaught TypeError: Cannot read property 'fullName' of undefined
- ☐ undefined
- ☒ Reference Error

Correct answer

- ☒ Uncaught TypeError: Cannot read property 'fullName' of undefined

Which of the following is not a combinatory while defining css 1/1 property? \*

- ☒ \*
- ☐ >
- ☐ ~
- ☐ +

#### Feedback

*Explanation: Combinator explains the relationship between the selectors. There are four combinators in CSS, child selector (>), adjacent sibling selector(+), general sibling selector(~), descendent selector (space).*

What would be the output of following code ? \*

1/1

```
(function() {  
    var greet = 'Hello World';  
    var toGreet = [].filter.call(greet, function(element, index) {  
        return index > 5;  
    });  
    console.log(toGreet);  
})();
```

- ☐ Hello World
- ☐ undefined
- ☐ World
- ☒ ['W', 'o', 'r', 'l', 'd']
- ☐ Other: .....

## What is a singleton pattern and where we can find it in Angularjs? \*

1/1

Is a great pattern that restricts the use of a class more than once. We can find singleton pattern in angular in dependency injection and in the services.

- ☒ In a sense, if the candidate does 2 times 'new Object()' without this pattern, the candidate will be allocating 2 pieces of memory for the same object. With singleton pattern, if the object exists, it'll be reused.

- ☐ Singleton pattern not used in AngularJS

### Feedback

<http://joelhooks.com/blog/2013/05/01/when-is-a-singleton-not-a-singleton/>

## What would be the output of following code ? \*

1/1

```
function mul(x) {  
  return function(y) {  
    return function(z) {  
      return function(w) {  
        return function(p) {  
          return x * y * z * w * p;  
        };  
      };  
    };  
  };  
}  
console.log(mul(2)(3)(4)(5)(6));
```

- ☐ undefined
- ☒ 720
- ☐ Reference Error
- ☐ Type Error

