

WEB PROGRAMMING

Subject code:23UCS2CC4

UNIT III

WHAT IS ANGULARJS?

The AngularJS is a framework of JavaScript. It can use HTML as a template language and can extend HTML's sentence structure to state an application's components plainly and briefly.

The syntax of AngularJS looks like this:

```
<div ng-app="" "> .....  
</div>
```

We know that div is an html tag, but ng-app is directive of Angular JS, which is used in div tag like an attribute.

BENEFITS OF ANGULARJS

AngularJS has capability to bind data with HTML.

AngularJS provides Single Page Application (SAP).

Angular JS's View is totally written in HTML, Controller is written in JavaScript, and Model is written in AngularJS. It is a MVC framework.

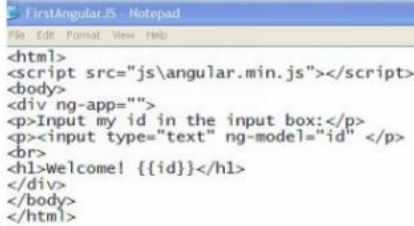
You can achieve more functionality with just a few lines of code.

AngularJS provides the reusability for codes and components.

FIRST ANGULARJS SCRIPT

Example 1.1

Open the Notepad, write codes as following. (Figure 1) <html>

```
<script src="js\angular.min.js"></script>  
<body> <div ng-app=""> <p>Input my id in the input box:</p> <p><input  
type="text" ng-model="id" </p> <br> <h1>Welcome! {{id}}</h1> </div>  
  
</body> </html>
```

(Figure 1) Please save the file with name "FirstAngularJS.html" at myFolder.

Note: make sure to use ".html" extension name.

Double click "FirstAngularJS.html" file, "FirstAngularJS.html" will be run by a browser, and see the output. (Figure 2) (Assume that I entered id "168168".)

Output:

Input my id in the input box:

Welcome! 168168

(Figure 2)

“<script src="js\angular.min.js" ></script>” adds AngularJS framework into current application.

“<div ng-app=""></div>” indicates you can write AngularJS application in here.

“ng-model="id” binds the inputted data from HTML controls (input, select, and textarea) to application data “id”.

{{id}} is synchronized with the “id” in ng-model="id”

{{id}}: displays the value of “id”.

If you want to edit the codes, right click the file “FirstAngularJS.html” > open with > Notepad.

DIRECTIVES

The directives of AngularJS

The AngularJS allows us to extend HTML in very simple way using attributes. The attributes are basically directives. There are different types of directives which can play different roles in Application. They are App Directive, Model Directive, Bind Directive, Init Directive, and Repeat Directive. Let’s discuss one by one in detail.

APP DIRECTIVE

ng-app= “ ”

The app directive defines the area of AngularJS application. The syntax of app directive is ng-app = “ ”; In here the ng is the namespace of AngularJS and app is the application area of Angular JS.

Program

```
<!DOCTYPE html>
<html >
<head>
<title>AngularJS for beginners</title>
<script src="js\angular.min.js"></script>
</head>
```

```
<body>
<div ng-app="" ">
The AngularJSapplication has been started.
</div>
</body>
</html>
```

Output:

The AngularJSapplication has been started.

In head section (<script src = "js\angular.min.js"> </script>), the framework of AngularJS is loaded, which means that we can use AngularJS framework now.

In body section (<div ng-app="" "> </div>), AngularJS application can be written in here.

In last tag (</div>), the AngularJSapplication is ended.

MODEL DIRECTIVE

ng-model = "data"

The model directive is used to bind the inputted value from HTML controls (input, checkbox and select etc.) to application data. The ng-model = "data" is the syntax of model directive. Let's take a program for better understanding.

Program

```
<!DOCTYPE html>
<html >
<head>
<title>AngularJSfor beginners</title> <script src="js\angular.min.js">
</script>
</head>
<body>
<div ng-app=""> <p>User Name: <br> <input type="text" ng-model ="Username"></p>
</div>
</body>
</html>
```

Output:

Username:

“ng-model = “Username”” binds the inputted value from HTML control to “Username”.

Now you cannot see the result because this program is missing the code `<p ng-bind="Username"></p>`.

BIND DIRECTIVE

`<p>ng-bind = “data”</p>`

The bind directive is used to bind the data value to an html element `<p>`; the syntax of bind directive is `<p>ng-bind = “data”</p>`. Let`s take an program for better understanding.

Program

```
<!DOCTYPE html>
<html >
<head>
<title>AngularJSfor beginners</title> <script src="js\angular.min.js">
</script> </head>
<body>
<div ng-app=""> <p>User Name: <br> <input type="text" ng-model =
"Username"></p> <p ng-bind ="Username"></p> </div>
</body>
</html>
```

Open the notepad and paste the above mentioned code with .html extension, and type username “Ray Yao” in the input box.

Output:

User Name:

Ray Yao

“`<p>ng-bind="Username"</p>`” binds the value of “Username” to `<p></p>` tag, and “shows” its value.

In the above program, (`<p ng-bind="Username"></p>`) can update the value of “Username” and writes it to `<p></p>` tag.

`<p>ng-bind="data"</p>` can “output” the value of “data” in specified html element `<p>`.

INIT DIRECTIVE

`ng-init = "data = 'value'"`

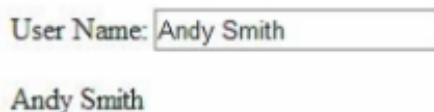
The init directive is used to initialize the data with a value. The syntax of init directive is `ng-init = "data = 'value'"`. Let`s take an program for better understanding.

Program

```
<!DOCTYPE html>
<html >
<head>
<title>AngularJSfor beginners</title> <script src="js\angular.min.js">
</script> </head>
<body>
<div ng-app="" ng-init="Username= 'Andy Smith' "> <p>User Name:<input type="text" ng-
model = "Username"></p> <p ng-bind="Username"></p> </div>
</body>
</html>
```

Open the notepad and paste the above mentioned code with .html extension.

Output:



User Name:
Andy Smith

`ng-init="Username= 'Andy Smith'"` initializes “Username” with a value “Andy Smith”.

In the above program, (`<div ng-app="" ng-init = "Username= 'Andy Smith'">`) initializes the value of User Name. When the page is loaded completely, the value of User Name “Andy Smith” is displayed.

REPEAT DIRECTIVE

`ng-repeat = “variable in array”`

The repeat directive works like a loop. The `ng-repeat` directive repeats to get the value of an array.

Program

```
<html >
<head>
<title>AngularJSfor beginners</title> <script src="js\angular.min.js">
</script> <meta http-equiv="Content-Type" content="text/html; charset=UTF-8"
/> </head>
<body>
<div ng-app="" ng-init = "ColorName = ['Pink', 'Red', 'Green', 'Blue', 'Black', 'White',
'Yellow', 'Gray']"> <p style="color:green; font-weight:bold">Colours Name:</p> <ol>
<li ng-repeat = "x in ColorName"> <p ng-bind="x"></p> </li>
</ol>
</div>
</body>
</html>
```

Open the notepad and paste the above code with .html extension.

Output

Colours Name:

1. Pink
2. Red
3. Green
4. Blue
5. Black
6. White
7. Yellow
8. Gray

ng-repeat = "x in ColorName" repeats to get the value of array "ColorName", and assigns all array values to "x" variable.

<p ng-bind="x"></p> repeats to output the value of "x".

VALID DIRECTIVE

input.ng-valid

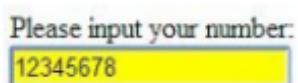
ng-valid directive can change the status of a specified element when an input is valid.

Program

```
<!DOCTYPE html>
<html ng-app="">
<head>
<script src="js\angular.min.js"></script>
<meta charset="utf-8">
<style>
input.ng-valid{ // check number if valid
background-color: yellow;
}
</style>
</head>
<body>
<form>
Please input your number:<br>
<input text="mynumber" ng-model="psw" required>
</form>
</body>
</html>
```

Output

Output:



Please input your number:
12345678

“input.ng-valid” change the input element status when the input is valid.

`<input text="mynumber" ng-model="psw" required>` is used to input numbers.

ng-model= “psw” need a property of “required”, otherwise this element cannot change status.

When you input some valid numbers to text field, the text field background becomes yellow.

CHECK EMAIL ADDRESS

`$error.email`

“`$error.email`” is used to check email address validity; and return error message if wrong email address format is being inputted.

Program 2.7

```
<!DOCTYPE html>
<html ng-app="">
<head>
<script src="js\angular.min.js"></script>
<meta charset="utf-8">
</head>
<body>
<form name="iForm">
Please input your email address: <br><br>
<input type="email" name="eMailAddr" ng-model="text"> <br><br>
<div ng-show="iForm.eMailAddr.$error.email"> Invalid email address!</div>
</form>
```

Output:

Please input your email address:

123456#yahoo.com

Invalid email address!

“`$error.email`” is used to check email address validity; and return error message if wrong email address format is being inputted.

When you input a wrong format email address, you will see an alert: Invalid email address!