CMR
INSTITUTE OF TECHNOLOGY

USN

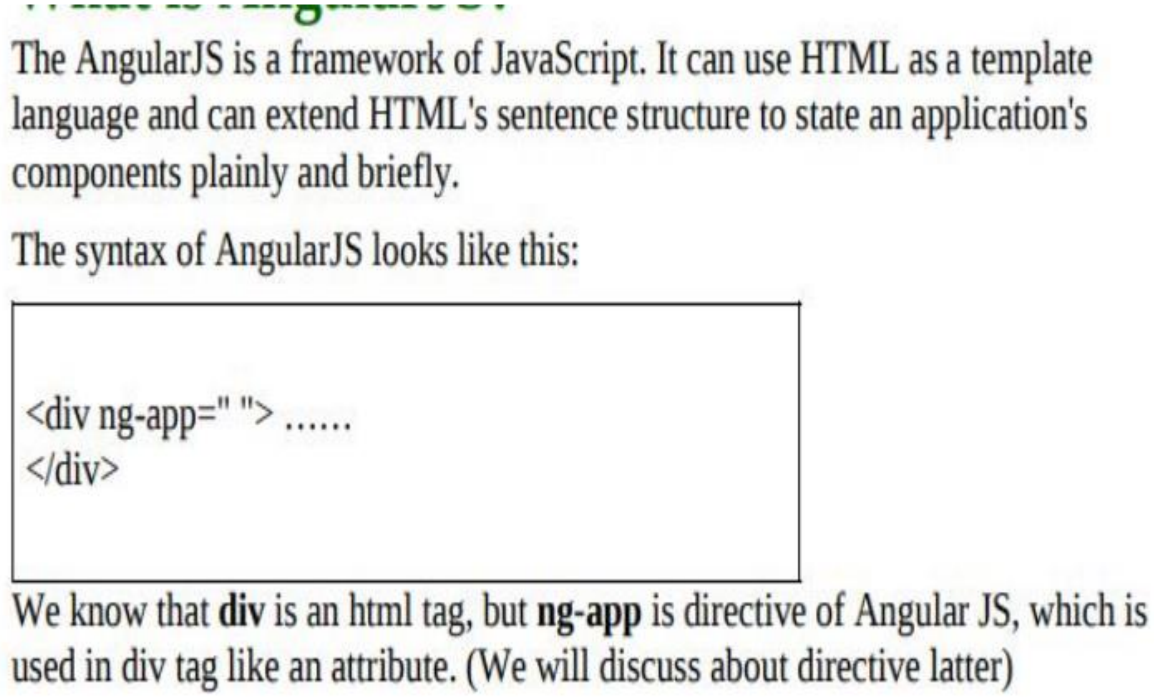
Internal Assessment Test III – September 2023

Sub:	**Web Technologies**							**Sub Code:**	**22MCA24**
Date:	**25/09/2023**	**Duration:**	**90 min's**	**Max Marks:**	**50**	**Sem:**	**II**	**Branch:**	**MCA**

Note : Answer FIVE FULL Questions, choosing ONE full question from each Module

	PART I	MARKS	OBE	
			CO	RBT
1	What is angular JS? Discuss the architecture of angular JS. Explain how to create first angular JS application with example **OR**	[10]	CO5	L2
2	Explain show() and hide() methods in jQuery	[10]	CO1	L2
3	**PART II** Explain how to use child, container and attribute selector in jQuery **OR**	[10]	CO1	L3
4	Discuss getting and setting style, adding and removing classes in jQuery.	[10]	CO1	L3
5	**PART III** Explain following method with appropriate examples 1. bind () 2.unbind() 3.one() **OR**	[10]	CO1	L3
6	What is expression in angular JS explain with example following expressions i)string expression ii)number expression iii)object expression iv) array expression	[10]	CO5	L4
7	**PART IV** what is filter, explain following with example i) uppercase ii)lowercase iii)orderby iv) currency **OR**	[10]	CO5,CO6,CO7	L3
8	What is scope and how to define method in controller explain with example?	[10]	CO5,CO6,CO7	L4
9	**PARTV** What are directives in angular JS? Explain following directive with examples? i)ng-app ii) ng-model and ng-bind iii) ng-init iv) ng-repeat **OR**	[10]	CO5,CO6,CO7	L5
10	Explain following method with appropriate examples i)add() ii) Not() iii) Filter().	[10]	CO1	L3

1. What is angular JS? Discuss the architecture of angular JS. Explain how to create first angular JS application with example

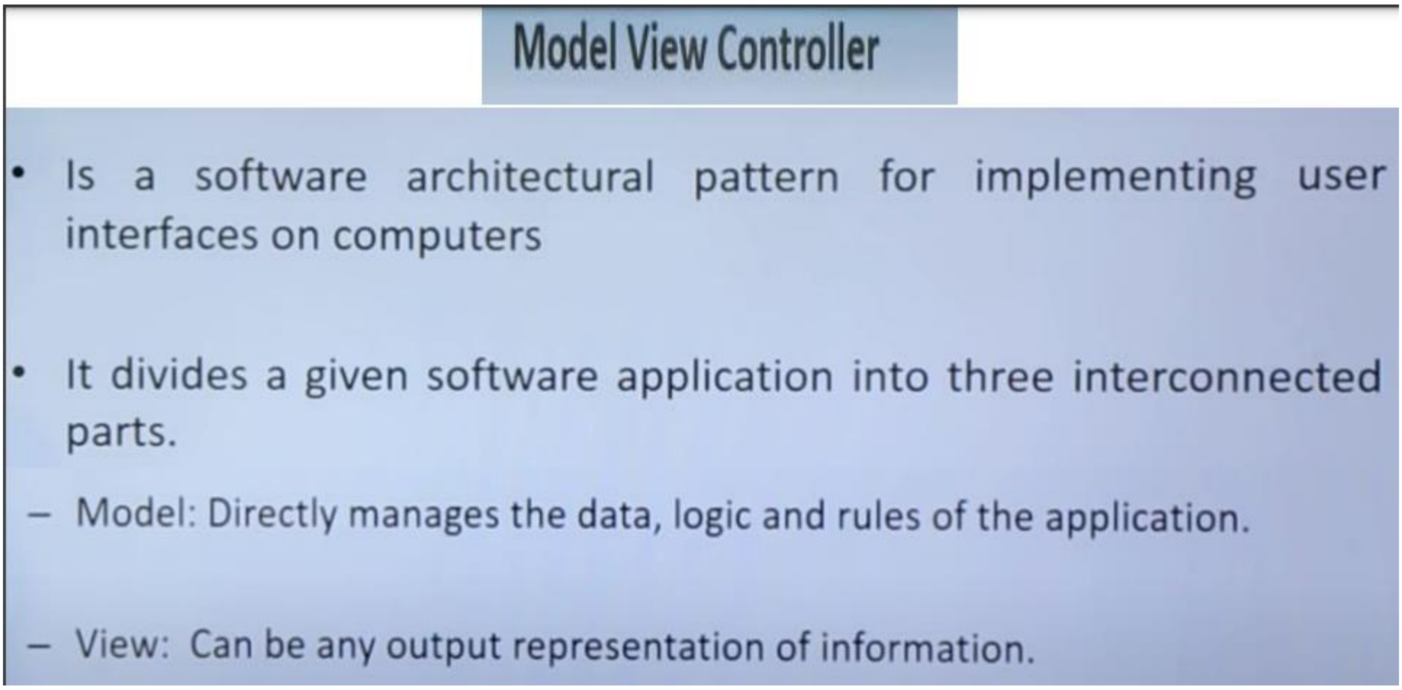


The AngularJS is a framework of JavaScript. It can use HTML as a template language and can extend HTML's sentence structure to state an application's components plainly and briefly.

The syntax of AngularJS looks like this:

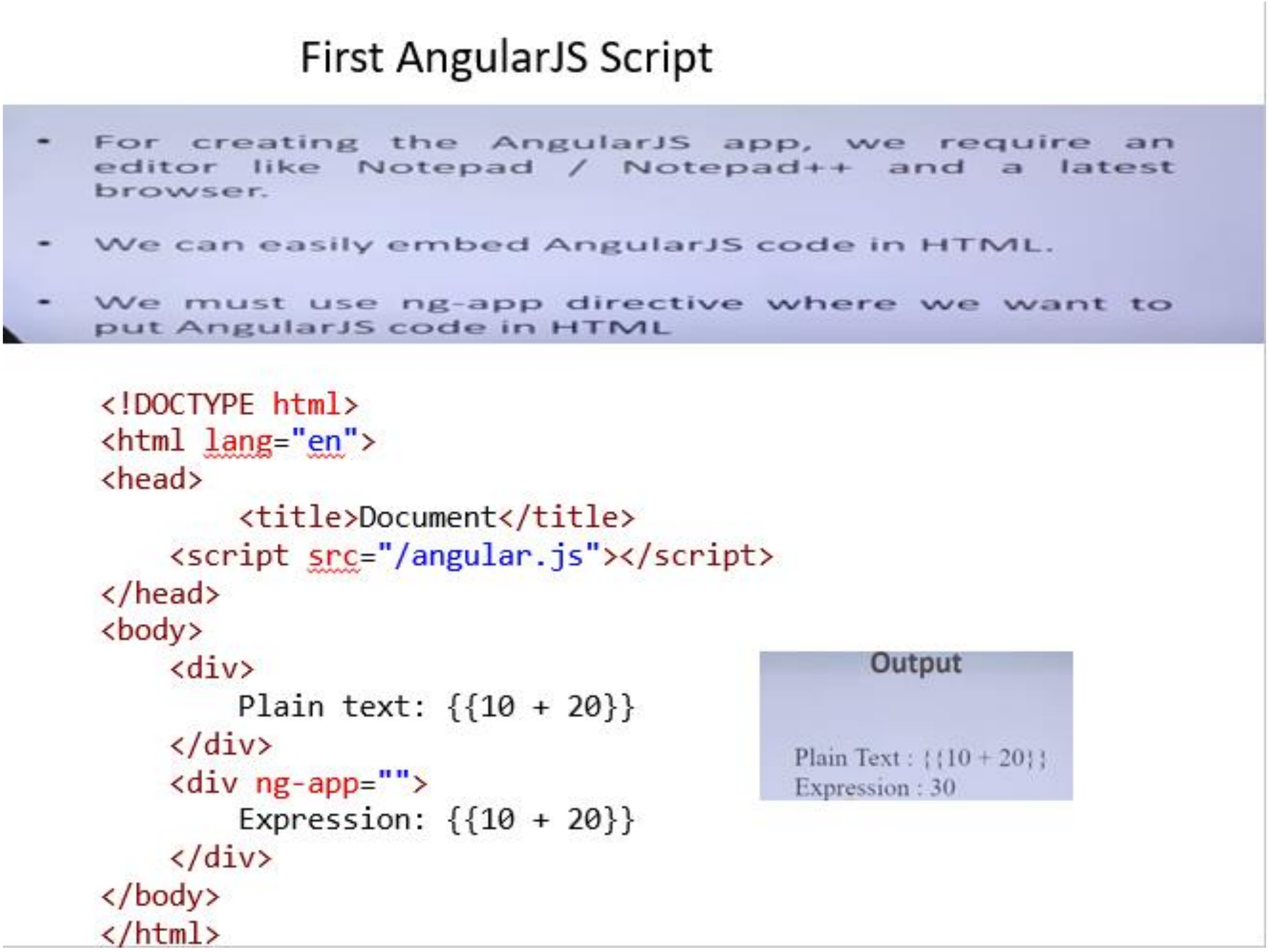
```
<div ng-app=" "> ......
</div>
```

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. (We will discuss about directive latter)



Model View Controller

- Is a software architectural pattern for implementing user interfaces on computers
- It divides a given software application into three interconnected parts.
  - Model: Directly manages the data, logic and rules of the application.
  - View: Can be any output representation of information.



First AngularJS Script

- For creating the AngularJS app, we require an editor like Notepad / Notepad++ and a latest browser.
- We can easily embed AngularJS code in HTML.
- We must use ng-app directive where we want to put AngularJS code in HTML

```
<!DOCTYPE html>
<html lang="en">
<head>
        <title>Document</title>
    <script src="/angular.js"></script>
</head>
<body>
    <div>
        Plain text: {{10 + 20}}
    </div>
    <div ng-app="">
        Expression: {{10 + 20}}
    </div>
</body>
</html>
```

Output

```
Plain Text : {{10 + 20}}
Expression : 30
```

2. Explain show() and hide() methods in jQuery

Chapter 5 - Sprucing up with animations and effects

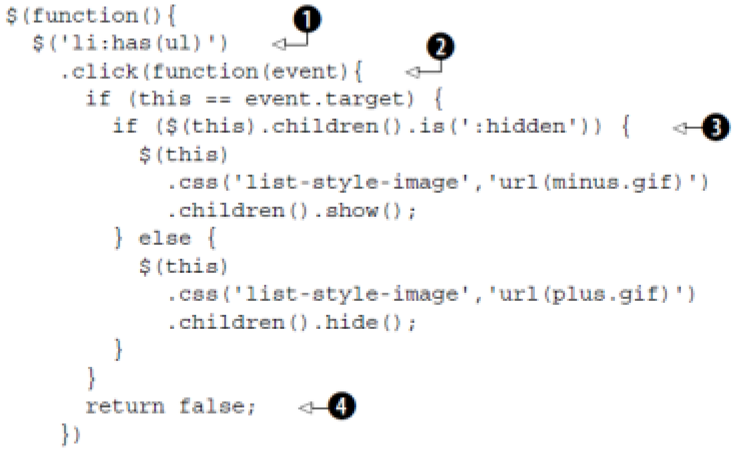
Showing and hiding elements

The commands for showing and hiding elements are :

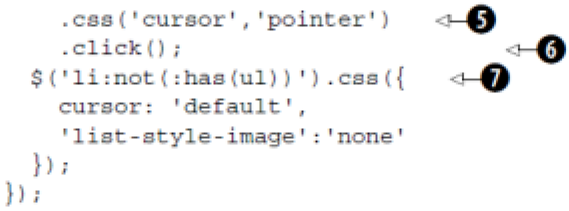
- show() to show the elements in a wrapped set
- hide() to hide them.

If the element starts out without an explicitly declared display value, and we use the jQuery hide() command to hide it, the show() command will remember the original value and restore it to that original display state.

So it's usually a good idea not to use style attributes on the elements we want initially hidden, but to apply the hide() command to them in the page's ready handler. This prevents them from being displayed on the client, and also makes sure everything is in a known initial state and will behave as expected during subsequent hide and show operations.



```
$(function(){
  $('li:has(ul)')                    ❶
    .click(function(event){               ❷
      if (this == event.target) {
        if ($(this).children().is(':hidden')) {     ❸
          $(this)
            .css('list-style-image','url(minus.gif)')
            .children().show();
        } else {
          $(this)
            .css('list-style-image','url(plus.gif)')
            .children().hide();
        }
      }
      return false;     ❹
    })
```



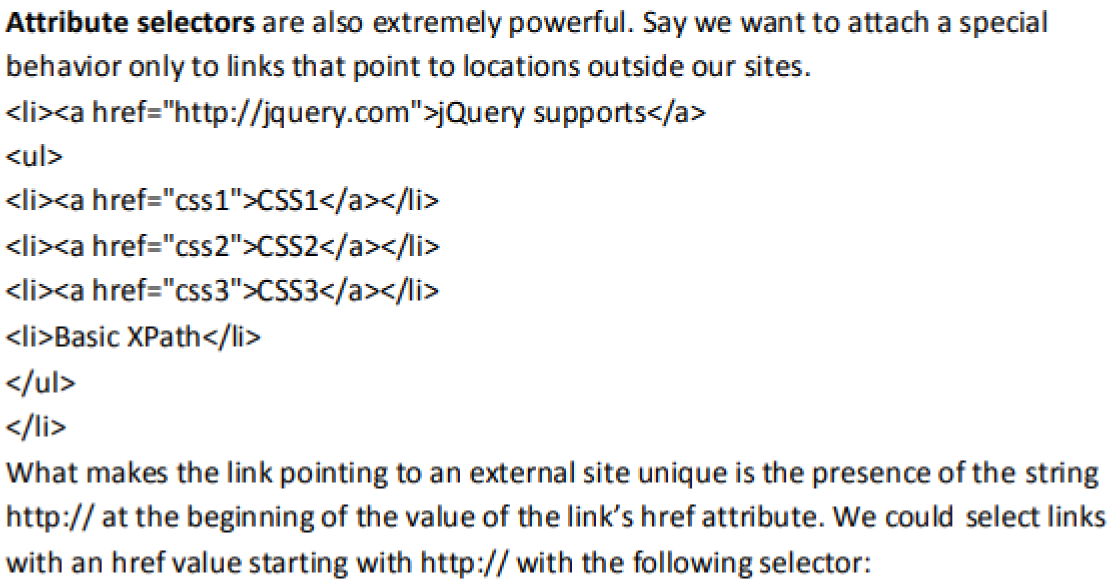
```
      .css('cursor','pointer')      ❺
      .click();                          ❻
    $('li:not(:has(ul))').css({      ❼
      cursor: 'default',
      'list-style-image':'none'
    });
  });
```

3. Explain how to use child, container and attribute selector in jQuery

Child selector

Consider the following HTML fragment:

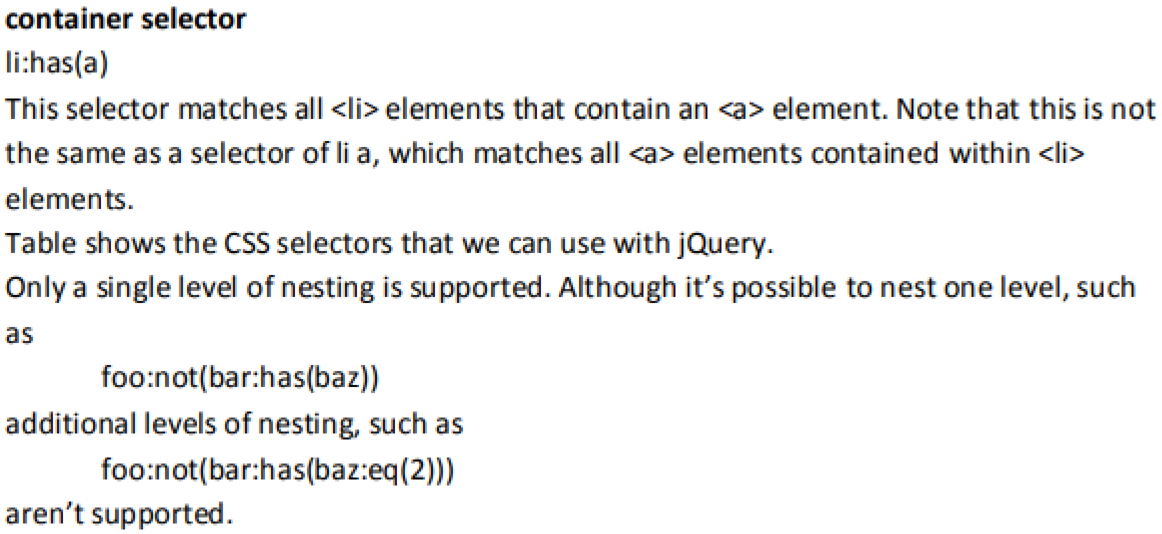
```
<ul class="myList">
        <li><a href="http://jquery.com">jQuery supports</a>
                <ul>
                        <li><a href="css1">CSS1</a></li>
                        <li><a href="css2">CSS2</a></li>
                        <li><a href="css3">CSS3</a></li>
                        <li>Basic XPath</li>
                </ul>


         </li>
         <li>jQuery also supports
                 <ul>
                         <li>Custom selectors</li>
                         <li>Form selectors</li>
                 </ul>
         </li>
 </ul>
```



Attribute selectors are also extremely powerful. Say we want to attach a special behavior only to links that point to locations outside our sites.

```
<li><a href="http://jquery.com">jQuery supports</a>
<ul>
<li><a href="css1">CSS1</a></li>
<li><a href="css2">CSS2</a></li>
<li><a href="css3">CSS3</a></li>
<li>Basic XPath</li>
</ul>
</li>
```

What makes the link pointing to an external site unique is the presence of the string http:// at the beginning of the value of the link's href attribute. We could select links with an href value starting with http:// with the following selector:



container selector

li:has(a)

This selector matches all <li> elements that contain an <a> element. Note that this is not the same as a selector of li a, which matches all <a> elements contained within <li> elements.

Table shows the CSS selectors that we can use with jQuery.

Only a single level of nesting is supported. Although it's possible to nest one level, such as

```
foo:not(bar:has(baz))
```

additional levels of nesting, such as

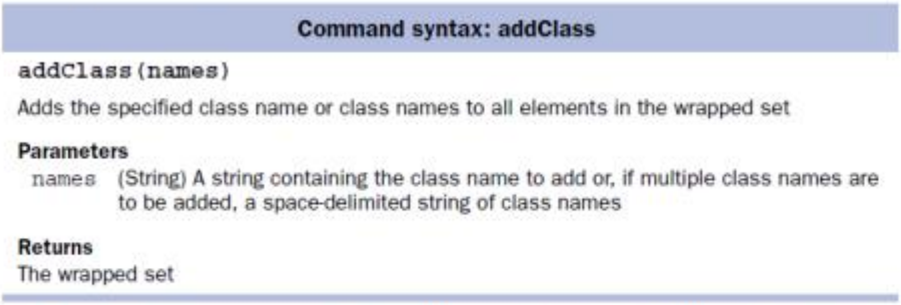
```
foo:not(bar:has(baz:eq(2)))
```

aren't supported.

4. Discuss getting and setting style, adding and removing classes in jQuery.

I. Adding and removing class names

Adding class names to all the elements of a matched set is an easy operation with the following addClass() command:



Command syntax: addClass

`addClass(names)`

Adds the specified class name or class names to all elements in the wrapped set

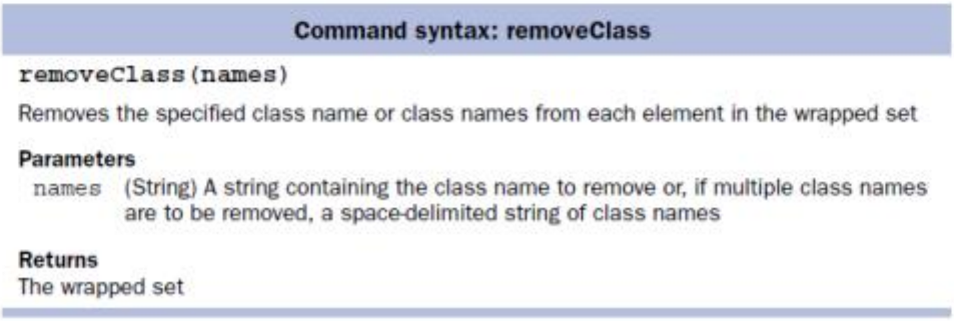
Parameters

`names` (String) A string containing the class name to add or, if multiple class names are to be added, a space-delimited string of class names

Returns

The wrapped set

Removing class names is as straightforward with the following removeClass() command:



Command syntax: removeClass

`removeClass(names)`

Removes the specified class name or class names from each element in the wrapped set

Parameters

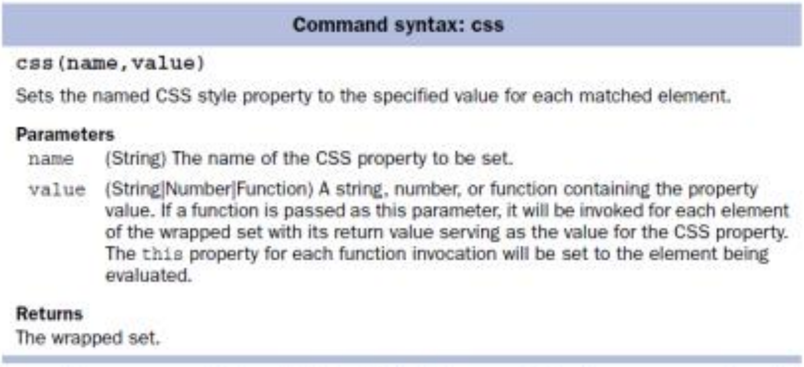
`names` (String) A string containing the class name to remove or, if multiple class names are to be removed, a space-delimited string of class names

Returns

The wrapped set

II. Getting and setting styles

The css() method works similarly to the attr() method, allowing us to set an individual CSS property by specifying its name and value, or a series of elements by passing in an object.



Command syntax: css

`css(name,value)`

Sets the named CSS style property to the specified value for each matched element.

Parameters

`name` (String) The name of the CSS property to be set.

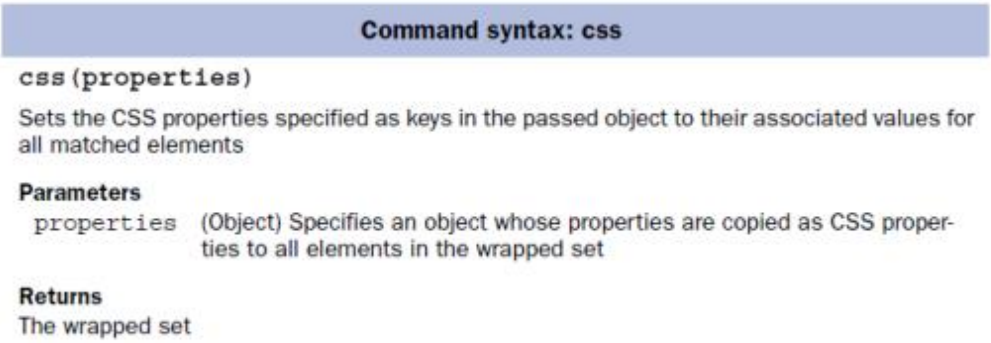
`value` (String|Number|Function) A string, number, or function containing the property value. If a function is passed as this parameter, it will be invoked for each element of the wrapped set with its return value serving as the value for the CSS property. The `this` property for each function invocation will be set to the element being evaluated.

Returns

The wrapped set.

Example: expand the width of all elements in the wrapped set by 20 pixels as follows:

```
$("div.expandable").css("width",function() {
        return $(this).width() + 20 + "px";
});
```



Command syntax: css

`css(properties)`

Sets the CSS properties specified as keys in the passed object to their associated values for all matched elements

Parameters

`properties` (Object) Specifies an object whose properties are copied as CSS properties to all elements in the wrapped set

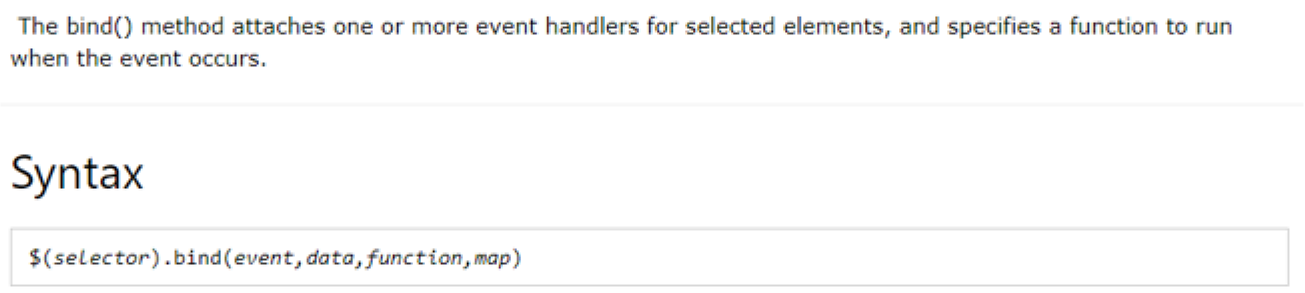
Returns

The wrapped set

5. Explain following method with appropriate examples

bind () 2.unbind() 3.one()

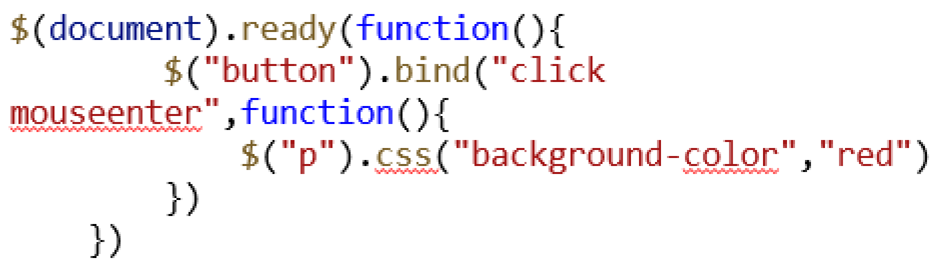
Binding event handlers using jQuery



The bind() method attaches one or more event handlers for selected elements, and specifies a function to run when the event occurs.

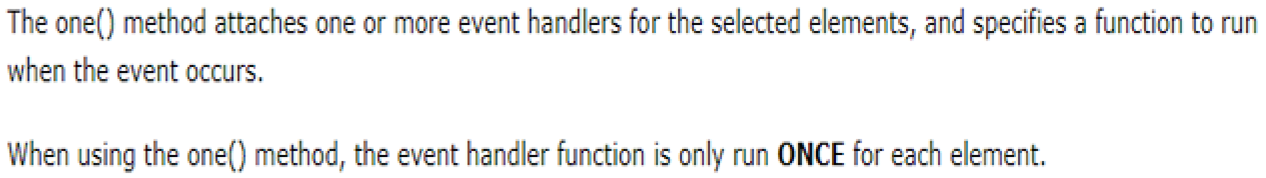
Syntax

```
$(selector).bind(event,data,function,map)
```



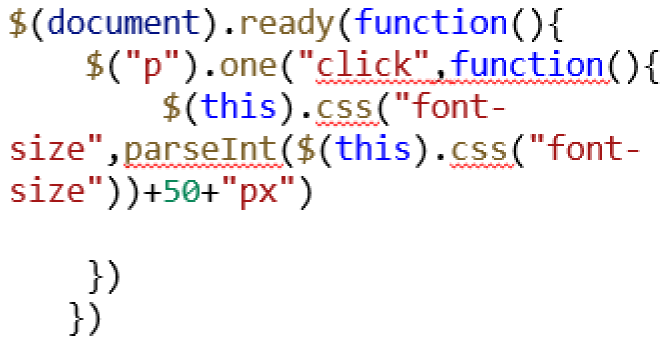
```
$(document).ready(function(){
        $("button").bind("click
mouseenter",function(){
            $("p").css("background-color","red")
        })
    })
```

jQuery one() Method

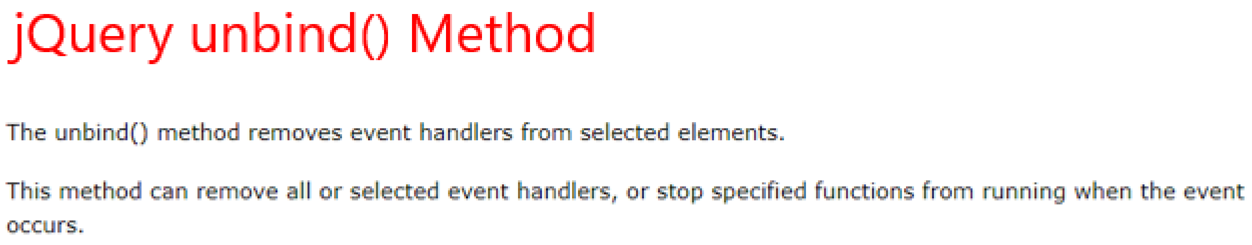


The one() method attaches one or more event handlers for the selected elements, and specifies a function to run when the event occurs.

When using the one() method, the event handler function is only run **ONCE** for each element.



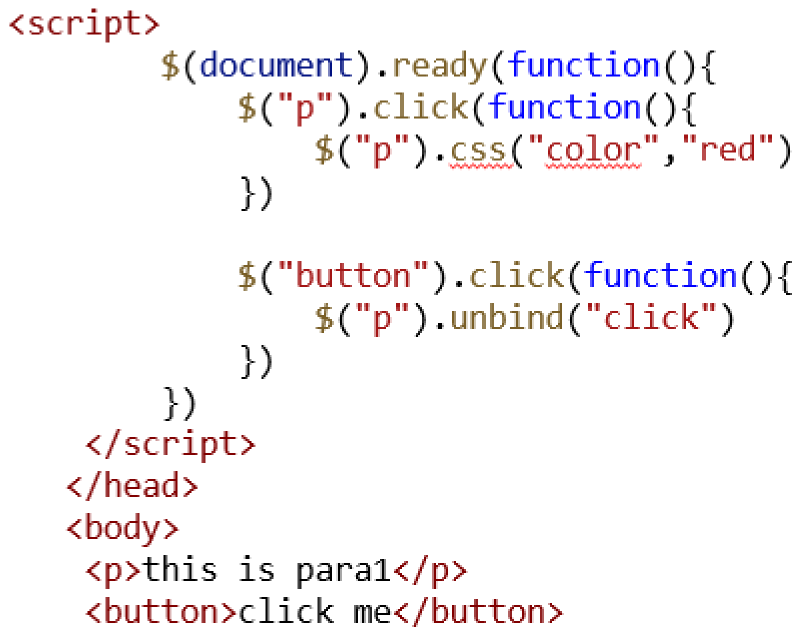
```
$(document).ready(function(){
    $("p").one("click",function(){
        $(this).css("font-
size",parseInt($(this).css("font-
size"))+50+"px")

    })
   })
```



jQuery unbind() Method

The unbind() method removes event handlers from selected elements.

This method can remove all or selected event handlers, or stop specified functions from running when the event occurs.



```
<script>
    $(document).ready(function(){
      $("p").click(function(){
        $("p").css("color","red")
      })

      $("button").click(function(){
        $("p").unbind("click")
      })
    })
  </script>
 </head>
 <body>
  <p>this is para1</p>
  <button>click me</button>
```

6. What is expression in angular JS explain with example following expressions
i)string expression ii)number expression iii)object expression iv) array expression

String Expression

We know that string is collection of characters. In AngularJSthe string expression looks like this.

```
<element> {{First String + Second String}}
</element>
```

Example 6.1

```
<!DOCTYPE html>
<html >
<head>
   <title>AngularJS for beginners</title> <script src="js/angular.min.js">
</script> </head>
<body>
<h4>Combine Two String Using String Expression</h4> <div ng-app="" >
First String &nbsp;&nbsp;&nbsp;: <input type="text" ng-
model="firstString"/><br><br> Second String: <input ng-
model="secondString"/><br><br> Resulting String:<p
style="color:blue;font-weight:bold;">{{firstString +" "+secondString}}
</p> </div>
</body>
</html>
```

Number Expression

In AngularJSyou can perform different mathematic operation by using Number Expression.

```
<element> {{First Number + Second Number}}
</element>
```

Example 6.2

```
<!DOCTYPE html>
<html >
<head>
   <title>AngularJS for beginners</title> <script src="js/angular.min.js"> </script> </head>
<body>
<h4>Multiply Two Number Using Number Expression</h4> <div ng-app="" ng-init="firstNumber=9;secondNumber=6"> First Number &nbsp;&nbsp;&nbsp;: <input type="number" ng-model="firstNumber"/> <br><br> Second Number: <input type="number" ng-model="secondNumber"/><br><br> Result:<p style="color:blue;font-weight:bold;">{{firstNumber * secondNumber}}</p> </div>
</body>
</html>
```

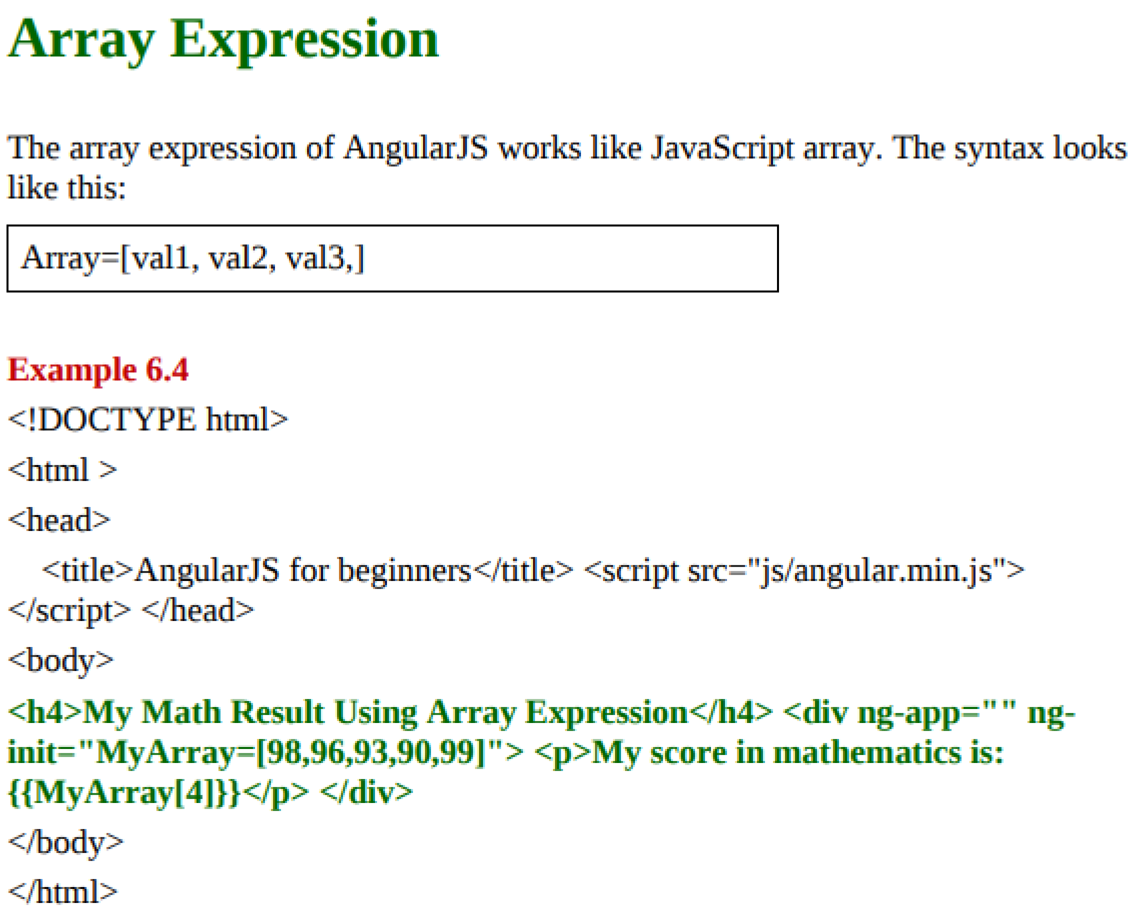
Object Expression

AngularJSobject works like a JavaScript object. The syntax looks like this:

```
object = {property: value}
```

Example 6.3

```
<html >
<script src= "js\angular.min.js"></script>
<body>
<h4>Object Expression</h4> <div ng-app=""  ng-init="EmployeeObject = {Emp_name: 'Jay Smith',    Emp_Month: 'June.15 2015',  Emp_salary: '$8000'}"> <p>Employee Name :  {{EmployeeObject.Emp_name}}</p> <p>Salary's Month:  {{EmployeeObject.Emp_Month}}</p> <p>Employee Salary:  {{EmployeeObject.Emp_salary}}</p> </div>
</body>
</html>
```



Array Expression

The array expression of AngularJS works like JavaScript array. The syntax looks like this:

```
Array=[val1, val2, val3,]
```

Example 6.4

```
<!DOCTYPE html>
<html >
<head>
   <title>AngularJS for beginners</title> <script src="js/angular.min.js">
</script> </head>
<body>
<h4>My Math Result Using Array Expression</h4> <div ng-app="" ng-
init="MyArray=[98,96,93,90,99]"> <p>My score in mathematics is:
{{MyArray[4]}}</p> </div>
</body>
</html>
```

7. what is filter, explain following with example

   i) uppercase ii)lowercase iii)orderby iv) currency

Uppercase filter

```
Value | uppercase
```

The uppercase filter changes the text to upper case. Suppose a user writes a text in lower case (e.g. ray) or title case (e.g. Ray) or in mixed case (e.g. rAy or RaY or rAY etc.), and you want the upper case result, then you will have to use upper case filter.

Example 3.1

```
<!DOCTYPE html>
<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>
<h3>Using Upper Case Filter</h3> <div ng-app=""  ng-init="Username=
'ray' "> <p>User Name: <input type="text" ng-model = "Username"></p>
<p style="color:red" ng-bind="Username | uppercase"></p> </div>
</body>
</html>
```

Lowercase filter

```
Value | lowercase
```

The lowercase filter changes the text to lower case. Suppose a user writes a text in upper case (e.g. RAY YAO) or title case (e.g. Ray Yao) or in mixed case (e.g. rAy or RaY or rAY etc.), and you want the lower case result, then you will have to use lower case filter.

Example 3.2

```
<html >
<head>
   <title>AngularJSfor beginners</title> <script src="js\angular.min.js">
</script> </head>
<body>
<h3>Using Lower Case Filter</h3> <div ng-app="" ng-init="Username=
'Ray YAO' "> <p>User Name: <input type="text" ng-model="Username">
</p> <p style="color:red" ng-bind="Username | lowercase"></p> </div>
</body>
</html>
```

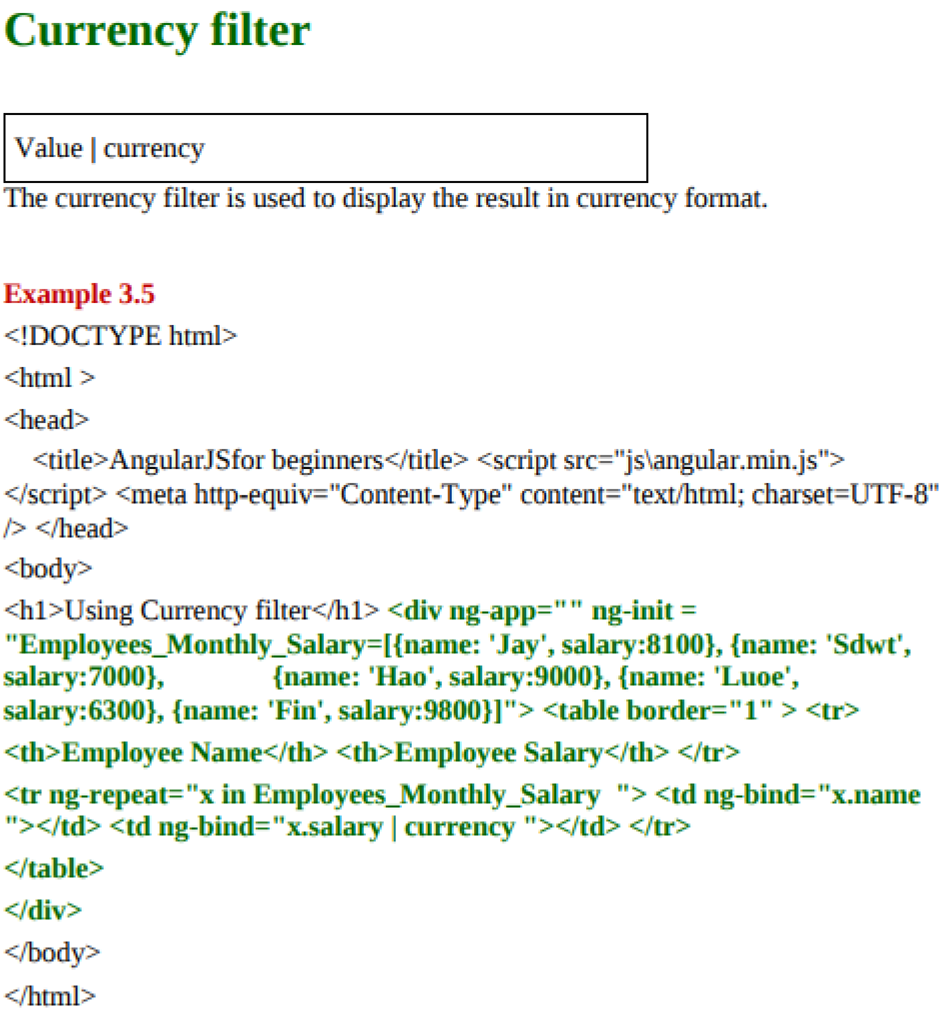
OrderBy filer is used to display values in ascending order or descending order.
The syntax of "orderBy" looks like this:

```
Value | orderBy: 'value'    //for ascending order
Value | orderBy: '-value'   //for descending order
```

Let`s take an example for better understanding.

Example 3.3

```
<!DOCTYPE html>
<html >
<head>
   <title>AngularJSfor beginners</title> <script src="js\angular.min.js">
</script> </head>
<body>
<h1>Using OrderBy filter</h1> <div ng-app="" ng-init="StudentsResult=
[{name: 'Tienq', marks:81},          {name: 'Svbrf', marks:70},
{name: 'Yaito', marks:90},           {name: 'Pewfn', marks:63}, {name:
'Riet', marks:98}]"> <table border="1" > <tr>
<th>Student Name</th> <th>Mathematics' Result</th> </tr>
<tr ng-repeat="x in StudentsResult | orderBy:'-marks' "> <td ng-
bind="x.name "></td> <td ng-bind="x.marks "></td> </tr>
</table>
</div>
</body>
</html>
```



Currency filter

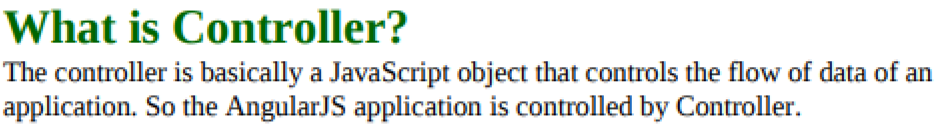
```
Value | currency
```

The currency filter is used to display the result in currency format.

Example 3.5

```
<!DOCTYPE html>
<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>
<h1>Using Currency filter</h1> <div ng-app="" ng-init =
"Employees_Monthly_Salary=[{name: 'Jay', salary:8100}, {name: 'Sdwt',
salary:7000},            {name: 'Hao', salary:9000}, {name: 'Luoe',
salary:6300}, {name: 'Fin', salary:9800}]"> <table border="1" > <tr>
<th>Employee Name</th> <th>Employee Salary</th> </tr>
<tr ng-repeat="x in Employees_Monthly_Salary  "> <td ng-bind="x.name
"></td> <td ng-bind="x.salary | currency "></td> </tr>
</table>
</div>
</body>
</html>
```

8. What is scope and how to define method in controller explain with example?



What is Controller?

The controller is basically a JavaScript object that controls the flow of data of an application. So the AngularJS application is controlled by Controller.

How to define Controller?

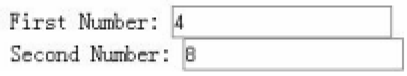
The **ng-controller** directive is used to define the Controller. We know that Controller is a JavaScript object which contains JavaScript function and properties. The syntax of Controller is as following:

```
<div ng-app=""  ng-controller="controllerName">
```

Example 7.1

```
<html>
<script src= "js\angular.min.js"></script> <body>
<div ng-app="Calculation" ng-controller="myController"> First Number:
<input type="number" ng-model="firstNumber"><br> Second Number:
<input type="number" ng-model="secondNumber"><br> <br>
Sum: {{firstNumber + secondNumber}}
</div>
<script>
var app = angular.module('Calculation', [ ]); app.controller('myController',
function($scope) {
   $scope.firstNumber = 4; $scope.secondNumber = 8; });
</script>
</body>
</html>
```

Output:



```
First Number: 4
Second Number: 8
```

What is Scope?

Scope is a JavaScript object which contains model data.

```
function ($scope) {  }
```

$scope is a parameter of JavaScript function which is called by a controller. Let`s take an example for understanding.

Example 7.2

```
<script>
function ($scope) {
   $scope.firstNumber = 23;
   $scope.secondNumber = 63;
}
```

</script> **Explanation:** In above example, the **$scope** is the parameter of the function ($scope) { }. $scope is an object in AngularJS.

$scope.**firstNumber and **$scope.secondNumber are models used in HTML. Model data is accessed by the $scope object. We assign values to the model with following formula: "$scope.property = value".

9. What are directives in angular JS? Explain following directive with examples?
   i)ng-app ii) ng-model and ng-bind iii) ng-init iv) ng-repeat

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.

Example 2.1

```
<!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>
```

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 an example for better understanding.

Example 2.2

```
<!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>
```

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 example for better understanding.

Example 2.3

```
<!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.

Init Directive

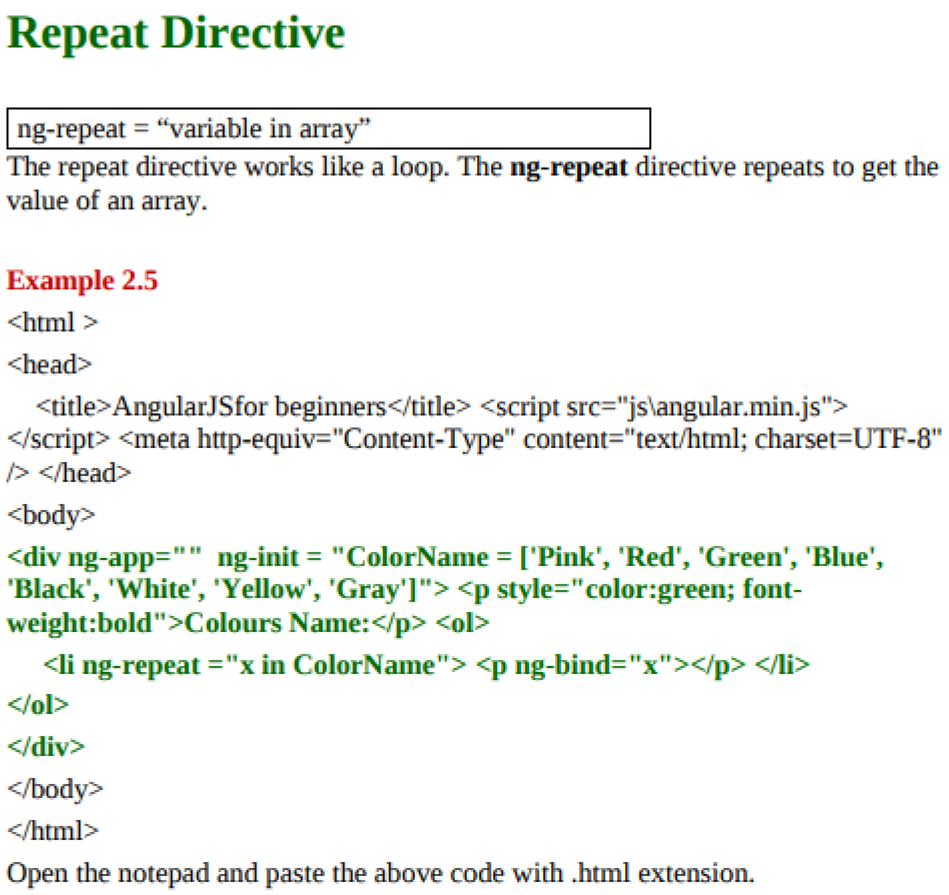
```
ng-init = "data = ‘value’”
```

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 example for better understanding.

Example 2.4

```
<!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.



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.

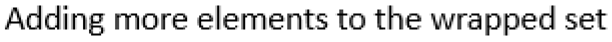
Example 2.5

```
<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>
```

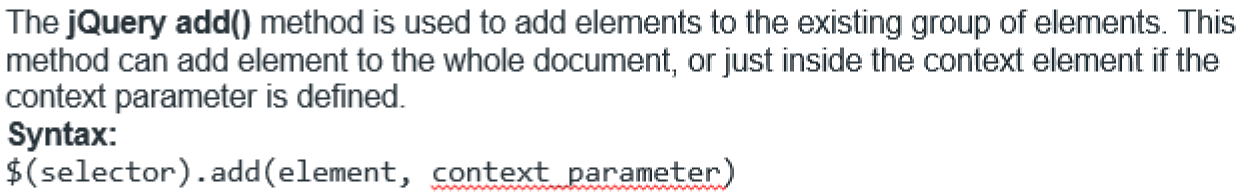
Open the notepad and paste the above code with .html extension.

10. Explain following method with appropriate examples

i)add() ii) Not() iii) Filter().



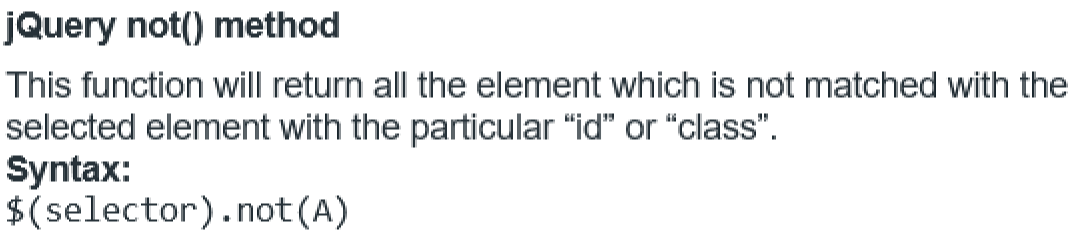
Adding more elements to the wrapped set



The **jQuery add()** method is used to add elements to the existing group of elements. This method can add element to the whole document, or just inside the context element if the context parameter is defined.

Syntax:

```
$(selector).add(element, context_parameter)
```

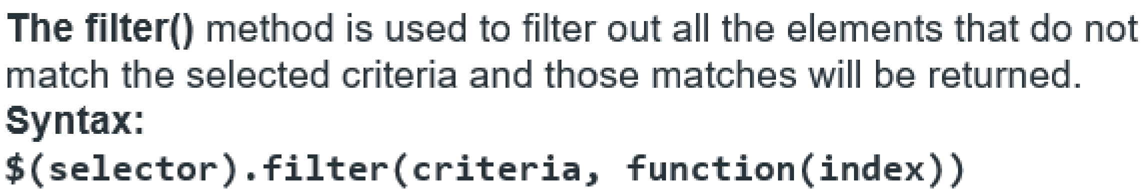


jQuery not() method

This function will return all the element which is not matched with the selected element with the particular "id" or "class".

Syntax:

```
$(selector).not(A)
```



The filter() method is used to filter out all the elements that do not match the selected criteria and those matches will be returned.

Syntax:

```
$(selector).filter(criteria, function(index))
```