

<script>



JavaScript

</script>

*JQuery*

## Jquery Part 1

# Topics will be covered...



1

- What is jQuery?

2

- How to use jQuery?

3

- Utilities and Features

4

- Problem: Using Multiple Libraries

# What is jQuery?



***jQuery simplifies HTML document traversing, event handling, animating, and Ajax interactions for Rapid Web Development***

# What is jQuery?(Cont.)



- ❑ jQuery is a fast and concise JavaScript Library created by John Resig in 2006 with a nice motto: **Write less, do more.**
- ❑ Most popular JavaScript library in use today (Google/Microsoft, IBM and Netflix).
- ❑ Current version 1.11.2 (December 2014).
- ❑ The core library size is small.
- ❑ Includes Effects and Animation in the core library.
- ❑ jQuery UI Extension provides very rich controls.

# What is jQuery?(Cont.)



❑ jQuery is a JavaScript library designed to **simplify various tasks by writing less code:**

- HTML/DOM manipulation
- CSS manipulation
- HTML event methods
- Effects and animations
- AJAX
- Utilities

❑ It's:

- Lightweight
- Cross Browser Support
- Latest Technology

# Why jQuery?



- ☐ Rich Internet Applications (RIA)
- ☐ Dynamic HTML (DHTML)
- ☐ Friendly and Elegant API's
- ☐ DOM traversing is very easy especially for complex search criteria
- ☐ The core library size is very small
- ☐ **jQuery UI** Extension provides very rich controls
- ☐ Cross Browser Support

# Things jQuery Provides...



- ☐ Select DOM (Document Object Model) elements on a page – one element or a group of them
- ☐ Set properties of DOM elements, in groups (“Find something, do something with it”)
- ☐ Creates, deletes, shows, hides DOM elements
- ☐ Defines event behavior on a page (click, mouse movement, dynamic styles, animations, dynamic content)
- ☐ Reduces browser inconsistencies
- ☐ **Plugins** are available to cover all needs

# Using jQuery



❑ You can download latest version from: [www.jquery.com](http://www.jquery.com)

❑ Include it in the script:

```
<script type="text/javascript" src="jquery-1.11.2.min.js"></script>
```

❑ You can also use **jQuery CDN** (Content Delivery Network), both Google and Microsoft host it:

**//Google CDN**

```
<script src="http://ajax.googleapis.com/ajax/libs/jquery/1.11.2/jquery.min.js"></script>
```

**//Microsoft CDN**

```
<script src="http://ajax.aspnetcdn.com/ajax/jQuery/jquery-1.11.2.min.js"></script>
```

**//Jquery CDN**

```
<script type="text/javascript" src="http://code.jquery.com/jquery-1.11.2.min.js"></script>
```



# Using jQuery (Cont.)



❑ jQuery is provided in main 2 JS files:

- `jquery-1.x.x.min.js`, that contains core functions.
- `jQuery-ui-1.x.x.js`, that contains:
  - Interactions (Draggable, Droppable, Resizable, Sortable,...).
  - Widgets (Accordion, Menu, Dialog, tabs,...).
  - Effects (Hide, Show, animate,...).
- `jQuery-ui-1.x.x.custom.js`

❑ There also a lot of JQuery plugins for more widgets and features.

# Jquery Syntax



- ❑ Basic JQuery syntax is: `$(selector).action()`
  - A \$ sign to define/access jQuery
  - A (selector) to "query (or find)" HTML elements (CSS Rule).
  - A jQuery action() to be performed on the element(s)

## ❑ Example:

```
$(".test").hide(); //hides all elements with class="test"  
$("p").hide(); //hides all <p> elements.  
$("#test").hide(); //hides the element with id="test".
```

# Jquery Syntax (Cont.)



## ❑ The Document Ready Event:

- It is good practice to wait for the document to be fully loaded and ready before working with it.
- To ensure that our events will start working we should use `$(document).ready()` Event.
- Everything inside it will load **as soon as the DOM is loaded** and before the page contents are loaded.

```
$(document).ready(function()  
{  
    // do stuff when DOM is ready  
});
```

- Another shorter method for the document ready event:

```
$(function(){  
    // do stuff when DOM is ready  
});
```

# Utilities and Features



- ☐ **Selectors**

- ☐ **Attributes**

- ☐ **Traversing**

- ☐ **CSS**

- ☐ **DOM**

- ☐ **Events**

- ☐ **UI**

- ☐ **Ajax**

# jQuery Selector



- ❑ jQuery can help you find elements based on their ID, classes, types, attributes, values of attributes and much, much more.
- ❑ **A jQuery Selector** is a function which makes use of expressions to find out matching elements from a DOM based on the given criteria.
- ❑ The jQuery library harnesses the power of **Cascading Style Sheets (CSS) selectors** to let us quickly and easily access elements or groups of elements in the Document Object Model (DOM).
- ❑ All type of selectors available in jQuery, always start with the dollar sign and parentheses: **\$()**.

# jQuery Selector(Cont.)



## □ \$():

- The factory function \$() makes use of following building blocks while selecting elements in a given document:

jQuery	Description
<b>Name</b>	Selects all elements which match with the given element <b>Name</b> .
<b>#ID</b>	Selects a single element which matches with the given <b>ID</b>
<b>.Class</b>	Selects all elements which match with the given <b>Class</b> .
<b>Universal (*)</b>	Selects <b>all</b> elements available in a DOM.
<b>Multiple Elements E, F, G</b>	Selects the combined results of all the specified selectors E, F or G.
<b>Othe Selectors</b>	Jquery Selector Ref: <a href="http://www.w3schools.com/jquery/jquery_ref_selectors.asp">http://www.w3schools.com/jquery/jquery_ref_selectors.asp</a>

# jQuery Selector(Cont.)



## □ Examples:

Selector	Result
<code>\$('*')</code>	all elements in the document.
<code>\$('p &gt; *')</code>	all elements that are children of a paragraph element.
<code>\$('#specialID')</code>	the element with id="specialID".
<code>\$('.specialClass')</code>	all the elements that have the class of <i>specialClass</i>
<code>\$('p:empty')</code>	all elements matched by <p> that have no children.
<code>\$('p a.specialClass')</code>	matches links with a class of <i>specialClass</i> declared within <p> elements.
<code>\$(':input')</code>	All Input Elements.
<code>\$(':radio')</code>	Inputs of “radio” type. Equals to: <code>\$('input[type=radio]')</code>
<code>\$('li:even')</code>	all <li> that have an even index value.
<code>\$('p:first')</code>	Selects the first <p> element.
<code>\$('p:eq(3)')</code>	Selects the third<p> element

# jQuery Selector (Cont.)



## ❑ CSS Attribute Selector:

Example	Result
<code>\$('[attr="val"]')</code>	attr equals val
<code>\$('[attr!="val"]')</code>	attr does not equal val
<code>\$('[attr~="val"]')</code>	attr containing the specific word “val”
<code>\$('[attr^="val"]')</code>	attr begins with val
<code>\$('[attr\$="val"]')</code>	attr ends with val
<code>\$('[attr*="val"]')</code>	attr contains val



# jQuery Selector (Cont.)



## ❑ jQuery Traversing – Filtering:

- You can use some filtering methods to filter the search elements.
- Most basic filtering methods are `first()`, `last()`, `eq()`, `filter()` and `not()`.
- Examples:

Example	Result
<code>\$("div p").first()</code>	selects the first <code>&lt;p&gt;</code> element inside the first <code>&lt;div&gt;</code> .
<code>\$("div p").last()</code>	selects the last <code>&lt;p&gt;</code> element inside the last <code>&lt;div&gt;</code> .
<code>\$("p").eq(1)</code>	selects the second <code>&lt;p&gt;</code> element (index starts at 0).
<code>\$("p").next()</code>	Return the next sibling element of each <code>&lt;p&gt;</code>
<code>\$("p").nextAll()</code>	Return all next sibling element of each <code>&lt;p&gt;</code>
<code>\$("li.start").siblings()</code>	Return all sibling elements of each <code>&lt;li&gt;</code> element
<code>\$("p.start").nextUntil("p.stop")</code>	Return all sibling elements between two <code>&lt;p&gt;</code> elements with class name "start" and "stop"
<code>\$('span:contains("A")')</code>	select spans that contains "A" in its content (based on its text content).

# jQuery Selector (Cont.)



## ❑ jQuery Traversing – Filtering (Cont.)

Example	Result
<code>\$('p').filter('.intro')</code>	returns all <p> elements with class name "intro". The same as: <code>\$('p.intro')</code>
<code>\$('p').not('.intro')</code>	returns all <p> elements that do not have class name "intro".
Other Traversing Methods	Jquery Traversing Methods Reference: <a href="http://www.w3schools.com/jquery/jquery_ref_traversing.asp">http://www.w3schools.com/jquery/jquery_ref_traversing.asp</a>

# jQuery Events



## ❑ JQuery Syntax For Event Methods:

- In jQuery, most DOM events have an equivalent jQuery method.
- To handle event to any selector in JQuery:

```
$(selector).event(function()  
{  
    // things you need to execute to handle the event  
});
```

- Example: (handles click event for all paragraphs):

```
$(document).ready(function () {  
    $("p").click(function ()  
    {  
        $(this).hide();  
    });  
});
```

# jQuery Events (Cont.)



## □ JQuery shorthand Events:

- .submit()
- .mouseover()
- .keydown()
- .change()
- .mouseout()
- .keypress()
- .focus()
- .mouseenter()
- .keyup()
- .focusin()
- .mouseleave()
- .scroll()
- .blur()
- .mousemove()
- .resize()
- .focusout()
- .dblclick()
- .error()

- JQuery Event Ref: [http://www.w3schools.com/jquery/jquery\\_ref\\_events.asp](http://www.w3schools.com/jquery/jquery_ref_events.asp)

# jQuery Callback Functions



- ❑ A callback function is executed after the current effect is finished.
- ❑ Typical syntax:

```
$(selector).hide(speed,callback);
```

- ❑ Example:

```
$("#button").click(function(){  
    $("#p").hide("slow",function(){  
        alert("The paragraph is now hidden");  
    });  
});
```

# Methods for manipulating HTML elements and attributes



- ❑ The basic components we can manipulate when it comes to DOM elements are the **properties** and **attributes** assigned to those elements.
- ❑ Most of these attributes are available through JavaScript as DOM node properties.

# Methods for manipulating HTML elements and attributes (Cont.)



## ❑ **attr():**

- ❑ Used to either **get** or **set** the value of an attribute from the first element in the matched set or from all matched elements.

```

```

- Get an Attribute value → `attr("attName")`

```
var imgTitle = $("#myimg").attr("title");
```

- Set an Attribute value: → `attr(name, value)`

```
$("#myimg").attr("src", "/images/jquery.jpg");
```

# Methods for manipulating HTML elements and attributes (Cont.)



## ❑ Attribute Methods that handles **html**

Method	Description
<b>html( )</b>	Get the html contents (innerHTML) of the first matched element.
<b>html( val )</b>	Set the html contents of every matched element.
<b>text( )</b>	Get the combined text contents of all matched elements.
<b>text( val )</b>	Set the text contents of all matched elements.
<b>val( )</b>	Get the input value of the first matched element.
<b>val( val )</b>	Set the value attribute of every matched element if it is called on <input> but if it is called on <select> with the passed <option> value then passed option would be selected, if it is called on check box or radio box then all the matching check box and radiobox would be checked.



# Methods for manipulating HTML elements and attributes (Cont.)



## ❑ Attribute Methods that handles **Style**

Method	Description
<b>.addClass(class)</b>	apply defined style sheets onto all the matched elements. You can specify multiple classes separated by space.
<b>.removeClass(class)</b>	Removes all or the specified class(es) from the set of matched elements.
<b>.toggleClass(class)</b>	Adds the specified class if it is not present, removes the specified class if it is present.
<b>.hasClass( class )</b>	Returns true if the specified class is present on at least one of the set of matched elements.
<b>.css(css attribute)</b> <b>.css(css attribute, css value)</b>	Set CSS rules for the element. Example: <pre>var pColor=\$("p").css("background-color"); \$("p").css("background-color", "Yellow"); \$("p").css({ "background-color":"yellow", "color":"black" });</pre>

# jQuery Method Chaining



- ❑ With jQuery, you can chain together actions/methods.
- ❑ Chaining allows us to run multiple jQuery methods (on the same element) within a single statement.
- ❑ Example:

```
$("#p1").css("color","red").slideUp(2000).slideDown(2000);
```

# jQuery Ajax methods



Method	Description
<a href="#"><u>\$.ajax()</u></a>	Performs an async AJAX request
<a href="#"><u>\$.get()</u></a>	Loads data from a server using an AJAX HTTP GET request
<a href="#"><u>\$.getJSON()</u></a>	Loads JSON-encoded data from a server using a HTTP GET request
<a href="#"><u>\$.post()</u></a>	Loads data from a server using an AJAX HTTP POST request
<a href="#"><u>load()</u></a>	Loads data from a server and puts the returned data into the selected element

❑ More details:

[https://www.w3schools.com/jquery/jquery\\_ref\\_ajax.asp](https://www.w3schools.com/jquery/jquery_ref_ajax.asp)

# Problem: Using Multiple Libraries



- ❑ The jQuery library, and virtually all of its plugins are constrained within the jQuery namespace.
- ❑ We can use multiple libraries all together without conflicting each others.
- ❑ For example we can use jQuery and Prototype javascript libraries together.

# Problem: Using Multiple Libraries(Cont.)



- ❑ By default, jQuery uses "\$" as a shortcut for "jQuery".
- ❑ Many JavaScript libraries use \$ as a function or variable name, just as jQuery does.
- ❑ In jQuery's case, \$ is just an alias for jQuery, so all functionality is available without using \$.
- ❑ We can override that default by calling **jQuery.noConflict()** at any point after jQuery and the other library have loaded.

# Solution 1



```
<script src="jquery.js"></script>
<script>
    jQuery.noConflict() ; // Use jQuery via jQuery(...)

    jQuery(document).ready(function()
    {
        jQuery("div").hide();
    });
</script>
<script src="prototype.js"></script>
//use $() with prototype
```

# Solution 2



```
<script src="jquery.js"></script>
<script>
    var $j = jQuery.noConflict(); // Use jQuery via $j(...)
    $j (document).ready(function()
    {
        $j ("div").hide();
    });
</script>

<script src="prototype.js"></script>
//use $() with prototype
```

# Self Study



## ☐ Search and try other JS libraries, like:

- Underscore.js
- D3.js
- Prototype.
- scriptaculous.
- MooTools.
- Dojo.
- YUI.
- Ext.

## ☐ What're the differences between JS libraries and JS frameworks? Mention some JS known frameworks.

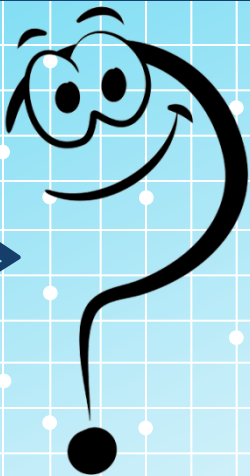


<script>



JavaScript

</script>

<SCRIPT>  </SCRIPT>

<script>document.writeln("Thank  
You!")</script>