

# Webonise Lab Induction Program

---

JavaScript and JQuery

# What is JavaScript?

- An interpreted programming language with object oriented capabilities.
- Not Java!
  - Originally called LiveScript, changed to JavaScript as a marketing ploy by Sun and Netscape. Can also be referred to as ECMAScript.
- Not simple!
  - Although it is loosely typed and can be used by web developers in a “cookbook” fashion (think image rollovers), JavaScript is a fully featured programming language with **many** advanced features.

# Client Side JavaScript

- When JavaScript is embedded in a web browser, it is referred to as Client Side JavaScript.
- Contains an extended set of functionality to interface with the web browser DOM (Document Object Model).
- Objects, such as **window** and **document**, and functions, like event detection and handling, are included in Client Side JavaScript.

# What is jQuery?

- A framework for Client Side JavaScript.
- Frameworks provide useful alternatives for common programming tasks, creating functionality which may not be available or cumbersome to use within a language.
- An open source project, maintained by a group of developers, with a very active support base and thorough, well written documentation.

# Why jQuery?

- Lightweight – 14kb (Minified and Gzipped)
- Cross-browser support (IE 6.0+, FF 1.5+, Safari 2.0+, Opera 9.0+)
- CSS-like syntax – easy for developers/non-developers to understand
- Active developer community
- Extensible - plugins

# What jQuery is not...

- A substitute for knowing JavaScript
  - jQuery is extraordinarily useful, but you should still know how JavaScript works and how to use it correctly. This means more than Googling a tutorial and calling yourself an expert.
- A solve all
  - There is still plenty of functionality built into JavaScript that should be utilized! Don't turn every project into a quest to 'jQuery-ize' the problem, use jQuery where it makes sense. Create solutions in environments where they belong.

# What is available with jQuery?

- Cross browser support and detection
- AJAX functions
- CSS functions
- DOM manipulation
- DOM transversal
- Attribute manipulation
- Event detection and handling

JavaScript animation

Hundreds of plugins for pre-built user interfaces, advanced animations, form validation, etc

Expandable functionality using custom plugins

Small foot print

# jQuery Syntax

**`$.func(...);`**

**or**

**`$(selector).func1(...).func2(...).funcN(...);`**

- \$ jQuery Object, can be used instead of jQuery
- selector Selector syntax, many different selectors allowed
- func Chainable, most functions return a jQuery object
- (...) Function parameters



# The jQuery/\$ Object

- Represented by both **\$** and **jQuery**
  - To use **jQuery** only, use `jQuery.noConflict()`, for other frameworks that use `$`
- By default, represents the jQuery object. When combined with a selector, can represent multiple DOM Elements, see next slide.
- Used with all jQuery functions.

# jQuery Selectors

- **`$( html )`**

Create DOM elements on-the-fly from the provided String of raw HTML.

- **`$( elems )`**

Wrap jQuery functionality around single or multiple DOM Elements.

- **`$( fn )`**

A shorthand for `$(document).ready()`, allowing you to bind a function to be executed when the DOM document has finished loading.

## **`$( expr, context )`**

This function accepts a string containing a CSS or basic XPath selector which is then used to match a set of elements. Default context is document. Used most often for DOM transversal.

Selectors will return a jQuery object, which can contain one or more elements, or contain no elements at all.

# jQuery Selector Examples

- **`$( html )`**
  - `$(' <p><a href="index.html">Click here!</a></p>')`
- **`$ ( elems )`**
  - `$(document)`, `$(window)`, `$(this)`
  - `$(document.getElementsByTagName("p"))`
- **`$ ( fn )`**
  - `$(function() { alert("Hello, World!") });`
- **`$ ( expr, context )`**
  - `$("p")`, `$("form")`, `$("input")`
  - `$("p#content")`, `$("#content")`, `$(".brandnew")`, `$("p span.brandnew:first-child, #content")`
  - `$("p/span")`, `$("p/span[@class=brandnew]")`, `$(p/span:first)`, `$(p:first/span:even)`
  - `$("input:checkbox[@checked]")`, `$("div:visible p[a]")`
  - `var xml = '<d><it w="h1"><nm>One</nm></it><it w="h2"><nm>Two</nm></it></d>';`  
`$( "d it nm:contains('One')", xml), $("it[@w^=h]",xml)`

# jQuery Functions

- Attached to the jQuery object or chained off of a selector statement.
- Most functions return the jQuery object they were originally passed, so you can perform many actions in a single line.
- The same function can perform an entirely different action based on the number and type of parameters.

# jQuery Usage Example

```
$( "li:odd" ).prepend( '<span>Changed</span>' ).css( {background:"red"} );
```

```
<ul>
  <li>
    First item
  </li>
  <li>
    Second item
  </li>
  <li>
    Third item
  </li>
</ul>
```

```
<ul>
  <li>
    <span>Changed</span>
    First item
  </li>
  <li>
    Second item
  </li>
  <li>
    <span>Changed</span>
    Third item
  </li>
</ul>
```

```
<ul>
  <li style="background:red;">
    <span>Changed</span>
    First item
  </li>
  <li>
    Second item
  </li>
  <li style="background:red;">
    <span>Changed</span>
    Third item
  </li>
</ul>
```

# jQuery Usage Example

```
$("#div:hidden").find(".foo").empty().text("Changed").end().show();
```

```
<div>
  <span class="foo">
    Some text
  </span>
</div>
<div style="display:none">
  <span>
    More text
  </span>
  <span class="foo">
    Goodbye cruel world.
  </span>
</div>
```

```
<div>
  <span class="foo">
    Some text
  </span>
</div>
<div style="display:none">
  <span>
    More text
  </span>
  <span class="foo">
    Goodbye cruel world.
  </span>
</div>
```

```
<div>
  <span class="foo">
    Some text
  </span>
</div>
<div style="display:none">
  <span>
    More text
  </span>
  <span class="foo">
    Goodbye cruel world.
  </span>
</div>
```

```
<div>
  <span class="foo">
    Some text
  </span>
</div>
<div style="display:none">
  <span>
    More text
  </span>
  <span class="foo">
    Changed
  </span>
</div>
```

```
<div>
  <span class="foo">
    Some text
  </span>
</div>
<div style="display:none">
  <span>
    More text
  </span>
  <span class="foo">
    Changed
  </span>
</div>
```

```
<div>
  <span class="foo">
    Some text
  </span>
</div>
<div style="display:block">
  <span>
    More text
  </span>
  <span class="foo">
    Changed
  </span>
</div>
```

# jQuery Advanced Example

```
$("span.none").click(  
  function() {  
    $(this).siblings(":checkbox").removeAttr("checked") ;  
  }  
);
```

```
$("span.all").click(  
  function() {  
    $(this).siblings(":checkbox").attr("checked","checked") ;  
  }  
);
```

```
$("span").click(  
  function() {  
    if($(this).text()=="Select All")  
      $(this).siblings(":checkbox").attr("checked","checked") ;  
    else if ($(this).attr("class")=="none")  
      $(this).siblings(":checkbox").removeAttr("checked") ;  
  }  
);
```

```
<div>  
  <span class="all">Select All</span>  
  <span class="none">Select None</span>  
  <input name="chk1" type="checkbox">  
  <input name="chk2" type="checkbox">  
  <input name="chk3" type="checkbox">  
</div>
```

```
<div>  
  <span class="all">Select All</span>  
  <span class="none">Select None</span>  
  <input name="chk4" type="checkbox">  
  <input name="chk5" type="checkbox">  
  <input name="chk6" type="checkbox">  
</div>
```

# jQuery & AJAX

- jQuery has a series of functions which provide a common interface for AJAX, no matter what browser you are using.
- Most of the upper level AJAX functions have a common layout:
  - `$.func(url[,params][,callback])`, [ ] optional
    - url: string representing server target
    - params: names and values to send to server
    - callback: function executed on successful communication.



# jQuery AJAX Functions

- `$.func(url[,params][,callback])`
    - `$.get`
    - `$.getJSON`
    - `$.getIfModified`
    - `$.getScript`
    - `$.post`
  - `$(selector), inject HTML`
    - `load`
    - `loadIfModified`
  - `$(selector), ajaxSetup` alts
    - `ajaxComplete`
    - `ajaxError`
    - `ajaxSend`
    - `ajaxStart`
    - `ajaxStop`
    - `ajaxSuccess`
- `$.ajax,`
- `$.ajaxSetup`
- `async`
  - `beforeSend`
  - `complete`
  - `contentType`
  - `data`
  - `dataType`
  - `error`
- `global`
  - `ifModified`
  - `processData`
  - `success`
  - `timeout`
  - `type`
  - `url`

# jQuery AJAX Example

```
<html>
<head>
<title>AJAX Demo</title>
<script type="text/javascript" src="jquery.js">
</script>
<script type="text/javascript">
var cnt = 0;
$(function(){
    $.ajaxSettings({
        error:function(){alert("Communication error!");}
    });
    $("button").click(function(){
        var input = {in:$(":textbox").val(),count:cnt};
        $.getJSON("ajax.php",input,function(json){
            cnt = json.cnt;
            $(".cnt").text(cnt)
            $(".msg").text(json.out);
        });
    });
});
</script>
</head>
<body>
<p>
    Input:
    <input type="textbox"/>
    <input type="button" value="Send"/>
    Output #
    <span class="cnt"></span>:
    <span class="msg"></span>
</p>
</body>
</html>
```

```
<?php
$output = '';

switch($_REQUEST['in']) {

    case 'hello':

        $output = 'Hello back.';

        break;

    case 'foo':

        $output = 'Foo you, too.';

        break;

    case 'bar':

        $output = 'Where Andy Capp can be found.';

        break;

    case 'foobar':

        $output = 'This is German, right?';

        break;

    default:

        $output = 'Unrecognized string.';

}

$count = $_REQUEST['count']+1;

echo json_encode(
    array(
        'out' => $output,
        'cnt' => $count
    )
);

exit;
?>
```

# jQuery Assignment

- ajax demo
- animation demo

# jQuery Resources

- Project website
  - <http://www.jquery.com>
- Learning Center
  - <http://docs.jquery.com/Tutorials>
  - <http://www.learningjquery.com/>
  - <http://15daysofjquery.com/>
- Support
  - <http://docs.jquery.com/Discussion>
  - <http://www.nabble.com/JQuery-f15494.html> mailing list archive
  - [irc.freenode.net](http://irc.freenode.net) irc room: #jquery
- Documentation
  - [http://docs.jquery.com/Main\\_Page](http://docs.jquery.com/Main_Page)
  - <http://www.visualjquery.com>
  - <http://jquery.bassistance.de/api-browser/>
- jQuery Success Stories
  - [http://docs.jquery.com/Sites\\_Using\\_jQuery](http://docs.jquery.com/Sites_Using_jQuery)

# Webonise Lab Induction Program

---

JavaScript and JQuery

[WWW.WEBONISELAB.COM](http://WWW.WEBONISELAB.COM)

# What is JavaScript?

- An interpreted programming language with object oriented capabilities.
- Not Java!
  - Originally called LiveScript, changed to JavaScript as a marketing ploy by Sun and Netscape. Can also be referred to as ECMAScript.
- Not simple!
  - Although it is loosely typed and can be used by web developers in a “cookbook” fashion (think image rollovers), JavaScript is a fully featured programming language with **many** advanced features.

# Client Side JavaScript

- When JavaScript is embedded in a web browser, it is referred to as Client Side JavaScript.
- Contains an extended set of functionality to interface with the web browser DOM (Document Object Model).
- Objects, such as **window** and **document**, and functions, like event detection and handling, are included in Client Side JavaScript.

# What is jQuery?

- A framework for Client Side JavaScript.
- Frameworks provide useful alternatives for common programming tasks, creating functionality which may not be available or cumbersome to use within a language.
- An open source project, maintained by a group of developers, with a very active support base and thorough, well written documentation.





# Why jQuery?

- Lightweight – 14kb (Minified and Gzipped)
- Cross-browser support (IE 6.0+, FF 1.5+, Safari 2.0+, Opera 9.0+)
- CSS-like syntax – easy for developers/non-developers to understand
- Active developer community
- Extensible - plugins

# What jQuery is not...

- A substitute for knowing JavaScript
  - jQuery is extraordinarily useful, but you should still know how JavaScript works and how to use it correctly. This means more than Googling a tutorial and calling yourself an expert.
- A solve all
  - There is still plenty of functionality built into JavaScript that should be utilized! Don't turn every project into a quest to 'jQuery-ize' the problem, use jQuery where it makes sense. Create solutions in environments where they belong.

# What is available with jQuery?

- Cross browser support and detection
- AJAX functions
- CSS functions
- DOM manipulation
- DOM transversal
- Attribute manipulation
- Event detection and handling

JavaScript animation

Hundreds of plugins for pre-built user interfaces, advanced animations, form validation, etc

Expandable functionality using custom plugins

Small foot print

# jQuery Syntax

**`$.func(...);`**

**or**

**`$(selector).func1(...).func2(...).funcN(...);`**

- \$ jQuery Object, can be used instead of jQuery
- selector Selector syntax, many different selectors allowed
- func Chainable, most functions return a jQuery object
- (...) Function parameters

# The jQuery/\$ Object

- Represented by both **\$** and **jQuery**
  - To use **jQuery** only, use `jQuery.noConflict()`, for other frameworks that use `$`
- By default, represents the jQuery object. When combined with a selector, can represent multiple DOM Elements, see next slide.
- Used with all jQuery functions.

# jQuery Selectors

- **`$( html )`**

Create DOM elements on-the-fly from the provided String of raw HTML.

- **`$( elems )`**

Wrap jQuery functionality around single or multiple DOM Elements.

- **`$( fn )`**

A shorthand for `$(document).ready()`, allowing you to bind a function to be executed when the DOM document has finished loading.

## **`$( expr, context )`**

This function accepts a string containing a CSS or basic XPath selector which is then used to match a set of elements. Default context is document. Used most often for DOM transversal.

Selectors will return a jQuery object, which can contain one or more elements, or contain no elements at all.

# jQuery Selector Examples

- **`$( html )`**
  - `$('<p><a href="index.html">Click here!</a></p>')`
- **`$( elems )`**
  - `$(document)`, `$(window)`, `$(this)`
  - `$(document.getElementsByTagName("p"))`
- **`$( fn )`**
  - `$(function() { alert("Hello, World!") });`
- **`$( expr, context )`**
  - `$("p")`, `$("form")`, `$("input")`
  - `$("#content")`, `("#content")`, `(".brandnew")`, `("p span.brandnew:first-child, #content")`
  - `("/p/span")`, `("/p/span[@class=brandnew]")`, `(p/span:first)`, `(p:first/span:even)`
  - `("input:checkbox[@checked]")`, `("div:visible p[a]")`
  - `var xml = '<d><it w="h1"><nm>One</nm></it><it w="h2"><nm>Two</nm></it></d>';`  
`$("d it nm:contains('One')", xml)`, `($"it[@w^=h]",xml)`

# jQuery Functions

- Attached to the jQuery object or chained off of a selector statement.
- Most functions return the jQuery object they were originally passed, so you can perform many actions in a single line.
- The same function can perform an entirely different action based on the number and type of parameters.



# jQuery Usage Example

```
$("#li:odd").prepend('<span>Changed</span>').css({background:"red"});
```

```
<ul>
  <li>
    First item
  </li>
  <li>
    Second item
  </li>
  <li>
    Third item
  </li>
</ul>
```

```
<ul>
  <li>
    <span>Changed</span>
    First item
  </li>
  <li>
    Second item
  </li>
  <li>
    <span>Changed</span>
    Third item
  </li>
</ul>
```

```
<ul>
  <li style="background:red;">
    <span>Changed</span>
    First item
  </li>
  <li>
    Second item
  </li>
  <li style="background:red;">
    <span>Changed</span>
    Third item
  </li>
</ul>
```

# jQuery Usage Example

```
$("#div:hidden").find(".foo").empty().text("Changed").end().show();
```

```
<div>
  <span class="foo">
    Some text
  </span>
</div>
<div style="display:none">
  <span>
    More text
  </span>
  <span class="foo">
    Goodbye cruel world.
  </span>
</div>
```

```
<div>
  <span class="foo">
    Some text
  </span>
</div>
<div style="display:none">
  <span>
    More text
  </span>
  <span class="foo">
    Goodbye cruel world.
  </span>
</div>
```

```
<div>
  <span class="foo">
    Some text
  </span>
</div>
<div style="display:none">
  <span>
    More text
  </span>
  <span class="foo">
  </span>
</div>
```

```
<div>
  <span class="foo">
    Some text
  </span>
</div>
<div style="display:none">
  <span>
    More text
  </span>
  <span class="foo">
    Changed
  </span>
</div>
```

```
<div>
  <span class="foo">
    Some text
  </span>
</div>
<div style="display:none">
  <span>
    More text
  </span>
  <span class="foo">
    Changed
  </span>
</div>
```

```
<div>
  <span class="foo">
    Some text
  </span>
</div>
<div style="display:block">
  <span>
    More text
  </span>
  <span class="foo">
    Changed
  </span>
</div>
```

WWW.WEBONSECURE.COM

# jQuery Advanced Example

```
$("#span.none").click(
function(){
    $(this).siblings(":checkbox").removeAttr("checked");
}
);
```

```
$("#span.all").click(
function(){
    $(this).siblings(":checkbox").attr("checked","checked");
}
);
```

```
$("#span").click(
function(){
    if($("#this").text()=="Select All")
    $("#this").siblings(":checkbox").attr("checked","checked");
    else if ($("#this").attr("class")=="none")
    $("#this").siblings(":checkbox").removeAttr("checked");
}
);
```

```
<div>
    <span class="all">Select All</span>
    <span class="none">Select None</span>
    <input name="chk1" type="checkbox"/>
    <input name="chk2" type="checkbox"/>
    <input name="chk3" type="checkbox"/>
</div>
```

```
<div>
    <span class="all">Select All</span>
    <span class="none">Select None</span>
    <input name="chk4" type="checkbox"/>
    <input name="chk5" type="checkbox"/>
    <input name="chk6" type="checkbox"/>
</div>
```

WWW.WEBONISELAB.COM

# jQuery & AJAX

- jQuery has a series of functions which provide a common interface for AJAX, no matter what browser you are using.
- Most of the upper level AJAX functions have a common layout:
  - `$.func(url[,params][,callback])`, [ ] optional
    - url: string representing server target
    - params: names and values to send to server
    - callback: function executed on successful communication.

# jQuery AJAX Functions

- \$.func(url[,params][,callback])
    - \$.get
    - \$.getJSON
    - \$.getIfModified
    - \$.getScript
    - \$.post
  - \$(selector), inject HTML
    - load
    - loadIfModified
  - \$(selector), ajaxSetup alts
    - ajaxComplete
    - ajaxError
    - ajaxSend
    - ajaxStart
    - ajaxStop
    - ajaxSuccess
- | \$.ajaxSetup  | \$.ajax,      |
|---------------|---------------|
| • async       | ▪ global      |
| • beforeSend  | ▪ ifModified  |
| • complete    | ▪ processData |
| • contentType | ▪ success     |
| • data        | ▪ timeout     |
| • dataType    | ▪ type        |
| • error       | ▪ url         |

# jQuery AJAX Example

```
<html>
<head>
<title>AJAX Demo</title>
<script type="text/javascript" src="jquery.js">
</script>
<script type="text/javascript">
var cnt = 0;
$(function () {
    $.ajaxSettings({
        error:function () {alert("Communication error!");}
    });
    $("button").click(function () {
        var input = (in:$(":textbox").val(),count:cnt);
        $.getJSON("ajax.php",input,function (json) {
            cnt = json.cnt;
            $(":cnt").text(cnt)
            $(":msg").text(json.out);
        });
    });
});
</script>
</head>
<body>
<p>
    Input:
    <input type="textbox"/>
    <input type="button" value="Send"/>
    Output #
    <span class="cnt"></span>:
    <span class="msg"></span>
</p>
</body>
</html>
```

```
<?php
$output = "";

switch($_REQUEST('in')) {
    case 'hello':
        $output = 'Hello back.';
        break;
    case 'foo':
        $output = 'Foo you, too.';
        break;
    case 'bar':
        $output = 'Where Andy Capp can be found.';
        break;
    case 'foobar':
        $output = 'This is German, right?';
        break;
    default:
        $output = 'Unrecognized string.';
}

$count = $_REQUEST('count')+1;

echo json_encode(
    array(
        'out' => $output,
        'cnt' => $count
    )
);

exit;
?>
```

www.webonsecab.com

# jQuery Assignment

- ajax demo
- animation demo

[WWW.WEBONISELAB.COM](http://WWW.WEBONISELAB.COM)

# jQuery Resources

- Project website
  - <http://www.jquery.com>
- Learning Center
  - <http://docs.jquery.com/Tutorials>
  - <http://www.learningjquery.com/>
  - <http://15daysofjquery.com/>
- Support
  - <http://docs.jquery.com/Discussion>
  - <http://www.nabble.com/JQuery-f15494.html> mailing list archive
  - [irc.freenode.net](http://irc.freenode.net) irc room: #jquery
- Documentation
  - [http://docs.jquery.com/Main\\_Page](http://docs.jquery.com/Main_Page)
  - <http://www.visualjquery.com>
  - <http://jquery.bassistance.de/api-browser/>
- jQuery Success Stories
  - [http://docs.jquery.com/Sites\\_Using\\_jQuery](http://docs.jquery.com/Sites_Using_jQuery)