

Introduction to jQuery

Some slides from: BreadFish

JavaScript

- **A script language interpreted by browser**
- **OOP (prototype based, not class based)**
- **Dynamic typing**
- **Run-time evaluation**

JavaScript

- **Cannot access host computer (except cookie)**
- **Same origin policy**
- **Non-persistence**
- **Cannot access history object**
- **Cannot write the value of file upload field**

JavaScript Libraries

- jQuery
- Mootools
- Prototype
- YUI

Introduction to jQuery

- Developed in 2006 by [John Resig](#) at Rochester Institute of Technology
- jQuery is a lightweight [JavaScript library](#) that emphasizes interaction between [JavaScript](#) and [HTML](#)
- jQuery is [free, open source software](#) [Dual-licensed](#) under the [MIT License](#) and the [GNU General Public License](#)
- Helps web developers to create simple pieces of interaction without being forced to write long, complex, book-length pieces of code

Introduction to jQuery

- Why do I want it
 - Rich Internet Applications (RIA)
 - Dynamic HTML (DHTML)
- How do I get it
 - www.jquery.com
- How can I learn it
 - jQuery in Action by Bibeault & Katz, Manning
 - jQuery Visual Quickstart Guide by Holzner, Peachpit
 - www.jquery.com
 - docs.jquery.com
 - www.visualjquery.com
 - www.Jqueryfordesigners.com
 - www.gscottolson.com/weblog/ - cheat sheet
 - www.javascripttoolbox.com/jquery
 - http://www.w3schools.com/jquery/jquery_examples.asp

Summary

- Introduction, installation, “Howdy World”, Ready function, DOM, Selecting and Formatting web page elements
- Events and Animations
- jQuery Plugin libraries
- AJAX

What jQuery Does

- “Unobtrusive” JavaScript
 - separation of behavior from structure
- CSS
 - separation of style from structure
- Allows adding JavaScript to your web pages
- Advantages over *just* JavaScript
 - Much easier to use
 - Eliminates cross-browser problems

5 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)
- AJAX calls

The DOM

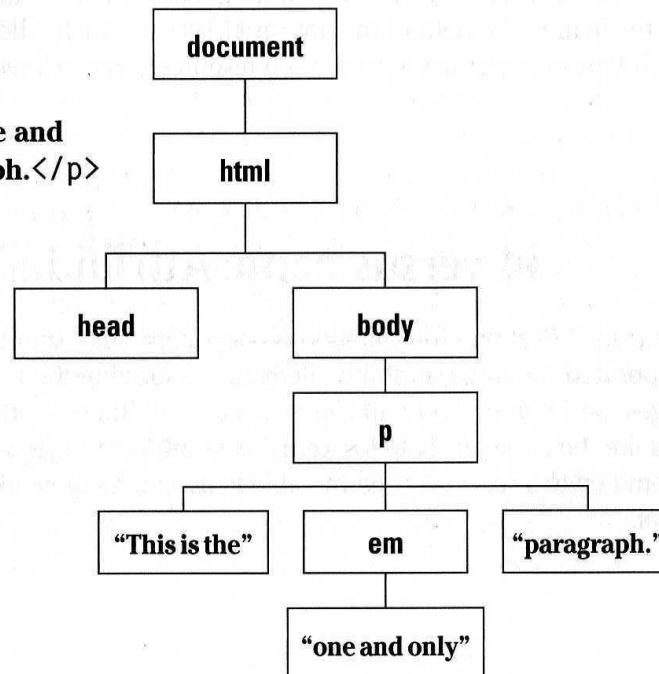
- Document Object Model
- jQuery is “DOM scripting”
- Hierarchical structure of a web page
- You can add and subtract DOM elements on the fly
- You can change the properties and contents of DOM elements on the fly

The DOM

“a [cross-platform](#) and [language](#)-independent convention for representing and interacting with [objects](#) in [HTML](#), [XHTML](#) and [XML](#) documents. Aspects of the DOM (such as its "Elements") may be addressed and manipulated within the syntax of the programming language in use.” Wikipedia

A simple HTML document node tree.

```
<html>
  <head></head>
  <body>
    <p>This is the <em>one and
      only</em> paragraph.</p>
  </body>
</html>
```



The jQuery Function

- `jQuery() = $()`
- `$(function)` The “Ready” handler
- `$('selector')` Element selector expression
- `$(element)` Specify element(s) directly
- `$('HTML')` HTML creation
- `$.function()` Execute a jQuery function
- `$.fn.myfunc(){}` Create jQuery function

The Ready Function

- Set up a basic HTML page and add jQuery
- Create a “ready” function
- Call a function
- 5 ways to specify the ready function
 - `$(document).ready(function(){...});`
 - `$.ready(function(){...});`
 - `$(function(){...});`
 - `jquery(dofunc);`
 - `$(dofunc);`

Selecting Elements – Creating a “wrapped set”

- \$(selector)
- selector:
 - \$('#id') id of element
 - \$('p') tag name
 - \$('.class') CSS class
 - \$('p.class') <p> elements having the CSS class
 - \$('p:first') \$('p:last') \$('p:odd') \$('p:even')
 - \$('p:eq(2)') gets the 2nd <p> element (1 based)
 - \$('p')[1] gets the 2nd <p> element (0 based)
 - \$('p:nth-child(3)) gets the 3rd <p> element of its parent.
 - \$('p:nth-child(odd)) gets the odd <p> element of its parent.
 - \$('p:nth-child(5n+1)') gets the 1st element after every 5th one
 - \$('p a') <a> elements, descended from a <p>
 - \$('p>a') <a> elements, direct child of a <p>
 - \$('p+a') <a> elements, directly following a <p>
 - \$('p, a') <p> and <a> elements
 - \$('li:has(ul)') elements that have at least one descendent
 - \$(':not(p)') all elements but <p> elements
 - \$('p:hidden') only <p> elements that are hidden
 - \$('p:empty') <p> elements that have no child elements

Selectors

```
<html>
<head>
<title>The jQuery Example</title>
<script type = "text/javascript"
src = "https://ajax.googleapis.com/ajax/libs/jquery/2.1.3/jquery.min.js">
</script>

<script type = "text/javascript" language = "javascript">

$(document).ready(function() {
    $("p").css("background-color", "yellow");
});

</script>
</head>

<body>
<div>
<p class = "myclass">This is a paragraph.</p>
<p id = "myid">This is second paragraph.</p>
<p>This is third paragraph.</p>
</div>
</body>
</html>
```

<https://www.tutorialspoint.com/jquery/jquery-selectors.htm>

Selecting Elements, cont.

<code>\$('img'[alt])</code>	<code></code> elements having an alt attribute
<code>\$('a'[href^=http://])</code>	<code><a></code> elements with an href attribute starting with 'http://'
<code>\$('a'[href\$=.pdf])</code>	<code><a></code> elements with an href attribute ending with '.pdf'
<code>\$('a'[href*=ntpcug])</code>	<code><a></code> elements with an href attribute containing 'ntpcug'

Useful jQuery Functions

<code>.each()</code>	iterate over the set
<code>.size()</code>	number of elements in set
<code>.end()</code>	reverts to the previous set
<code>.get(n)</code>	get just the nth element (0 based)
<code>.eq(n)</code>	get just the nth element (0 based) also <code>.lt(n)</code> & <code>.gt(n)</code>
<code>.slice(n,m)</code>	gets only nth to (m-1)th elements
<code>.not('p')</code>	don't include 'p' elements in set
<code>.add('p')</code>	add <p> elements to set
<code>.remove()</code>	removes all the elements from the page DOM
<code>.empty()</code>	removes the contents of all the elements
<code>.filter(fn/sel)</code>	selects elements where the func returns true or sel
<code>.find(selector)</code>	selects elements meeting the selector criteria
<code>.parent()</code>	returns the parent of each element in set
<code>.children()</code>	returns all the children of each element in set
<code>.next()</code>	gets next element of each element in set
<code>.prev()</code>	gets previous element of each element in set
<code>.siblings()</code>	gets all the siblings of the current element

<https://www.tutorialspoint.com/jquery/jquery-utilities.htm>

css(property, value)

```
<html>
<head>
<title>The jQuery Example</title>
<script type = "text/javascript"
src = "https://ajax.googleapis.com/ajax/libs/jquery/2.1.3/jquery.min.js">
</script>

<script type = "text/javascript" language = "javascript">
    $(document).ready(function() {
        $("div:first").width(100);
        $("div:first").css("background-color", "blue");
    });
</script>

<style>
div {
width:70px; height:50px; float:left;
margin:5px; background:red; cursor:pointer;
}
</style>
</head>

<body>
<div></div>
<div>d</div>
<div>d</div>
<div>d</div>
<div>d</div>
</body>
</html>
```

Formatting Elements-html()

```
<html>
<head>
<title>The jQuery Example</title>
<script type = "text/javascript"
src = "https://ajax.googleapis.com/ajax/libs/jquery/2.1.3/jquery.min.js">
</script>

<script type = "text/javascript" language = "javascript">
$(document).ready(function() {
$("#div").click(function () {
    var content = $(this).html();
    $("#result").text( content );
});
});
</script>

<style>
#division{ margin:10px;padding:12px; border:2px solid #666; width:60px;}
</style>
</head>

<body>
<p>Click on the square below:</p>
<span id = "result"> </span>

<div id = "division" style = "background-color:blue;">
This is Blue Square!!
</div>
</body>
</html>
```

Add Page Elements

- `$('#target').before('<p>Inserted before #target</p>');`
- `$('#target').after('<p>This is added after #target</p>');`
- `$('#target').append('<p>Goes inside #target, at end</p>');`
- `$('#target').wrap('<div></div>');`

Adding Events

- Mouseover events – bind, hover, toggle
- Button click events
- Keystrokes

Event Background

- DOM Level 2 Event Model
 - Multiple event handlers, or listeners, can be established on an element
 - These handlers cannot be relied upon to run in any particular order
 - When triggered, the event propagates from the top down (capture phase) or bottom up (bubble phase)
 - IE doesn't support the "capture phase"

Basic Syntax of Event Binding

- `$('#img').bind('click', function(event){alert('Howdy';)});`
- `$('#img').bind('click', imgclick(event));`
 - Allows unbinding the function
- `$('#img').unbind('click', imgclick());`
- `$('#img').unbind('click');`
- `$('#img').one('click', imgclick(event));`
 - Only works once
- `$('#img').click(imgclick);`
- `$('#img').toggle(click1, click2);`
- `$('#img').hover(mouseover, mouseout);`
- <https://www.tutorialspoint.com/jquery/jquery-events.htm>

Basic Syntax of Event Binding

```
<html>
<head>
<title>The jQuery Example</title>
<script type = "text/javascript"
src = "https://ajax.googleapis.com/ajax/libs/jquery/2.1.3/jquery.min.js">
</script>

<script type = "text/javascript" language = "javascript">
    $(document).ready(function() {
        $('div').bind('click', function( event ){
            alert('Event type is ' + event.type);
            alert('pageX : ' + event.pageX);
            alert('pageY : ' + event.pageY);
            alert('Target : ' + event.target.innerHTML);
        });
    });
</script>

<style>
.div{ margin:10px;padding:12px; border:2px solid #666; width:60px;}
</style>
</head>

<body>
<p>Click on any square below to see the result:</p>

<div class = "div" style = "background-color:blue;">ONE</div>
<div class = "div" style = "background-color:green;">TWO</div>
<div class = "div" style = "background-color:red;">THREE</div>
</body>
</html>
```


Element Properties – “this”

- this
- this.id
- this.tagName
- this.attr
- this.src
- this.className
- this.title
- this.alt
- this.value (for form elements)

'Event' properties

- `event.target` → ref to element triggering event
- `Event.target.id` → id of element triggering event
- `event.currentTarget`
- `event.type` → type of event triggered
- `event.data` → second parm in the `bind()` func
- Various mouse coordinate properties
- Various keystroke related properties

Event Methods

- `.stopPropagation()` no bubbling
- `.preventDefault()` no `<a>` link, no `<form>` submit
- `.trigger(eventType)` does not actually trigger the event, but calls the appropriate function specified as the one tied to the `eventType`

`.click()`, `blur()`, `focus()`, `select()`, `submit()`

With no parameter, invokes the event handlers, like `trigger` does, for all the elements in the wrapped set

Shortcut Event Binding

- `.click(func)`
- `.submit(func)`
- `.dblclick(func)`
- `.mouseover(func)`
- `.mouseout(func)`
- `.select(func)`

Useful Event Functions

<code>.hide()</code>	<code>display:none</code>
<code>.show()</code>	<code>display:true</code>
<code>.toggle(func1, func2)</code>	first click calls func1, next click executes func2
<code>.hover(over, out)</code>	mouseover, mouseout

<https://www.tutorialspoint.com/jquery/jquery-effects.htm>

Useful Event Functions

```
<html>
<head>
<title>The jQuery Example</title>
<script type = "text/javascript"
src = "https://ajax.googleapis.com/ajax/libs/jquery/2.1.3/jquery.min.js">
</script>

<script type = "text/javascript" language = "javascript">
$(document).ready(function() {
    $("#show").click(function () {
        $(".mydiv").show( 1000 );
    });

    $("#hide").click(function () {
        $(".mydiv").hide( 1000 );
    });
});
</script>
<style>
.mydiv{
margin:10px;
padding:12px;
border:2px solid #666;
width:100px;
height:100px;
}
</style>
</head>

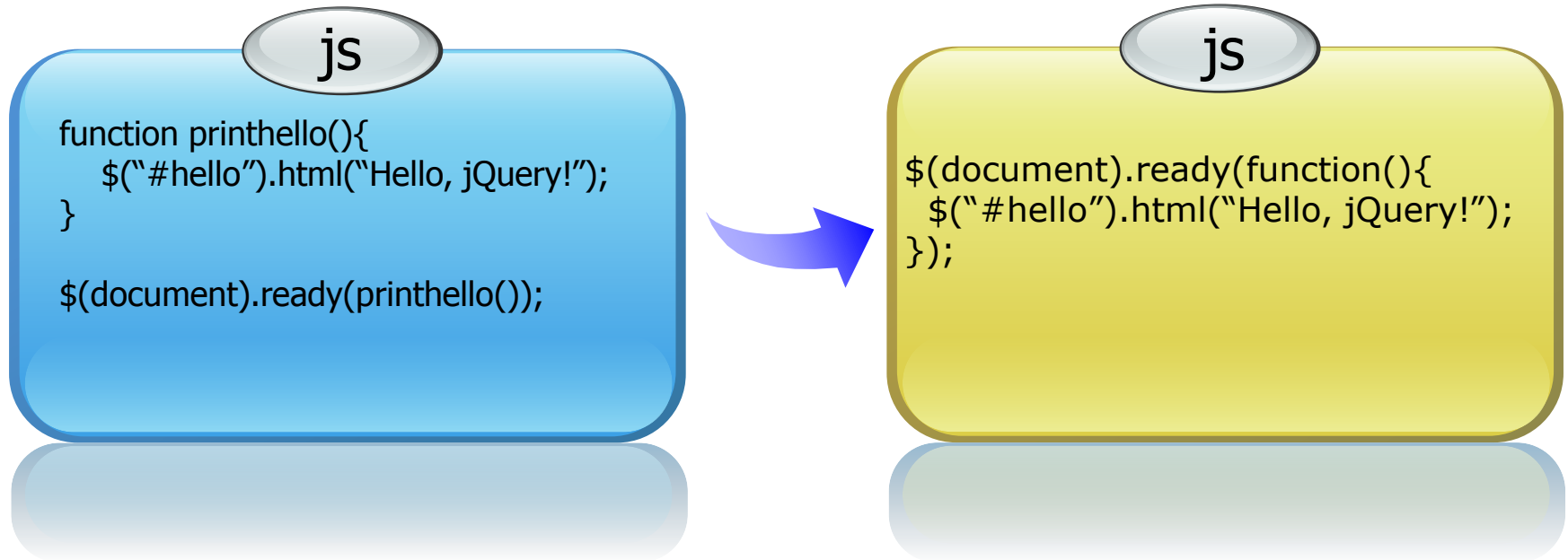
<body>
<div class = "mydiv">
This is a SQUARE
</div>
<input id = "hide" type = "button" value = "Hide" />
<input id = "show" type = "button" value = "Show" />
</body>
</html>
```

jQuery Core

- **jQuery(selector, [context])**: Accepts a string containing a CSS selector which is then used to match a set of elements and returns a jQuery object.
 - jQuery(element)
 - jQuery(elementArray)
 - jQuery(jQuery object)
 - jQuery()
- can be written as **\$()**

jQuery Events

- `.ready(handler)` : execute handler when the DOM is fully loaded.



❖ **Same as window.onload???**

jQuery Events

- .bind()
- .blur()
- .change()
- .click()
- .focus()
- .hover()
- .select()
- .toggle()
- .trigger()
- .submit()
- .mousedown()
- .mouseenter()
- .mouseleave()
- .keypress()
- .keyup()

jQuery Events

```
$(document).keyup(function(event) {  
    switch(event.which) {  
        case 32:  
            alert("32 pressed");  
            break;  
    }  
});
```

- ❖ `event.preventDefault()`
- ❖ `event.stopPropagation()`

jQuery Selectors

- follows CSS1~3 Selectors

<http://www.w3.org/TR/css3-selectors>

- :animated
- :has()
- :gt()

jQuery Attributes

- **.addClass()**
- **.removeClass()**
- **.hasClass()**
- **.toggleClass()**
- **.attr()**
- **.removeAttr()**
- **.val()**
- **.html()**

jQuery Each

- **.each()** : Iterate over a jQuery object, executing a function for each matched element.

html

```
<ul>
  <li>garbage</li>
  <li>food</li>
  <li>abroad</li>
</ul>
```

js

```
$(document).ready(function(){
  $('li').each(function(index) {
    alert(index + ': ' + $(this).text());
  });
});
```

jQuery Traversing

- **.add()**
- **.children()**
- **.contents()**
- **.filter()**
- **.find()**
- **.next()**
- **.not()**
- **.prev()**

jQuery Manipulations

- **.append()**
- **.appendTo()**
- **.clone()**
- **.detach()**
- **.insertAfter()**
- **.insertBefore()**
- **.remove()**

jQuery CSS

- **.css()**
- **.height()**
- **.width()**
- **.position()**
- **.offset()**
- **.scrollTop()**
- **.scrollLeft()**

jQuery Effect

- **.animate()**

html

```
<button id="left">left</button>  
<button id="right">right</button>  
<div class="block"></div>
```

js

```
$(document).ready(function(){  
  $(".block").css({  
    'position': 'absolute',  
    'backgroundColor': "#abc",  
    'left': '100px',  
    'width': '90px',  
    'height': '90px',  
    'margin': '5px' });  
  $("#left").click(function(){  
    $(".block").animate({left: "-=50px"}, "slow");  
  });  
  $("#right").click(function(){  
    $(".block").animate({left: "+=50px"}, "slow");  
  });  
});
```

jQuery Effect

- **.fadeIn()**
- **.hide()**
- **.show()**
- **.toggle()**

AJAX

- Asynchronous JavaScript And XML
- The basic AJAX function – XMLHttpRequest
- Initiating a request
- Getting the response

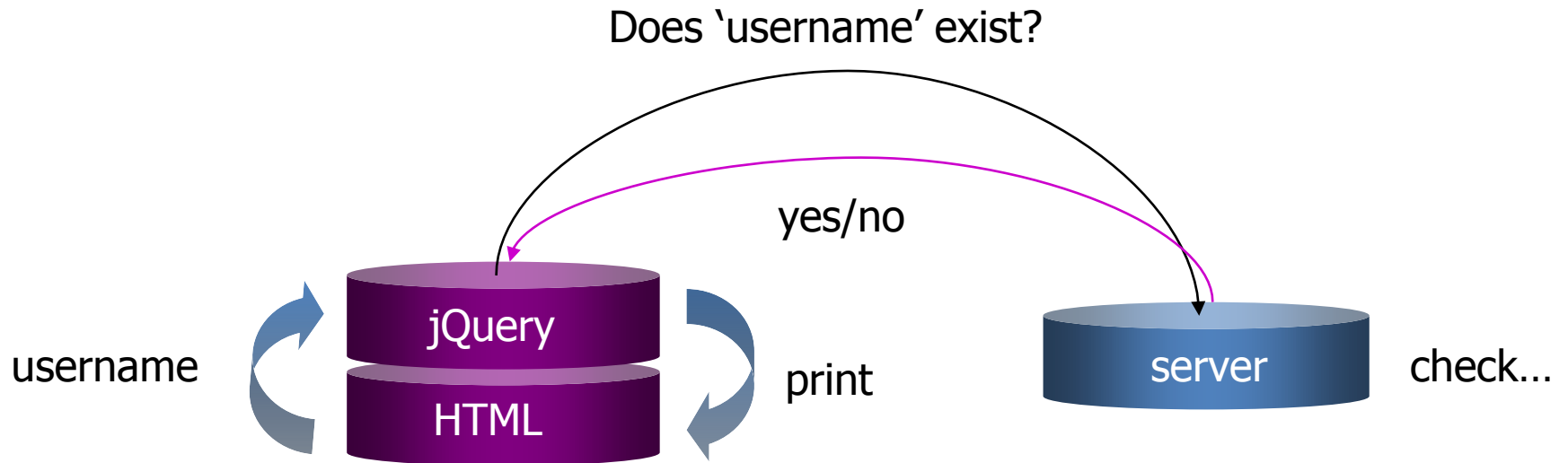
jQuery AJAX

- [jQuery.get](#)(url, [data], [callback(data, textStatus, XMLHttpRequest)], [dataType])

Returns: [XMLHttpRequest](#)

- [jQuery.post\(\)](#)
- [jQuerygetJSON\(\)](#)
- [jQuery.load\(\)](#)
- [jQuery.getScript\(\)](#)

AJAX Example



jQuery AJAX

html

```
<div id="user_check">  
  <input id="username" type="text"></input>  
  <input id="username_submit" type="button"  
value="submit"></input>  
  <p id="check_result"></p>  
</div>
```

js

```
$(document).ready(function(){  
  $("#username_submit").click(function(){  
    $.get('jqtest.cgi', {"username" : $("#username").val()},  
function(data){  
    $('#check_result').html(data);  
    });  
  });  
});
```