

Doing More with Less: Javascript and jQuery joy

John Ryding
trippinbilly25@gmail.com

Javascript Basics

if (boolean)

while (boolean)

for(var i=0; i < array.length; i++)

for (var i in object)

Variable Funkiness

-Loosely typed language

```
function declare() {  
  var fun = 1;  
  fun = "cool man!"  
  return fun;  
}
```

```
declare() == "cool man!"
```

Variables

- Variables are scoped to the **function** not the block

```
function test() {  
    for ( var i = 0, i < 10, i++) {}  
  
    return i;  
}  
  
test() == 10;
```

Variables

There are a few primitives offered:

-integers/floats: 4, 1.5

-Strings: "hello world"

-boolean: true/false

Type coercion

noValue == false

null == false

"yay string!" == true

Declaring primitives:

```
var bird : string = "robbin"
```

```
var value : double = 42.3
```

Objects

```
var obj = new Object;  
obj.property = value;  
var foo = { key1: "first", key2: "second" }  
foo.key1 == "first";  
foo["key1"] == "first";
```

Arrays:

```
var values = [ 1, 2, 3 ];  
values.push( 4 );  
var stuff = [];
```

There are no associative arrays though, only emulated

Functions are first order objects

Prototypes

-There is always an object "above" your current object

-In the DOM, the window object is the overall global object

```
var foo = function obj1() {} //reference obj1
```

```
foo = obj1.prototype //now referencing window object
```

Challenge!

```
function Building() {  
    this.location = "McBryde";  
}
```

```
Building.prototype.getLocation = function() {  
    return this.location;  
}
```

```
var foo = new Building();  
var getLocation = foo.getLocation();  
var locale = getLocation();  
//what is the value of locale?
```

In Between Functional and OO

-Functions are considered to be first order objects

```
function circle( radius ){  
    this.radius = radius;  
}
```

-The ability to have anonymous functions:

```
var foo = function(){};  
setTimeout( function(){  
    alert( "anonymous!" );  
}, 1000 );
```

-Closures are supported

Closures

- Remember that variables only exist **within** the **function** they are called
- Closures allows us to encapsulate and reference data by calling other functions within the function (wtf?)

```
function Bus() {  
    var passengers = 0;  
    this.getPassengers = function() {  
        return passengers;  
    }  
    this.addPassenger = function(){  
        passengers++;  
    }  
}
```

```
var bus = new Bus();  
bus.addPassenger;  
bus.getPassengers == 1;  
bus.passengers; //does not exist
```

Enter jQuery

- Frameworks take care of cross-browser consistency
- very lightweight (15 kb)
- CSS1, 2, 3 compliant
- do lots with less code!

Selectors

-Use CSS and XPath selectors to grab a variety of nodes

-MUCH more versatile than document.getElementById

```
$( "#idValue" ) //grab node that has given id
```

```
$( "#id > a" ) //grab all anchors that are childs of id
```

```
$( "ul > li:not(:first)" )  
  //grab all li nodes in all unordered lists EXCEPT the  
  // first choice
```

```
$("#id").find("a") //get all child anchors of id node
```

```
$("#id").parents("ul") //get parent node of id node
```

Easy Funtionality

-Add functionality when the page loads

```
$(document).ready( function(){} );
```

Example:

```
$(document).ready( function() {  
    $("div a").click( function() {  
        alert( "peanut butter jelly time!" );  
    });  
});
```

Equivalent to:

```
<a onclick="alert('peanut butter jelly time!');" />
```

Event Functionality

```
$(document).ready(function() {  
  
  //get the last li node of the list  
  $("#orderedlist li:last").hover(function() {  
  
    //when the mouse hovers over this node, set the  
    // class attribute to "green"  
    $(this).addClass("green");  
  
  }, function(){  
  
    //when mouse moves away, remove the class  
    $(this).removeClass("green");  
  });  
});
```

jQuery For-Each

```
$(document).ready(function() {  
  
    //grab all li child elements and iterate through them  
    $("#orderedlist").find("li").each(function(i) {  
  
        //append "Bam!" to the current li node iterator is on  
        $(this).append( " BAM! " );  
  
    });  
  
});
```

Animations

```
$("#right").click(function(){  
    //move the block right 50 pixels  
    $(".block").animate({"left": "+=50px"}, "slow");  
});
```

```
$("#left").click(function(){  
    //move the block left 50 pixels  
    $(".block").animate({"left": "-=50px"}, "slow");  
});
```

Ajax

```
//save data to the server and alert the user when done $.ajax({  
type: "POST",  
url: "some.php",  
data: "name=John&location=Boston",  
success: function(msg){  
alert( "Data Saved: " + msg );  
}  
});
```

AJAX cont.

```
//load an html page into all divs  
4("div").load( "somePage.html" );
```

```
//GET call  
$.get( "test.php", { name: "John", score: "80" }  
    , function( data ){  
        alert( "Data loaded: " + data );  
    });
```

```
//JSON data supported  
$.getJSON("test.js", function(json){  
    alert("JSON Data: " + json.users[3].name);  
});
```

Chaining

- Kind of like a built in stack for jQuery
- Attach multiple commands to one line of code

```
$( div ).addClass( 'green' );  
$( div ).show();
```

Becomes:

```
$( div ).addClass( 'green' ).show();
```

Advanced Chaining

```
$( "ul" )  
    .find( "li" )  
        .appendTo( "Think Think Think" )  
        .addClass( "green" );  
    .end  
    .show();
```

Plug-ins

- jQuery is an extensible framework
- Help encapsulate your work
- Simply made by extending the jQuery Object:
`jQuery.fn.methodName = function() {}`

Simple Plug-in

```
$("#input[@type='checkbox']").each(function() {  
    this.checked = true;  
});
```

Becomes:

```
jQuery.fn.check = function() {  
    return this.each( function() {  
        this.checked = true;  
    });  
};
```

```
$("#input[@type='checkbox']").check();
```

Dev Environments and Links

- Aptana Studio <http://aptana.com/>
- Visual Studio:
<http://jquery.com/blog/2008/09/28/jquery-microsoft-nokia/>
- www.jquery.com
- www.ejohn.org
- www.ajaxian.com
- <http://alex.dojotoolkit.org/> <---- wtf?

Bibliography

Bennett, James. "The functional language that's right under your nose." The B-List. 11 Oct. 2006. 12 Oct. <<http://www.b-list.org/weblog/2006/oct/11/functional-language-s-right-under-your-nose/>>.

Elaine. "Javascript and internships." Weblog post. Meebo Blog. 5 Dec. 2007. 12 Oct. 2008 <<http://blog.meebo.com/?p=385>>.

Resign, John. "Javascript and jQuery." Northeastern ACM Tech Talk. Massachusetts, Boston. 12 Oct. 2008. John Resig. 27 Mar. 2008. 12 Oct. 2008 <<http://ejohn.org/blog/javascript-talk-at-northeastern/>>.

Zaefferer, Jörn. "Tutorials:Getting Started with jQuery From jQuery JavaScript Library Jump to: navigation, search." JQuery. 12 Oct. 2008 <http://docs.jquery.com/tutorials:getting_started_with_jquery>.

Thanks!