



# JavaScript and jQuery for Designers

## Lesson 1: The Basics

Michael Slater, CEO  
Christopher Haupt, CTO  
[course-support@webvanta.com](mailto:course-support@webvanta.com)  
888.670.6793

**[www.webvanta.com](http://www.webvanta.com)**

# Welcome!



- Four sessions
  - **1: JavaScript & jQuery Basics**
  - 2: Setting Up Plugins
  - 3: Making Forms Work Well
  - 4: Getting Fancy
- Course Portal has all the links you need
  - [www.online-web-courses.com](http://www.online-web-courses.com)
  - You need your personal login

# JavaScript & jQuery Basics

- What is JavaScript, and why do we need it?
- Elementary JavaScript programming
- What jQuery is, and why it makes our life so much easier
- Where to put JavaScript in your pages, and how to be "unobtrusive"
- Example: Hide and show

# Questions

- Ask questions via chat during the live course
  - Second instructor standing by
- Join the discussion group
  - Link at the course portal
- Email us
  - [course-support@webvanta.com](mailto:course-support@webvanta.com)
  - *Please use the discussion group* for non-private questions



Michael Slater  
Cofounder & CEO



Christopher Haupt  
Cofounder & CTO

# Goals for This Course

- Enable non-programmers to use jQuery widgets
- Make cut-and-paste coding less mysterious and more reliable
- Start down the path of being able to write your own JavaScript code
- *Non-goal: we're not trying to turn you into programmers*

# Testing Your Code

- The only way to learn is through lots of practice
- You want to be in complete control, and be able to iterate quickly
  - Use a simple text editor
    - **Not** Dreamweaver or other visual editor
  - View files locally using your browser
  - Use Firebug or Chrome Dev Tools to inspect and modify your pages

# Code Editors

- Use an editor that provides HTML syntax highlighting (makes it easier to spot mistakes)
- Autocomplete or menu-driven markup *might* be helpful
- Good Mac solutions
  - TextMate, BBEdit, Coda, Espresso, Sublime Text
- Good Windows solutions
  - Sublime Text, Ultraedit, HTML-Kit, EditPlus, TextPad

```
<html lang="en" >
  <head>
  </head>
  <body>
    <h1>Headline</h1>
    <p>Body text</p>
  </body>
</html>
```

# Why Do We Need JavaScript?

- HTML and CSS are markup and styling languages, **not** programming languages
  - No variables, conditionals, loops
- Basic web model is document based
  - Server delivers document
  - Browser displays document
- JavaScript allows the "document" to include software

# Some Things JS Can Do

- Hide or show page elements in response to user actions
- Sort or restructure content
- Modify markup or styles
- Make things move over time (animation)
- Fetch new content from the server and insert into the page (Ajax)
- Validate user input on forms and display validation messages

# Won't HTML5 and CSS3 Do All This For Me?

- Some of the tasks commonly done in JS can be done without it using HTML5 and CSS3
  - You just need to wait until IE10 is the *oldest* browser you care about supporting!
  - So for your current career, you need JS
- Most advanced approach:
  - Use HTML5/CSS3 techniques where available
  - Use JavaScript fallbacks for less-capable browsers

# Why jQuery?

- JavaScript is a low-level language
  - jQuery provides higher-level functions so you can do more with less code
- JavaScript manipulation of the DOM (document object model) needs to be done differently depending upon the specific browser
- Allows use of CSS selectors in JS code
- Provides a mechanism to bundle up standard solutions (plugins)

# jQuery and Friends

- **jQuery** is an open-source project
  - Led by John Resig
  - Quite mature, now in version 1.7.1
- **jQuery UI** is a library of user interface elements that builds upon jQuery
- **jQuery Mobile** is a mobile website framework that builds upon jQuery
- Thousands of third-party plugins available, mostly open-source

# JavaScript Coding Basics

- Insert code anywhere in an HTML page

```
<script>  
    alert("Hello World!");  
</script>
```

- Load an external JavaScript file

```
<script src="/js/myfile.js"></script>
```

- No need for `type="text/javascript"`

# Basic JavaScript Syntax

- JavaScript is a complex, messy language with more choices than are good for you
  - Our approach: pragmatic, simple rules for simple programs (not a rigorous language specification)
- Each statement ends with a semicolon (;)
- Put each statement on a separate line
- Use braces to enclose blocks { }
- Use brackets to enclose arrays [ ]
- Use commas to separate items
- Use quotes around strings

# Variables

- Variables are named pieces of memory: the workhorse of all programs
- In JavaScript, variables are extremely flexible
  - Can hold a value, an object, or a function
- Commonly named in camelCase
  - Also ok to use under\_score
- Have global scope unless inside a function
- Use keyword "var" when defining a variable
  - We'll explore this more in later lessons

# Objects

- A flexible structure for encapsulating an arrangement of data and, optionally, functions

```
var course = {  
  name: "jQuery for Designers",  
  sessions: 4,  
  firstDate: "March 13, 2012",  
  classTime: "1 pm PST"  
}
```

```
course.name => "jQuery for Designers"
```

# Functions

- Functions encapsulate a block of code
- Most commonly, you will be invoking existing functions
  - `alert("Hello World");`
- You can also create your own, which can have names or be "anonymous" (most common)
- Inside the parenthesis, there may be nothing, a single parameter, or a list of parameters
- Functions may return a value, or just perform an action independently

# Creating Functions

```
function doSomething(param1, param2, ...) {  
    // code that does something goes here  
};
```

# Loading jQuery

- JavaScript is built into the browser, but jQuery is not
- Must load jQuery before any statements that use jQuery
- Best practice: load from Google CDN
  - Persists in browser cache, even across sites
- Also OK to load from local file, especially during development
- Use minified version unless you need to want to see jQuery source

# Loading jQuery

- From Google's CDN:

```
<script src="https://ajax.googleapis.com/
ajax/libs/jquery/1.7.1/jquery.min.js"></
script>
```

- From your local files, assuming you put it in a folder named "js":

```
<script src="/js/jquery.min.js">
```

# Using jQuery

- Access the jQuery library using statements that begin with \$ (the magic jQuery selector)
  - Long-form "jquery" can also be used
  - Some other JS libraries define \$ differently, so conflicts can occur
- Usually, you'll start by using jQuery to access an element of the HTML document (i.e., the DOM)

# Using jQuery to Access a Page's HTML

- jQuery makes it easy to use any CSS selector to access DOM elements via JavaScript
- Syntax is the same as in a stylesheet
  - `$("#special")` - returns an array of all DOM elements with an ID of "special" (even if just one can exist)
  - `(".specials")` - returns an array of all DOM elements with a class of "specials"
  - `("h2")` - returns array of all h2 elements
  - `("#article h2")` - returns array of h2 elements that are descendants of an element with the ID of "article"

# Once You Have an Element...

- You can change it

```
$("#h2").css("color", "green");
```

```
$("#special").hide();
```

- You can get its contents

```
h2text = $("#h2").html();
```

- You can invoke other functions (such as jQuery plugins) on it

```
$("#alertbox").colorbox();
```

# Unobtrusive JavaScript

- No JavaScript code mixed with HTML
  - Never put any JS in an HTML statement
- JavaScript code connects to the HTML via the DOM
  - Attach event handler functions to items on which you want to detect click, hover, ready, etc.
  - Read and write HTML content via the DOM
- Ensures HTML runs without JS, and keeps the code together

# Detecting Events

- **Event Handler:** function that is executed when a specified event occurs
- Typically an *anonymous function*

```
$("#show-alert").click(function() {  
    $("#hello").show();  
});
```

# Anonymous Functions

## Simplify Code

```
$("#show-alert").click(function() {  
    $("#hello").show();  
});
```

Otherwise, you would need something like this:

```
$("#show-alert").click(showHello);  
  
function showHello() {  
    $("#hello").show();  
};
```

# Creating an Alert Within the Page

```
<div id="hello" style="display:none">  
  <p style="color:red;">Hello World!</p>  
</div>
```

```
<button id='show-alert'>Show Alert</button>
```

```
<script>  
  $("#show-alert").click(function() {  
    $("#hello").show();  
  });  
</script>
```

# When JavaScript Runs

- Unless you do something to prevent it, JS runs when the browser encounters the code
- Need to be sure that any DOM elements the JS is referencing have already been loaded
- JS not only does not need to be in the head section, it generally should not be
  - Browser stops rendering until file is loaded
  - Best place in general is just before `</body>`

# Running When DOM is Ready

- The modern replacement for `<body onload= ... >`

```
$(document).ready(function() {  
    // code to run when DOM is ready  
});
```

Shorthand version:

```
$(function() {  
    // code to run when DOM is ready  
});
```

# DOM Ready < Content Loaded

- DOM Ready fires as soon as the HTML has been parsed into the DOM
  - There may still be images and other files to be loaded
  - Sizes of elements may not be final
- Use `.load()` instead when you want to wait for everything to be ready

# Homework

1. Create an HTML page with two sections of content.
2. Add two buttons, one to turn each section on and off.
3. Add event handlers to the buttons so clicking on that button turns on or off that section of content.
4. Make the content areas sit on top of each other, and make the buttons act like tabs.

# Help Spread the Word!

- It's not too late for your friends and colleagues to join the course
  - [www.webvanta.com/courses](http://www.webvanta.com/courses)
  - Click the share links on each of the course pages
- Help spread the word so we can develop more courses
  - Share via your favorite social media networks
  - Email a friend or a mailing list
  - Write something on your blog
- Thanks!