

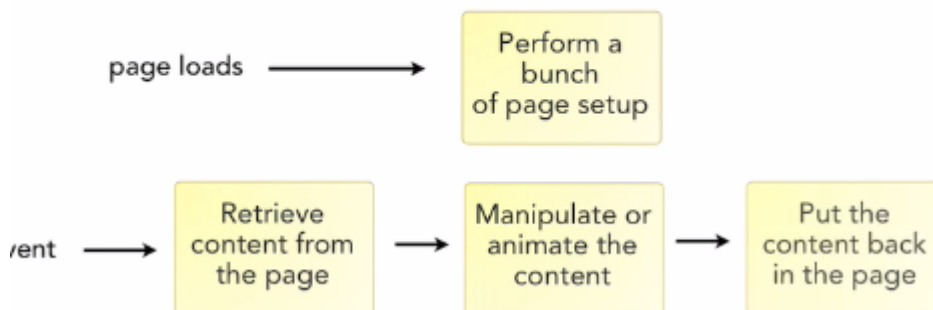
JQuery Essentials 2013

Introduction

What is JQuery?

- JavaScript Library
- User does something, now need to respond
- Leverages CSS
- Works with sets of elements
- Performs multiple operations on a set of elements with one line of code (known as statement chaining)
- Hides various browser quirks (so you can concentrate on the end result)
- Is extensible (so you can use third-party-plugins to perform specialized tasks, or write your own)
-

- Most modern web development scenarios involve common patterns



BROWSER	WORKS WITH	KNOWN ISSUES WITH
Internet Explorer	6.0 and greater	1.0 through 5.x
Safari	3 and greater	1.0 through 2.1
Chrome	1 and greater	N/A
Firefox	2 and greater	1.0.x
Opera	9 and greater	1.0 through 8.x

CHAPTER 01

Downloading JQuery (www.jquery.com)

Development – easier to debug and learn
Production – compressed so smaller/faster



CREATING A SIMPLE JQUERY-ENABLED PAGE

For our first jQuery example, we're going to keep things simple:

1. Include a reference to the jQuery library
2. Install and event handler that responds to the page loading in the browser window
3. Display an alert that indicates that the event handler was called and that jQuery is therefore working

Typically, code that you want to execute when the page is loaded is written like this:

```
function runOnLoad() {  
  alert("the page just loaded!");  
}  
window.onload = runOnLoad;
```

The on-load event only fires after *all* of the page content has downloaded, including images

It's also harder to add multiple load functions

jQuery provides a way to run code when the DOM of the page is ready

This is called the `document.ready` event

Document.ready event

- This event will occur when the entire DOM has been loaded.
- Better than `window.onload`

```
$( "document" ).ready(function() {  
  alert("The page just loaded!");  
});
```

```
} );
```

JQUERY 8 CATEGORIES

1. Core Functionality
2. Selection and Traversal
3. Manipulation and CSS
4. Events
5. Effects
6. AJAX
7. User Interface
8. Extensibility

Core Functionality Implements core jQuery functions as well as commonly used utilities	Selection and Traversal Provides functions for finding content in documents and navigating among the contents of the document	Manipulation & CSS Provides functions for editing and changing documents content and working with CSS data such as positioning info	Events Simplifies working with the modern DOM events and provides common event helper functions
Effects Provides functions for creating basic animations and effects, such as hiding and showing elements and moving objects around	Ajax Provides utilities for working with Ajax, such as loading content from pages and dealing with JSON data	User Interface Provides an official plug-in with commonly used interface widgets, like slider controls, progress bars, accordions, etc	Extensibility Enables the construction of jQuery plug-ins that enhance the functionality of the base library

jQuery Selectors and Filters: Overview

jQuery selectors and filters retrieve content from the document so it can be manipulated using other functions (think of this as the “Query” part of the “jQuery”)

- jQuery selectors return an array of objects that match the selection criteria
- jQuery filters operate on a selector to further refine the results array that the selector returns

This array is *not* a set of DOM elements

It is a collection of jQuery objects that provide a large number of predefined functions for further operating on the objects

Using Basic jQuery Selectors

- Using jQuery vs. using the plain browser DOM

:SS-style selectors and filters are based on familiar CSS syntax, and work exactly the same way as CSS does

The CSS selectors listed here correspond directly to their CSS counterparts

ELECTOR	PURPOSE
<code>tagname</code>	Finds all elements that are named <i>tagname</i>
<code>identifier</code>	Finds all elements with ID of <i>identifier</i>
<code>className</code>	Finds all elements that have class attribute with the value of <i>className</i>
<code>tag.className</code>	Gets elements of type <i>tag</i> that have a class attribute with the value of <i>className</i>
<code>tag#id.className</code>	Retrieves the <i>tag</i> element that has an ID of <i>id</i> and a class attribute with the value of <i>className</i>
<code>*</code>	Finds all of the elements on the page

Using jQuery vs. using the plain browser DOM

<code><html></code>	Get all <code><p></code> tags (DOM):
<code><head></code>	<code>document.getElementsByTagName("p");</code>
<code><title>Document</title></code>	Get all <code><p></code> tags (jQuery):
<code><head></code>	<code>\$("p");</code>
<code><body></code>	Get the tag with id "list1" (DOM):
<code><ul id="list1"></code>	<code>document.getElementById("list1");</code>
<code><li class="a">item 1</code>	Get the tag with id "list1" (jQuery):
<code><li class="a">item 2</code>	<code>\$("#list1");</code>
<code><li class="b">item 3</code>	Get all <code></code> tags with class "a" (jQuery):
<code><li class="b">item 3</code>	<code>\$("li.a");</code>
<code></code>	Get all tags with class "b", but only if they are inside a <code></code> (jQuery):
<code><p>This is paragraph 1</p></code>	<code>\$("ul .b");</code>
<code><p>This is paragraph 2</p></code>	
<code><p>This is paragraph 3</p></code>	
<code><p>This is paragraph 4</p></code>	
<code></body></code>	
<code></html></code>	

```
Get all <p> tags (jQuery):
$("p");
$("#list1");
```

Using Basic jQuery Selectors

he hierarchy and combination selectors allow you to get a little more advanced in selecting page content

ou can select elements based on hierarchical relationships or on a series of common criteria

SELECTOR	PURPOSE
<code>selector, selector, ...</code>	Finds all of the specified selectors
<code>.class1.class2</code>	Finds all elements with both <i>.class1</i> and <i>.class2</i> applied
<code>parent > child</code>	Finds all <i>child</i> elements that are direct children of elements of type <i>parent</i>
<code>ancestor descendant</code>	Finds all <i>descendant</i> elements that are contained within elements of type <i>ancestor</i>
<code>prev + next</code>	Finds all <i>next</i> elements that are next to a <i>prev</i> element
<code>prev ~ siblings</code>	Finds all sibling elements that come after <i>prev</i> and match the <i>siblings</i> selector

Select all <p> tags and put a red border

```
<script type="text/javascript" src="jquery/Exercise Files/jquery-1.3.2.js"></script>
<script>
$ ("document").ready(function() {
    $ ("p").css("border","3px solid red")
});
</script>
```

jQuery Selectors and Filters: Using Filters

- Filters work with selectors to provide even more fine-grained control over how elements are selected in the document
- jQuery filters fall into six different categories

Basic Provides basic filtering, like finding the first, last, and n- and odd-numbered elements in a returned set	Content Filters a set of elements based on the content, like whether an element contains a particular string	Visibility Filters a set of elements using the visibility setting of each element as a test
Attribute Examines a given attribute on an element to determine whether it should be filtered out	Child Selects elements based upon their relationship with their parent element	Form Provides specialized filters that operate on form elements

- You can refine a selector by including elements that match certain conditions, like position or index

ALTER	PURPOSE
first	Selects only the first instance of the selector's returned set
last	Selects only the last instance of the selector's returned set
even	Selects only even-numbered elements in the selector's returned set
odd	Selects only odd-numbered elements in the selector's returned set
eq(n)	Filters out elements that are not positioned at the given index
lt(n)	Includes elements that are past the given index
gt(n)	Includes elements that are before the given index
header	Selects all header elements (H1, H2, H3, etc)
animated	Selects all elements that are currently being animated in some way
not(selector)	Includes elements that do not match the given selector

```

<script type="text/javascript" src="jquery/Exercise Files/jquery-
1.3.2.js"></script>
<script>
$("document").ready(function() {
    $("p").css("border", "3px solid red")
});
</script>

```

