# Create HTML5 Apps with jQuery Jump Start

**Useful blogs:**

<http://rachelappel.com>

<http://codefaster.com>

<http://codefaster.com/codechat>

<http://codefaster.com/codeshow>

[www.codecademy.com/tracks/jquery](http://www.codecademy.com/tracks/jquery)

### Module 1: jQuery 101

**What is Jquery?**

* JQuery is a library. Earlier people widely used javascript but we had issues with traditional javascript not exactly with the javascript but with the DOM of the browser. So avoid these issues, John Resig came up with JQuery mostly for cross browser application events, DOM selection, and for effects and Ajax.
* In simple words, JQuery is a helper library which erases all the problems with DOM.

**What is DOM?**

* DOM is called as Document Object Model. When the browser renders HTML and other assets, it creates a tree i.e., DOM tree. We need to work with those elements in order to able to manipulate things on the page.

**Jquery is..**

* Fast, small (32k), open, feature-rich
* HTML document traversal and manipulation
* Event handling
* Animation
* Ajax
* Very compatible
* Extensible
* Cross browser

**When to use jquery?**

* Web sites
* Small to medium websites
* Brownfield applications (legacy applications) – Greenfield applications (Brand new)

**When not to use jquery?**

* As primary structure in large web apps
* When using one or two jQuery features.

**jQuery ready function:**

The Ready function wraps up all the rest of functions and methods of scripts in each page. When the DOM is loaded, we can go and start working with elements, that’s when the Ready function allows our script to run. Simply, a Ready function is a wrapper function and everything goes in the page.

**JQuery Version History:**

| **Version number** | **Release date** | **Latest update** | **Size Prod (KB)** | **Additional notes** |
| --- | --- | --- | --- | --- |
| 1.0 | August 26, 2006 |  |  | First stable release |
| 1.1 | January 14, 2007 |  |  |  |
| 1.10 | May 24, 2013 | 1.10.2 (July 3, 2013) |  | Incorporated bug fixes and differences reported from both the 1.9 and 2.0 beta cycles |
| 1.11 | January 24, 2014 | 1.11.3 (April 28, 2015) | 95.9 |  |
| 1.12 | January 8, 2016 | 1.12.4 (May 20, 2016) | 95 |  |
| 1.2 | September 10, 2007 |  |  |  |
| 1.3 | January 14, 2009 |  | 55.9 | Sizzle Selector Engine introduced into core |
| 1.4 | January 14, 2010 |  |  |  |
| 1.5 | January 31, 2011 |  |  | Deferred callback management, ajax module rewrite |
| 1.6 | May 3, 2011 |  |  | Significant performance improvements to the attr() and val() functions |
| 1.7 | November 3, 2011 | 1.7.2 (March 21, 2012) |  | New Event APIs: .on() and .off(), while the old APIs are still supported. |
| 1.8 | August 9, 2012 | 1.8.3 (November 13, 2012) | 91.4 | Sizzle Selector Engine rewritten, improved animations and $(html, props) flexibility. |
| 1.9 | January 15, 2013 | 1.9.1 (February 4, 2013) |  | Removal of deprecated interfaces and code cleanup |
| 2.0 | April 18, 2013 | 2.0.3 (July 3, 2013) | 81.1 | Dropped IE 6–8 support for performance improvements and reduction in filesize |
| 2.1 | January 24, 2014 | 2.1.4 (April 28, 2015) | 82.4 |  |
| 2.2 | January 8, 2016 | 2.2.4 (May 20, 2016) | 85.6 |  |
| 3.0[[28]](https://en.wikipedia.org/wiki/JQuery#cite_note-28) | June 9, 2016 | 3.0.0 (June 9, 2016) | 86.3 | Promises/A+ support for Deferreds, $.ajax and $.when, .data() HTML5-compatible |
| 3.1 | July 7, 2016 | 3.1.1 (September 23, 2016) | 86.3 | jQuery.readyException added, ready handler errors are now not silenced |
| 3.2 | March 16, 2017 | 3.2.1 (March 20, 2017) | 84.6 |  |

### Module 2: Selection

**JQuery Selectors:**

Selectors are the core part of the Jquery. Jquery selectors are derived from CSS Selectors.

Useful link: <http://api.jquery.com/category/selectors/>

**JQuery Selectors for:**

* Elements, by ID, attribute, name, etc.
* Attributes
* Form Elements
* Heirarchy (Parent, Child, Descendant, Sibling)
* Elements by State (Enabled, Disabled, Hidden, selected)
* Position
* Class

**Basic syntax of Selector:**

$(Selector);

$Selector.method();

**JQuery Selectors: Multiple:**

We can select multiple elements out of the DOM with Selectors

$(“selector1,selector2,selectorN”);

$(“element[attr=’value’][attr=’value2”];

Wrapped sets

$(“selector”)[0];

$(“selector”).get(0);

Add to wrapped set but not DOM

$(“selector”).add(“<div>Added content</div>”)

**Link to explanation of all Selectors:**

[**https://api.jquery.com/category/selectors/**](https://api.jquery.com/category/selectors/)

**Basic Selectors:**

$(“\*”);

$(“elementName”);

$(“#elementId”);

$(“.class”);

**Attribute Selectors:**

$(“selector[attr]”);

$(“selector[attr=’value’]”);

$(“selector[attr^’value’]”);

$(“selector[attr$’value’]”);

$(“selector[attr\*’value’]”);

**Hierarchy Selectors:**

$(“ancestor descendant”);

$(“parent elem > child elem”);

$(“prev elem + next elem”);

$(“prev sibling + next sibling”);

**Positional Selectors:**

$(“element:first”);

$(“element:last”);

**Jquery Filters:**

$(“:button”);

$(“:checkbox”);

$(“:password”);

$(“:submit”);

$(“:text”);

$(“:file”);

$(“:contains(text)”);

$(:”not(selector)”);

**Chaining:**

Allows us to run multiple commands on a wrapped set while only accessing those DOM elements once

$(“selector”).method();

$(“selector”).method1().method2();

### Module 3: Manipulating DOM

**Modifying Element Data:**

(“#p”).html(“bla bla <em>lazy</em> dog);

(“#p”).text(“bla bla”);

(#div p:first”).text(“bla bla #div div);

**Modifying inside Elements:**

$(“content”).appendTo(“selector”);

$(“content”).prependTo(“selector”);

$(“selector”).append(“<li>this is after</li>”);

$(“selector”).prepend(“<li>this is after</li>”);

**Modifying Outside Elements:**

$(“content”).insertAfter(“selector”);

$(“content”).insertBefore(“selector”);

$(“selector”).after(“<li>this is after</li>”);

$(“selector).before(“<li>this is after</li>”);

$(selector).before(content,function(index){});

**Copying Elements:**

$(“selector”).clone().appendTo(“selector2”);

**Creating new Element:**

$(“<a/>”, {

html: “This is a <strong> new </strong> link”,

“class”:”link”,

href:”page.html”

});

**Removing Elements:**

$(“selector”).remove();

$(“selector”).detach();

$(“selector”).empty();

**Replacing Elements:**

$(“selector”).replaceAll(“content”);

$(“selector”).replaceWith(“content”)

**Wrapping Elements:**

$(“selector”).wrap();

$(“selector”).wrapAll();

$(“selector”).wrapInner();

$(“selector”).unwrap();

**Modifying Attributes:**

$(“selector”).attr(“name”,”value”);

$(“selector”).prop(“name”,”value”);

$(“selector”).removeAttr(“name”);

$(“selector”).removeProp(“prop”);

$(“selector”).attr(“name”, function(attr) {…});

**Style Elements:**

$(“selector”).CSS(“style”,”value”);

$(“selector”).CSS({“style”:”value”,” style”:”value”});

$(“selector”).CSS({“style”, function(index, value) {…});

**Styling & Dimensions:**

$(“selector”).hasClass(“className”);

$(“selector”).addClass(“className”);

$(“selector”).addClass(“className className2”);

$(“selector”).addClass(“className”).removeClass(“className”);

$(“selector”).removeClass(“className”);

$(“selector”).toggleClass(“className className2”);

$(“selector”).width(“100px”);

$(“selector”).width();

$(“selector”).height(“100px”);

$(“selector”).height();

$(“selector”).position();

### Module 4: Events

**Events Overview:**

* Attachment and the event Object
* Browser Events
* Document Loading events
* Form Events
* Keyboard events
* Mouse events

**Attaching to events:**

**.on() and .off()**

.bind() and .unbind() -- deprecated

.delegate() and .undelegated() -- deprecated

.one() -- deprecated

.live() and .die() -- deprecated

**.trigger() and .triggerHandler()**

**The Event Object:**

* Holds all of the event properties
* Helpful properties include

.currentTarget

.type

.which

.timestamp

.target

.data

Browser Events:

.error(): Called if an element is not loaded correctly

.resize(): when the size of the browser window changes

.scroll(): When the user scrolls to a different place in an element

**Document Loading Events:**

.load(): When an element is being loaded

.ready(): When the DOM is loaded and ready (minus content like images). Use $(function) as a shortcut

.unload(): Opposite of load, when an element is being loaded.

**Form Events:**

**.change():** When an element value changes

**.focus():** When an element receives focus

**.blur():** when an element loses focus

**.focusin():** When an element or any nested elements receives focus

.**focusout():** When an element loses focus

**.select():** When the user makes a text selection inside

**.submit():**When the user is attempting to submit a form

**Keyboard Events:**

**.keydown():** When a key is pressed down

**.keypress():** When the browser registers keyboard input (may vary by browser). Not triggered by Shift, Esc, Delete, etc

**.keyup():** When a key returns to the up position

**Mouse Events:**

.click()

.dblclick()

.hover()

.mousedown()

.mouseenter()

.mouseleave()

.mousemove()

.mouseout()

.mouseover()

.mouseup()

### Module 5: Effects

**Basic Effects:**

.hide()

.show()

**.toggle():** If it’s showing, hide it. If its hiding, show it

**Fading:**

Fading adjusts the opacity of the selected element

.fadeIn()

.fadeOut()

**.fadeTo():** Set the opacity to a set value

**.fadeToggle():** If it’s showing, fade it out. If it’s hidden, fade it in

**Sliding:**

**.slideUp():** Hides an element sing a sliding motion

**.slideDown():** Shows an element using a sliding motion

**.slideToggle():** If it’s showing, then hide it,. If it’s hiding, then show it.

**Custom Effects:**

**.animate():** Performs a custom animation of a set of CSS properties

**.queue(), .dequeue(), and .clearQueue():** Control the queue of functions to be executed on the matched elements

**.delay():** Set a timer to delay execution of subsequent items in the queue.

**.stop():** Stop the animation

**.finish():** Stop the animation and clear queue

### Module 6: Ajax and Async

**Ajax Overview:**

* Global Ajax Event Handlers
* Helper Functions
* Low-Level Interface
* Shorthand Methods
* Differed Object

**Global Ajax Event handlers:**

.ajaxSend()

.ajaxStart()

.ajaxStop()

.ajaxComplete()

.ajaxSuccess()

.ajaxError()

**Helper Functions:**

$.param()

.serialize()

.serializeArray()

**Low Level Interface:**

**$.ajax():** Perform an asynchronous HTTP (ajax) request

**$.ajaxPrefilter():** Handle custom ajax options before requests are sent

**$.ajaxSetup():** Set default values for future ajax requests (not recommended)

**$.ajaxTransport():** Allows for ajax transmission of custom data types.

**Shorthand Methods:**

**$.get():** Load data from the server using a HTTP GET request

**$.post():** Load data from the server using a HTTP POST request

**$.getJSON():** Load JSON-encoded data from the server using a HTTP GET request

**$.getScript():** Load a JS file from the server using a HTTP GET request and then execute it

**.load():** Load data from the server and place the returned HTML into the matched element

**Deferred Object:**

The Deferred object can register and invoke callback queues and relay the callbacks’ success or failure asynchronously. It can be chained, but is a different object than $.

$.Deferred ()

deferred.promise()

deferred.then() and .done()

deferred.fail() and .always()

$.when()

.promise()

### Module 7 : Libraries (jQuery Libraries JavaScript Libraries)

**Useful Links:**

<https://jqueryui.com/>

<http://microjs.com/#>

<http://www.datejs.com/>

<https://momentjs.com/>

**jQuery UI**

* CSS & JavaScript libraries for Web UI

1. Widgets
2. CSS
3. Script

* Animations
* Drag and Drop
* Widgets
* Accordion
* Autocomplete
* Button
* Datepicker
* Dialog
* Menu
* Progressbar
* Selectmenu
* Slider
* Spinner
* Tabs
* Tooltip

**jQuery Mobile:**

* CSS & JavaScript for mobile web development
* Touch optimized
* Data attributes
* Themes

**JavaScript Libraries:**

* YUI
* Mootools
* Prototype
* momentjs
* jquery.complexify.js
* justgage.com
* three.js

**Useful links for jQuery Mobile:**

<http://jqueryui.com>

<http://jquerymobile.com/>

<http://themeroller.jquerymobile.com/>

<http://api.jquerymobile.com/1.3/data-attribute/>