



THE DOM & JQUERY

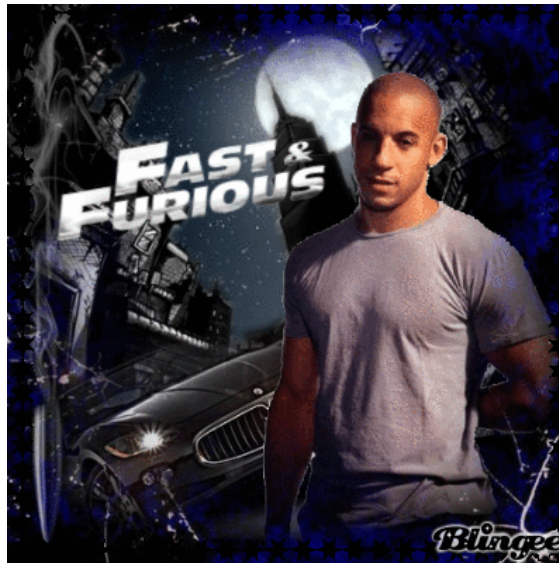
Code 301

Welcome to class!

AGENDA

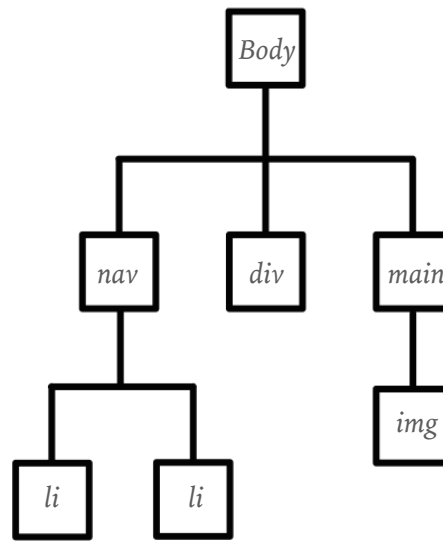
- Quick DOM Review
- jQuery

THE DOM



NOPE.

This guy's name is Dom. This is not the DOM we are looking for.



YUP.

That's better. If not as fast or furious.

THE DOM

- What is the DOM?
 - Document Object Model
 - What the browser does with HTML
 - Internal representation of a web page

So tell me: What is The DOM?

This is a great interview question, you should have an answer ready.

Collect a few student responses.

“

The DOM states what your script can ask the browser about the current page, and how to tell the browser to update what is being shown to the user.

-Jon Duckett

This is the first part of MVC: The DOM is the View.

But we aren't updating a user's screen pixel by pixel, thankfully! The DOM serves as a layer of abstraction over how our page is presented.

So how do we use it? What does DOM access allow us to do?

THE DOM

- What can you do with the DOM?
 - Find elements:
 - `var el = document.getElementById('articles');`
 - `var els = document.getElementsByTagName('li');`
 - `var els = document.querySelectorAll('#articles > li');`
 - Change elements:
 - `document.getElementById('articles').innerHTML = '<h1>News</h1>';`
 - `document.getElementById('articles').textContent = 'News';`
 - `document.getElementById('articles').innerText = 'News1';`
 - Also:
 - Add and remove elements
 - Read attributes and values
 - Navigate the tree... Manage user interactions, events, etc...

There is lots of very useful stuff you can DO with the DOM.

In fact, client side web apps are totally reliant on this ability.

Wouldn't it be nice if it was a little less clunky?

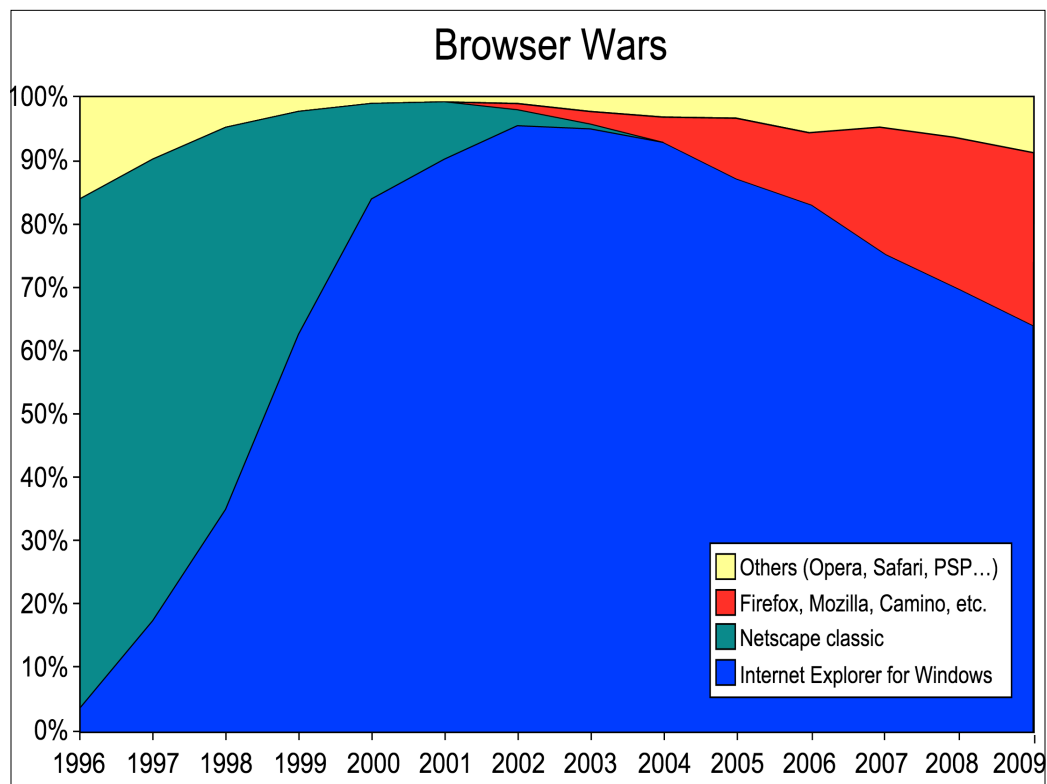
A little more efficient for devs?

Easier to work with?

Supported the same features across different browsers?

JQUERY

Let me introduce you to my friend jQuery!



Born from the Browser Wars in 2006, jQuery came to the aid of web developers everywhere to help with writing more expressive code. Code that could rise above the `document.getElementById` weeds.

JQUERY

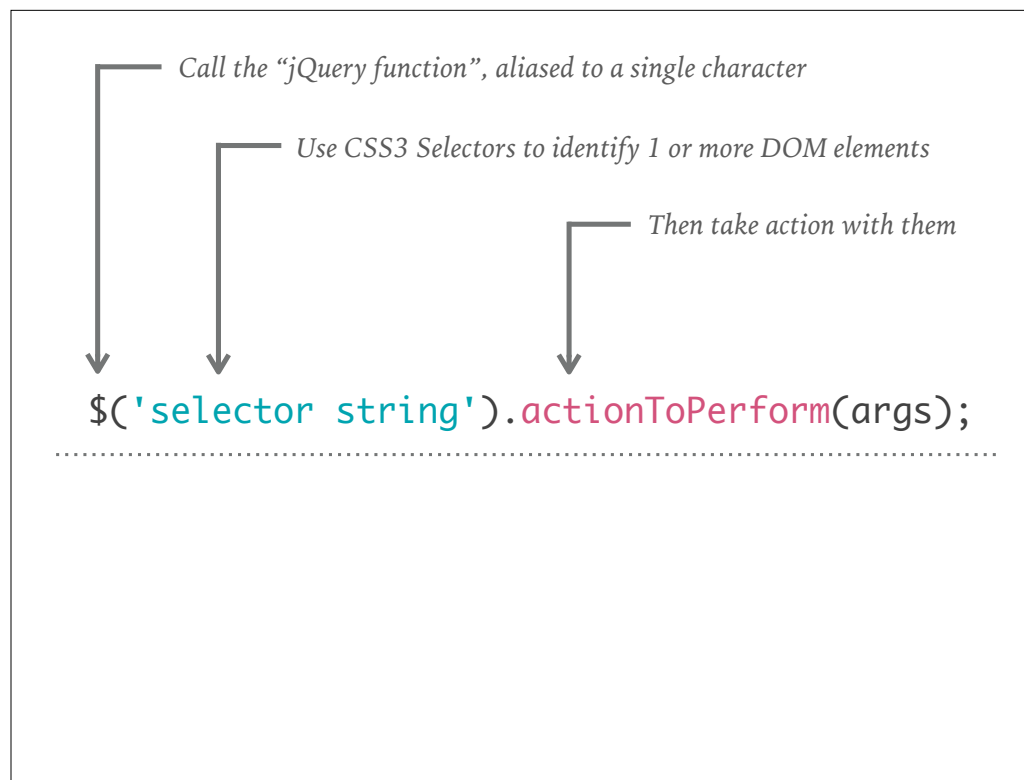
- What is it, exactly?
 - A JavaScript library (6,358 LOC)
 - That provides an API for:
 - DOM selection
 - DOM manipulation
 - DOM traversal
 - Event management
 - AJAX
 - ...and more

Created in 2006 by John Resig. jQuery offered web developers a single, unified API for accessing the DOM, and doing things with it.

Know what I mean by API?

Any DOM-related JS suddenly became more expressive, faster to work with, more compatible, more comprehensible.

That's a lot of promises! How does it actually work?



The entire jQuery API is available behind a single function call: `jQuery()` or `$()`.

The `$` is your gateway into the DOM.

Use any CSS selector string you like. jQuery even adds some other handy selectors.

Then you can decide what to do with your selection. Clone it? Hide it? Animate it? Move it elsewhere? Attach an event? You have lots of options.

VANILLA JAVASCRIPT: FADE IN

```
1 function fadeIn(el) {
2   el.style.opacity = 0;
3
4   var last = +new Date();
5   var tick = function() {
6     el.style.opacity = +el.style.opacity + (new Date() - last) / 400;
7     last = +new Date();
8
9     if (+el.style.opacity < 1) {
10      (window.requestAnimationFrame && requestAnimationFrame(tick)) ||
11        setTimeout(tick, 16)
12    }
13  };
14  tick();
15 }
16
17 fadeIn(el);
```

JQUERY: FADE IN

```
1 $(el).fadeIn();
```

http://youmightnotneedjquery.com/#fade_in

You may be thinking... “But I know how to do all that DOM stuff already, is this really going to help me?”

Let’s take a look at some examples from <http://youmightnotneedjquery.com/>, which offers some vanilla js translations of common jQuery functions.

Up first: Make a DOM element slowly appear on the page...

VANILLA JAVASCRIPT: FILTER

```
1 Array.prototype.filter.call(document.querySelectorAll(selector), filterFn);
```

JQUERY: FILTER

```
1 $(selector).filter(filterFn);
```

http://youmightnotneedjquery.com/#fade_in

You may be thinking... “But I know how to do DOM stuff already, why do I need jQuery?”

Let’s take a look at some examples from <http://youmightnotneedjquery.com/>, which offers some vanilla js translations of common jQuery functions.

Up first: Make a DOM element slowly appear on the page...

VANILLA JAVASCRIPT: MATCHES SELECTOR?

```
1 var matches = function(el, selector) {  
2   return (el.matches  
           || el.matchesSelector  
           || el.msMatchesSelector  
           || el.mozMatchesSelector  
           || el.webkitMatchesSelector  
           || el.oMatchesSelector).call(el, selector);  
3 };  
4  
5 matches(el, '.my-class');
```

JQUERY: MATCHES SELECTOR?

```
1 $(el).is('.my-class');
```

http://youmightnotneedjquery.com/#matches_selector

Browser-compatibility-hell: resolved.

VANILLA JAVASCRIPT: TOGGLE A CLASS

```
1 if (el.classList) {  
2   el.classList.toggle(className);  
3 } else {  
4   var classes = el.className.split(' ');  
5   var existingIndex = classes.indexOf(className);  
6  
7   if (existingIndex >= 0)  
8     classes.splice(existingIndex, 1);  
9   else  
10    classes.push(className);  
11  
12   el.className = classes.join(' ');  
13 }
```

JQUERY: TOGGLE A CLASS

```
1 $(el).toggleClass(className);
```

http://youmightnotneedjquery.com/#toggle_class

Expressive, isn't it! When you look at this code 2 weeks from now, you will instantly know what it does.

VANILLA JAVASCRIPT: GET JSON FROM A SERVER

```
1 var request = new XMLHttpRequest();
2 request.open('GET', '/my/url', true);
4 request.onload = function() {
5   if (request.status >= 200 && request.status < 400) {
6     // Success!
7     var data = JSON.parse(request.responseText);
8   } else {
9     // Server returned an error
10  }
11 };
12 request.onerror = function() {
13   // There was a connection error
14 };
15 request.send();
```

JQUERY: GET JSON FROM A SERVER

```
1 $.getJSON('/my/url', function(data) {
2   // Do something with the data
3 });
```

<http://youmightnotneedjquery.com/#json>

JQUERY CHEAT SHEET

<http://oscarotero.com/jquery/>

Here's a great resource to keep in front of you.

Look how this page is organized. Each colored column identifies another main thing you can do with jQuery: Selectors, manipulations, traversal, events, effects, ajax, etc.

Click anything to see the jQuery docs: which are pretty good, even if they do take a little getting used to.

Looks like a lot? It is... but this is all there is to it. This is the entirety of the jQuery API.

JQUERY IN ACTION

Let's walk through some of the examples from the book, to see jQuery in action. The book has great examples, you should "Play along" as you read.

Download the code from: <http://javascriptbook.com/code/>

Unzip, move to your ~/projects/ dir, and cd into the folder.

Run `live-server` to see the index of all the chapters. We are in Chapter 7.

Highlights: Watch how jQuery works as "syntactic sugar" for DOM management. Let's you work with a whole collection of "enriched" DOM nodes. Allows chaining.

JQUERY

- From jquery.com:
- jQuery is a fast, small, and feature-rich JavaScript library.
- It makes things like HTML document traversal and manipulation, event handling, animation, and Ajax much simpler with an easy-to-use API that works across a multitude of browsers.

jQuery is a fast, small, and feature-rich JavaScript library.



It makes things like HTML document traversal and manipulation, event handling, animation, and Ajax much simpler with an easy-to-use API that works across a multitude of browsers.

With a combination of versatility and extensibility, jQuery has changed the way that millions of people write JavaScript.

-jquery.com

In summary, here's a few words from jquery.com

RECAP

RECAP

- You need work closely with the DOM
- Your app is logic to dictate what the DOM looks like, and what it does
- jQuery will boost your DOM skills
- ...which makes your apps more maintainable, and more fun to write.

RESOURCES AND ATTRIBUTIONS

- <https://blingee.com/blingee/view/113822655-Dom-Fast-Furious->
- https://en.wikipedia.org/wiki/Browser_wars