JQUERY:
BUILDING BLOCKS

Lesson 1 Handout

Welcome To jQuery

ABOUT THIS HANDOUT

This handout includes the following:

- A list of the core concepts covered in this lesson.
- The assignment(s) for this lesson.
- A list of readings and resources for this lesson including books, articles and websites mentioned in the videos by the instructor, plus bonus readings and resources hand-picked by the instructor.
- A transcript of the lecture videos for this lesson

CORE CONCEPTS

At its most basic, jQuery helps you quickly find HTML elements in your document, and easily do something to them. jQuery helps you out in four fundamental ways:

1. jQuery helps you avoid annoyance. It takes tasks that are complex and annoying to program – like building cross-platform code to iterate through every element in your document and find all the ones with a certain class – and makes this incredibly easy. For example, this syntax performs that task:
   `$('.certain-class')`
2. jQuery hides cross-browser issues, both with out-of-date browsers and with new, not-yet-universal features. Old versions of IE still in widespread use handle transparency, for example, in a completely different, nonstandard way, but jQuery takes care of it for you. Meanwhile, new features like CSS animations require handwritten JavaScript fallbacks when run in browsers more than a year or two old; again, jQuery takes care of that for you.
3. jQuery makes hard things easy. For example, animating anything smoothly requires sophisticated "Bézier curve" math, which jQuery includes under the hood so you don't have to learn it.
4. jQuery does... everything else. Its simple but sophisticated plugin architecture has given rise to an enormous ecosystem of plugins that can do just about anything you need them to do. In this course we'll be looking at two – a slideshow creator and a Twitter plugin – as well as showing you how to write your own.
   - jQuery has become the most popular JavaScript library because its simple, elegant syntax hides a ton of power and optimization, while its plugin ecosystem means that the basics are just the beginning. Furthermore, its popularity with web developers has led to a popularity with employers, too.
   - At its most basic, jQuery helps you quickly find HTML elements in your document, and easily do something with them. To find elements, you pass a selector string into the jQuery function (`$`). jQuery selectors are the same as CSS selectors. For example:
     - Select all image elements: `$('img')`.
     - To select by ID, use a hashtag symbol: `$('#element-id')`.
     - To select by class name, use a period: `$('.class-name')`.

- As in CSS, you can string selectors together to select children, descendants, etc.: `$('.class-name-1.class-name-2 > img')`.
- The jQuery function returns a jQuery array of matching elements, which lets you perform actions or do further selections: `$('.gallery-items').first()`.

ASSIGNMENTS

1. Quiz
2. Review basic and hierarchical selector syntax at http://oscarotero.com/jquery/.

RESOURCES

- Andrée's introduction to jQuery is excellent. So far we've covered the first three sections.
  http://andreehansson.se/the-basics-of-jquery/

- Since jQuery selectors are a superset of CSS selectors, it's a good idea to keep a CSS cheat sheet around for reference. This is an excellent one.
  http://www.addedbytes.com/download/css-cheat-sheet-v2/png/

INTRODUCTION

(Note: This is an edited transcript of the jQuery: Building Blocks lecture videos. Some students work better with written material than by watching videos alone, so we're offering this to you as an optional, helpful resource. Some elements of the instruction, like live coding, can't be recreated in a document like this one.)

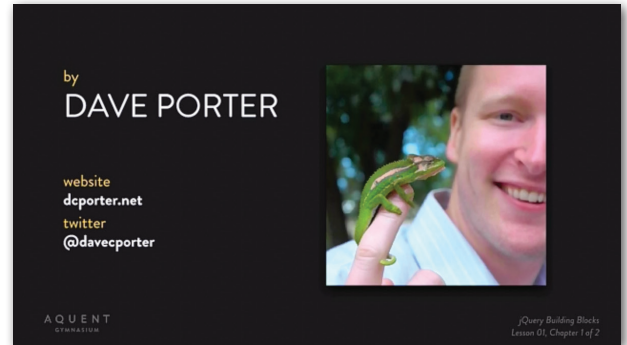Hello. And welcome to an Aquent Gymnasium production of jQuery building blocks, five ways to cut your web development time in half.

My name's Dave Porter. I'm a web developer here in sunny Boston, Massachusetts. I've been doing this for about 10 years, so that's since back before the dawn of jQuery. I've got plenty of stories from the good old days that I'm sure I'll tell you. I'm currently spending most of my time doing rich web application work with SproutCore, which is a rich web application development framework. It's very powerful, and it's built on top of jQuery. You can visit my website at dcporter.net. As of this recording, that is a half-built at best. Or you can follow me on Twitter @davecporter.

This course is a practical introduction to jQuery for folks who are familiar with the basics of HTML, CSS, and JavaScript. I'm assuming that this is not your first time behind the keyboard, that you are familiar with the big three web technologies, but that you're new to jQuery, and that you're looking to learn to do more interesting things. You're looking to learn to do them faster. And you want to learn it quickly.

So with those things in mind, over the course of the next six lessons, I'm going to teach you how to use jQuery. I'm going to teach you five specific techniques for getting your projects into production faster, and to make those projects do more interesting, more dynamic sorts of things.

Here in Lesson 1, I'm going to introduce you to the concepts and benefits of jQuery at a high level. And we'll go over some basic syntax toward the end of the lesson. We'll continue syntax in Lesson 2. I'll show you how easy it is with jQuery to add those little touches, the little flourishes, that make a website really stand out.

In Lesson 3, we're going to dive into the vast world of jQuery plug-ins. If there's something that you're trying to do, odds are pretty good that somewhere out there, there's a plug-in that can help you do it. Well be focusing on doing a photo slide show with a carousel plug-in. The reason we're focusing on the slide show motif is that at some point in your career, you're guaranteed to come across a project with a need for a slide show or a carousel. They're on the front page of every corporate website out there. They're very common.
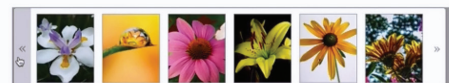
And there are other things you can do with a carousel besides just a photo slide show, so we'll touch on some of those, too.

But it's important for me to show you that, thanks to jQuery, and thanks to its plug-in ecosystem, this is a five-minute requirement project, not a five-hour requirement.

In Lesson 4, we're going to continue our look at plug-ins by showing you how to take code that you don't ever want to look at again, because it's quirky or complicated or took you a long time to bang out all the details. And you can wrap it up in your own plug-in and distribute it to your team, like, if it's something that's going to be reused across your application a lot, like a navigation bar that's just a little too dynamic for a simple Server Side Include, you can wrap it up in a jQuery plug-in, distribute it out to your team. They won't have to worry about all the quirks in the code or worry about getting it wrong. They're not going to have to copy/paste it, which is great, because copy/pasting is a sure-fire way to ensure that something's going to go wrong later on with your project. And you're going to look like a hero.

In Lesson 5, which I'm calling Introspective Dynamism, I'm going to show you how jQuery gives you the tools to react to your users' actions on the fly instead of waiting for a server call. We're going to be doing that in the context of form validation, which is prosaic but very practical.

And finally, in Lesson 6, we're going to turn to external dynamism. The Internet is just a vast world of information. Right? That's the whole point. And a lot of that information is structured in such a way that you can pull it into your web page and use it any way you want to. And once you've got a handle on updating your page on the fly, just the whole other world of dynamism becomes available to you.

I'd like to touch briefly on the very minimal software requirements for this course. First off, you need a browser. I'll be using Google Chrome. It's got really impressive built-in developer tools, which let you look at the structure of your web page, lets you look at the code as it runs. It's really easy to use, and it's really powerful. So I'll be using that and showing you how. If you're already familiar with something else, like Firebug or Firefox, feel free to use that instead. You should be able to follow along, no problem. But if you're not familiar with something already, then please make sure you have a copy of Chrome installed on your machine.

Second, not in this lesson, but starting next lesson, you'll need an IDE. That's an Integrated Development Environment. Basically, it's a fancy text editor for your code. It helps with syntax and things, makes sure you got semicolons in the right place. I personally use Sublime Text 2. Everyone I know uses it. It's awesome. It's super speedy, very flexible, and it's free for a trial period. So again, if you don't have an IDE that you're familiar with, then please download Sublime Text 2 in time for your next class. If you do have one you like then, cheers, carry on with it.
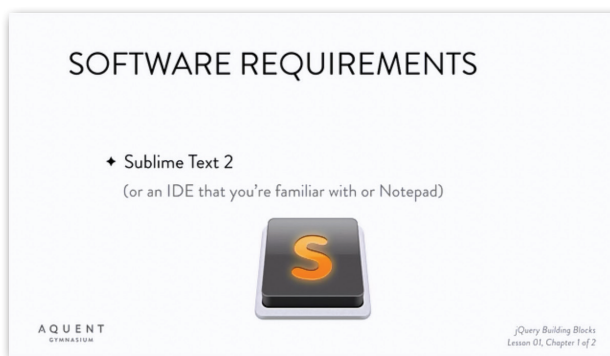
And so, without further ado, Lesson 1. Welcome to the thing that you're here to learn about. jQuery's tag line is, "Write less, do more." And you're going to find yourself writing a lot less and doing a lot more. Per jQuery's website, it is a fast, small, and feature-rich JavaScript library, helps you out with a ton of different things, works across a ton of different browsers. And it's changed the way that millions of people write JavaScript.

And that's all true advertising, especially the last bit. jQuery has really taken over the web by virtue of its simplicity, extensibility, and just general awesomeness. At its core, jQuery is an easy way to do a lot, whether you're building a personal portfolio for yourself, or whether you have a boss or a client that needs a blog for the corporate site, you're going to be using jQuery to really improve the quality of the output and the speed with which you get there. OK. But, how?

The first way is probably the most important way in terms of the amount of time it's going to save you. It's also the first way that you're going to forget about. In a couple of years, you're not going to have any memory of how much jQuery is helping you out, and that's a good thing. jQuery helps you avoid annoyance. There's so much code in plain old JavaScript and HTML that's a pain to write. jQuery takes things that you're going to do often, and it makes them really easy. Let's take a look at an example.

Here's a generic snippet of HTML for a generic web page. We've got a header at the top, a big, splashy title. We got a body section with some elements scattered around. Now let's say we need to, for whatever reason, change the text and the header, which means that we need to get that element. And here's how you do it in plain old JavaScript: document, dot, getElementById, whatever the ID is, dot, innerText, equals whatever.
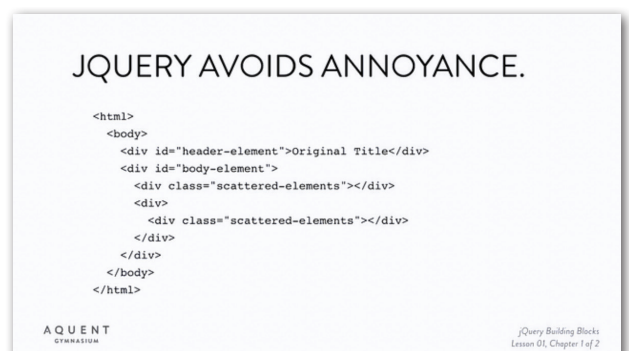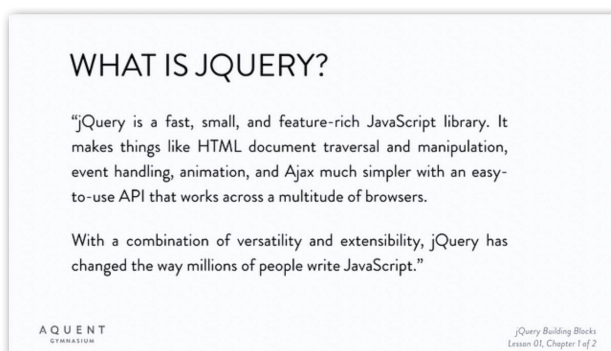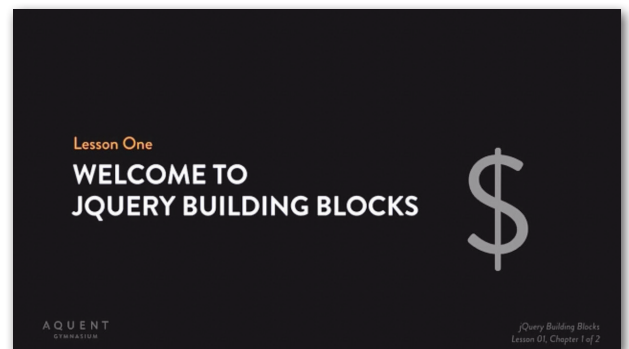
And that's not bad. That's not terrible. I mean, the fact that ID is capitalized differently here than anywhere else in the known universe may have tripped me up a couple of times when I was a beginner, I'll cop to it. And the fact that its inner text is a little weird. But whatever. It's not so bad.

Here it is in jQuery. It's a little bit shorter, a little more concise. There are a few odd little things that you have to remember. So there it is. Now let's say you have to select all of the elements in your document that have a particular class. You need to change their color, you need to make them pop out at the user a little bit or something.

Now there is no "get element by class" method in plain old JavaScript. There is something that you can use in brand new browsers. I think it just got put in late last year or this year, which jQuery sort of pushed to have in there, by the way, which you can use. But you can't be sure that everyone that comes to your website is going to have it, because it's brand-new. So if you're going to build this thing yourself, you're really going to have to do it.

GET ELEMENT BY ID

```
$('#header-element').text('Lesson 1 Portfolio');
                        (jQuery)
```

AQUENT
GYMNASIUM

jQuery Building Blocks
Lesson 01, Chapter 1 of 2

Now what does that method look like? Well, you're going to have to recursively comb through every single element of every single element in every single part of your document, which means you're going to have to know what the heck recursion is. You're going to have to have enough experience with string searches to know that if you don't do it the right way, you're going to risk getting back false positives, because strings can be contained within strings. And you're going to have this big chunk of code that you wrote while you were annoyed, because it's annoying to write, which you're not going to feel like revisiting. It's probably going to have some bugs in it.

And, just, why bother? Because here it is in jQuery. One incredibly simple command handles all the stuff that you don't feel like dealing with. And the code that does it in jQuery wasn't written by you in five minutes when you wished you were doing something else. It was written by thousands of developers, all over the planet, who are all super proud about making it as fast and optimized as they possibly can. They're going to bang on it a million times. They're going to make sure there's no bugs in it. They're going to take that brand-new, super-speedy, native method that I mentioned before, that browsers have finally put in, and they're going to detect it. And they're going to use it if it's there. It's all stuff that you just don't have to know about any of it. And it's great.

So jQuery helps you avoid annoyance, whether you're selecting an element by ID—again, not that bad, but it helps a little bit—whether you're selecting an element by class, and this is where it starts to get crucial and really wonderful. Let's say you're doing something like a simple animation. Let's say you're moving something from over here to 20 pixels over there, over the course of a second, and there's math in there that

GET ELEMENT BY CLASS

```
function getElementsByClass(className, context) {
    if (!context) context = document.body;
    var ret = [];
    var i = 0, len = context.childNodes.length, node, isMatch;
    for (i = 0; i < 0; i++) {
        node = context.childNode[i];
        isMatch = false;
        // Test node.
        if (node.className.substr(0, className.length + 1) === className + ' ') isMatch = true;
        if (node.className.substr(node.className.length - className.length) === ' ' + className) isMatch = true
        if (node.className.indexOf(' ' + className + ' ') !== -1) isMatch = true;
        if (isMatch) ret.push(node);
        // Get results from node's children.
        if (node.childNodes.length) ret.push.apply(ret, getElementsByClass(className, node));
    }
    return ret;
}
// Or just install and use jQuery.
```

AQUENT
GYMNASIUM

(JavaScript)

jQuery Building Blocks
Lesson 01, Chapter 1 of 2

you just don't feel like doing. You have to set up a bunch of timers, fire it off, and you just don't want to deal with it. jQuery makes it quick and easy.

And then there's Ajax. Ajax is last decade's sort of buzzword for reaching out to the Internet, grabbing a document, pulling it back, and using it in your web page without reloading. It's a great capability, but it's annoying to deal with. And jQuery wraps it up in a nice, simple API that we'll be touching lightly on in Lesson 6.

Next, and this is the second-most important thing that jQuery does for you, jQuery hides browser issues. There are just so many browser issues. You know? It's better than it used to be, guys, but it's not great. We're looking at all the versions of IE. Internet Explorer 6, that's still around. We've got IE7. We've got IE8. We've got Compatibility Modes that failover, depending on whether you've got the doctype set up correctly, that you just don't want to deal with all of that. So jQuery, in a lot of cases, will wrap up those differences and hide them completely from you.

Then at the other end of the spectrum, there's new browsers. Right now is the best time in history to be in the web market. Things are just moving so quickly. But that means that folks coming to your website aren't guaranteed to have the latest and greatest of every technology. So that means if you want to use some shiny new feature like CSS animation, or like local storage, you need to design a fallback. You need a way so that, if the user comes by without the most up-to-date browser, your website doesn't just completely fall apart.

And jQuery, in a lot of cases, will wrap up those new features with internal, hidden, kind of best-practice fallbacks, that you don't have to worry about. Something as simple as, "I'm not kidding, when is my page ready?" is handled differently across different browsers. There are little differences and quirks between them, even between ones that look similar. jQuery wraps that all up in one simple, reliable event, that fires at a reliable time across all browsers and it's great. And Ajax, especially old versions of IE, where Ajax just worked completely differently.

And there are weird little things. Like there's this version of Chrome for Android phones, where the Ajax system just throws a different status code (and this weird, little edge case) than every other version of the browser. And that's stuff you don't want to have to memorize and deal with. And jQuery wraps most of that up and makes it so you don't have to deal with it.
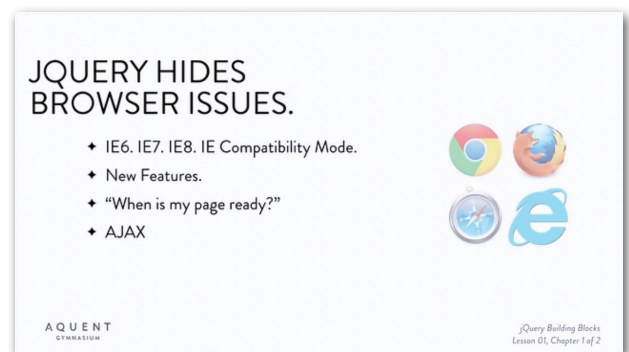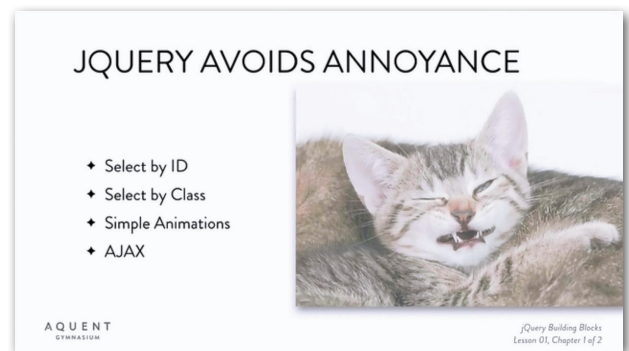
This is the third thing that jQuery does for you. It takes code that is legitimately sophisticated, and thanks to the magic of abstraction and jQuery's awesome, streamlined, abstract syntax, it wraps it all up for you in a way that's incredibly easy to use.

For example, do you remember the linear animations that I was talking about a couple of minutes ago? Those are annoying to work with, like I said, but they also look bad. If you want something that looks natural, you're going to need it to start slow, speed up, then slow back down at the end, like these guys are doing.

But then you're not just talking about linear animations anymore. You're talking about something called Bezier curves. And the code for Bezier curves is just, I mean, I have no idea what's going on here. And I don't want to. I don't have enough space in my brain for all of this. So forget about it.

What you want is, you want someone with a Ph.D. in something that I can't even pronounce, to go and write this code for you and then wrap it up and make it super-simple to use. And that's it. It's about that simple.

This is a demo from themeforest.net. We're not actually going to be going over this demo in this course, but after a couple lessons, you're going to be able to go out to the Internet, track down this tutorial, and add this awesome, parallax animation effect to your blog, just this easily.
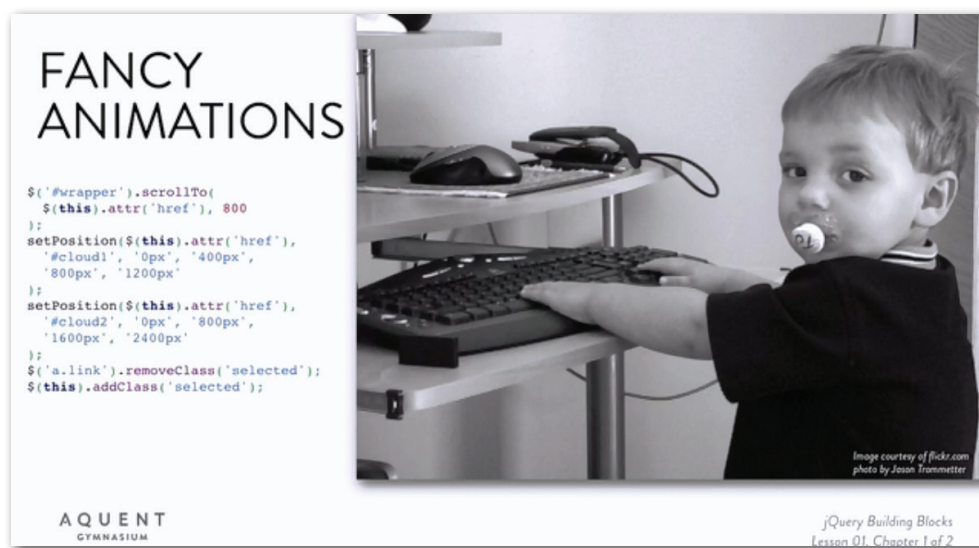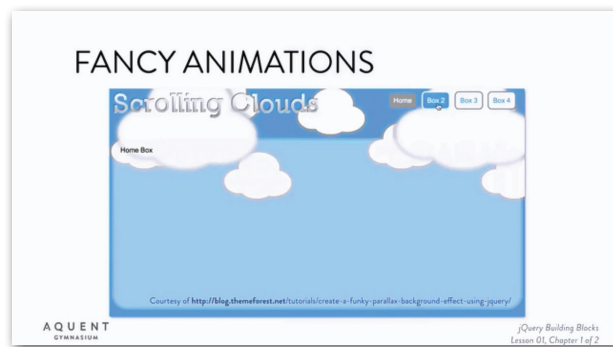
Now this is a more, kind of, aesthetic kind of sophistication. This is a slide show, where the challenge is to take these things that are difficult to fit together, and make them look seamless, make them look like it was really easy. I built one of these back in the day, pre-jQuery. And it's complex. It's tough to make these things move together seamlessly and all show up at the right time. But all you have to worry about is listing your images and making a call.

Now here's a widget that we're going to check in on again later in this lesson. And then we're going to take a real good, close look at it in Lesson 3. I'm going to teach you all about it.

So jQuery makes hard things easy. It wraps up complex math. It wraps up complex widgets. And of course, it wraps up Ajax.

Now the last thing that jQuery does for you is everything else. It's got a really well-done plug-in architecture and a really massive plug-in ecosystem. So if you have a problem, chances are that jQuery has a solution. For example, if you need some really well-done plug-and-play user interface widgets with easy drag and drop—I mean, drag and drop is just really touchy to build yourself—there's a plug-in called jQuery UI. It comes with a ton of prefab widgets that are this easy to put together.

We'll be taking a little bit of a closer look at this in Lesson 3. Now here's this guy again, the slide show from earlier. It's actually a plug-in. But along with doing simple slide shows— whoa. So I think both of these effects were inspired by the big drop blocks from Super Mario Brothers. So in proper homage, here's your code.

This plug-in just kind of makes all the different kind of carousel work that you might need to do, really easy. You just include the plug-in. You make some easy declarative calls like this. And you're good to go. Now that's kind of a small example. jQuery plug-ins can be massive, too. jQuery Mobile is an entire platform, basically, for building mobile websites. Really, like, you build your entire website and it really takes it over in a way that gives you easy access to this kind of mobile app-feeling features.

And they can range all the way down to the very small. jTweets is a very focused plug-in that I wrote that does absolutely nothing but reach out and fetch tweets for you. Now, it's still fairly complicated code. It embodies a lot of research that I did into how the Twitter API works. But that's all stuff I don't need to know anymore, because any time I want tweets in my web page, which we're going to touch on in Lesson 6, then this is all you need to do.

So jQuery extends, whether you're looking for a library of great user interface widgets, whether you're looking for an entire mobile platform to build your website on, or whether you're looking for something focused, like some kind of specialized Ajax.
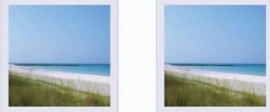
So why are we talking about jQuery instead of something else? Right? Where did jQuery come from? There were a lot of forces at work in the middle of the last decade that meant that something was going to emerge to kind of handle browser weirdnesses and to start innovating where the browsers weren't innovating at the time. But why jQuery?

Well, one reason is its simple and elegant syntax, which I'm going to show you a little bit of in a couple of minutes. But I think you're going to really appreciate the syntax in the next couple of lessons. Now inside that syntax, you've got a lot of power. The jQuery syntax really lends itself to abstracting complexity, way behind these really simple, declarative commands. And thanks to whatever witchcraft gave John Resig the foresight to build this incredible plug-in architecture, loads of people have built stuff for it. So it's got this giant, extensible ecosystem.

And of course there are competitors. Like, there's Prototype. I know that Apple uses that on its front page on apple.com. So that can't be too shabby. MooTools, that exists. YUI came out of Yahoo real early on. And they've been iterating on it for a long time. Bootstrap is out of Twitter, really recently actually. That's pretty fresh. And these are good. Right? They're solid.

But none of them have really taken over the web the way that jQuery has. And that means that nothing has really taken over the job market like jQuery either, by the way. Another reason to learn jQuery is that employers really like it. I jumped on a popular job site yesterday, pulled up some numbers. So let's see. Just that want Prototype, 59. All right. That's not bad. MooTools was 24. Yahoo was 86. That's solid. Yahoo must be hiring. And then Bootstrap was zero. That's tough. Bootstrap is pretty cool. It deserved a little better than that. But jQuery was 1,100 jobs. That's a couple orders of magnitude over anyone else on here.

So in conclusion, what can jQuery do for you? At its most basic, it's going to go out and get elements and let you do stuff to them. It does widgets really easily. It does dynamicness, changing your page on the fly, really easily. It extends really easily. You can grab one of the million different plugins from the Internet and use it. You can wrap your own plug-in and pass it out to your team.

And key point, since it's so popular, it helps you find a job. jQuery helps you write less and do more. So I'm going to take the last few minutes of the lesson and we're going to go over some introductory jQuery selector syntax.

If you could open up your lesson materials, and go to the portfolio page. Just open it up in Chrome. Like I said earlier, we're not going to be doing any of IDE work today. We're not going to be editing our code directly. Instead, I'm going to show you how to use the Chrome Developer Tools, which, along with being incredibly useful for watching your code run, also have instant feedback, which is—we'll call it fun. I don't mind calling it fun.

This is the page, how it's going to look when we're done with it. The title of the top, we've got a lovely little portfolio running of all of the pictures from today's lesson. It's not what it looks like right now.

Right now, this is what it looks like. Page title section's there, but it looks a little different. The images are all there, but they're all stacked up on each other. And that copyright footer is—well, we're not going to do anything with that, so that guy's still there.

Now we're going to use the Chrome Developer Tools to really dig into this page, and get some stuff done. At any time you've got this code or a structure of some kind, and it's running in a black box, it's tough to get a look at it inside. It's tough to see what's happening on the inside.


You can put things in there. You can put in debugger lines. You can pop up alerts and stuff. There's ways to get information out of that black box. It's annoying and it's not fun, and there are cases where you just can't give you the information you need.

So anytime that you have access to a debugger where you can watch your code run, where you can pause it and check values in the middle of running, you are going to automatically increase your effectiveness as a developer by a factor of some very large number. So right now, I'm going to introduce you to the Chrome Debugging Tools, and have you do the most important thing that you're ever going to do with a right-click.

Please go up to the top on or next to the page title, right-click, and select Inspector Element. This is the DOM inspector here, which looks over your HTML. Here's the element we're looking for.

We are going to need to select it by ID, which means we need to find the ID. And there it is. I put it helpfully right in the element, but let's not cheat.

Now along with this guy showing up, you'll see that the entire rest of your page is in here, as well. And this looks like the HTML that got fed into your page, but it's not just text. It's live. This stuff is real.
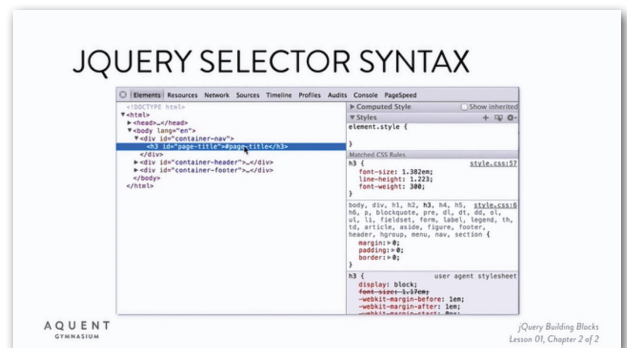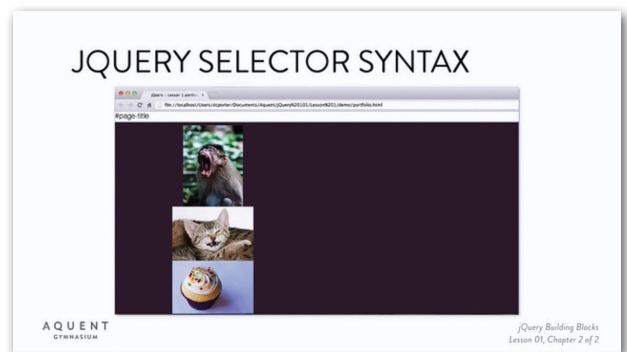
For example, you can add or remove class names. You can change the IDs of things, stuff like that. So this is great. We're going to spend a lot more time in this later. But I wanted to just give you a quick glimpse of it right now.

And up here, resources, network sources, timeline. We've got some other advanced stuff, stuff that's really useful once you've got a big, weighty chunk of code running, and you need to optimize it. But the thing that we're going to look at right now is the Console, so if you click on that for me.

Now this is effectively a command-line interface with your application. You can write commands here and hit Enter, and they'll be executed just as though they were going on in your actual code. For example, that'll pop up a little alert greeting.

Now, that's not super interesting, but you can do more complicated commands, and you actually have access to the full inner life of your application, all of the global variables, all of these things. And since this is a jQuery lesson, I included jQuery in this page. Let's take a look at it.
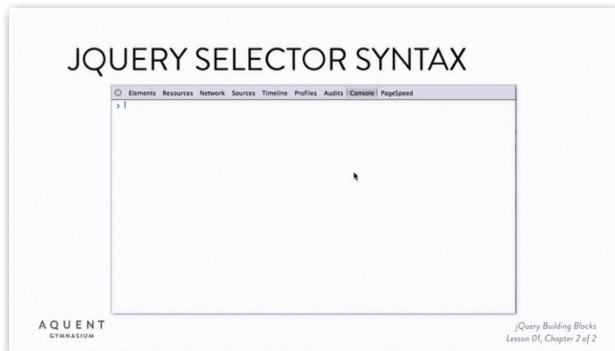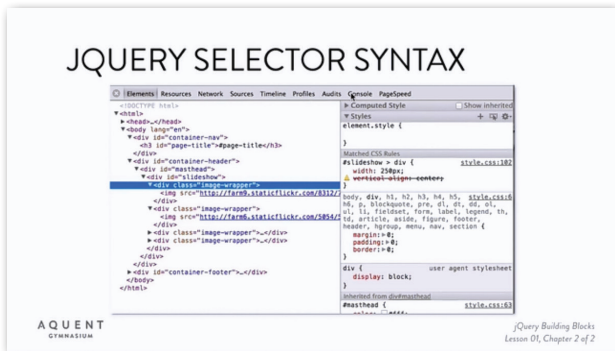

I'd like you to type "$," and then hit Enter. You may have to hit Enter twice. Now this dollar sign, this little simple symbol, is jQuery. It's your window into this whole world of introspection and plug-ins, and all of these things we've been talking about. Everything starts with the dollar sign.

Now as you can see here, the dollar sign is a function. But just to tease a little bit of some of the stuff we're covering later, I want you to type "$," and then dot. And then using the dot notation, you can see here that, thanks to the magic of JavaScript, jQuery is also an object. And it's got a bunch of stuff on there. Anyway, we'll go over that later.

As a function, the dollar sign's primary role is to select elements. And that's what we're showing you right now. And by the way, that's what you're going to be quizzed on later on.

Let's start with an ID. First thing we want to do is we want to select this element with a page title. And so drop in a hashtag, and page dash title, and hit Enter.
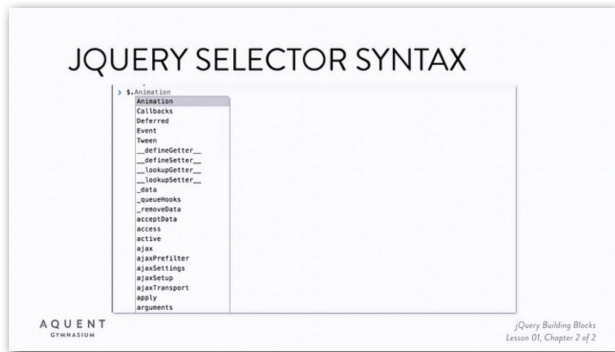
Now this has selected that element for us. There it is. Now this hashtag thing—pound sign for anyone that's not in the Twitter universe—that probably looks familiar to you, right? If you've used CSS, that should look very familiar to you.

This is just a plain old CSS selector. Use the pound sign or the hashtag for ID. If you want to select it by tag name, just do the name of the tag.

And just to prove that this is really CSS selectors at work, you can even chain them together like this. So H3 ID page title. This is going to select all level three headings with an ID of page title. There it is.

Now, let's do something with this guy. You'll notice that it returns the element, but it returns it inside an array. And that's not just an array. That's a special augmented j array, which has a whole bunch of other things to it.

For example, text. Text is going to pull out the stuff that's inside. And again, this isn't something that we're quizzing you on today. We'll just tease you a little bit or some of the stuff that we're going to cover next time.

Now let's change that text. Let's change it to "Lesson one Portfolio," and hit Enter. And there you go. It's been changed.
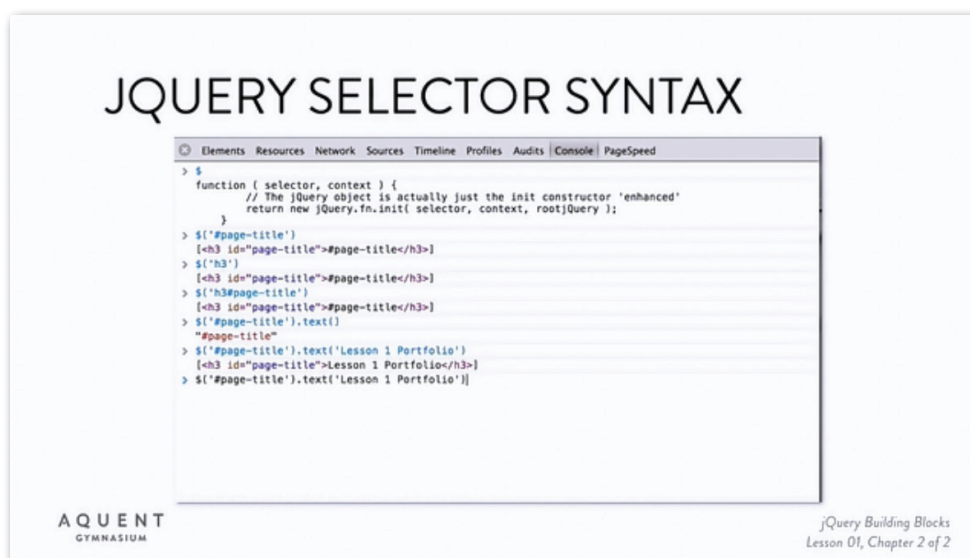
By the way, if you want to get to the previous, you'll notice I've been typing very fast. What's actually happening is that if you hit the up button on your keyboard while in the console, you'll go to the previous command. And you can edit that. Very useful for when you're iterating quickly over similar commands.

So we've now updated the title section of our page. Let's do something else. Let's say we want to select all the elements that have the class image wrapper. There they are.

I wrapped each one of these images in a DIV, gave it some CSS positioning, just to so it'll be a little prettier and line up. We're not actually going to do anything with those. Instead, we are going to grab the slide show element—again, by ID.
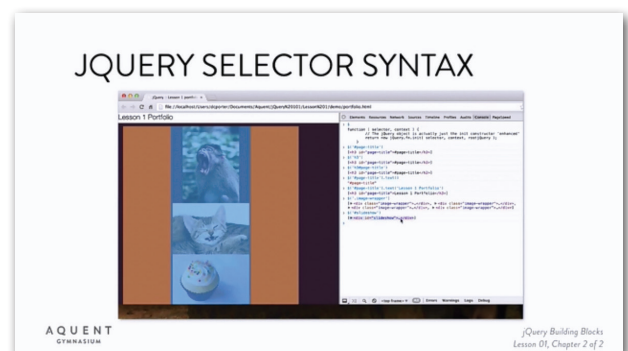
There it is. It's got the whole thing. It's got all these images stacked up inside it.

The way that we access the different CSS styles of this element is with this CSS command. So let's say, dot CSS background color. And if you just say that, it's going to tell you what it currently is. There's the default value of nothing.

But let's change it white using a second argument there. Boom, now it's white. Now just to make it look a little prettier, let's give it white border, as well. 2px solid white. And now, it's got a little bit of order in there.

One last thing, and this is the magic of jQuery plug-ins. If we wanted to turn this thing into an actual slide show ourselves, it would take a lot of effort. But since I included the cycle plug-in on this page, this is all you have to do to turn this stack of images into a slide show. And with that, we've turned some very simple HTML into a nice, sophisticated slide show widget.

That took three commands on the slide show element, so let me show you a fun thing you can do with jQuery. Anytime you've got a jQuery selector or action method—for example, this returns some elements. And if you want to set the background color to white, this guy also returns the impacted elements.

What that means is that you can do this cool thing called chaining. It's standard in jQuery. It's something that you're going to get used to quickly, but the first time you see it, it looks a little weird. It took me some time to get used to this when I first started out.

So let's chain all of these commands together. We're going to select the element. We're going to change the border.

We're going to change the background color all in one command. Ready? There it is.

Now chaining is great for doing a few things. It'll often allow you to do stuff without declaring extra variables, things like that. It's really nice.

Chaining, however, also opens up the possibility for you to write a single line of code which does 16 million different things. That's a bad idea. So don't do it.

There's a balance to be had there, like anything else. And we'll be touching a little bit later on in the course about keeping your code neat and clean. That would definitely be an example of unneat, unclean code.

But just for now, I want you to have a look at chaining, because it's probably the best example of very low-level ways that jQuery changes the way that you write JavaScript. Things with jQuery code that have jQuery in it just look and feel a little bit different. And once you're used to it, that's for the best.
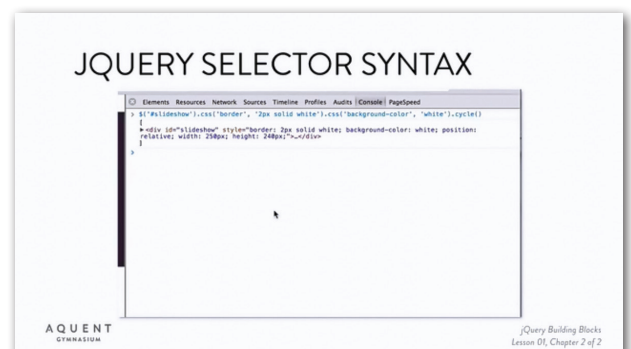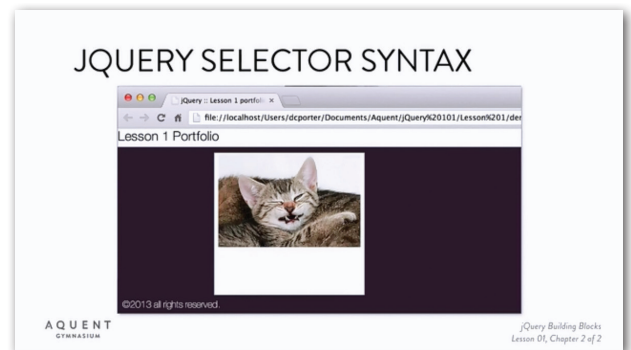
So, to review, the dollar sign is jQuery. It's your window into jQuery's view on your website. It's both a function and an object, which is a fact that has implications that we'll explore more later.

As a function, if you pass a string, it's going to return a jQuery list of matching elements. And that string is just a CSS selector using the CSS notation that you're familiar with. For example, if you want to select all of the elements of a certain tag, just pass in the name of the tag.

If you want to select by ID, you use the hashtag. And if you want to select by classes, use the dot notation. And just like CSS, you can mix and match, and use hierarchical selectors, and get very specific with the elements that you're selecting.

Here, for example, we're selecting an image element that is the child of an element with these two particular classes. And finally—merely an element of note at this time—don't forget you can change a query result list. It returns a list of objects, which can then have further jQuery things done to them.

Well, that's it for today. I have a couple of homework assignments for you. As you're going through them, please make liberal use of the pause button.

First up, I've got a short selector quiz in your class materials. And secondly, for next time, please review the selectors. There's a great documentation page at oscarotera.com/jquery. On that page, please review the basic and hierarchy selectors.

If you've got any questions or run into any issues, just jump on the forum. We'll see you there. And we'll see you next time in Lesson 2, That Certain Sparkle. This is jQuery Building Blocks—Five Techniques to Cut Your Web Development Time in Half. This is an Aquent production.