INTRO TO JQUERY

CLASS 1

SLIDES AVAILABLE AT:

HTTP://ALEXISGO.GITHUB.IO/JQUERY-INTRO/

WELCOME!

Girl Develop It is here to provide affordable and accessible programs to learn software through mentorship and hands-on instruction.

Some "rules"

- We are here for you!
- Every question is important
- Help each other
- Have fun

WHAT IS JQUERY?

jQuery is a library of JavaScript functions.

It contains many functions to help simplify your programming, including:

- HTML element selection & manipulation
- CSS manipulation
- HTML events
- JavaScript effects and animations

WHAT IS A LIBRARY?

- Software libraries hold functions
  - functions are like a collection of helpful code you want to run again & again
- When you include a library, you can use all the functions in that library
- That means: you get to take advantage of other people's experience!
- And... Save time!

WHY USE JQUERY?

- The most popular JavaScript library
- jQuery empowers you to "write less, do more."
- Great **documentation and tutorials**
- Used by nearly 20 million(!) websites

HISTORY OF JAVASCRIPT

- Developed by Brendan Eich of Netscape in 1995
- Standardized in 1997
- Java -- Actually JavaScript has nothing to do with the language Java. Java was just the cool kid in town at the time
- Script -- Instructions that a computer can run one line at a time

HISTORY OF JAVASCRIPT

- "AJAX" -- a way to communicate to servers was created in 2005
- jQuery -- a super-popular JavaScript Library 2006
- Node.js -- a way for JavaScript to perform back end functions in 2010
- 2012 -- spec for JavaScript "nearly" finished

WHAT DOES JAVASCRIPT DO?

- **Image Galleries and Lightboxes**
- **Games and all sorts of Google Doodles**
- **Interactions, like show/hide and accordians**
- **Retrieving data from other websites (through APIs)**
- All sorts of awesomeness, including this slideshow!

JQUERY: A BRIEF HISTORY

- jQuery was created by John Resig, a JavaScript tool developer at Mozilla.
- January 2006: John announced jQuery at BarCampNYC: BarCampNYC Wrap-up
- September 2007: A new user interface library is added to jQuery: jQuery UI: Interactions and Widgets
- September 2008: Microsoft and Nokia announce their support for jQuery
- December 2009: jQuery wins .Net Magazine's Award for Best Open Source Application

INCLUDING JQUERY

Two ways to include jQuery on your page:
Download the library, store it locally:

```
<head>
  <script type="text/javascript" src="jquery.js"></script>
</head>
```

Include the the live library:

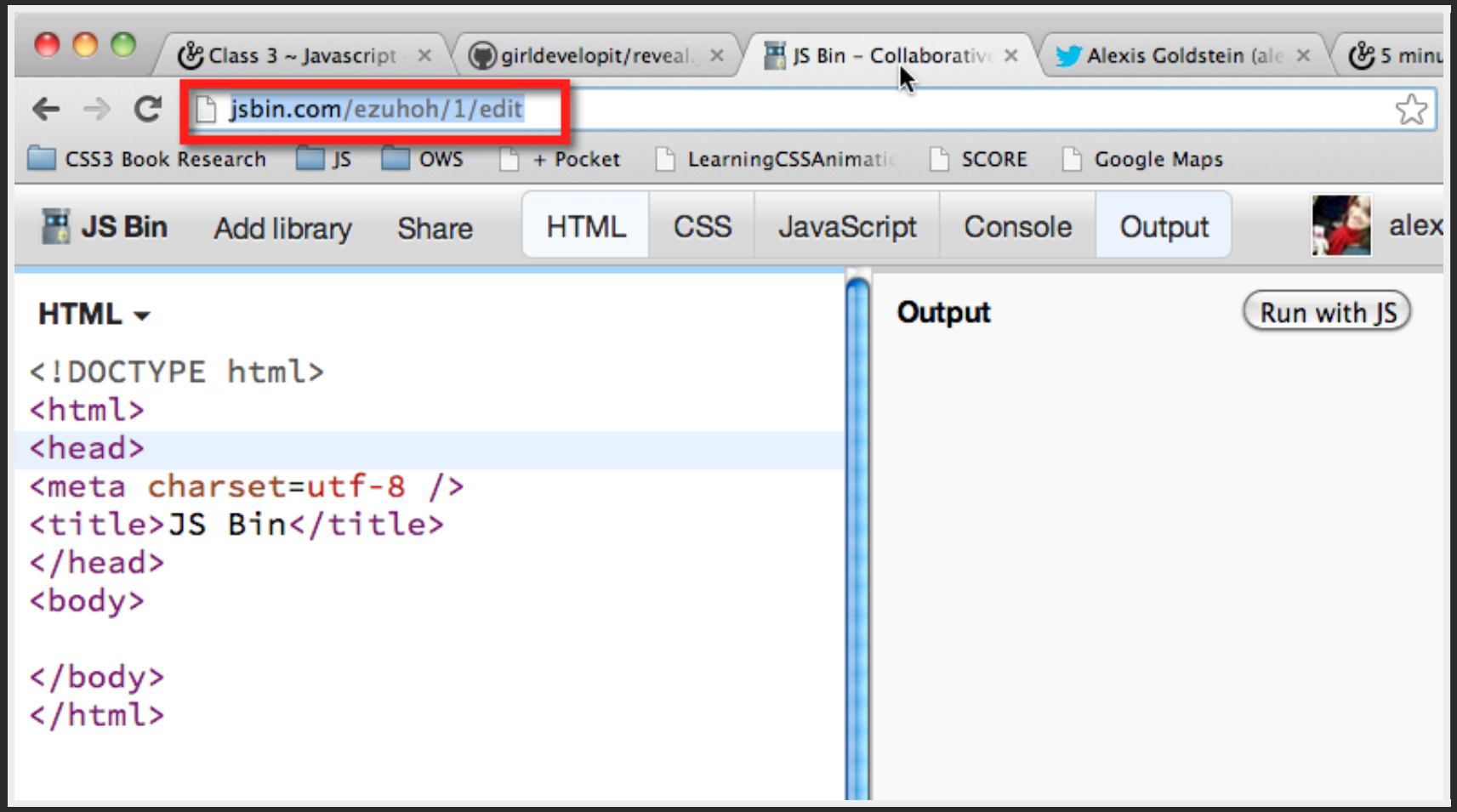
```
<head>
  <script type="text/javascript" src="http://code.jquery.com/jquery-1.8.3.min.js">
  </script>
</head>
```

Note: live code can change! It's always best to download

JSBIN

Today, we'll be using the online editor **JSBin**.

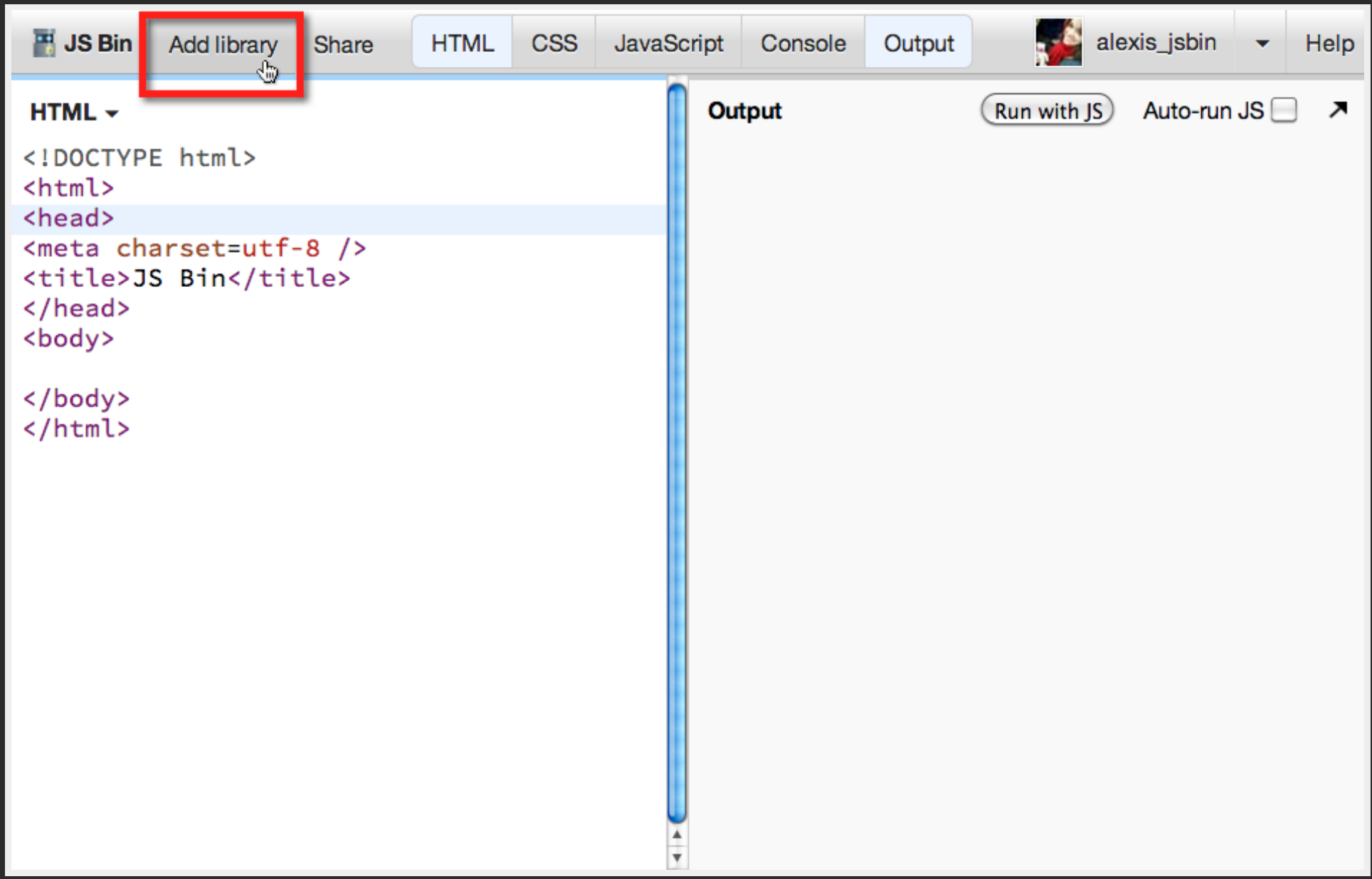
- **JSBin** allows us to create simple web pages from anywhere.
- Every time you create a new JSBin, it creates a new URL for that "file".
  - You can return to your work, or share it with others, by sharing this unique URL.
- You can also lock a revision of a JSBin by clicking Share and then Lock Revision


Class 3 ~ Javascript
girldevelopit/reveal
JS Bin – Collaborative
Alexis Goldstein (ale
5 minu
jsbin.com/ezuhoh/1/edit
CSS3 Book Research
JS
OWS
+ Pocket
LearningCSSAnimatic
SCORE
Google Maps
JS Bin
Add library
Share
HTML
CSS
JavaScript
Console
Output
alex
HTML
<!DOCTYPE html>
<html>
<head>
<meta charset=utf-8 />
<title>JS Bin</title>
</head>
<body>
</body>
</html>
Output
Run with JS

INCLUDING JQUERY

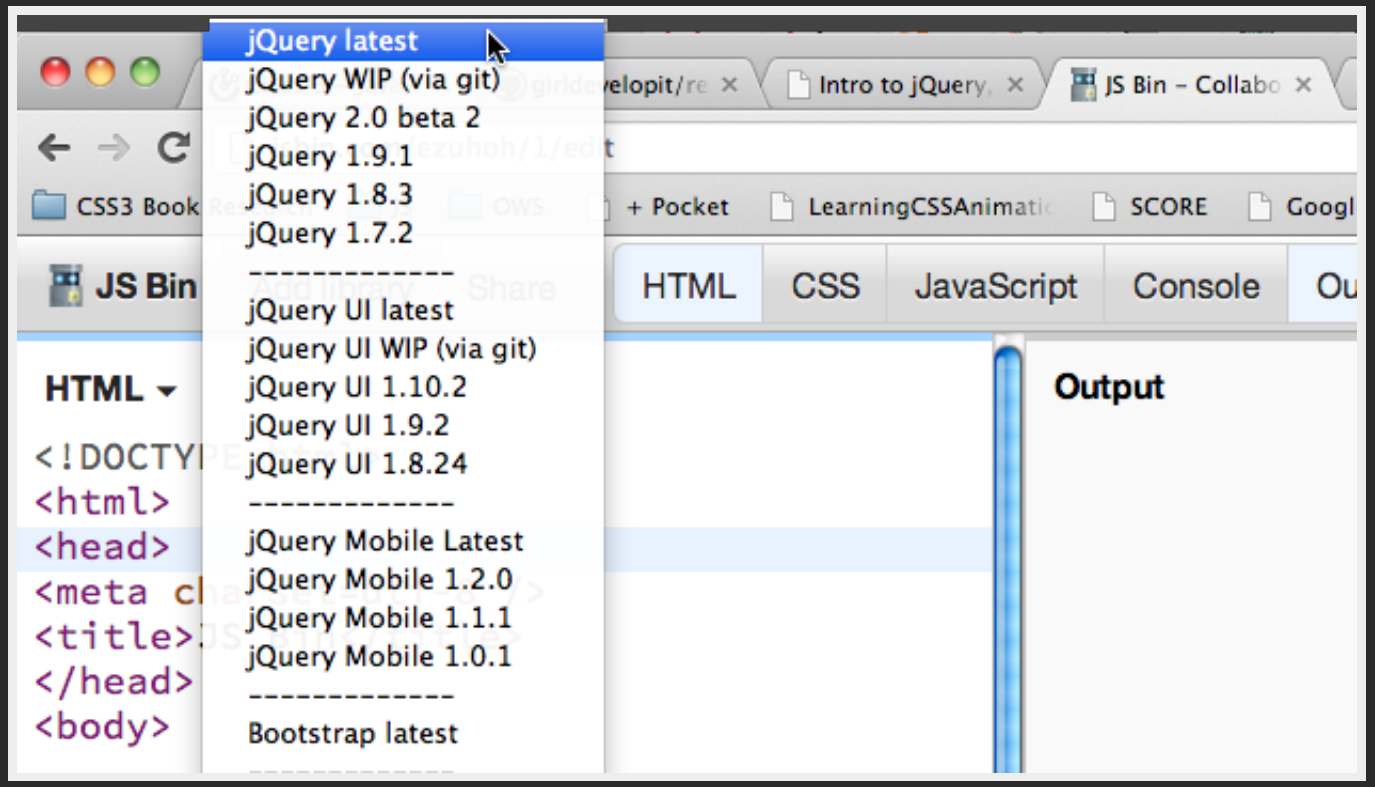
Let's begin by adding the latest version of jQuery to a new **JSBin**. JSBin has support for including many JavaScript libraries, including jQuery.

We begin by clicking the Add Library button.


JS Bin
Add library
Share
HTML
CSS
JavaScript
Console
Output
alexis_jsbin
Help
HTML
<!DOCTYPE html>
<html>
<head>
<meta charset=utf-8 />
<title>JS Bin</title>
</head>
<body>
</body>
</html>
Output
Run with JS
Auto-run JS

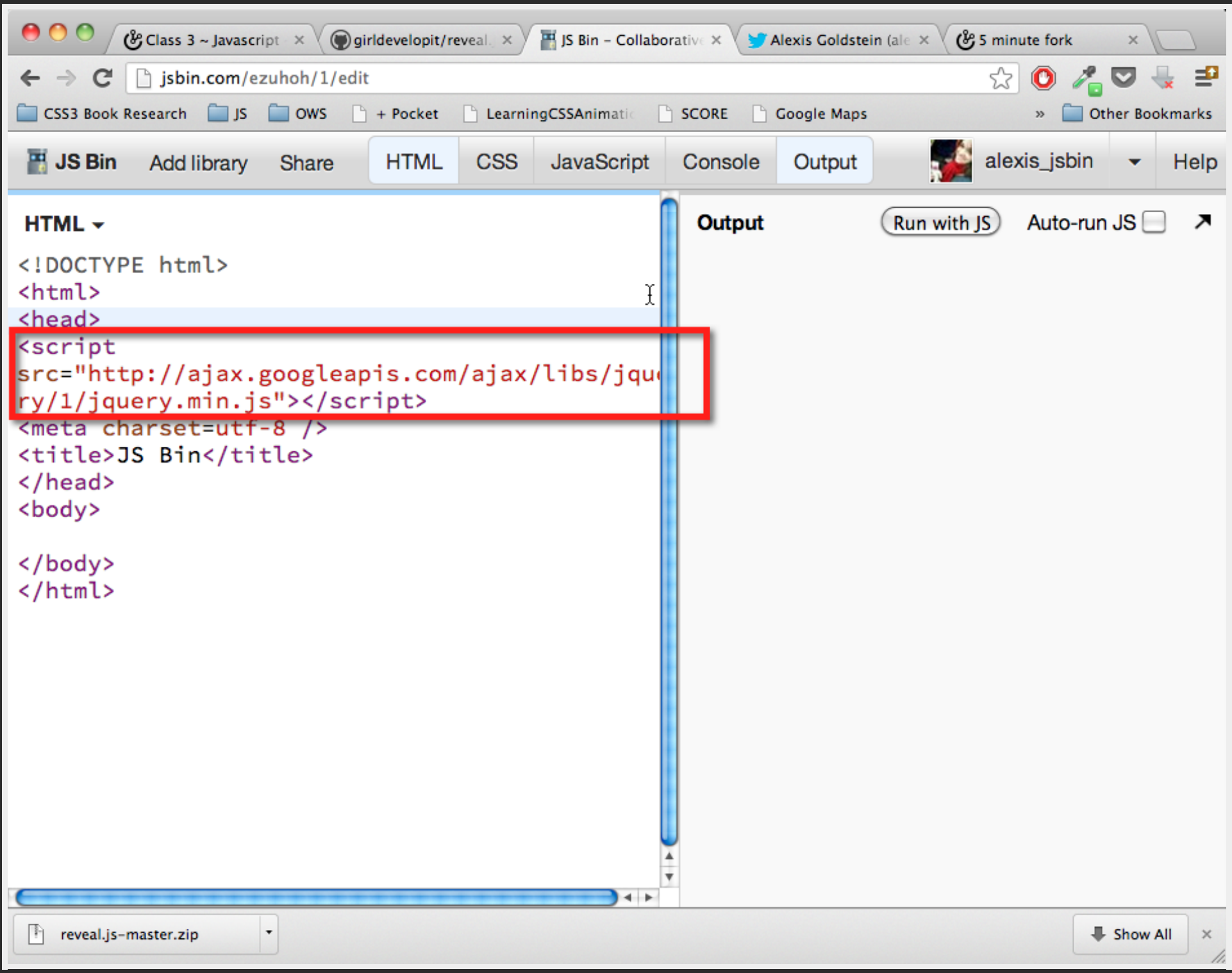
INCLUDING JQUERY

We'll use the latest version of the jQuery library.



INCLUDING JQUERY

Clicking Add Library > jQuery latest will result in the jQuery library being linked to in a `script` element in the `head` section of our JSBin's HTML.


Class 3 ~ Javascript
girldevelopit/reveal
JS Bin – Collaborative
Alexis Goldstein (ale
5 minute fork
jsbin.com/ezuhoh/1/edit
CSS3 Book Research
JS
OWS
+ Pocket
LearningCSSAnimatic
SCORE
Google Maps
Other Bookmarks
JS Bin
Add library
Share
HTML
CSS
JavaScript
Console
Output
alexis_jsbin
Help
HTML
<!DOCTYPE html>
<html>
<head>
<script
src="http://ajax.googleapis.com/ajax/libs/jqu
ry/1/jquery.min.js"></script>
<meta charset=utf-8 />
<title>JS Bin</title>
</head>
<body>
</body>
</html>
Output
Run with JS
Auto-run JS
reveal.js-master.zip
Show All

JQUERY SELECTORS

One of the most important things we can do with jQuery is select HTML elements from existing websites. In jQuery, we do this through selectors

jQuery selectors are much like CSS selectors. Instead of finding to an html element to apply styles, you can find an html element to modify, remove or replace.

JQUERY ELEMENT SELECTORS

jQuery selectors let you get elements by:

- Element name (div, p)

```
var divs = jQuery("div");
```

JQUERY ELEMENT SELECTORS

Let's practice using jQuery's element selector.

- Open the following JSBin: **http://jsbin.com/ezuhoh/7/edit**

- Let's select all the paragraphs by entering the following code in the JavaScript window.

```
var paragraphs = jQuery("p");
```

- After storing all the paragraphs in the variable paragraphs, we can log all the text of these paragraphs to the console:

```
var paragraphs = jQuery("p");
console.log(paragraphs.text());
```

IMPORTANT JAVASCRIPT VOCABULARY

Before we go any further, it's important to review a few basic programming terms that are relevant to jQuery.

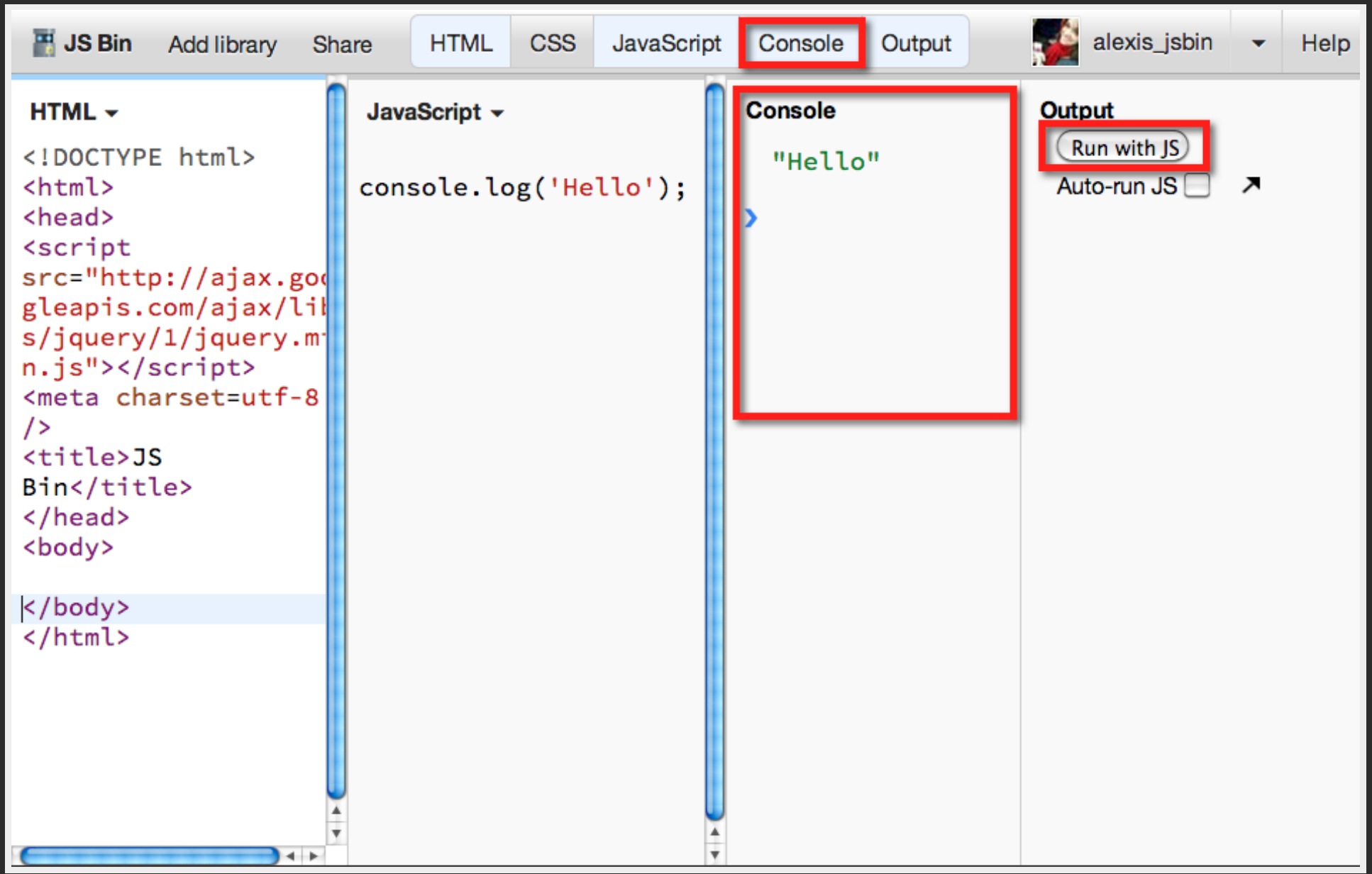
- Statements
- Variables
- Functions

STATEMENTS

- Each line in JavaScript is an instruction, aka a statement
- When the browser reads it, it "executes the script"

- Try typing the following into JSBin, and then clicking "Run with JS".
  - Make sure you've clicked the Console button so you can see the results logged to the console.

```
console.log('Hello');
```


JS Bin
Add library
Share
HTML
CSS
JavaScript
Console
Output
alexis_jsbin
Help
HTML
<!DOCTYPE html>
<html>
<head>
<script
src="http://ajax.goo
gleapis.com/ajax/lib
s/jquery/1/jquery.mi
n.js"></script>
<meta charset=utf-8
/>
<title>JS
Bin</title>
</head>
<body>
</body>
</html>
JavaScript
console.log('Hello');
Console
"Hello"
Output
Run with JS
Auto-run JS

VARIABLES

Variables hold content

Words, numbers, true/false, basically any kind of content

Declare a variable (Give it a name)

```
var bananas;
```

Initialize variable (Give it a value)

```
bananas = 5;
```

VARIABLES

Declare and initialize at the same time!

```
var bananas = 5;
```

Change value of variable

```
bananas = 4;
```

(I ate a banana)

NAMING RULES

Begin with a letter, _, or $

Contain letters, numbers, _ and $

```
var hello;
var _hello;
var $hello;
var hello2;
```

Names are case sensitive

```
var hello;
var Hello;
var heLLO;
```

BACK TO JQUERY SELECTORS!

JQUERY SELECTORS

jQuery selectors allow us to select HTML elements from existing websites.

In addition to selecting all elements of a given type, jQuery selectors *also* let you get elements by:

- ID name (#mainpic, #results)

```
var img = jQuery("#mainpic"); //img with id mainpic
```

- Class name (.result, .picture)

```
//All images with class picture
 var images = jQuery(".picture");
```

USING JQUERY'S CLASS SELECTOR

- Re-open the following JSBin: **http://jsbin.com/ezuhoh/7/edit**
- Let's select all the images with the class picture by entering the following code in the JavaScript window.

```
var images = jQuery(".picture");
```

- Next, we'll use jQuery's addClass() to add the class pic-border to all the matched elements.
  - The CSS for this JSBin has previously defined the class pic-border to apply a 5px black border

```
var images = jQuery(".picture");
images.addClass('pic-border');
```

- Don't forget to click the Run with JS button to see the changes take effect.

USING JQUERY'S ID SELECTOR

- Re-open the following JSBin: **http://jsbin.com/ezuhoh/7/edit**
- Let's select the images with the id mainpic by entering the following code in the JavaScript window.

```
var mainpic = jQuery("#mainpic");
```

- Next, we'll use jQuery's addClass() to add the class float-right to the matched element.
  - The CSS for this JSBin has previously defined the id selector float-left to make any matched element float right.

```
var mainpic = jQuery("#mainpic");
mainpic.addClass('float-right');
```

- Don't forget to click the Run with JS button to see the changes take effect.

THE JQUERY FUNCTION

We've seen jQuery appearing a lot in the code we've written thus far.

The jQuery library actually stores all of its power in what is called a function.

Every time we have called jQuery in our code thus far, we've been calling the jQuery function.

```
var mainpic = jQuery("#mainpic");
var images = jQuery(".picture");
```

JQUERY == $

There is a shorter way to call the jQuery function. We can simply use the shorthand name for the jQuery function, which is simply the $ character.

```
var mainpic = $("#mainpic");
var images = $(".picture");
```

FUNCTIONS

Functions are re-usable collections of statements.

In other words, they're a way to package up and easily reuse chunks of code!

Declare a function

```
function sayHi(){
  console.log('Hi!');
}
```

Call the function

```
sayHi();
```

ARGUMENTS

Functions can take named arguments

```
function sayHi(name){
  console.log('Hi' + name + '!');
}
sayHi('Mitch, the dinosaur');
sayHi('Harold, the hippo');

var name = 'Pip, the mouse';
sayHi(name);
```

ARGUMENTS

Functions can take MULTIPLE named arguments

```
function addNumbers(num1, num2){
  var result = num1 + num2;
  console.log(result);
}

addNumbers(5, 6);

var number1 = 12;
var number2 = 15;
addNumbers(number1, number2);
```

RETURN VALUES

Functions can return a value

```
function addNumbers(num1, num2){
  var result = num1 + num2;
  return result; //Anything after this line won't be read
}
var sum  = addNumbers(5, 6);
```

THE JQUERY ($) FUNCTION

All of jQuery's power is all stored inside a single function, jQuery.

Again, we can call the jQuery function, or its shorthand, $. These two statements are identical:

```
jQuery('p');
$('p');
```

JQUERY ACTIONS

Now that we have talked about selectors, and the jQuery shorthand ($), we can tell you the foundation of all jQuery: $(selector).action();

jQuery has hundreds of actions that can be performed on any element

All the actions are methods

As methods they are called with dot notation

Action format

```
$(selector).action();
```

UPDATING VALUES AND HTML

```
<div class = "results">Boo!</div>
```

Get and set html value

```
var div = $('#results');
div.html();
div.html('New html!');
```

APPEND AND PREPEND

```
<div class = "results">Boo!</div>
```

Append html

```
var div = $('#results');
div.append('Additional html');
```

Prepend html

```
var div = $('#results');
div.prepend('Additional html (on top)');
```

CREATING NEW ELEMENT

```
var newDiv = $('<div></div>');
```

Seriously. That's it!

DOCUMENT READY

Webpages take time to load

Almost always, you don't want the JavaScript to be called until the page is loaded

Document ready is a method called when the page is loaded

```
$(document).ready(function(){

});
```

Note: The function() inside is an "anonymous function". It has no name, but still performs like a function.

UPDATING ATTRIBUTES AND CSS

```
<img id="mainpicture" src="http://girldevelopit.com/assets/pink-logo.png">
```

Attribute get and set

```
var img = $('#mainpicture');
img.attr('src');
img.attr('src', 'http://girldevelopit.com/assets/pink-logo.png');
```

CSS property get and set

```
var img = $('#mainpicture');
img.css('width');
img.css('width', '200px');
```

UPDATING CSS: A BETTER WAY

Rather than directly updating the CSS as we saw on the previous slide, it's better to modify styles on your page by toggling existing CSS classes on and off. This way, we are not adding CSS via an inline style, which is messy.

We are also preserving the separation of Content (HTML) and formatting (CSS) when we take this approach.

UPDATING CSS: A BETTER WAY

If we've previously defined a CSS selector to set a size...

```
.shrink {
  width: 200px;
}
```

We can leverage that class, and trigger the change by simply adding the class via jQuery:

```
var img = $('#mainpic');
img.addClass('shrink');
```

FADING ELEMENTS OUT

We can fade out an element by calling the action fadeOut on it.

```
var img = $('#mainpic');
img.fadeOut();
```

FADING ELEMENTS OUT ON CLICK

We will talk in detail about responding to user actions with jQuery next week.

However, as a preview, open up the following JSBin:
http://jsbin.com/ubeney/3/edit

We can fade out an element in response to a user clicking on it by adding the following jQuery:

```
$('#mainpic').click(
  function() {
    $(this).fadeOut();
  }
);
```

SHOWING AND HIDING ELEMENTS

We can hide an element by calling the action hide on it.

```
$('#mainpic').click(
  function() {
    $(this).hide();
  }
);
```

Unlike fadeOut(), which happens over time, hide() happens instantly.

SHOWING AND HIDING ELEMENTS

Open the following JSBin:
http://jsbin.com/owarum/1/edit

Add in the following jQuery to make the buttons "Show" and "Hide" respond appropriately.

```
$('#show').click(
  function() {
    $('#mainpic').show();
  }
);

$('#hide').click(
  function() {
    $('#mainpic').hide();
  }
);
```

HOMEWORK!

Reading:

- **JavaScript 101: Getting Started**
- **How jQuery Works**
- **Selecting Elements**
- **jQuery Basics** from jQuery Fundamentals, by Rebecca Murphey

If you'd like a book recommendation to go along with this course, I'd recommend **jQuery: Novice to Ninja**

QUESTIONS?