

GRAND
CIRCUS
DETROITGRAND
CIRCUS
DETROITGRAND
CIRCUS
DETROITGRAND
CIRCUS
DETROITGRAND
CIRCUS
DETROITGRAND
CIRCUS
DETROITGRAND
CIRCUS
DETROIT

JQUERY

GRAND
CIRCUS
DETROITGRAND
CIRCUS
DETROITGRAND
CIRCUS
DETROITGRAND
CIRCUS

GRAND
CIRCUS
DETROIT

GRAND
CIRCUS
DETROIT

GRAND
CIRCUS
DETROIT

DETROIT

GRAND
CIRCUS
DETROIT

GRAND
CIRCUS
DETROIT

GRAND
CIRCUS
DETROIT

WELCOME BACK!

DETROIT

GRAND
CIRCUS
DETROIT

GRAND
CIRCUS

GRAND
CIRCUS
DETROIT

GRAND
CIRCUS

GRAND
CIRCUS
DETROIT

GOALS FOR THIS UNIT

1. Review
2. Intro to jQuery
3. Selectors
4. Actions

INTRO TO JQUERY



WHAT IS IT?

jQuery is an extremely popular JavaScript library
(read: huge collection of functions) that makes writing
JavaScript easier and faster.

HIGHLIGHTS

jQuery is:

- small
- fast
- feature-rich
- cross-browser
- extensively documented
- extremely popular



CIRCUUS
DETROIT

SITE PERFORMANCE

We care that jQuery is kept lean and mean because it's going to make a smaller (negative) impact on our sites' performance.

DON'T REINVENT THE WHEEL

jQuery makes the common tasks many web developers have to accomplish available out of the box.

- fade elements in and out
- add/remove elements
- handle events
- change attribute/property values
- animation

CROSS-BROWSER

jQuery uses automatic feature detection to shield the developer from the many inconsistencies in the way different browsers approach tasks.

CIRCUS
DETROIT

GRAND
CIRCUS
DETROIT

CIRCUS
DETROIT

GRAND
CIRCUS
DETROIT

SOLID DOCS

A huge plus of using jQuery is that it has very thorough, clear documentation.

GRAND
CIRCUS
DETROIT

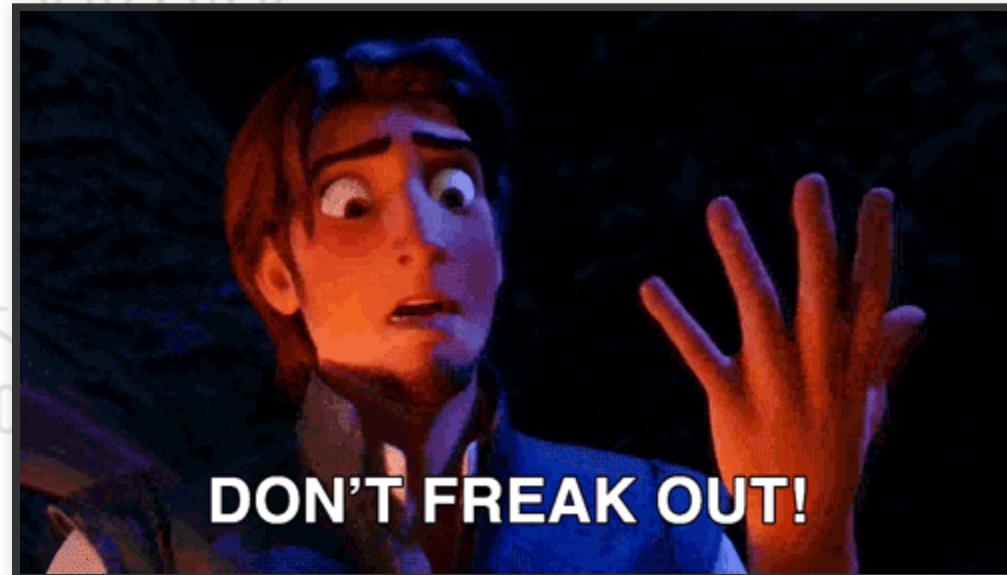
GRAND
CIRCUS
DETROIT

GRAND
CIRCUS
DETROIT

COMMUNITY

The enduring popularity of jQuery means that it has an enormous following and tons of people and online resources to learn from. If you run into a problem or question, chances are that someone has already found a solution and written it down on the internet.

GRAND
CIRCUS
DETROIT



GRAND
CIRCUS
DETROIT

GRAND
CIRCUS

jQuery is still just JavaScript!

GRAND
CIRCUS
DETROIT

GRAND
CIRCUS
DETROIT

GRAND
CIRCUS

GRAND
CIRCUS
DETROIT

GRAND
CIRCUS

GRAND
CIRCUS
DETROIT

CIRCUS
DETROIT

GRAND
CIRCUS
DETROIT

GRAND
CIRCUS
DETROIT

GRAND
CIRCUS
DETROIT

GRAND
CIRCUS
DETROIT

INCLUDING JQUERY

GRAND
CIRCUS
DETROIT

GRAND

GR

There are a couple of ways that we can include
jQuery in our projects:

- Download the library and store it locally in your project folder
- Link to a live version of the library via CDN

PROS AND CONS

download

CDN

stable

can change

bloats your
site

makes a call every time you
refresh

LIRLUS
DETROIT

GRAND
CIRCUS
DETROIT

LIRLUS
DETROIT

GRAND
CIRCUS
DETROIT

GRAND
CIRCUS
DETROIT

GRAND
CIRCUS
DETROIT

DOWNLOADING JQUERY

CIRCUS
DETROIT

GRAND
CIRCUS
DETROIT

CIRCUS
DETROIT

GRAND
CIRCUS
DETROIT

GRAND
CIRCUS
DETROIT

1. Go to jquery.com/download
2. Click the link Download the uncompressed, development jQuery 3.2.1. This takes you to a raw text file.
3. Select File > Save Page as... > save As 'jquery-3.2.1.js' somewhere useful



Or just link to a hosted CDN in a script tag on your page. Both script tags link to a working jQuery file.

```
<script src="js/jquery-3.2.1.js"></script>
<script src="https://code.jquery.com/jquery-3.2.1.js"></script>
```

LIRLUS
DETROIT

GRAND
CIRCUS
DETROIT

JQUERY SELECTORS

GRAND
CIRCUS
DETROIT

GRAND

GR
CIR

CIRCUS
DETROIT

GRAND
CIRCUS
DETROIT

Anything look familiar?

```
var divs = $("div");
// All divs on the page

var happyThing = $("#happy");
// Element with id "happy"

var rounds = $(".roundedCorner");
// Elements with class "roundedCorner"
```

GRAND
CIRCUS
DETROIT

GRAND
CIRCUS
DETROIT

GRAND
CIRCUS
DETROIT

GRAND
CIRCUS
DETROIT

GRAND

GR



That's right! jQuery allows us to use the same selectors we've already used in our CSS. Selecting elements this way is faster, more accurate, and requires less code than other methods.

JQUERY FILTERS

jQuery even adds CSS-like **filters** to make selecting elements even easier.

Example

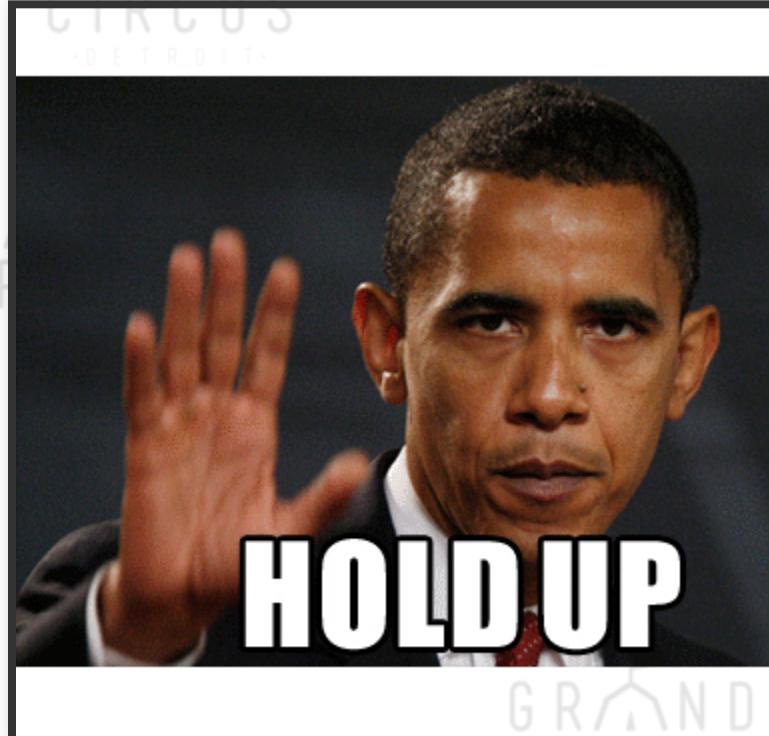
This code would select all `li` elements, then *filter* that list using the psuedo-selector `:even` which would return every other

```
$( "li" ).filter( ":even" );
// This is a jQuery equivalent to --
$( "li" ).filter( ":nth-child(even)" );
```

CIRCUS
DETROIT

CIRCUS
DETROIT

GRAND
CIRCUS
DETROIT



What was that dollar sign thingy?!

GRAND
CIRCUS
DETROIT

GRAND
CIRCUS
DETROIT

GRAND

GRAND
CIRCUS
DETROIT

GRAND

GRAND
CIRCUS
DETROIT



\$()

\$() is a tiny piece of jQuery magic (read: function) that turns whatever is inside the parentheses into a jQuery object.



GRAND
FITCH

READY METHOD

AD
GRAND
CIRCUS
DETROIT

GRAND
CIRCUS
DETROIT

GET READY!

`.ready()` is a jQuery method which tells a script to wait to execute until an HTML document has loaded.

```
$(document).ready(function() {  
    $("#header").slideDown("slow");  
});
```

CIRCUS
DETROIT



WHY WE NEED IT

Just like plain ol' JavaScript, jQuery needs the browser to construct the DOM before it can select anything.





THE SOLUTION

We can place our script tags at the very end of the body of our HTML. That way, all the HTML has already been loaded before the scripts are run.

THE (PORTABLE) SOLUTION

Wrapping our code in the `.ready()` method means the browser will still wait for the DOM to load before running any scripts, regardless of where they're placed.

```
$(document).ready(function() {  
    // JavaScript-y things here  
});
```

THE (PORTABLE, SHORTHAND) SOLUTION

We can simplify this method by using a shorthand one! Code inside this very common shorthand method will be run automatically once the page loads.

```
$(function() {  
    // JavaScript-y things here  
});
```

JQUERY ACTIONS



jQuery has tons of actions that can be performed on any element. All of these are functions (or methods).

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



jQuery makes it simple to accomplish a huge number of common tasks, such as updating elements' attributes or **CSS** (and pretty much anything else).

```
var img = $("#myPicture");
img.attr("src");
img.attr("src", "http://myPictureLivesHere.com");

img.css("width");
img.css("width", "200px");
```



CHAINING

We can use dot notation to run **several methods** on the same selector.

```
$(".hideBox").hide().delay(1200).fadeIn(800);
```



EXERCISE

This is all done using jQuery in your script file.

1. Create a webpage that has the following elements:
p, ul, li's inside the ul, and a button.
2. The ul should be hidden and then fade in once it loads.
3. There should be code to make the p tag's background color purple.
4. All even li's should have a height of 40px.

GRAND
CIRCUS
DETROIT

GRAND
CIRCUS
DETROIT

GRAND
CIRCUS
DETROIT

JQUERY LAB PART 1



GRAND
CIRCUS
DETROIT

GRAND
CIRCUS

GRAND
CIRCUS



NEW
CONTENT
AHEAD!





-DETROIT-



-DETROIT-



JQUERY METHODS & EVENTS

GOALS FOR THIS UNIT

1. Review
2. More on Methods
3. Effects
4. Events

RETRIEVING CONTENT

ONLINE

The `.html()` method retrieves the HTML inside the first element matched by the selector. This includes any descendants. It can also be used to update that content.

```
<div class="demo-container">
  <div class="demo-box">Demonstration Box</div>
</div>
```

```
$( "div.demo-container" ).html();
```

The `.text()` method retrieves the text inside the first element matched by the selector. This also includes any descendants. It can also be used to update that content.

```
$( "div.demo-box" ).text( "New Demo Box Text!" );
```

The above would change the text from "Demonstration Box" to "New Demo Box Text!".

GRAND CIRCUS DETROIT

Demo

Demo



UPDATING CONTENT



GRAND
CIRCUS
DETROIT



GRAND
CIRCUS
DETROIT



GRAND
CIRCUS
DETROIT

The `.html()` and `.text()` methods can not only retrieve content, but update it as well.

GRAND
CIRCUS
DETROIT

GRAND
CIRCUS

DEMO

DEMO 2



The `.replaceWith()` method replaces matched elements with new content and returns the replaced elements.





The `.remove()` method removes any elements in the matched set.

INSERTING CONTENT

GRAND
CIRCUS
DETROIT

GRAND
CIRCUS
DETROIT

DETROIT

GRAND
CIRCUS
DETROIT

The `.before()` method inserts content before the matched element(s).

```
$(".inner").before( "<p>Test</p>" );
```



DETROIT



The `.after()` method inserts content after the matched element(s).

```
$(".inner").after( "<p>Test</p>" );
```

GRAND
CIRCUS
DETROIT

CIRCUUS
DETROIT

GRAND
CIRCUS
DETROIT

GRAND
CIRCUS
DETROIT

GRAND
CIRCUS
DETROIT

GRAND
CIRCUS
DETROIT

The `.prepend()` method inserts content inside the matched element(s) immediately after the opening tag.

```
$(".inner").prepend( "<p>Test</p>" );
```

CIRCUS
DETROIT

CIRCUS
DETROIT

CIRCUS
DETROIT

GRAND
CIRCUS
DETROIT

GRAND
CIRCUS
DETROIT

The `.append()` method inserts content inside the matched element(s) immediately before the closing tag.

```
$(".inner").append( "<p>Test</p>" );
```



DEMO



CIRCUS
DETROIT

GRAND
CIRCUS
DETROIT

CIRCUS
DETROIT

GRAND
CIRCUS
DETROIT

JQUERY EFFECTS

EFFECTS

Effects generally enhance the interactive components of our sites. These days, we can accomplish many of the same tasks with pure CSS, but certain older browsers don't play nicely with our shiny CSS3 animations.



TOGGLING

Methods with `toggle` in their name will look at the current state of the selected element(s) and switch to the opposite.



GRAND
CIRCUS
DETROIT

CIRCUUS
DETROIT

GRAND
CIRCUS
DETROIT

BASIC EFFECTS



CIRCUS
DETROIT



CIRCUS
DETROIT

Method Description

.hide() Hide the matched elements.

.show() Display the matched elements.

.toggle() Display or hide the matched elements.

ČIRCUS
DETROIT

CIRKUS
DETROIT

ČIRCUS
DETROIT

GRAND
CIRCUS
DETROIT

CUSTOM EFFECTS

GRAND
CIRCUS
DETROIT

GRAND
CIRCUS
DETROIT

GRAND
CIRCUS
DETROIT

GRAND
CIRCUS
DETROIT

GRAND
CIRCUS

GRAND
CIRCUS
DETROIT

GRAND
CIRCUS
DETROIT

GRAND
CIRCUS

Method Description

.animate() Perform a custom animation of a set of CSS properties.

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

.finish() Stop the currently-running animation, remove all queued animations, and complete all animations for the matched elements.

.stop() Stop the currently-running animation on the matched elements.

CIRCUS
DETROIT

GRAND
CIRCUS
DETROIT

CIRCUS
DETROIT

GRAND
CIRCUS
DETROIT

FADING EFFECTS

Method

Description

.fadeIn()

Display the matched elements by fading them to opaque.

.fadeOut()

Hide the matched elements by fading them to transparent.

.fadeTo()

Adjust the opacity of the matched elements.

.fadeToggle()

Display or hide the matched elements by animating their opacity.

LIRCUIS
DETROIT

GRAND
CIRCUS
DETROIT

SLIDING EFFECTS

GRAND
CIRCUS
DETROIT

GRAND

GRAND
CIRCUS
DETROIT

GR
CIR

Method

Description

.slideUp()

Hide the matched elements with a sliding motion.

.slideDown()

Display the matched elements with a sliding motion.

.slideToggle()

Display or hide the matched elements with a sliding motion.

GRAND
CIRCUS
DETROIT

ČIRČUŠ
DETROIT

GRAND
CIRCUS
DETROIT

ČIR
DET

GRAND
CIRCUS

GRAND
CIRCUS
DETROIT

GRAND
CIRCUS
DETROIT

GRAND
CIRCUS
DETROIT

GR
CIR
DET

GRAND
CIRCUS
DETROIT

GRAND
CIRCUS
DETROIT

GRAND
CIRCUS
DETROIT

GR
CIR
DET

NEW
CONTENT
AHEAD!



GRAND
CIRCUS
DETROIT

GRAND

GRAND

GRAND
CIRCUS
DETROIT

CIRCUS
DETROIT

GRAND
CIRCUS
DETROIT

GRAND
CIRCUS
DETROIT

GRAND
CIRCUS
DETROIT

GRAND
CIRCUS
DETROIT

JQUERY EVENTS

GRAND
CIRCUS
DETROIT

GRAND

GRAND
CIRCUS
DETROIT

GRAND

EVENTS

jQuery's event methods are used to cause certain code to take effect when the user interacts with the browser. This code is called a handler because it tells JavaScript how to handle the event.

ONE METHOD TO RULE THEM ALL



As of jQuery 1.7, the `.on()` method is all we need to attach handlers to events. Previously, there were a few different ways to do it, including `bind()`, `live()` and `delegate()`. Those are now deprecated.



THE BREAKDOWN

1. First, we need to select the affected element(s).

```
$( "button" )
```

CIRCUS
DETROIT

GRAND
CIRCUS
DETROIT

GRAND
CIRCUS
DETROIT

GRAND
CIRCUS
DETROIT

THE BREAKDOWN

2. We'll use the `.on()` method to handle the event.

```
$( "button" ).on()
```

THE BREAKDOWN

3. The first argument we pass to the `.on()` method is the event we want to respond to.

```
$( "button" ).on( "click" )
```

THE BREAKDOWN

4. The second argument we pass to the `.on()` method is the code we want to be triggered by the event in the form of either a named or anonymous function.

```
$( "button" ).on( "click", function() {  
    console.log( "You clicked a button!" );  
});
```

SHORTCUT METHODS

Most of the events also have shortcut methods, which do the same thing.

```
$( "button" ).click(function() {  
  console.log("You clicked a button!");  
});
```

is the same as

```
$( "button" ).on("click", function() {  
  console.log("You clicked a button!");  
});
```

THE EVENT TARGET

Sometimes you need to know what HTML element was clicked or where the mouse moved. jQuery passes an *event* to your function. This event includes a lot of information about what happened. One bit of information is the `event.target` which indicates what was clicked.

```
$("button").on("click", function(event) {  
    console.log("This element was clicked: " + event.target);  
});
```

THIS

There is another way to identify what was clicked. Just use the `this` keyword.

```
$( "button" ).on( "click", function() {  
    console.log( "This element was clicked: " + this );  
});
```

EVENT.TARGET VS. THIS

Consider the code below. If I click the word "First", am I clicking the "first" div or the "wrapper" div?

```
<div id="wrapper">
  <div id="first">First</div>
  <div id="second">Second</div>
</div>
```

- `event.target` will be the *most specific* element.
- `this` will always be the element for which we *registered* the event. <**DEMO**>



GRAND
CIRCUS
DETROIT

HOMEWORK

From JavaScript & jQuery:

- Chapter 8: 367-408

GRAND
CIRCUS
DETROIT

JQUERY LAB PART 2





NEW
CONTENT
AHEAD!



GRAND





AJAX & JSON



GRAND CIRCUS

GRAND CIRCUS
DETROIT

GRAND CIRCUS

GOALS FOR THIS UNIT

1. Review
2. Requests & Responses
3. Submitting Forms
4. Error Handling



-DETROIT-



REVIEW

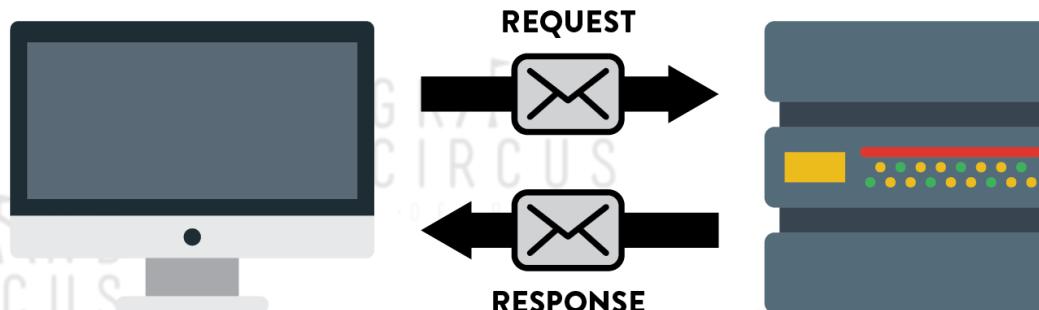


HOW THE INTERNET WORKS

Time for a bit of crucial theory.

CLIENT / SERVER

REQUEST-RESPONSE



The browser (client) sends a *request* to the server.
Then the server sends back a *response*.

THE REQUEST

The request kicks off the communication. It is a message sent from the client to the server. It includes several pieces of information:

1. The envelope: Where is the request going? What server?
2. The payload: What do we want the server to do?

THE RESPONSE

Unless the internet breaks, a response always comes back from the server. This message includes:

1. Status: Was the server able to do what the client wanted?
2. Data: Any information that was requested.

GRAND
CIRCUS
DETROIT

AJAX

GRAND
CIRCUS
DETROIT

HISTORY LESSON

AJAX was originally an acronym for Asynchronous JavaScript And XML. These days, people still use it, but the term Ajax actually refers to a group of web technologies used for asynchronous programming.

We're going to use "AJAX" moving forward.



ASYNCHRONOUS PROCESSING

When we use AJAX, the browser can request data without needing to wait to load the rest of the page. This works for loading an entire page as well as just parts of a page.

GRAND
CIRCUS
DETROIT

GRAND
CIRCUS
DETROIT

GRAND

GRAND
CIRCUS
DETROIT

GR
CIR
DET



grand circus fax

grand circus fax

grand circus

graphing calculator

grantland



Remove

Remove

About 12,000,000 results (0.38 seconds)

Grand Circus: Coding Bootcamps In Detroit

grandcircus.co/ ▾

Join a bootcamp in downtown Detroit, kickstart your career. At Grand Circus we train people intensively for new careers.

You've visited this page 2 times. Last visit: 6/14/15

GRAND
CIRCUS
DETROIT

GRAND
CIRCUS
DETROIT

GRAND
CIRCUS

GRAND
CIRCUS
DETROIT

GRAND
CIRCUS
DETROIT

GRAND
CIRCUS

GRAND
CIRCUS
DETROIT

GRAND
CIRCUS
DETROIT

GRAND
CIRCUS
DETROIT

DETROIT

GRAND
CIRCUS
DETROIT

GRAND
CIRCUS
DETROIT

GRAND
CIRCUS
DETROIT

PROMISES

GRAND
CIRCUS
DETROIT

GRAND
CIRCUS

GRAND
CIRCUS
DETROIT

GRAND
CIRCUS

GRAND
CIRCUS
DETROIT

PROMISE

Analogy: Restaurant Pager



GRAND
CIRCUS
DETROIT

GRAND
CIRCUS

GRAND
CIRCUS
DETROIT

PROMISE

In JavaScript, there is a Promise object. This object has a few properties, most important are two methods: `resolve` and `reject`. When you construct a promise, these two methods are going to be used to handle the state of the promise. Resolve is used when the operation is completed, reject is used when the operation fails.

```
var p = new Promise(function(resolve, reject) {  
});
```

PROMISE

Promises are sort've like callback functions, but get to handle success or errors. Let's look at an example of handling a success first.

```
var p = new Promise(function(resolve, reject) {  
  var status = true;  
  if(status === true) {  
    resolve("The promise is a success");  
  }  
});
```

PROMISE

We can call the resolve() method from within our promise, and pass any value we want.

```
p.then(function(resolve) {  
  console.log(resolve);  
})
```

PROMISE

If there is an error, we can call the reject() method.
Instead of chaining a .then() method, we use the
.catch() method to handle any errors.

```
var p = new Promise(function(resolve, reject) {  
  
    var status = true;  
    if(status === true) {  
        resolve("The promise is a success");  
    } else {  
        reject("The promise was not resolved");  
    }  
});  
  
p.then(function(resolve) {  
    console.log(resolve);  
}).catch(function(reject) {  
    console.log(reject);  
});
```

PROMISE

We can chain promises together, using multiple .then() methods. In this example, we are calling the resolve() method with a value of 0. We are then chaining together some promises that will keep returning updated values.

```
var p = new Promise(function(resolve, reject) {
  resolve(0);
});

p.then(function(resolve) {
  console.log(resolve + 20);
  return resolve + 20;
}).then(function(resolve) {
  console.log(resolve + 10);
  return resolve + 10;
})
```

CIRCUS
DETROIT

GRAND
CIRCUS
DETROIT

CIRCUS
DETROIT

GRAND
CIRCUS
DETROIT

GRAND
CIRCUS
DETROIT

GRAND
CIRCUS
DETROIT

GRAND
CIRCUS
DETROIT

PROMISE

Using a promise, however, is a good choice when you need something to run only *after* the code has completed.

GRAND
CIRCUS
DETROIT

JQUERY & AJAX



GRAND CIRCUS DETROIT



METHODS

jQuery provides four methods to handle our AJAX requests.

- `$.get()`
- `$.post()`
- `$.getJSON()`
- `$.getScript()`

GRAND
CIRCUS
DETROIT

CIRCUS
DETROIT

GRAND
CIRCUS
DETROIT

GRAND

GRAND
CIRCUS
DETROIT

GRAND

USING \$.GET()

GRAND
CIRCUS
DETROIT

CIRCUS
DETROIT

GRAND
CIRCUS
DETROIT

GRAND
CIRCUS
DETROIT

GRAND
CIRCUS
DETROIT

GRAND
CIRCUS
DETROIT

USING `$.GET()`

1. Define the request.
2. Define a function to handle the response.

DEMO



CIRKUS
DETROIT



GRAND
CIRCUS
DETROIT

ASYNCHRONOUS

```
var message = "before";  
  
$("button").on("click", function() {  
    message = "success";  
});  
  
console.log(message); // > "before"
```

ASYNCHRONOUS

```
var message = "before";  
  
$.get("https://www.reddit.com/r/aww/.json", function(responseBody) {  
    message = "success";  
});  
  
console.log(message); // > "before"
```

CIRCUS
DETROIT

GRAND
CIRCUS
DETROIT

CIRCUS
DETROIT

GRAND
CIRCUS
DETROIT

GRAND
CIRCUS
DETROIT

GRAND
CIRCUS
DETROIT

ASYNCHRONOUS

```
var message;  
  
$.get("https://www.reddit.com/r/aww/.json", function(responseBody) {  
    message = responseBody;  
});  
  
console.log(message); // > undefined  
console.log(responseBody); // > undefined
```



ASYNCHRONOUS

```
$.get("https://www.reddit.com/r/aww/.json", function(responseBody) {  
  // Put all your code to handle the response inside this function.  
  console.log(responseBody);  
});
```





SUCCESS & FAILURE



FAILURE HAPPENS

Every so often, you will make a request of the server and it will fail. This is inevitable, so be ready! Plan ahead for moments like these so your page isn't completely broken with every little thing that doesn't go as expected.

GRAND
CIRCUS
DETROIT

ČÍRCÚŠ
DETROIT

GRAND
CIRCUS
DETROIT

ČÍR
DET

GRAND
CIRCUS
DETROIT

HANDLING IT EITHER
WAY

GRAND
CIRCUS
DETROIT

GRAND
CIRCUS
DETROIT

GRAND
CIRCUS
DETROIT

GRAND
CIRCUS
DETROIT

GRAND

GRAND

GRAND

GRAND
CIRCUS
DETROIT

CIRCUS
DETROIT

GRAND
CIRCUS
DETROIT

The following methods can be chained after any of
jQuery's shorthand AJAX methods.

GRAND
CIRCUS
DETROIT

GRAND

GRAND
CIRCUS
DETROIT

GRAND

CIRCUS
DETROIT

GRAND
CIRCUS
DETROIT

CIRCUS
DETROIT

GRAND
CIRCUS
DETROIT

GRAND
CIRCUS
DETROIT

GRAND
CIRCUS
DETROIT

SUCCESS!

Code passed to the `done()` method will only run if the request is completed successfully.



FAILURE :(

Code passed to the `fail()` method will only run if the request is *not* completed successfully.

EVERY TIME

Code passed to the `always()` method will run regardless of the status of the request.

CIRCUS
DETROIT

GRAND
CIRCUS
DETROIT

GRAND
CIRCUS
DETROIT

GRAND
CIRCUS
DETROIT

GRAND

GRAND
CIRCUS
DETROIT

DEMO

GRAND

```
$.get("https://www.reddit.com/r/aww/.json").done(function(responseBody) {  
  console.log("DONE", responseBody.data.children[0].data.title);  
}).fail(function() {  
  console.log("FAIL");  
}).always(function() {  
  console.log("ALWAYS");  
});
```

See it live

GRAND
CIRCUS
DETROIT

GRAND
CIRCUS
DETROIT

GRAND
CIRCUS
DETROIT

GRAND
CIRCUS
DETROIT

GRAND

GRAND
CIRCUS
DETROIT

GRAND
CIRCUS
DETROIT

CIRCUS
DETROIT



HOMEWORK

From JavaScript & jQuery:

- Chapter 7: 359-366



GRAND
CIRCUS
DETROIT

GRAND
CIRCUS
DETROIT



GRAND
CIRCUS
DETROIT

GRAND
CIRCUS
DETROIT

NEW
CONTENT
AHEAD!



GRAND
CIRCUS
DETROIT

GRAND
CIRCUS
DETROIT



GRAND
CIRCUS
DETROIT



API GOTCHAS



GRAND

GRAND



API GOTCHAS

1. Same Origin Policy / CORS
2. API Keys
3. OAuth

SAME ORIGIN POLICY / CORS

A security feature of web browsers. You cannot make an AJAX requests to a different domain unless the API you're calling specifically allows it.

This means some APIs you simply *cannot use from the browser*. There's nothing you can do about it.



SAME ORIGIN POLICY / CORS

The solution is you have to write a back-end for your web app to access these APIs. We'll learn more about that later.



GRAND
CIRCUS
DETROIT

GRAND
CIRCUS
DETROIT

GRAND
CIRCUS
DETROIT

API KEYS

Most APIs require you to register and get a "key", kinda like a password.

Every time you use the API, you have to send the key with the request.



API KEYS

Sometimes you send the key as a URL parameter.

```
$.get({
  url: "https://api.flickr.com/services/rest/?method=flickr.test.echo",
  data: {
    name: "Hello Flickr!",
    api_key: "YOUR KEY HERE"
  },
  function(responseBody) {
    console.log(responseBody);
  });
}
```



API KEYS

Sometimes you send the key as a header.

```
$ .get({
  url: "https://yoda.p.mashape.com/yoda?sentence=I+have+a+bad+feeling+about+this!",
  headers: {
    "X-Mashape-Key": "YOUR KEY HERE",
    "Accept": "text/plain"
  },
  function(responseBody) {
    console.log(responseBody);
  });
}
```

OAUTH

This is a pretty complicated system that many APIs require you to use in order to have people sign into their service so that you can use their data.

We won't cover the details. Plenty of groups for final projects have tackled OAuth for final projects. With every API being different, it really just comes to reading the documentation.



NEW
CONTENT
AHEAD!





-DETROIT-



PLUGINS



GOALS FOR THIS UNIT

1. Review
2. Popular plugins
3. Incorporating plugins

GRAND
CIRCUS
DETROIT

REVIEW

GRAND
CIRCUS
DETROIT

GRAND
CIRCUS
DETROIT

GRAND
CIRCUS
DETROIT

GRAND
CIRCUS
DETROIT

GRAND
CIRCUS

GRAND
CIRCUS
DETROIT

DETROIT

GRAND
CIRCUS

GRAND
CIRCUS
DETROIT

GRAND
CIRCUS
DETROIT

GRAND
CIRCUS
DETROIT

WHAT'S A PLUGIN?

Plugins extend the functionality of jQuery by providing developers with additional methods that can be used on jQuery selections.

FINDING A PLUGIN

Curb your urge to Google for just a moment and have a look at these lists of jQuery plugins:

- npm
- Sitepoint's Popular jQuery Plugins List
- Unheap

CHOOSING A PLUGIN

Some things to keep in mind when deciding whether or not a particular plugin is right for your project:

- How many times has it been downloaded?
- When was the last time it was updated?
- How many open issues are associated with it?
- How responsive are its publishers to issues and feature requests?



POPULAR PLUGINS

LIGHTBOX

GRAND
CIRCUS
DETROITGRAND
CIRCUS
DETROITGRAND
CIRCUS
DETROITGRAND
CIRCUS
DETROITGRAND
CIRCUS
DETROITGRAND
CIRCUS
DETROITGRAND
CIRCUSGRAND
CIRCUS
DETROITGRAND
CIRCUS
DETROITGRAND
CIRCUS

THE OG OF LIGHTBOX PLUGINS

There are more plugins to display larger versions of images in a modal view than you could shake a stick at, but the original [Lightbox](#) is still a solid choice.

GRAND
CIRCUS
DETROIT

DETROIT

GRAND
CIRCUS
DETROIT

DETROIT

GRAND
CIRCUS
DETROIT

DEMO

GRAND
CIRCUS
DETROIT

GRAND
CIRCUS
DETROIT

GRAND
CIRCUS
DETROIT



GRAND
CIRCUS
DETROIT

GRAND
CIRCUS

GRAND
CIRCUS
DETROIT

GRAND
CIRCUS
DETROIT

GRAND
CIRCUS
DETROIT

GRAND
CIRCUS
DETROIT

GRAND
CIRCUS

ANIMATION



THE NEW KID

There are plenty of plugins to make creating interactive animations easier. One popular (relative) newcomer to the game is [animatedModal.js](#).

GRAD CIRCUS

DETROIT

Animate a modal!



DETROIT



DETROIT



DETROIT



DETROIT



DETROIT



DETROIT



DETROIT



DETROIT



DETROIT



DETROIT



GR
CI

JQUERY UI

GRAND
CIRCUS
DETROIT

CIRCUUS
DETROIT

GRAND
CIRCUS
DETROIT

GRAND
CIRCUS
DETROIT

GRAND
CIRCUS
DETROIT

GRAND
CIRCUS
DETROIT

SISTERS? WE'RE CLOSE!

jQuery UI is a sister project to jQuery. It has a wide variety of user interface functionality that layers on top of jQuery itself.

THE DOWNLOAD BUILDER

jQuery UI gives us a ton of options to keep our site as light as possible. Using the [Download Builder](#), we can specify only the interactions, widgets, and effects we actually need.

HOMEWORK

- JS & jQuery pgs. 359-366

EXERCISE

Work with your pair. Choose a jQuery plugin from the list on the next slide. Implement a simple website featuring this plugin demonstrating it's basic functionality.

JQUERY PLUGINS

Some Plugins ya'll

Alertify.js Swipebox Reveal

Countdown Complexify.js SpaceGallery

Pickadate.js Hook.js Airport

Waypoints HeatMap Skrollr

Pizza ScrollPath 20x20

Lettering.js Bacon! gridster.js

Cycle2 Backstretch Tubular

Raptorize Fool.js BallDroppings.js