

GRAND
CIRCUS

DETROIT

Build a tip calculator that takes a subtotal from a text input box and one of three percentages, 15, 20 or 25, that can be chosen with radio buttons. Your calculator should then append the total + tip to the page.

Once done, try refactoring to allow a user to enter a custom tip percentage, but any percentage lower than 15% is automatically updated to 15%

JQUERY

WELCOME BACK!

Group Projects!

1. Show us your final product, but most importantly, show us your code!
2. What's something you (individually) learned?
3. What's one thing you struggled with?
4. What's one thing you learned that will be helpful for your next group project?

GOALS FOR THIS UNIT

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

REVIEW

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

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.

SOLID DOCS

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

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.



jQuery is still just JavaScript!

INCLUDING JQUERY

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

WHICH IS BEST?

For active development, it's fine to link.

For production sites, I usually play it safe and source jQuery locally.

DOWNLOADING JQUERY

1. Go to **jquery.com/download**
2. Click the link Download the uncompressed, development jQuery **1.11.3**. This takes you to a raw text file.
3. Select All > copy > paste into empty file > save As 'jquery-1.11.3.js'
4. Place the saved file into a 'lib' folder in your site project

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

```
ib/jquery-1.11.3.js"></script>  
tpp://code.jquery.com/jquery-1.11.3.min.js"></script>
```

JQUERY SELECTORS

Anything look familiar?

```
div");
the page
= $( "#happy" );
h id "happy"
(".roundedCorner");
th class "roundedCorner"
```

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

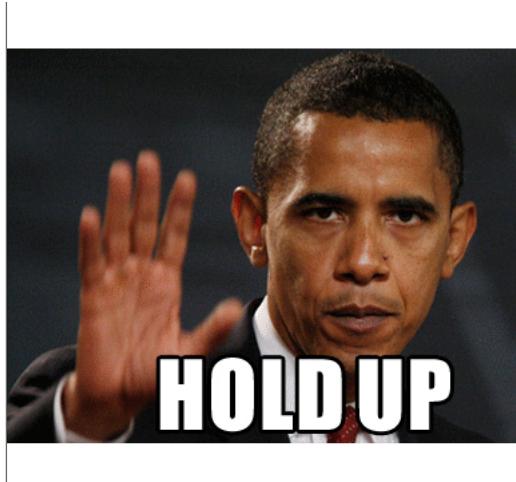
jQuery even adds CSS-like **filters** to make selecting elements even easier.

```
er( ":even" ).css( "background-color", "red" );
```

STORING JQUERY OBJECTS

jQuery objects are typically stored in variables. Variables containing jQuery objects conventionally begin with "\$" to make it clear that they're *jQuery* objects rather than DOM elements.

```
$( 'slow' , 0.5 );  
img );  
low, 0.5 );
```



What was that dollar sign thingy?!

`$()`

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

GETTING READY

[] is a jQuery method which tells a script to wait to execute until an HTML document has loaded.

```
ady(function() {  
slideDown('slow');
```

JQUERY ACTIONS

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

```
tion( );
```

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

```
myPicture') ;  
) ; 'http://myPictureLivesHere.com') ;  
' ) ;  
' , '200px') ;
```

CHAINING

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

```
hide().delay(1200).fadeIn(800);
```

LAB 5

SEAT RESERVATION - JQUERY



**WELL I DON'T SEE ANYBODY SITTING
HERE**

4GIFs.com

INSTRUCTIONS

Complete the seat reservation app by using jQuery to make it interactive.

- Users should be able to choose an available seat to reserve and enter their information into a form.
- Submitting this form should create an object that represents that user and is associated with that seat. Users should be able to choose multiple seats.
- Use jQuery selectors and actions in your JavaScript.

Group members should make a branch for the feature they're working on.

INSTRUCTIONS (CONT.)

- On hover (for reserved seats), make the associated user's information appear on the screen.
- If a user attempts to reserve an unavailable seat, the form should remain hidden and an error message should display.
- Different price tiers for different seats. The seats in the front two rows should have a higher ticket price than the seats in the back two rows.
- A user should be presented with a total cost of their order.

INSTRUCTIONS (CONT.)

- Validate the name and email fields to prevent the user from entering incorrectly formatted info, leaving fields blank, or making multiple reservations.

FIGURE IT OUT

PART 1 - Create a form with at least five separate inputs. Hide all but the first and show the rest only when the previous input is in focus.

PART 2 - Use jQuery to build a "form-appending machine" that, when invoked by your event handler, appends a new input field to the page. PLEASE NOTE - you will need a template that your jQuery can use to make copies of, you are advised not to start from scratch with no HTML.

EXERISE

REFACTOR your email and password validator using jQuery. If email and password are correct, those fields should turn green. If email is correct but password is not, the password field should turn green. If email is incorrect but password is correct, both fields should be red. If both are incorrect, both fields should turn red.



JQUERY METHODS & EVENTS

GOALS FOR THIS UNIT

1. Review
2. Ready method
3. More on Methods
4. Effects
5. Events

REVIEW

READY METHOD

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 method means the browser will still wait for the DOM to load before running any scripts, regardless of where they're placed.

```
ady(function(){  
t-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.

t-y things here

READY VS. EVERYBODY

Other options include relying on the load event (if your script depends on other assets) or simply placing scripts at the very end of the body of your HTML. In the latter case, developers will often still use the `[]` method in case the script tag is moved later.

RETRIEVING CONTENT

The `_____` 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.

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

```
ontainer" ).html();
```

The above would return:

```
emo-box" >Demonstration Box < /div >
```

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

```
ox" ).text( "New Demo Box Text!" );
```

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

UPDATING CONTENT

The and methods can not only retrieve content, but update it as well.

The method replaces matched elements with new content and returns the replaced elements.

The method removes any elements in the matched set.

EXERCISE

Create a bare HTML document and include jQuery in your head section. IN CONSOLE and using JQUERY, add a div with a class-name that you can reference. Then, using jQuery, add text to your div, change the background-color, the text-color, the size of your text, and at least one other property.

INSERTING CONTENT

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

```
before( "<p>Test</p>" );
```

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

```
after( "<p>Test</p>" );
```

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

```
prepend( "<p>Test</p>" );
```

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

```
append( "<p>Test</p>" );
```

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.

MAKING SPACE

When an element is hidden, other elements may move to take up the empty space left in the page. When an element is shown, others will move to make room.

TOGLING

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

BASIC EFFECTS

Method Description

.hide()	Hide the matched elements.
.show()	Display the matched elements.
.toggle()	Display or hide the matched elements.

CUSTOM EFFECTS

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.

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.

SLIDING EFFECTS

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.



JQUERY EVENTS

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, we would've used different methods for each type of event we wanted our code to respond to.

THE BREAKDOWN

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

THE BREAKDOWN

2. We'll use the method to handle the event.

()

THE BREAKDOWN

3. The first argument we pass to the method is the event we want to respond to.
-

('click')

THE BREAKDOWN

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

```
('click', function(){  
  You clicked a button!});
```

LAB 6

TOGGLE CLASSES

INSTRUCTIONS

Practice using jQuery methods on a simple web page.

- Create a simple HTML page that includes an unordered list.
- Each should fade to 0.25 opacity when the user hovers over it.
- Each should change background colors when clicked on.
- When the user clicks on a , the word 'Clicked!' should appear inside it.
- Changes triggered by a click should change back when the is clicked again.

Apply what we've learned about effects and events to enhance your shopping list app

Build a credit card validator with jQuery that accepts and verifies - a credit card number that must be 16 numbers in length, a CVV number that must be 3 numbers in length, a name and an expiration date. If the validations for an input field pass upon a change in focus, your jQuery must change the input field's background color to green. For example, if a credit card number is 16 numbers, send the focus to the CVV field.

FIGURE IT OUT

Create a class that specifies both the appearance and position of a given element.
Toggle this class when matched elements are clicked on.



AJAX & JSON

GOALS FOR THIS UNIT

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

REVIEW

AJAX

HISTORY LESSON

AJAX was originally an acronym for Asynchronous JavaScript And XML. These day 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.

The screenshot shows a Google search results page. The search query in the bar is "grand circus fax". Below the bar, there are four suggestions with "Remove" buttons: "grand circus fax", "grand circus", "graphing calculator", and "grantland". The first result is a link to "Grand Circus: Coding Bootcamps In Detroit" from "grandcircus.co/". The snippet for this result reads: "Join a bootcamp in downtown Detroit, kickstart your career. At Grand Circus we train people intensively for new careers." Below the snippet, it says "You've visited this page 2 times. Last visit: 6/14/15".

THE 1, 2 PUNCH

AJAX works in two steps:

1. The browser requests information from a server.
2. The browser processes the server's response.

The stuff the server does in between these two steps? That's a bunch of server-side nonsense that we don't need to worry about right now.

ALL-IN-ONE OBJECT

The browser uses the XMLHttpRequest object in both AJAX steps.

REQUEST

THE BREAKDOWN

1. We begin by creating an instance of the XMLHttpRequest object and storing it in a variable.
-

```
XMLHttpRequest();
```

THE BREAKDOWN

2. We use the method to prepare the request.
-

```
XMLHttpRequest();
```

THE BREAKDOWN

- 2a. The first parameter of the method takes either 'GET' or 'POST' and specifies how the request should be sent.
-

```
XMLHttpRequest( );  
);
```

THE BREAKDOWN

2b. The second parameter of the method takes the path to the page that will handle the request.

```
XMLHttpRequest();  
, 'data/data.html');
```

THE BREAKDOWN

2c. The third parameter of the method takes a Boolean value that indicates whether or not the request is asynchronous.

```
XMLHttpRequest();  
, 'data/data.html', true);
```

THE BREAKDOWN

3. Once the request has been prepared, we use the method to (you guessed it) send the request to the server.
-

```
XMLHttpRequest();  
, 'data/data.html', true);
```

THE BREAKDOWN

BONUS!

3a. Optionally, we can pass additional information to the server using the method. In the absence of any such information, you may see the keyword passed.

```
XMLHttpRequest( );
, 'data/data.html', true);
;
```

RESPONSE

THE BREAKDOWN

1. The event is triggered when a response is received from the server and loaded.
-

```
XMLHttpRequest( );  
  
, 'data/data.html', true);  
;
```

THE BREAKDOWN

2. We can use the event to trigger a function.

```
XMLHttpRequest( );  
unction() {  
  
, 'data/data.html', true);  
;
```

THE BREAKDOWN

3. Within that function, we need to make sure the server responded successfully by checking for a status of 200.
-

```
XMLHttpRequest();  
unction() {  
us === 200) {  
  
, 'data/data.html', true);  
;  
;
```

THE BREAKDOWN

4. If the response is okay, we can go ahead and use the information from the server to update the page.
-

```
XMLHttpRequest();  
  
unction() {  
us === 200) {  
hat uses the server's response  
  
, 'data/data.html', true);  
;  

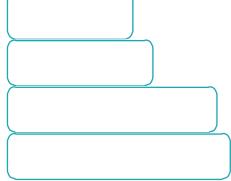

---


```

JQUERY & AJAX

METHODS

jQuery provides four methods to handle our AJAX requests.

- 
-
-
-

SENDING & SUBMITTING FORMS

COLLECTING DATA

In addition to the four methods already mentioned, jQuery provides us with the `serialize` method, which takes all the information contained in a form, puts it into a string we can send to the server (encoding characters when necessary).

THE BREAKDOWN

1. Select the form.
-
-

THE BREAKDOWN

2. Using the method, we'll create a block of code to be run when the form is submitted. The first argument is the trigger and the second is the function that will respond to that trigger.

```
.on('submit', function(e){
```

THE BREAKDOWN

3. We can prevent the form from submitting immediately (which would be its default behavior).
-

```
.on('submit', function(e){  
    e.preventDefault();  
});
```

THE BREAKDOWN

4. To prepare the form data to be sent to the server, we use the method.

```
.on('submit', function(e){  
    e.preventDefault();  
    var form = $('#register').serialize();  
    // do something with the data  
});
```

THE BREAKDOWN

5. The method sends the data.

```
.on('submit', function(e){  
  e.preventDefault();  
  var $form = $('#register').serialize();  
  $.post('register.php', $form, function(data) {  
    // handle response  
  });  
});
```

THE BREAKDOWN

6. The function passed to the method indicates where the result should be displayed.
-

```
.on('submit', function(e){  
  e.preventDefault();  
  var $form = $('#register').serialize();  
  $.post('register.php', details, function(data) {  
    $('#register').html(data);  
  });  
});
```

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.

HANDLING IT EITHER WAY

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

SUCCESS!

Code passed to the method will only run if the request is completed successfully.

FAILURE :(

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

EVERY TIME

Code passed to the method will run regardless of the status of the request

LAB 7

POOR MAN'S REDDIT

INSTRUCTIONS

Take the JSON payload available from [/r/aww](#) (check with the instructor before using anything else) and use the data to create a “feed”.

1. Break into small groups and delegate tasks.
2. Look at the raw JSON to see how objects are organized.
3. Create a webpage that pulls information from the payload.

FIGURE IT OUT

Create a "back to top" button that persists near the bottom of the screen and scrolls (as opposed to jumps) back to the top of the page without reloading it.



PLUGINS

GOALS FOR THIS UNIT

1. Review
2. Popular plugins
3. Incorporating plugins
4. Building plugins

REVIEW

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

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.

DEMO



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](#).

DEMO

Animate a modal!

JQUERY UI

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.

LAB 8

Choose a JavaScript library and use it to create a widget that can be added to a web page.

EXAMPLES

- [Leaflet](#)
- [ChessboardJS](#)
- [Bounce.js](#)
- [Please](#)
- [Shepherd](#)