
WEEKLY OVERVIEW

WEEK 5

jQuery / JS Lab

WEEK 6

Interactions / Functions

LEARNING OBJECTIVES

- Practice listening for and responding to events such as scroll, submit, keyup, and mouseenter.
- Use the `val()` method to get and set information for form fields.
- Practice using jQuery methods and events, JavaScript control flow, and variables by building out a Mad Libs game.

AGENDA

Review

jQuery Events

Working with User Input

Mad Libs Lab

JS LAB

REVIEW

JQUERY — REVIEW



EXERCISE

KEY OBJECTIVE

- Review jQuery selectors, methods and events

TYPE OF EXERCISE

- Groups of 2 - 3

STEPS

8 *min*

1. In the start files, review `select_fun > js > main.js`.
2. With your HTML and JS files open, chat through each part of the lab and describe what is happening, in human terms.
3. Have one person in the group jot down 1 - 2 questions

USING JQUERY TO MANIPULATE THE DOM

1

Select an element/elements

2

Work with those elements

JQUERY — SELECTING ELEMENTS

Selector

```
$('li').addClass('selected');
```

jQuery Function:

- ▶ Lets us find one or more elements in the page
- ▶ Creates a *jQuery object* which holds references to those elements
- ▶ We'll be using the shorthand in this class: `$()`
- ▶ `$(selector)` is the same as `jQuery(selector)`

USING JQUERY TO MANIPULATE THE DOM

1

Select an element/elements

2

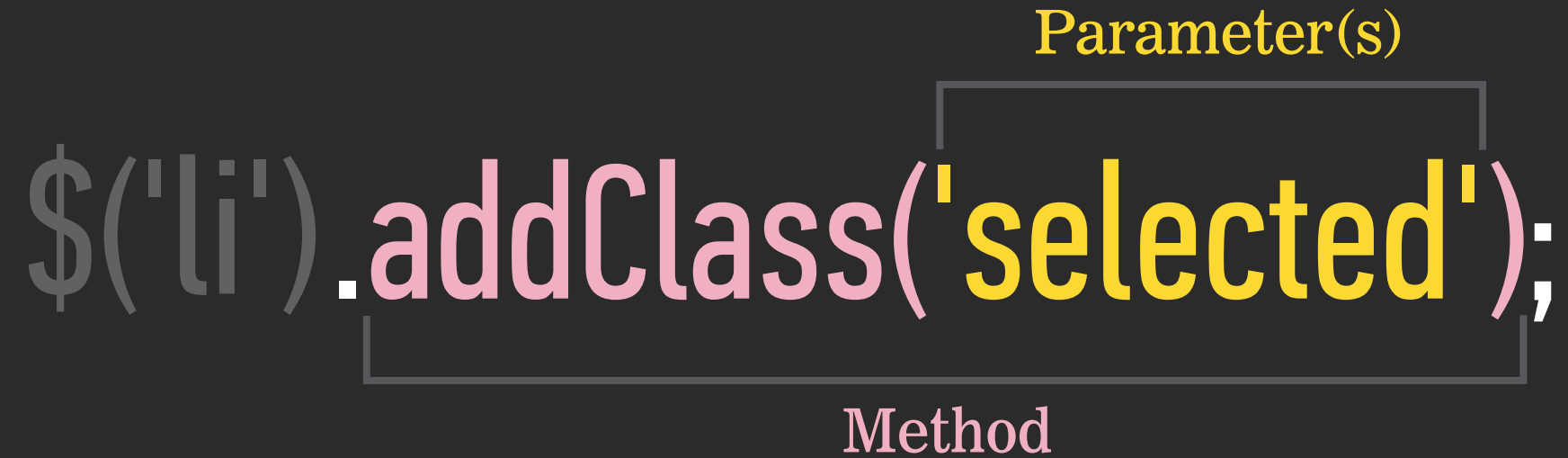
Work with those elements

JQUERY — WORKING WITH THOSE ELEMENTS

Parameter(s)

```
$('.li').addClass('selected');
```

Method

The diagram shows the jQuery code snippet `$('.li').addClass('selected');` with two annotations. A bracket above the code spans from the opening curly brace of the selector to the closing parenthesis of the method call, with the label "Parameter(s)" centered above it. A bracket below the code spans from the dot after the selector to the closing parenthesis of the method call, with the label "Method" centered below it.

JQUERY METHODS — TRAVERSING THE DOM

TRAVERSE THE DOM

- ▶ These methods to find/select elements to work with & traverse the DOM
- ▶ Think of these as filters, or part of the selection process.
- ▶ They must come *directly after another selection*

METHODS	EXAMPLES
<code>.find()</code> <i>finds all descendants</i>	<code>\$('h1').find('a');</code>
<code>.parent()</code>	<code>\$('#box1').parent();</code>
<code>.siblings()</code>	<code>\$('p').siblings('.important');</code>
<code>.children()</code>	<code>\$('ul').children('li');</code>

What goes in the parentheses?
A css-style selector

JQUERY METHODS — GETTING/SETTING CONTENT

GET/SET CONTENT

Get/change content of elements and attributes

METHODS	EXAMPLES
<code>.html()</code>	<code>\$('#h1').html('Content to insert goes here');</code>
<code>.attr()</code>	<code>\$('#img').attr('src', 'images/bike.png');</code>
<code>.css()</code>	<code>\$('#box1').css('color', 'red');</code>
<code>.addClass()</code>	<code>\$('#p').addClass('success');</code>
<code>.removeClass()</code>	<code>\$('#p').removeClass('my-class-here');</code>
<code>.toggleClass()</code>	<code>\$('#p').toggleClass('special');</code>

What goes in the parentheses?
The **html**, **styles**, **classes** you want to change.

ADD CLASS

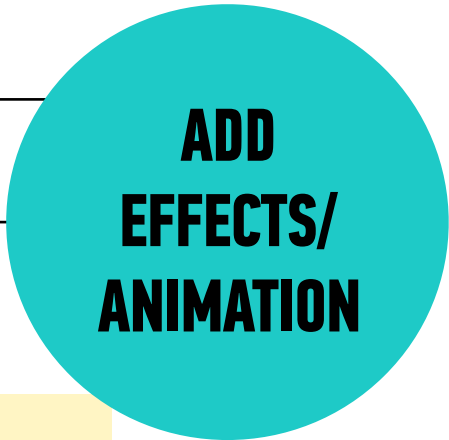
`$('h1').addClass('x fun')`



REMEMBER — NO PERIOD!!

`$('h1').addClass(' fun')`

JQUERY METHODS — EFFECTS/ANIMATION



Add effects and animation to parts of the page

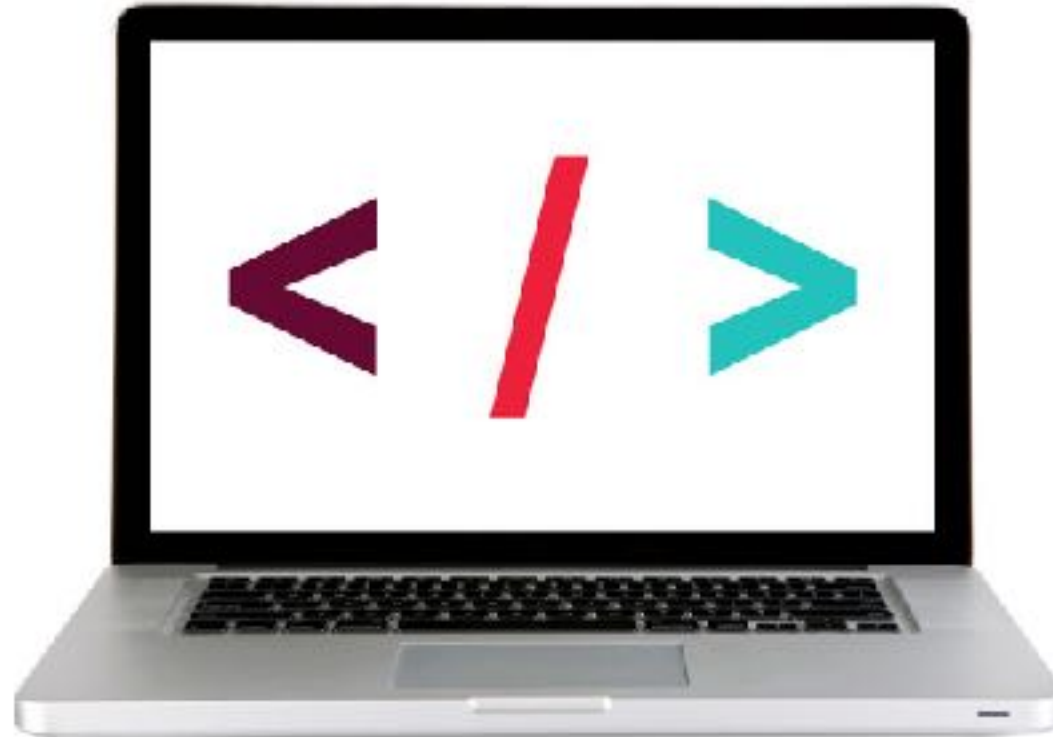
METHODS	EXAMPLES
<code>.show()</code>	<code>\$('#h1').show();</code>
<code>.hide()</code>	<code>\$('#ul').hide();</code>
<code>.fadeIn()</code>	<code>\$('#h1').fadeIn(300);</code>
<code>.fadeOut()</code>	<code>\$('.special').fadeOut('fast');</code>
<code>.slideUp()</code>	<code>\$('#div').slideUp();</code>
<code>.slideDown()</code>	<code>\$('#box1').slideDown('slow');</code>
<code>.slideToggle(), .fadeToggle()</code>	<code>\$('#p').slideToggle(300);</code>

What goes in the parenthesis?
An animation speed

JQUERY

METHOD CHAINING

ACTIVITY — METHOD CHAINING



METHOD CHAINING!!!

JQUERY METHODS — METHOD CHAINING

```
$( )
```

```
.slideUp( )
```

```
'li'
```

```
'slow'
```

JQUERY METHODS — METHOD CHAINING

```
$( 'li' )
```

```
.slideUp( 'slow' )
```

JQUERY METHODS — METHOD CHAINING

```
$( 'li' ).slideUp( 'slow' );
```

JQUERY METHODS — METHOD CHAINING

`$()`

`.addClass()`

`'li'`

`'.complete'`

`'complete'`

JQUERY METHODS — METHOD CHAINING

```
$( 'li' )
```

```
.addClass( 'complete' )
```

JQUERY METHODS — METHOD CHAINING

```
$( 'li' ).addClass( 'complete' );
```

JQUERY METHODS — METHOD CHAINING

`$()`

`.html()`

`'li'`

`300`

`'Feed cat'`

JQUERY METHODS — METHOD CHAINING

```
$( 'li' )
```

```
.html( '<li>Feed cat</li>' )
```

JQUERY METHODS — METHOD CHAINING

```
$( 'li' ).html( '<li>Feed cat</li>' );
```

JQUERY METHODS — METHOD CHAINING

`$()`

`.show()`

`.siblings()`

`'h3'`

`'p'`

JQUERY METHODS — METHOD CHAINING

```
$('#h3')
```

```
.show()
```

```
.siblings('p')
```

JQUERY METHODS — METHOD CHAINING

```
$('#h3').siblings('p').show();
```

JQUERY METHODS — METHOD CHAINING

`$()`

`.slideUp()`

`.find()`

`' .item '`

`300`

`'h2'`

JQUERY METHODS — METHOD CHAINING

```
$('.item')
```

```
.slideUp(300)
```

```
.find('h2')
```

JQUERY METHODS — METHOD CHAINING

```
$('.item').find('h2').slideUp(300);
```

JQUERY METHODS — METHOD CHAINING

`$()`

`.fadeOut()`

`.children()`

`'#main'`

`'slow'`

`'p'`

JQUERY METHODS — METHOD CHAINING

```
$('#main')
```

```
.fadeOut('slow')
```

```
.children('p')
```

JQUERY METHODS — METHOD CHAINING

```
$('#main').children('p').fadeOut('slow')
```

JQUERY EVENTS - PART II

JQUERY METHODS — EVENTS!



**CREATE
EVENT
LISTENERS**

We can use the `on()` method to handle all events in jQuery.

JQUERY METHODS — EVENTS!

**CREATE
EVENT
LISTENERS**

selector

```
$('li').on('click', function() {  
    // your code here  
});
```

JQUERY METHODS — EVENTS!

**CREATE
EVENT
LISTENERS**

method for all events

```
$( 'li' ).on( 'click', function() {  
    // your code here  
} );
```

JQUERY METHODS — EVENTS!

**CREATE
EVENT
LISTENERS**

type of event

```
$( 'li' ).on( 'click', function() {  
    // your code here  
});
```

MOUSE

click
dblclick
mouseenter
mouseleave

KEYBOARD

keypress
keydown
keyup

FORM

submit
change
focus
blur

DOCUMENT

resize
scroll



```
$('#li').on('eventGoesHere', function() {  
  // your code here  
});
```

JQUERY METHODS — EVENTS!



CREATE EVENT LISTENERS

```
$('.li').on('click', function() {  
    // your code here  
});
```

function to run
when event is
triggered

JQUERY METHODS — EVENTS!

CREATE EVENT LISTENERS

selector method for
all events type of
event

```
$('li').on('click', function() {  
    // your code here  
});
```

function to run
when event is
triggered

JQUERY — REVIEW



EXERCISE

KEY OBJECTIVE

- Review jQuery selectors and events, get practice looking up new event types

TYPE OF EXERCISE

- Individual/paired

SMALL GROUP PLANNING

5 min

1. Follow the instructions in Starter Code > jquery_events > js/main.js

WORKING WITH USER INPUT

JQUERY FORM EVENTS



KEYBOARD

keypress

keydown

keyup



FORM

submit

change

focus

blur

```
$('#li').on('eventGoesHere', function() {  
    // your code here  
});
```

SUBMIT EVENT

```
$( 'form' ).on( 'submit', function(event) {  
    event.preventDefault();  
    // your code here  
});
```

JQUERY METHODS — THE EVENT OBJECT

- The event object has properties and methods that tell you more about the event that took place.
- By using the `preventDefault()` method, the default action of the event will not be triggered.

Parameter name

```
$('#a').on('click', function(event) {  
    event.preventDefault();  
});
```

Use that name in the function and use dot notation to access its properties and methods.

WHEN TO PREVENT THE DEFAULT ACTION

- So when do I need to prevent the default action?

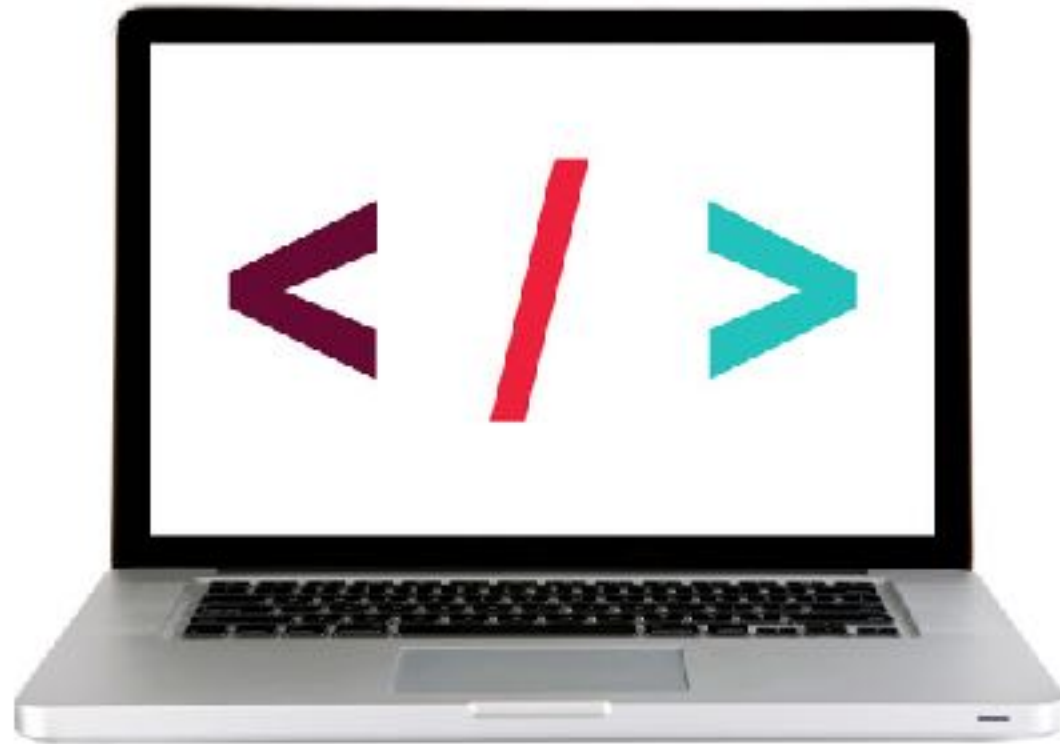
1. WHEN THE SELECTOR FOR OUR EVENT IS AN ANCHOR:

```
$('#a').on('click', function(event) {  
    event.preventDefault();  
});
```

2. WHEN THE TYPE OF EVENT IS A SUBMIT EVENT:

```
$('#form').on('submit', function(event) {  
    event.preventDefault();  
});
```

LET'S TAKE A CLOSER LOOK



SOME NEW METHODS!

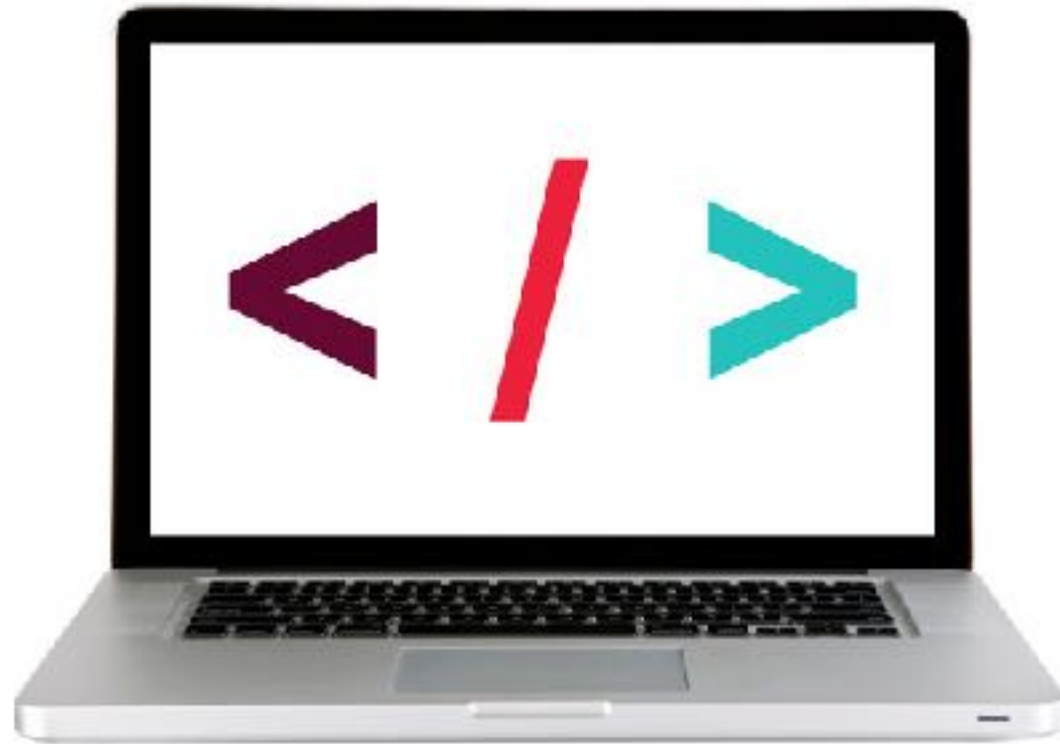
GET/SET CONTENT

Get/change content of elements, attributes, text nodes (part 2!)

METHODS	GOAL	EXAMPLES
.val()	Get value from input	<code>\$('input').val();</code>
	Change value in input	<code>\$('input').val('New Value');</code>

What goes in the parentheses?
The **html or content** you want to add/change

LET'S TAKE A CLOSER LOOK



LAB

MAD LIBS — PART 1



EXERCISE

KEY OBJECTIVE

- ▶ Review the HTML and CSS to create forms by building the interface for the Mad Libs game.

TYPE OF EXERCISE

- ▶ Individual/paired

SMALL GROUP PLANNING

15 min

1. Using the image in the starter code folder as a guide, write HTML and CSS to create the interface for the Mad Libs game.

MAD LIBS — PART 2



EXERCISE

KEY OBJECTIVE

- ▶ Practice breaking down a problem into steps using pseudo code.

TYPE OF EXERCISE

- ▶ Groups of 2 - 3

SMALL GROUP PLANNING

10 min

1. In groups of 2 - 3, write pseudo code for the Mad Libs game.
2. Together: Review the pseudo code for the application

MAD LIBS — PART 3



EXERCISE

KEY OBJECTIVE

- ▶ Practice using events, jQuery methods, and JavaScript conditionals to create a Mad Libs game.

TYPE OF EXERCISE

- ▶ Individual / Partner

SMALL GROUP PLANNING

1 hour

1. Write JS to add functionality to the Mad Libs game.
2. Bonus: Clear out all form fields after user has played game.
3. Bonus: Validate the form every time the user lifts a key up (keyup event).
4. Bonus: Add a color scheme switcher to the game.

LEARNING OBJECTIVES

- Practice listening for and responding to events such as scroll, submit, keyup, and mouseenter.
- Use the `val()` method to get and set information for form fields.
- Practice using jQuery methods and events, JavaScript control flow, and variables by building out a Mad Libs game.

WEEKLY OVERVIEW

WEEK 5

jQuery / JS Lab

WEEK 6

Interactions / Functions

EXIT TICKETS!