GENERAL ASSEMBLY

JAVASCRIPT DEVELOPMENT

Sasha Vodnik, Instructor

HELLO!

1. Pull changes from the `svodnik/JS-SF-13-resources` repo to your computer
2. Open the `07-jquery` > `starter-code` folder in your code editor

JAVASCRIPT DEVELOPMENT

Intro to jQuery

LEARNING OBJECTIVES

At the end of this class, you will be able to

- Create DOM event handlers using vanilla JavaScript.
- Select DOM elements and properties using jQuery.
- Manipulate the DOM by using jQuery selectors and functions.
- Create DOM event handlers using jQuery.

AGENDA

- JavaScript events
- jQuery
- Getting and setting DOM elements with jQuery
- Responding to events with jQuery

WEEKLY OVERVIEW

WEEK 5	DOM & jQuery / Events & jQuery
WEEK 6	Ajax & APIs / Asynchronous JS & callbacks

Break

WEEK 7	Advanced APIs / Project 2 lab

HOMEWORK — GROUP DISCUSSION

EXERCISE

TYPE OF EXERCISE

- Full class

TIMING

6 min

1. Show off your bot! What can it do?
2. Share a challenge you encountered, and how you overcame it.

HOMEWORK — GROUP DISCUSSION

EXERCISE

TYPE OF EXERCISE

- Pairs

TIMING

4 min

1. Share your solutions for the JSON & create-append homework.
2. Share a challenge you encountered, and how you overcame it.

EXIT TICKET QUESTIONS

1. What do APIs do? What other operators can we use for with query selector?
2. Still lots to learn about dom query selectors, obviously there's lots of syntax to learn or memorize there.
3. I don't think I could write the code for a for in loop off the top of my head yet without making a few mistakes!
4. Why didn't the nth-of-type selector work when in the .js file but ran without errors when run in the console?

EXERCISE — CATCH PHRASE

EXERCISE

TYPE OF EXERCISE

- Pairs

TIMING

4 min

1. Get your partner to guess the word on each piece of paper by giving clues describing it **without saying the word itself**.
2. Take turns giving clues and guessing words.

EVENTS

EVENT LISTENERS

selecting element

```
let button = document.querySelector(‘.submitBtn’);
```

element reference

```
button.addEventListener('click', function() {
  // your code here
}, false);
```

EVENT LISTENERS

```
let button = document.querySelector(‘.submitBtn’);
```

method to add event listener

```
button.addEventListener('click', function() {
  // your code here
}, false);
```

EVENT LISTENERS

```
let button = document.querySelector(‘.submitBtn’);
```

type of event

```
button.addEventListener('click', function() {
  // your code here
}, false);
```

MOUSE	KEYBOARD	FORM	DOCUMENT
click	keypress	submit	resize
dblclick	keydown	change	scroll
mouseenter	keyup	focus	
mouseleave		blur	

```
button.addEventListener('eventgoeshere', function() {
  // your code here
}, false);
```

EVENT LISTENERS

```
let button = document.querySelector(‘.submitBtn’);
```

```
button.addEventListener('click', function() {
  // your code here
}, false);
```

function to run when event is triggered

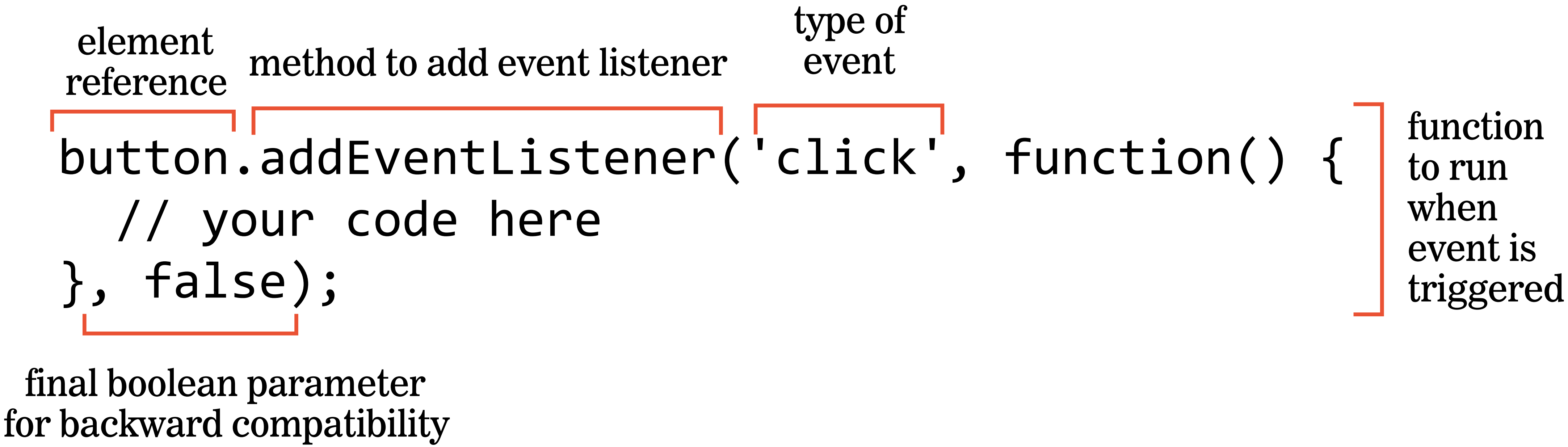
EVENT LISTENERS

```
let button = document.querySelector(‘.submitBtn’);
```

```
button.addEventListener('click', function() {
  // your code here
}, false);
```

final boolean parameter
for backward compatibility

EVENT LISTENERS



element reference

method to add event listener

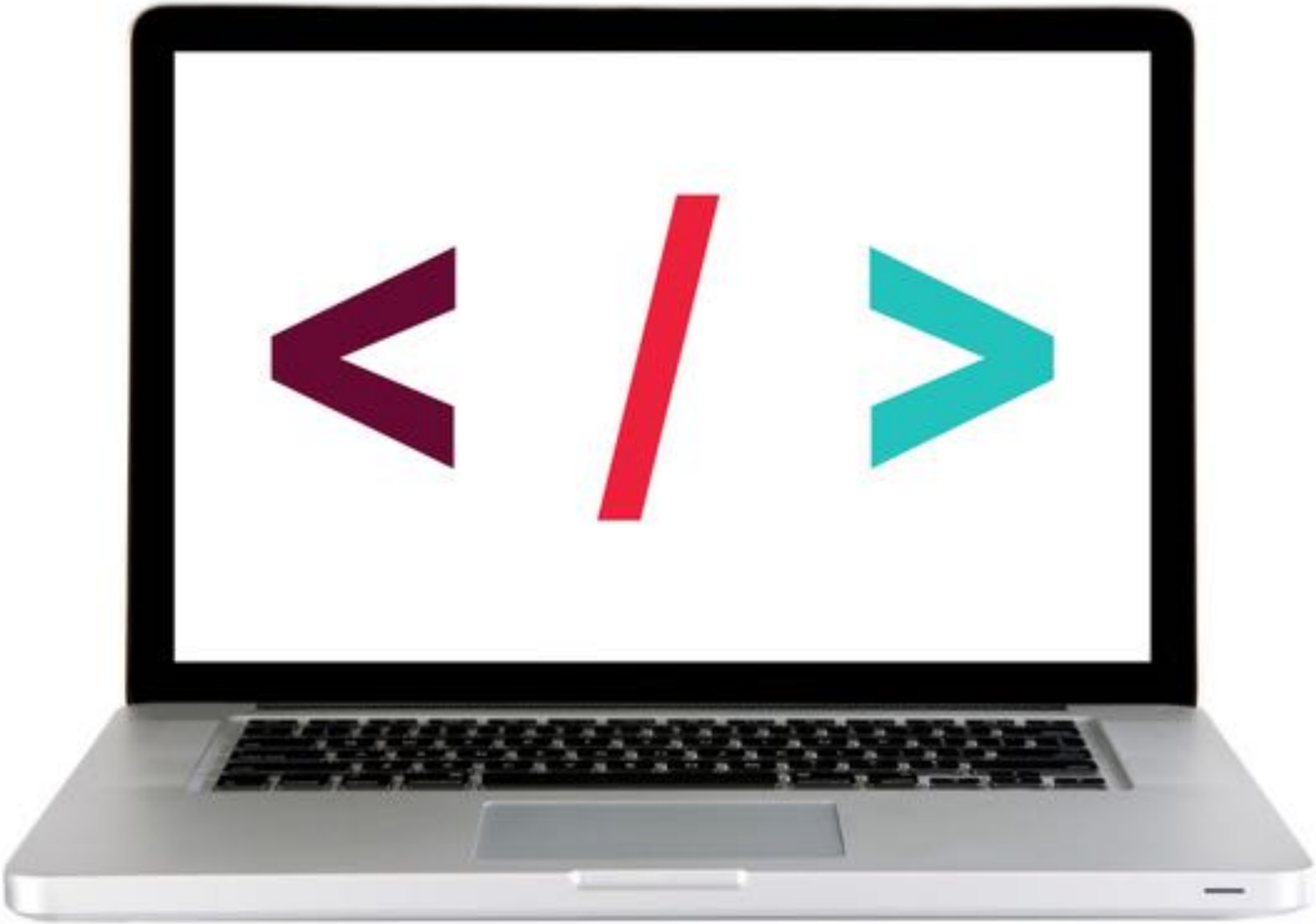
type of event

```
button.addEventListener('click', function() {
  // your code here
}, false);
```

function to run when event is triggered

final boolean parameter for backward compatibility

LET'S TAKE A CLOSER LOOK



ACTIVITY

EXERCISE

KEY OBJECTIVE

- Create DOM event handlers using vanilla JavaScript

TYPE OF EXERCISE

- Individual/Partner

TIMING

8 min `1-events-exercise`

1. Add event listeners to the 3 buttons at the top of the page.
2. Clicking each button should hide the block below it with the corresponding color.
3. Use handout/slides as a guide for syntax

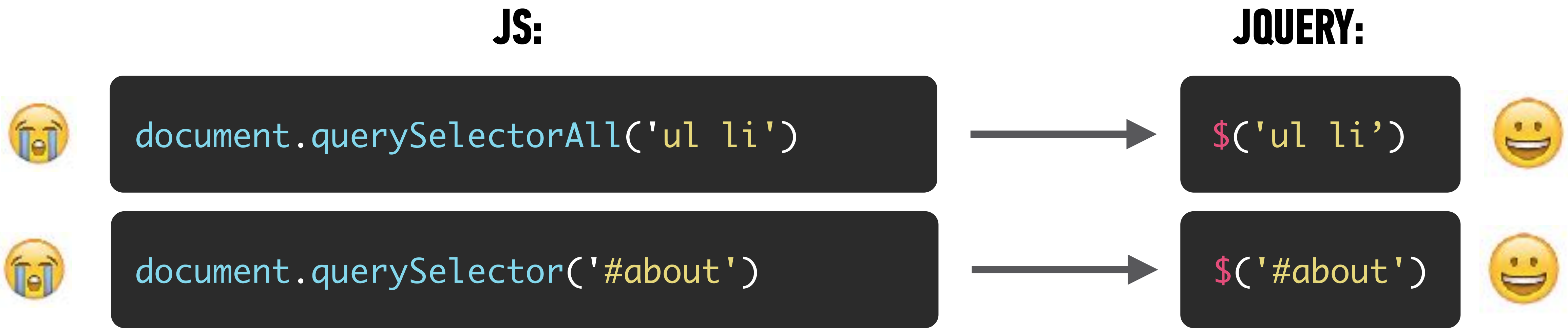
JQUERY

INTRO TO JQUERY — YOUR NEW BEST FRIEND!

jQuery is a JavaScript library you include in your pages.

JQUERY VS. JAVASCRIPT

jQuery allows us to keep using the CSS-style selectors that we know and love — but more concisely! Yay!



JS:

JQUERY:

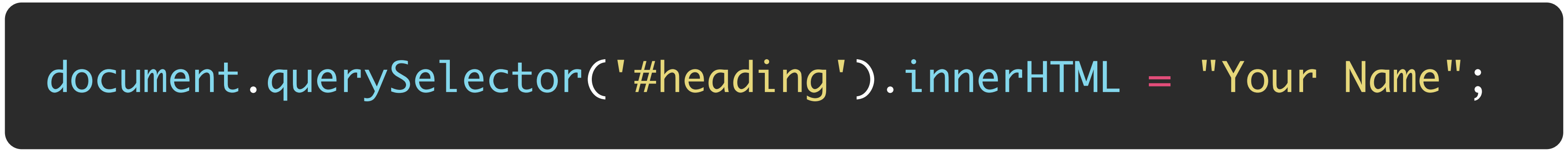
😭
```
document.querySelectorAll('ul li')
```
→
```
$('ul li’)
```
😃

😭
```
document.querySelector('#about')
```
→
```
$('#about')
```
😃

JQUERY VS. JAVASCRIPT

jQuery statements for DOM manipulation are also more concise!

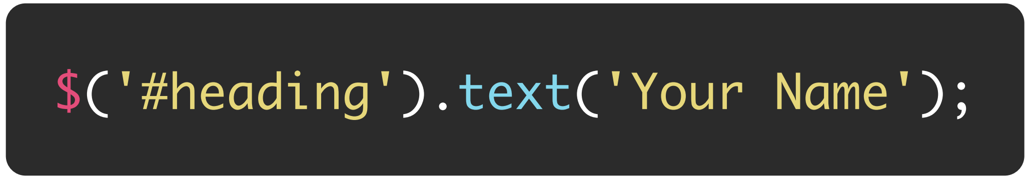
JS:



```
document.querySelector('#heading').innerHTML = "Your Name";
```

😭

JQUERY:



```
$('#heading').text('Your Name');
```

😀

You could do everything jQuery does with plain-old vanilla Javascript

JQUERY VS. JAVASCRIPT — A COMPARISON OF BENEFITS

JQUERY

- Write way less code to achieve the same tasks

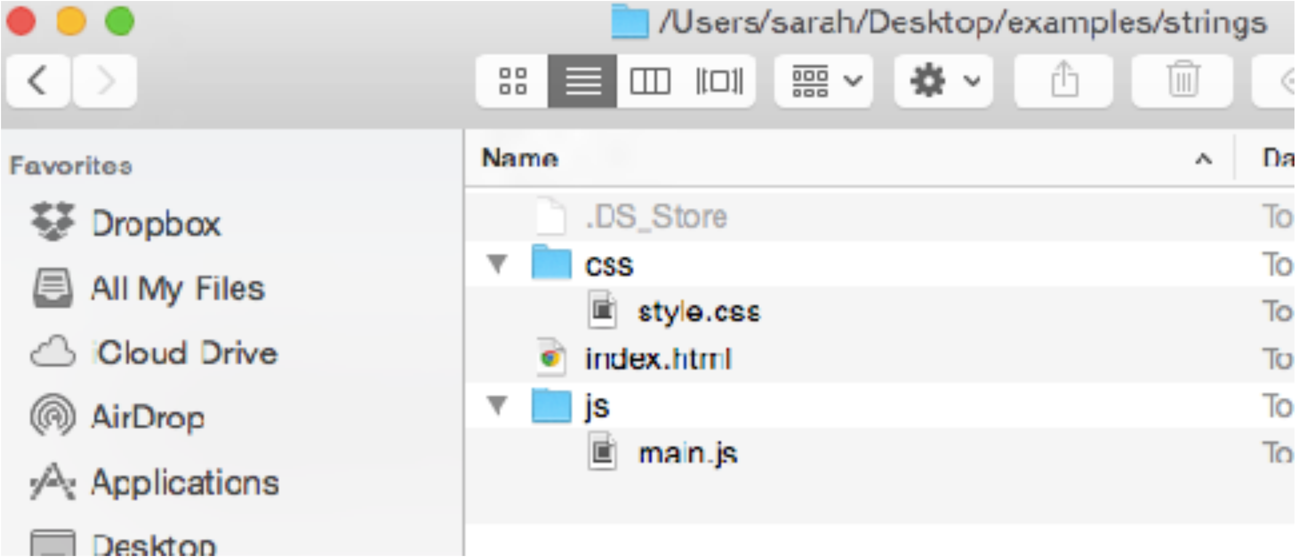
PURE JAVASCRIPT

- Better performance
- Faster

ADDING JQUERY TO YOUR PROJECT

KEEP IT ON THE UP AND UP!

- It is considered **best practice** to keep Javascript files organized in one folder.
- Usually people name this folder *scripts*, *js*, or *javascript*.

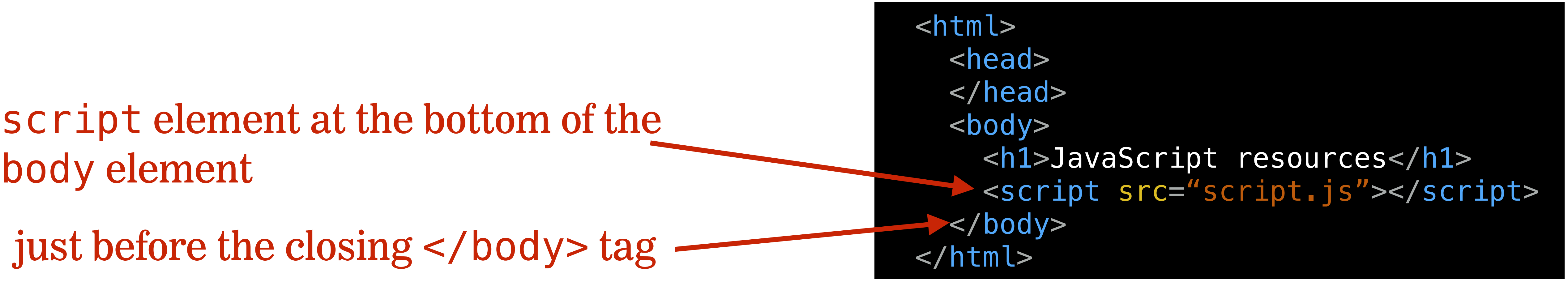


/Users/sarah/Desktop/examples/strings

```
.DS_Store
▼ css
    style.css
index.html
▼ js
    main.js
```

Remember - use an underscore or dash between words in folder names instead of a space. And try to avoid characters/symbols in file names (*really_cool_page.html* or *really-cool-page.html*).

REFERENCING A SCRIPT IN HTML



script element at the bottom of the body element

just before the closing </body> tag

```
<html>
  <head>
  </head>
  <body>
    <h1>JavaScript resources</h1>
    <script src=“script.js”></script>
  </body>
</html>
```

STEP 1: ADD JQUERY TO YOUR WEBSITE

1. Download the jQuery script (version 3.x, compressed).
2. Add a js folder to your project
3. Move the jQuery file you downloaded to the js folder
4. Use a `<script>` tag to include the jQuery file after your HTML content and before any other JavaScript files that use it.

```
<body>
  <!-- HTML content here -->
  <script src="js/jquery-3.2.1.min.js"></script>
  <script src="js/main.js"></script>
</body>
```

STEP 2: ADD A JAVASCRIPT FILE

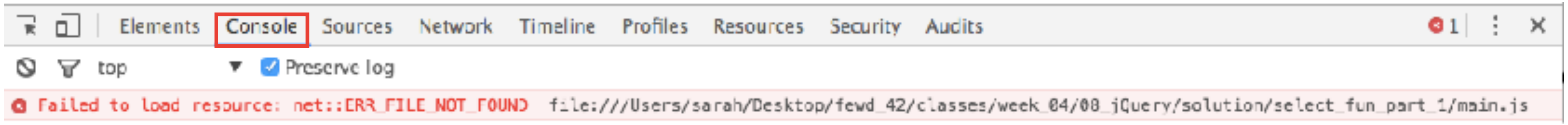
1. Create your custom JavaScript file with a .js extension (example: main.js)
2. Link to the JavaScript file from your HTML page using the `<script>` element. Add this **right before the closing `</body>` tag and after the `<script>` element for your jQuery file.**

```
<body>
  <!-- HTML content here -->
  <script src=“js/jquery-3.2.1.min.js"></script>
  <script src="js/main.js"></script>
</body>
```

ORDER IS IMPORTANT!!!!

MAKE SURE YOUR JS IS HOOKED UP PROPERLY

- Open the page in Chrome, then open the console (command + option + J [Mac] or Ctrl + Alt + J [Win]) and make sure there are no errors.



```
Failed to load resource: net::ERR_FILE_NOT_FOUND   file:///Users/sarah/Desktop/fewd_42/classes/week_04/08_jQuery/solution/select_fun_part_1/main.js
```

This error means the file can't be found. Check your url in your <script> tag. Make sure the file exists.

PART 1 — SELECT AN ELEMENT

A JQUERY STATEMENT INVOLVES 2 PARTS

1

Select an element/elements

2

Work with those elements

INTRO TO JQUERY

1

Select an element/elements

2

Work with those elements

JQUERY — SELECTING ELEMENTS

Selector

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

JQUERY OBJECTS — FINDING ELEMENTS: SOME EXAMPLES

	CSS	JQUERY
ELEMENT	`a { color: blue; }`	`$('a')`
ID	`#special { color: blue; }`	`$('#special')`
CLASS	`.info { color: blue; }`	`$('.info')`
NESTED SELECTOR	`div span { color: blue; }`	`$('div span')`

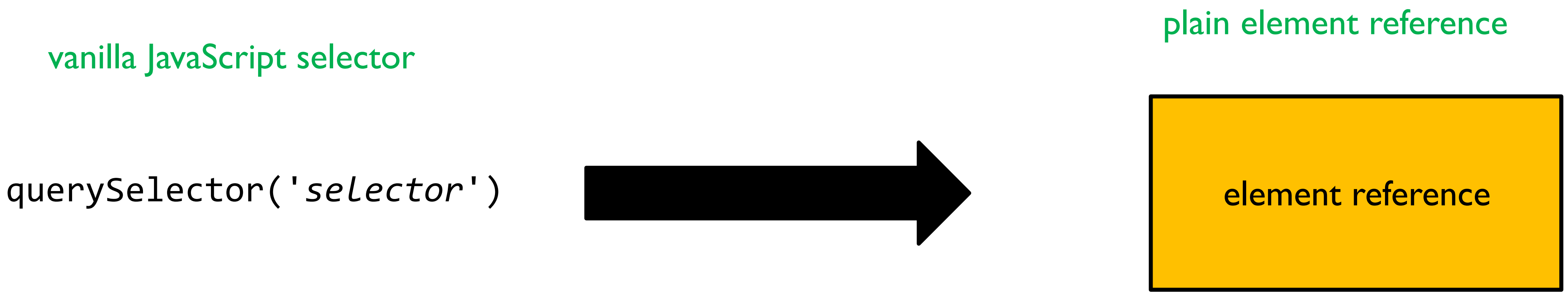
```html
<button id="form-submit">Submit</button>

<li class="circle">One</li>

<h1>Color Scheme Switcher</h1>
```

JQUERY OBJECTS

- Selecting elements with vanilla JavaScript returns an element reference (`querySelector`) or a collection of element references (`querySelectorAll`)



vanilla JavaScript selector

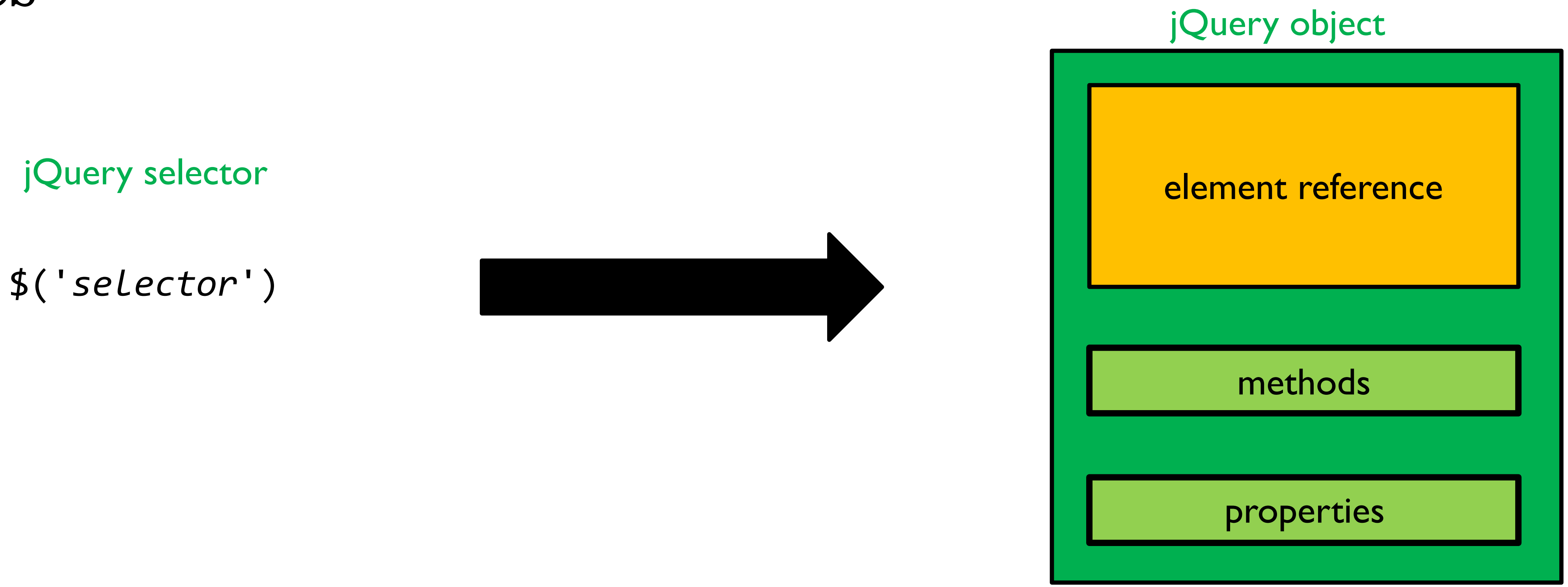
`querySelector('selector')`

plain element reference

element reference

JQUERY OBJECTS

- Selecting elements with jQuery returns a **jQuery object**, which is one or more element references packaged with jQuery methods and properties



NAMING VARIABLES WHEN USING JQUERY

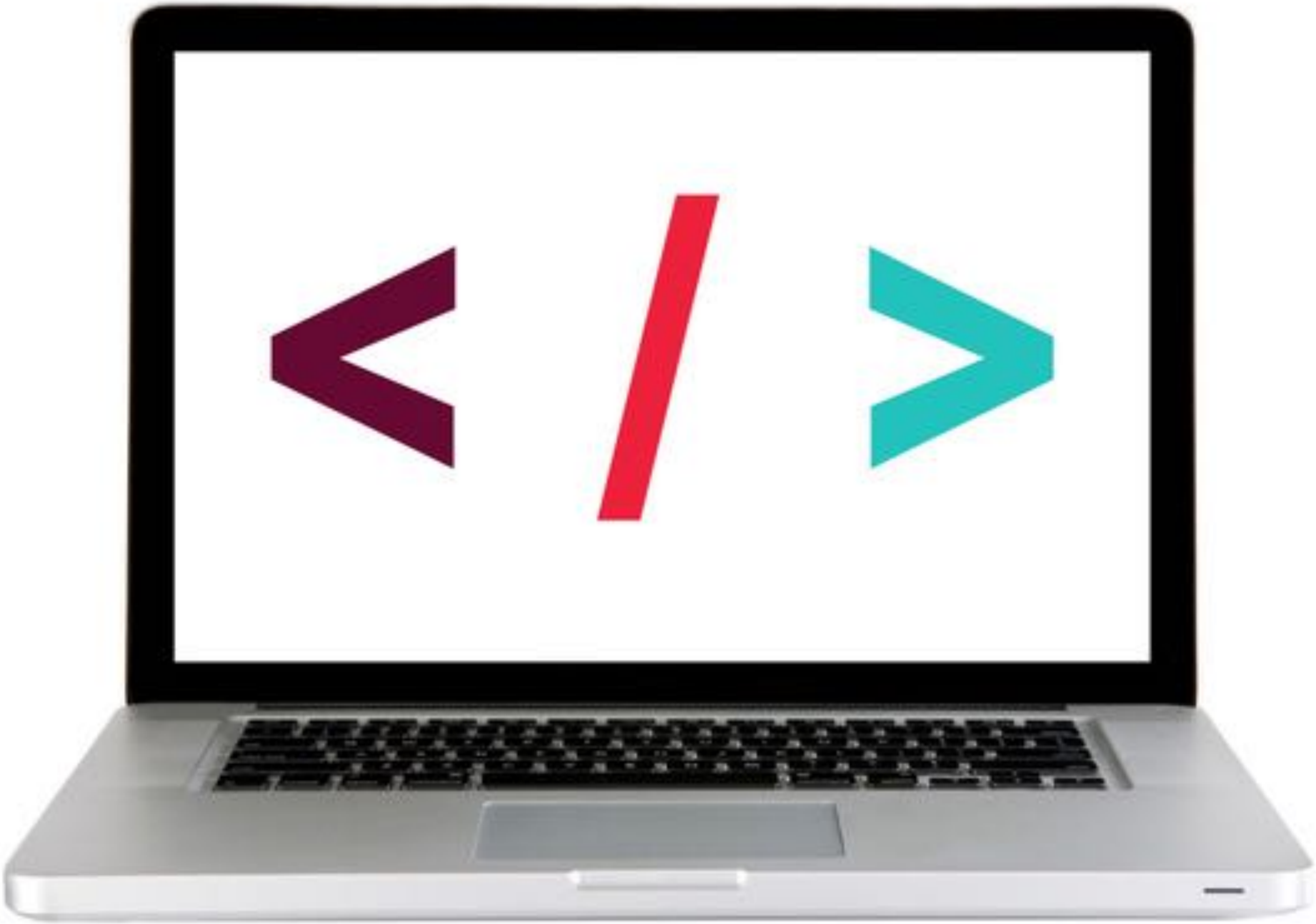
include $ at start of variable name to indicate that its value is a jQuery object

```
let $openTab = $(‘.open’);
```

it’s not an error to name the variable with out the $ — it just wouldn’t give us as much information

```
let openTab = $(‘.open’);
```

LET'S TAKE A CLOSER LOOK



PART 2 — ADD A METHOD

USING JQUERY TO MANIPULATE THE DOM

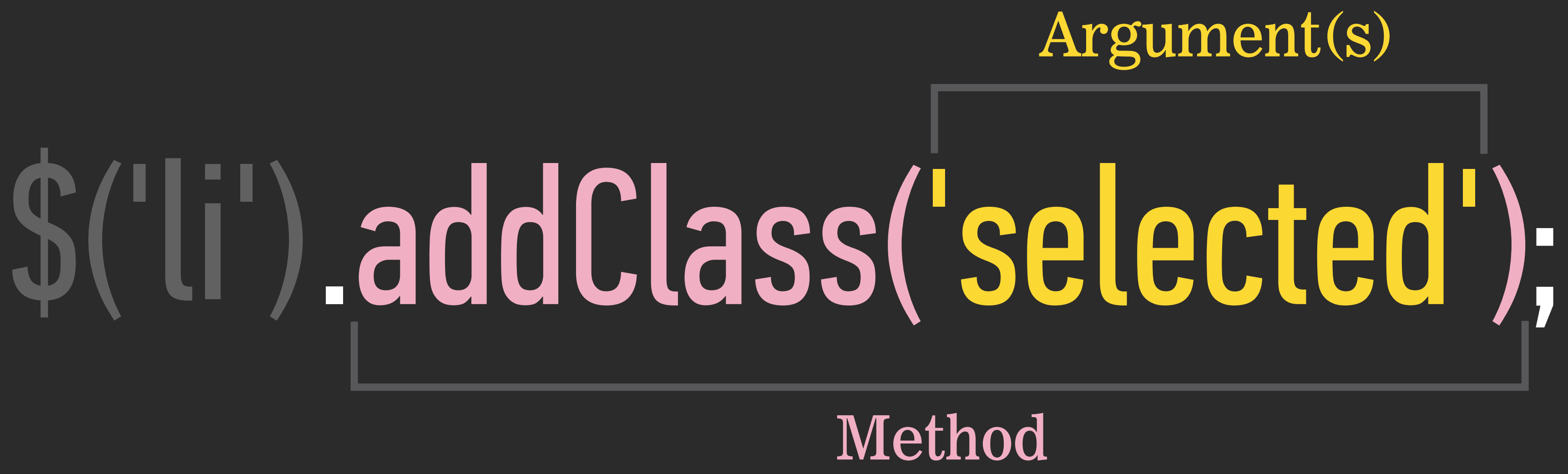
1

Select an element/elements

2

Work with those elements

JQUERY — WORKING WITH THOSE ELEMENTS



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

JQUERY METHODS

Be forewarned!

There are a lot of methods!

Do not feel like you need to sit down and memorize these. The important things is knowing that they're there and **being able to look them up** in the documentation.

api.jquery.com

JQUERY METHODS — WORKING WITH THOSE ELEMENTS

After we've selected elements, we can use jQuery methods to:

FIND ELEMENTS

GET/SET CONTENT

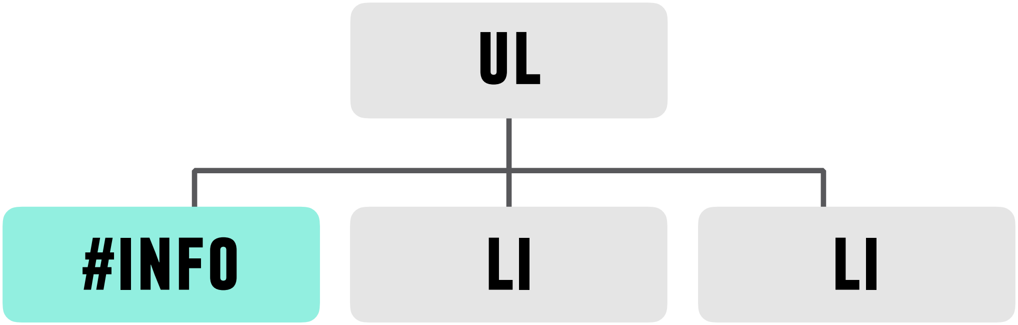
ADD EFFECTS/ ANIMATION

CREATE EVENT LISTENERS

See your handout or the jQuery docs for list!

TRAVERSING THE DOM?

```
$('#info').parent();
```



UL

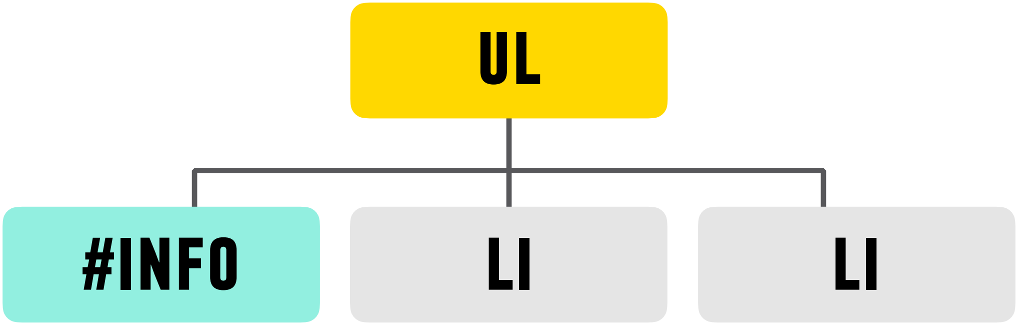
#INFO

LI

LI

TRAVERSING THE DOM?

```
$('#info').parent();
```



JQUERY METHODS — TRAVERSING THE DOM

FIND ELEMENTS

- Think of these as filters, or part of the selection process.
- They must come *directly after another selection*

METHODS	EXAMPLES
`.find()` *finds all descendants*	`$('h1').find('a');`
`.parent()`	`$('#box1').parent();`
`.siblings()`	`$('p').siblings('.important');`
`.children()`	`$('ul').children('li');`

What goes in the parentheses?
A **css-style selector**

JQUERY METHODS — WORKING WITH THOSE ELEMENTS

After we've selected elements, we can use jQuery methods to:

FIND ELEMENTS

GET/SET CONTENT

ADD EFFECTS/ ANIMATION

CREATE EVENT LISTENERS

See your handout or the [jQuery docs]() for list!

GETTING/SETTING CONTENT — PART 1

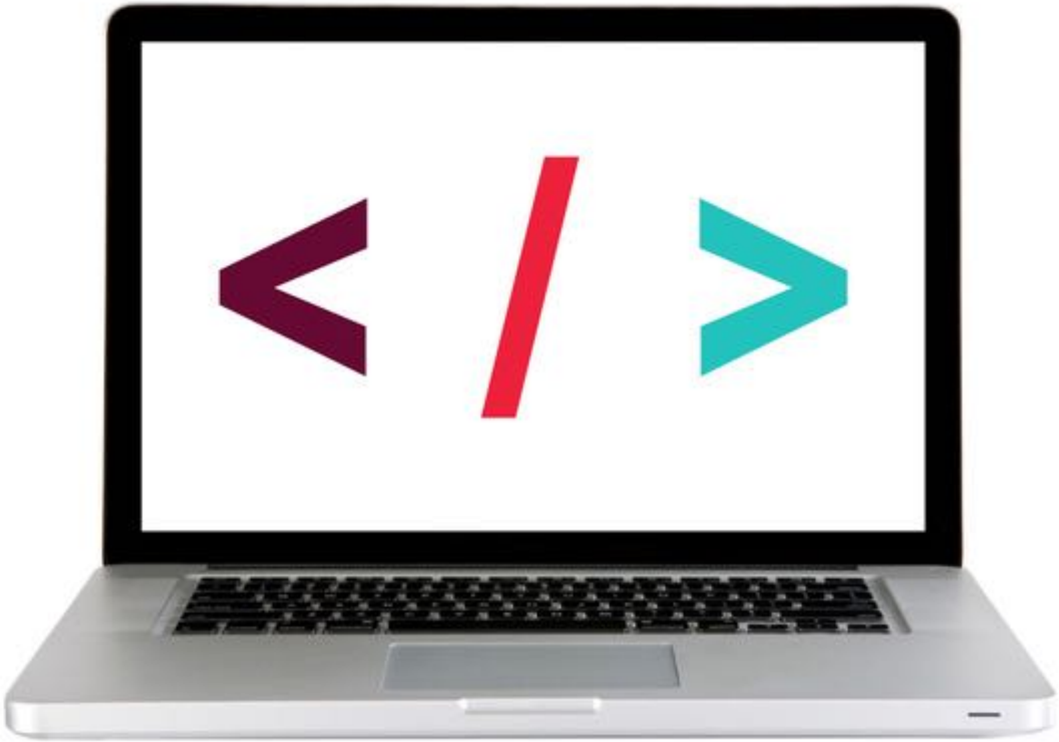
GET/SET CONTENT

Get/change content of elements and attributes

METHODS	EXAMPLES
.html()	$(‘h1').html('<strong>Content</strong>');
.text()	$(‘h1’).text('Just text content!’);
.attr()	$('img').attr('src', ‘images/bike.png');

What goes in the parentheses?
The **content** you want to change.

LET'S TAKE A CLOSER LOOK



GETTING/SETTING CONTENT — PART 2

GET/SET CONTENT

Get/change content of elements and attributes

METHODS	EXAMPLES
`.addClass()`	`$('p').addClass('success');`
`.removeClass()`	`$('p').removeClass('my-class-here');`
`.toggleClass()`	`$('p').toggleClass('special');`

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

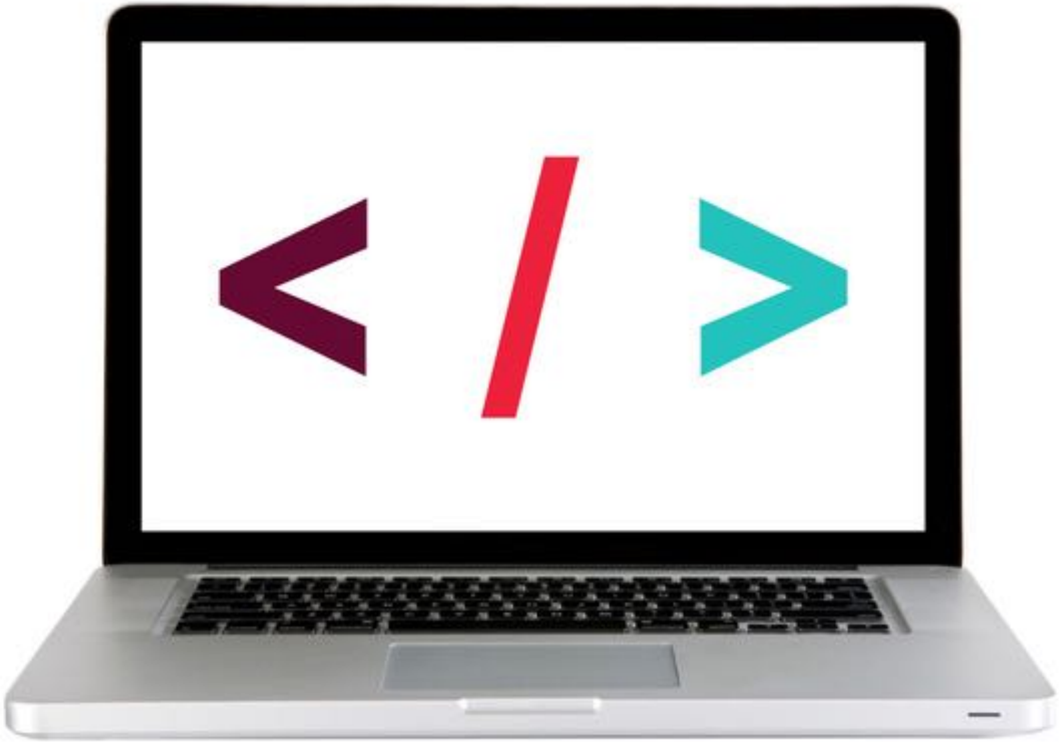
JQUERY METHODS — GETTING/SETTING CONTENT

GET/SET
CONTENT

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

NO PERIOD!!!

LET'S TAKE A CLOSER LOOK



ACTIVITY

EXERCISE

KEY OBJECTIVE

- Utilize jQuery to access and manipulate DOM elements.

TYPE OF EXERCISE

- Individual/Partner

TIMING

5 min `3-jquery-exercise`

1. Follow the instructions under part 1 in main.js
2. Use handout/slides as a guide for syntax

JQUERY METHODS — WORKING WITH THOSE ELEMENTS

After we've selected elements, we can use jQuery methods to:

FIND ELEMENTS

GET/SET CONTENT

ADD EFFECTS/ ANIMATION

CREATE EVENT LISTENERS

See your handout or the [jQuery docs](#) for list!

JQUERY METHODS — EFFECTS/ANIMATION

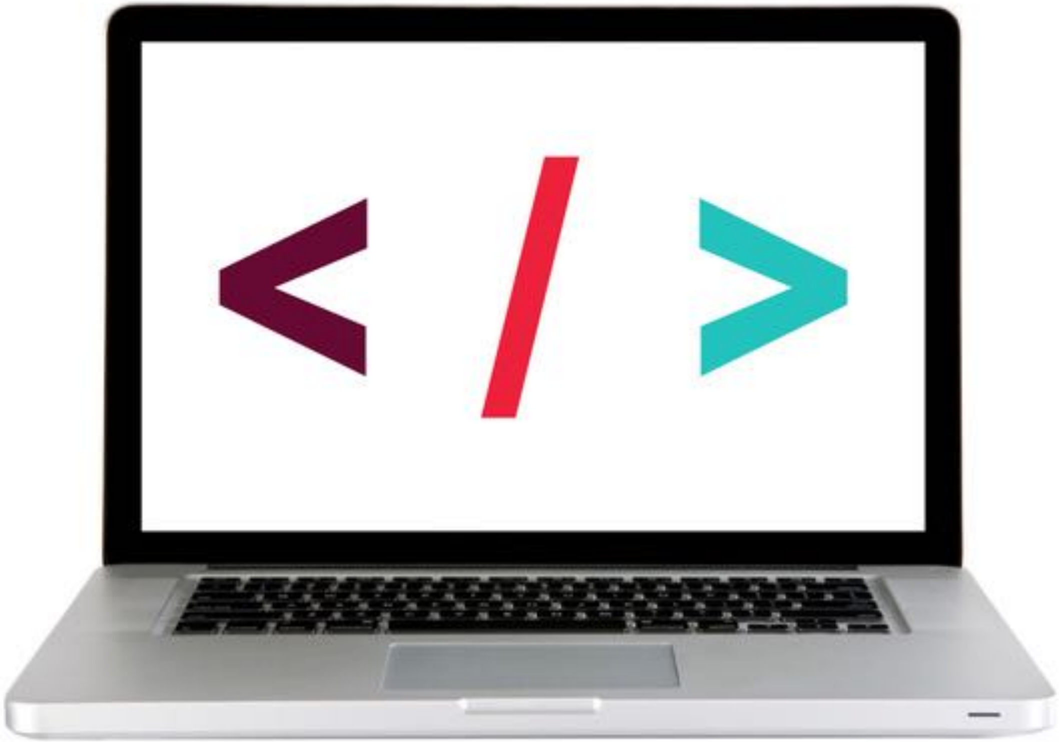
ADD EFFECTS/ ANIMATION

Add effects and animation to parts of the page

METHODS	EXAMPLES
`.show()`	`$('h1').show();`
`.hide()`	`$('ul').hide();`
`.fadeIn()`	`$('h1').fadeIn(300);`
`.fadeOut()`	`$('.special').fadeOut('fast');`
`.slideUp()`	`$('div').slideUp();`
`.slideDown()`	`$('#box1').slideDown('slow');`
`.slideToggle()`	`$('p').slideToggle(300);`

What goes in the parenthesis?
An animation speed

LET'S TAKE A CLOSER LOOK



JQUERY METHODS — WORKING WITH THOSE ELEMENTS

After we've selected elements, we can use jQuery methods to:

FIND ELEMENTS

GET/SET CONTENT

ADD EFFECTS/ ANIMATION

CREATE EVENT LISTENERS

See your handout or the jQuery docs for list!

EVENTS

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	KEYBOARD	FORM	DOCUMENT
click	keypress	submit	resize
dblclick	keydown	change	scroll
mouseenter	keyup	focus	
mouseleave		blur	

```
$('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

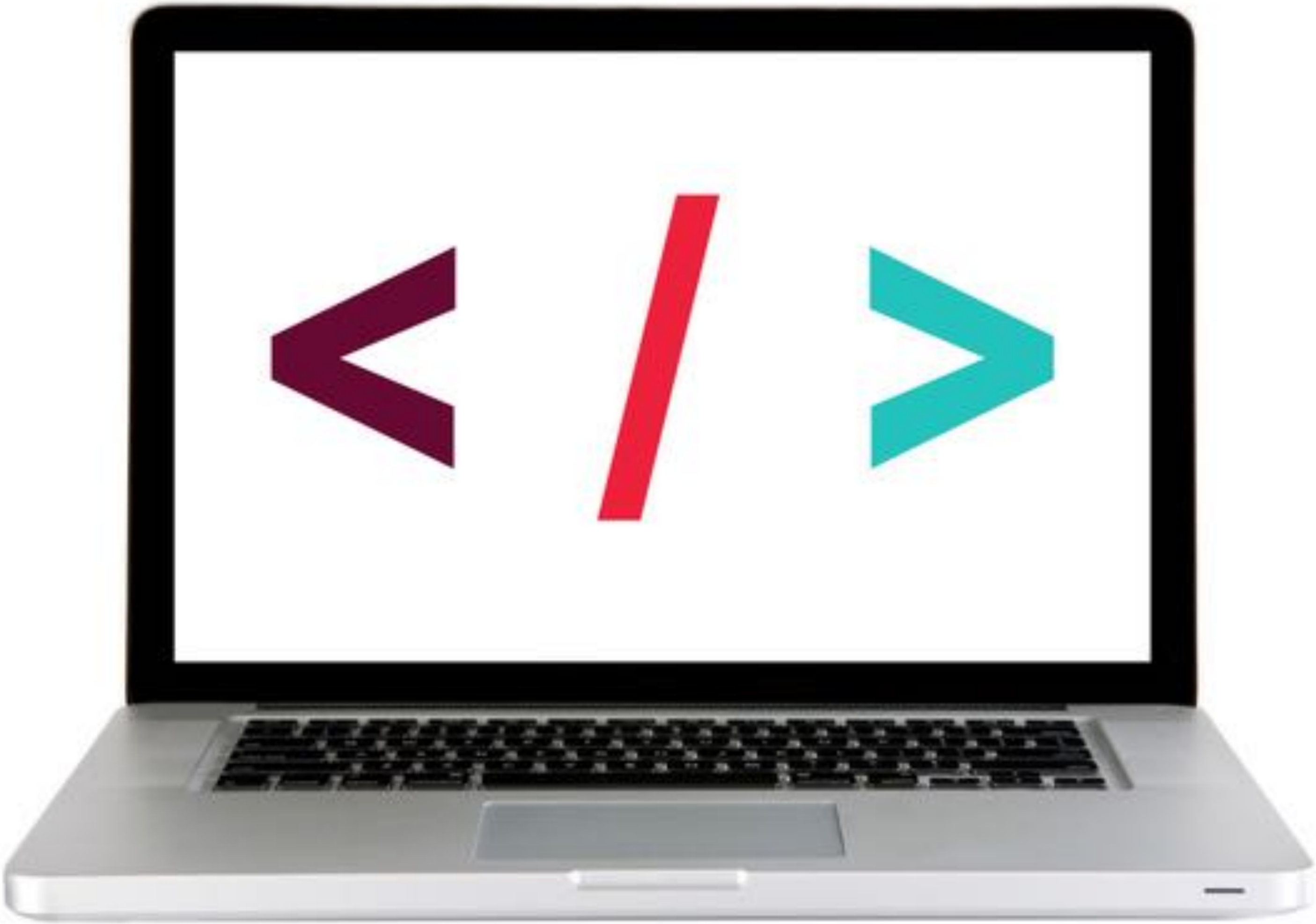
method for all events

type of event

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

function to run when event is triggered

LET'S TAKE A LOOK



ACTIVITY

EXERCISE

KEY OBJECTIVE

- Utilize jQuery to access and manipulate DOM elements.

TYPE OF EXERCISE

- Individual/Partner

TIMING

5 min

Continue with `3-jquery-exercise`

1. Follow the instructions under Part 2 in main.js
2. Use handout/slides as a guide for syntax

ACTIVITY

EXERCISE

KEY OBJECTIVE

- Create DOM event handlers to respond to user actions

TYPE OF EXERCISE

- Individual/Partner

AS A CLASS

6 min Return to `1-events-exercise` folder

1. Rewrite your vanilla JavaScript code to use jQuery instead.
2. Use handout/slides as a guide for syntax

CREATING & APPENDING DOM NODES

document.ready()

- specifies code to run only after the DOM has finished loading
- Syntax:

```
$(document).ready(function() {
  // code goes here
});
```

- Shorthand version (best practice):

```
$(function() {
  // code goes here
});
```

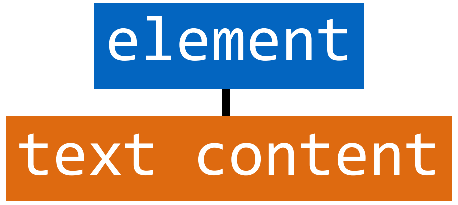
Adding content to the DOM

1. create a new element with *`$(‘<element>’)`*

element

Adding content to the DOM

1. create a new element with `$(‘<element>’)`
2. add new content to that element with `.text()` or `.html()`

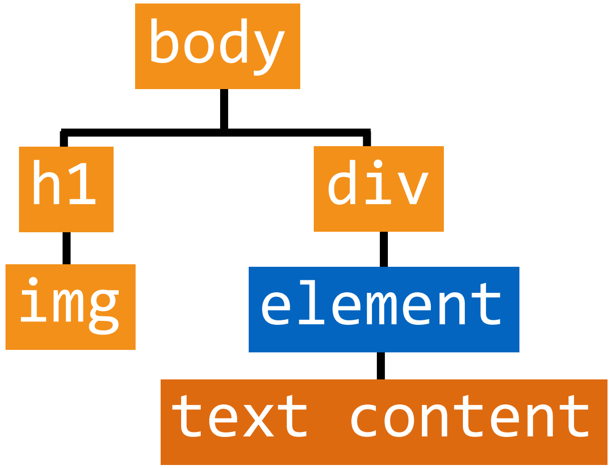


element

text content

Adding content to the DOM

1. create a new element with `$(‘<element>’)`
2. add new content to that element with `.text()` or `.html()`
3. attach the new element to the DOM with `.append()`



$(‘<element>’)

- Creates a new element

```
$(‘<li>’); // creates an li element
```

- Created element isn’t attached to DOM
  - » assign variable when creating so you can reference later

```
let item1 = $(‘<li>’);
let item2 = $(‘<li>’);
```

`.text()` or `.html()`

- Creates and adds text content as the child of an element
- Easiest to add method to same statement that creates element

```
let item1 = $(‘<li>’).text(‘banana’);
let item2 = $(‘<li>’).text(‘apple’);
```

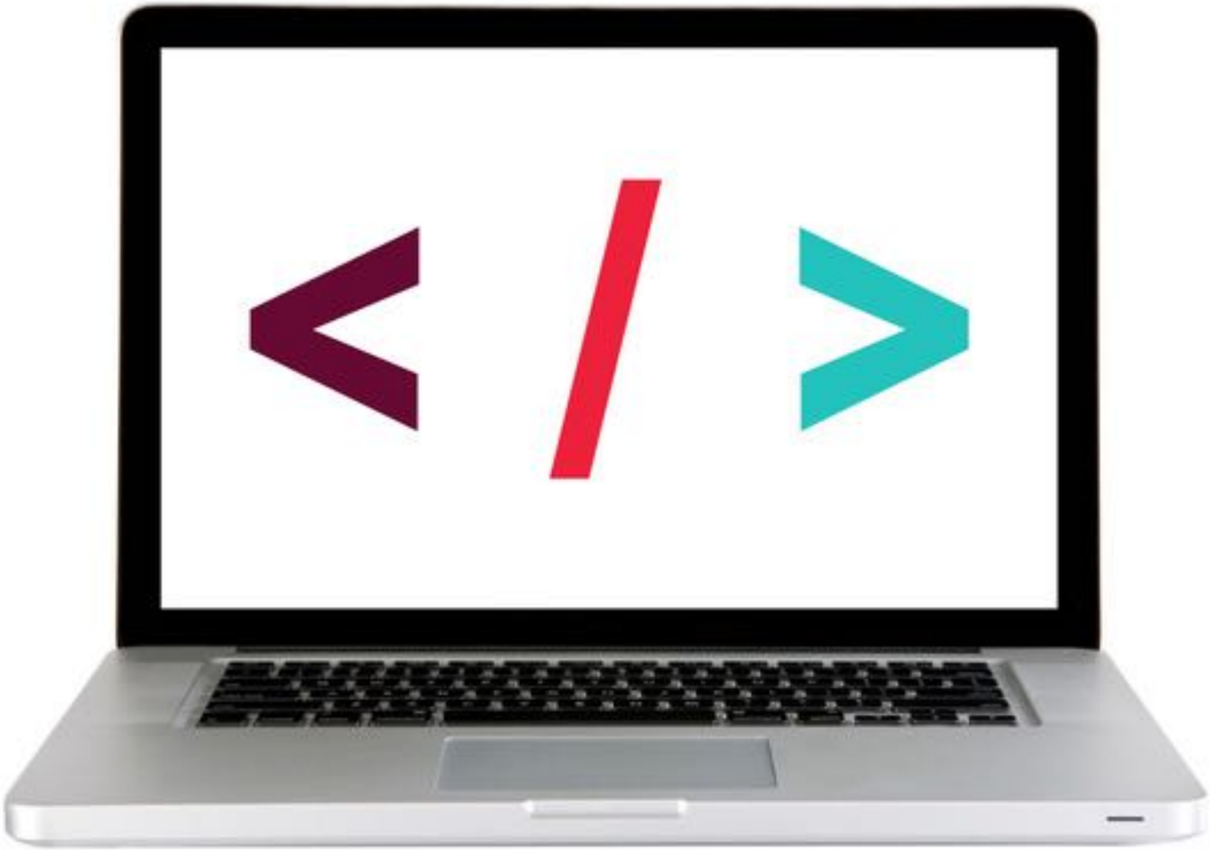
```
let item1 = $(‘<li>’).html(‘<strong>Every</strong> dinosaur’);
let item2 = $(‘<li>’).html(‘Books (<em>not</em> ebooks)’);
```

.append()

- Attaches element or node as child of specified element
  - Attaching to a DOM element makes it part of the DOM
- Syntax:
  $(*parent*).append(*child*);

```
const list = $(‘ul’);  // selects ul element
list.append(item1);    // adds item1 li to list ul
list.append(item2);    // adds item2 li to list ul
```

EVENTS & JQUERY



LET'S TAKE A CLOSER LOOK

EXERCISE - ADD CONTENT TO A WEB PAGE USING JQUERY

EXERCISE

LOCATION

- starter-code > 6-create-append—exercise

TIMING

10 min

1. Open preview.png. Your task is to use DOM manipulation to build the sidebar shown in the image and add it to the blog.html web page.
2. Open app.js in your editor, then follow the instructions to create and the “About us” heading and the 2 paragraphs of text to the sidebar.
3. BONUS 1: Open preview-bonus.png, then write JavaScript code to add the image shown to the sidebar. (Filename and location in app.js.)
4. BONUS 2: Create and append the “Recent issues” heading and list.

WORKING WITH EVENT OBJECTS

preventDefault()

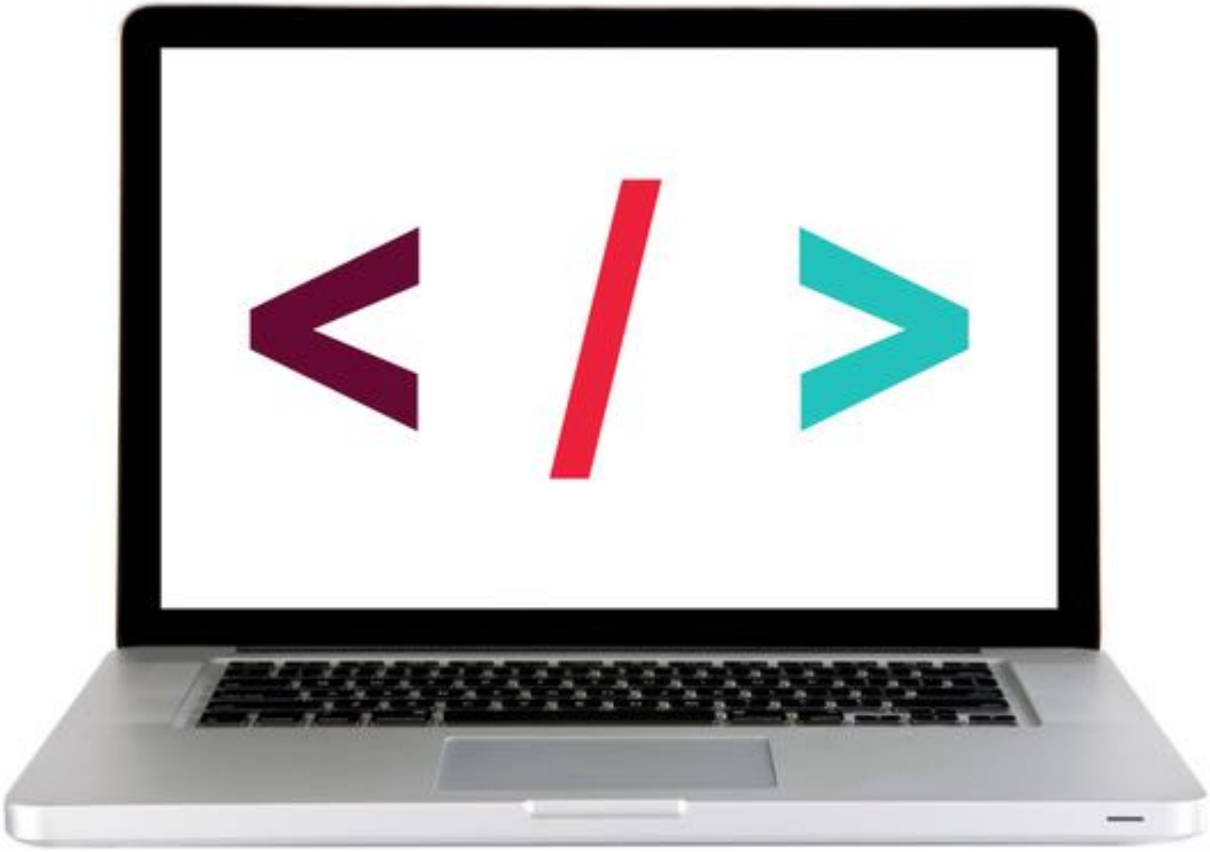
- Prevents element from executing default behavior in response to an event

Referencing an event

- An object containing information about the triggering event is passed to a function called in response to an event
- Specify a parameter to be able to reference this event in your code
  - By convention, we use `event`, `evt`, or `e`

```
submitButton.on(‘click’, function(event) {
  event.preventDefault();
  ...
)};
```

EVENTS & JQUERY



LET'S TAKE A CLOSER LOOK

EXERCISE

EXERCISE

LOCATION

▸ `starter-code > 8-event object—exercise`

TIMING

2 min

1. Update the code to prevent the form from submitting when the button is clicked.
2. Test your code in the browser and check the URL to verify that the form is not being submitted.

EXERCISE

EXERCISE

OBJECTIVE

- Manipulate the DOM and create DOM event handlers using jQuery

LOCATION

- `starter-code > 9-dice-lab`

TIMING

until 9:20

1. Use jQuery to create a page where every time the user hits the "Roll Dice" button, the screen randomly updates the two dice.
2. BONUS: Refactor your code to replace all jQuery with vanilla JavaScript.

Exit Tickets!

(Class #7)

LEARNING OBJECTIVES - REVIEW

- Select DOM elements and properties using jQuery.
- Manipulate the DOM by using jQuery selectors and functions.
- Create DOM event handlers using jQuery.

NEXT CLASS PREVIEW

Advanced jQuery

- Use event delegation to manage dynamic content.
- Use implicit iteration to update elements of a jQuery selection

Q&A