

JAVASCRIPT DEVELOPMENT

Sasha Vodnik, Instructor

HELLO!

1. Pull changes from the `svodnik/JS-SF-14-resources` repo to your computer
2. Open the `07-events-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

INTRO TO 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



TYPE OF EXERCISE

- ▶ Groups of 3

TIMING

6 min

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

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.

VANILLA JAVASCRIPT DOM MANIPULATION METHODS

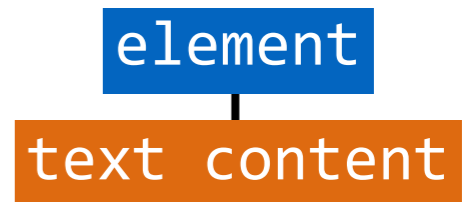
Adding content to the DOM

1. create a new element with `document.createElement()`

element

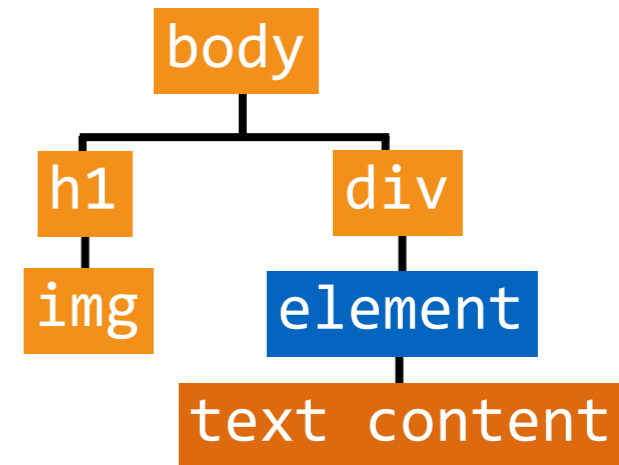
Adding content to the DOM

1. create a new element with `document.createElement()`
2. add text content to that element with the `innerHTML` or `textContent` property



Adding content to the DOM

1. create a new element with `document.createElement()`
2. add text content to that element with the `innerHTML` or `textContent` property
3. **attach the new element to the DOM with `appendChild()`**



createElement()

- Creates a new element

```
document.createElement('li'); // creates an li element
```

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

```
let item1 = document.createElement('li');  
let item2 = document.createElement('li');
```

textContent

- Specifies text content of an element
- Any nested HTML tags are parsed as text, not HTML

```
item1.innerHTML = 'banana';  
item2.innerHTML = 'apple';
```

innerHTML

- Specifies text content of an element
- May include nested HTML tags

```
item1.innerHTML = '<strong>banana</strong>';  
item2.innerHTML = '<em>apple</em>';
```

appendChild()

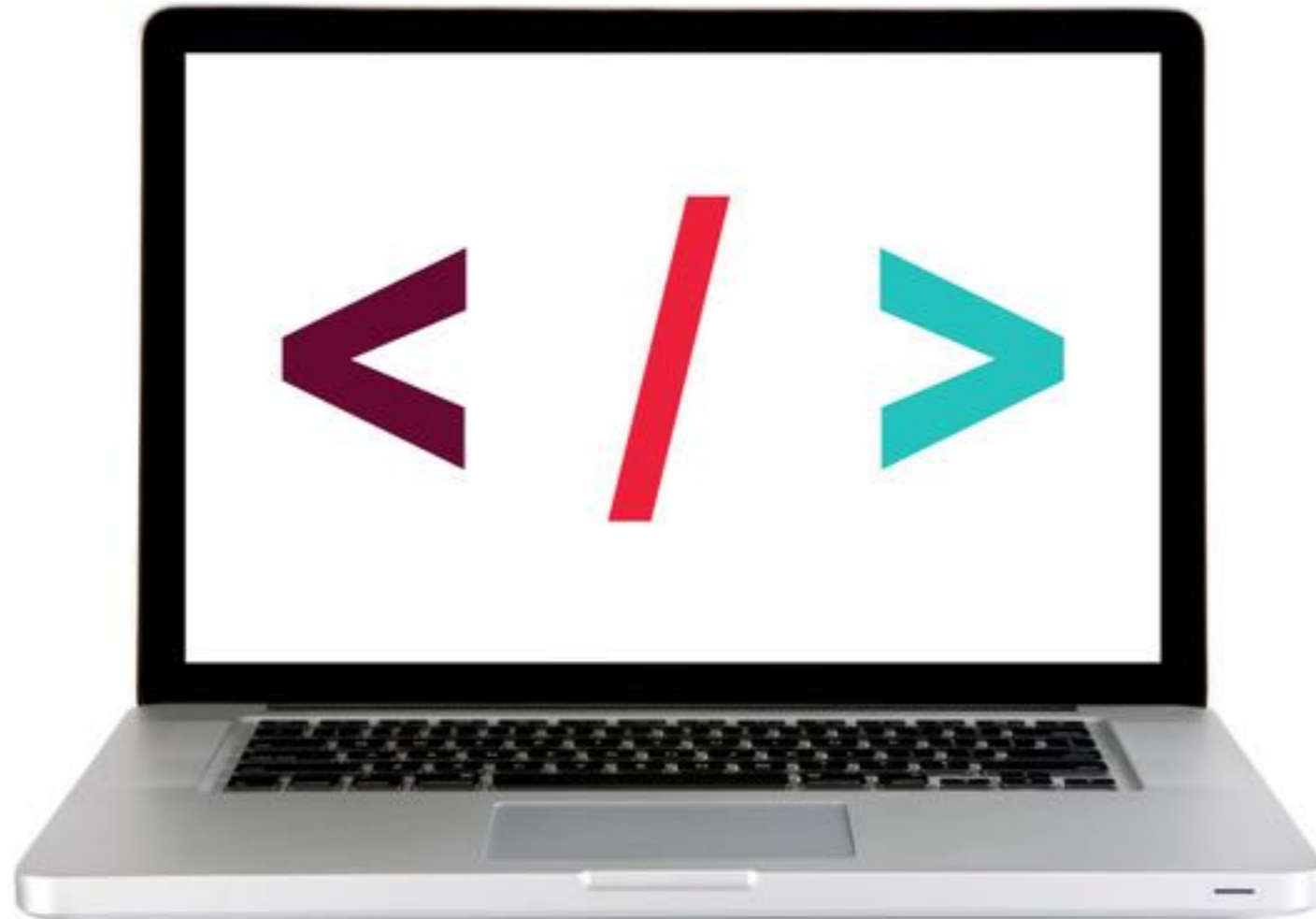
- Attaches element or node as child of specified element
 - » Attaching to a DOM element makes it part of the DOM

- Syntax:

```
parent.appendChild(child);
```

```
let list = document.querySelector('ul'); // selects ul element
list.appendChild(item1); // adds item1 li to list ul
list.appendChild(item2); // adds item2 li to list ul
```

LET'S TAKE A LOOK



EXERCISE



EXERCISE

KEY OBJECTIVE

- ▶ Explain and use JavaScript methods and properties for DOM manipulation.

TYPE OF EXERCISE

- ▶ Pairs

TIMING

2 min

1. Work together to create and complete a list of the three steps in DOM manipulation.
2. For each step in your list, add the method/property used.

EXERCISE – ADD CONTENT TO A WEB PAGE USING JAVASCRIPT



EXERCISE

LOCATION

▶ starter-code > 1-create-append-exercise

TIMING

20 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.

EVENTS

EVENT LISTENERS

selecting element

```
let button = document.querySelector('.submitBtn');
```

element
reference

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

EVENT LISTENERS

```
let button = document.querySelector('.submitBtn');
```

method to add event listener

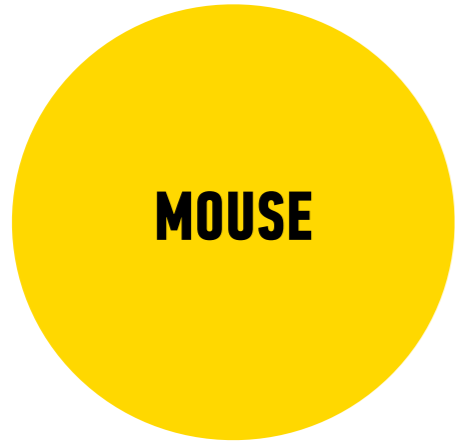
```
button.addEventListener('click', function() {  
  // your code here  
});
```

EVENT LISTENERS

```
let button = document.querySelector('.submitBtn');
```

type of event

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



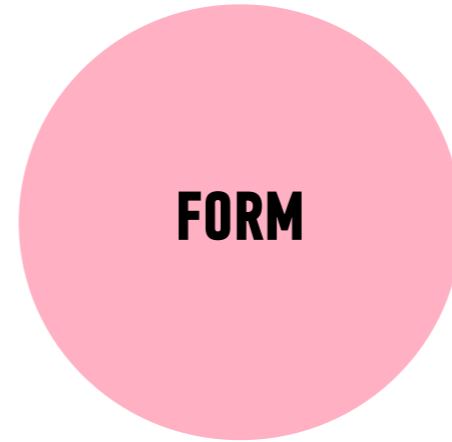
MOUSE

click
dblclick
mouseenter
mouseleave



KEYBOARD

keypress
keydown
keyup



FORM

submit
change
focus
blur



DOCUMENT

resize
scroll



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


EVENT LISTENERS

```
let button = document.querySelector('.submitBtn');
```

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

function to run
when event is
triggered

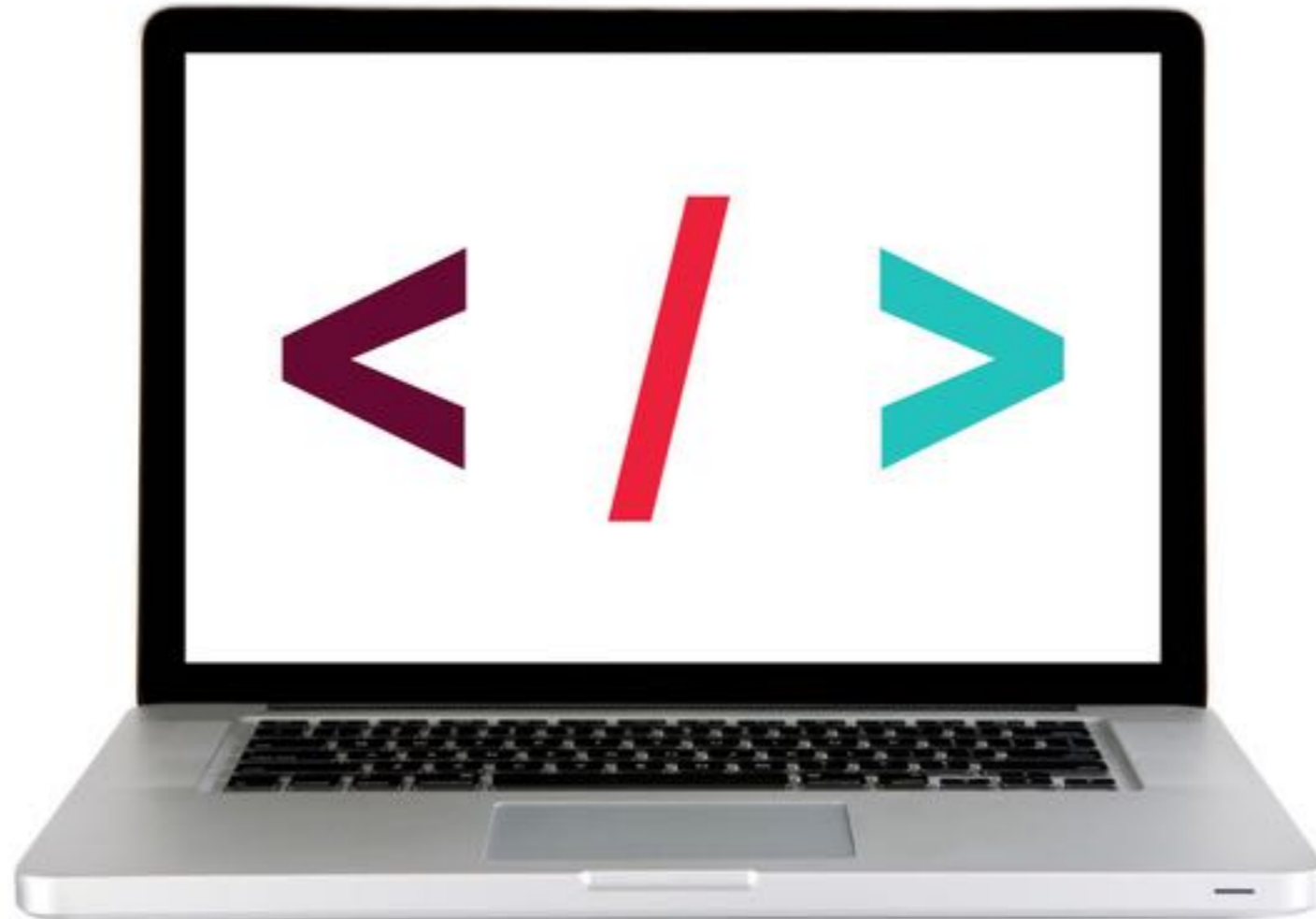
EVENT LISTENERS

element reference method to add event listener type of event

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

function to run when event is triggered

LET'S TAKE A CLOSER LOOK



ACTIVITY



KEY OBJECTIVE

- ▶ Create DOM event handlers using vanilla JavaScript

TYPE OF EXERCISE

- ▶ Individual/Partner

TIMING

8 min

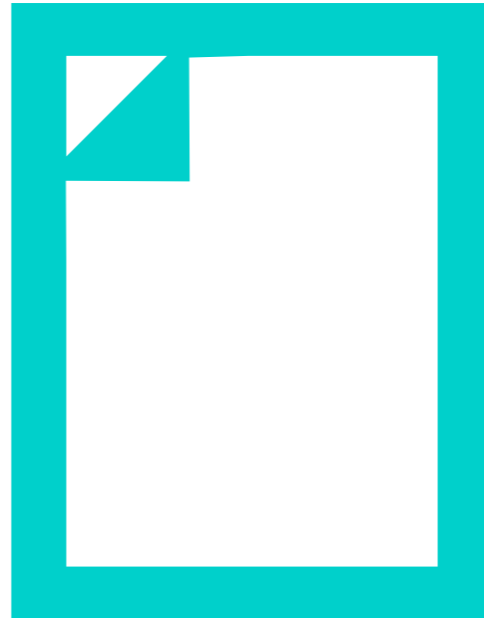
3-events-exercise

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:



```
document.querySelectorAll('ul li')
```



```
document.querySelector('#about')
```



JQUERY:

```
$('#ul li')
```



```
$('#about')
```



JQUERY VS. JAVASCRIPT

jQuery statements for DOM manipulation are also more concise!

JS:

```
document.querySelector('#heading').textContent = "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

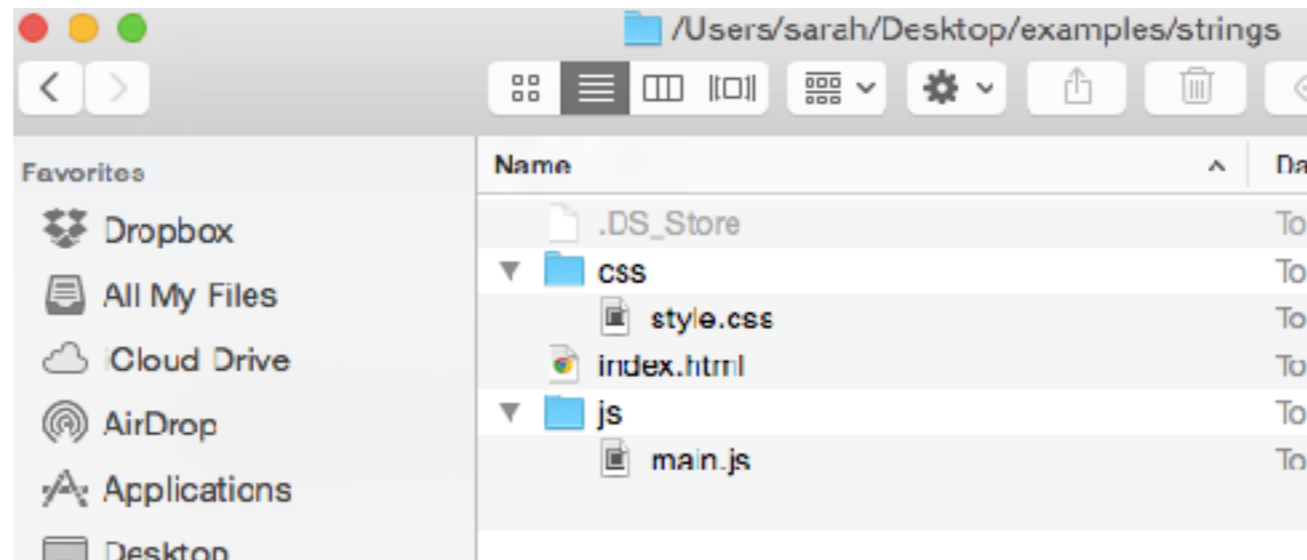
- ▶ Better performance
- ▶ Faster

JQUERY

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*.



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.**

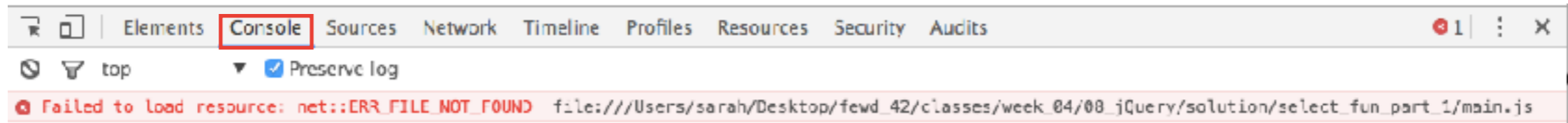
```
<body>  
  <!-- HTML content here -->  
  <script src="js/jquery-3.2.1.min.js"></script>  
  <script src="js/main.js"></script>  
</body>
```



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.



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

JQUERY

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	<code>a { color: blue; }</code>	<code>\$('a')</code>
ID	<code>#special { color: blue; }</code>	<code>\$('#special')</code>
CLASS	<code>.info { color: blue; }</code>	<code>\$('.info')</code>
NESTED SELECTOR	<code>div span { color: blue; }</code>	<code>\$('div span')</code>

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

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

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

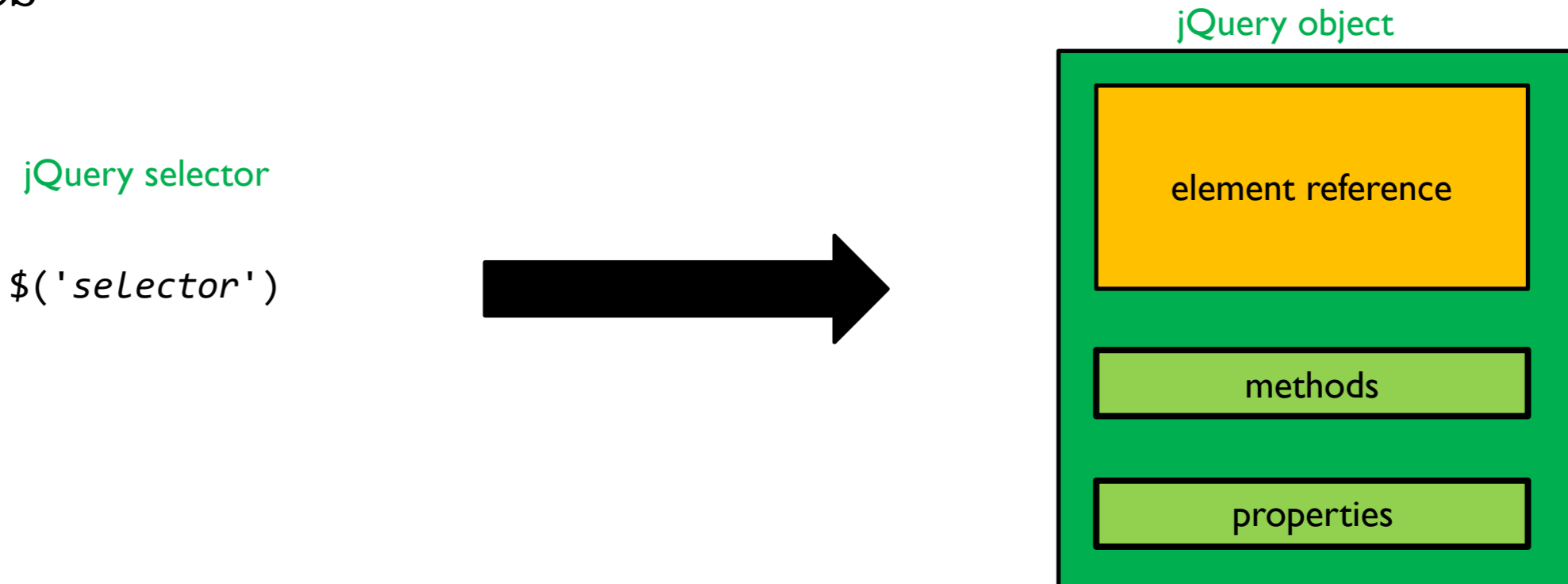
JQUERY OBJECTS

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



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

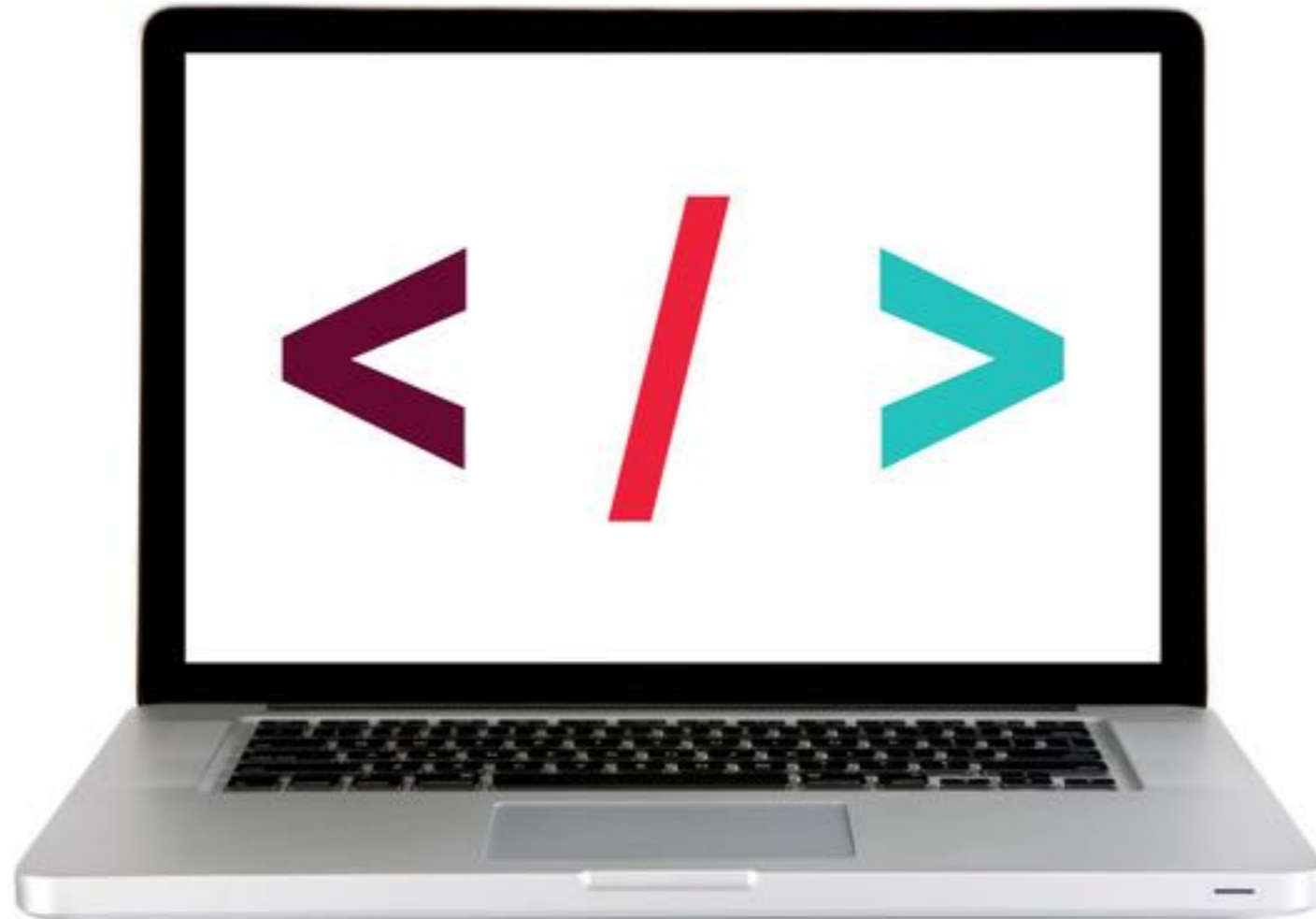
```
let $openTab = $('.open');
```



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



JQUERY

PART 2 — ADD A METHOD

USING JQUERY TO MANIPULATE THE DOM

1

Select an element/elements

2

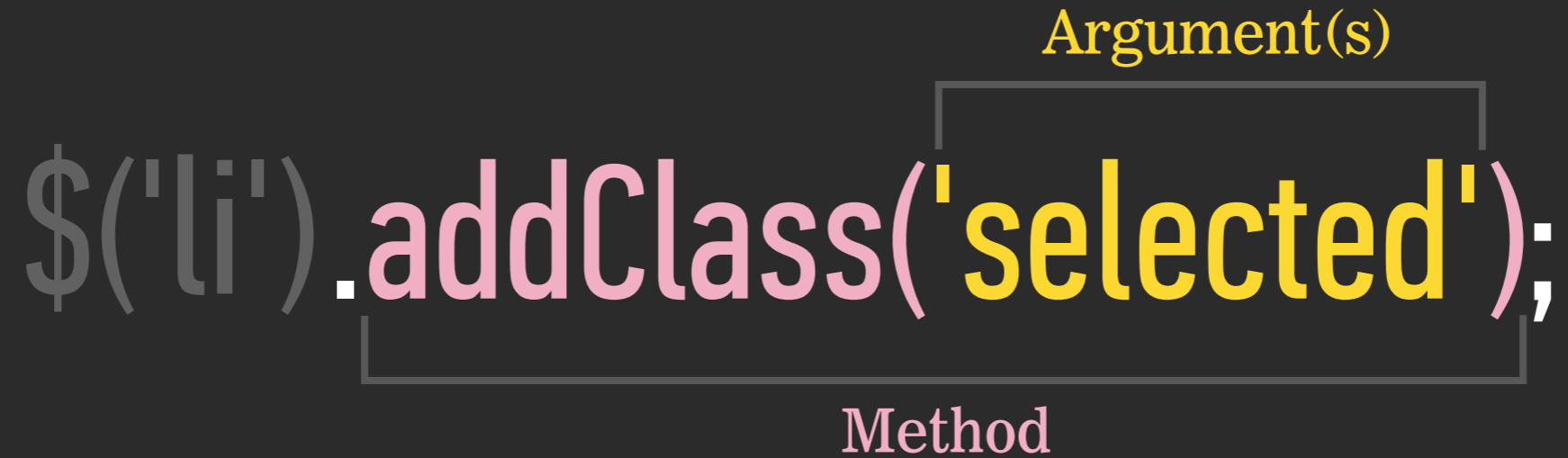
Work with those elements

JQUERY — WORKING WITH THOSE ELEMENTS

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

Argument(s)

Method

The image shows the jQuery code snippet `$('.li').addClass('selected');` with two annotations. A bracket above the string `'selected'` is labeled "Argument(s)" in yellow text. A bracket below the `addClass` method name is labeled "Method" in pink text. The `addClass` text is also highlighted in pink, and the string `'selected'` is highlighted in yellow.

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**

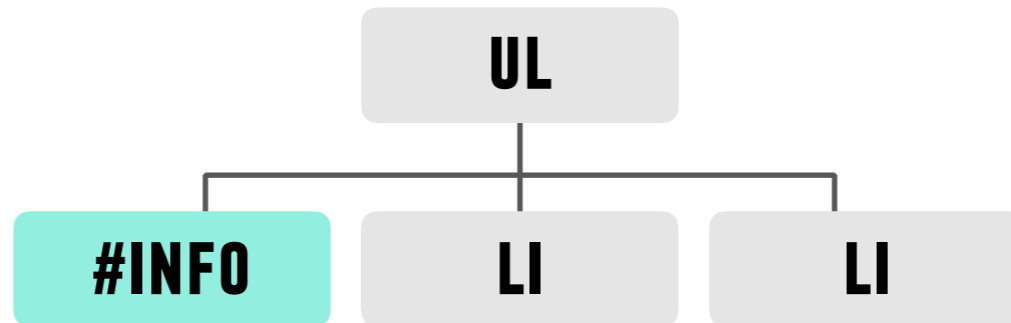
**CREATE
EVENT
LISTENERS**



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

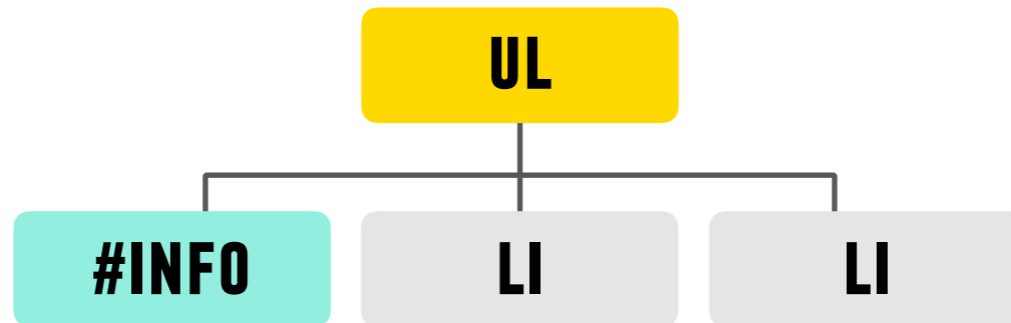
TRaversing the DOM?

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



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
<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 — 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**



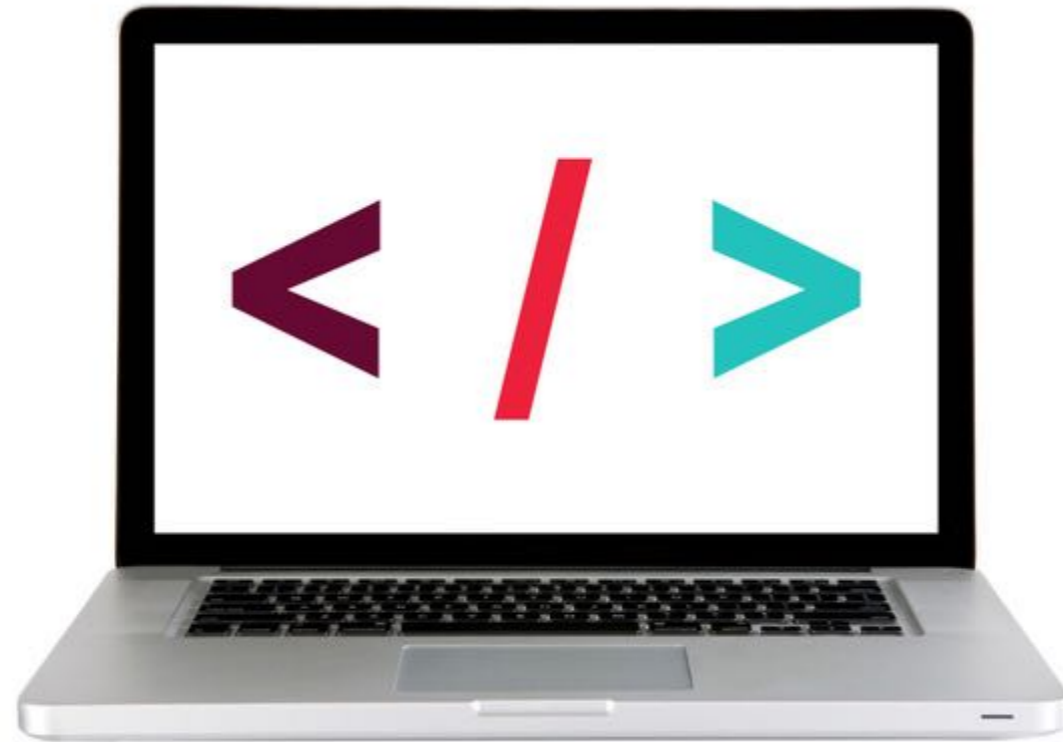
See your handout or the [jQuery docs](#) for list!

Get/change content of elements and attributes

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

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

LET'S TAKE A CLOSER LOOK



Get/change content of elements and attributes

METHODS	EXAMPLES
<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>

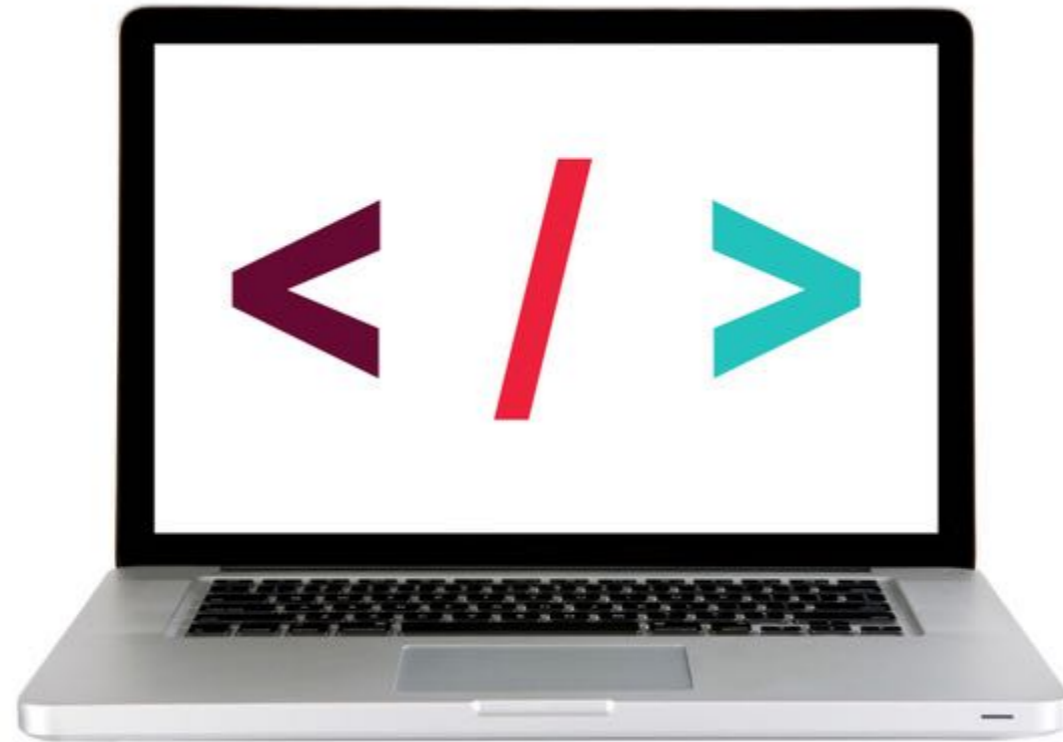
What goes in the parentheses?
The **classes** you want to change.

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



NO PERIOD!!!

LET'S TAKE A CLOSER LOOK



ACTIVITY



KEY OBJECTIVE

- ▶ Utilize jQuery to access and manipulate DOM elements.

TYPE OF EXERCISE

- ▶ Individual/Partner

TIMING

5 min

5-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

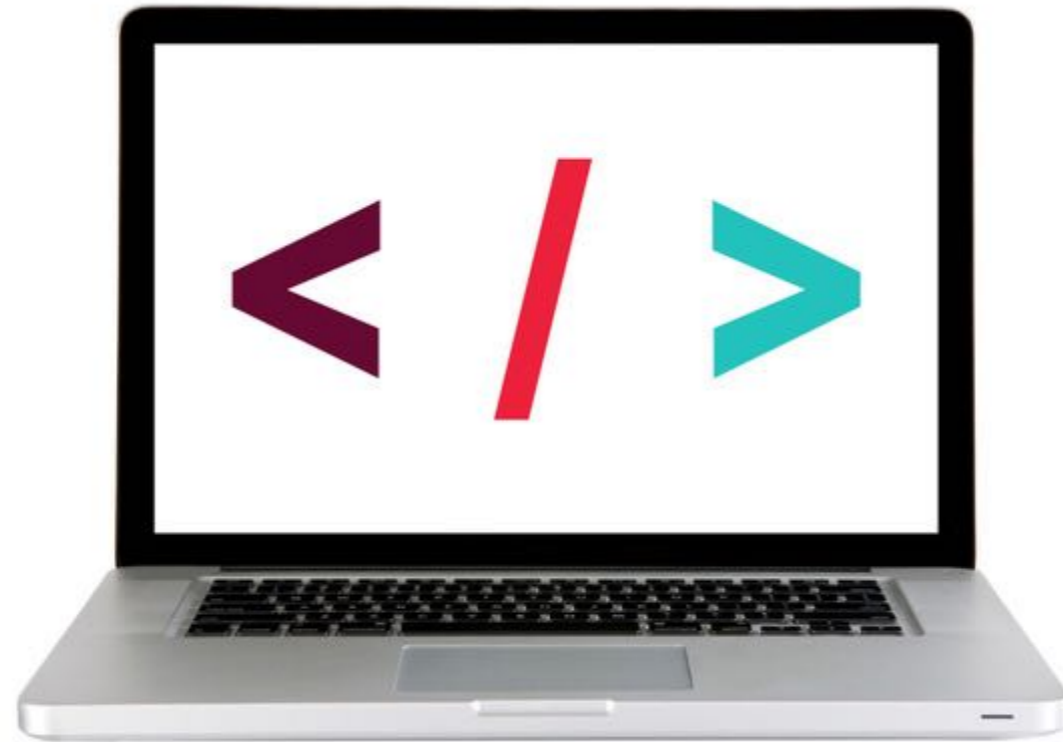
**ADD
EFFECTS/
ANIMATION**

Add effects and animation to parts of the page

METHODS	EXAMPLES
<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()</code>	<code>\$('#p').slideToggle(300);</code>

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  
});
```

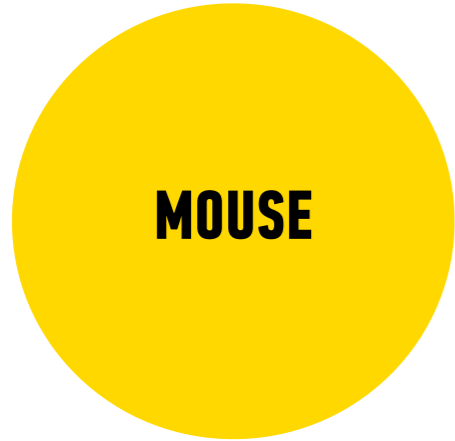
method for all events

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

JQUERY METHODS — EVENTS!

**CREATE
EVENT
LISTENERS**

```
                                type of event  
                                ┌──────────┐  
$( 'li' ).on( 'click', function() {  
    // your code here  
});
```



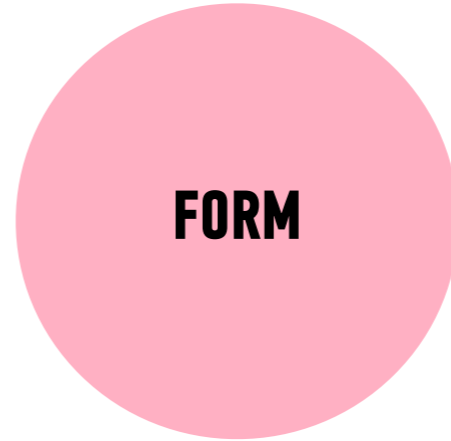
MOUSE

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!

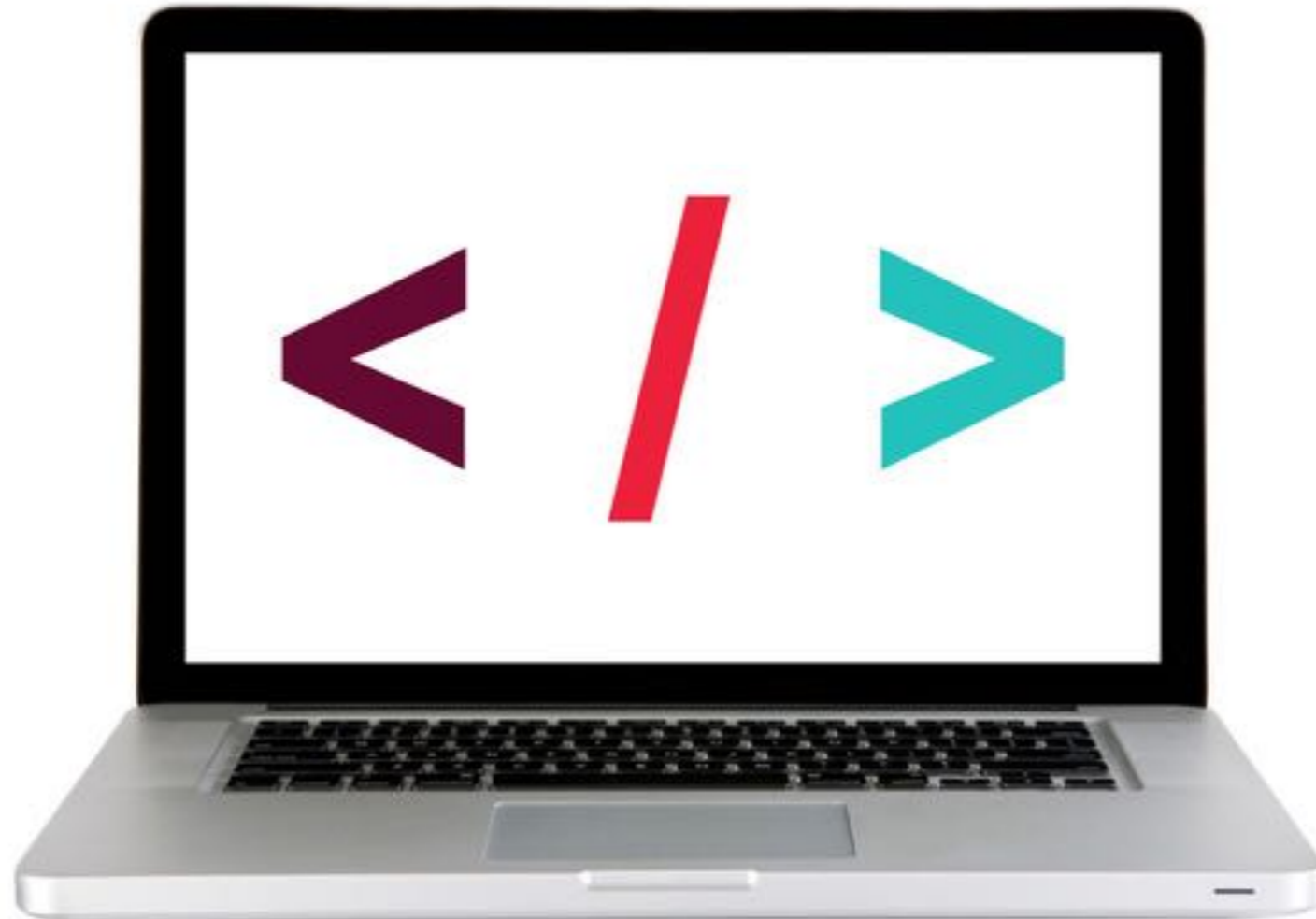
CREATE EVENT LISTENERS

selector method for
 all events type of
 event

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

function to run
when event is
triggered

LET'S TAKE A LOOK



ACTIVITY



KEY OBJECTIVE

- ▶ Utilize jQuery to access and manipulate DOM elements.

TYPE OF EXERCISE

- ▶ Individual/Partner

TIMING

5 min

Continue with 5-jquery-exercise

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

ACTIVITY



KEY OBJECTIVE

- ▶ Create DOM event handlers to respond to user actions

TYPE OF EXERCISE

- ▶ Individual/Partner

AS A CLASS

6 min

Return to 3-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 WITH JQUERY

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  
});
```

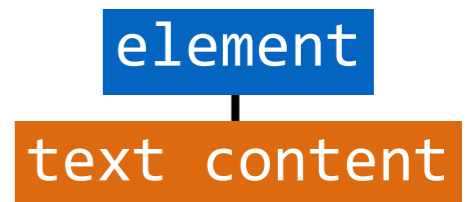
Adding content to the DOM

1. create a new element with
`$('<eLement>')`

element

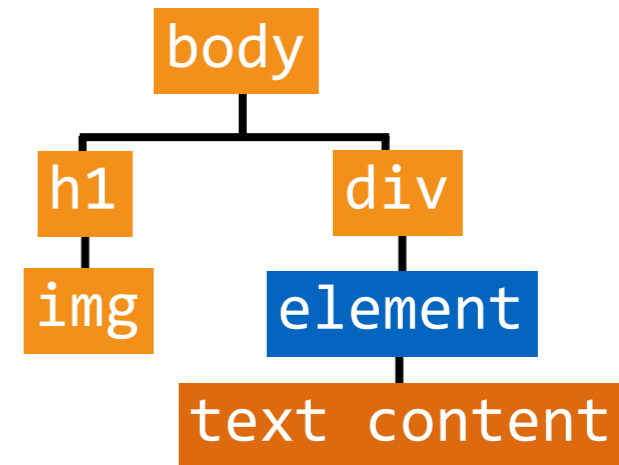
Adding content to the DOM

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



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');
```

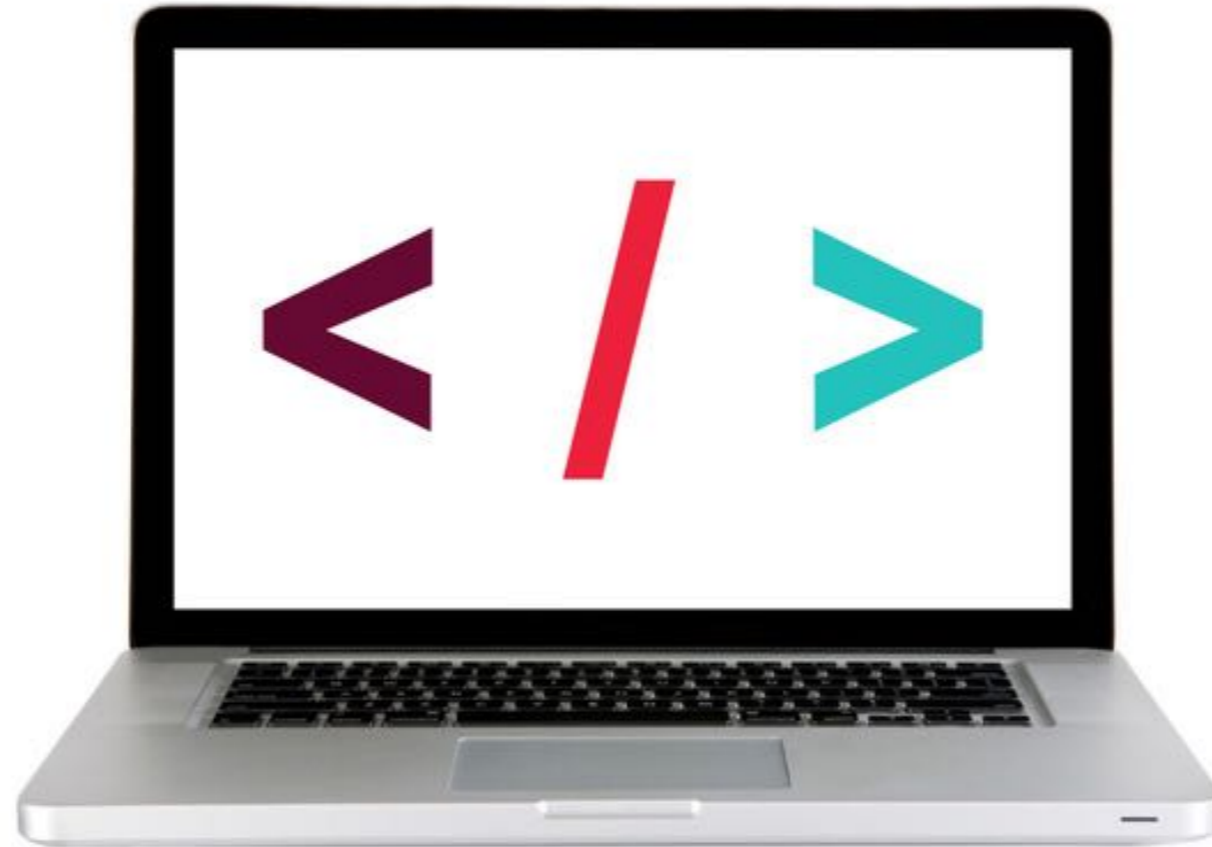
```
let item1 = $('<li>').html('<strong>Every</strong> dinosaur');  
let item2 = $('<li>').html('Books (<em>not</em> ebooks)');
```

• `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 > 8-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()

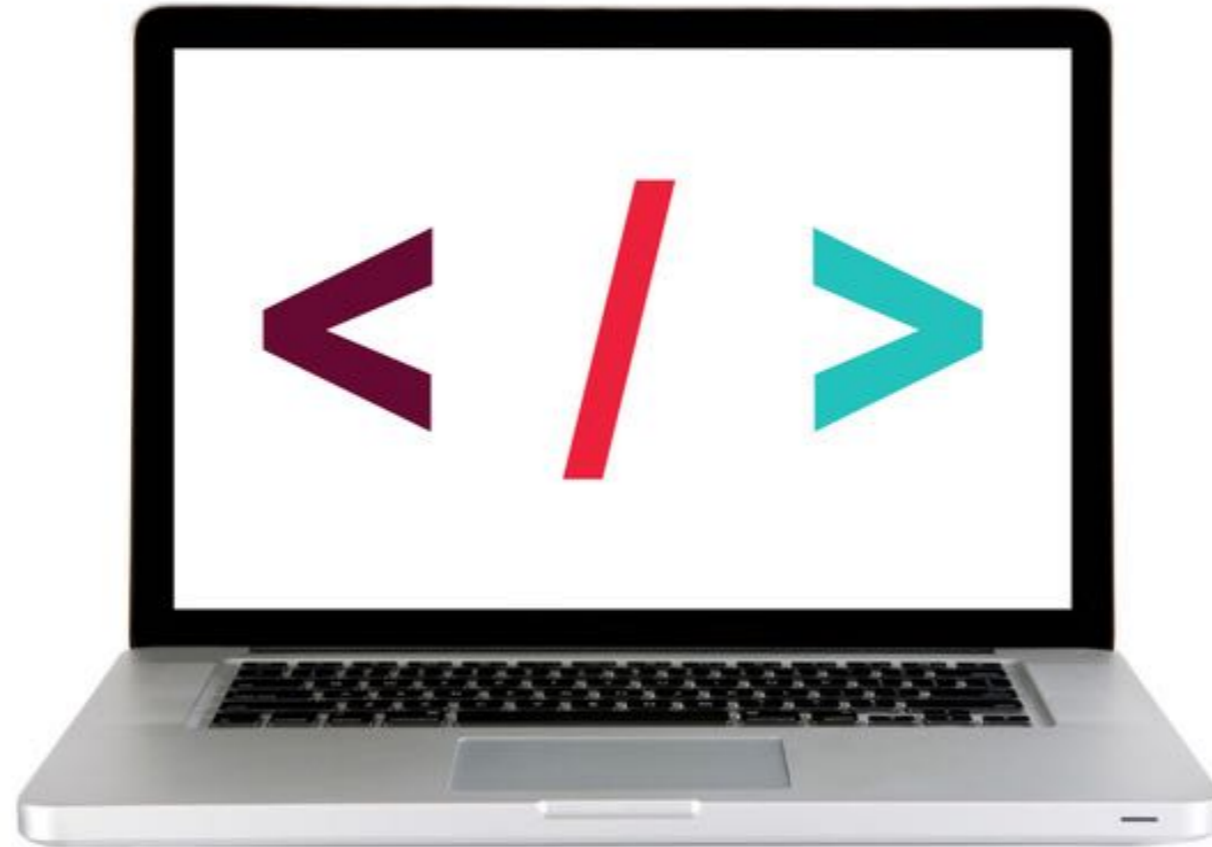
- Prevents element from executing default behavior in response to an event

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



LOCATION

▶ `starter-code > 10-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



OBJECTIVE

- ▶ Manipulate the DOM and create DOM event handlers using jQuery

LOCATION

- ▶ `starter-code > 11-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

- 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.

NEXT CLASS PREVIEW

Advanced jQuery

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

Q&A