

JAVASCRIPT DEVELOPMENT

Sasha Vodnik, Instructor

HELLO!

1. Pull changes from the `svodnik/JS-SF-12-resources` repo to your computer
2. Open the `07-dom-jquery > starter-code` folder in your code editor

JAVASCRIPT DEVELOPMENT

Intro the the DOM & jQuery

LEARNING OBJECTIVES

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

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

AGENDA

- Intro the the DOM
- jQuery
- Getting and setting DOM elements
- Responding to events

INTRO THE THE DOM & JQUERY

WEEKLY OVERVIEW

WEEK 4

Objects & JSON / Intro to the DOM & jQuery

WEEK 5

Events & jQuery / Ajax & APIs

WEEK 6

Asynchronous JS & callbacks / Advanced APIs

EXIT TICKET QUESTIONS

1. Is coercion always bad?
2. How will we use JSON?

WARM-UP EXERCISE – DOM MANIPULATION



EXERCISE

KEY OBJECTIVE

- ▶ Identify web page features that respond to user actions or other events

TYPE OF EXERCISE

- ▶ Groups of 2-3

TIMING

2 min

1. On a website you use regularly, identify at least one thing that changes after the page loads (for instance, showing new data after you click, or updating itself on a set interval).
2. Demonstrate the change to your partner/group.

THE DOCUMENT OBJECT MODEL (DOM)

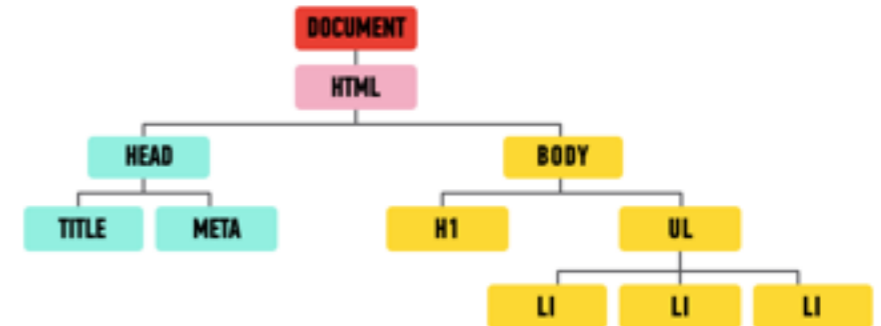
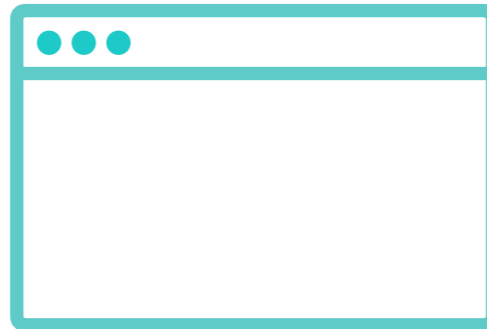
DOM TREE — HTML FILE

```
index.html *
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <title>The Evolution of Denim</title>
6 </head>
7 <body>
8
9   <h1>The Evolution of Denim</h1>
10  <p>
11    Chambray retro plaid gentrify letterpress.
    Taxidermy ennui cliche Intelligentsia. Echo
    Park umami authentic before they sold out. <a
    href="https://placekitten.com/">Forage
    wayfarers</a> listicle Kickstarter, Pitchfork
    cray messenger bag fap High Life tilde pug
    Blue Bottle mumblecore.
12  </p>
13  <ul>
14    <li>Dark Wash</li>
15    <li>Stone Wash</li>
16    <li>Chambray</li>
17  </ul>
18
19 </body>
20 </html>
```

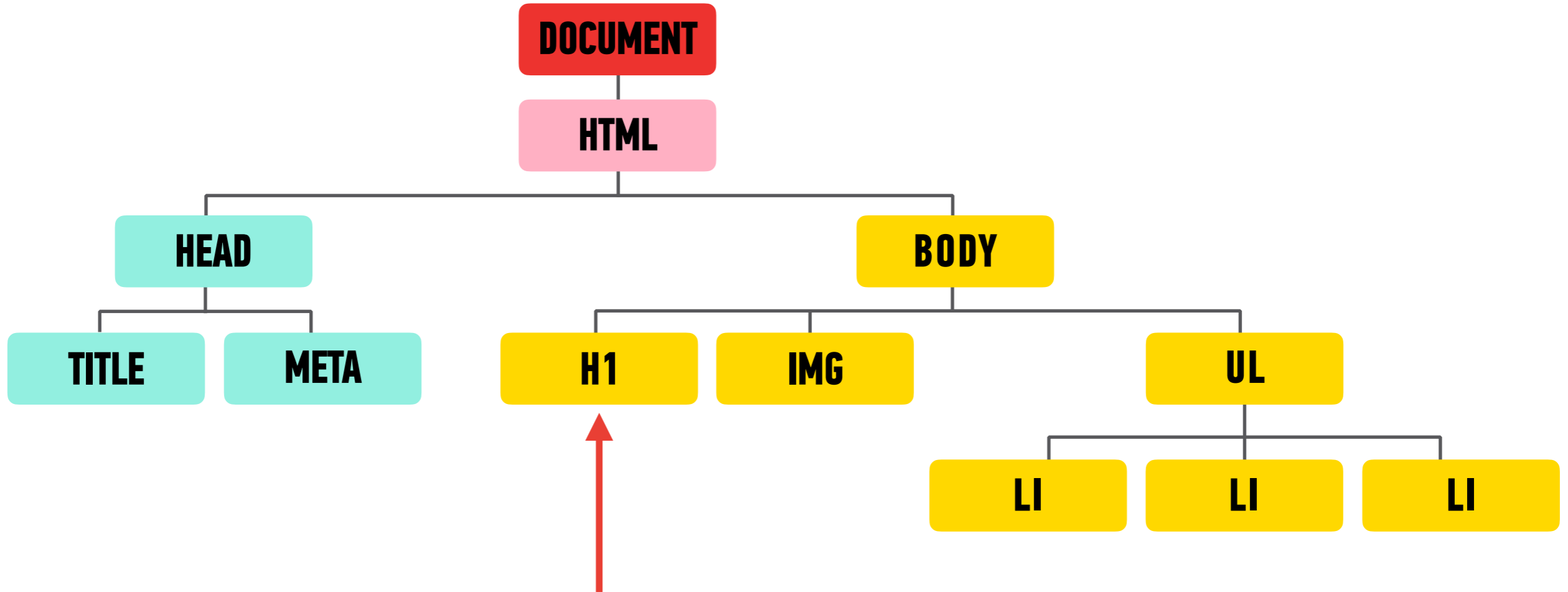
DOM TREE

- ▶ The browser pulls in this HTML document, analyzes it, and creates an *object model* of the page in memory.
- ▶ This model is called the *Document Object Model (DOM)*.
- ▶ The DOM is structured like a tree, a DOM Tree, like in the model below:

```
index.html
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4 <meta charset="UTF-8">
5 <title>The Evolution of Denim</title>
6 </head>
7 <body>
8 <h1>The Evolution of Denim</h1>
9 <p>
10 Chambray retro plaid gentrify letterpress.
11 Taxidermy ennui cliche Intelligentsia. Echo
12 Park umami authentic before they sold out. <a
13 href="https://placekitten.com/">Forage
14 wayfarers</a> listicle Kickstarter, Pitchfork
15 cray messenger bag fao High Life tilde pug
16 Blue Bottle mumblecore.
17 </p>
18 <ul>
19 <li>Dark Wash</li>
20 <li>Stone Wash</li>
21 <li>Chambray</li>
22 </ul>
23 </body>
24 </html>
```



DOM TREE



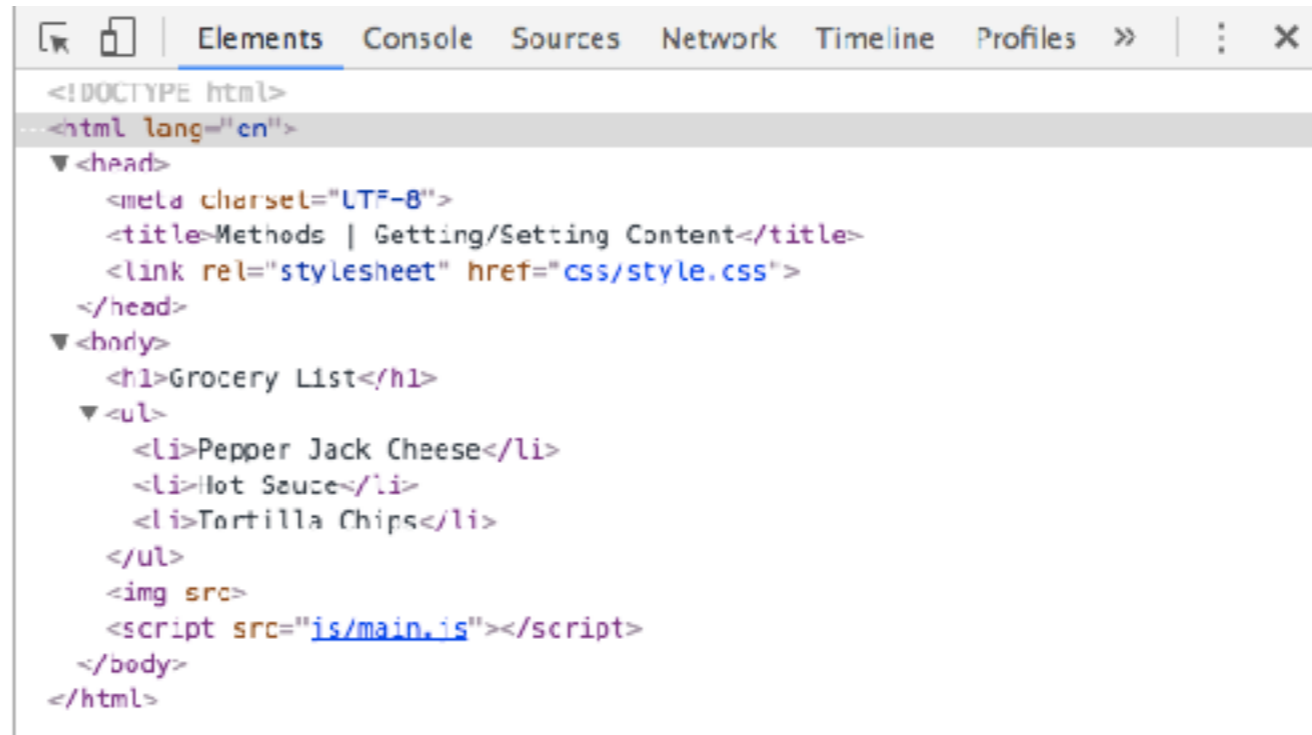
- ▶ Each element in the HTML document is represented by a *DOM node*.
- ▶ You can think of a node as a live object that you can access and change using JavaScript.
- ▶ When the model is updated, those changes are reflected on screen.

DOM TREE

- ▶ In Chrome, you can go to View > Developer > Developer Tools and click on the Elements panel to take a look at the DOM tree.

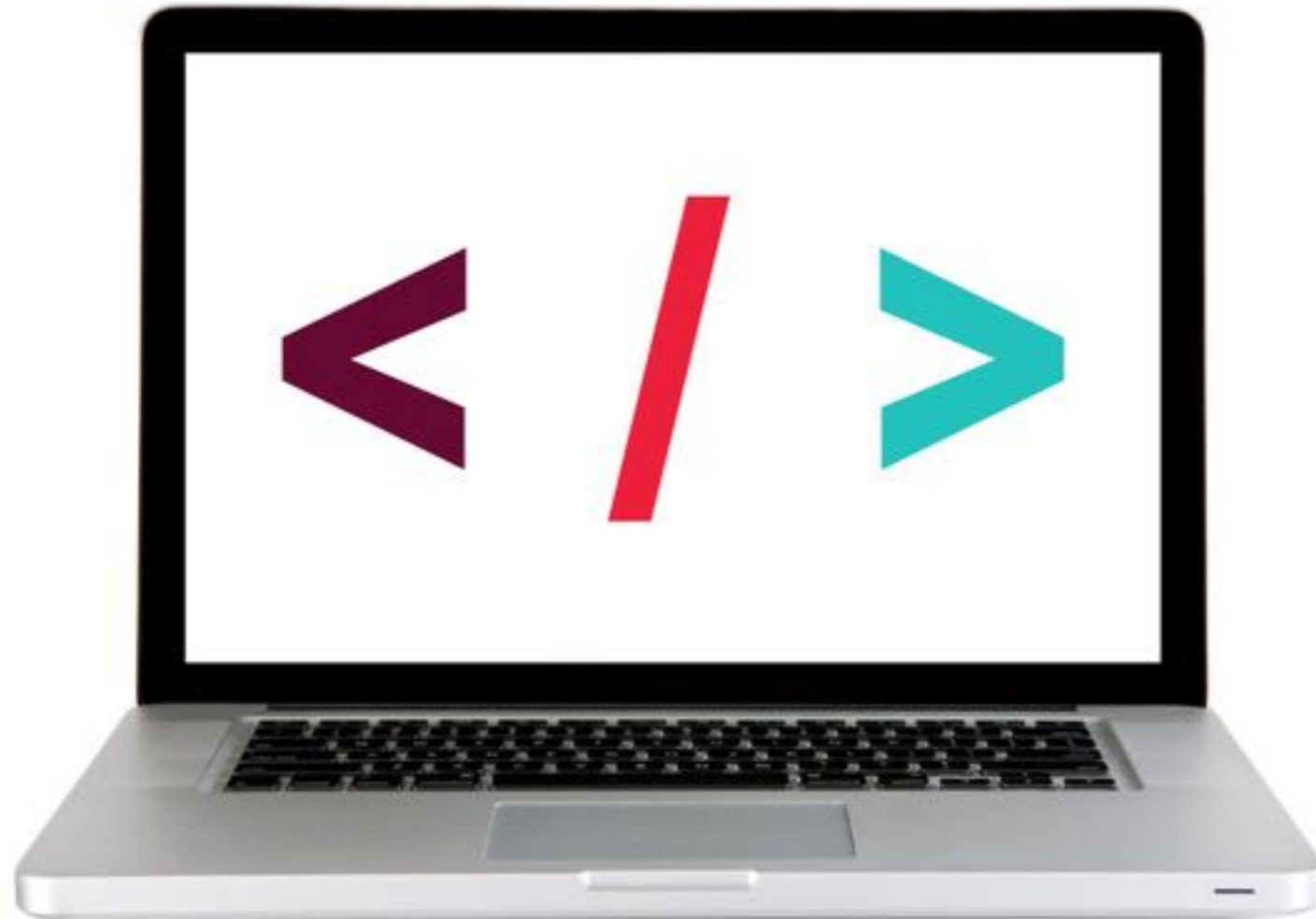
Grocery List

- Pepper Jack Cheese
- Hot Sauce
- Tortilla Chips



```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8">
    <title>Methods | Getting/Setting Content</title>
    <link rel="stylesheet" href="css/style.css">
  </head>
  <body>
    <h1>Grocery List</h1>
    <ul>
      <li>Pepper Jack Cheese</li>
      <li>Hot Sauce</li>
      <li>Tortilla Chips</li>
    </ul>
    <img src="">
    <script src="js/main.js"></script>
  </body>
</html>
```

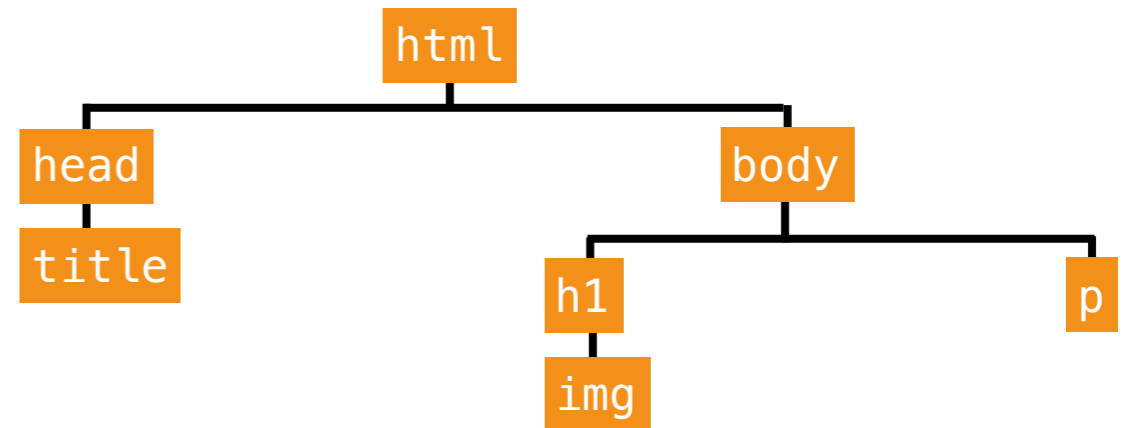
LET'S TAKE A LOOK



Web page elements

DOM Tree

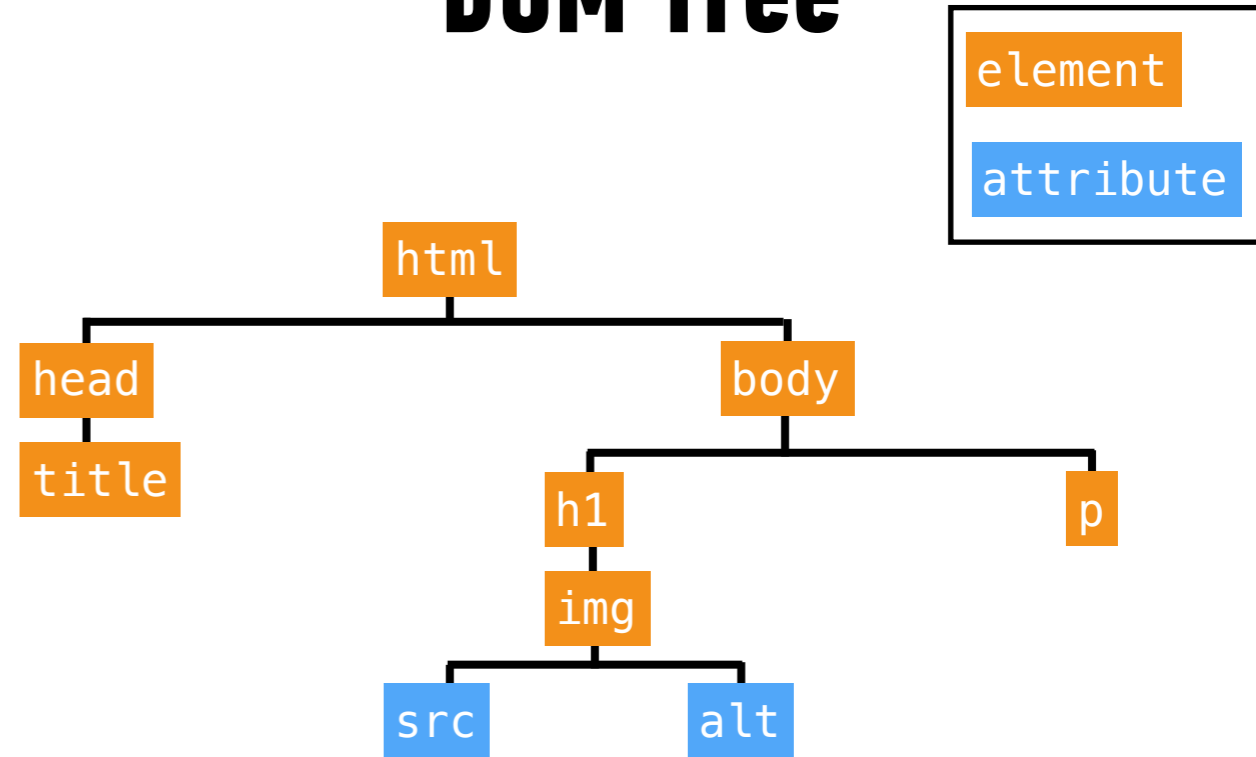
```
<html>
  <head>
    <title>JavaScript Basics</title>
  </head>
  <body>
    <h1>
      
    </h1>
    <p>First, master HTML and CSS.</p>
  </body>
</html>
```



Web page elements

```
<html>
  <head>
    <title>JavaScript Basics</title>
  </head>
  <body>
    <h1>
      
    </h1>
    <p>First, master HTML and CSS.</p>
  </body>
</html>
```

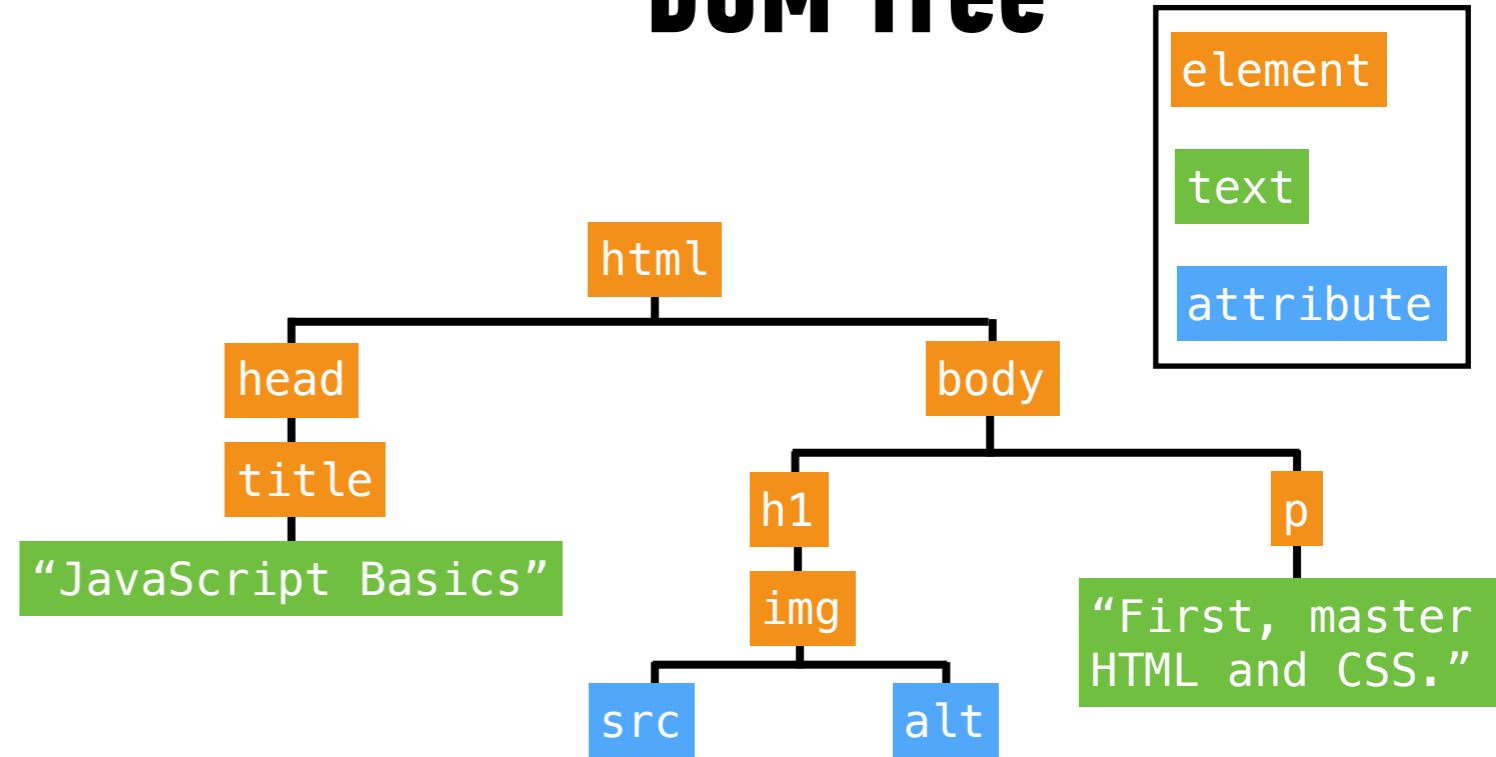
DOM Tree



Web page elements

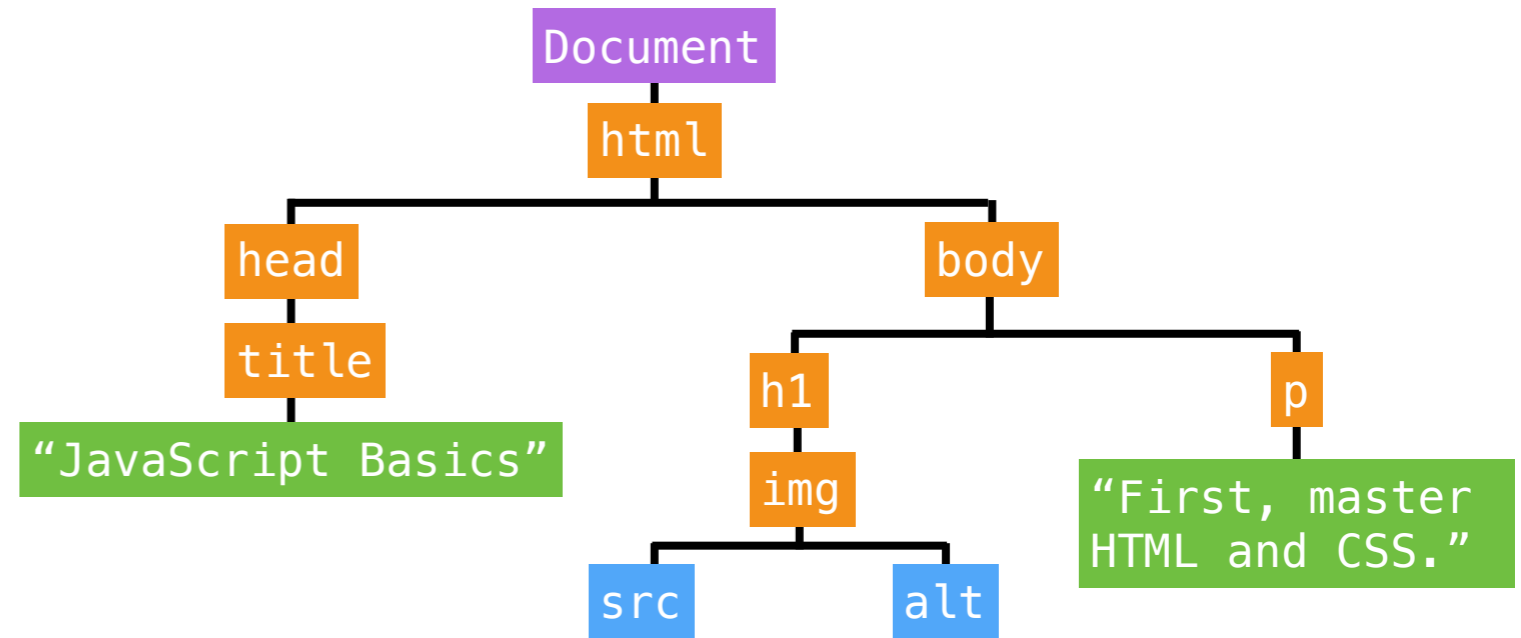
```
<html>
  <head>
    <title>JavaScript Basics</title>
  </head>
  <body>
    <h1>
      
    </h1>
    <p>First, master HTML and CSS.</p>
  </body>
</html>
```

DOM Tree



The Document object

- ▶ Created by the browser
- ▶ Contains all web page elements as descendant objects
- ▶ Also includes its own properties and methods



EXERCISE



EXERCISE

KEY OBJECTIVE

- ▶ Identify differences between the DOM and HTML

TYPE OF EXERCISE

- ▶ Pairs

TIMING

2 min

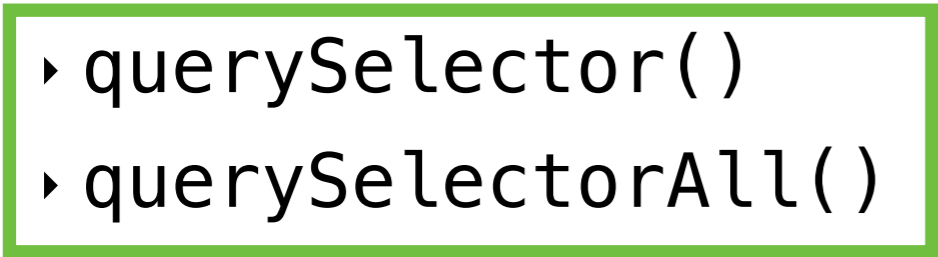
1. How is the DOM different from a page's HTML?

DOM MANIPULATION

Selecting an element in the DOM

- getElementById()
- getElementsByClassName()
- getElementsByTagName()
- querySelector()
- querySelectorAll()

Let us select DOM elements using CSS selector syntax



querySelector()

- Takes a single argument, a string containing CSS selector

HTML

```
<body>
...
<p id="main">Lorem ipsum</p>
...
</body>
```

JavaScript

```
document.querySelector('#main');
```

querySelector()

- Selects the **first** DOM element that matches the specified CSS selector

```
<body>
  ...
  <ul>
    <li>Lorem ipsum</li>
    <li>Lorem ipsum</li>
    <li>Lorem ipsum</li>
  </ul>
  ...
</body>
```

JavaScript

```
document.querySelector('li');
```

querySelectorAll()

- Takes a single argument, a string containing CSS selector
- Selects all DOM elements that match this CSS selector
- Returns a NodeList, which is similar to an array

```
<body>
...
<ul>
  <li>Lorem ipsum</li>
  <li>Lorem ipsum</li>
  <li>Lorem ipsum</li>
</ul>
...
</body>
```

JavaScript

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


What can we do with a selected element?

- Get and set its text content with the `innerHTML` property
- Get and set its attribute values by referencing them directly (`id`, `src`, etc.)

innerHTML

- › Gets the existing content of an element, including any nested HTML tags
- › Sets new content in an element

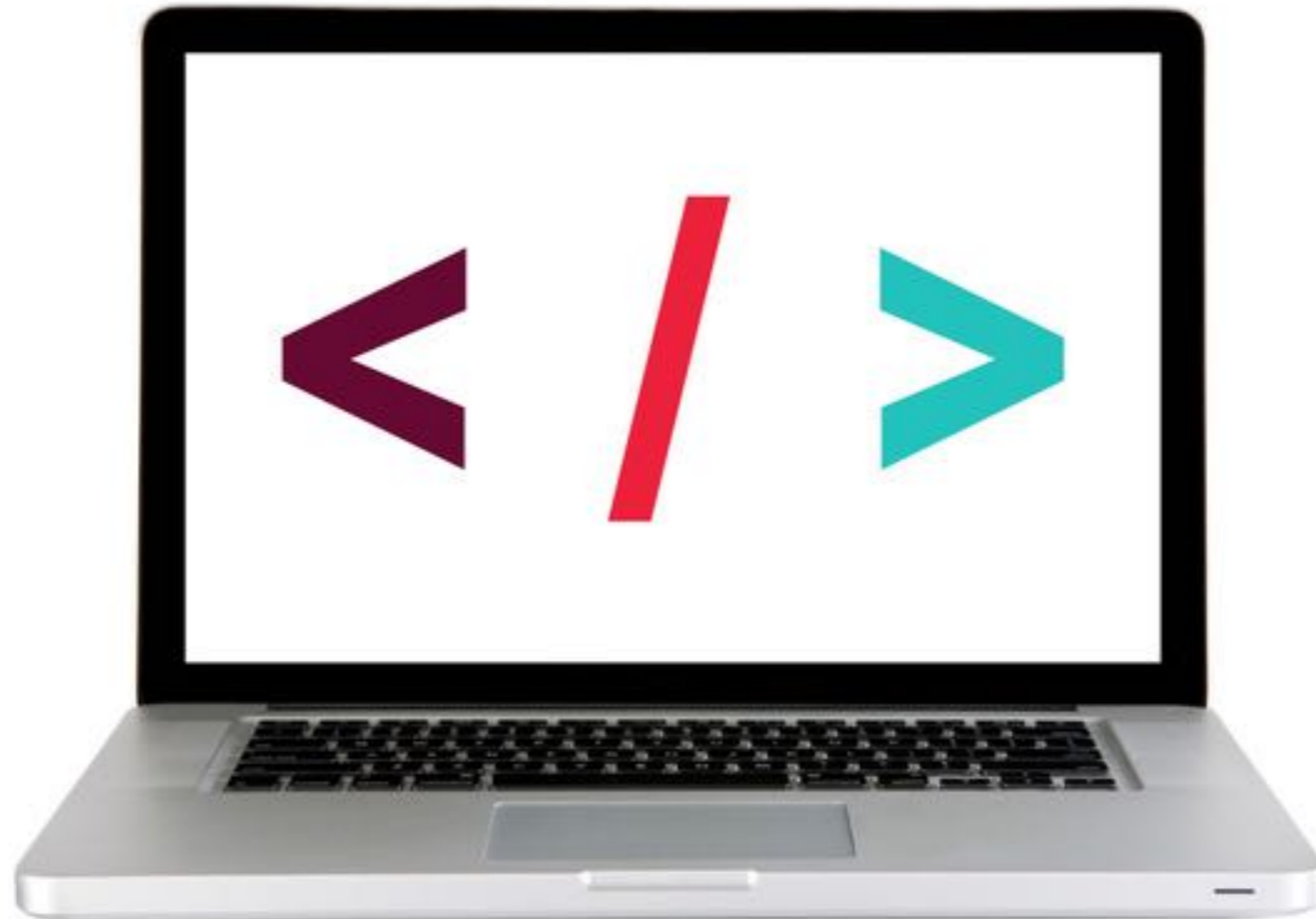
```
var item = document.querySelector('li');  
console.log(item.innerHTML) // Gets value: "Lorem ipsum"  
item.innerHTML = 'Apples' // Sets value: 'Apples'
```

className property

- Gets/sets an element's class attribute value
- CSS style sheet contains a style rule for each class
 - » Appearance of element changes based on which class is applied
 - » This is the best practice.

```
var item = document.querySelector('li');  
  
console.log(item.className) // Gets value: 'default'  
  
item.className = 'selected'  
// Sets value: 'selected'
```

LET'S TAKE A LOOK



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').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

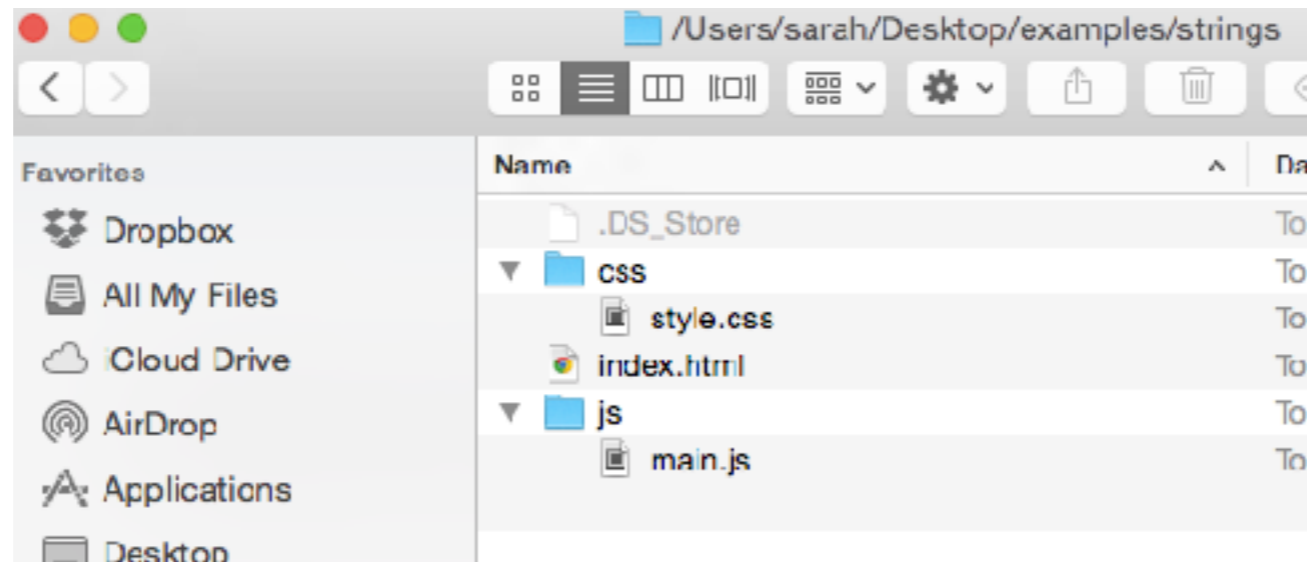
- ▶ Better performance
- ▶ Faster

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.**

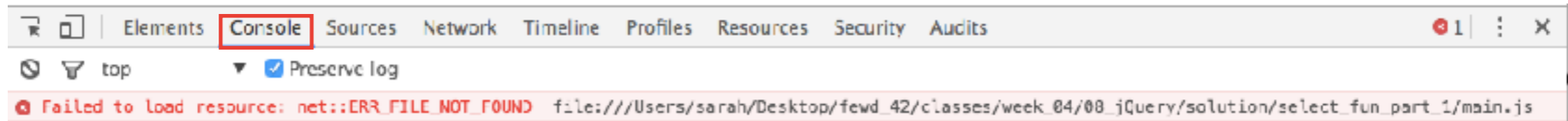
```
<body>  
  <!-- HTML content here -->  
  <script src="js/jquery-3.2.1.min.js"></script>  
  <script src="js/main.js"></script>  
</body>
```



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>
```

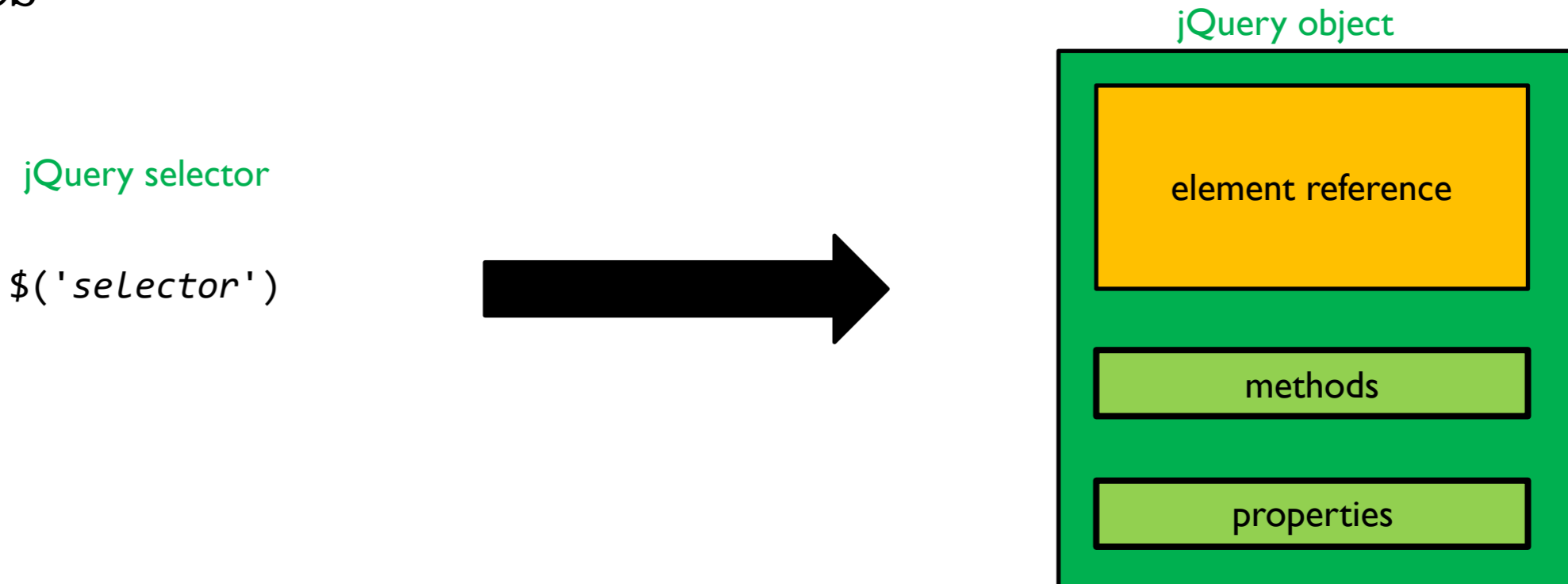
JQUERY OBJECTS

- ▶ 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

- Best practice: include \$ as the first character of any variable whose value is a jQuery object
- This is not required by jQuery, but helps us keep track of what parts of our code rely on the jQuery library

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

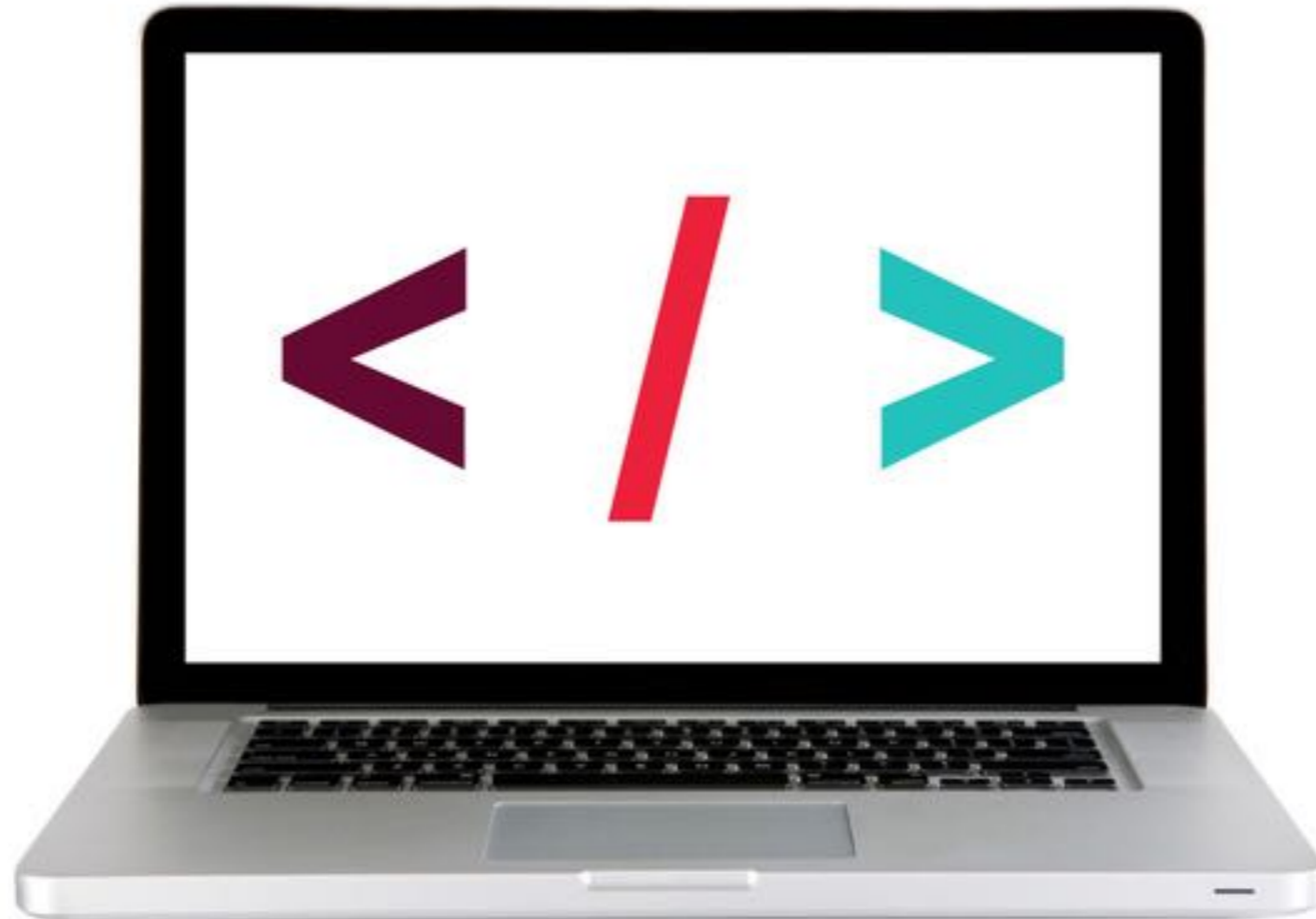
```
let $openTab = $(' .open ');
```



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

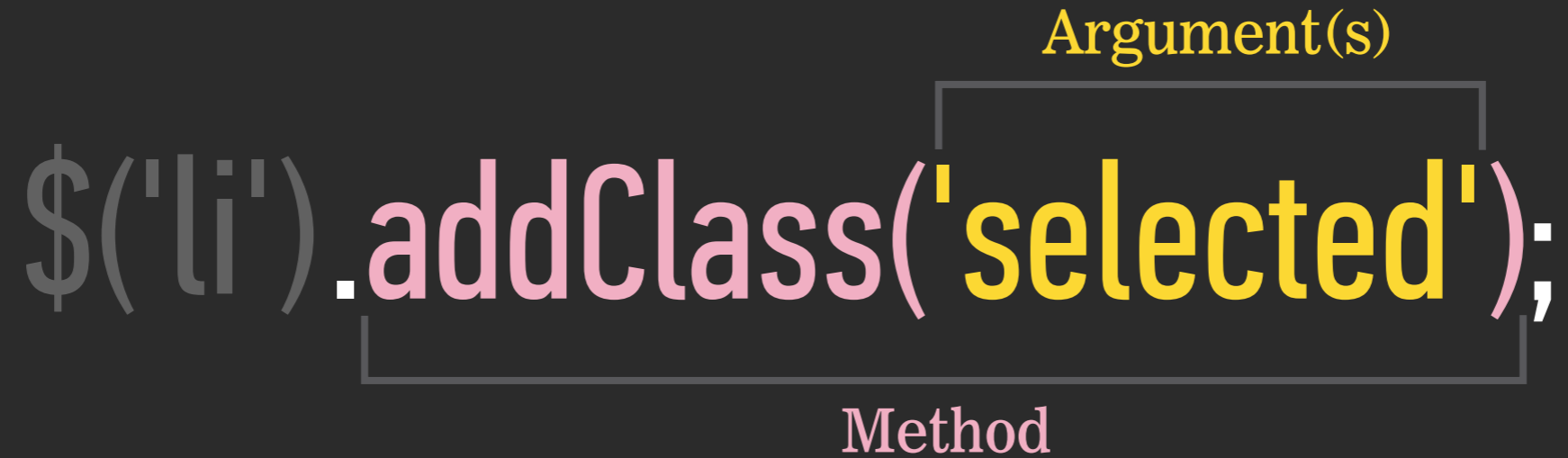
Work with those elements

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**

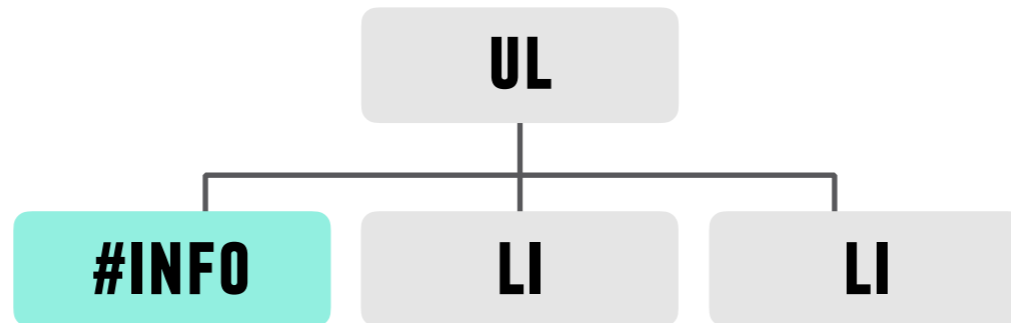
**CREATE
EVENT
LISTENERS**



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

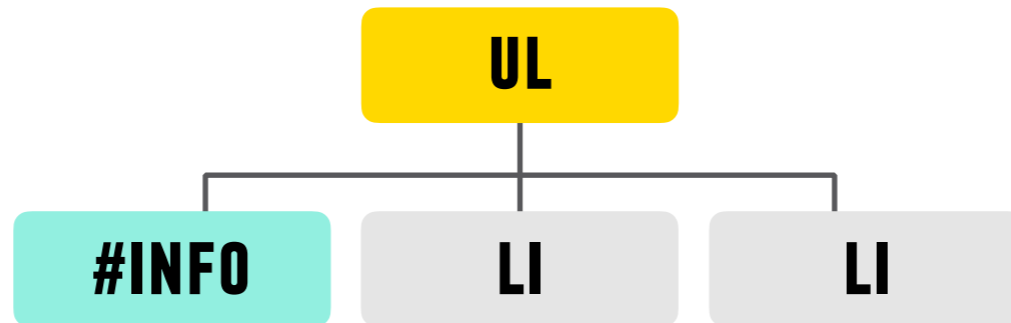
TRaversing the DOM?

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



TRaversing the DOM?

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



JQUERY METHODS — TRAVERSING THE DOM



**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 — 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**



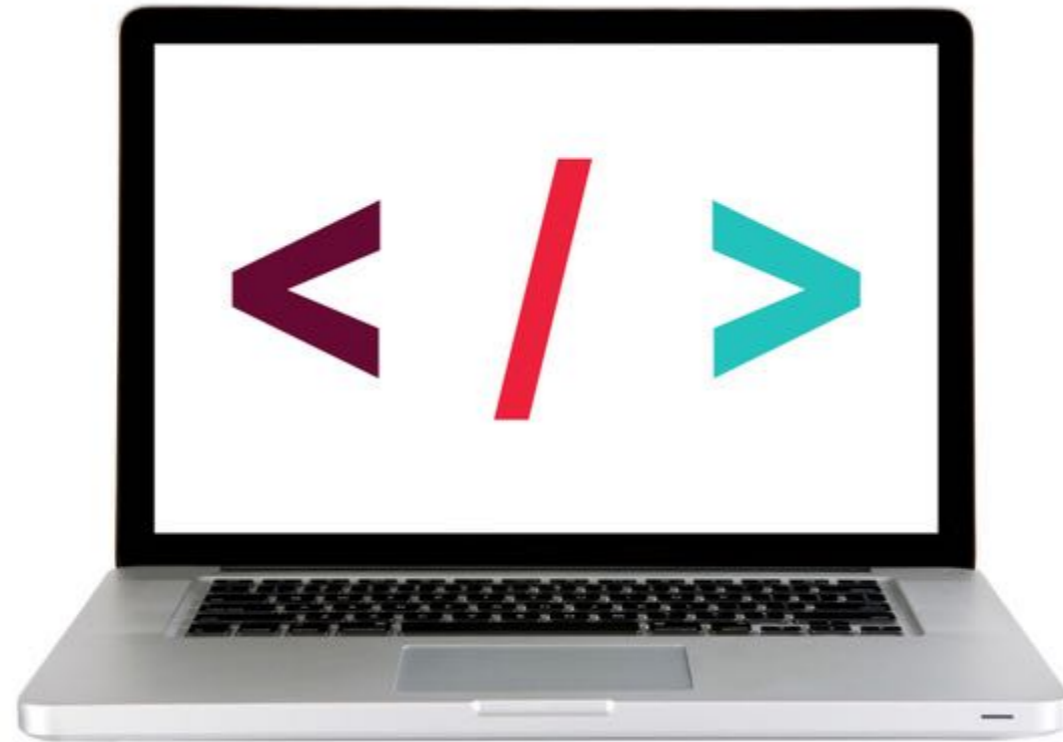
See your handout or the [jQuery docs](#) for list!

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>

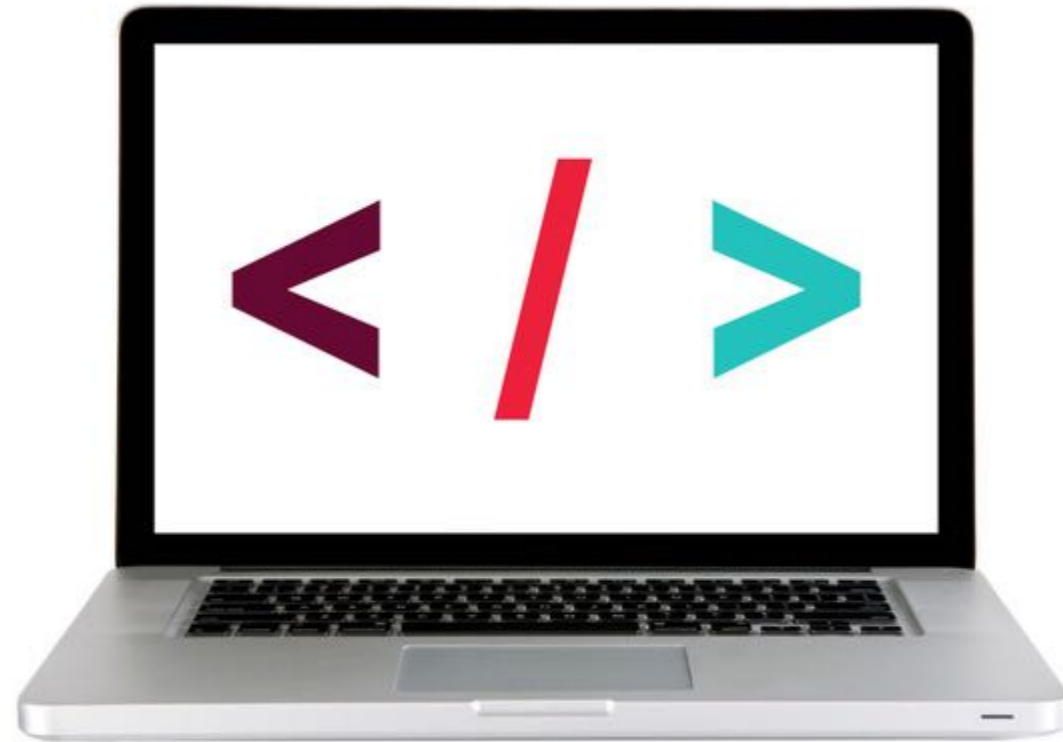
What goes in the parentheses?
The **classes** you want to change.

```
$('.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

2-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

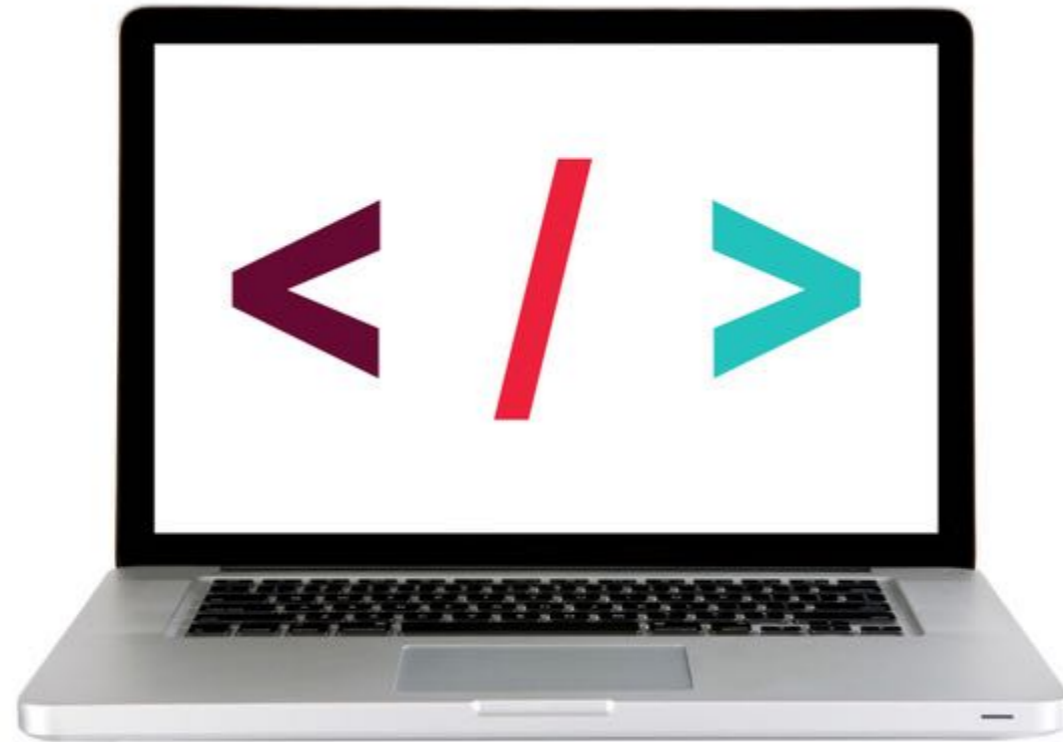
**ADD
EFFECTS/
ANIMATION**

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!

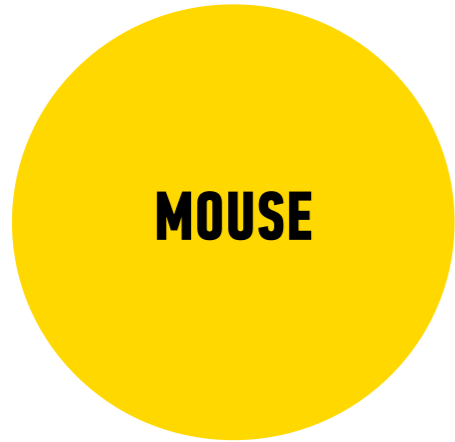
EVENT LISTENERS

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



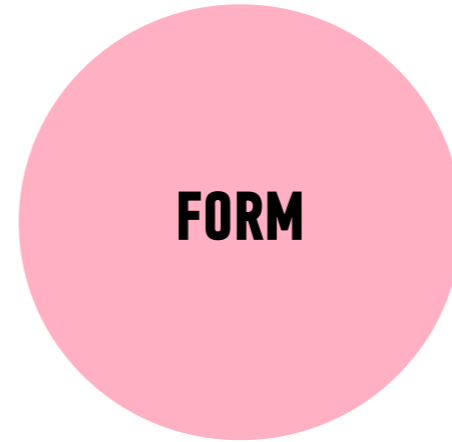
MOUSE

click
dblclick
mouseenter
mouseleave



KEYBOARD

keypress
keydown
keyup



FORM

submit
change
focus
blur



DOCUMENT

resize
scroll



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

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**

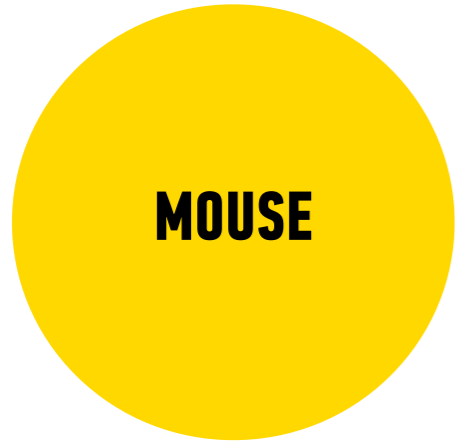
method for all events

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

JQUERY METHODS — EVENTS!

**CREATE
EVENT
LISTENERS**

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



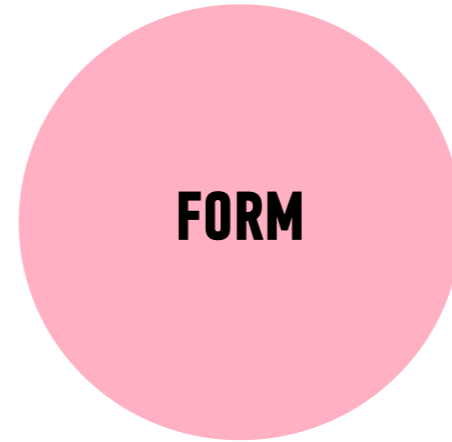
MOUSE

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!

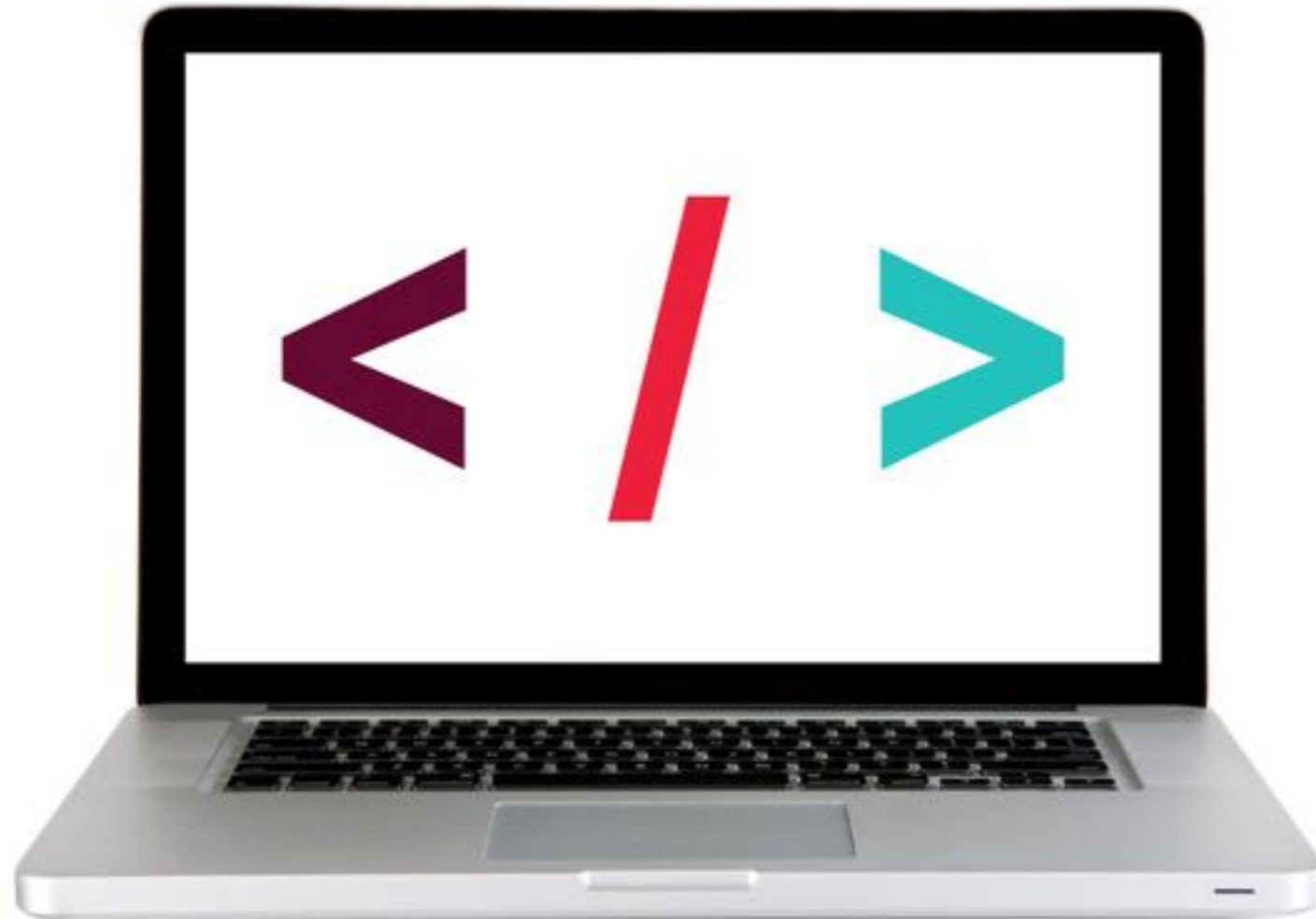


**CREATE
EVENT
LISTENERS**

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

function to run
when event is
triggered

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 2-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

10 min

Exercise is in 4-events-exercise folder

1. Add event listeners to the 3 buttons at the top of the page. Clicking each button should hide the block below it with the corresponding color.
2. Use cheat sheet/slides as a guide for syntax
3. **BONUS:** Add an event listener for the "Show all blocks" button that removes the hidden class from all the colored block elements.

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.onclick = function(event) {  
    event.preventDefault();  
    ...  
}
```

EXERCISE



KEY OBJECTIVE

- ▶ Create DOM event handlers to respond to user actions

LOCATION

- ▶ `starter-code > 5-js-dom-exercise`

TIMING

10 min

1. Open `index.html` in your browser.
2. Open `main.js` in your editor, then follow the instructions to make the submit button functional and use DOM manipulation to add items to the list.
3. **BONUS:** Add functionality that adds a message to the page to alert the user when they click Submit without typing anything. (Use DOM manipulation, not the `alert` method.)

Exit Tickets!

(Class #7)

LEARNING OBJECTIVES – REVIEW

- Describe the difference between the DOM and HTML.
- 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