

JAVASCRIPT DEVELOPMENT

Sasha Vodnik, Instructor

HELLO!

- 1. Pull changes from the svodnik/JS-SF-16-resources repo to your computer
- 2. Open the 07-events-jquery folder in your editor

Intro to the DOM & iQuery

LEARNING OBJECTIVES

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

- Identify differences between the DOM and HTML.
- Use vanilla JavaScript methods and properties to create and modify DOM nodes.
- Create DOM event handlers using vanilla JavaScript.
- Select DOM elements and properties using jQuery.
- Manipulate the DOM by using jQuery selectors and functions.

AGENDA

- Intro to the DOM
- Creating DOM nodes with vanilla JS
- JavaScript events
- jQuery
- Getting and setting DOM elements with jQuery

INTRO TO JQUERY

WEEKLY OVERVIEW

WEEK 4

Objects & JSON / Intro to DOM & jQuery

WEEK 5

Advanced jQuery / Ajax & APIs

WEEK 6

Asynchronous JS & callbacks / Advanced APIs

EXIT TICKET QUESTIONS

- 1. Can I build a bot just search any site? without google?
- 2. Can slackbots reference or trigger other slack apps?
- 3. Would like to spend more time on JSON. given the importance of how prevalent it is used in programming
- 4. Are there any common JSON methods or prebuilt functions that we can leverage?

Whoops

Unable to successfully deserialize JSON object

Ok

WARM-UP EXERCISE - DOM MANIPULATION



KEY OBJECTIVE

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

TYPE OF EXERCISE

Pairs

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.

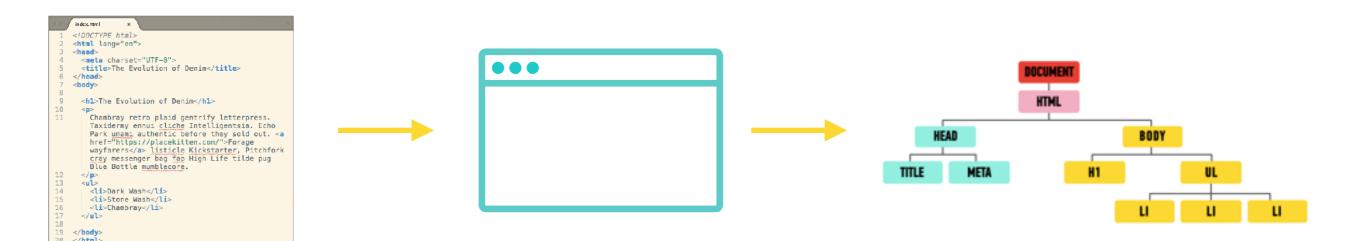
THE DOCUMENT OBJECT MODEL (DOM)

DOM TREE — HTML FILE

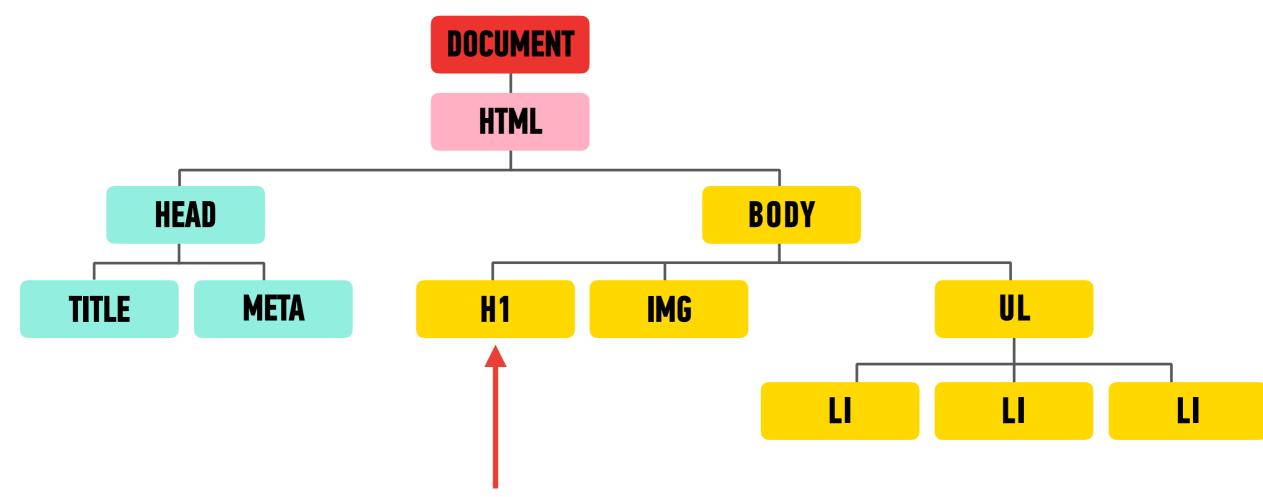
```
index.html
    <!DOCTYPE html>
    <html lang="en">
   <head>
     <meta charset="UTF-8">
     <title>The Evolution of Denim</title>
   </head>
    <body>
8
9
     <h1>The Evolution of Denim</h1>
10
11
     Chambray retro plaid gentrify letterpress.
       Taxidermy ennui cliche Intelligentsia. Echo
       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.
     < 11 |
14
      >li>Dark Wash
     Stone Wash
15
16
     Chambray
17
     18
19
   </body>
   </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:



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

```
Elements Console Sources Network Timeline Profiles >>
<!DOCTYPE html>
<html lang="en">
▼<head>
   <meta charset="UTF-8">
   <title>Methods | Getting/Setting Content</title>
   k rel="stylesheet" href="css/style.css">
 </head>
▼<body>
   <h1>Grocery List</h1>
 ▼
    Pepper Jack Cheese
    Hot Sauce
    Iortilla Chips
   <img src>
   <script src="is/main.is"></script>
 </body>
</html>
```

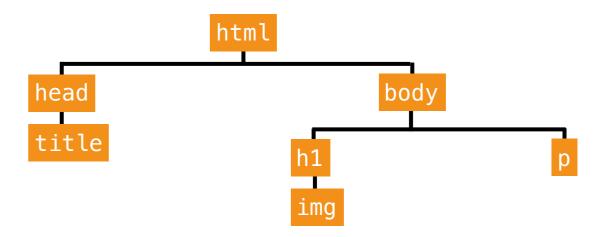
LET'S TAKE A LOOK



Web page elements

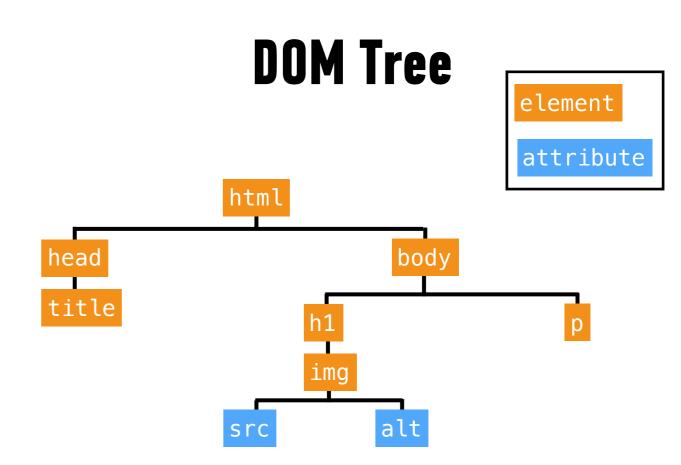
```
<html>
    <head>
        <title>JavaScript Basics</title>
    </head>
    <body>
        <h1>
            <img src="logo.png" alt="JS Basics">
              </h1>
            First, master HTML and CSS.
        </body>
</html>
```

DOM Tree



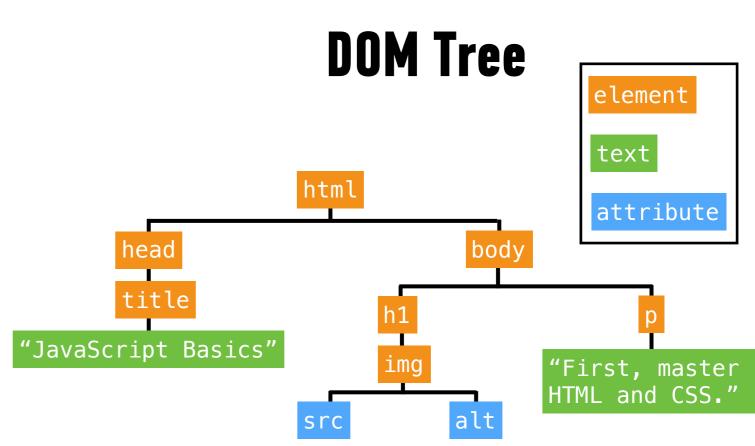
Web page elements

```
<html>
    <head>
        <title>JavaScript Basics</title>
    </head>
    <body>
        <h1>
            <img src="logo.png" alt="JS Basics">
            </h1>
            First, master HTML and CSS.
        </body>
</html>
```



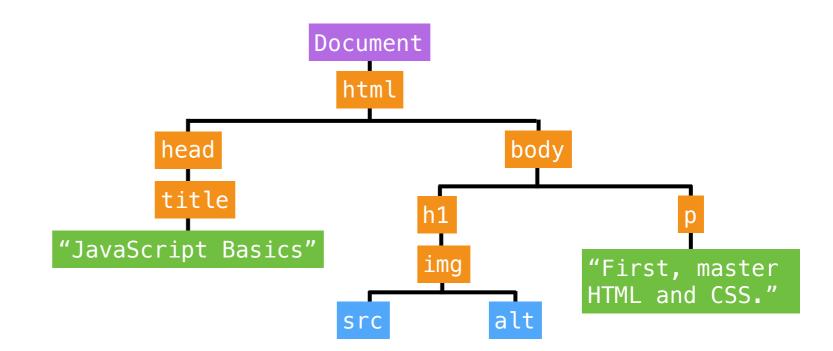
Web page elements

```
<html>
    <head>
        <title>JavaScript Basics</title>
    </head>
    <body>
        <h1>
            <img src="logo.png" alt="JS Basics">
              </h1>
            First, master HTML and CSS.
        </body>
</html>
```



The Document object

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



EXERCISE



KEY OBJECTIVE

▶ Identify differences between the DOM and HTML

TYPE OF EXERCISE

Pairs

TIMING

2 min

1. Discuss how the DOM is different from a page's HTML

DOM MANIPULATION

SELECTING DOM ELEMENTS

Selecting an element in the DOM

```
• getElementById()
```

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

Let us select DOM elements using CSS selector syntax

24

INTRO THE THE DOM & JQUERY

CSS SELECTORS

CSS SELECTORS REVIEW

Selecting all instances of an element

the name of the element as a string

'div'

Selecting all instances of multiple elements

the names of the elements separated by commas, as a single string

'div, span'

Selecting the element with a specified ID value

the hash symbol (#) followed by the name of the ID value

'#top'

Selecting all elements with a specified class value

a period (.) followed by the name of the class value

'.pullquote'

Selecting all elements with a given relationship

the hierarchical relationship, ending with the element to select

'div p a'

Selecting all elements with a given relationship

the hierarchical relationship, ending with the element to select

'div p .pullquote'

CSS SELECTORS QUIZ

Select all paragraph elements and list item elements

'p li'

Select all paragraph elements and list item elements

'p, li'

Select unordered lists within divs

'ul div'

Select unordered lists within divs

'div ul'

Select the element with the ID value 'main'

'main'

Select the element with the ID value 'main'

'#main'

Select the elements with the class value 'col3'

':col3'

Select the elements with the class value 'col3'

'.co13'

Select the elements with the class value 'col3' nested within div elements

'div.col3'

Select the elements with the class value 'col3' nested within div elements

'div .col3'

querySelector()

Takes a single argument, a string containing CSS selector

HTML JavaScript

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

document.querySelector('#main');

querySelector()

Selects the first DOM element that matches the specified CSS selector

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

JavaScript

document.querySelectorAll('li');

WORKING WITH SELECTED ELEMENTS

What can we do with a selected element?

- Get and set the text within it with the textContent property
- Get and set the HTML within it with the innerHTML property
- Get and set its attribute values by referencing them directly (id, src, etc.)

textContent

• Gets the existing text content of an element

```
let item = document.querySelector('li');
console.log(item.textContent)
// Gets value: "Lorem ipsum"
```

textContent

Sets new text content in an element

```
let item = document.querySelector('li');
item.textContent = 'Apples'
// Sets value: 'Apples'
```

innerHTML

• Gets the existing content of an element, including any nested HTML tags

```
let item = document.querySelector('li');
console.log(item.innerHTML)
// Gets value: "<strong>Lorem ipsum</strong>""
```

innerHTML

Sets new content in an element, including nested HTML tags

```
let item = document.querySelector('li');
item.innerHTML = '<em>Apples</em>'
// Sets value: '<em>Apples</em>'
```

className property

• Gets an element's class attribute value

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

className property

- 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.

```
let item = document.querySelector('li');
item.className = 'selected'
// Sets value: 'selected'
```

classList.add method

Adds specific class attribute value(s)

```
let item = document.querySelector('li');
item.classList.add('selected');
// Adds value: 'selected'
```

classList.remove method

Removes specific class attribute value(s)

```
let item = document.querySelector('li');
item.classList.remove('default');
// Removes value: 'default'
```

classList.toggle method

Removes existing or adds missing class attribute value(s)

```
let item = document.querySelector('li');
item.classList.toggle('selected');
// Adds value 'selected' if not already in collection
// Removes value 'selected' if already in collection
```

LET'S TAKE A LOOK



LAB — DOM



KEY OBJECTIVE

Use vanilla JavaScript methods and properties to create and modify DOM nodes.

TYPE OF EXERCISE

Individual or pair

TIMING

5 min

- Open starter-code > 01-dom-exercise > app.js in your editor.
- 2. Follow the instructions to write code that selects and modifies the indicated elements and content.

LAB — DOM



KEY OBJECTIVE

 Use vanilla JavaScript methods and properties to create and modify DOM nodes.

TYPE OF EXERCISE

Individual or pair

TIMING

5 min

- 2. Follow the instructions to write code that selects and modifies the indicated elements and content.

MANUSCR PI DOM CREATON

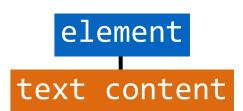
Adding content to the DOM

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



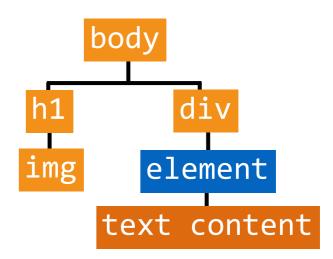
Adding content to the DOM

- create a new element with document.createElement()
- 2. add content to that element with a property such as textContent, innerHTML, or src



Adding content to the DOM

- create a new element with document.createElement()
- 2. add content to that element with a property such as textContent, innerHTML, or src
- 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.textContent ='banana';
item2.textContent ='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



KEY OBJECTIVE

▶ Use vanilla JavaScript methods and properties to create and modify DOM nodes.

TYPE OF EXERCISE

Pairs

TIMING

2 min

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

EXERCISE - ADD CONTENT TO A WEB PAGE USING JAVASCRIPT



LOCATION

starter-code > 04-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', () => {
    // your code here
});
```

EVENT LISTENERS

```
let button = document.querySelector('.submitBtn');
    method to add event listener

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

EVENT LISTENERS

MOUSE **KEYBOARD FORM** keypress click submit resize dblclick keydown change scroll focus keyup mouseenter mouseleave blur button.addEventListener('eventgoeshere', () => { // your code here }, false);

EVENT LISTENERS

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

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

function to run when event is triggered

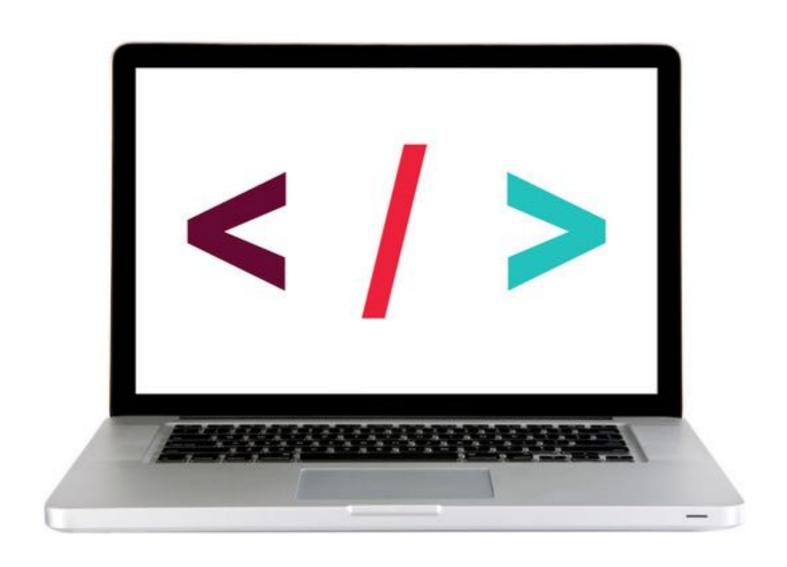
EVENT LISTENERS

```
element reference method to add event listener type of event

button.addEventListener('click', () => {
    // 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

06-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

WORKING WITH EVENT OBJECTS

preventDefault()

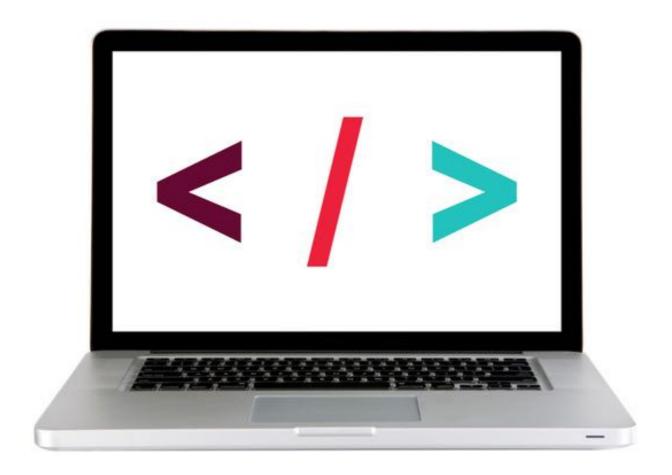
Prevents element from executing default behavior in response to an event

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',(event) => {
  event.preventDefault();
   ...
)};
```

EVENTS & JQUERY



LET'S TAKE A CLOSER LOOK

EXERCISE



LOCATION

starter-code > 08-event object—exercise

TIMING

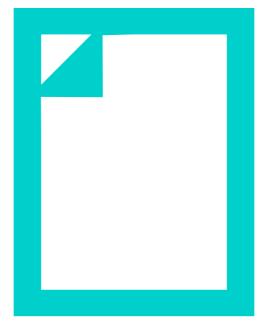
2 min

- 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.

JQUERY

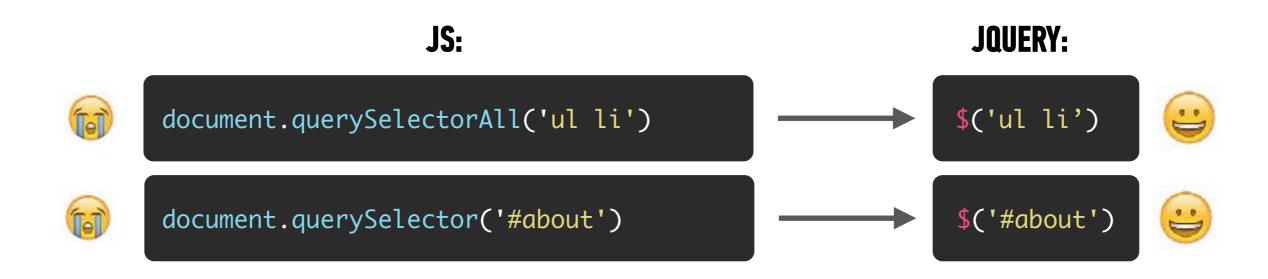
INTRO TO JQUERY — YOUR NEW BEST FRIEND!

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!



JQUERY VS. JAVASCRIPT

jQuery statements for DOM manipulation are also more concise!

```
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

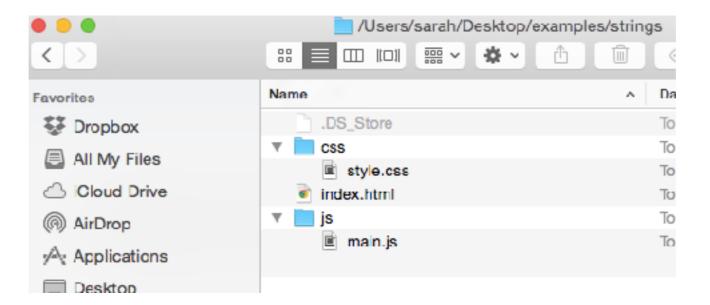
PURE JAVASCRIPT

- Better performance
- Faster

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 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>
```



JQUERY

PART 1 —— SELECT AN ELEMENT

A JQUERY STATEMENT INVOLVES 2 PARTS

Select an element/elements

2 Work with those elements

Select an element/elements

Work with those elements

JQUERY — **SELECTING ELEMENTS**

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

JQUERY OBJECTS — FINDING ELEMENTS: SOME EXAMPLES

| | CSS | JQUERY |
|-----------------|--------------------------------------|---------------------------|
| ELEMENT | <pre>a { color: blue; }</pre> | \$('a') |
| ID | <pre>#special { color: blue; }</pre> | <pre>\$('#special')</pre> |
| CLASS | <pre>.info { color: blue; }</pre> | \$('.info') |
| NESTED SELECTOR | <pre>div span { color: blue; }</pre> | \$('div span') |

```
<button id="form-submit">Submit</button>
One
<h1>Color Scheme Switcher</h1>
```

JQUERY OBJECTS

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

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

jQuery selector
\$('selector')

methods

properties

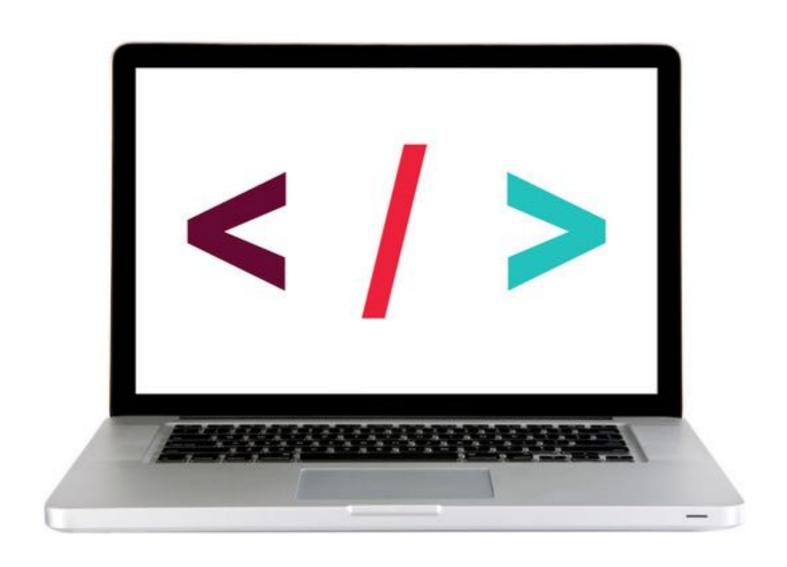
NAMING VARIABLES WHEN USING JQUERY

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

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

Select an element/elements

Work with those elements

JQUERY — WORKING WITH THOSE ELEMENTS

\$('li').addClass('selected'); Method

JQUERY METHODS

Be forewarned!

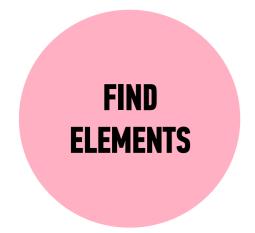
There are a lot of methods!

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:



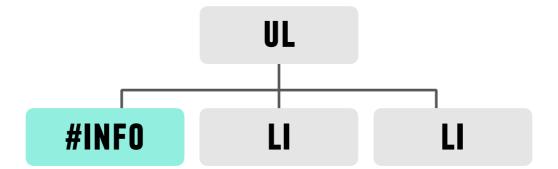
GET/SET CONTENT



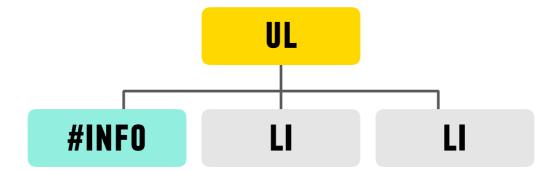




TRAVERSING THE DOM?



TRAVERSING THE DOM?



JQUERY METHODS — TRAVERSING THE DOM

FIND ELEMENTS

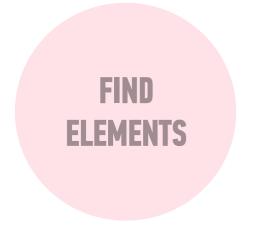
- ▶ Think of these as filters, or part of the selection process.
- ▶ They must come *directly after another selection*

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

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:











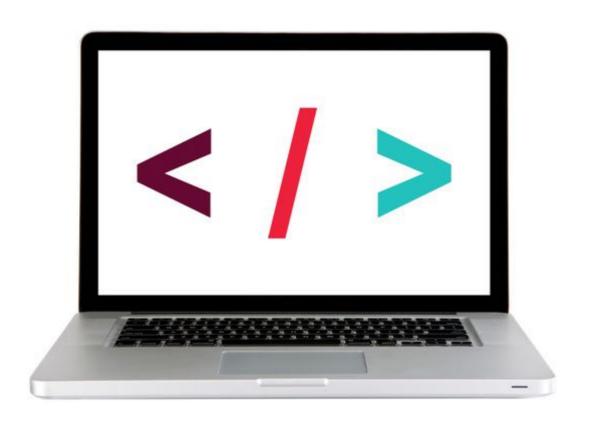
GETTING/SETTING CONTENT — PART 1

Get/change content of elements and attributes

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

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

LET'S TAKE A CLOSER LOOK



GETTING/SETTING CONTENT — PART 2

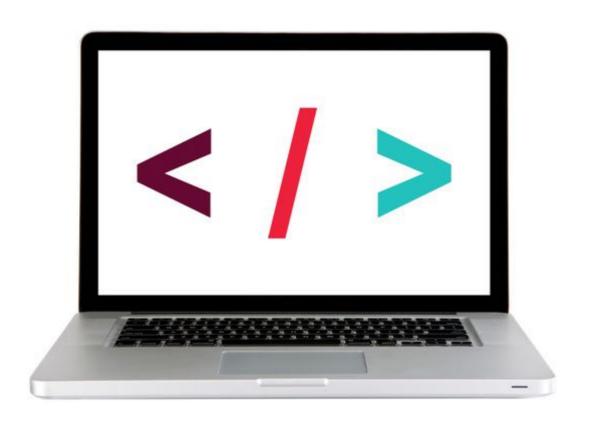
Get/change content of elements and attributes

| METHODS | EXAMPLES |
|----------------|--------------------------------------------------|
| .addClass() | <pre>\$('p').addClass('success');</pre> |
| .removeClass() | <pre>\$('p').removeClass('my-class-here');</pre> |
| .toggleClass() | <pre>\$('p').toggleClass('special');</pre> |

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

10-jquery-statements-exercise

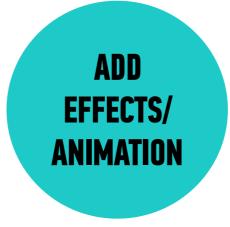
- 1. Follow the instructions under part 1 in main.js
- 2. Use handout/slides as a guide for syntax

JQUERY METHODS — WORKING WITH THOSE ELEMENTS

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



GET/SET CONTENT







JQUERY METHODS — EFFECTS/ANIMATION

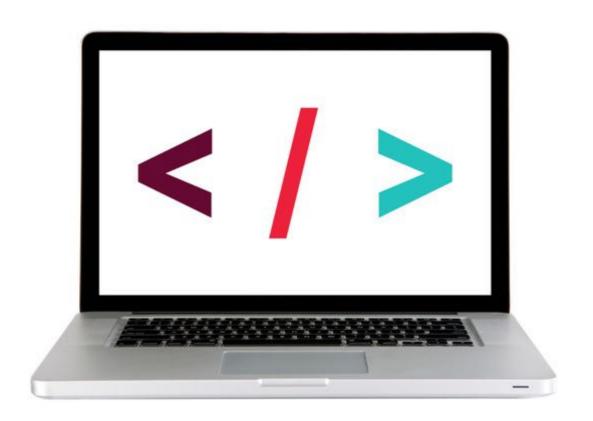
ADD EFFECTS/ ANIMATION

Add effects and animation to parts of the page

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

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:



GET/SET CONTENT







EVENTS



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

CREATE EVENT LISTENERS

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

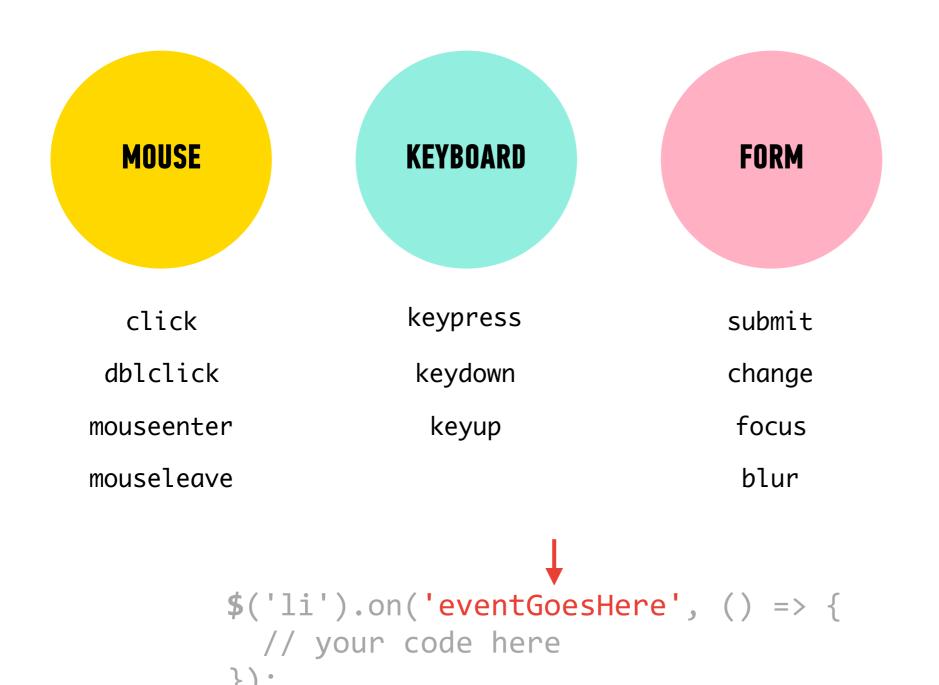


method for all events

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



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



DOCUMENT

resize

scroll

CREATE EVENT LISTENERS

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

function to run when event is triggered

CREATE EVENT LISTENERS

```
selector method for all events event

$('li').on('click', () => {
    // 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 10-jquery-statements-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 06-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

document.ready()

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

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

Shorthand version (best practice):

```
$(() => {
   // code goes here
});
```

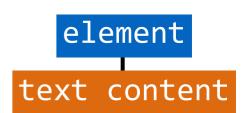
Adding content to the DOM

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

element

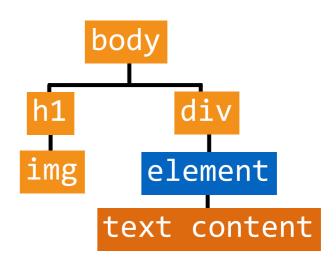
Adding content to the DOM

- 1. create a new element with
 \$('<element>')
- 2. add new content to that element with a
 method like .text(), .html(),
 or .attr()



Adding content to the DOM

- 1. create a new element with
 \$('<element>')
- 2. add new content to that element with a
 method like .text(), .html(),
 or .attr()
- 3. attach the new element to the DOM with .append()



\$('<eLement>')

Creates a new element

```
$(''); // creates an li element
```

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

```
let item1 = $('');
let item2 = $('');
```

.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 = $('').text('banana');
let item2 = $('').text('apple');
```

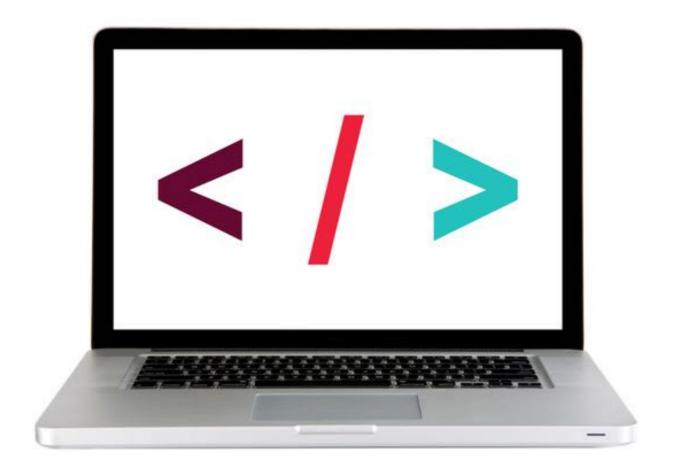
```
let item1 = $('').html('<strong>Every</strong> dinosaur');
let item2 = $('').html('Books (<em>not</em> ebooks)');
```

.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



LOCATION

starter-code > Homework-3 > create-append—homework

TIMING

until 9:20

- 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.

Exit Tickets!

(Class #7)

LEARNING OBJECTIVES - REVIEW

- Identify differences between the DOM and HTML.
- Use vanilla JavaScript methods and properties to create and modify DOM nodes.
- Create DOM event handlers using vanilla JavaScript.
- Select DOM elements and properties using jQuery.
- Manipulate the DOM by using jQuery selectors and functions.

NEXT CLASS PREVIEW

Advanced jQuery

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

