Week 10

Intro to jQuery

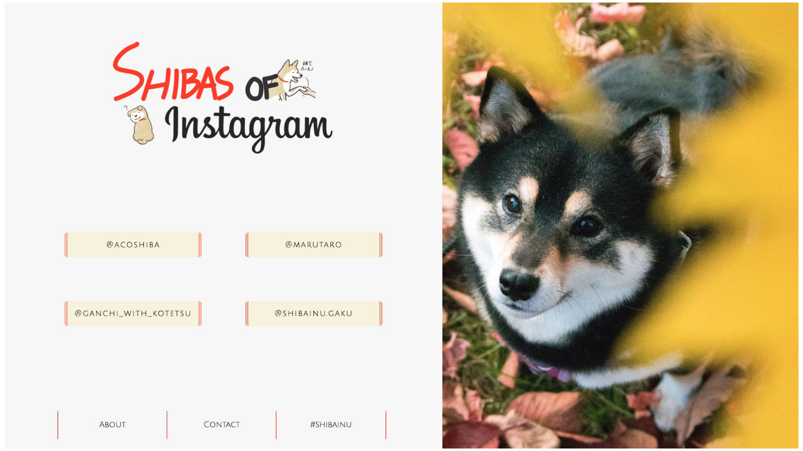
What is jQuery?

JavaScript **library** that makes **manipulating** HTML easier and more efficient

Manipulation?

What jQuery can do

Previous student work: brandonlchow.github.io/shibas_of_instagram



Library?

Barilla
PENNE
"AL DENTE" PERFECTION IN 11-12 MINUTES
ITALY'S #1 BRAND OF PASTA
ENRICHED MACARONI PRODUCT
NET WT. 1 LB (454 g)
NON GMO Project VERIFIED
Nutrition Facts

Pure JavaScript vs. jQuery

```
let el = document.getElementById("div1");

function fadeIn(el) {
  el.style.opacity = 0;
  const tick = function() {
    el.style.opacity = +el.style.opacity + 0.01;
    if (+el.style.opacity < 1) {
      (window.requestAnimationFrame &&
          requestAnimationFrame(tick)) ||
          setTimeout(tick, 16)
    }
  };
  tick();
}
fadeIn(el);
```

```
$("#div1").fadeIn()
```

tl;dr

It makes writing JavaScript easier

Questions?

Using jQuery

Linking HTML and JavaScript file

How will the HTML file know where to find its codin' ?!

Inside the `<head>` tag, add this line:

```
<script type="text/javascript"
src="assets/scripts/main.js"></script>
```

*No need to memorize this — just copy and paste or something like that

Adding the jQuery Library

How will the HTML file know where to find its jQuery ?!

Inside the `<head>` tag, add this line:

```
<script type="text/javascript"
src="http://code.jquery.com/jquery-1.11.3.min.js"></script>
```

Make sure to include this **before** your other JavaScript files

Adding the jQuery Library

How will the HTML file know where to find its jQuery ?!

Inside the `<head>` tag, add this line:

```
<script type="text/javascript"
src="http://code.jquery.com/jquery-1.11.3.min.js"></script>
```

Make sure to include this **before** your other JavaScript files

Setting up jQuery in your JavaScript file

Add this line to your JavaScript file, and write all your code within the curly braces.

```
$(document).ready(function() {
    // jQuery and JavaScript code goes in here!
});
```

Quick note about loading scripts

- Order of execution dependent on load order
  - Loading is unpredictable: HTML first or JS first? Depends.
  - Old convention: add scripts after `body` tags
- New convention: add **`defer`** attribute to script tag
  - `defer`: execute script **after** HTML document is parsed

```
<script type="text/javascript" src="assets/scripts/main.js" defer></script>
```

More on `script` attributes

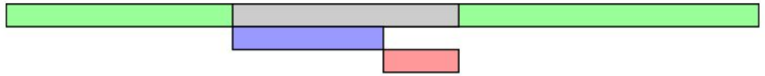
More reading on script tag attributes:
https://stackoverflow.com/questions/10808109/script-tag-async-defer

Legend

- 🟩 HTML parsing
- ⬜ HTML parsing paused
- 🟦 Script download
- 🟥 Script execution

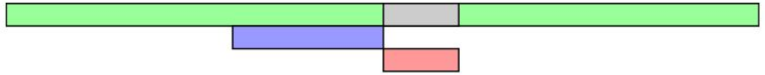
`<script>`

Let's start by defining what `<script>` without any attributes does. The HTML file will be parsed until the script file is hit, at that point parsing will stop and a request will be made to fetch the file (if it's external). The script will then be executed before parsing is resumed.



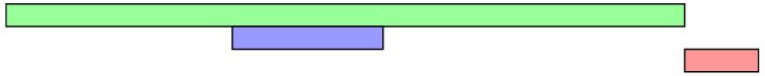
`<script async>`

`async` downloads the file during HTML parsing and will pause the HTML parser to execute it when it has finished downloading.



`<script defer>`

`defer` downloads the file during HTML parsing and will only execute it after the parser has completed. `defer` scripts are also guarenteed to execute in the order that they appear in the document.



Questions?

How we select + identify elements

Grabbing elements by id and class: HTML

HTML

```
<div id="id-name"></div>
<div
class="class-name"></div>
<div></div>
```

Grabbing elements by id and class: CSS

HTML

```html
<div id="id-name"></div>
<div
class="class-name"></div>
<div></div>
```

$\longleftrightarrow$

CSS

```css
#id-name {}
.class-name {}
div {}
```

Grabbing elements by id and class: jQuery!

HTML

```
<div id="id-name"></div>
<div
class="class-name"></div>
<div></div>
```

↔

CSS

```
#id-name {}
.class-name {}
div {}
```

↔

jQuery

```
$("#id-name")
$(".class-name")
$("div")
```

jQuery Selectors

$("   ")

```
$("            ")
```

↑

selector goes here

```
$("#id-name")
```

```
$("#id-name")
```

selects

```
<div id="id-name"></div>
```

Can also use more advanced selectors!

For whichever combination of CSS selectors, we can also use to select elements in jQuery.

```
// all dog class elements with div ancestors
$("div .dog")

// selects all elements with both class1 and
class2
$(".class1.class2")

// selects elements with either the id id-name
or the class class3 (or both!)
$("#id-name, .class3")
```

Questions?

jQuery Methods

What are methods?

“actions” objects can **execute** or **respond** to

Execute

- fadeIn
- fadeOut

Respond

- hover
- click

Dot notation

- How methods are called
- Perform actions on jQuery selectors
  - Each one does different things!

```
// when button is hovered over
$(".button").hover()

// when button is clicked
$(".button”).click()

// make image fade in
$("#image").fadeIn()

// made div fade out
$("div").fadeOut()
```

Dot notation: selected element

- This is how I select my element
- I want to use this element with the id of “image”

```
// make image fade in
$("#image").fadeIn()
```

Dot notation: action

- This is what I want the selected element to do
- In this case, I want it to fade in

```
// make image fade in
$("#image").fadeIn()
```

Dot notation: additional info

- Inside the parentheses, additional information is given to the action/method
  - What/how to do it
- In this case, take 200ms (0.2s) to fade in

```
// make image fade in
$("#image").fadeIn(200)
```

Dot notation: additional info

- For “responding” jQuery methods, the additional information will be a function to execute in response to the event

```
// when we click on the button, the
// image will fade in.
$("#button").click(function() {
    // code that should execute
    // when the button is clicked
    // on.
    $("#image").fadeIn(200)
})
```

Questions?

Setting up jQuery in your JavaScript file - revisited!

- This makes sure the HTML document is parsed and **ready** before executing jQuery code
- Step-by-step:
  - Grabs **document** with selector
  - Uses ready to method to check
  - Executes everything inside the function

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

Some useful jQuery

Name	Use
`$(".elem").click()`	Run the code below when you click on the element with class ‘elem’

`$(".elem").hover(`
`    cursor-enter, cursor-exit)`
	Run first function when your cursor enters the element and run second function when it leaves the element

`$(".elem").css("color”, "red")`	Change the color of elem to red
`$(".elem").fadeIn()`	Fade in element
`$(".elem").fadeOut()`	Fade out element
`$(".elem").show() / $(".elem").hide()`	Add/remove **`display: none`** from element

jQuery Resources

- Some good references for more information
  - http://api.jquery.com/
  - https://www.w3schools.com/jquery/

Demo

https://tinyurl.com/wdd-jquery-click

Takeaways

- It’s really easy to change the CSS of an element in jQuery
- It’s really easy to change the CSS of an element in response to **another** element in jQuery
- `$(this)` = the **specific** element from a set of elements with the same class

Demo

https://tinyurl.com/wdd-jquery-toggle

Takeaways

- It’s really easy to change the CSS of an element in jQuery
- It’s really easy to change the CSS of an element in response to **another** element in jQuery
- `$(this)` = the **specific** element from a set of elements with the same class
- It’s really easy to change the CSS of an element using **another** element in jQuery

Demo

https://tinyurl.com/wdd-jquery-shapeshift

Takeaways

- It’s really easy to change the CSS of an element in jQuery
- It’s really easy to change the CSS of an element in response to **another** element in jQuery
- `$(this)` = the **specific** element from a set of elements with the same class
- It’s really easy to change the CSS of an element using **another** element in jQuery
- Adding new “sets” of styling (that is, **classes**) to an element is really easy

Questions?