

Lab 3: Interactive Web Pages



Tricia Ngoon

slides by Michael Bernstein, Scott Klemmer, &

Philip Guo

Review questions

- `$("#project1")` is how you would select an HTML element w/ the id "project1" in jQuery
- `$(project1)` won't work. Why?
- `$(#project1)` won't work. Why?

console.log()

- A way to check if your code worked
- It's like a text message your code is sending you.
- It does NOT actually fix your bugs for you or add any sort of functionality or interactivity.

Review: CSS Properties & jQuery

Selector → `h1` {
Property → `color: red;`
}

`$(".project").click(listenerFunction);`

↑ ↑ ↑ ↑
jQuery Selected Event Action to take
shortcut element on event

CSS Selectors: **NO** dot means HTML element type

All `<p>` elements:

`$(“p”)`

```
➡ <p>Projects are below.</p>
  <div class="project" id="project1">
    ➡ <p>Waiting in line</p>
  </div>
  <div class="project" id="project2">
    ➡ <p>Needfinding</p>
  </div>
```


CSS Selectors: **NO** spaces for logical “and”

All <p> elements with class “project”:

`$(“p.project”)`

```
<p>Projects are below.</p>
<div class="project" id="project1">
  <p>Waiting in line</p>
</div>
<div class="project" id="project2">
  <p>Needfinding</p>
</div>
```


CSS Selectors: # for CSS element id

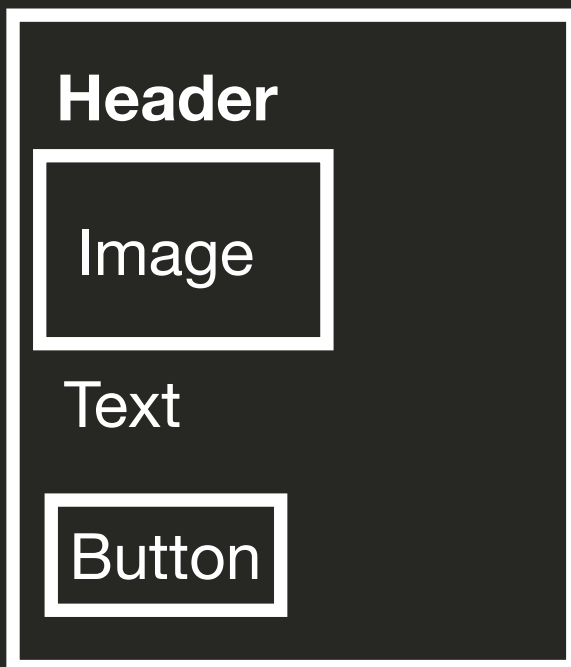
The element with id “project2”:

`$("#project2")`

```
<p>Projects are below.</p>
<div class="project" id="project1">
  <p>Waiting in line</p>
</div>
➔ <div class="project" id="project2">
  <p>Needfinding</p>
</div>
```

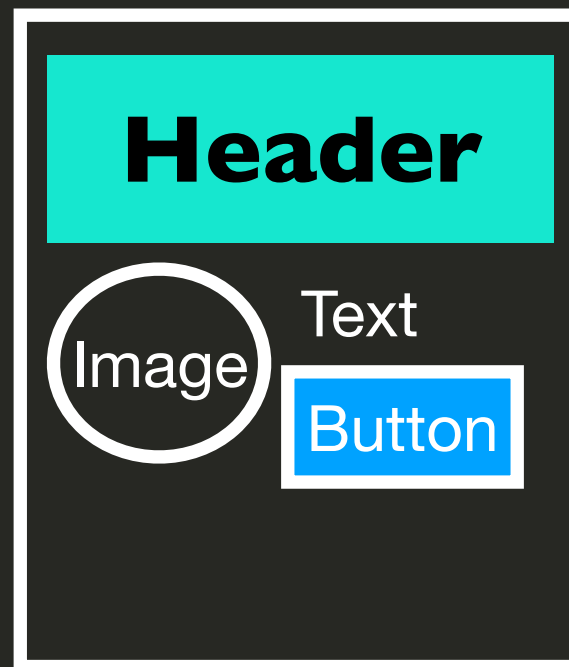
Putting it all together

Site Structure



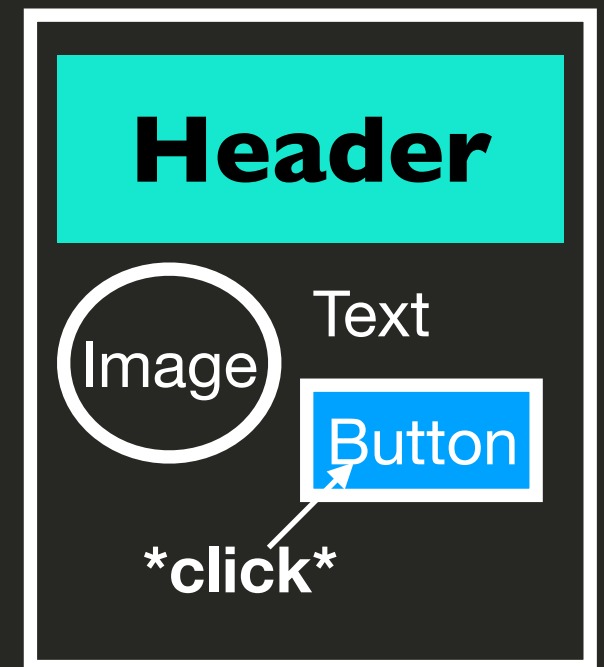
HTML <>

Site Styling



CSS {}

Site Interactivity



Javascript/
jQuery \$()

Replace instead of append

- `$("#some_selector").append("some_html")`: adds content
- `$("#some_selector").html("some_html")`: replaces content
- `$("#some_selector").text("some_html")`: replaces text
- `$("#some_selector").remove()`: gets rid of the element

Example:

```
$(".description").text("<p>New text</p>");
```

Adding a CSS class

- Select the element, then call `addClass()`

Example:

```
var description = $(".description");  
description.addClass("active");
```

```
.description.active {  
  border: 1px solid #ee3223;  
}
```

Remove, and toggle CSS classes

- You can remove classes with `removeClass()`
- Or, toggle them on and off with `toggleClass()`:

```
$(".description").click(function(e) {  
→ $(this).toggleClass("active");  
});
```

Why interactive webpages?

- We can interact with elements of a webpage without needing to reload the entire page.

Objective: Let's add some interactivity to your website

- Learn to write Javascript functions
- Hide/show your project text

Week 3 Goals

- Add navigation to your website
- Add interactivity to your projects

The console is your friend

- Add debugging statements in your functions to check if they worked

```
console.log("please work");
```

- Print some debugging statements to your console

Other jQuery events you might care about

- `$('#myelement').click(functionName)`

when the user clicks

- `$('#myelement').keypress(functionName)`

when the user types

- `$('#myelement').mouseenter(functionName)`

when the mouse enters the element

- `$('#myelement').mousemove(functionName)`

every time the mouse moves within the element

- `$('#myelement').submit(functionName)`

when a form is submitted

- <http://learn.jquery.com/events/event-basics/>

Append to an element

- `$("#some_selector").append("some_html")`

adds *some_html* to the end of the element

Example:

```
$(".description").append("<p>New text</p>");
```

Adding buttons

- Bootstrap has preset button classes for different types of buttons: <https://getbootstrap.com/docs/4.0/components/buttons/>



Copy

```
<button type="button" class="btn btn-primary">Primary</button>
<button type="button" class="btn btn-secondary">Secondary</button>
<button type="button" class="btn btn-success">Success</button>
<button type="button" class="btn btn-danger">Danger</button>
<button type="button" class="btn btn-warning">Warning</button>
<button type="button" class="btn btn-info">Info</button>
<button type="button" class="btn btn-light">Light</button>
<button type="button" class="btn btn-dark">Dark</button>

<button type="button" class="btn btn-link">Link</button>
```

Make your navigation bar

- Bootstrap has great pre-made examples of navbars:
<https://getbootstrap.com/docs/4.0/components/navbar/>
- Copy/paste a navbar example you like to your code.
- *Remember that divs are structured vertically, so your navbar should be at the top of your content in your `<body>`.

Change your navigation items

Examples:

```
<a class="nav-item nav-link href="#">About</a>
```

```
<a class="nav-item nav-link href="#">Projects</a>
```

```
<a class="nav-item nav-link href="#">Contact</a>
```

Let's make that navigation interactive

- Give each of your sections ids
 - `<div id="#about"></div>`
 - `<div id="#projects"></div>`
 - `<div id="#contact"></div>`

Let's make that navigation interactive

- Then make the hrefs of your nav items these ids:
 - `About`
 - `Projects`
 - `Contact`

What does this do?

Show which nav item is active

- What do we want to do?
 - We want the nav item we click on to have the class “active”
 - All other nav items should not have the class “active”
- *Hint: we'll use the `.toggleClass()` and `.removeClass()` events

Hide/show project info

- You can show and hide content on your page using JavaScript as well
- In your projectClick() function, add this line:

```
$(this).parent().next().toggle();
```

Week 3 Goals

- Add a navbar to your website
- Add interactivity to your projects

Commit and Push all your
work to Github, then
submit Assignment 3 at:

<https://tinyurl.com/dsgn90-a3>