LEARNING OBJECTIVES

‣ Gain an overview of the JavaScript landscape and its placement in the web ecosystem.

‣ Practice programmatic thinking by writing pseudo-code.

‣ Write expressions that both assign and evaluate variables.

‣ Explain the difference between jQuery and vanilla JS.

‣ Register and trigger event handlers for jQuery events.

INSTRUCTOR TIPS

- ‣ There's a lot to fit into this class! Don't worry about diving too deep—this should be an introductory lesson.

- ‣ Keep track of time, so you can get through all the content.

- ‣ Be aware of differing skill levels and try to differentiate as needed.

MATERIALS

‣ Laptop
‣ Google Chrome
‣ Sublime Text or Atom

PRE-WORK

‣ DASH Project 1
‣ Install either Sublime Text 3 or Atom on to computer (choose one before telling students which one to install).

‣ Write learning objectives on board.

‣ Have a way to distribute starter code for code challenges at the beginning of class.

‣ Make sure wifi network and password (GA Guest, yellowpencil) is written on board, since students will need to be online to use codepen.io.

JAVASCRIPT 101

Brett Haymaker

Web Developer, c|change inc

GREETINGS + THE GA EXEPERIENCE

LEARNING OBJECTIVES

‣ Gain an overview of the JavaScript landscape and its placement in the web ecosystem.

‣ Practice programmatic thinking by writing pseudo-code.

‣ Write expressions that both assign and evaluate variables.

‣ Explain the difference between jQuery and vanilla JS.

‣ Register and trigger event handlers for jQuery events.

WEB PRODUCTION WORKFLOW

WEB PRODUCTION WORKFLOW

UX → Design → Front End → Back End → LAUNCH!!

- ‣ Take the design and turn it into code and assets
- ‣ Uses HTML/CSS to create the structure for a page and add styles
- ‣ Add interactions with JavaScript
- ‣ Other responsibilities: accessibility, performance, cross-browser and cross-device functionality
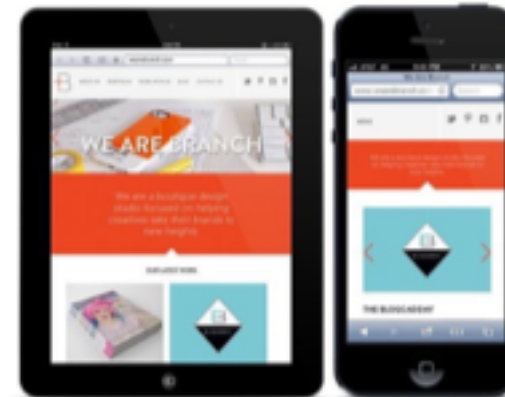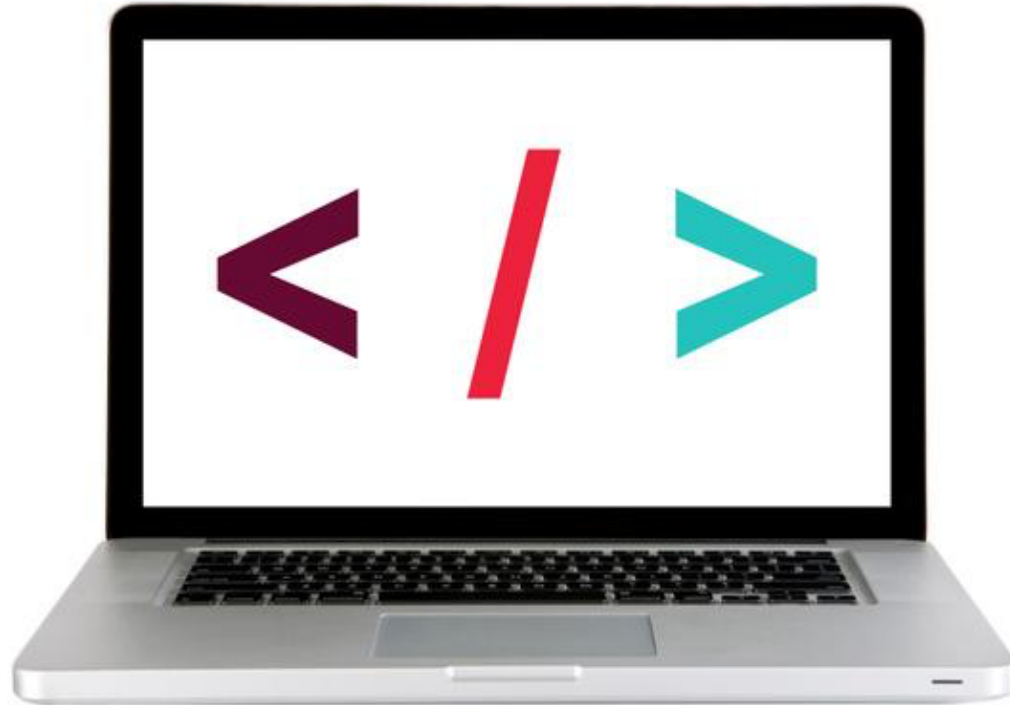
WHAT DOES FRONT END EVEN MEAN?

"A mix of programming and layout that powers the <u>visuals</u> and <u>interactions</u> of the web."

FRONT END IS MADE UP OF…

CSS **HTML** **JavaScript**

LET'S TAKE A LOOK

http://egwineco.com/

IN SUMMARY

- The Front End is what the user sees.

- It powers the visuals and interactions of the web.

- It is meant to be pretty, but doesn't always happen that way.

- Made up of HTML, CSS and Javascript.

WHAT IS JAVASCRIPT?

JAVASCRIPT IS RESPONSIBLE FOR...

CONTENT BEHAVIOR

INTERACTIONS

ANIMATIONS

WITH JAVASCRIPT YOU CAN...

1 Access Content

2 Modify Content

3 Program Rules

14 React to Events

MANIPULATE THE DOM

PRACTICE READING JAVASCRIPT

DIRECTIONS

1. Let's visit this codepen together
2. Turn to someone next to you and as a team try to figure out on a high level what is happening
3. With your partner try to make it so that the slow button changes the bulb to yellow
4. With your partner try to make it so that the go button works
5. This exercise is simply to help introduce you to what JavaScript looks like, do not get caught up in all the details just yet

EXERCISE

JAVASCRIPT 101

INTRO TO PSEUDO CODE

THE IMPORTANCE OF PLANNING

PSEUDO CODE IS...

‣ A way to 'plan out' your program before coding it.

‣ A detailed yet readable description of what a computer program must do.

‣ Expressed in plain english.

PSEUDO CODE — EXAMPLE

A program that lets a player know whether he or she has passed the current level.

```
passingScore = 50 points
get playerScore


if playerScore >= passingScore
    display message "Current level: Passed"
otherwise
    display message "Current level: Failed"
```

PSEUDO CODE — THERMOSTAT

DIRECTIONS

1. Write pseudo code for an application that would monitor the room temperature and adjust it so the room remains at a certain temperature.

EXERCISE

VARIABLES + DATA TYPES

WHAT ARE VARIABLES?

‣ We can tell our program to remember (store) values for us to use later on.

‣ The 'container' we use to store the value is called a variable.

‣ A variable has a name and a value.

‣ The value can change.
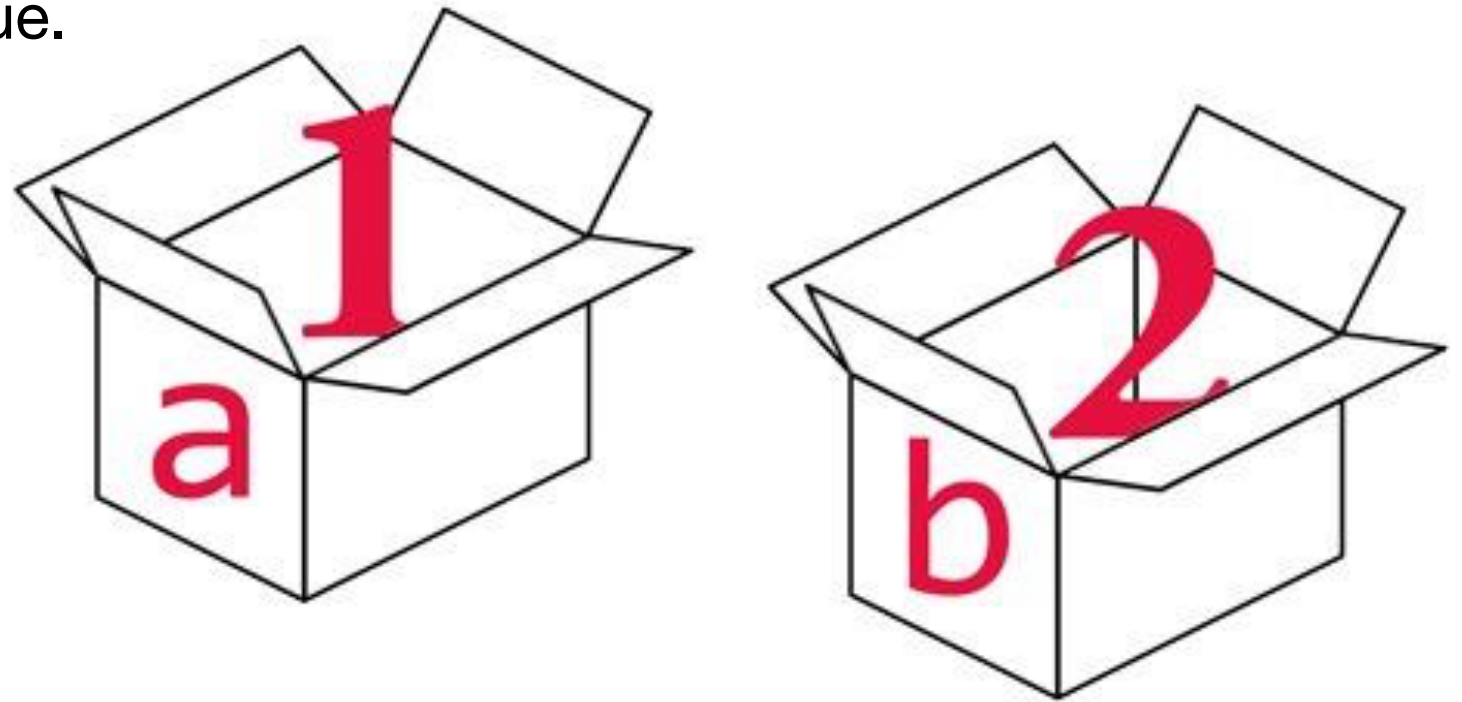
var age = 29;

VARIABLES

‣ **Reassigning a variable** means you are changing the value you set initially

‣ How is this done?:

$$var\ age = 29;$$
$$age = 30;$$

USING THE CONSOLE

EXERCISE

DIRECTIONS

1. We'll be using the console to practice creating variables. It's where JavaScript is interpreted and run. You can use it to practice writing JavaScript!
2. Open up Google Chrome
3. Right click and go to "inspect"
4. Select "console"
5. Follow along!

WHAT CAN BE STORED IN VARIABLES?

DATA TYPES:

STRINGS

"Today is Monday"

Letters and other characters enclosed in quotes

NUMBERS

10   22.75

‣ Positive numbers
‣ Negative numbers
‣ Decimals

BOOLEANS

true   false

Can have one of two values:

‣ True
‣ False

ARITHMETIC OPERATORS

NAME:	OPERATOR:	EXAMPLE:	RESULT:
ADDITION	+	2 + 4	6
SUBTRACTION	-	8 - 1	7
MULTIPLICATION	*	2 * 3	6
DIVISION	/	4 / 2	2

ASSIGNMENT OPERATORS

	INITIAL VALUE:	OPERATOR:	EXAMPLE:	RESULT:
ASSIGN VALUE TO VARIABLE	var num = 8	=	num = 6	6
ADD VALUE TO VARIABLE	var num = 8	+=	num += 6	14
SUBTRACT VALUE FROM VARIABLE	var num = 8	-=	num -= 6	2

GUIDED/INDEPENDENT PRACTICE: VARIABLES

EXERCISE

DIRECTIONS

1. Open up the project called "variables-code-along" in Sublime Text. I will walk you through how to do this
2. Go to the file called main.js
3. We will do Part 1 together
4. You will do Part 2 on your own

JAVASCRIPT IN ACTION

METHODS AND PROPERTIES OF STRINGS

MAKE STRING LOWERCASE:

```
var str = "Hello World";
var res = str.toLowerCase();
// the result of res will be:
// hello world
```

LENGTH OF A STRING (PROPERTY):

```
var str = "Hello World";
var n = str.length;
// the result of n will be 11
```

MAKE STRING UPPERCASE:

```
var str = "Hello World";
var res = str.toUpperCase();
// the result of res will be:
// HELLO WORLD
```

**Find a whole list of methods and properties for strings here

STRING CONCATENATION

‣ To take two strings and stick them together, use the + operator.
‣ This is called string concatenation.

```
var book = "Happy";
var summary = "Best book ever.";
var review = book + ": " + summary;
// Result will be: Happy: Best book ever.
```

JQUERY: A JS LIBRARY

WHAT IS JQUERY?

‣ jQuery is an open-source project that was released in 2006, and it's currently the most widely used JavaScript library on the web.

‣ jQuery is a JavaScript file you include in your pages.

‣ Makes it faster and easier to write cross-browser JavaScript.

‣ Allows us to find elements using CSS-style selectors and then do something to them using jQuery methods.

JQUERY VS. JAVASCRIPT

JS:

```
document.getElementById('heading').innerHTML = "Your Name";
```

JQUERY:

```
$('#heading').html('Your Name');
```

Vanilla JavaScript (or plain JavaScript) is like driving a car with a manual transmission.

Using a library such as jQuery, that is comprised of plain JavaScript, gives us syntactic shortcuts. It's the equivalent of driving a car with an automatic transmission. We don't have to worry about as many things, and we can do more with less.

**Personal anecdote about driving a manual transmission car. Mi scusi.

JQUERY VS. JAVASCRIPT

‣ jQuery allows us to use the CSS-style selectors that we know and love!

JS: **JQUERY:**

```
document.getElementsByTagName('body')[0]
```
→
```
$('body')
```

```
document.getElementById('about')
```
→
```
$('#about')
```

LOADING JQUERY

‣ jQuery is a javascript file, but before you start using jQuery, you'll need to include it in our html page. Here are two ways to get jQuery.

‣ 1. Download the compressed version [online](https://jquery.com/download/).

‣ 2. Alternatively, you can link to the jQuery file in the <script> tag in your HTML like so:

```html
<head>
<script src="https://code.jquery.com/jquery-3.1.0.min.js"></script>
</head>
```
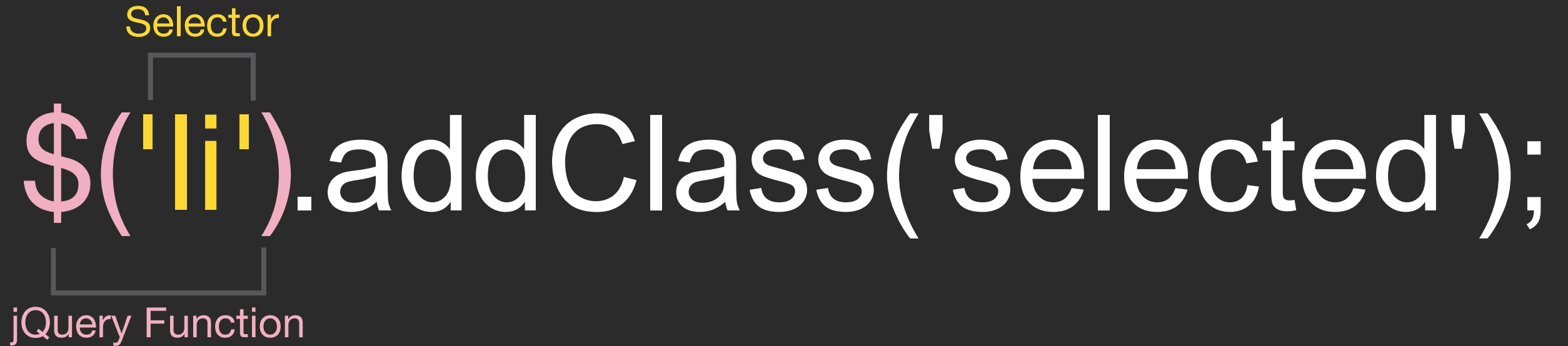
JAVASCRIPT 101

HOW DOES jQUERY WORK?

USING JQUERY TO MANIPULATE THE DOM

1
Select an element/elements

2
Work with those elements

Selector

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

jQuery Function

jQuery Function:

‣ Lets us find one or more elements in the page
‣ Creates a jQuery object which holds references to those elements

JQUERY OBJECTS — FINDING ELEMENTS: SOME EXAMPLES

‣ You can use your CSS-style selectors!!!

SELECTOR:	CSS:	JQUERY:
CLASS	.className	$('.className')
ID	#idName	$('#idName')
MULTIPLE SELECTORS	h1, h2, h3	$('h1, h2, h3')
DESCENDANT	li a	$('li a')

& tons more!!!

USING JQUERY TO MANIPULATE THE DOM
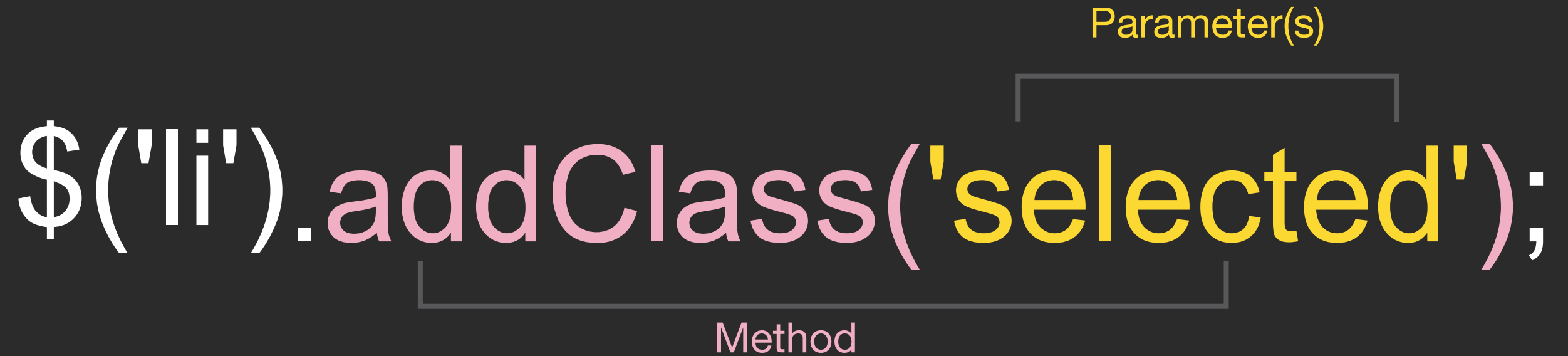
1

Select an element/elements

2

Work with those elements

Parameter(s)

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

Method

JQUERY METHODS — WORKING WITH THOSE ELEMENTS

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

GET/SET CONTENT

ADD EFFECTS / ANIMATION

CREATE EVENT LISTENERS
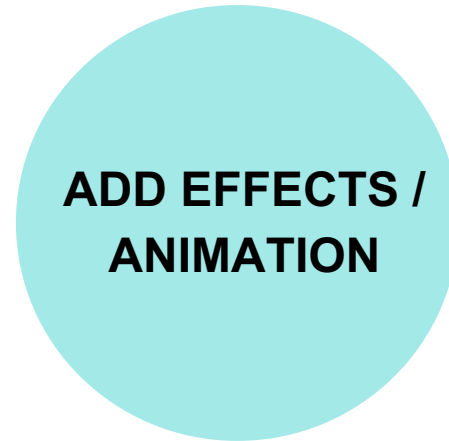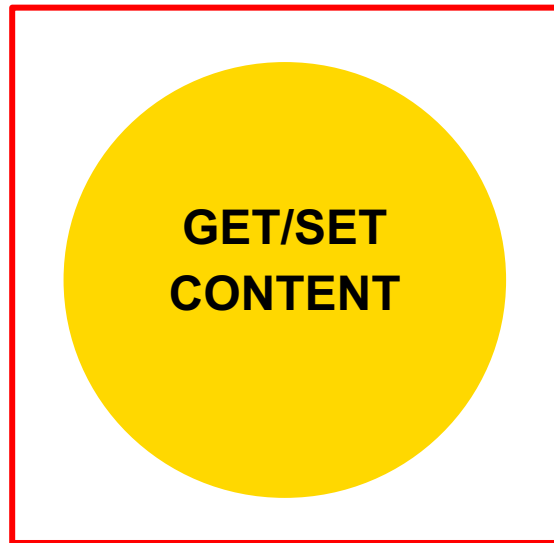
Refer to the jQuery docs for list

JQUERY METHODS — GETTING/SETTING CONTENT

Get/change content of elements, attributes, text nodes

Some methods available to us:
- **.html( )**
- **.attr( )**
- **.css( )**
- **.addClass( )**
- **.removeClass( )**
- **.toggleClass( )**

What goes in the parentheses?
The html, styles, classes you want to add/change

Examples of adding/changing content:

```
$('h1').html('Content to insert goes here');
$('img').attr('src', 'images/bike.png');
$('#box1').css('color', 'red');
$('p').addClass('success');
$('p').removeClass('my-class-here');
```

JQUERY CODEALONG PART 1

EXERCISE

KEY OBJECTIVE

‣ Utilize jQuery to access and manipulate DOM elements.

TYPE OF EXERCISE

‣ Individual/Partner

AS A CLASS

Exercise is in starter_code > jquery_code_along

1. Follow the instructions under part 1 in main.js

JQUERY METHODS — WORKING WITH THOSE ELEMENTS

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

GET/SET CONTENT

ADD EFFECTS / ANIMATION

CREATE EVENT LISTENERS

Refer to the jQuery docs for list

JQUERY METHODS — EFFECTS/ANIMATION

Add effects and animation to parts of the page

Some methods available to us:
- **.show( )**
- **.hide( )**
- **.fadeIn( )**
- **.fadeOut( )**
- **.slideUp( )**
- **.slideDown( )**
- **.slideToggle( )**

What goes in the parenthesis?
An animation speed

Examples:

```
$('h1').fadeOut(200);
$('#box1').slideDown('slow');
$('h1').fadeIn();
```

JQUERY METHODS — WORKING WITH THOSE ELEMENTS

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

GET/SET CONTENT

ADD EFFECTS / ANIMATION

CREATE EVENT LISTENERS

Refer to the jQuery docs for list

JQUERY METHODS — EVENTS!

The .on() method is used to handle all events.

Syntax: $('selector').on('event', code_that_should_run);

Example:

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

JQUERY METHODS — EVENTS!

Some events that .on() deals with:

- ‣ UI: focus, blur, change
- ‣ Keyboard: keydown, keyup
- ‣ Mouse: click, mouseup, mousedown, mouseover
- ‣ Form: submit
- ‣ Browser: resize, scroll

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

JQUERY CODEALONG PART 2

EXERCISE

KEY OBJECTIVE

‣ Utilize jQuery to access and manipulate DOM elements.

TYPE OF EXERCISE

‣ Individual/Partner

AS A CLASS

Exercise is in starter_code > jquery_code_along

1. Follow the instructions under Part 2 in main.js

IF

CONDITIONALS

CONDITIONAL LOGIC

If something is true, do one thing. If it is not, do something else. This type of logic or statement is a condition.

In JavaScript (and coding in general) you'll need to make comparisons all the time:

‣ Is a user logged in?

‣ Has the user chosen three or more colors?

‣ Is the password correct?

‣ Does a user have enough money in their bank account?

‣ etc.

JAVASCRIPT — COMPARISON OPERATORS

>= Greater than or equal to

<= Less than or equal to

> Greater than

< Less than

Equal to ===

Not equal to !==

ASSIGNMENT VS. COMPARISON — DON'T GET THEM CONFUSED!

ASSIGNMENT

=

var number = 7;

COMPARISON

===

or

==

if (number === 8) {
// Do something
}

IF STATEMENTS

```javascript
if (age > 65) {
    $('h1').html("Senior Discount Applied");
}
```

IF STATEMENTS

```
if (age > 65) {
    $('h1').html("Senior Discount Applied");

} else {
    $('h1').html("Sorry, you do not qualify for a discount.");
}
```

```javascript
if (answer === 38) {
    // Do something if first condition is true
} else if (answer === 30) {
    // Do something second condition is true
} else {
    // Do something if all above conditions are false
}
```

IF STATEMENTS

```
if (age > 65) {
    $('h1').html("Senior Discount Applied");

} else if (age < 18) {
    $('h1').html("Student Discount Applied");

} else {
        $('h1').html("Sorry, you don't qualify for a discount");
}
```

JAVASCRIPT — LOGICAL OPERATORS

& and

|| or

! not

MULTIPLE CONDITIONS

```
if (name == "GA"  && password == "YellowPencil"){
    //Allow access to internet
}
```

WRAP-UP

SO, WHO NEEDS TO KNOW ABOUT THIS STUFF?

WRAP UP

Business
Managers

Product
Managers

Designers

Marketers

Programmers

JAVASCRIPT 101

HOW TO LEARN THIS AT GA?

PART-TIME:

JS/FEWD

ONLINE:

HCD

FULL-TIME:

WDI / WDIR

REFERENCE PAGES:

- **Mozilla Developer Network**
- W3Schools
- CSS Tricks
- Web Field Manual
- Free IT Books
- JSDB.io
- Site Point
- Codrops

STAYING CURRENT

- Smashing Magazine
- Sidebar.io
- Codrops
- Medium
- A List Apart
- Web Designer Weekly
- Creative Bloq
- CSS Tricks
- Codepen.io

SITE INSPIRATION

- Awwwards
- Site Inspire
- Mediaqueri.es
- One Page Love
- Little Big Details
- The Best Designs
- CSS Design Awards
- Web Design File
- Behance
- Dribbble
- Responsive Patterns

A COUPLE OF LAST THINGS

You are going to get lots of errors…

Google first, ask questions later (but don't be afraid to ask)
Stack Overflow is great for this stuff!

Get immersed! Coding is hard if you don't get really involved.

AFTER CLASS YOU CAN...

DIRECTIONS

EXERCISE

1. Go to dash.ga.co
2. The last two projects cover what we went over tonight

Q & A

THANKS!

Brett Haymaker

‣ Contact Information:

‣ brett.haymaker@gmail.com

‣ brett-haymaker.com

‣ @BrettHaymaker