1) HAVE SLACK OPEN

2) HEY! CODEPEN!

SNACKS & DESIGN

TODAY!

ALLISON SCHAFFER & KEVIN SELLA

(GOOGLE SHEET IS PINNED IN SLACK)

Q & A

“What's some of the most advanced stuff that you can do/have done with Javascript? “

http://threejs.org/examples

“How much can you do with javascript before needing to introduce back-end stuff?”

“How do CSS animations work?”

“What is something that is most commonly attributed to Javascript when it shouldn't be? (Like, should actually be done in CSS or HTML)”

“As a graphic designer I have a lot of quick practices I do to stay in tune with the practice, example being once a week I spend an hour max redesigning popular logos. What are some potential options in the same vain for coding. “

“To what extent do/can jQuery and Node.js interact?”

“For our final projects,
would you recommend we use jQuery or Vanilla JS? “

“I'm still trying to figure out the order or composition of larger, more complex javascript.”

*“Convention for using spaces within functions and variables in JS...? (e.g. "(a * (b + c)" vs. “(a*(b+c)")"*

“What situations would I need to use Javascript instead of jQuery in?”

GENERAL ASSEMBLY

JS – VARIABLES AND CONDITIONALS

Eric Boyer

COMING UP...

MARCH 15: VARIABLES AND CONDITIONALS

MARCH 17: FUNCTIONS AND ARRAYS

REVIEW

USING JQUERY TO MANIPULATE THE DOM

1

Select an element/elements

2

Work with those elements

JQUERY — SELECTING ELEMENTS

jQuery Function:

- Lets us find one or more elements in the page
- Creates a *jQuery object* which holds references to those elements
- We'll be using the shorthand in this class: $( )
- $(selector) is the same as jQuery(selector)

USING JQUERY TO MANIPULATE THE DOM

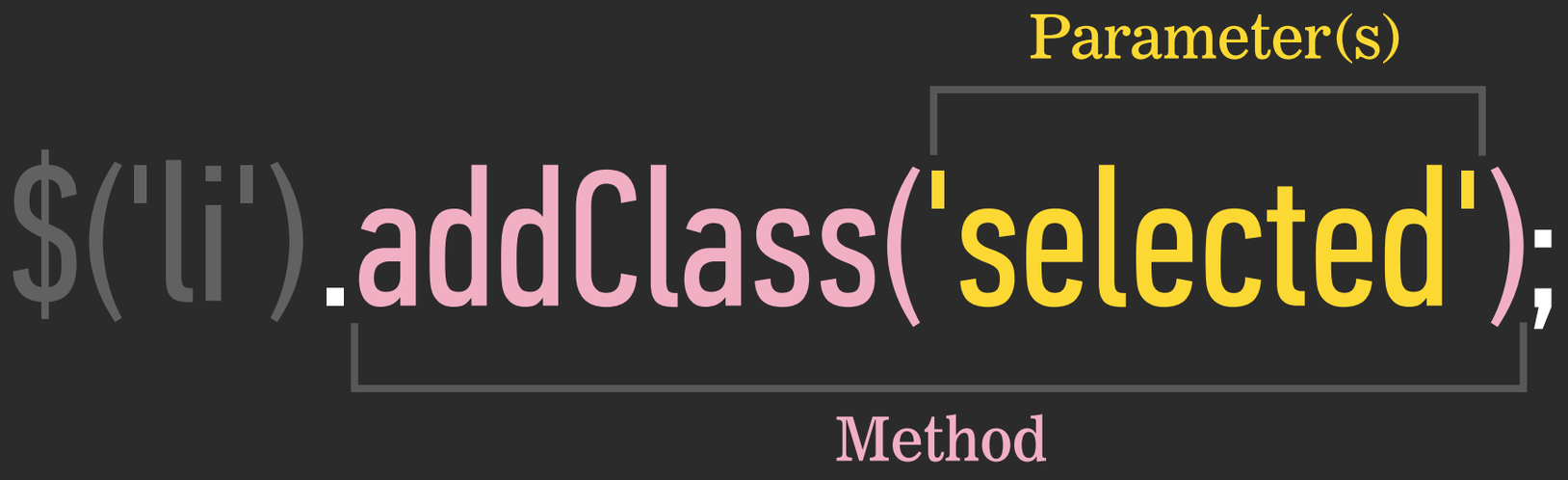
1

Select an element/elements

2

Work with those elements

JQUERY — WORKING WITH THOSE ELEMENTS



JQUERY METHODS — TRAVERSING THE DOM

TRAVERSE THE DOM

jQuery provides us with methods to find/select elements to work with & traverse the DOM

Some methods available to us:

- .find( ) (finds all descendants)
- .children( )
- .parent( )
- .siblings( )

What goes in the parentheses?
A css-style **selector**

Examples:

```
$('h1').find('a');
$('#box1').parent();
$('p').siblings('.important');
```

**Think of this as part of the selection process, must come directly after another selection*

JQUERY METHODS — GETTING/SETTING CONTENT

GET/SET 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');
```

ADD CLASS

```
$('h1').addClass('.fun')
```

REMEMBER — NO PERIOD!!

```
$('h1').addClass('fun')
```

JQUERY METHODS — EFFECTS/ANIMATION

ADD 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 — EVENTS!

CREATE EVENT LISTENERS

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!

CREATE EVENT LISTENERS

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 — REVIEW

EXERCISE

KEY OBJECTIVE

- Review jQuery selectors and events, get practice looking up new methods

TYPE OF EXERCISE

- Individual/paired

SMALL GROUP PLANNING

8 min

1. Follow the instructions in lesson8_starter_code > [0] jquery_review > js/main.js

SOME NEW METHODS!

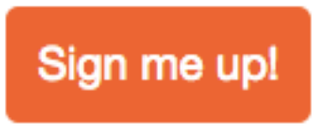
INPUT AND BUTTON ELEMENTS

TEXT INPUT ELEMENT

```
<input type="text" placeholder="Enter your name">
```

BUTTON ELEMENT

```
<button type="button">Sign me up!</button>
```



SOME NEW METHODS!

GET/SET CONTENT

Get/change content of elements, attributes, text nodes (part 2!)

Some new methods

- .html( )
- .append( )
- .prepend( )
- .val( )

What goes in the parentheses?
The **html or content you want to add/change**

Examples of **adding/changing** content:

```
$('h1').html('Summer in Chicago rocks!');
$('.box').html('<h1>Best box ever!</h1>');
$('ul').prepend('<li>First list item</li>');
$('ul').append('<li>Last list item</li>');
$('input').val();
```

LEARNING OBJECTIVES

- Define variables and identify best cases to use them.
- Differentiate between strings, integers and floats.
- Apply conditionals to change the program's control flow

AGENDA

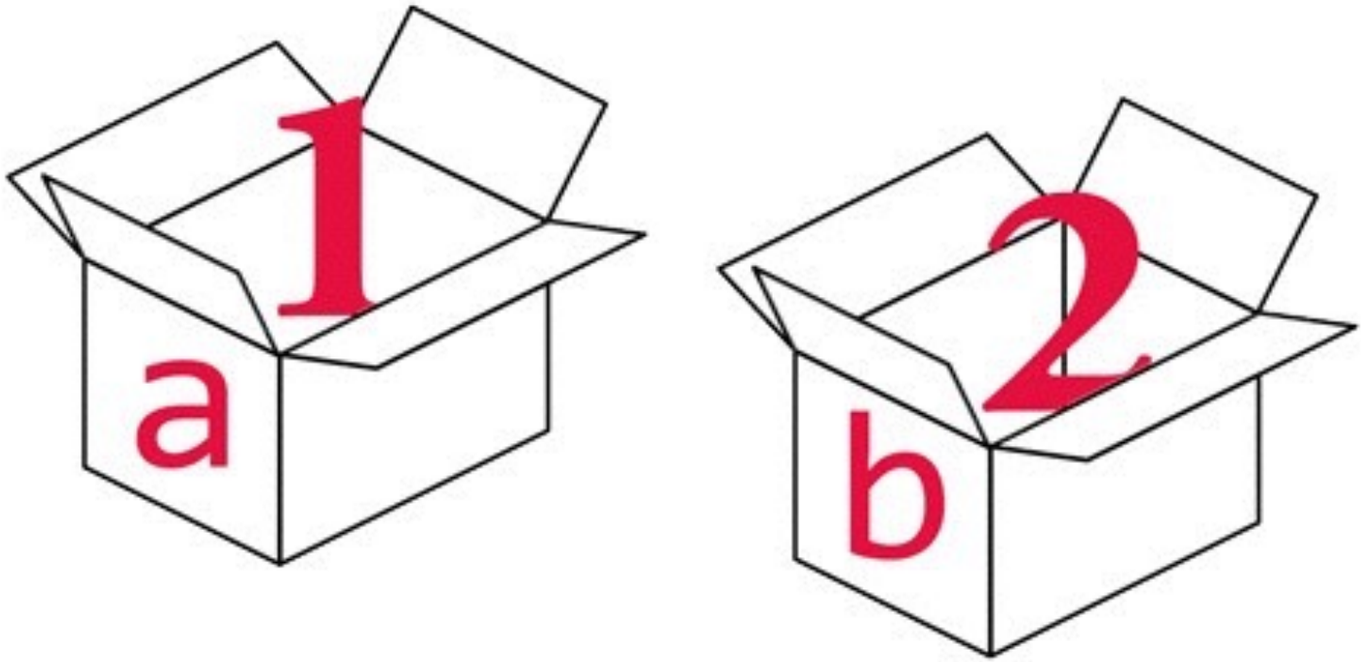
- Variables
- Data Types
- Conditionals
- Lab — Temperature Converter

VARIABLES

WHAT ARE VARIABLES?

WHAT ARE VARIABLES?

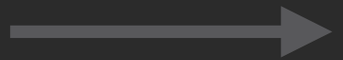
- We can tell our program to remember (store) values for us to use later on.
- The 'container' we use to store the value is called a **variable**



SYNTAX

JAVASCRIPT — VARIABLES

Declaring a variable



```
var age;
```

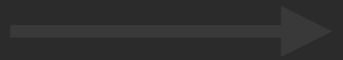
Keyword Name

Assigning

```
age
```

Name Value

Both in one step



```
var
```

JAVASCRIPT — VARIABLES

Declaring

```
var
```

Keyword Name

Assigning a variable

```
age = 29;
```

Name Value

Both in one step

```
var
```

JAVASCRIPT — VARIABLES

Declaring → `var`

Keyword: `var` — Name: `age`

Assigning → `age`

Name: `age` — Value: `29`

Both in one step → `var age = 29;`

JAVASCRIPT — VARIABLES

Declaring a variable → `var age;` ← Semicolon!

Assigning a variable → `age = 29;` ← Semicolon!

Both in one step → `var age = 29;` ← Semicolon!

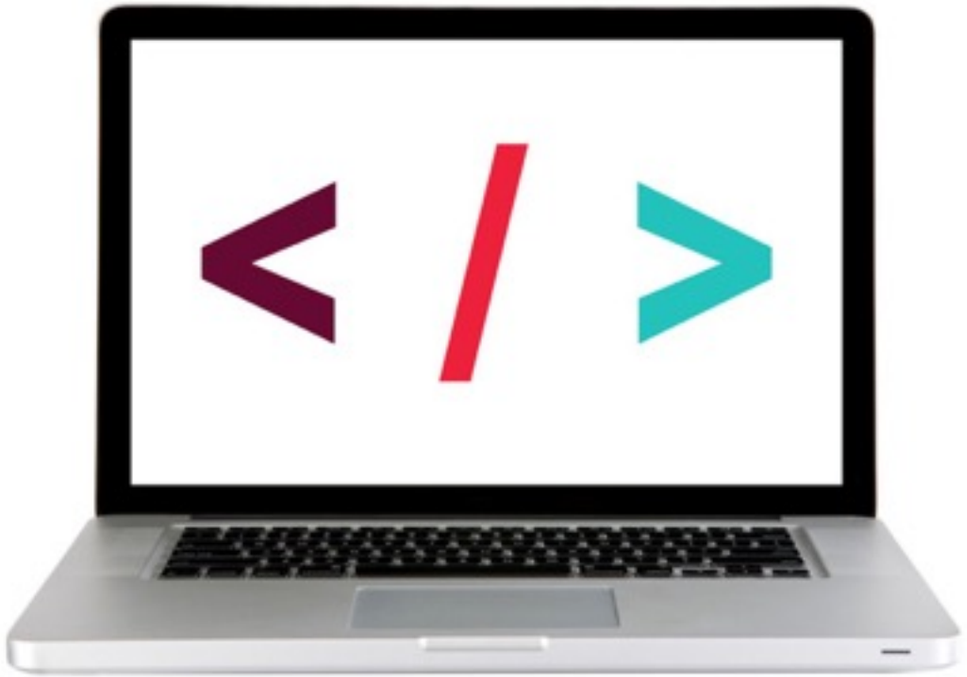
JAVASCRIPT — VARIABLE RE-ASSIGNMENT

```
var champion = “Eric”;
champion = “Adriana”;
```

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

CODE ALONG — SCORE KEEPER



Score Keeper (Codepen)

RULES

VARIABLE CONVENTIONS

1. Variables start with a **lowercase** letter


```
var numberOfStudents = 10;
```

```
var NumberOfStudents = 10;
```

2. If they contain multiple words, subsequent words start with an upper case letter.


```
var firstName = "Eric";
```

```
var firstname = "Eric";
```

```
var first name = "Eric";
```

3. Names can only contain: letters, numbers, $ and _ (no dashes - or periods .)


```
var number1 = 5.5;
```

```
var number-1 = 10;
```

```
var number.1 = 10;
```

VARIABLE CONVENTIONS

3. Variables cannot start with a number


```
var number1 = 10;
```

```
var 1number = 10;
```

4. Case sensitive - numberofstudents is not the same as numberOfStudents

5. Names should be descriptive


```
var lastName = "Boyer";
```

```
var x = "Boyer";
```

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

** Note: we'll meet some more data types later on down the road, too!*

TO SUMMARIZE

1. A variable has both a “name” and a “value”
2. That value can change
3. A variable can be used multiple times throughout the code

ORDER IS IMPORTANT!!!

```
var name = "Matt";
```


~~"Matt" = var name;~~

WRONG!!!!!

CODE ALONG — VARIABLES PT. 1

lesson8_starter_code > [1] variables

LAB — TEMP CONVERTER — PART 2

EXERCISE

KEY OBJECTIVE

- Practice declaring and assigning variables

TYPE OF EXERCISE

- Individual/paired

LOCATION

- lesson8_starter_code > [1] variables

EXECUTION

6 min 1. Follow the instructions under Part 2

DATA TYPES

NUMBERS

MORE ABOUT NUMBERS

INTEGERS:

Integers are whole numbers `10`

FLOATS:

Number that uses a decimal to represent a fraction `22.75`

**Can perform arithmetic on number data types*

ARITHMETIC OPERATORS

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

STRINGS

MORE ABOUT STRINGS

A STRING:

- Stores textual information
- Is surrounded by quotes

```
"How is the weather today?"
```

```
'Cold'
```

STRINGS

DOUBLE QUOTES VS. SINGLE QUOTES

```
"It's a beautiful day"
```

```
'They "purchased" it'
```


ESCAPING

```
'It\'s a beautiful day'
```

```
"They \"purchased\" it"
```

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.
```

BOOLEANS

BOOLEANS

Can have one of two values: `true` `false`

CONVERTING DATA TYPES

DATA TYPE CONVERSION

STRING TO INTEGER:

```
var intString = "4";
var intNumber = parseInt(intString, 10);
```

STRING TO FLOAT:

```
var floatString = "3.14159";
var floatNumber = parseFloat(floatString);
```

NUMBER TO STRING

```
var number = 4;
number.toString(); => "4";
```

CODE ALONG — SCORE KEEPER

Let's code! Score Keeper (Codepen)

CONDITIONALS

WHAT ARE CONDITIONALS?

IF STATEMENTS

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.

COMPARISON OPERATORS

JAVASCRIPT — COMPARISON OPERATORS

`>=`	Greater than or equal to	`===`	Equal to
`<=`	Less than or equal to	`!==`	Not equal to
`>`	Greater than		
`<`	Less than		

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

ASSIGNMENT

```
var number = 7;
```

COMPARISON

or


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

IF STATEMENTS

JAVASCRIPT — IF STATEMENT

Condition

```
if (answer === 38) {
    // Do something if true
}
```

IF STATEMENTS

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

JAVASCRIPT — IF/ELSE STATEMENT

```
if (answer === 38) {
    // Do something if true
} else {
    // Do something if false
}
```

IF STATEMENTS

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

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

JAVASCRIPT — IF/ELSE IF/ELSE

```
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");
}
```

CODE ALONG — SCORE KEEPER

Let's code! lesson8_starter_code > [2] conditionals

LOGICAL OPERATORS

JAVASCRIPT — LOGICAL OPERATORS

&& and

|| or

! not

MULTIPLE CONDITIONS

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

EXERCISE — BLACKOUT

KEY OBJECTIVE

- Practice combining conditionals with jQuery to create a simple interaction.

TYPE OF EXERCISE

- Individual/Paired

EXECUTION

5 min	1. lesson8_starter_code > [3] blackout
8 min	2. Together: write pseudo code for the light/dark switcher.
	3. In pairs, write code to complete the light/dark switcher.

CODE ALONG — COMPARE TWO NUMBERS

Let's code! Compare Two Numbers (Codepen)

LAB

LAB — TEMP CONVERTER

LAB — TEMP CONVERTER — PART 1

EXERCISE

KEY OBJECTIVE

- Build an application using HTML/CSS and JS that converts a temperature from Fahrenheit to Celsius

TYPE OF EXERCISE

- Groups of 3-4

SMALL GROUP PLANNING

Until 8:45

1. In groups of 3-4 test out the functional temperature converter and write pseudo code to convert a temperature from Fahrenheit to Celsius

LAB — TEMP CONVERTER — PART 2 (NEXT CLASS)

EXERCISE

KEY OBJECTIVE

- Build an application using HTML/CSS and JS that converts a temperature from Fahrenheit to Celsius

EXECUTION

Until 8:50

1. Write .js to make the temperature converter functional.
2. **Bonus #1**: Change the background-color depending on what temperature the user enters
3. **Bonus #2**: Add error styles if the user doesn't enter a value in the form
4. **Bonus #3**: Add your own styles to the temperature converter

***For reference, see the Compare Two Numbers and Score Keeper*

LAB — TEMP CONVERTER — FORMULAS

Formula to convert fahrenheit to celsius: `(fahrenheit - 32) / 1.8;`

Formula to convert celsius to fahrenheit: `1.8 * celsius + 32;`

JQUERY METHODS — EVENTS!

CREATE EVENT LISTENERS

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
});
```

FINAL PROJECT ROADMAP

~~SUNDAY MARCH 13 - PROPOSALS DUE~~

TUESDAY MARCH 15
THURSDAY MARCH 17 - PROPOSAL DISCUSSIONS W/ ERIC

***

SUNDAY MARCH 27 - WIREFRAMES DUE

FINAL PROJECT MILESTONE 2: PROPOSAL CHECK-INS W/ ERIC

TUESDAY MARCH 15TH

Kimberly Baird
Tom Bunting
Michael Fischer
Jim Howes
Jon Iler
Peyton Lee
Blanca Leon-Carter
Anna Matras
Christopher Zalek

LEARNING OBJECTIVES

- Define variables and identify best cases to use them.
- Differentiate between strings, integers and floats.
- Apply conditionals to change the program's control flow.

SNACKS & DESIGN

THURSDAY MAR 17TH

MICHAEL SENA & NICHOLAS SKEBA

(GOOGLE SHEET IS PINNED IN SLACK)

EXIT TICKETS