LEARNING OBJECTIVES

- Define variables and identify best cases to use them.
- Describe strings, numbers, and boolean variable types.
- Use comparison operators to evaluate and compare statements.
- Apply conditionals to change the program's control flow
- Create Functions

AGENDA

- Review
- Variables
- Data Types
- Conditionals
- Lab

JS BASICS

REVIEW

USING JQUERY TO MANIPULATE THE DOM

1

Select an element/elements

2

Work with those elements

JQUERY — SELECTING ELEMENTS

Selector

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

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

Parameter(s)

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

Method

JQUERY METHODS — TRAVERSING THE DOM

TRAVERSE THE DOM

- These methods to find/select elements to work with & traverse the DOM
- 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()`	`$('p').siblings('.important');`
`.children()`	`$('ul').children('li');`

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

JQUERY METHODS — GETTING/SETTING CONTENT

GET/SET CONTENT

Get/change content of elements and attributes

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

What goes in the parentheses?
The **html, styles, classes** you want to change.

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

METHODS	EXAMPLES
`.show()`	`$('h1').show();`
`.hide()`	`$('ul').hide();`
`.fadeIn()`	`$('h1').fadeIn(300);`
`.fadeOut()`	`$('.special').fadeOut('fast');`
`.slideUp()`	`$('div').slideUp();`
`.slideDown()`	`$('#box1').slideDown('slow');`
`.slideToggle(), .fadeToggle()`	`$('p').slideToggle(300);`

What goes in the parenthesis?
An animation speed

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 — REVIEW

EXERCISE

KEY OBJECTIVE

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

TYPE OF EXERCISE

- Individual/paired

SMALL GROUP PLANNING

5 min

1. Follow the instructions in Starter Code> jquery_review > js/main.js

VARIABLES

EXERCISE — READING AND GUESSING

EXERCISE

KEY OBJECTIVE

- Read a sample JavaScript file and see if you can guess what will happen.

TYPE OF EXERCISE

- Reading exercise (Groups of 3 - 4)

EXECUTION

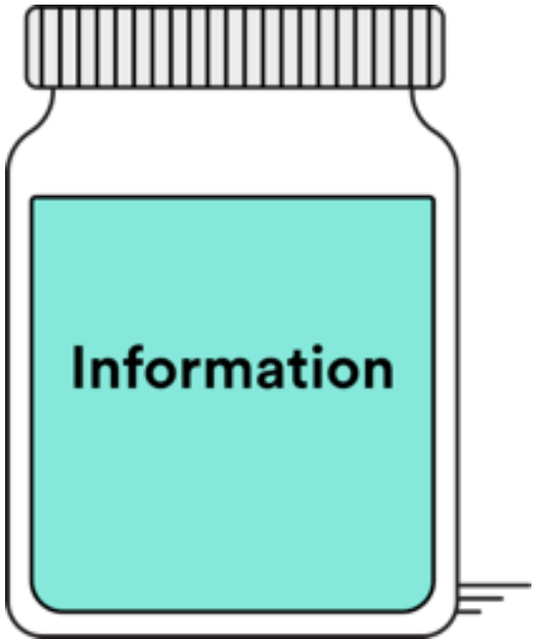
1 min

1. Follow the instructions in starter code > reading_js > main.js

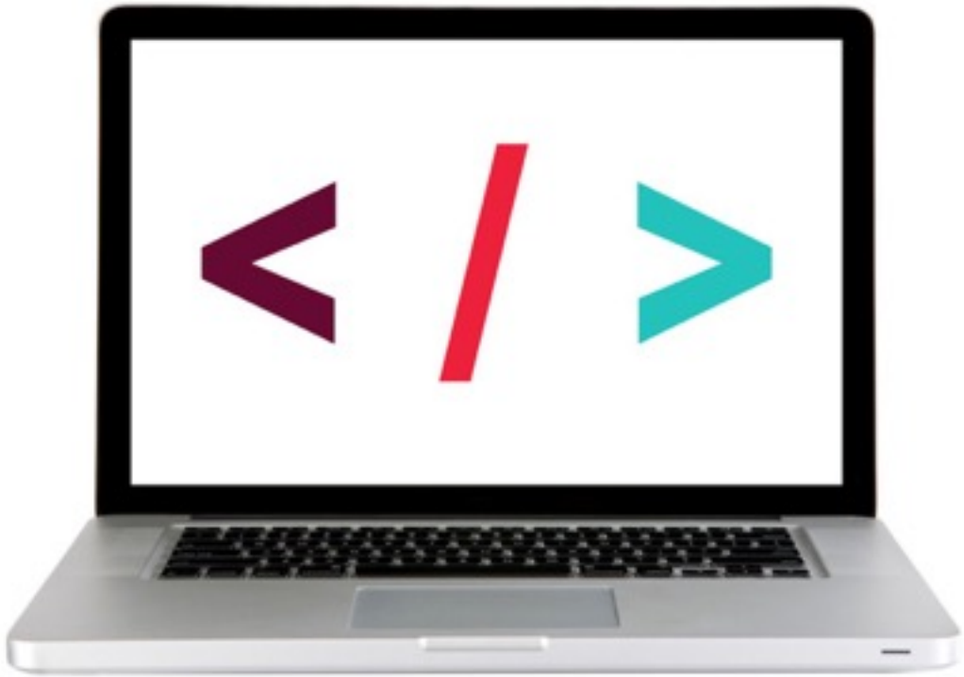
WHAT ARE VARIABLES?

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**



CODE ALONG — SCORE KEEPER



Score Keeper (Codepen)

LEARNING OBJECTIVES

- Define variables and identify best cases to use them.
- Describe strings, numbers, and boolean variable types.
- Use comparison operators to evaluate and compare statements.
- Apply conditionals to change the program's control flow

SYNTAX

CREATING VARIABLES

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!

DECLARING A VARIABLE

```
var age = 29;
```

VARIABLE ASSIGNMENT

JAVASCRIPT — UPDATING THE VALUE OF A VARIABLE

Declaring a variable:

```
var champion = "Sarah";
```

Update the value of the variable:

```
champion = "Celeste";
```

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

ASSIGNMENT OPERATORS

```
var totalAmount = 6;
totalAmount += 4;
totalAmount -= 2;
```

What will total amount be equal to?

ASSIGNMENT OPERATORS

```
var score = 6;
score + 2;
```

What will score be equal to?

RULES

VARIABLE CONVENTIONS

RULES:

1. Should be "camel case" — First word starts with a lowercase letter and any following words start with an uppercase letter.
2. Names can only contain: letters, numbers, $ and _
3. No dashes, no periods.
4. Cannot start with a number
5. Case sensitive - numberofstudents is not the same as numberOfStudents

```
var numberOfStudents = 10;
```

Guideline: Names should be descriptive:

```
var lastName = "Holden";
```

```
var x = "Holden";
```

DATA TYPES

WHAT CAN BE STORED IN VARIABLES?

DATA TYPES:

1. Numeric	2. String	3. Boolean
Handles numbers	Consists of letters and/or other characters	Handles true or false values
Ex: 200.54 Ex: 893	Ex: 'GA@ga.co' Ex: "How are you user?"	Ex: true Ex: false
Used for tasks that involve counting or calculating	Used when working with any kind of text Written with single or double quotes	Used when there are two options for a value (i.e. yes/no, on/off, true/false)

NUMBERS

MORE ABOUT NUMBERS

INTEGERS

```
10
```

Whole numbers

FLOATS

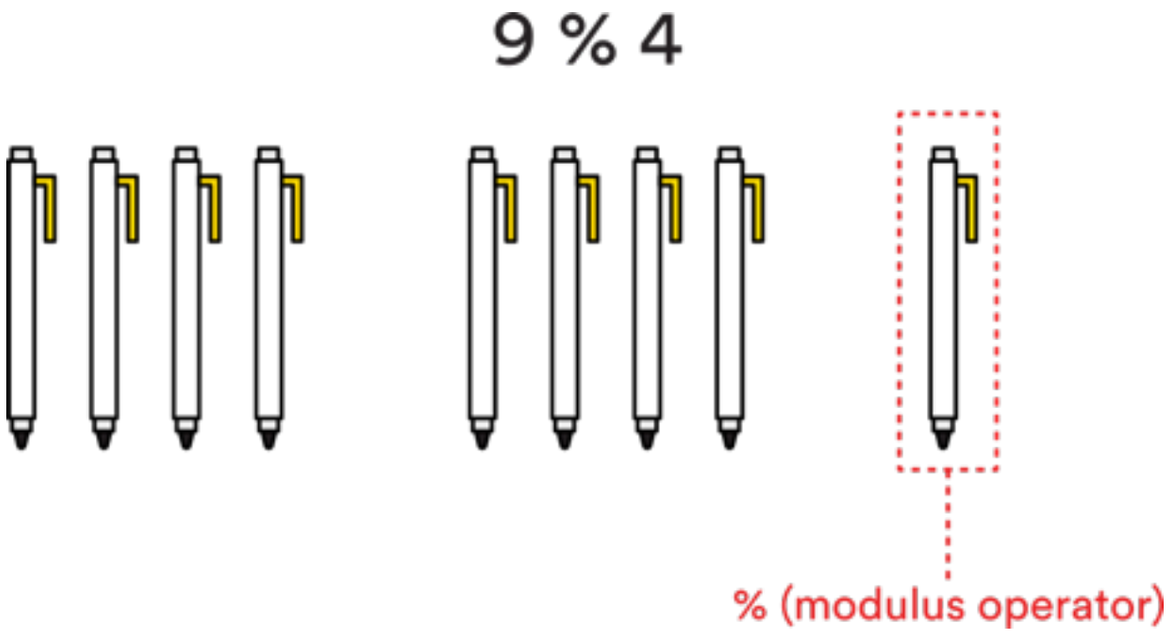
```
22.75
```

Number that uses a decimal to represent a fraction

ARITHMETIC OPERATORS

	Operator	Example	Result
Addition	+	2 + 4	6
Subtraction	-	8 - 1	7
Multiplication	*	2 * 3	6
Division	/	4 / 2	2
Modulus	%	4 % 2	0

ARITHMETIC OPERATORS



EXERCISE — VARIABLES

EXERCISE

KEY OBJECTIVE

- Practice declaring and assigning variables

TYPE OF EXERCISE

- Individual/paired

LOCATION

- Score Keeper (Codepen)

EXECUTION

5 min 1. Hook up the +10, -1 and -5 buttons

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"
```

GETTING THE LENGTH OF A STRING

```
"Hello World".length;
// result: 11
```

```
var userInput = "Suzie"
userInput.length;
// result: 5
```

STRING CONCATENATION

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

```
var name = "Suzie Smith";
var greeting = "Hello " + name;
// greeting will be: "Hello Suzie Smith"
```

COMMON MISTAKES

```
"Bill" = var name;
```

COMMON MISTAKES

```
var name = "Bill";
```

COMMON MISTAKES

```
var total score = 20;
```

COMMON MISTAKES

```
var totalScore = 20;
```

COMMON MISTAKES

```
var fullName = Suzie Smith;
```

COMMON MISTAKES

```
var fullName = "Suzie Smith";
```

COMMON MISTAKES

```
Var fullName = "Bill Smith";
```

COMMON MISTAKES

```
var fullName = "Bill Smith";
```

COMMON MISTAKES

```
var score = "5";
score += "6";
```

COMMON MISTAKES

```
var score = 5;
score += 6;
```

CONDITIONALS

IF STATEMENTS

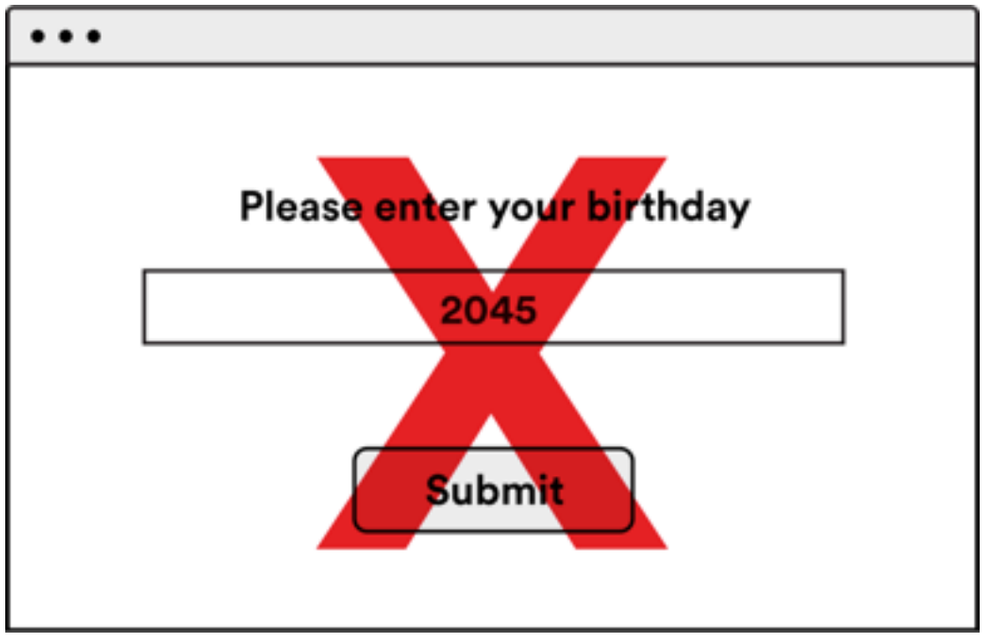
CONDITIONAL LOGIC

If something is true, do one thing. If it is not, do something else.

This type of logic or statement is a condition.

CONDITIONAL LOGIC

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



Is the year less than or equal to 2016 and more than or equal to 1900? If the answer to this question is "true," then we know the user has entered a valid year, and that he or she was born somewhere between 1900 and 2016.

COMPARISON OPERATORS

COMPARISON OPERATORS

Comparison Operators	
<	Less than
>	Greater than
<=	Less than or equal to
>=	Greater than or equal to

COMPARISON OPERATORS

Equality Operators	
`===`	Strict equal to
`==`	Equal to
`!==`	Strict not equal to
`!=`	Not equal to

COMPARISON OPERATORS

```
7 === 7  //true
7 === "7"  //false
0 === false  //false
false === "false"  //false
```

[This is because a string and number are not the same]

[0 is always false in boolean statements]

['false' is a string and JS will not evaluate strings that has the words false as false]

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

Assignment	Comparison
=	== == ==

var number = 7;	if (number === 8) {
	// Do something
	}

EXERCISE — VARIABLES

EXERCISE

KEY OBJECTIVE

- Use comparison operators to evaluate and compare statements.

LOCATION

- Console

EXECUTION

2 min

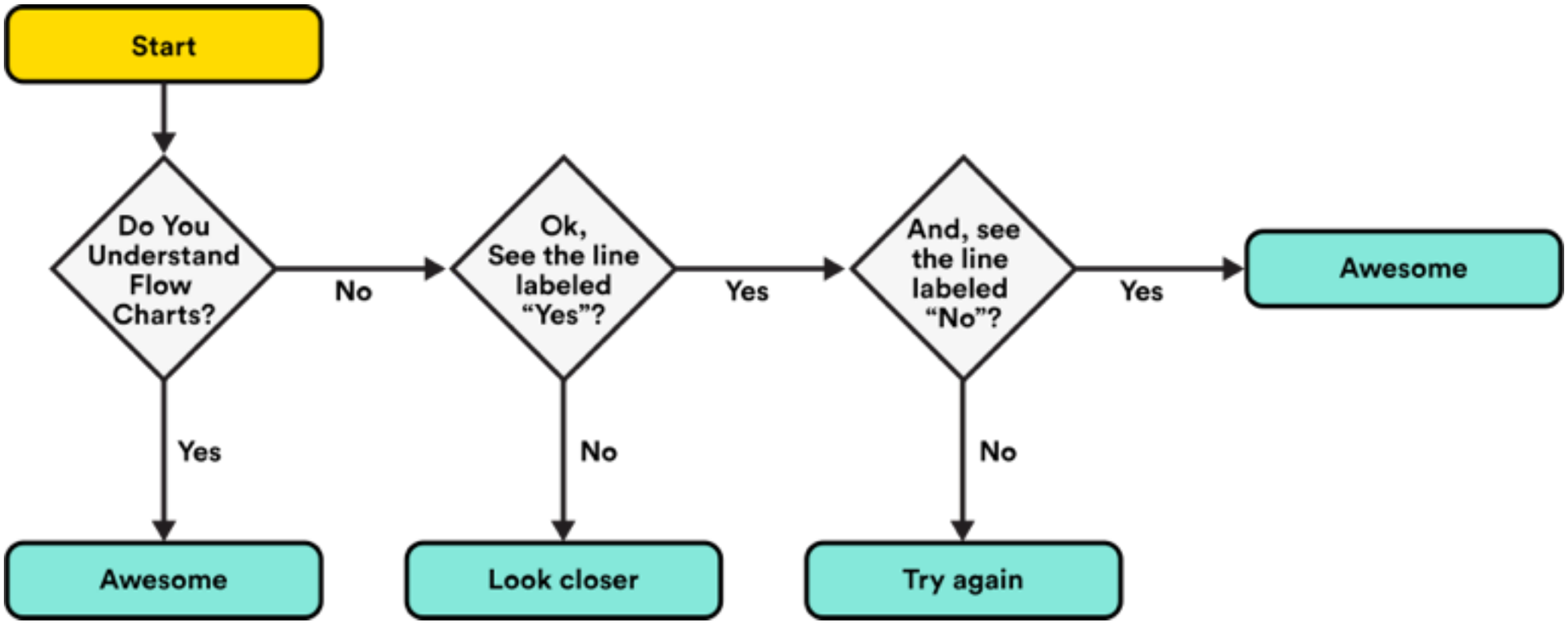
1. Type each command in the console. Before you press enter, take a moment to think about what value the console will return.

```
7 === 7
7 === "7"
7 >= 3
false === "false".
```

```
7 !== "7"
7 != "7"
6 < 3
4 >= 1
```

IF STATEMENTS

CONTROL FLOW



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");
}
```

LOGICAL OPERATORS

MULTIPLE CONDITIONS

&&	and
||	or
!	not

MULTIPLE CONDITIONS

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

EXERCISE — CONDITIONALS

EXERCISE

KEY OBJECTIVE

- Practice writing conditionals

TYPE OF EXERCISE

- Individual/paired

LOCATION

- Starter Code > conditionals

EXECUTION

6 min

1. Follow the instructions in main.js. Refer to your cheat sheet!