1) HAVE SLACK OPEN

2) CREATE/LOGIN TO CODEPEN

SNACKS & DESIGN

TODAY!

MICHAEL SENA & NICHOLAS SKEBA

(GOOGLE SHEET IS PINNED IN SLACK)

Q & A

“How does "this" apply to input values?”

.val()

“Coding anything that changes the number (+ -)”

“What's a good, simplifying rule of thumb to decide when you should or shouldn't set something up using an if else?”

(Context: I started thinking about setting up this temperature converter using an if else for the respective formulas needed for each button, realized eventually that wasn't needed. So how would one decide what is and isn't an appropriate circumstance for making your jQuery magic happen with an if/else?)

“My internet sometimes sucks sometimes (because Comcast). I've noticed that when I'm having connection problems, my browser won't load even local files (even if they don't link to web addresses for Google Fonts or jQuery, etc. Any tips for bypassing that so I can code offline?”

“What’s the benefit to using .on(“event”...) as opposed to just .event?”

.hover()

Exit tickets plz

Context of JS/Programming

HTML Boilerplate

lesson9_starter_code > [-] final_project_boilerplate

Katie's JS question

GENERAL ASSEMBLY

FUNCTIONS

Eric Boyer

LEARNING OBJECTIVES

- Describe arguments as they relate to functions.
- Predict values returned by a given function.

AGENDA

- Review
- Functions — What are functions?
- Functions — Syntax
- Functions — Return Values
- Functions — Scope
- Lab Time — Temperature Converter

REVIEW

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 name = “Matt”;

name = “Ana”;
```

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!*

JAVASCRIPT — COMPARISON OPERATORS

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

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
}
```

JAVASCRIPT — LOGICAL OPERATORS

&& and

|| or

! not

REVIEW EXERCISE — CONDITIONALS

EXERCISE

KEY OBJECTIVE

- Review and practice using variables and conditionals

TYPE OF EXERCISE

- Individual/paired

EXECUTION

12 min

1. Follow the instructions in starter_code_lesson_10 > conditionals > main.js (Part 1)
2. If you finish early, work on the **bonus** section

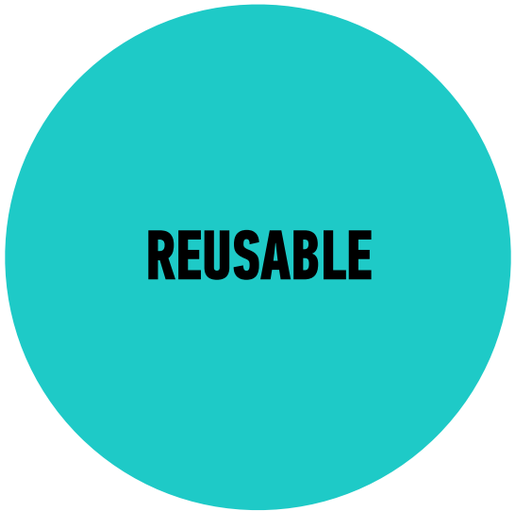
FUNCTIONS

WHAT ARE FUNCTIONS?

FUNCTIONS

FUNCTIONS

Allow us to group a series of statements together to perform a specific task

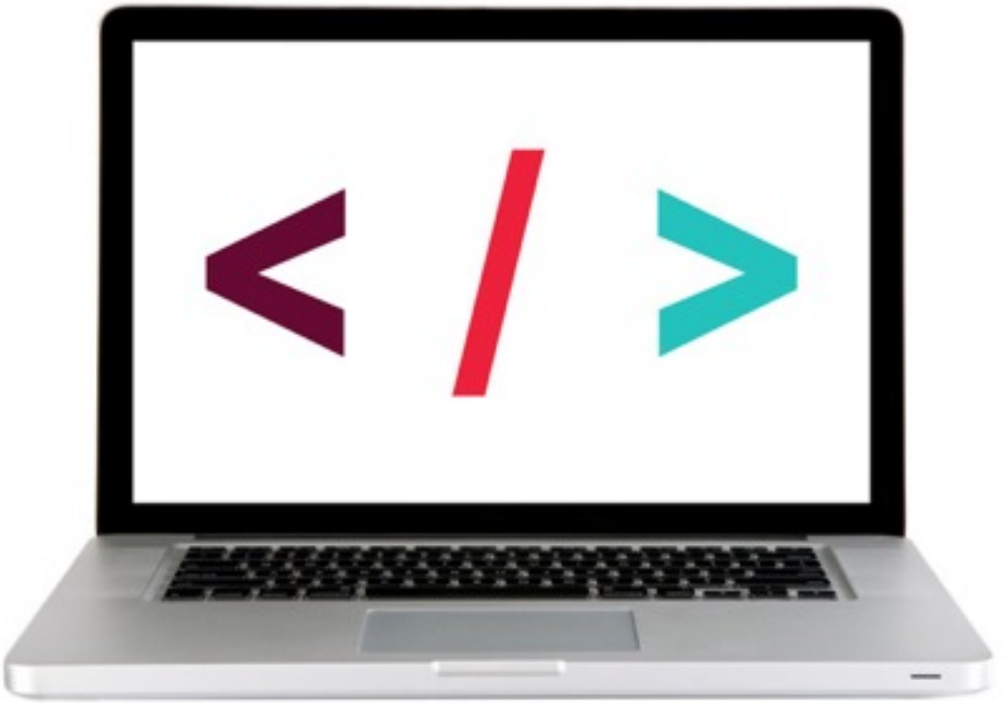


We can use the same function multiple times

Not always executed when a page loads. Provide us with a way to 'store' the steps needed to achieve a task.

DRY — DON'T REPEAT YOURSELF

LET'S TAKE A CLOSER LOOK



jQuery Traffic Light

SYNTAX

SYNTAX — DECLARING A FUNCTION

Keyword

Name

```
function pickADescriptiveName() {
    // Series of statements to execute
}
```

Code block

SYNTAX — CALLING A FUNCTION

- To run the code in a function, we '**call**' the function by using the function name followed by parenthesis.

```
pickADescriptiveName();
```

Function name

FUNCTIONS — TAKING ATTENDANCE

```
function takeAttendance () {
  // Count the number of students in the classroom
  // Write the number of students on the board
}
```

FUNCTIONS — TAKING ATTENDANCE

```
takeAttendance();
```

CODE ALONG — FUNCTIONS

Let's code! lesson9_starter_code > functions (part 1)

SYNTAX — DECLARING A FUNCTION (WITH PARAMETERS)

Parameters

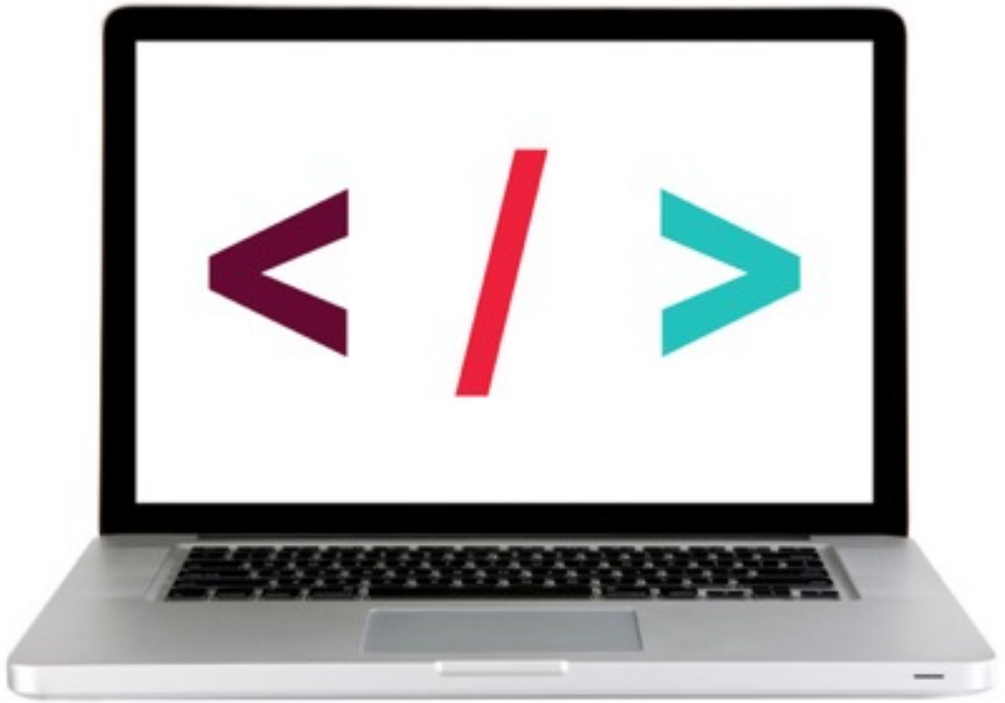
```
function multiply(param1, param2) {
  var result =  param1 * param2;

  $('h1').html(result);
}
```

We can use these parameters like variables from within our function

SYNTAX — CALLING A FUNCTION (WITH ARGUMENTS)

LET'S TAKE A CLOSER LOOK



Multiply on CodePen

FUNCTIONS — GREET

```
function greet (firstName) {
  console.log("Hello " + firstName);
}
```

FUNCTIONS — GREET

```
greet("Michelle");
```

CODE ALONG — FUNCTIONS

Let's code! lesson9_starter_code > functions (part 2)

RETURN VALUES

RETURNING VALUES FROM A FUNCTION

- To return a value from a function, we use the **return** keyword
- From within a function, the **return** keyword 'hands' a value back to the code that called the function
- We can then do something with that value, or store it in a variable for use later in the script

```
function convertToCurrency (entry) {
    // Cut number to two decimal point
    var currency = entry.toFixed(2);
    // Prepend the dollar sign
    currency = '$' + currency;

    return currency;
}
```

```
var amountInDollars = convertToCurrency(entry);
$('ul').append('<li>' + amountInDollars + '</li>');
```

SCOPE

VARIABLE SCOPE

LOCAL VARIABLES

- A **local** variable is a variable that is declared *inside* a function.
- It can **only be used in that function**, and cannot be accessed outside of that function

GLOBAL VARIABLES

- A **global** variable is a variable that is declared *outside* of a function.
- It can be used **anywhere in the script**.

LAB TIME!

LAB

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
});
```

LAB — TEMP CONVERTER — PART 1

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

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

CODE ALONG — FUNCTIONS

Let's code! lesson9_starter_code > [2] temp_converter

LAB — TEMP CONVERTER — PART 2

EXERCISE

KEY OBJECTIVE

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

EXECUTION

Until 8:50

1. Start with the functional temp converter
2. Create getCelsius() and getFahrenheit() functions
3. **Bonus #1**: Change the background-color depending on what temperature the user enters (example)
4. **Bonus #2**: Add error styles if the user doesn't enter a value in the form (example)

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

LEARNING OBJECTIVES

- Describe arguments as they relate to functions.
- Predict values returned by a given function.

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

~~Katie Allyn~~
~~Kimberly Baird~~
Tom Bunting
Michael Fischer
Jim Howes
~~Jon Iler~~
Peyton Lee
Anna Matras
~~Christopher Zalek~~
Corin Menuge
Colin Quirk
Allison Schaffer
Kevin Sella
Michael Sena
Micholas Skeba
Ashley Sullivan
Lisa Vasquez
Katrina Wang
Xi Yue Li

LAB

HOMEWORK

HOMEWORK

FEEDBACK GROUPS FOR CASH_REGISTER

IF YOU ARE STUCK...

1) Use the “Chrome Inspector” to look at your code, use the console, Google for answers.

2) Ask your question in the Slack channel and see if any fellow students might know the answer

3) Ask Eric & Adriana

* When using your fellow students and instructors, pushing your code to Github is a great place to share where all of your code is currently at.

FEEDBACK GROUPS

GROUP 1 (ADRIANA)

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

TO: ADRIANA.LCS316@GMAIL.COM

GROUP 2 (ERIC)

Corin Menuge
Colin Quirk
Allison Schaffer
Kevin Sella
Michael Sena
Micholas Skeba
Ashley Sullivan
Lisa Vasquez
Katrina Wang
Xi Yue Li

TO: EBOYER@GMAIL.COM

FEEDBACK GROUPS

DUE 11:59PM, SUNDAY MAR 13TH

- Complete the “temp_converter” and “cash_register” lab assignments
- Continue to think about your final project, *March 27th wireframes are due*
- Add files, Commit, and Push/Publish your homework to your class Github Repo

*Feedback given between Monday AM - Wednesday PM

SNACKS & DESIGN

TUESDAY MAR 22ND

ASHLEY SULLIVAN, LISA VASQUEZ

(GOOGLE SHEET IS PINNED IN SLACK)

EXIT TICKETS