LET'S GET EVERYTHING SET UP!

- 1. Navigate to the FEWD 42 Dashboard (<u>saraheholden.com/fewd_dashboard/</u>) and download the Lesson 10 starter code and slides. You'll want to keep the dashboard open for other links and resources we'll be referencing in class.
- 2. Move the starter code and slides from your Downloads folder to the **fewd** folder on your desktop.
- 3. Double-click on starter_code_lesson_10.zip to unzip it
- 4. After you've unzipped, be sure to delete the original .zip file!
- 5. Open the entire **fewd** folder with Sublime Text (either drag and drop the folder on the Sublime icon in the dock on Mac, or open Sublime and go to file > open... and select the **fewd** folder.
- 6. Log in to the FEWD 42 Slack (<u>fewd42.slack.com</u>) and join the class10 channel.
- 7. Open the Github desktop app

WEEKLY OVERVIEW

WEEK 4	Intro to Programming / Intro to jQuery
WEEK 5	Intro to JavaScript — Variables, Conditionals, Functions
WEEK 6	JavaScript Continued — Arrays / Lab

NOTE!

HAPPY HOUR!!

Date: Thursday, April 7th

When: Right after class

Location: TBD

NOTE!

BEWD

- ▶ Many people take BEWD after FEWD as a next step and to add back end functionality
- ▶ Ruby on Rails
- ▶ Will be taught by the amazing Kisha!
- We won't be running BEWD for the rest of the year and don't have anything scheduled for 2017
- > Students will receive a \$500 alumni discount
- ▶ A little bit of overlap but the course launches April 20th-June 29th (M & W)

FINAL PROJECT MILESTONE #1

LET'S GIT IT!

IN FINDER:

Move the "wireframes" folder from today's starter code folder to your homework folder.

IN THE GITHUB APP:

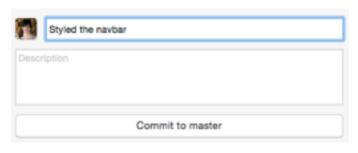
ADD

Make sure the boxes next to each file you've edited are checked.

travel_blog_part1/images/ad.gif

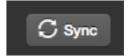
COMMIT

Enter a commit message and click "commit to master."



PUSH

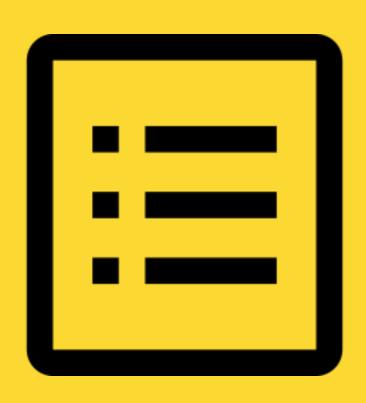
Click "sync" in upper right corner.



LEARNING OBJECTIVES

- Define a function with one or more parameters
- Execute a function within a program
- Given a function and a set of arguments, predict the output of a function

AGENDA



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



FEWD

REVIEW

JAVASCRIPT — VARIABLES

Declaring a variable

Semicolon!

Assigning a variable

$$\rightarrow$$
 age = 29; \leftarrow Semicolon!

Both in one step

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

WHAT CAN BE STORED IN VARIABLES?

DATA TYPES:

STRINGS

"Today is Monday"

Letters and other characters enclosed in quotes

NUMBERS

10 22.75

- Positive numbersNegative 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

> = Greater than or equal to

Equal to = = =

Less than or equal to

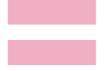
Not equal to ==

> Greater than

Less than

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

ASSIGNMENT



var number = 7;

COMPARISON



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

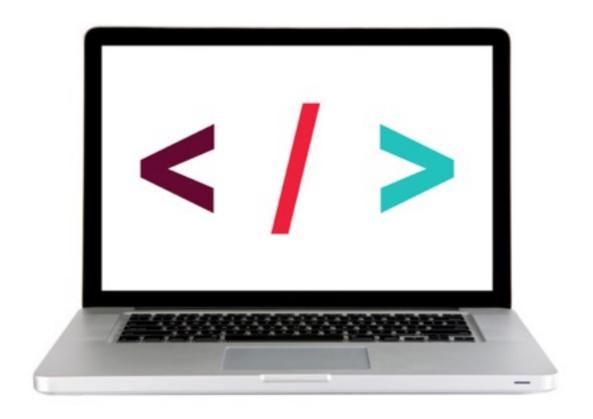
JAVASCRIPT — IF/ELSE IF/ELSE

```
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



LET'S TAKE A CLOSER LOOK



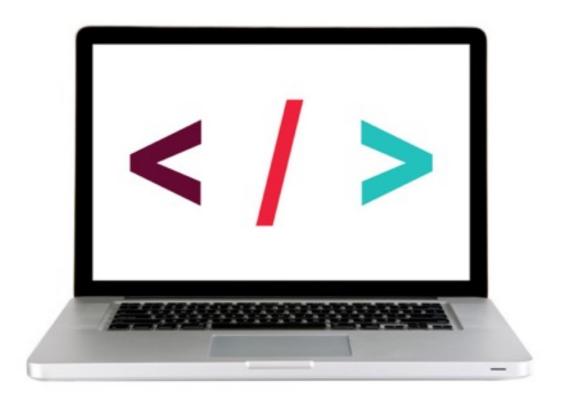
starter_code_lesson_10 > compare_two_numbers

EXIT TICKET QUESTIONS

- Best practice sites for Javascript? <u>Codecademy</u>, <u>CodeSchool</u>, <u>Dash</u>, <u>30 Days to Learn</u> jQuery series
- ▶ Will we learn to "refactor" our jQuery? The way we learned how to refactor CSS?
- ▶ Side question...what is the best way to share our final project? CodePen? any free versions of CodePen?
- ▶ Are there exceptions to setting CSS rules in Javascript? i.e. Display: none; ???

CASH REGISTER PT. 1

CASH REGISTER

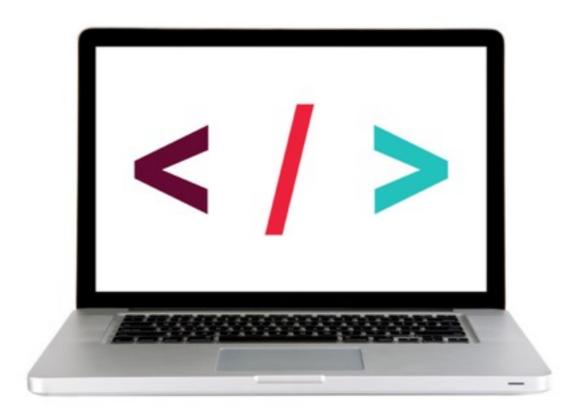


Cash Register Pt. 1

FUNCTIONS

WHAT ARE FUNCTIONS?

LET'S TAKE A CLOSER LOOK

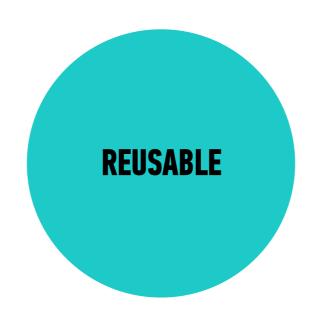


jQuery Traffic Light





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

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();

EXERCISE — WRITING FUNCTIONS



KEY OBJECTIVE

Practice defining and executing functions

TYPE OF EXERCISE

Individual/paired

LOCATION

starter_code_lesson_10 > functions (part 1)

EXECUTION

4 min

1. Follow the instructions under Part 1

SYNTAX — **DECLARING A FUNCTION (WITH PARAMETERS)**

Parameters

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

We can use these parameters like variables from within our function

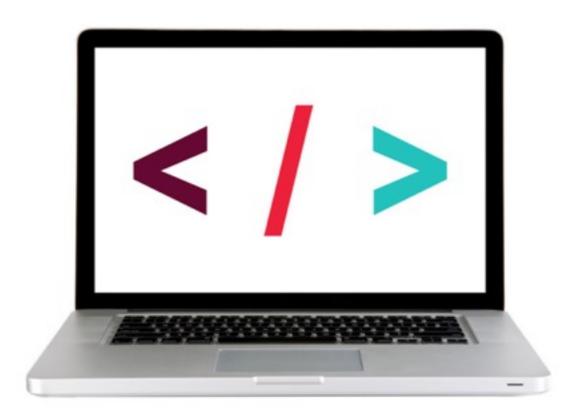
```
$('h1').html(result);
```

SYNTAX — CALLING A FUNCTION (WITH ARGUMENTS)

Arguments

multiply(350, 140)

LET'S TAKE A CLOSER LOOK



Multiply on CodePen

FUNCTIONS — CALCULATE THE AREA

```
function getArea(width, height) {
  var area = width * height;
  // write area on board
}
```

getArea(5, 2);

EXERCISE — **READING FUNCTIONS**



KEY OBJECTIVE

 Given a function and a set of arguments, predict the output of a function

TYPE OF EXERCISE

▶ Groups of 2 - 3

LOCATION

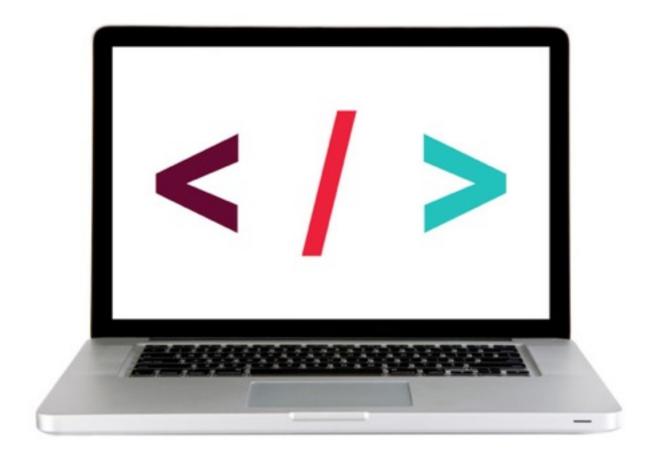
starter_code_lesson_10 > functions (part 2)

EXECUTION

- 1 *min* 1. Look at Part 2A. Predict what will happen when the function is called.
- 1 *min* 2. Look at Part 2B. Predict what will happen when the function is called.
- 1 *min* 3. Look at Part 2C. Predict what will happen when the function is called.

SCOPE

LET'S TAKE A CLOSER LOOK



View example in **Codepen**

VARIABLE SCOPE

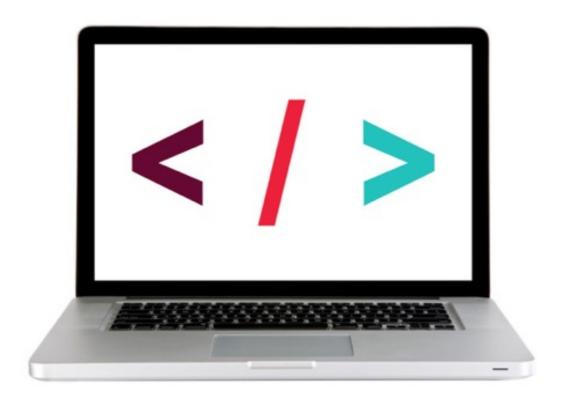
LOCAL VARIABLES

- 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.

CODE ALONG — SCORE KEEPER



Score Keeper (Codepen)

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 multiply (num1, num2) {
    return num1 * num2;
}
```

```
var result = multiply(6, 8);
console.log(result);
```

RETURNING VALUES FROM A FUNCTION

```
function convertToDollars (number) {
    // Cut number to two decimal points
    // Prepend the dollar sign
    // return dollar amount
}
```

```
var amountInDollars = convertToDollars(3.5012);
// Write amountInDollars on board
```

EXERCISE — WRITING FUNCTIONS



KEY OBJECTIVE

Practice returning values from a function

TYPE OF EXERCISE

Individual/paired

LOCATION

starter_code_lesson_10 > functions (part 3)

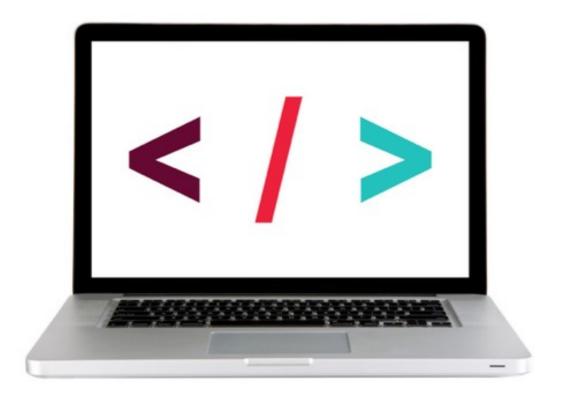
EXECUTION

6 min

1. Follow the instructions under Part 3

GROUP CHALLENGE

CASH REGISTER



Cash Register Pt. 2

LAB TIME!



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 2



KEY OBJECTIVE

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

WHAT WE WANT OUR END RESULT TO BE

Use the <u>live version</u> to test the functionality

EXECUTION

Until 9:15

- 1. Write .js to make the temperature converter functional.
- 2. Change the background-color depending on what temperature the user enters
- 3. **Bonus #1**: Add error styles if the user doesn't enter a value in the form
- 4. **Bonus #2**: Add error styles if the user a value that's not a number (you'll need to do some google research)
- 5. **Bonus #3**: Add your own styles to the temperature converter (example, example)

LEARNING OBJECTIVES

- Define a function with one or more parameters
- Execute a function within a program
- Given a function and a set of arguments, predict the output of a function

HOMEWORK

HOMEWORK

Be sure to read the specs on the <u>FEWD 42 Dashboard</u>.

ADVANCED CSS

EXIT TICKETS

http://goo.gl/forms/vPhCOlfESf