WEEKLY OVERVIEW

WEEK 6 Functions / CSS Positioning & Animation

WEEK 7 Interactions / Lab

LEARNING OBJECTIVES

- Describe why functions are useful
- Describe how parameters relate to functions
- Given a function and a set of arguments, predict the output of a function
- Define and call a function
- Compare global and local scope

AGENDA

Review

Functions — What are Functions?

Functions — Syntax

Functions — Scope

Lab — Rock, Paper, Scissors

REVIEW / DEBUGGING

DEBUGGING — STEPS

- 1. Check your console for errors.
- 2. Check to make sure you're using # for ids and . for classes
- 3. Check to make sure your classes and ids are spelled the same in your HTML and JS file
- 4. Walk through your code in the debugger ("Sources" panel) and check values
- 5. Google it!

EXERCISE —HOMEWORK REVIEW



KEY OBJECTIVE

▶ Review homework solution

TYPE OF EXERCISE

▶ Groups of 2 - 3

EXECUTION

10 min

- 1. Use the console and the debugger to fix the four errors in the HW solution (mad_libs folder).
- 2. Walk through the solution code line-by-line and write a comment above each line to describe what is happening.

LAB —— PART 1

EXERCISE — ROCK, PAPER, SCISSORS



KEY OBJECTIVE

Practice JavaScript basics, then refactor code to avoid repetition.

TYPE OF EXERCISE

▶ Groups of 2 - 3

LOCATION

starter code > Rock, Paper, Scissors

EXECUTION

10 min

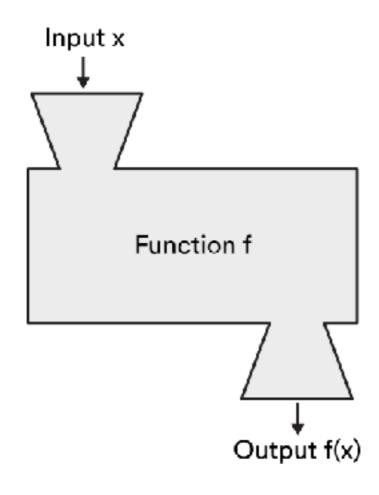
- 1. In groups of 2 3, write pseudo code for the game.
- 2. As a class: Review pseudo code

45 min

3. Write JS to make the game functional. For this portion of the class we are only working on the "Rock" button.

FUNCTIONS

WHAT ARE FUNCTIONS?



x	f(x)
-1	-2
0	0
1	2
2	4
3	6

EXERCISE — FUNCTIONS INTRO



KEY OBJECTIVE

Practice reading code and guessing what's happening.

TYPE OF EXERCISE

• Groups of 3 - 4

LOCATION

starter code > functions_intro

EXECUTION

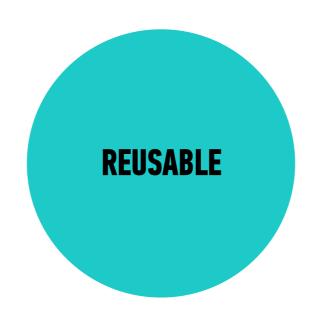
8 min

- 1. Walk through the questions at the top of main.js and discuss.
- 2. Walk through the questions in part 2 and discuss.





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

```
var movie = "Saving Private Ryan";
var cost = 100;
if (cost < 1000) {
    console.log(movie + " director says LET'S RESCUE MATT DAMON!");
} else {
    console.log(movie + " director says FORGET MATT DAMON!");
var movie = "Interstellar";
var cost = 2000;
if (cost < 1000) {
    console.log(movie + " director says LET'S RESCUE MATT DAMONI");
} else {
    console.log(movie + " director says FORGET MATT DAMON!");
var movie = "The Martian";
var cost = 700;
if (cost < 1000) {
    console.log(movie + " director says LET'S RESCUE MATT DAMON!");
} else {
    console.log(movie + " director says FORGET MATT DAMON!");
```

```
function mattDamon (movie, cost) {
   if (cost < 1000) {
      console.log(movie + " director says LET'S RESCUE MATT DAMON!");
   } else {
      console.log(movie + " director says FORGET MATT DAMON!");
   }
}

mattDamon("Saving Private Ryan", 100);
mattDamon("Interstellar", 2000);
mattDamon("The Martian", 700);
mattDamon("Elysium", 200);</pre>
```

mattDamon("movieTitleHere", costHere)

LEARNING OBJECTIVES

- Describe why functions are useful
- Describe how parameters relate to functions
- Given a function and a set of arguments, predict the output of a function
- Define and call a function
- Compare global and local scope

SYNTAX

SYNTAX — **DECLARING A FUNCTION**

```
Keyword Name

function pickADescriptiveName() {
    // Code to run
}
```

SYNTAX — CALLING A FUNCTION

```
function pickADescriptiveName() {
    // Code to run
}
```

To run the function, we need to *call* it. We can do so like this:

```
pickADescriptiveName();
```

Function name + parentheses

FUNCTIONS — LET'S PLAY SOME MUSIC!

```
function playSong () {
   var song = "Don't Stop 'Til You Get Enough";
   var artist = "Michael Jackson";

   $('#nowPlaying').html('Now playing: ' + song + ' by ' + artist);
}
```

FUNCTIONS — LET'S PLAY SOME MUSIC!

```
function playSong () {
    var song = "Don't Stop 'Til You Get Enough";
    var artist = "Michael Jackson";

    $('#nowPlaying').html('Now playing: ' + song + ' by ' + artist);
}

playSong();
```

EXERCISE — WRITING FUNCTIONS



KEY OBJECTIVE

Practice defining and executing functions

TYPE OF EXERCISE

Individual/paired

LOCATION

starter code > functions (part 1)

EXECUTION

4 min

1. Follow the instructions under Part 1

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

```
function multiply (param1, param2) {
    $('h1').html(param1 * param2);
}

We can use these parameters like variables from within our function
```

```
Arguments multiply(350, 140)
```

FUNCTIONS — LET'S PLAY SOME MUSIC!

```
function playSong (song, artist) {
    $('#nowPlaying').html('Now playing: ' + song + ' by ' + artist);
}
```

FUNCTIONS — LET'S PLAY SOME MUSIC!

```
function playSong (song, artist) {
    $('#nowPlaying').html('Now playing: ' + song + ' by ' + artist);
}
playSong("Don't Stop 'Til You Get Enough", "Michael Jackson");
```

```
function sayHello (name) {
    $('.greeting').html("Hello " + name);
}
```

What will be the html that gets added to .greeting if we call the function using these arguments:

```
sayHello("Drake");
```

```
function sayHello (name) {
    $('.greeting').html("Hello " + name);
}
```

How about now?

```
sayHello("Sally");
```

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 > functions (part 2)

EXECUTION

3 min

1. Look at Part 2 A and B. Predict what will happen when each function is called.

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 > functions (part 3)

EXECUTION

8 min

- 1. See if you can write one function that takes some parameters and combines the functionality of the *makeAPizza* and *makeAVeggiePizza* functions.
- 2. BONUS: Create your own function with parameters. This function could do anything!

EXERCISE — FUNCTIONS



KEY OBJECTIVE

- Describe why functions are useful
- Describe how parameters relate to functions

TYPE OF EXERCISE

Turn and Talk

EXECUTION

1 min

- 1. Summarize why we would use functions in our programs. What purpose do they serve?
- 2. How do parameters relate to functions?

LEARNING OBJECTIVES

- Describe why functions are useful
- Describe how parameters relate to functions
- Given a function and a set of arguments, predict the output of a function
- Define and call a function
- Compare global and local scope

SCOPE

VARIABLE SCOPE

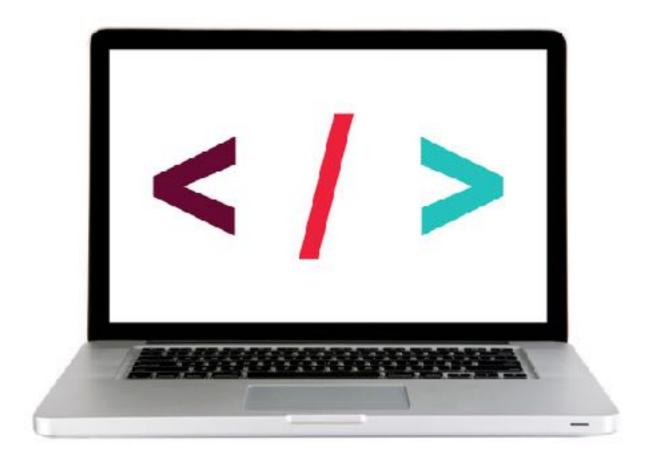
LOCAL VARIABLES

- 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.

LET'S TAKE A CLOSER LOOK



View example in **Codepen**

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 > functions (part 4)

EXECUTION

8 min

1. Follow the instructions under Part 4 in main.js

LEARNING OBJECTIVES

- Describe why functions are useful
- Describe how parameters relate to functions
- Given a function and a set of arguments, predict the output of a function
- Define and call a function
- Compare global and local scope

LAB TIME!

EXERCISE — ROCK, PAPER, SCISSORS



KEY OBJECTIVE

Practice JavaScript basics, then refactor code to avoid repetition.

TYPE OF EXERCISE

▶ Groups of 2 - 3

LOCATION

starter code > Rock, Paper, Scissors

EXECUTION

Until 9:20

- 1. Continue building the game, enabling the Paper and Scissors buttons, as well as feedback in the browser window.
- 2. Use functions to group together any repeated code.
- 3. **Bonus**: Create a "Best of three" version of the game. After three rounds, there should be some visual feedback for the user as to who is the "final" winner. Get creative!



LEARNING OBJECTIVES

- Describe why functions are useful
- Describe how parameters relate to functions
- Given a function and a set of arguments, predict the output of a function
- Define and call a function
- Compare global and local scope

WEEKLY OVERVIEW

WEEK 6 Functions / CSS Positioning & Animation

WEEK 7 Interactions / Lab

EXIT TICKETS!