# 1 Warm up 💪🏼

1. *Declare* and *invoke* a **function** named *greeting* that takes a **string** name as an argument and returns a greeting for the given name. The test cases should pass.
2. **// Test cases**
3. **console.log(greeting("Kana")); // => "Hello, Kana!"**
4. **console.log(greeting("Kimiko")); // => "Hello, Kimiko!"**
5. Write a **function** *average* that takes two **number**s as inputs and returns the average of those numbers. Don’t forget to write tests!

# 2 function Parts

## Lecture Slides

## Vocabulary

* ***function*** keyword - the built in word **function** that you start a function with to let the computer know that you’re declaring a **function**
* function name - the name of your **function**. It follows the special **function** keyword
* parameter - the name(s) passed to your function declaration that describe the inputs. They are declared inside the parentheses that follow the function name. The order of the parameters correspond with the order that inputs are given during function invocation.
* argument - the values passed to the function during function invocation.
* return value - the value that is returned from the function. A function can only return a single value.
* body - the main part of the function declaration. Located after the parentheses and in between two curly braces.
* return statement - the line that contains the **return** keyword. The **return** keyword, similar to the **function** keyword, is a special keyword that you can only use inside of the function. The expression that happens after the **return** keyword is what gets returned by the function after it is invoked. Any lines in the function after a return statement will not be executed. The **return** keyword also indicates the end of the function.
* function invocation operator - the parentheses that follow the function name during invocation.

## Exercises

There are no exercises associated with this lesson!

# 3Project HTML + CSS

## Lecture Slides

**IMPORTANT** Due before Saturday, Week 3 **IMPORTANT**

## Objectives

* Use HTML elements to place items on a page
* Be able to link your CSS page to your HTML page
* Use basic HTML and CSS techniques 🌱
* Create simple webpage introducing yourself
* Use class and ids to categorize and identify elements on your page

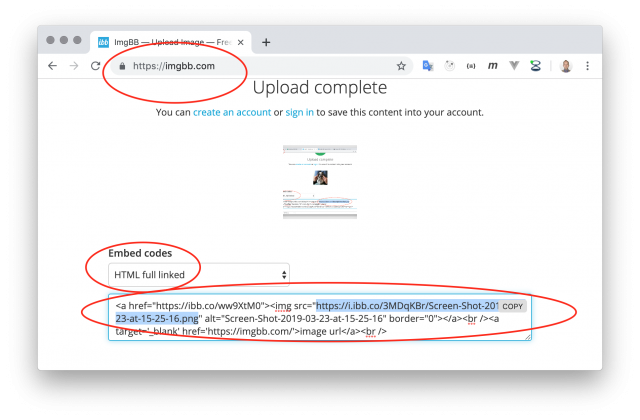
## Exercises

### Basic Requirements

1. Please choose **one** of the following to do. If you find another similar resource that works for you, feel free to use it as long as it covers the same content matter as the below and helps you finish the project on this page. There is no need to pay for the premium versions.
   * More interactive: [Khanacademy](https://www.khanacademy.org/computing/computer-programming/html-css" \l "web-development-tools" \t "_blank)
     + Do the following chapters:
       - Intro to HTML
       - Intro to CSS
       - More HTML Tags
       - CSS Text Properties
       - CSS layout (only through box model)
   * More academic: [Shay Howe’s HTML/CSS tutorials](http://learn.shayhowe.com/html-css/)
     + Read Chapters 1 - 3, 8
     + Skim Chapters 4, 5, 10
2. In your Foundations folder, create a copy of **\_lesson-template** called **html-css-homework**.
3. Create a file named **styles.css** in your **html-css-homework** folder.
4. [Link your css file](https://teamtreehouse.com/community/htmlcss-linking) to the **index.html**, so that your styles appear on the page.
5. In the folder, create a personal webpage that satisfies the following:
   * Your name in **<h1>** tags with an id of **profile-name**
   * Your occupation (or whatever title you want to give yourself) in **<h2>** tags
   * A picture of yourself with an id of **profile-pic**
   * Where you call home in **<h3>** tags
   * Five pictures of anything of significance to you: family, friends, animals, anything. Give each image a class of **profile-favorite**
   * An unordered list of your favorite hobbies each with a class of **profile-hobby**

You do not need to worry about **display**, **position** or **flex-box** for now.

**Note**: We suggest using [imgbb.com](https://imgbb.com/) to host your images. After you upload the image, use the **src** for the image in your code. Like this:



1. Copy and paste your HTML and CSS into a [CodePen](https://www.codepen.io/" \t "_blank) and share the link on your class Slack channel.

### Medium Requirements

1. Make it look nice using CSS by:
   * Changing the background color
   * Adding a border to your images
   * Resizing your images

### Advanced Requirements

1. If you chose Khanacademy, do the rest of the course. If you chose to read Shay Howe’s tutorials, read the rest!
2. Get started with flexbox here: [A Complete Guide to Flexbox](https://css-tricks.com/snippets/css/a-guide-to-flexbox/)
   * Play this: [Flexbox Froggy](https://flexboxfroggy.com/) 🐸
3. Skim more thoroughly: [Beginner Concepts: How CSS Selectors work](https://css-tricks.com/how-css-selectors-work/)
4. Test your knowledge here: [CSS Sushi Selector Game](https://flukeout.github.io/)
5. Skim more thoroughly articles about the CSS box model: [MDN](https://developer.mozilla.org/en-US/docs/Learn/CSS/Introduction_to_CSS/Box_model) and [CSS-tricks](https://css-tricks.com/the-css-box-model/)

### Resources

* [HTML dog](http://www.htmldog.com/)
* [HTML & CSS is Hard](https://internetingishard.com/html-and-css/)

## Homework

There will be no more dedicated class time for this project. Complete this project before before Saturday, Week 3. **IMPORTANT**

# 4 Warm up 💪🏼

1. Declare and invoke a **function** named isMultipleOfThree that returns whether or not the given number is a multiple of three.
2. **function isMultipleOfThree(/\*???\*/) {**
3. **// ???**
4. **}**
5. **// Test cases**
6. **console.log(isMultipleOfThree(6)); // => true**
7. **console.log(isMultipleOfThree(10)); // => false**
8. Declare and invoke a **function** named isMultipleOf that returns whether or not the given first number is a multiple of the given second number.
9. **function isMultipleOf(/\*???\*/) {**
10. **// ???**
11. **}**
12. **// Test cases**
13. **console.log(isMultipleOf(6, 3)); // => true**

**console.log(isMultipleOf(10, 4)); // => false**

# 5 Imposter Syndrome

## Lecture Slides

## Resources

* [Imposter Syndrome Resources](http://www.cs.stir.ac.uk/sciencegrrl/impostor/)
* [5 Types of Imposter Syndrome and How to Beat Them - Fast Company](https://www.fastcompany.com/40421352/the-five-types-of-impostor-syndrome-and-how-to-beat-them)
* [Harvard Business Review - Everyone Suffers from Imposter Syndrome and Here’s How to Handle It](https://hbr.org/2016/07/everyone-suffers-from-imposter-syndrome-heres-how-to-handle-it)
* [Ada Initiative Training](https://adainitiative.org/continue-our-work/impostor-syndrome-training/)

# 6 Intro to Conditionals

## Lecture Slides

## Vocabulary

* conditional keywords - **if**, **else if**, and **else**
* expression - You should know this! It is any piece of code that evaluates to a value. Ex. **1+1**, **"hello"**, **true !== false**

## Exercises

### Basic Requirements

1. Fix the code below.
2. **if ((1 = 1)) {**
3. **console.log(true);**
4. **}**
5. Fix the code below.
6. **function simplePasswordLock(password) {**
7. **if ((password = "password")) {**
8. **return "Correct! Welcome.";**
9. **}**
10. **return "Incorrect password, please try again.";**
11. **}**

Tests:

**console.log(simplePasswordLock("qwerty")); // should return "Incorrect password, please try again."**

**console.log(simplePasswordLock("password")); // => "Correct! Welcome."**

1. Do you remember **.length**? The quickest way to find out the length of a string is to use its **.length** property.

What does that mean? Run the lines below to see it in action!

**console.log("Hello".length);**

**console.log("The length of this string is 31".length);**

**console.log(" spaces! ".length);**

1. Declare a **function** called isItTooLong that takes a **string** as input.

Using the **.length** property you learned above, your **function** should return **true** if the input **string** is LONGER than **10** and **false** if the input string is SHORTER than or EQUAL to **10**.

Write some test cases to test it!

1. Add conditionals to function below so that it returns whichever number is bigger. Look at the test cases below to see how it’s supposed to work.
2. **function biggerNumber(numOne, numTwo) {}**

Test cases:

**console.log(biggerNumber(4, 3)); // should print 'The first argument is bigger.'**

**console.log(biggerNumber(3, 4)); // => 'The second argument is bigger.'**

1. Declare a **function** printDataType that prints the data type that is passed in as an argument. You will use a JavaScript operator **typeof**.

You can learn more about this operator [here](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Operators/typeof).

The first has been done for you below. Use the test cases below the **function** to see how printDataType is supposed to work!

**function printDataType(data) {**

**if (typeof data === "string") {**

**console.log("This is a string.");**

**}**

**.**

**}**

Test cases:

**printDataType("Hello!"); // should print "This is a string."**

**printDataType(true); // => "This is a boolean."**

**printDataType(42); // => "This is a number."**

**printDataType(undefined); // should print "This is not a string, boolean, or number."**

1. Create a **function** called greeting that takes in two parameters: a name and a language. It should return a greeting for the person in the language they have specified.

Test cases:

**console.log(greeting("Harry Potter", "Japanese")); // should print "Konnichiwa, Harry Potter!"**

**console.log(greeting("Harry Potter", "English")); // => "Hello, Harry Potter!"**

**console.log(greeting("Harry Potter", "German")); // => "Gutentag, Harry Potter!"**

**console.log(greeting("Harry Potter", "Spanish")); // => "Hola, Harry Potter!"**

1. Create a **function** called isEven that returns **true** IF the **number** is even.

**Hint**: for this, you may want to use the remainder operator (%).

**Bonus**: add a conditional before everything else  
that returns “This is not a number.” IF the argument  
passed in is not a number.

Test cases:

**console.log(isEven(4)); // should print true**

**console.log(isEven(7)); // => false**

### Medium Requirements

Like you did above, finish writing the functions described below. Each must take one **number** as an argument:

1. isOdd: returns **true** IF its input is an odd number.
2. isPositive: returns **true** IF its input is positive, and **false** IF it is not positive.
3. isNegative: returns **true** IF its input is negative, and **false** IF it is not negative.
4. isZero: returns **true** IF its input is zero, and **false** IF it is not zero.

The built-in **Math** object (we’ll learn what objects are at a later time) has a lot of useful mathematical properties and methods (methods are a kind of function…more about that later!). A few useful methods are **Math.random**, **Math.floor** and**Math.ceil**. Look up these functions on [MDN](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Math) to learn how they work, and use them to implement the following:

1. randomNumber: This **function** should accept a **number** as its input and return a random **number** between **0** (inclusive) and **number** (inclusive).
2. guessMyNumber: This **function** accepts a **number** argument and compares it to a random integer (whole number) between **0** (inclusive) and **5** (inclusive). It should return **"YES!"** if the argument matches the random number and **"NO!"** if the argument does not match.

### Advanced Requirements

For these exercises, you may need to use techniques that we haven’t covered in class.

1. Declare a **function** called **randomStopLight**. It should create a random **number** from **0** to **10**. If the **number** generated is less than **3**, the **function** should return **"red"**. If the **number** generated is between **3** and **6** (inclusive of both), it should return **"yellow"**. If the number generated is **7** or greater, it should return **"green"**.

**Hint**: You may need to use **Math.random()**.

## Homework

* Complete all Basic Requirements for this lesson.
* **IMPORTANT**: Make sure you are on track for your HTML/CSS online course.

### Additional Reading 📖

* Read [Eloquent JavaScript - Chapter 3](https://eloquentjavascript.net/2nd_edition/03_functions.html).
* [Top 8 Developer Habits](https://www.youtube.com/watch?v=RleN-6uMF04&list=PL0zVEGEvSaeGY3RMjGo4CgMPN42_U9Glu&index=2)

# 7 Retrospective

Let’s take some time to discuss:

* What went well during the last week? 😃
* What was “meh”? 😐
* What didn’t go well? 😞
* What can we do to improve our experience in this class?