Just The Beginning

Let's get some practice with everything you've learned today! Using terminal and reading errors and using loops and getting used to basic JavaScript syntax are all key to web development and you'll be using them throughout the entire course -- so let's get comfortable with them by doing some reps!

Learning Objectives

- Terminal
- Errors
- Boolean expressions
- While loops
- For loops

Prerequisites

- Terminal
- JavaScript

Getting Started - Terminal

1. Open Terminal
2. Navigate to your Documents directory /User/username/sei/day02
3. Make a directory inside Documents called w1d1_lab
4. Go into the w1d2_lab directory
5. Inside w1d2_lab, make a file called first_lab.js
6. Open the w1d2_lab directory in your text editor
7. Write the answers to the rest of this lab in the first_lab.js file.
8. Make sure you are in the same directory as the file, otherwise it won't run.
9. Press the up arrow to retrieve the node first_lab.js command. That way, you don't have to type it each time to run the file.

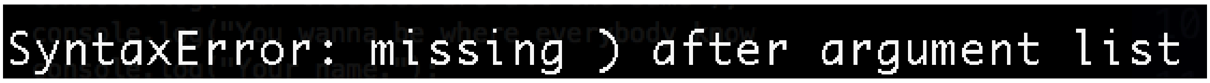
Errors

Fix the code by following the errors in Terminal

Cut and paste the following code into your text editor (the "Cheers" song lyrics). The code is broken -- there are **three errors**. You might already see the errors, but --

Run the code and read the error message in the terminal. Using information from the error message (line numbers, etc.), debug the code and make it work.

At the same time, write a comment below the code that specifies what *type* of error it was. Example, if you get this:



```
SyntaxError: missing ) after argument list
```

Write

```
// SyntaxError: missing ) after argument list
```

Cut and paste:

```
console.log("Making your way in the world today takes everything you've got.");
console.log("Taking a break from all your worries, sure would help a lot.");
console.log("Wouldn't you like to get away?");
console.log("Sometimes you want to go");
console.log("Where everybody knows your name,");
console.lo("and they're always glad you came.");
console.log("You wanna be where you can see,";
console.log("our troubles are all the same");
console.log("You wanna be where" "everybody knows");
console.log("Your name.");
```

Make it so you do not get any more errors!

Boolean Expressions

... and arithmetic

By just looking at the following expressions, determine in your mind whether or not each will evaluate to **true** or **false**

1. `999 > 999`
2. `999 == 999`
3. `999 != 999`
4. `-5 >= -4`
5. `100 <= -100`
6. `20 + 5 < 5`
7. `81 / 9 == 9`
8. `9 != 8 + 1`

Assignment Operator vs. Equality Operator

Write: What is the difference between:

the **assignment operator** `=`
and the **equality operator** `==`

While Loops: Reps and Reps

1 Write a *while* loop that will log in the console

`'Ginger chocolate honey patties'`
1000 times. You can test it out with a smaller number first, such as 10, and when that works, use 1000.

Make sure you do not run an infinite loop! If you do, close your Terminal. Oops!

2 Write another *while* loop that counts from 0 to 1000 and will log in the console the current loop number.

=> 0

=> 1

=> 2

=> 3

etc.

3 Write another *while* loop that prints a message to the console *and* concatenates the current loop number. Make it count from 0 to 1000.

=> "Current loop number is: 0"

=> "Current loop number is: 1"

=> "Current loop number is: 2"

=> "Current loop number is: 3"

NOTE: You should not need to see the 'correct answers' in this markdown for these loops. Either they work, or they don't. **Test** them thoroughly to make sure they are giving you the desired output.

For Loops

- Write a **for loop** that counts from 0 to 100 and console.logs each number.

- Write another **for loop** that counts from 7 to 635 (no more, no less!), and console.logs each number.

- Declare a variable `let start = 0,` declare a variable `const limit = 100`

  Write a for loop that counts from the value of `start` to the value of `limit`, using those variables in the **control panel** of the loop.

- Think of something in real life, or nature, or wherever / whenever that displays **looping** behavior.

Use a **for loop** to model the looping behavior of that thing.

Where does the loop begin? Where should it end? Does it simply count from one number to another? Or does it change or mutate data?

Conclusion

Congrats, you finished the lab! There is no need to submit this lab.