**Truthy and Falsy**

Let’s consider how non-boolean data types, like strings or numbers, are evaluated when checked inside a condition.

Sometimes, you’ll want to check if a variable exists and you won’t necessarily want it to equal a specific value — you’ll only check to see if the variable has been assigned a value.

Here’s an example:

let myVariable = 'I Exist!';  
   
if (myVariable) {  
   console.log(myVariable)  
} else {  
   console.log('The variable does not exist.')  
}

The code block in the if statement will run because myVariable has a *truthy* value; even though the value of myVariable is not explicitly the value true, when used in a boolean or conditional context, it evaluates to true because it has been assigned a non-falsy value.

So which values are *falsy*— or evaluate to false when checked as a condition? The list of falsy values includes:

* 0
* Empty strings like "" or ''
* null which represent when there is no value at all
* undefined which represent when a declared variable lacks a value
* NaN, or Not a Number

Here’s an example with numbers:

let numberOfApples = 0;  
   
if (numberOfApples){  
   console.log('Let us eat apples!');  
} else {  
   console.log('No apples left!');  
}  
   
// Prints 'No apples left!'

The condition evaluates to false because the value of the numberOfApples is 0. Since 0 is a falsy value, the code block in the else statement will run.

**Instructions**

**1.**

Change the value of wordCount so that it is truthy. This value should still be a number.

After you make this change and run your code, 'Great! You've started your work!' should log to the console.

Stuck? Get a hint

**2.**

Change the value of favoritePhrase so that it is still a string but falsy.

After you make this change and run your code, 'This string is definitely empty.' should log to the console.