1. Complete this function to convert the inputted **feet** value to meters and **return** the result. Conversion units: 39.37 inches = 1 meter;

```
function convertDistance(feet) {
    // convert feet to meters
}
```

2. Improve the previous function so that it can convert in both directions: feet-to-meters and meters-to-feet. You will need to provide some way for the function to know which conversion to do.

3. Make a function that:

    - takes one number as its input (argument)
    - squares that number (multiplies it by itself)
    - logs the answer to the Console. So, if you input 4, it logs 16. Run the function three times with different inputs.

4. Make another version of the previous function, so that:

    - instead of logging the answer, it returns the answer.
    - When you call the function, save the return value to a variable.
    - Log the variable to see if the function works. So, if you input 5, it returns 25. Run the function three times with different inputs.

5. Make a function that:

    - takes one number as its input (argument)
    - cubes the input number
    - returns the answer So, if you input 4, you get back 64. Run the function three times with different values.

6. Make another version of the previous function that:

    - takes one number as its input (argument)
    - if the number is even, it squares the number
    - but if the number is odd, it cubes the number
    - returns the answer. So, if you input 3, you get back 27. So, if you input 4, you get back 16. Run the function three times with different values.

7. The weight of anything on the moon is one-sixth its weight on earth. Make a function called **calcMoonWeight** that takes in an earth weight number as its argument, calculates the the moon

weight equivalent, and returns that number. So if the input is 180, the output is 30.

8. Declare a function called **introducePet**, that:

    - has four parameters: **pet**, **name**, **age** and **sound**, each of which gets set when the function is called.
    - uses string interploation to make a message that includes all four arguments.
    - returns a message, such that if the arguments are **cat**, **Fluffy**, **3** and **Meow**, the returned message is: **Meow! My name is Fluffy! I am a 3-year-old cat!**. Run the function three times, with different pet inputs each time.

9. Declare a function **introduceSelf** with two parameters: **name** and **city**. The function does not have a return value, but instead logs a message, such as: **Hi! My name is Jill. I am from Miami!**

10. Declare a function with two parameters, **num1** and **num2**. The function call passes in two arguments, both numbers.
    The function does the following math:

    - If the **num1** is greater than **num2**, subtract **num2** from **num1**
    - If **num1** is less than or equal to **num2**, add the numbers together.
      Return the answer. Run the function twice, once with the numbers being subtracted, the other with the numbers being added.

11. Given: two sides of a right triangle as global variables **sideA** and **sideB**

- Write a function with parameters **a** and **b**
    - Function uses the Pythagorean Theorem (**$a^2 + b^2 = c^2$**) to find the hypotenuse, **c**, of **a** and **b**.
    - Function returns **c**, the hyotenuse.
    - Call the function, passing in **sideA** and **sideB** as its two arguments.

12. Write a function that:

- takes in numbers of pennies, nickels, dimes and quarters
- calculates the total value of all coins
- returns the total as dollars and cents, to two decimal places and with dollar-sign

END Lab 03.01
SEE Lab 03.01 Solution