Lab 03.01 - Solution:

1. **calcMoonWeight** takes an earth weight as its parameter, converts that to moon weight by dividing by 6 and returns that value:

```
function calcMoonWeight(earthWt) {
    // convert earth wt on moon wt
    return earthWt / 6;
}

let earthWt = 300;
let moonWt = calcMoonWeight(earthWt);
console.log(moonWt); // 50
```

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

```
function convertFeetToMeters(feet) {
    // convert feet to meters
    return feet * 12 / 39.37;
}
```

3. Make a function called **squareIt** that:
    - takes in one number as its input
    - squares that number and returns the result So, if you input 4, it logs 16. Run the function three times with different inputs.

```
function squareNum(x) {
    return x ** 2; // or: x * x;
}

console.log(squareNum(4)); // 16
console.log(squareNum(5)); // 25
console.log(squareNum(6)); // 36
```

4. Make a function called **powerUp** that:
    - takes TWO positive integers as its inputs (arguments)
    - raises the first number to the power of the second number
    - returns the answer So, if you input (5, 3), you get back 125. Run the function three times with different inputs.

```
function powerUp(x,y) {
    return x ** y;
}
```

```
let pow44 = powerUp(4,4);
console.log(pow44); // 256

let pow53 = powerUp(5,3);
console.log(pow53); // 125

let pow62 = powerUp(6,2);
console.log(pow62); // 36
```

5. 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.

```
function squareOrCubeIt(x) {
    let answer = 0;
    // if x is even, dividing by 2 yields remainder of 0
    if(x % 2 == 0) { // if x is even, square it
        answer = x * x;
    } else { // x is odd, so cube it
        answer = x ** 3;
    }
    return answer;
}

let a = squareOrCubeIt(5); // cube it
console.log(a); // 125

let b = squareOrCubeIt(6); // square it
console.log(b); // 36

let c = squareOrCubeIt(7); // cube it
console.log(c); // 343
```

6. Declare a function called **introducePet**, that:
    - has four parameters: **pet**, **name**, **age** and **sound**
    - return 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.

```
function introducePet(pet, name, age, sound) {
    return `${sound}! My name is ${name}! I am a ${age}-year-old
${pet}`;
}
```

```
let petIntro1 = introducePet('cat', 'Fluffy', 3, 'Meow');
console.log(petIntro1);
// Meow! My name is Fluffy! I am a 3-year-old cat!

let petIntro2 = introducePet('dog', 'King', 2, 'Woof');
console.log(petIntro2);
// Woof! My name is King! I am a 2-year-old cat!

let petIntro3 = introducePet('canary', 'Tweety', 4, 'Chirp');
console.log(petIntro3);
// Chirp! My name is Tweety! I am a 4-year-old canary!
```

7. 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.

```
function addOrSubtractNums(num1, num2) {
    let answer = 0;
    if(num1 > num2) {
        answer = num1 - num2;
    } else {
        answer = num1 + num2;
    }
    return answer;
}

let answr1 = addOrSubtractNums(30, 20); // 10
let answr2 = addOrSubtractNums(20, 30); // 50
```

8. 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.

```
let sideA = 3;
let sideB = 4;

function findHypotenuse(a, b) {
    let cSquared = a**2 + b**2;
    let c = Math.sqrt(cSquared);
    return c;
}
```

```
    let c1 = findHypotenuse(sideA, sideB);
    console.log(c1); // 5

    let c2 = findHypotenuse(6, 8);
    console.log(c2); // 10

    let c3 = findHypotenuse(9, 12);
    console.log(c3); // 15
```

9. 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

```
    function countCoins(pennies, nickels, dimes, quarters)  {
        let cents = (pennies) + (nickels * 5) + (dimes * 10) + (quarters *
25);
        return '$' + ((cents / 100).toFixed(2));
    }

    let money1 = countCoins(250, 50, 25, 10); // 2.50 + 2.50 + 2.50 + 2.50
= $10.00
    console.log(money1);

    let money2 = countCoins(100, 100, 100, 100); // 1 + 5 + 10 + 25 =
$41.00
    console.log(money2);
```

10. Define a function that:

- takes the radius of a circle as its input
- calculates the area of the circle using the formula `A = πr2`
- returns (outputs) the area

```
    function findAreaOfCircle(r) {
        let area = Math.PI * r**2;
        return area;
    }

    let area1 = findAreaOfCircle(6);
    console.log(area1); // 113.09733552923255
```

END Lab 03.01 Solution
NEXT: Lesson 03.02