05.05 Lab Solution

Passing Arrays to Functions

```
    const nums = [4, 6, 10, 12, 18, 21, 24, 28, 33, 36];
```

1. Write a function that takes the nums array as its argument. The function returns the first odd number along with its index:

```
    function findFirstOddNum(arr) {

        for(let i = 0; i < arr.length; i++) {
            if(arr[i] % 2 == 1) { // if num is not even
                return arr[i];
            }
        }
        return 'no odd card found';

    }

    let firstOddity = findFirstOddNum(nums);
    console.log('firstOddity:', firstOddity);
```

2. Write a function that takes the nums array as its argument. Find the value and position of the first even number.

```
    function findFirstEvenNum(arr) {

        for(let i = 0; i < arr.length; i++) {
            if(arr[i] % 2 == 0) { // if num is even
                return `The first even number, ${arr[i]}, is at index
${i}.`;
            }
        }
        return 'no even number found';

    }

    let firstEven = findFirstEvenNum(cards);
    console.log('firstEven:', firstEven);
```

3. Write a function that takes the nums array as its argument. Return an array that only contains those numbers that are multiples of 3:

```
    function collectMultiplesOf3(arr) {

    let multiplesOf3 = [];

    for(let i = 0; i < arr.length; i++) {
        if(arr[i] % 3 == 0) { // if num is divisible by 3
            multiplesOf3.push(arr[i]);
        }
    }
    return multiplesOf3;

    }

    let multiplesOf3 = collectMultiplesOf3(nums);
    console.log('multiplesOf3:', multiplesOf3);
```

4. Write a function that takes the nums array as its argument. Return an array called multis that has all the values from nums multiplied by its index. So, if the number at index 3 is 5, the new value to push into multis is 5 x 3.

```
    function multiplyNumberByItsIndex(arr) {

    let multis = [];

    for(let i = 0; i < arr.length; i++) {
        let m = arr[i] * i;
        multis.push(m);
    }
    return multis;

    }

    let multi = multiplyNumberByItsIndex(nums);
    console.log('multi:', multi);
```

5. Write a function that takes in the fruits array and returns an object with the following properties

- fruitCount (number of words in the array)
- totChars (number of characters in the array)
- totVowels (number of vowels in the array)
- initVowels (number of words that start with a vowel)
- berrries (number of berries) HINT: The returned object does not need a name. The function call should be set equal to a variable, and in this way, the object will be assigned its name.

```
    const fruits = ["apple", "apricot", "banana",
"blackberry","blueberry", "boysenberry", "cherry", "cranberry",
"dragonfruit", "elderberry"];
```

6. Make the function work for any array, by passing in the veggies array and returning the same object:

```
    const veggies = ["arugula", "broccoli", "carrot", "celery","cucumber",
"daikon", "eggplant", "iceberg lettuce", "onion", "radish"];

    function makeWordObj(fruits) {

    }
```

BONUS:

7. Make a function that takes in a big string, such as a page or chapter of text as its argument, and returns an object with each unique word as a key the value of which is the number of times the word occus in the passage. So if the passage is:

"It is often said that that is the way that the proverbial cookie crumbles." The returned object should be:

```
    { "It": 1, "is": 2, "often": 1, "said": 1, "that": 3, "the": 2, "way":
1, "proverbial": 1, "cookie": 1, "crumbles": 1 }
```

NEXT: Lesson 05.06