Passing Arrays to Functions

In this lesson, we will be writing functions that take arrays as their arguments.

function #1: add article in front of word

The first function will take an array of words and add the article "a" or "an" in front of each word. If the word starts with "a", "e", "i" or "o", it should start with "an". The final product ("an apple", "a banana", etc.) will be saved to a new array.

The idea is that we can pass in any array of strings and the function will produce the same result, so we start by declaring two arrays:

1. Declare two arrays of strings:

```
    const fruits = ['apple', 'apricot', 'banana', 'blueberry',
'blueberry', 'cherry', 'elderberry', 'grape', 'kiwi', 'lemon', 'lime',
'mango', 'orange', 'papaya', 'peach', 'pear', 'pineapple', 'plum',
'raspberry', 'strawberry', 'tangerine', 'watermelon'];

    const cars = ['Acura', 'Alfa Romeo', 'Audi', 'BMW', 'Chevrolet',
'Dodge', 'Edsel', 'Eagle', 'Fiat', 'Ford', 'Honda', 'Kia', 'Infiniti',
'Isuzu', 'Mazda', 'Nissan', 'Opel', 'Porsche', 'Rolls Royce', 'Tesla',
'Toyota', 'Volvo'];
```

2. Declare the function and assign it a parameter, **arr**:

```
    function addArticle(arr) {
    }
```

3. The function needs a few variables:

- an empty array to hold the results
- a variable set equal to the article followed by a space
- a variable for checking if a word starts with a, e, i, o

```
```

```
function addArticle(arr) {
    let newArr = [];
    let article = 'a ';
    let vowels = 'aeio';
}
```

4. The function needs to check each item in the array, so set up a loop that iterates the array:

```
function addArticle(arr) {
    let newArr = [];
    let article = 'a ';
    let vowels = 'aeio';

    for(let i = 0; i < arr.length; i++) {
    }
}
```

5. Each time through the loop, save the current item, as well as the first letter of that item:

```
function addArticle(arr) {
    let newArr = [];
    let article = 'a ';
    let vowels = 'aeio';

    for(let i = 0; i < arr.length; i++) {
        let word = arr[i]; // apple
        let firstChar = word[0]; // a
    }

}
```

6. Add an if-statement that uses the **string.includes()** method to check if the first character, set to lowercase, is "a", "e", "i" or "u". If it is, set the value of **article** to "an":

```
    for(let i = 0; i < arr.length; i++>) {
        let word = arr[0]; // apple
        let firstChar = word[0]; // a
        if(vowels.includes(firstChar)) {
            article = 'an ';
        }
    }
```

7. Just for practice, comment out the if-else and switch to a ternary expression:

```
vowels.includes(firstChar) ? article='an ' : article='a ';
```

8. Concatenate **article** with the original word the original word, **arr[i]**, which may be capitalized. Push it into newArr, and return it:

```
    for(let i = 0; i < arr.length; i++>) {
        let word = arr[0]; // apple
        let firstChar = word[0]; // a
        if(vowels.includes(firstChar)) {
            article = "an ";
        }
        newArr.push(article + word);
    }
    return newArr;
```

Be sure to push *inside* the loop and return *outside* the loop. If the **return** is inside the loop, the loop will only run once, because **return** always ends a function.

9. Call the function twice, passing in a different array of strings each time. Save the return values to variables and log them:

```
let fruitsArr = addArticle(fruits);
console.log(fruitsArr);

let carsArr = addArticle(cars);
console.log(carsArr);
```

function #2: find target number in array

This next function has not just an array parameter, but also a number param. The function loops through the array, looking for the number.

- If the number is found, the function returns the index.
- If the number is NOT found, the function returns -1.

10. Declare an array of numbers, followed by a function:

```
const numsArr = [5, 7, 9, 12, 14, 16, 20, 25, 30, 40, 50];
```

11. Write the function with two parameters.

```
function findNum(arr, num) {
}
```

12. Iterate the array with a for loop:

```
function findNum(arr, num) {
    for(let i = 0; i < arr.length; i++) {
    }
}
```

13. Check if the current array item equals the target number, and if it does, return the index:

```
function findNum(arr, num) {
    for(let i = 0; i < arr.length; i++) {
        if(arr[i] === num) {
            return i;
        }
    }
}
```

14. If the loop ends without finding the target number, return -1:

```
function findNum(arr, num) {
    for(let i = 0; i < arr.length; i++) {
        if(arr[i] === num) {
            return i;
        }
    }
    return -1;
}
```

15. Call the function. The function returns a value, so set the call equal to a variable to "capture" the return value:

```
let find9 = findNum(numsArr, 12);
console.log(find9);  // 3

let find8 = findNum(numsArr, 123);
console.log(find8);  // -1
```

16. Instead of just returning the index, let's return a string, specifying the number being looked for and where it was found. Do this as a new function:

```
function findNumber(arr, num) {
    for(let i = 0; i < arr.length; i++) {
        if(num === arr[i]) {
            return `${num} found at index ${i}`;
```

```
            }
        }
        return `${num} not found`;
    }

    let find25 = findNumber(numsArr, 25);
    console.log(find25);  // 25 found at index 7

    let find250 = findNumber(numsArr, 250);
    console.log(find250);  // 250 not found
```

function #3: check array to see if all numbers are even

Let's try one more function that takes an array for its argument. The function checks each number to see if it is odd or even.

- If all numbers are even, the function returns true.
- If an odd number or float are found, the function returns false.

17. Declare three arrays of numbers:

- The first array contains only even numbers.
- The second array contains an odd number; the rest are even.
- The third array contains a float; the rest are even integers.

```
    let nums1 = [2, 4, 6, 8, 10];
    let nums2 = [2, 4, 7, 8, 10];
    let nums3 = [2, 4, 6.78, 10];
```

18. Write a function that expects an array to be passed to it:

```
    function allEven(arr) {
    }
```

19. Have the function iterate over the array:

```
    function allEven(arr) {
        for(let i = 0; i < arr.length; i++) {
        }
    }
```

20. Each time through the loop, run an if statement that uses the modulo operator **%** to divide the current number by 2. If the reminder is *not equal to* **!=** 0, the number is *not* even, the number is *not* even, so return false:

```
function allEven(arr) {
    for(let i = 0; i < arr.length; i++) {
        if(arr[i] % 2 != 0) {
            return false;
        }
    }
}
```

21. If the loop ends without finding a non-even number, all numbers in the array must be even, so return true:

```
function allEven(arr) {

    for(let i = 0; i < arr.length; i++) {
        if(arr[i] % 2 != 0) { // if num is not even
            return false;
        }
    }
    return true;
}
```

22. Call the function three times, once for each array; save the return value to a variable and log it:

```
let allEven1 = allEven(nums);
console.log('allEven1', allEven1); // true

let allEven2 = allEven(nums);
console.log('allEven2', allEven1); // false

let allEven3 = allEven(nums);
console.log('allEven3', allEven1); // false
```

END Lesson 05.05 NEXT: Lab 05.05 Lesson 05.06