Loops

1. Write a function that takes an array of numbers and returns the sum of all the numbers in the array.

```
console.log(sum([1, 2, 3, 4, 5])); // Output: 15
```

2. Write a function that takes an array of strings and returns an array of the strings with all vowels removed. (you can user string.replace)

```
console.log(removeVowels(['hello', 'world'])); // Output: ['hll', 'wrd']
```

3. Write a function that takes an array of objects and returns an array of the objects sorted by a given property in ascending order.

```
const books = [
 { title: 'The Great Gatsby', pages: 180 },
 { title: 'To Kill a Mockingbird', pages: 280 },
```

```
  { title: 'Pride and Prejudice', pages: 150 }
];

console.log(sortByProperty(books, 'pages'));
/* Output:
[
  { title: 'Pride and Prejudice', pages: 150 },
  { title: 'The Great Gatsby', pages: 180 },
  { title: 'To Kill a Mockingbird', pages: 280 }
]
*/
```

4. Write a function that takes a string and returns a new string with all of the characters in the original string reversed.

```
console.log(reverseString('hello')); // Output: 'olleh'
```

5. Write a function that takes an array of numbers and returns an array of the numbers squared. make it with an arrow function

```
console.log(square([1, 2, 3, 4])); // Output: [1, 4, 9, 16]
```

6. Write a function that takes an array of objects and a property name, and returns an array of the values of the specified property for each object.

```
 const books = [ { title: 'The Great Gatsby', pages: 180 }, { title: 'To Kill a Mockingbird', pages: 280 }, { title: 'Pride and Prejudice', pages: 150 } ]; console.log(getPropertyValues(books, 'pages')); // Output: [180, 280, 150]
```

7. Write a function that takes an array of numbers and returns the product of all the numbers in the array.

```
console.log(product([1, 2, 3, 4, 5])); // Output: 120
```

8. Write a function that takes an array of strings and returns an array of the strings sorted alphabetically.

```
console.log(sortAlphabetically(['c', 'a', 'b'])); // Output: ['a', 'b', 'c']
```

Methods:

1. Easy: Write a function that takes in a string and returns the string with all vowels removed.

   For example,
   `removeVowels("hello") //output hll.`
   This can be done using the replace method and a regular expression that matches vowels.

2. Medium: Write a function that takes in two numbers and returns their sum.

   For example,

```
add(2, 3) //output 5.
```

3. Medium: Write a function that takes in an array of numbers and returns the average of the numbers.

   For example,

```
average([1, 2, 3]) //output 2.
```

This can be done by using the reduce method to sum up all the numbers in the array and then dividing by the length of the array.

4. Medium: Write a function that takes in a string and returns the string with all characters in reverse order.

   For example,

```
reverseString("hello") //output olleh.
```

This can be done using the split, reverse, and join methods.

5. Hard: Write a function that takes in an array of numbers and returns an array of the squares of those numbers.

   For example,

```
squareNumbers([1, 2, 3]) //output [1, 4, 9].
```

This can be done using the map method and an arrow function that squares each number.

6. Easy: Write a function that takes in a string and returns the string with all occurrences of the letter "a" replaced with the letter "z".

   For example:

```
aToZ("I am an Armadillo") //output "I zm zn Zrmzdilo"
```

Use the replace method to accomplish this.

7. Medium: Write a function that takes in a string and returns an array of all the words in the string.

   For example:

```
stringToArray("hello world, I am Armadillo") //output
["hello", "world", "I", "am", "Armadillo"]
```

Use the split method to accomplish this.

8. Hard: Write a function that takes in an array of strings and returns an array of the strings sorted by length, with the shortest strings first.

   For example:

```
sortedStrings("hello world, I am Armadillo")//output
["I", "am", "hello", "world", "Armadillo"]
```

   Use the sort method to accomplish this.

9. Easy: Write a function that takes in a string and returns the string with the first letter capitalized.

   For example:

```
upperCase("hello") //output "Hello"
```

Use the charAt and toUpperCase methods to accomplish this.

10. Medium: Write a function that takes in an array of numbers and returns the minimum value in the array.

    For example:

    ```
    minNum(5,8,3,78,2,1,8,5) //output 1
    ```

    Use the Math.min method and the apply method to accomplish this.

11. Medium: Write a function that takes in an array of strings and returns a string containing all the elements of the array, separated by a space.

    For Example:

```
separator(["hello", "world"]) // output hello world
```

Use the join method to accomplish this.

12. Hard: Write a function that takes in an array of numbers and returns the second-largest number in the array.

    For example:

```
secondLargest([1,2,3,4,5,6,7]) //output 6
```

Use the sort method to accomplish this.

13. Medium: Write a function that takes in an array of numbers and returns the sum of all the elements in the array.

    For example:

```
sumOf([1,2,3,4,5]) // output 15
```

Use the reduce method to accomplish this.

14. Medium: Write a function that takes in a string and returns an array of all the words in the string that have an even length.

    For example

```
arrOfAllWords("hi world") //output ["hi"]
```

Use the split and filter methods to accomplish this.

15. Hard: Write a function that takes in an array of strings and returns an array of the strings sorted in alphabetical order, with the exception of words that begin with the letter "z", which should be placed at the end of the array.

    For example:

```
withoutZ(["hello", "world", "zero", "one"]) //output
["hello", "one","zero", "world"]
```

Use the sort method to accomplish this.