JavaScript Code Exercises 4

Exercise 1: Sum All Numbers in a Range

Write a function that takes an array of two numbers as input and returns the sum of all the numbers between them, inclusive.

```
function sumAll(arr) {
  let min = Math.min(...arr);
  let max = Math.max(...arr);
  let sum = 0;
  for (let i = min; i <= max; i++) {
```

```
    sum += i;
  }
  return sum;
}

console.log(sumAll([1, 4])); // Output: 10
```

In this example, we first use the Math.min() and Math.max() functions to find the minimum and maximum values in the input array, respectively. We then use a for loop to iterate from the minimum value to the maximum value, adding each number to the sum variable. Finally, we return the sum.

Exercise 2: Diff Two Arrays

Write a function that takes two arrays as input and returns a new array containing the elements that are unique to each array.

```
function diffArray(arr1, arr2) {
  return arr1.filter((item) => !arr2.includes(item))
    .concat(arr2.filter((item) =>
!arr1.includes(item)));
}

console.log(diffArray([1, 2, 3, 4], [3, 4, 5, 6])); //
Output: [1, 2, 5, 6]
```

In this example, we first use the filter() method to create a new array that contains only the elements from arr1 that are not in arr2. We then concatenate this array with a similar array created from arr2. The filter() method with the includes() method inside of it allows us to filter out the duplicate elements between both arrays. Finally, we return the concatenated array.

Exercise 3: Seek and Destroy

Write a function that takes an arbitrary number of arguments, the first of which is an array, and removes all other arguments from the array.

```
function destroyer(arr, ...args) {
  return arr.filter((item) => !args.includes(item));
}

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

In this example, we use the rest parameter syntax to allow the function to accept an arbitrary number of arguments. We then use the filter() method to create a new array that contains only the elements from arr that are not in args. Finally, we return the filtered array.

Exercise 4: Wherefore art thou

Write a function that takes two arrays of objects as input, and returns an array of all the objects in the first array that have matching property-value pairs to the objects in the second array.

```
function whatIsInAName(collection, source) {
  return collection.filter((obj) => {
    for (let key in source) {
      if (!obj.hasOwnProperty(key) || obj[key] !==
source[key]) {
        return false;
      }
    }
```

```
    return true;
  });
}

console.log(whatIsInAName([{name: "John", age: 25},
{name: "Jane", age: 27}], {age: 25})); // Output:
[{name: "John", age: 25}]
```

In this example, we use the filter() method to create a new array that contains only the objects from collection that match the property-value pairs in source. We loop through each key in source and check if the object in collection has that key