JavaScript Coding Examples

JavaScript coding exercises with full code and explanations:

Write a function that takes an array of integers and returns the average of the numbers. **7**

Write a function that takes two arrays of integers and returns a new array with only the numbers that are in both arrays. **8**

Write a function to calculate the factorial of a given number.

```
function factorial(n) {
  if (n === 0) {
    return 1;
  } else {
    return n * factorial(n - 1);
  }
}
console.log(factorial(5)); // Output: 120
```

This function uses recursion to calculate the factorial of a given number. If the number is 0, the function returns 1. Otherwise, it multiplies the number by the factorial of the number minus 1.

Write a function that takes an array of integers and returns the sum of all the even numbers.

```
function sumOfEvenNumbers(arr) {
  let sum = 0;
  for (let i = 0; i < arr.length; i++) {
    if (arr[i] % 2 === 0) {
      sum += arr[i];
```

```
    }
  }
  return sum;
}

console.log(sumOfEvenNumbers([1, 2, 3, 4, 5, 6])); //
Output: 12
```

This function uses a for loop to iterate through the array of integers. If a number is even, it adds it to the sum.

Write a function that takes a string and returns the reversed version of the string.

```
function reverseString(str) {
  return str.split("").reverse().join("");
}
console.log(reverseString("hello")); // Output: "olleh"
```

This function splits the string into an array of individual characters, reverses the order of the characters, and joins them back together into a string.

Write a function that takes an array of strings and returns the longest string in the array.

```
function longestString(arr) {
  let longest = "";
  for (let i = 0; i < arr.length; i++) {
    if (arr[i].length > longest.length) {
      longest = arr[i];
    }
```

```
  }
  return longest;
}

console.log(longestString(["cat", "dog", "elephant"]));
// Output: "elephant"
```

This function uses a for loop to iterate through the array of strings. If a string is longer than the current longest string, it becomes the new longest string.

Write a function that takes a string and returns the number of vowels in the string.

```
function countVowels(str) {
  const vowels = "aeiou";
  let count = 0;
  for (let i = 0; i < str.length; i++) {
    if (vowels.includes(str[i])) {
      count++;
    }
  }
  return count;
}
console.log(countVowels("hello")); // Output: 2
```

This function uses a for loop to iterate through the string. If a character is a vowel, it increments the count.

Write a function that takes an array of integers and returns the second largest number in the array.

```
function secondLargestNumber(arr) {
  let largest = arr[0];
  let secondLargest = arr[0];
  for (let i = 0; i < arr.length; i++) {
    if (arr[i] > largest) {
      secondLargest = largest;
      largest = arr[i];
    } else if (arr[i] > secondLargest && arr[i] !==
largest) {
      secondLargest = arr[i];
    }
  }
  return secondLargest;
}
console.log(secondLargestNumber([1, 2, 3, 4, 5])); //
Output: 4
```

This function initializes the largest and second largest numbers to the first element in the array. It then iterates through the array, updating the largest and second largest numbers as necessary. If a number is larger than the current largest number, it becomes the new largest number and the current largest number becomes the second largest number. If a number is larger than the current second largest number but not the largest number, it becomes the new second largest number.

Write a function that takes an array of integers and returns a new array with only the unique numbers.

```
function uniqueNumbers(arr) {
  let uniqueArr = [];
  for (let i = 0; i < arr.length; i++) {
    if (!uniqueArr.includes(arr[i])) {
      uniqueArr.push(arr[i]);
    }
  }
  return uniqueArr;
}

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

This function initializes an empty array for the unique numbers. It then iterates through the array of integers and checks if each number is already in the unique array. If not, it adds the number to the unique array.

Write a function that takes a string and returns true if the string is a palindrome and false if it is not.

```
function isPalindrome(str) {
  const reversedStr = str.split("").reverse().join("");
  return str === reversedStr;
}
```

console.log(isPalindrome("racecar")); // Output: true
This function splits the string into an array of individual characters, reverses the order of the characters, and joins them back together into a new string. It then checks if the original string is the same as the reversed string.

Write a function that takes an array of integers and returns the average of the numbers.

```
function average(arr) {
  let sum = 0;
  for (let i = 0; i < arr.length; i++) {
    sum += arr[i];
  }
  return sum / arr.length;
}

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

This function uses a for loop to iterate through the array of integers and add up the numbers. It then divides the sum by the length of the array to get the average.

Write a function that takes two arrays of integers and returns a new array with only the numbers that are in both arrays.

```
function commonNumbers(arr1, arr2) {
  let commonArr = [];
  for (let i = 0; i < arr1.length; i++) {
    if (arr2.includes(arr1[i]) &&
!commonArr.includes(arr1[i])) {
      commonArr.push(arr1[i]);
    }
  }
  return commonArr;
}

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

This function initializes an empty array for the common numbers. It then iterates through the first array of integers and checks if each number is in the second array and is not already in the common array. If it is, it adds the number to the common array.