






# LEARN JAVASCRIPT

 Boost Your JavaScript Skills:  
Master Functions with These 10  
Practical Exercises! 

**Coding EXERCISES explanations!**


<b>Exercise: Write a simple JavaScript function to print "Hello, World!".</b>	<b>2</b>
<b>Exercise: Create a function that takes two numbers as arguments and returns their sum.</b>	<b>2</b>
<b>Exercise: Write a function that takes a string as an argument and returns the string in reverse.</b>	<b>3</b>
<b>Exercise: Create a function that calculates the factorial of a number.</b>	<b>3</b>
<b>Exercise: Write a JavaScript function that checks whether a passed string is a palindrome or not.</b>	<b>4</b>
<b>Exercise: Create a function that removes duplicates from an array.</b>	<b>4</b>
<b>Exercise: Write a function to convert Celsius to Fahrenheit.</b>	<b>4</b>
<b>Exercise: Create a function that finds the largest number in an array.</b>	<b>5</b>
<b>Exercise: Write a function that checks if a number is even.</b>	<b>5</b>
<b>Exercise: Create a function that takes an array of numbers and returns a new array containing only the positive numbers.</b>	<b>6</b>

Inside the Challenge:

-  Functional Programming Basics
-  Recursion and Higher-Order Functions
-  Real-World Scenarios to Apply Your Knowledge

Learn more about JavaScript with Examples and Source Code Laurence Svekis  
Courses <https://basescripts.com/>

## Complete Explanations and Code Included

Embark on these challenges to refine your understanding of one of JavaScript's core concepts - functions. From simple operations to complex logical constructs, these exercises will help you think more functionally! 

### Exercise: Write a simple JavaScript function to print "Hello, World!".

```
function sayHello() {  
  console.log("Hello, World!");  
}  
sayHello();
```

Explanation: This exercise demonstrates a basic function declaration in JavaScript. The function `sayHello` when called, executes the `console.log` statement to print "Hello, World!".

### Exercise: Create a function that takes two numbers as arguments and returns their sum.

```
function add(a, b) {  
  return a + b;  
}  
console.log(add(2, 3)); // Output: 5
```

Explanation: The function `add` takes two parameters and returns their sum. When `add(2, 3)` is called, it returns 5.

**Exercise: Write a function that takes a string as an argument and returns the string in reverse.**

```
function reverseString(str) {  
  return str.split('').reverse().join('');  
}
```

```
console.log(reverseString("Hello")); // Output: "olleH"
```

Explanation: This function reverses a string by splitting it into an array of characters (`split('')`), reversing the array (`reverse()`), and then joining the characters back into a string (`join('')`).

**Exercise: Create a function that calculates the factorial of a number.**

```
function factorial(num) {  
  if (num === 0 || num === 1) {  
    return 1;  
  }  
  return num * factorial(num - 1);  
}
```

```
console.log(factorial(5)); // Output: 120
```

Explanation: This is a recursive function that calculates the factorial of a number. The base case returns 1 when num is 0 or 1. Otherwise, it calls itself with num - 1.

**Exercise: Write a JavaScript function that checks whether a passed string is a palindrome or not.**

```
function isPalindrome(str) {  
  return str === str.split('').reverse().join('');  
}  
  
console.log(isPalindrome("madam")); // Output: true  
console.log(isPalindrome("hello")); // Output: false
```

Explanation: This function checks if a string is a palindrome by comparing the string with its reversed version. It uses the same string reversal method as in exercise 3.

**Exercise: Create a function that removes duplicates from an array.**

```
function removeDuplicates(arr) {  
  return [...new Set(arr)];  
}  
  
console.log(removeDuplicates([1, 2, 2, 3])); // Output: [1, 2, 3]
```

Explanation: This function removes duplicates from an array using the Set object, which only stores unique values, and then converts it back to an array.

**Exercise: Write a function to convert Celsius to Fahrenheit.**

```
function celsiusToFahrenheit(celsius) {  
  return celsius * 9/5 + 32;
```

```
}  
console.log(celsiusToFahrenheit(0)); // Output: 32
```

Explanation: The function converts Celsius to Fahrenheit using the formula  $F = C * 9/5 + 32$ .

**Exercise: Create a function that finds the largest number in an array.**

```
function findLargest(arr) {  
  return Math.max(...arr);  
}  
console.log(findLargest([1, 2, 3, 4])); // Output: 4
```

Explanation: This function finds the largest number in an array using the `Math.max()` function along with the spread operator to pass all array elements as individual arguments.

**Exercise: Write a function that checks if a number is even.**

```
function isEven(num) {  
  return num % 2 === 0;  
}  
console.log(isEven(4)); // Output: true  
console.log(isEven(5)); // Output: false
```

Explanation: This function checks if a number is even by using the modulus operator (%). If the number is divisible by 2 with no remainder, it returns true, indicating the number is even.

Exercise: Create a function that takes an array of numbers and returns a new array containing only the positive numbers.

```
function filterPositiveNumbers(arr) {  
  return arr.filter(num => num > 0);  
}
```

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

Explanation: This function uses the `filter()` method to create a new array with only positive numbers. The `filter()` method creates a new array with all elements that pass the test implemented by the provided function, in this case, `num > 0`.

These exercises cover various aspects of JavaScript functions, demonstrating different uses and capabilities of functions in JavaScript, from basic operations to more advanced concepts like recursion and higher-order functions.