

# JavaScript-Functions

---

## Functions in JavaScript

In JavaScript, a function is a block of code that can be executed multiple times from different parts of your program. It's a way to encapsulate a specific task or set of instructions that you want to repeat.

## Function Declaration

A function is declared using the `function` keyword followed by the name of the function and its parameters in parentheses.

```
function add(x, y) {  
  return x + y;  
}
```

In this example, we have a function named `add` that takes two parameters, `x` and `y`.

## Function Invocation

To execute a function, you call it by its name followed by the required arguments in parentheses.

```
console.log(add(5, 7)); // Output: 12
```

In this example, we're calling the `add` function with arguments `5` and `7`, and logging the result to the console.

## Function Arguments

Functions can take multiple arguments, which are passed in when the function is called.

```
function greet(name, age) {  
  return `Hello, ${name}! You are ${age} years old.`;  
}  
  
console.log(greet('John Doe', 30)); // Output: Hello, John Doe! You are 30 years old.
```

In this example, we have a function named `greet` that takes two arguments, `name` and `age`.

## Function Return Values

Functions can return values using the `return` statement.

```
function getFullName(first, last) {  
  return `${first} ${last}`;  
}  
  
const fullName = getFullName('John', 'Doe');  
console.log(fullName); // Output: John Doe
```

In this example, we have a function named `getFullName` that returns a concatenated string of the first and last names.

## Function Expressions

Functions can also be defined using function expressions.

```
const add = function(x, y) {  
  return x + y;  
};  
  
console.log(add(5, 7)); // Output: 12
```

In this example, we have a function expression that defines the `add` function.

## Arrow Functions

Functions can also be defined using arrow functions.

```
const add = (x, y) => {  
  return x + y;  
};  
  
console.log(add(5, 7)); // Output: 12
```

In this example, we have an arrow function that defines the `add` function.

## Function Properties

Functions can also have properties attached to them.

```
function add(x, y) {  
  this.message = 'Added successfully!';  
}  
  
const result = add(5, 7);  
console.log(result.message); // Output: Added successfully!
```

In this example, we have a function named `add` that has a property called `message`.

---

*Curated by Brajesh Kumar*