

JS - Functions

1. Basic Function Declaration

Functions are blocks of reusable code that can be called to perform a specific task. You can declare a function using the function keyword:

```
// Basic function declaration
function greet() {
    console.log("Hello, World!");
}

// Call the function
greet(); // Output: Hello, World!
```

2. Function with Parameters

Functions can accept input values known as parameters, which can be used within the function body:

```
// Function with parameters
function greet(name) {
    console.log("Hello, " + name + "!");
}

// Call the function with an argument
greet("Alice"); // Output: Hello, Alice!
```

3. Function with Return Value

A function can return a value to the caller using the return statement:

```
// Function with a return value
function add(a, b) {
    return a + b;
}

// Call the function and store the result
let result = add(5, 3);
console.log(result); // Output: 8
```

4. Function Expression

Functions can also be assigned to variables, creating function expressions:

```
// Function expression
let greet = function(name) {
    return "Hello, " + name + "!";
};

// Call the function expression
console.log(greet("Bob")); // Output: Hello, Bob!
```

5. Arrow Functions

Arrow functions are a shorthand syntax for function expressions. They are more concise and do not have their own this value:

```
// Arrow function
let add = (a, b) => a + b;

// Call the arrow function
console.log(add(2, 4)); // Output: 6
```

6. Function with Default Parameters

You can set default values for function parameters:

```
// Function with default parameters
function greet(name = "Guest") {
    console.log("Hello, " + name + "!");
}

// Call the function without passing an argument
greet(); // Output: Hello, Guest!

// Call the function with a custom argument
greet("Charlie"); // Output: Hello, Charlie!
```

7. Anonymous Functions

An anonymous function is a function without a name, often used for short, one-time tasks:

```
// Anonymous function used in an array method
let numbers = [1, 2, 3, 4, 5];
let squares = numbers.map(function(num) {
    return num * num;
});

console.log(squares); // Output: [1, 4, 9, 16, 25]
```

8. Function as Arguments

Functions can be passed as arguments to other functions:

```
// Function that accepts another function as an argument
function performOperation(a, b, operation) {
    return operation(a, b);
}

// Add function
function add(a, b) {
    return a + b;
}

// Call the function with 'add' as an argument
console.log(performOperation(5, 3, add)); // Output: 8
```

9. Summary of Functions

Type	Description
Basic function	Declares a reusable block of code.
Function parameters	Functions can accept inputs as parameters.
Return values	Functions can return a value to the caller.
Function expression	Functions assigned to variables, often used as callbacks.
Arrow functions	Concise syntax for function expressions, without their own 'this'.
Default parameters	Functions can have default values for parameters.
Anonymous functions	Functions without names, used for one-time operations.
Functions as args	Functions can be passed as arguments to other functions.