

# JavaScript Functions, Arguments, Scope & Hoisting

## 1. Functions in JavaScript

Functions are reusable blocks of code that perform a specific task. Functions can be declared or assigned to variables.

```
function greet() {  
  console.log("Hello!");  
}  
  
const sayHi = function() {  
  console.log("Hi!");  
};  
  
const arrowGreet = () => {  
  console.log("Greetings!");  
};
```

## 2. Function Arguments & Return

Functions can accept arguments and return values.

```
function add(a, b) {  
  return a + b;  
}  
  
console.log(add(3, 4)); // 7
```

## 3. Scope in JavaScript

Scope determines where variables are accessible. JavaScript has global, function, and block scope.

```
let globalVar = "I am global";  
  
function testScope() {  
  let localVar = "I am local";  
  console.log(globalVar); // Accessible  
  console.log(localVar);  // Accessible  
}  
  
testScope();  
// console.log(localVar); // Error: not defined
```

## 4. Hoisting

JavaScript moves declarations to the top of the scope during execution. This is called hoisting.

```
greet(); // Works due to hoisting
```

## JavaScript Functions, Arguments, Scope & Hoisting

---

```
function greet() {  
  console.log("Hello");  
}  
  
// This won't work:  
console.log(msg); // undefined  
var msg = "Hi";
```