Return Values in JavaScript

Return values are what functions send back to the code that called them. They're how functions communicate their results.

Basic Return Values

When a function executes a return statement, it immediately exits and sends the specified value back to the caller:

```
function add(a, b) {
  return a + b;
}
let result = add(5, 3);  // result is 8
```

Without an explicit return statement, functions return undefined by default:

```
function noReturn() {
 let x = 5;
}
console.log(noReturn());  // undefined
```

Returning Different Data Types

Functions can return any JavaScript value:

```
// Return a number
function getAge() {
  return 25;
}

// Return a string
function greet() {
  return "Hello!";
}

// Return a boolean
```

```
function isEven(num) {
  return num % 2 === 0;
}

// Return an object
function createUser() {
  return { name: "Alice", age: 30 };
}

// Return an array
function getColors() {
  return ["red", "blue", "green"];
}
```

Early Returns

You can use return statements to exit a function early:

```
function checkAge(age) {
  if (age < 0) {
    return "Invalid age";
  }
  if (age < 18) {
    return "Minor";
  }
  return "Adult";
}
```

Using Return Values

Return values are commonly used in expressions, assignments, and as arguments to other functions:

```
function multiply(a, b) {
  return a * b;
}

// Used in expressions
let total = multiply(4, 5) + 10;  // 30
```

```
// Passed to other functions
console.log(multiply(3, 7));  // 21

// Used in conditionals
if (multiply(2, 3) > 5) {
  console.log("Greater than 5");
}
```

Arrow Functions

Arrow functions have implicit returns for single expressions:

```
// Explicit return
const square = (x) => {
  return x * x;
};

// Implicit return (no curly braces needed)
const cube = (x) => x * x * x;

console.log(cube(3));  // 27
```

Built-in Functions vs User-Defined Functions

Both built-in and user-defined functions work the same way with return values:

```
// Built-in function
let upperCase = "hello".toUpperCase();  // "HELLO"
let mathResult = Math.max(5, 10, 3);    // 10

// User-defined function
function findMax(a, b) {
  return a > b ? a : b;
}

let myMax = findMax(5, 10);  // 10
```

The key difference is that you control what your functions return, while built-in functions have predefined behavior. Both allow you to capture and use their return values in the same ways.