



# 9 must-know shorthands

for **JavaScript**





# Destructuring Assignment

Allows you to **extract values** from arrays or objects into **separate variables**.



```
const [a, b] = [1, 2];
```

destructuring  
array

```
const { x, y } = { x: 1, y: 2 };
```

destructuring  
object

# Object Property Shorthand

**Assigns** variables with the **same name** as object properties automatically.



```
const x = 1;
```

```
const y = 2;
```

```
const obj = { x, y };
```





# Object Property Value Shorthand



```
const x = 1;  
const y = 2;  
const obj = { x, y };
```

Assigns object properties with the **same name** as variables automatically.

# Default Parameter Values

Sets **default** values for function parameters **if no argument** is provided.



```
function foo(x = 10) {  
  console.log(x);  
}  
foo(); // Output: 10  
foo(5); // Output: 5
```





# Arrow Functions

Concise syntax for **defining anonymous functions**.



```
const add = (a, b) => a + b;
```

# Ternary Operator

Evaluates a **condition** and returns one of two values based on the result.



```
const result = condition ? value1 : value2;
```





# Short Circuit Evaluation

Uses logical operators to conditionally **assign** values based on **truthiness**.



```
const value = a || b;  
const value = a && b;
```

# Template Literals

Enables string interpolation and multiline strings using backticks.



```
const name = "John";  
const greeting = `Hello, ${name}!`;
```





# Spread Syntax

**Expands** arrays or objects into **individual** elements or properties.



```
const arr = [1, 2, 3];  
const newArr = [...arr, 3, 4, 5];  
console.log(newArr);  
// Output: [1, 2, 3, 4, 5]  
  
const obj = { x: 1, y: 2 };  
const newObj = { ...obj, z: 3 };  
console.log(newObj);  
// Output: { "x": 1, "y": 2, "z": 3 }
```







codewith**sloba**.com

Get a weekly digest of my tips and  
tutorials by subscribing now.