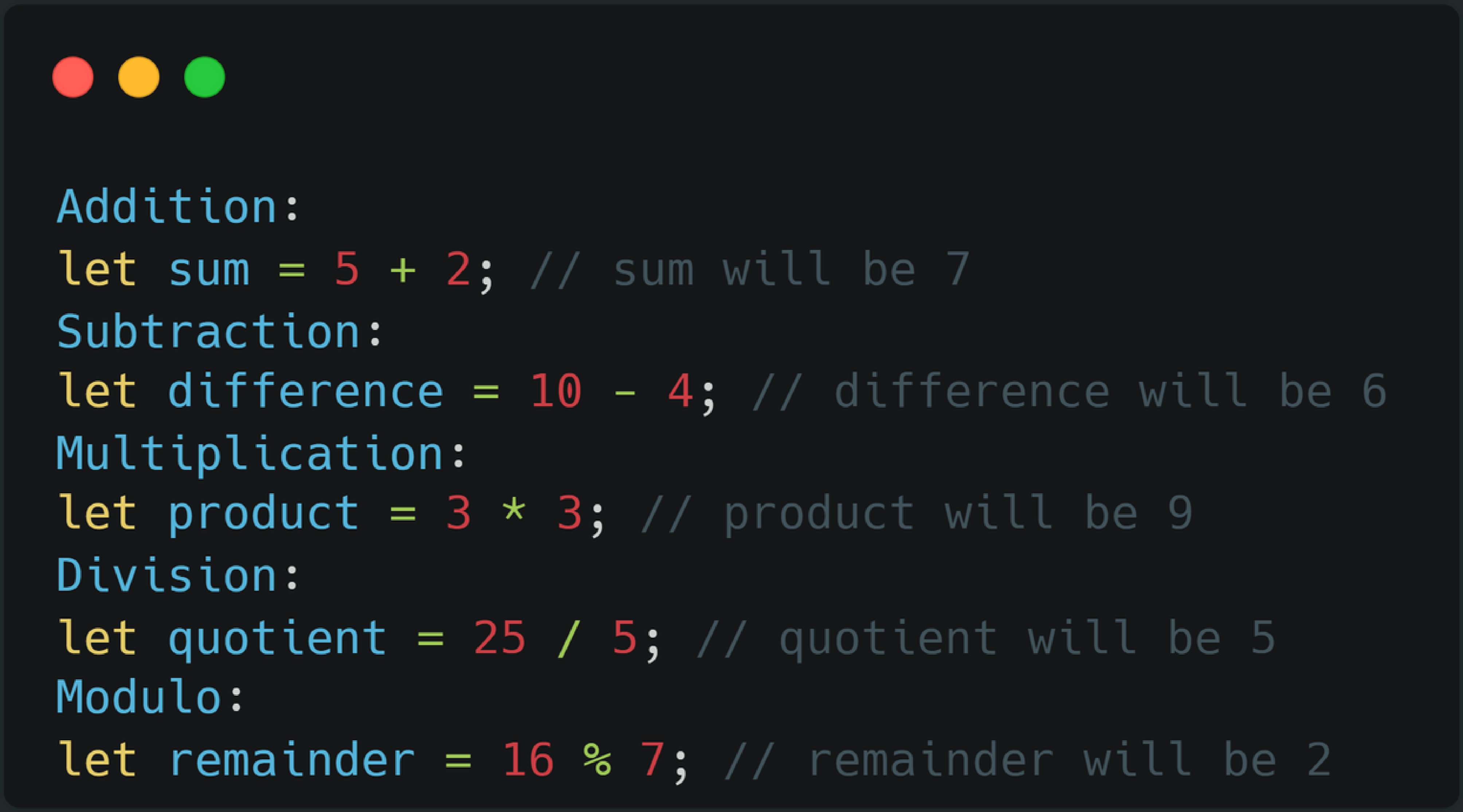
Understanding JavaScript Operators through Example

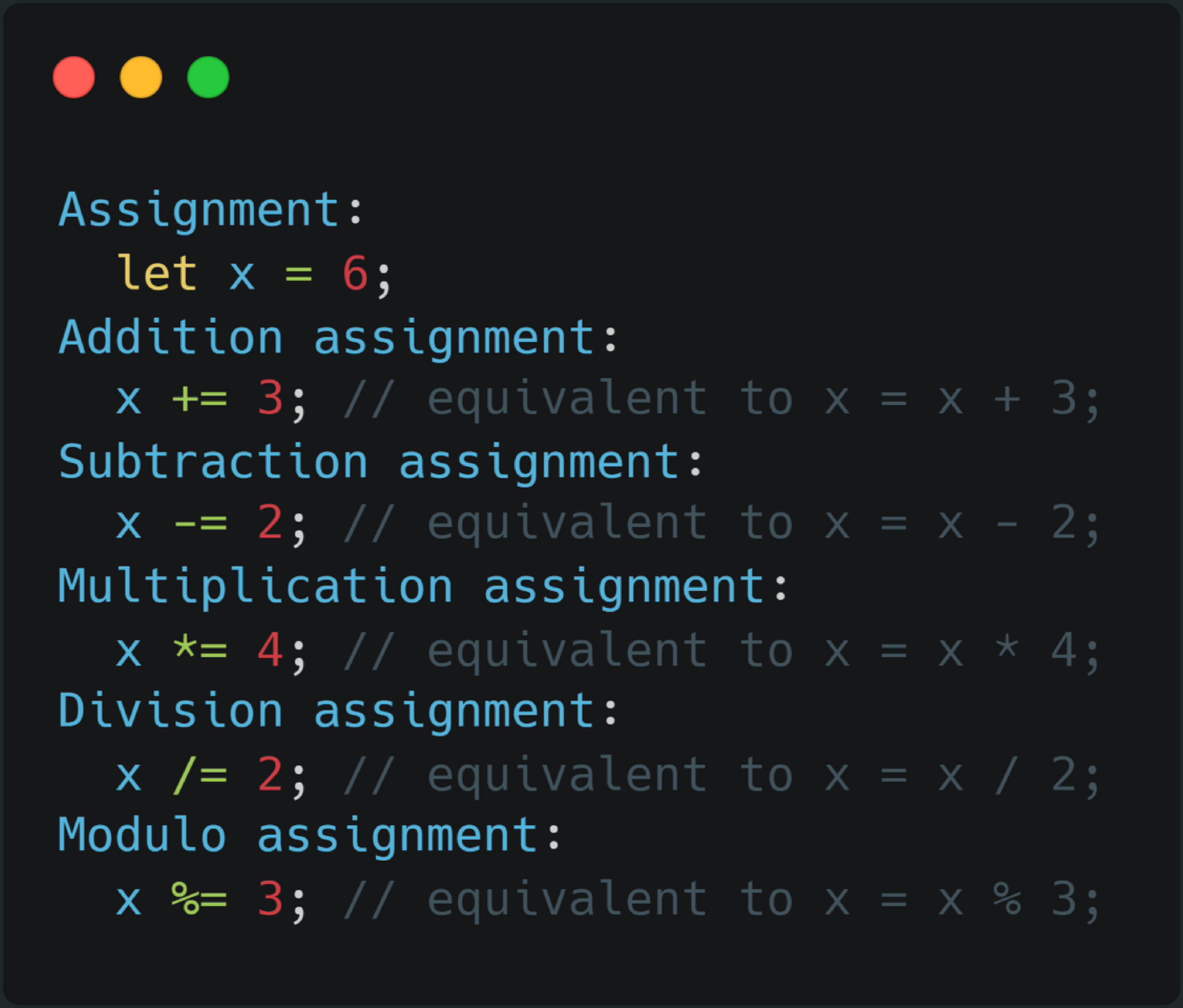
- Arithmetic Operators
- Comparison Operators
- Unary Operators
- String Operators
- Assignment Operators
- Logical Operators
- Type Operators

Arithmetic Operators



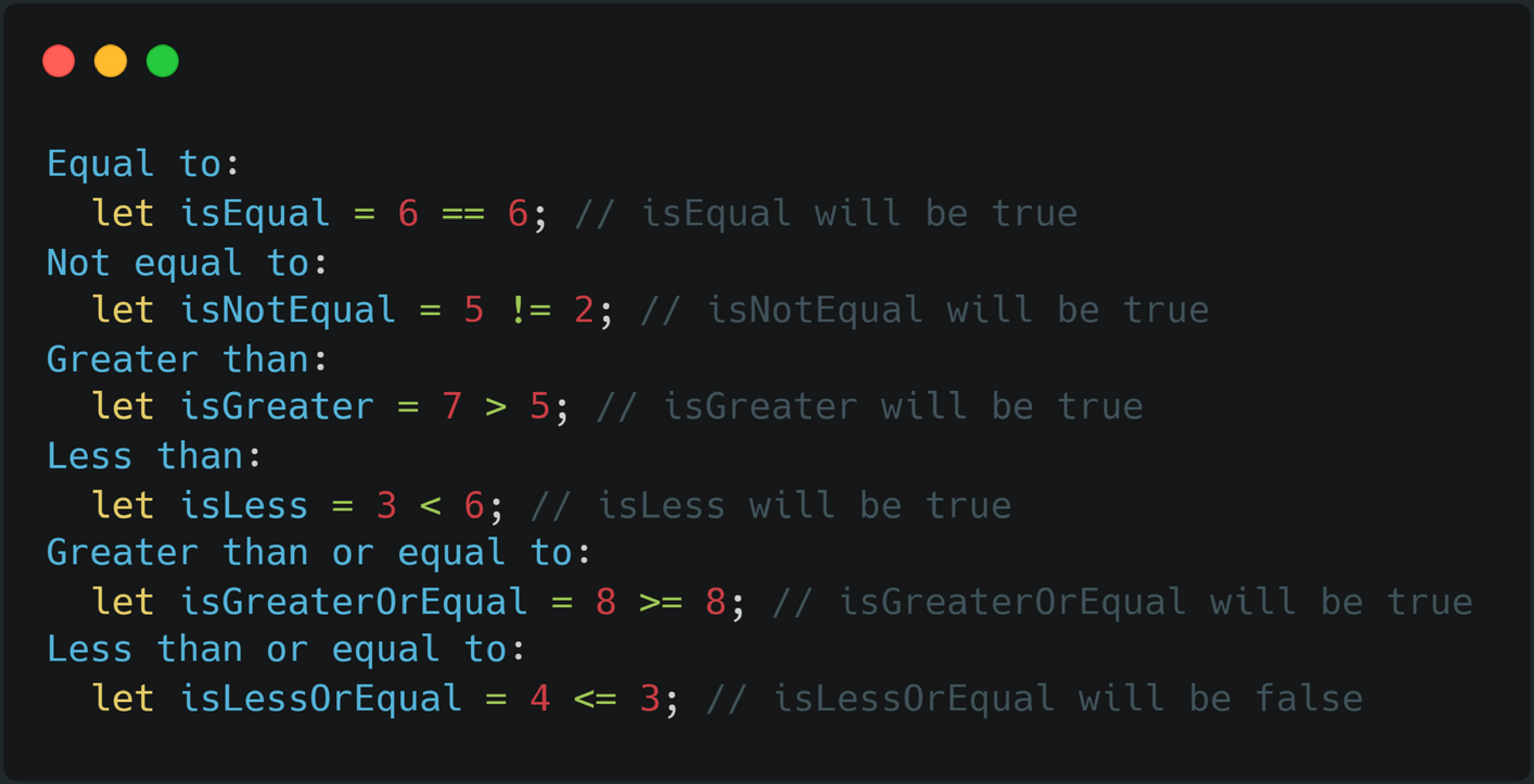
```
Addition:
let sum = 5 + 2; // sum will be 7
Subtraction:
let difference = 10 - 4; // difference will be 6
Multiplication:
let product = 3 * 3; // product will be 9
Division:
let quotient = 25 / 5; // quotient will be 5
Modulo:
let remainder = 16 % 7; // remainder will be 2
```

Assignment Operators



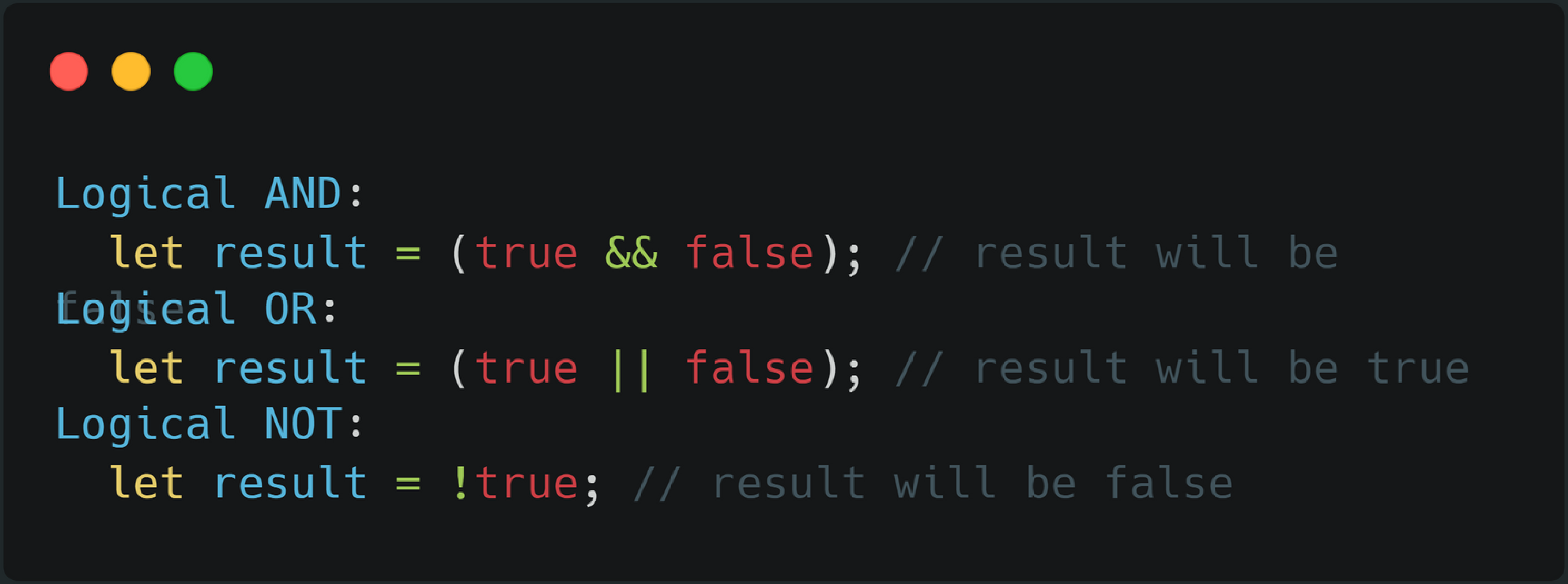
```
Assignment:
  let x = 6;
Addition assignment:
  x += 3; // equivalent to x = x + 3;
Subtraction assignment:
  x -= 2; // equivalent to x = x - 2;
Multiplication assignment:
  x *= 4; // equivalent to x = x * 4;
Division assignment:
  x /= 2; // equivalent to x = x / 2;
Modulo assignment:
  x %= 3; // equivalent to x = x % 3;
```

Comparison Operators



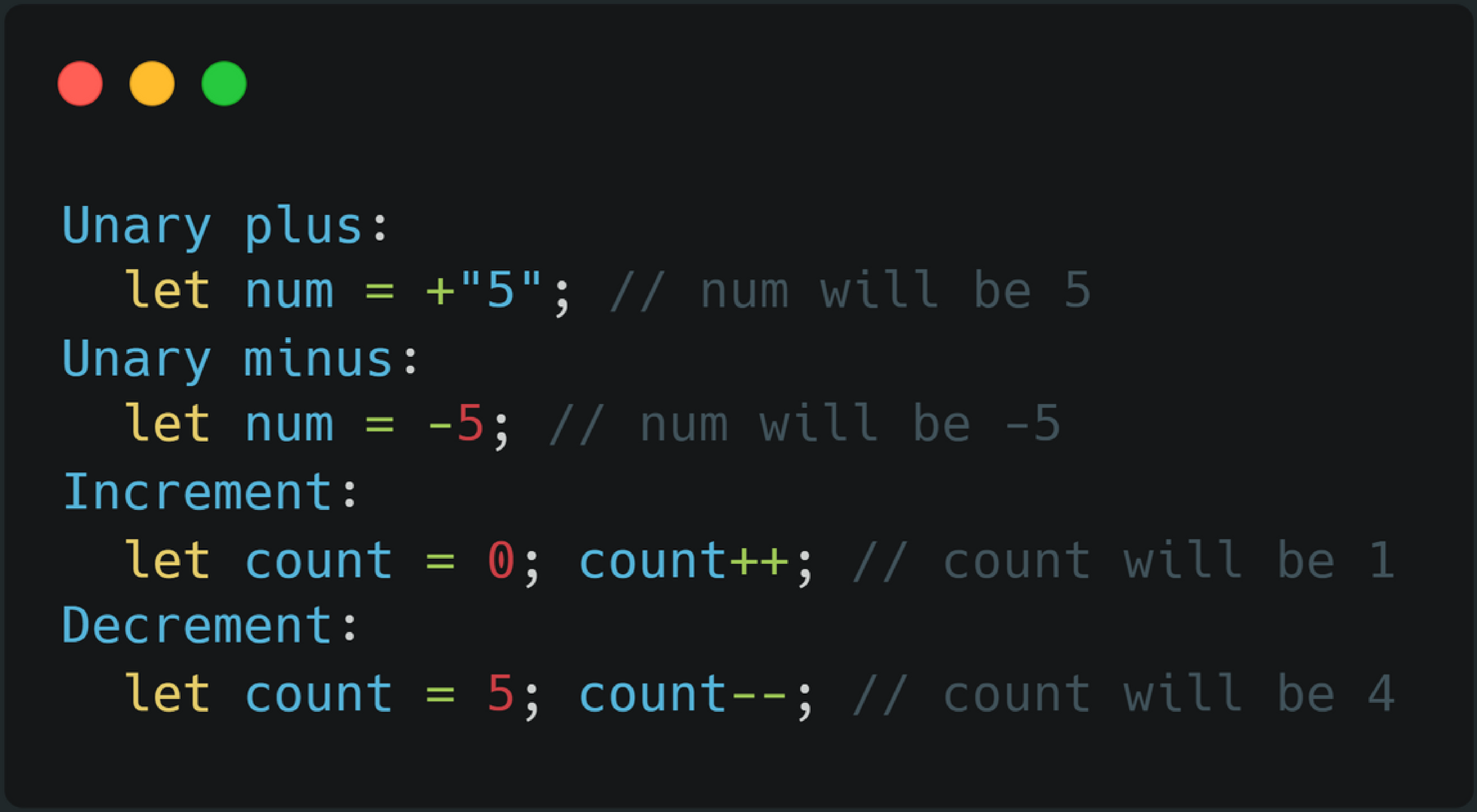
```
Equal to:
  let isEqual = 6 == 6; // isEqual will be true
Not equal to:
  let isNotEqual = 5 != 2; // isNotEqual will be true
Greater than:
  let isGreater = 7 > 5; // isGreater will be true
Less than:
  let isLess = 3 < 6; // isLess will be true
Greater than or equal to:
  let isGreaterOrEqual = 8 >= 8; // isGreaterOrEqual will be true
Less than or equal to:
  let isLessOrEqual = 4 <= 3; // isLessOrEqual will be false
```

Logical Operators



```
Logical AND:
  let result = (true && false); // result will be
false
Logical OR:
  let result = (true || false); // result will be true
Logical NOT:
  let result = !true; // result will be false
```

Unary Operators

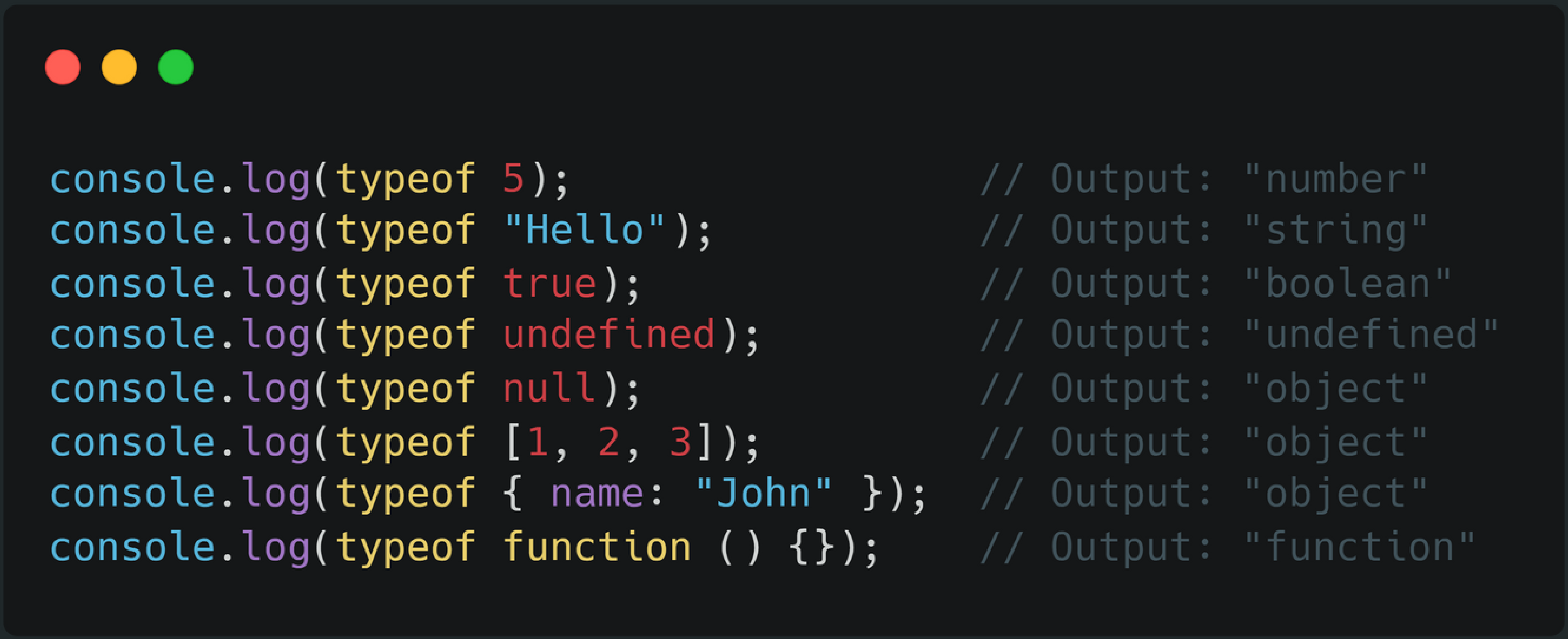


```
Unary plus:
  let num = +"5"; // num will be 5
Unary minus:
  let num = -5; // num will be -5
Increment:
  let count = 0; count++; // count will be 1
Decrement:
  let count = 5; count--; // count will be 4
```

Type Operators

Unary typeof

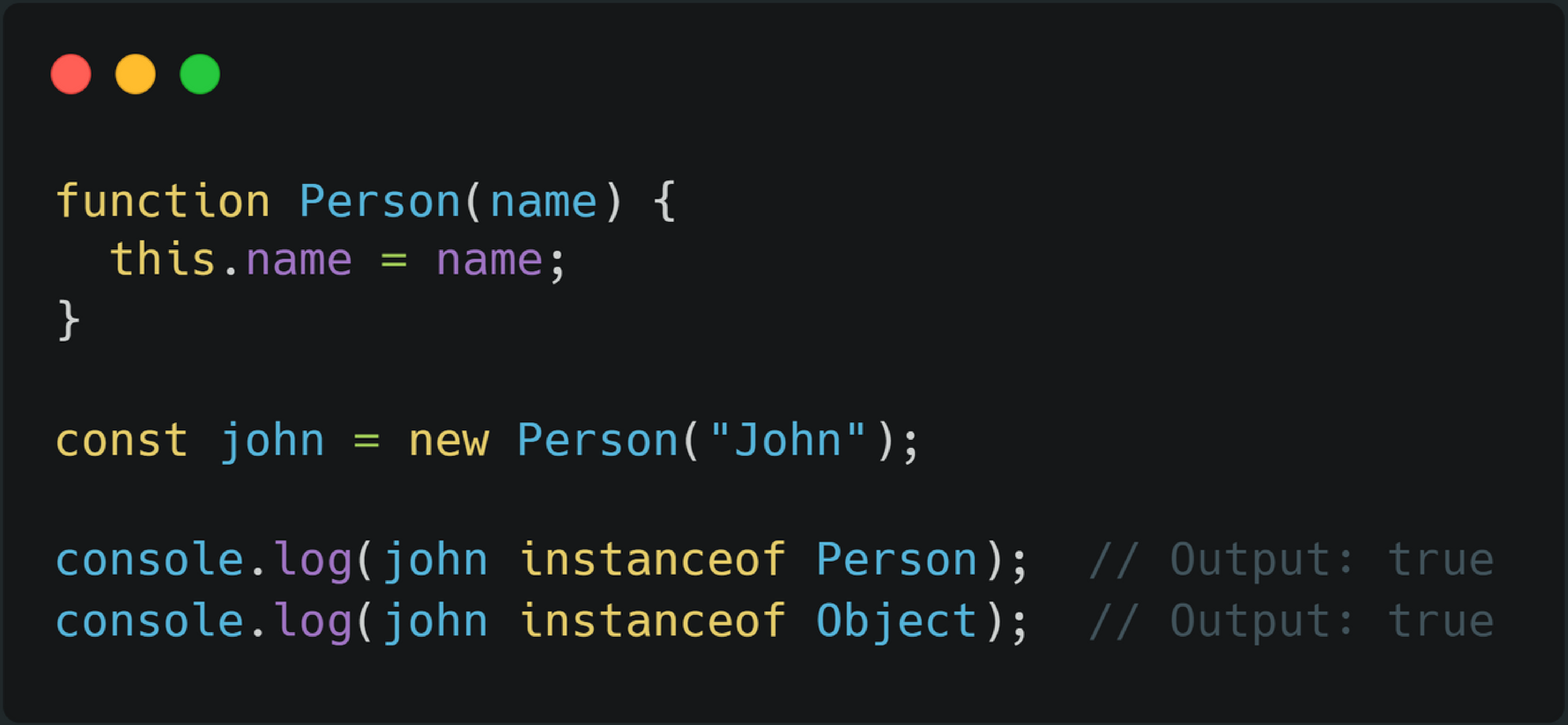
The unary typeof operator is used to determine the data type of a value or variable. It returns a string indicating the type of the operand.



```
console.log(typeof 5);                 // Output: "number"
console.log(typeof "Hello");           // Output: "string"
console.log(typeof true);              // Output: "boolean"
console.log(typeof undefined);         // Output: "undefined"
console.log(typeof null);              // Output: "object"
console.log(typeof [1, 2, 3]);         // Output: "object"
console.log(typeof { name: "John" });  // Output: "object"
console.log(typeof function () {});    // Output: "function"
```

instanceof Operator

The instanceof operator is used to check if an object belongs to a specific class or constructor function. It returns a boolean value.



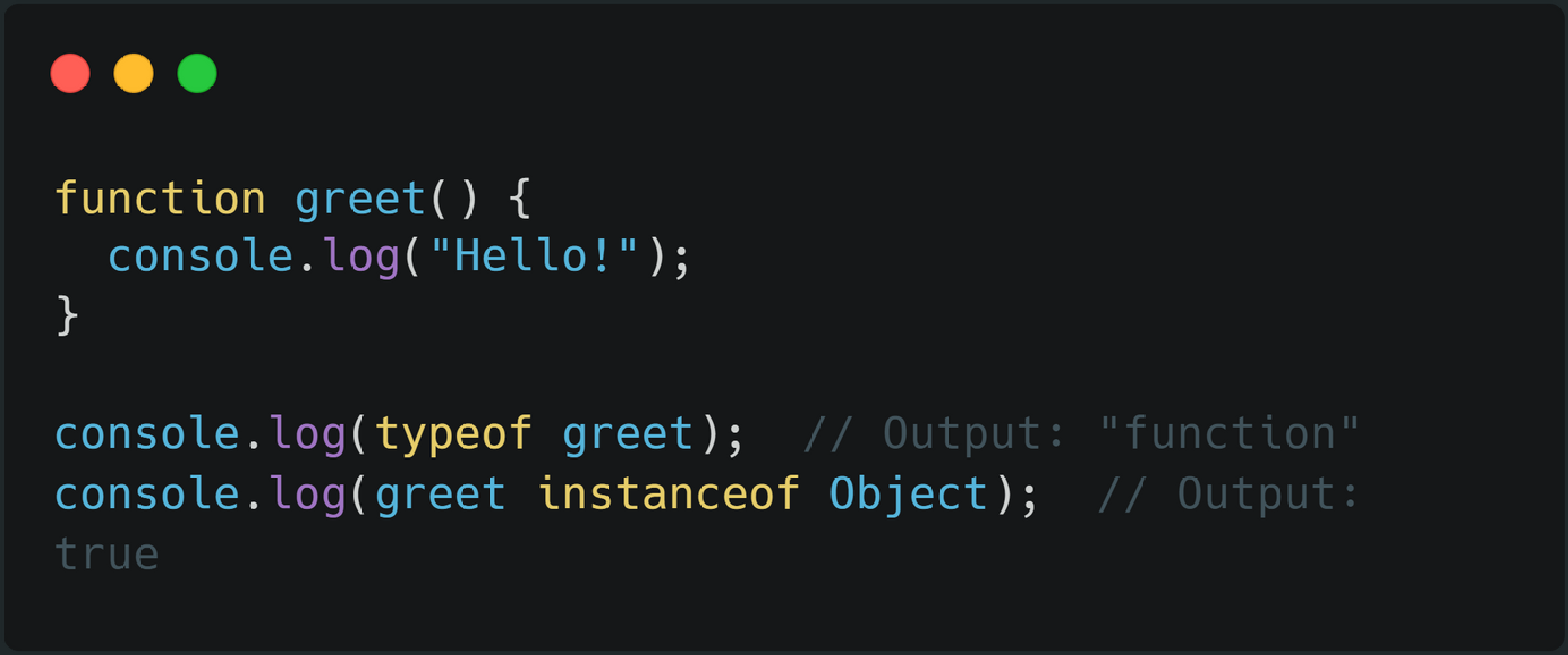
```
function Person(name) {
  this.name = name;
}

const john = new Person("John");

console.log(john instanceof Person);  // Output: true
console.log(john instanceof Object);  // Output: true
```

typeof Operator with Functions

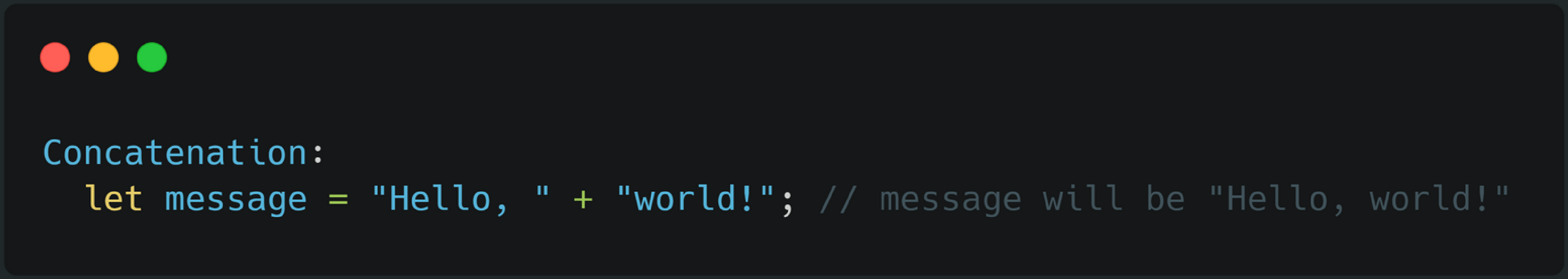
When used with functions, the typeof operator returns "function". However, functions are also objects in JavaScript.



```javascript
function greet() {
  console.log("Hello!");
}

console.log(typeof greet);  // Output: "function"
console.log(greet instanceof Object);  // Output:
true
```

String Operators



```
Concatenation:
  let message = "Hello, " + "world!"; // message will be "Hello, world!"
```

Thank You

Like

Share

Follow

Shahroz Khalid

Save for Later