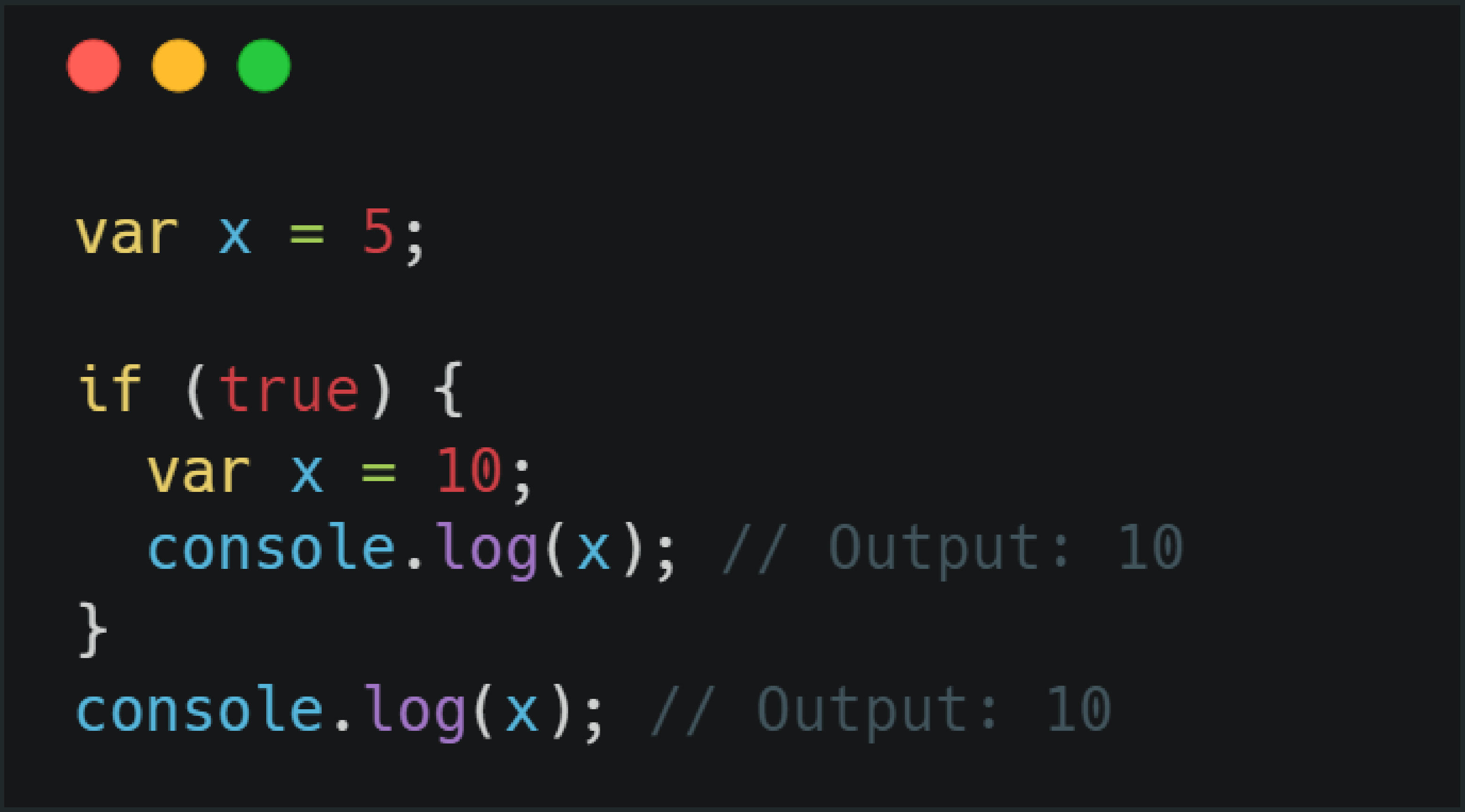
let, const and var

JS

1. var

Oldest way to declare variables in JavaScript. Has function-level scope, can be accessed outside of blocks, and allows redeclaration and reassignment.

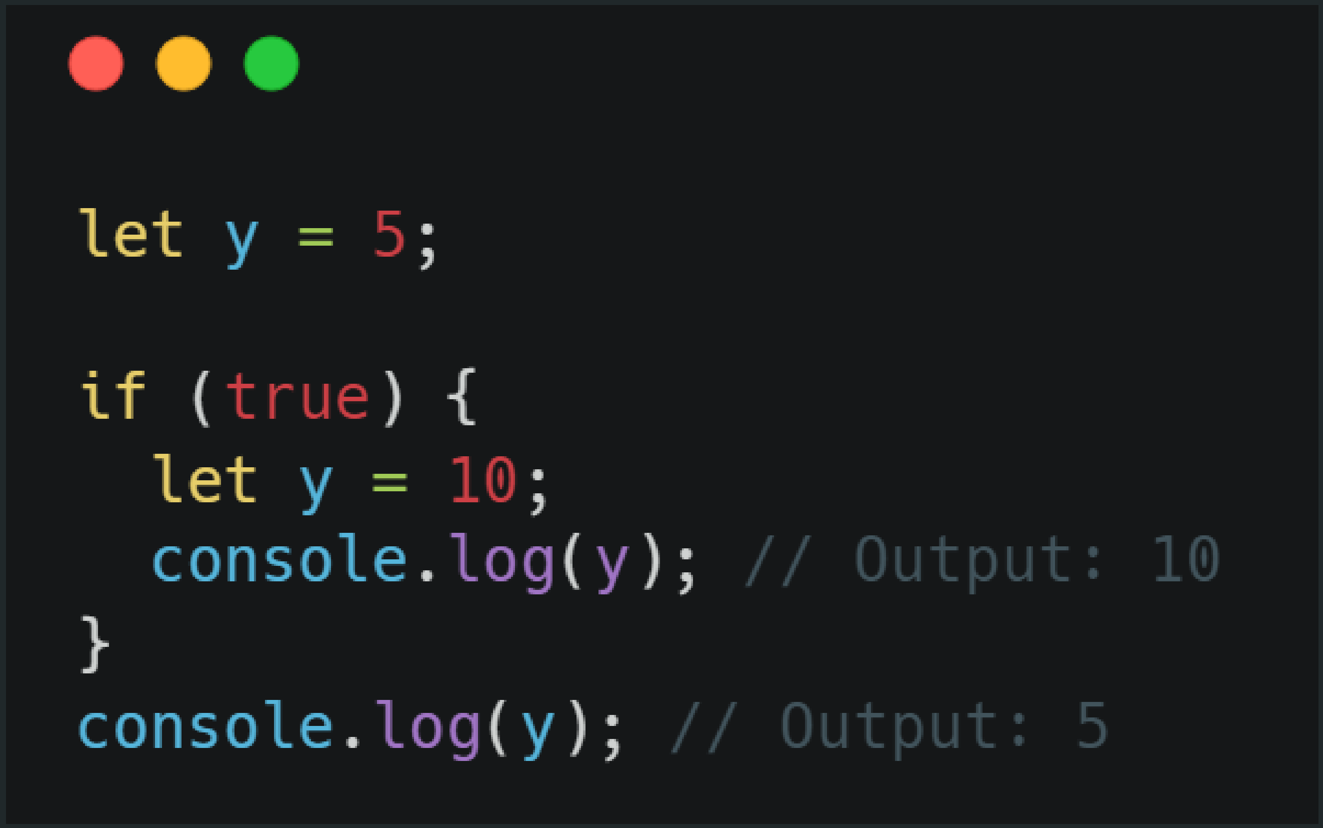


```
var x = 5;

if (true) {
  var x = 10;
  console.log(x); // Output: 10
}
console.log(x); // Output: 10
```

2. let

Introduced in ES6, provides block-level scope. Limited to the block it is declared in, allows reassignment but not redeclaration within the same scope.

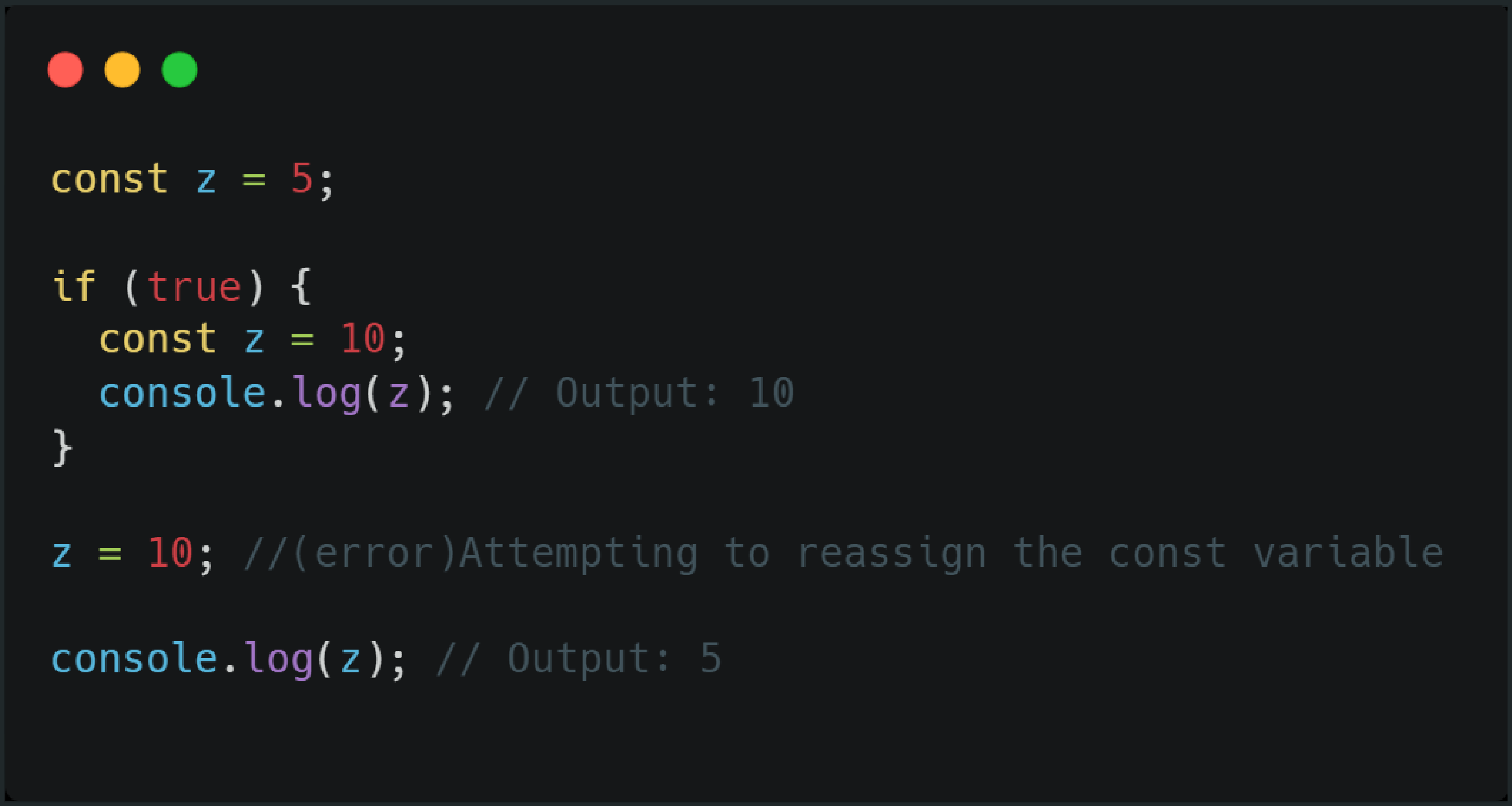


```
let y = 5;

if (true) {
  let y = 10;
  console.log(y); // Output: 10
}
console.log(y); // Output: 5
```

3. const

Also introduced in ES6, offers block-level scope. Declares constants that cannot be reassigned after initialization. Not redeclarable within the same scope



```
const z = 5;

if (true) {
  const z = 10;
  console.log(z); // Output: 10
}

z = 10; //(error)Attempting to reassign the const variable

console.log(z); // Output: 5
```

When to use which keyword?

- Use var when targeting older JavaScript environments or when you specifically need function-level scoping.
- Use let when you need block-level scoping; the variable's value might change over time.
- Use const when you need block-level scoping and want to declare a constant value that won't be reassigned.

Thank You

Like

Share

Follow

in Uzair Ahmad

Save for Later