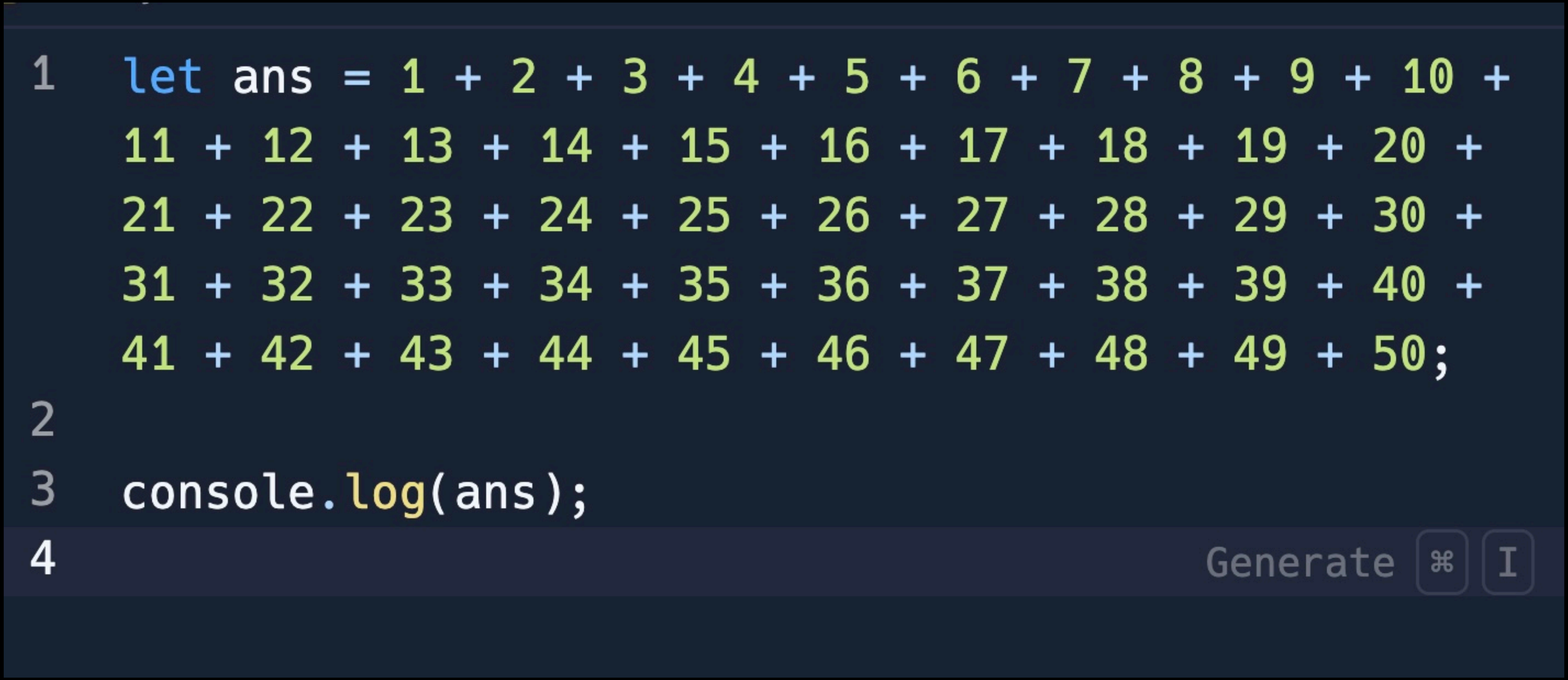
Loops, Functions and Callbacks in JS

Loops

Calculate sum from 0 to 100

Loops

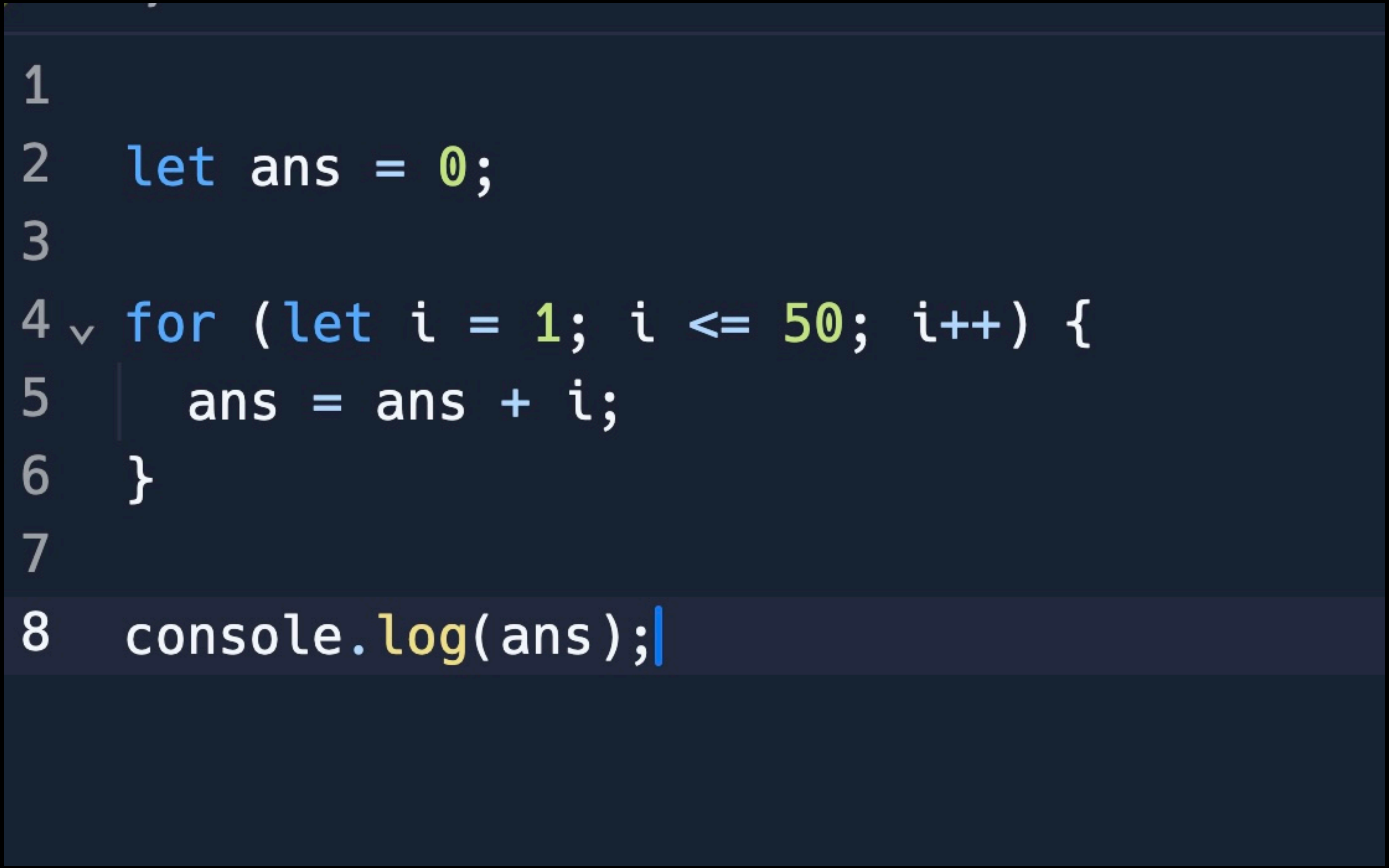
Dumb way



```
let ans = 1 + 2 + 3 + 4 + 5 + 6 + 7 + 8 + 9 + 10 +
11 + 12 + 13 + 14 + 15 + 16 + 17 + 18 + 19 + 20 +
21 + 22 + 23 + 24 + 25 + 26 + 27 + 28 + 29 + 30 +
31 + 32 + 33 + 34 + 35 + 36 + 37 + 38 + 39 + 40 +
41 + 42 + 43 + 44 + 45 + 46 + 47 + 48 + 49 + 50;

console.log(ans);

```

Loops

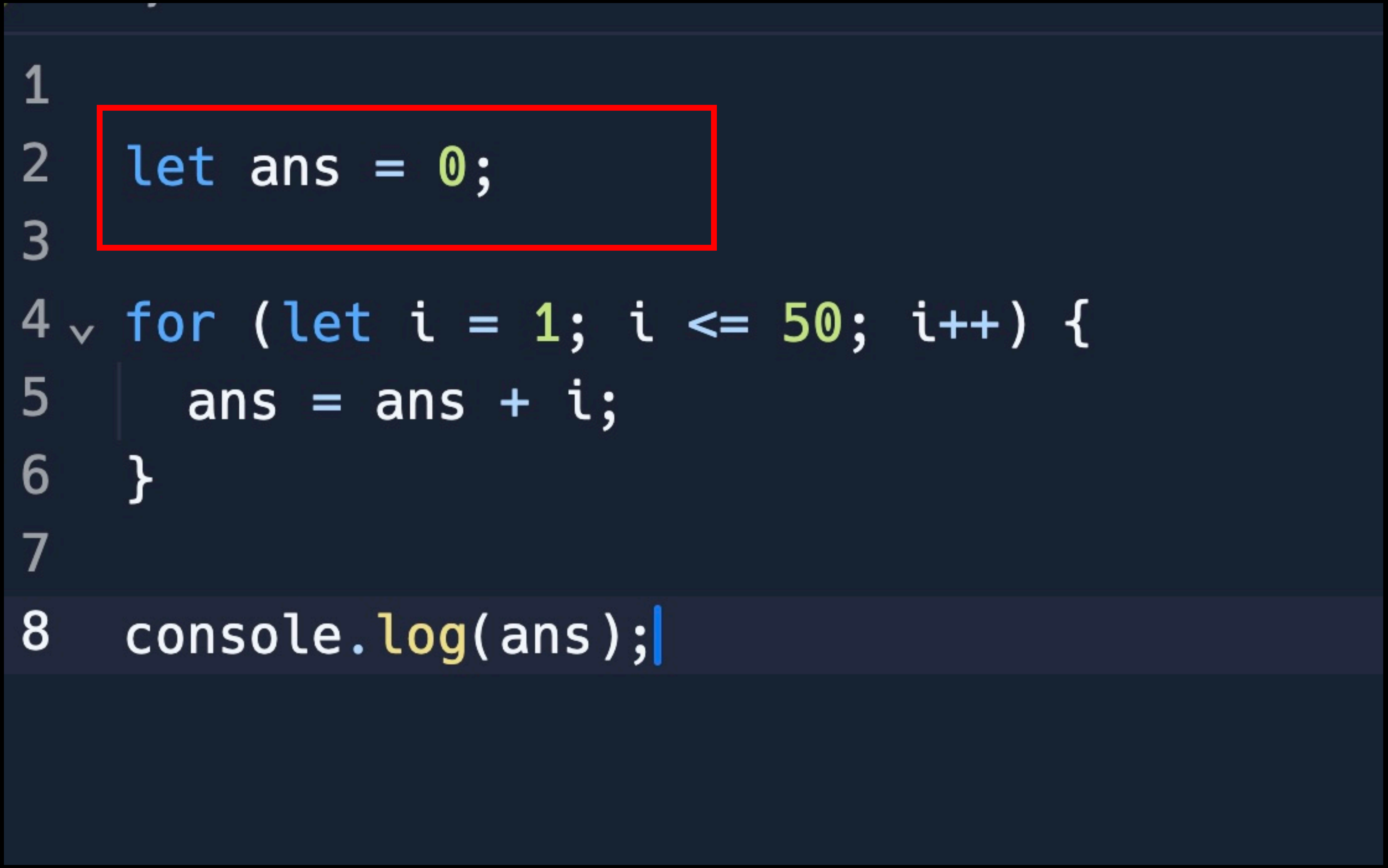
Better way - For loops



```

let ans = 0;

for (let i = 1; i <= 50; i++) {
  ans = ans + i;
}

console.log(ans);
```

Loops

Better way - For loops

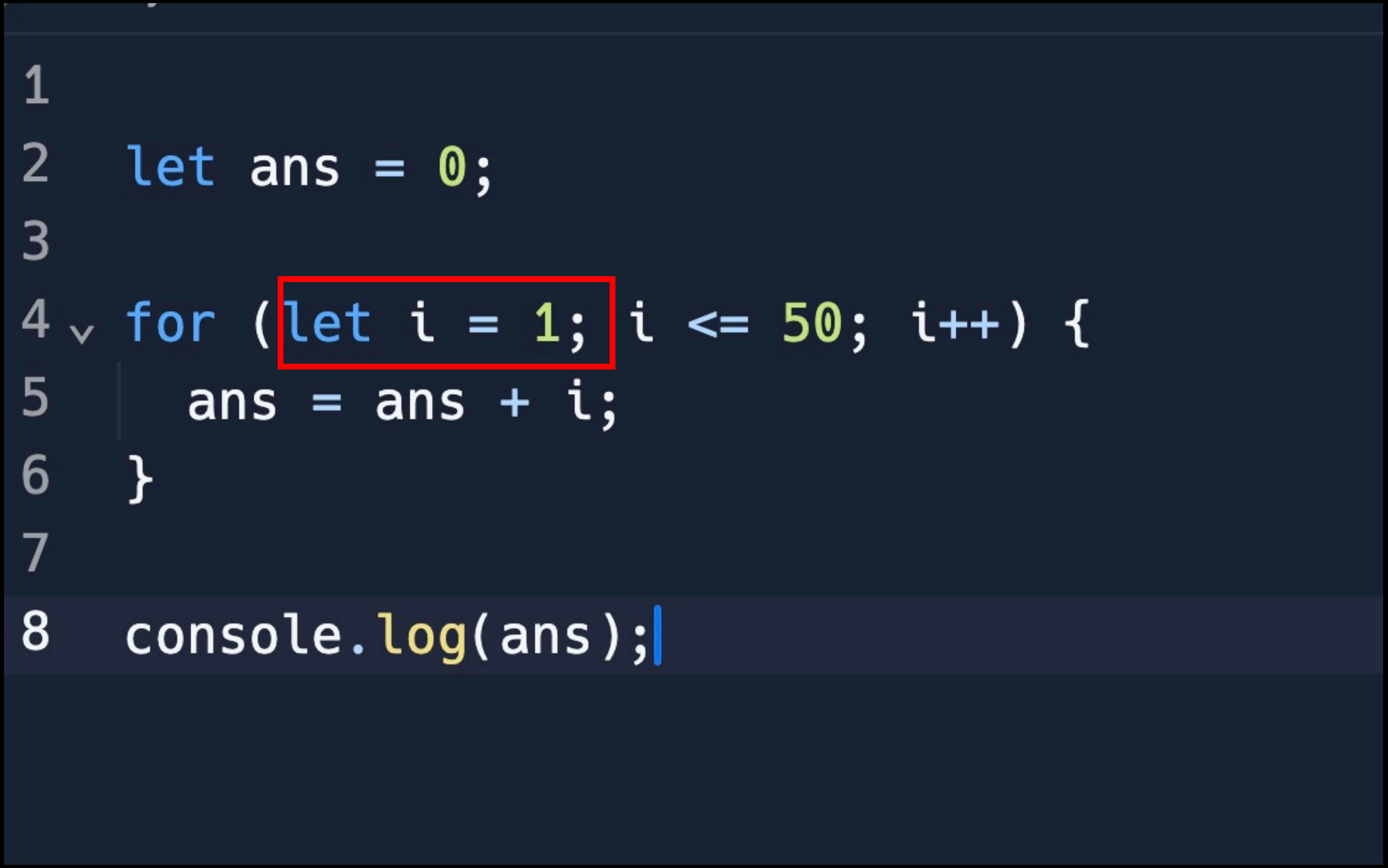


```
let ans = 0;

for (let i = 1; i <= 50; i++) {
  ans = ans + i;
}

console.log(ans);
```

ans = 0

Loops

Better way - For loops



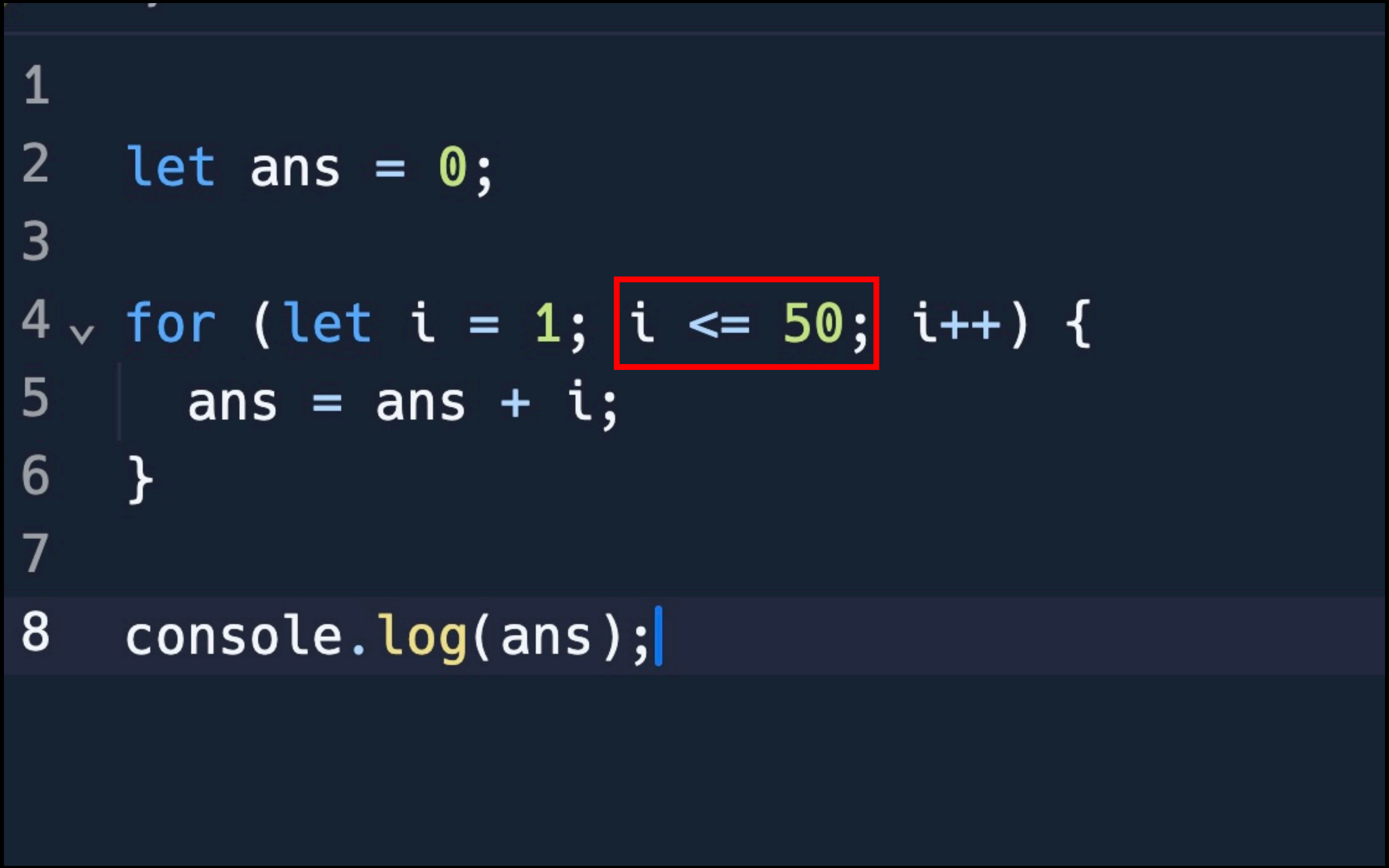
```
let ans = 0;

for (let i = 1; i <= 50; i++) {
  ans = ans + i;
}

console.log(ans);
```

ans = 0

I = 1

Loops

Better way - For loops



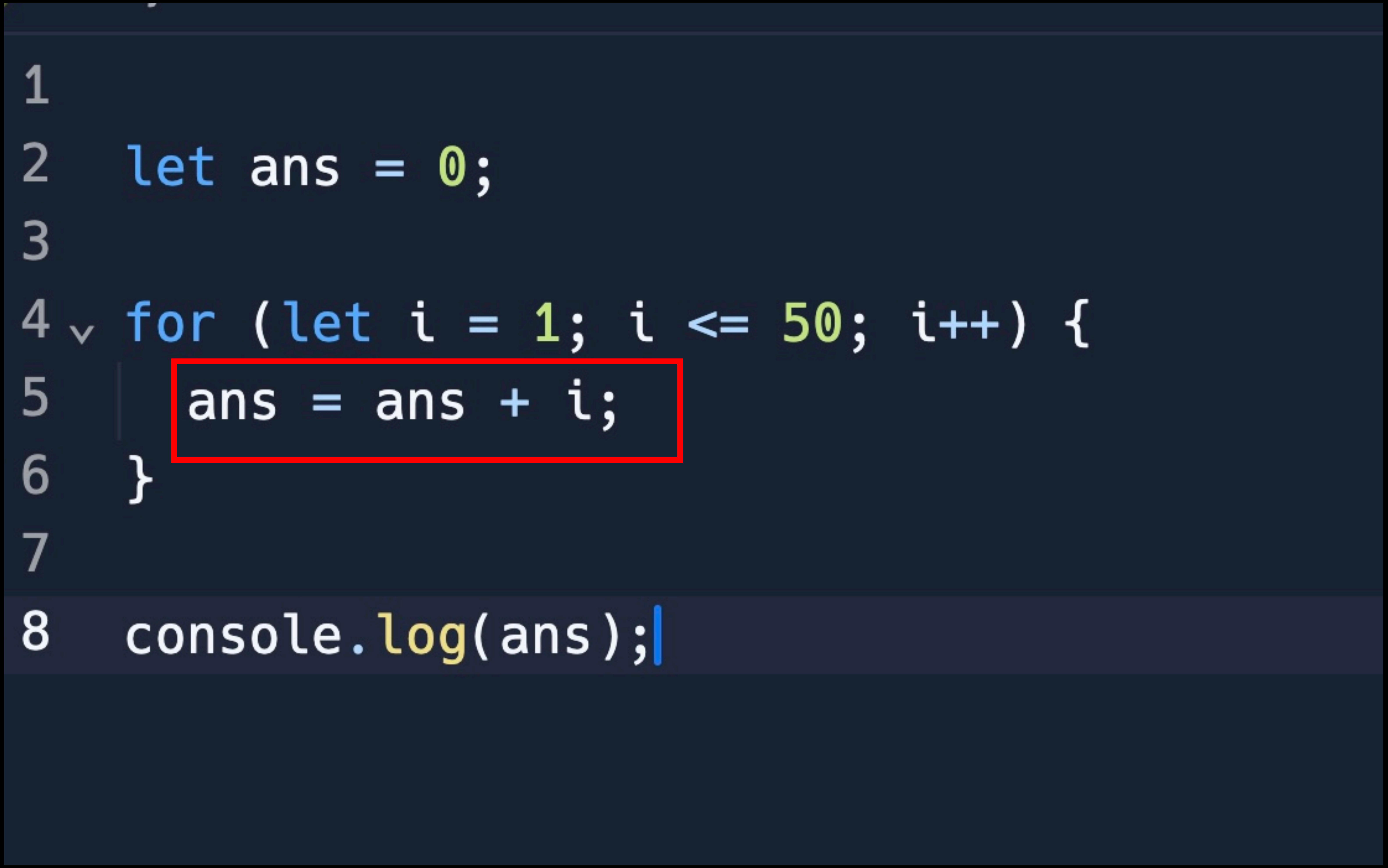
```
let ans = 0;

for (let i = 1; i <= 50; i++) {
  ans = ans + i;
}

console.log(ans);
```

ans = 0

I = 1

Loops

Better way - For loops



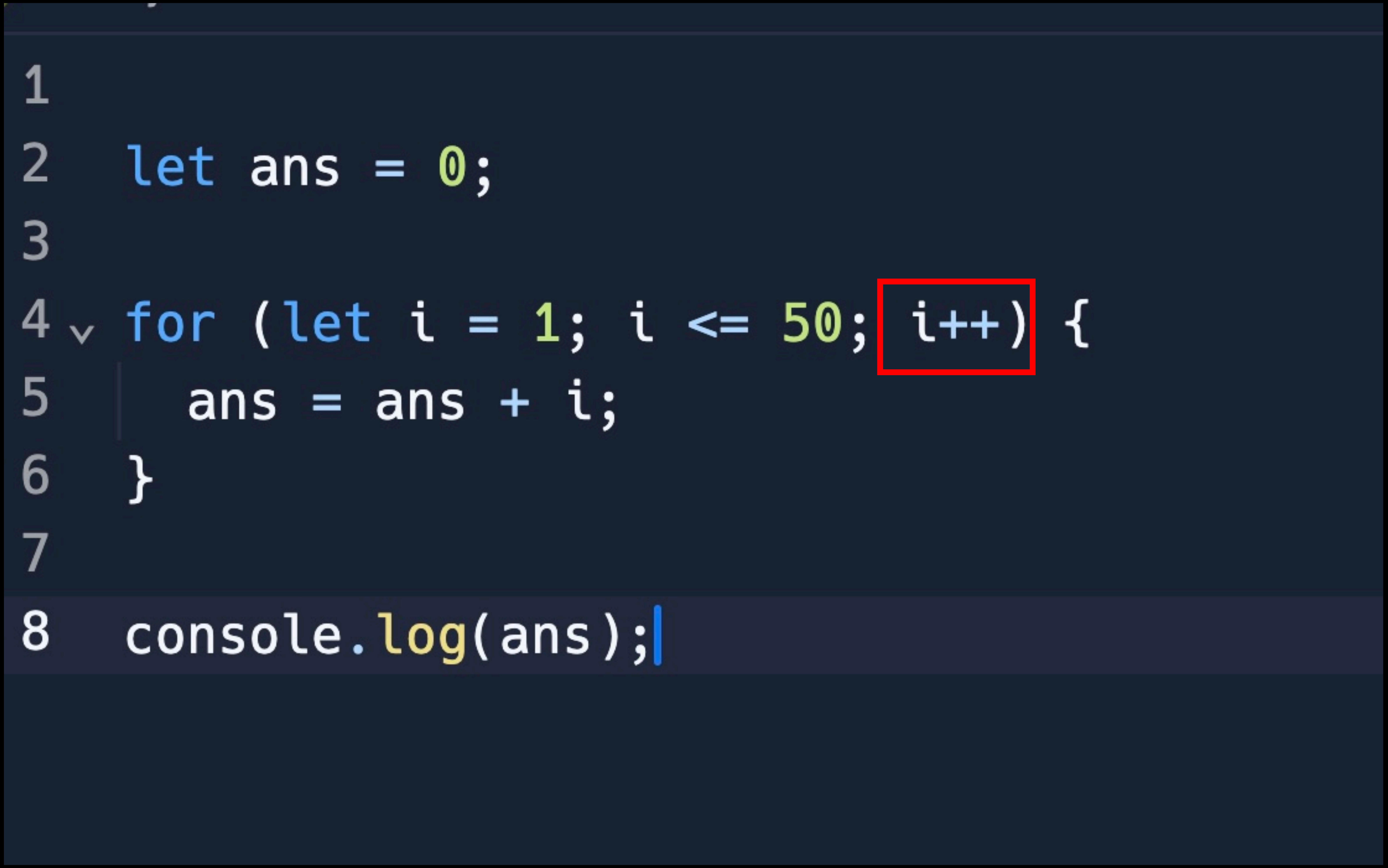
```
let ans = 0;

for (let i = 1; i <= 50; i++) {
  ans = ans + i;
}

console.log(ans);
```

ans = 1

I = 1

Loops

Better way - For loops



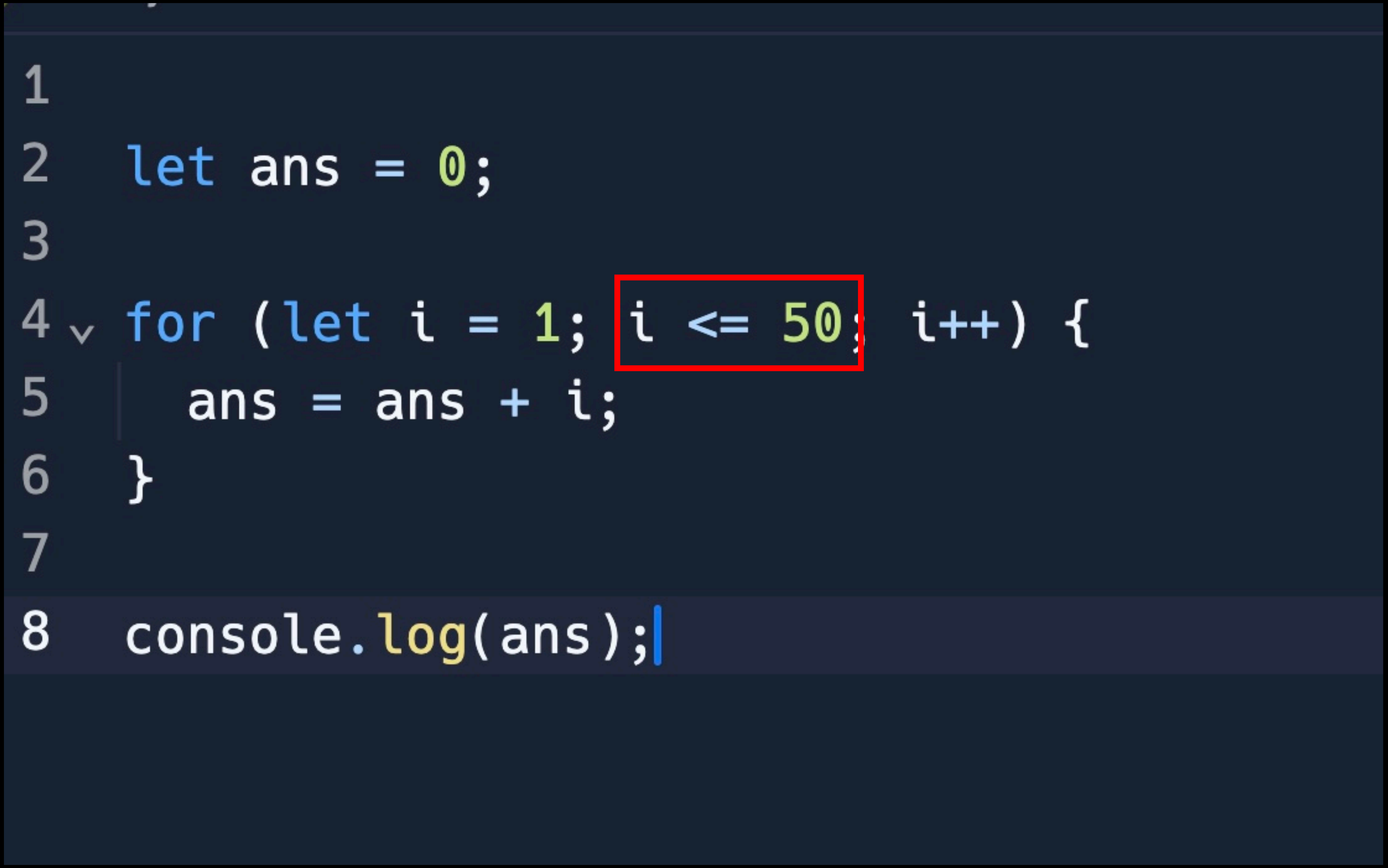
```
let ans = 0;

for (let i = 1; i <= 50; i++) {
  ans = ans + i;
}

console.log(ans);
```

ans = 1

I = 2

Loops

Better way - For loops



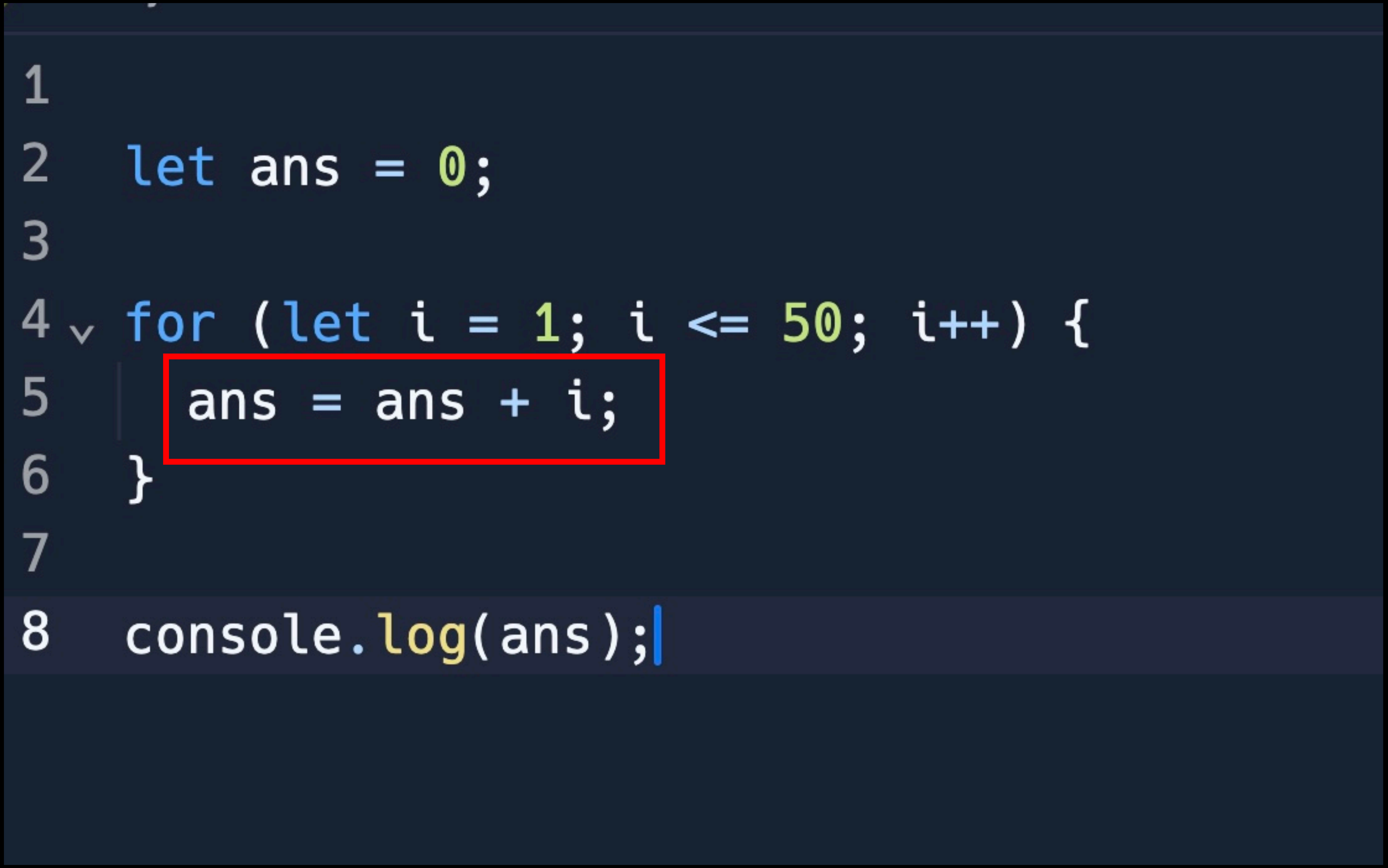
```

let ans = 0;

for (let i = 1; i <= 50; i++) {
  ans = ans + i;
}

console.log(ans);
```

ans = 1

I = 2

Loops

Better way - For loops



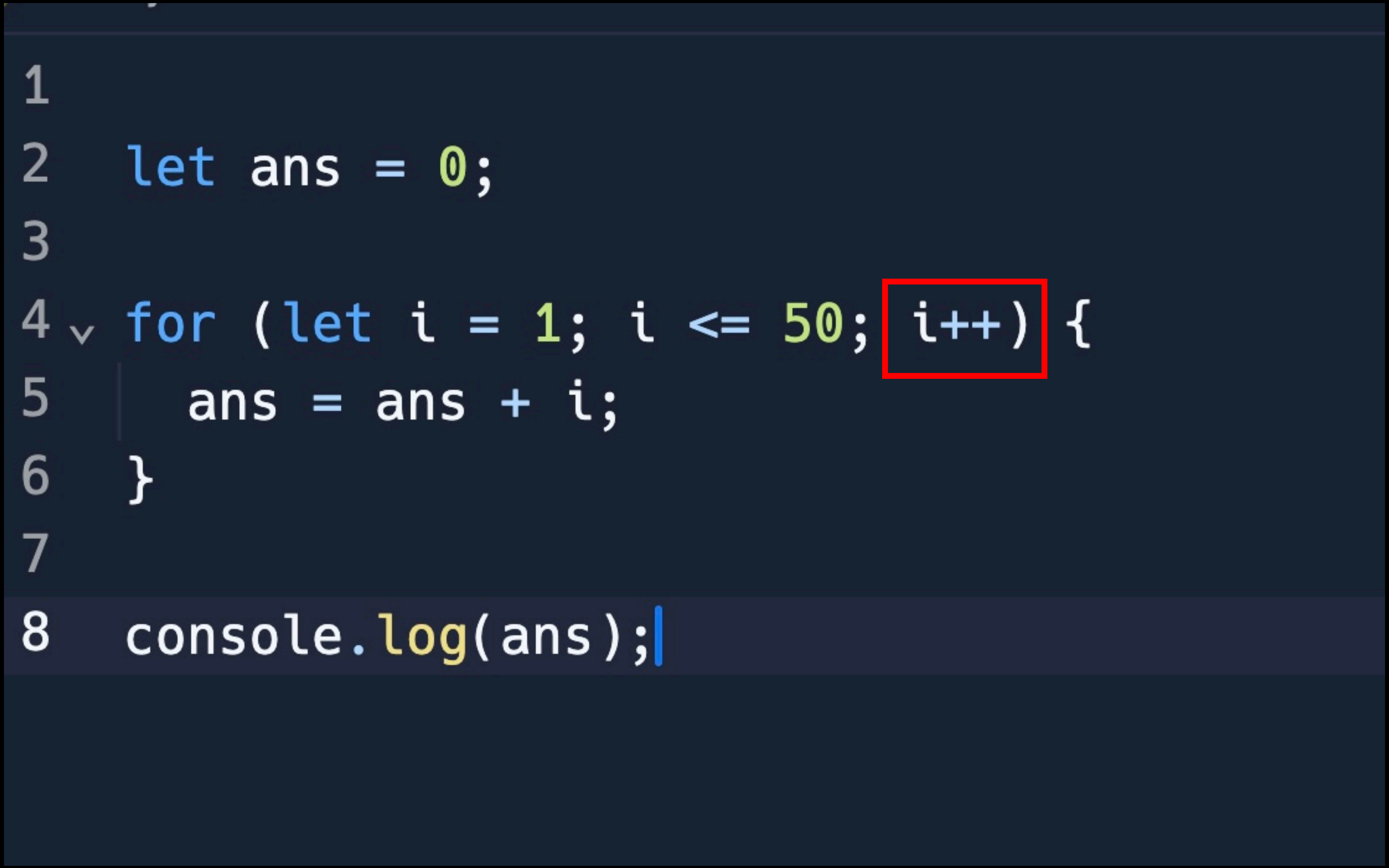
```
let ans = 0;

for (let i = 1; i <= 50; i++) {
  ans = ans + i;
}

console.log(ans);
```

ans = 3

I = 2

Loops

Better way - For loops



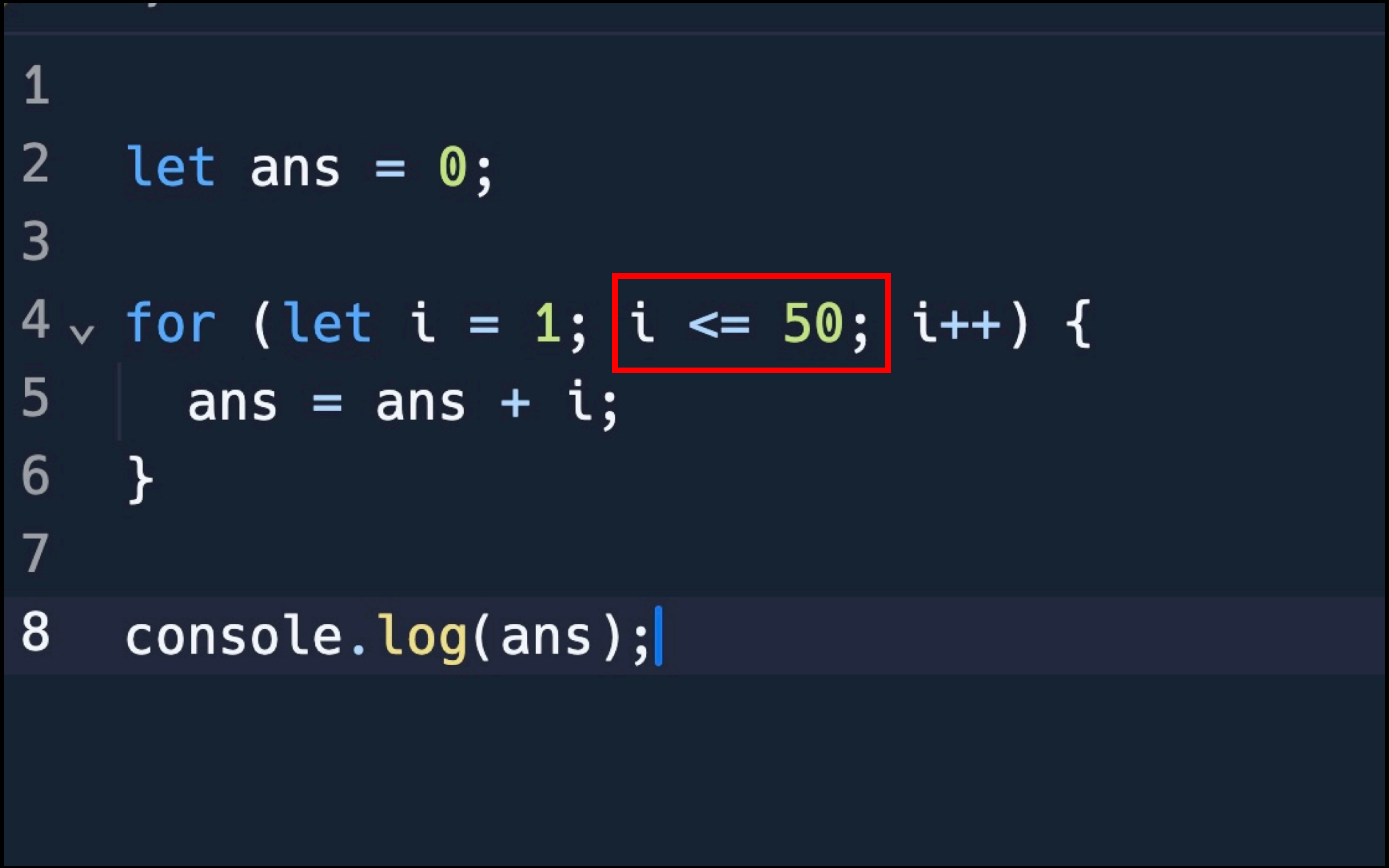
```
let ans = 0;

for (let i = 1; i <= 50; i++) {
  ans = ans + i;
}

console.log(ans);
```

ans = 3

I = 3

Loops

Better way - For loops



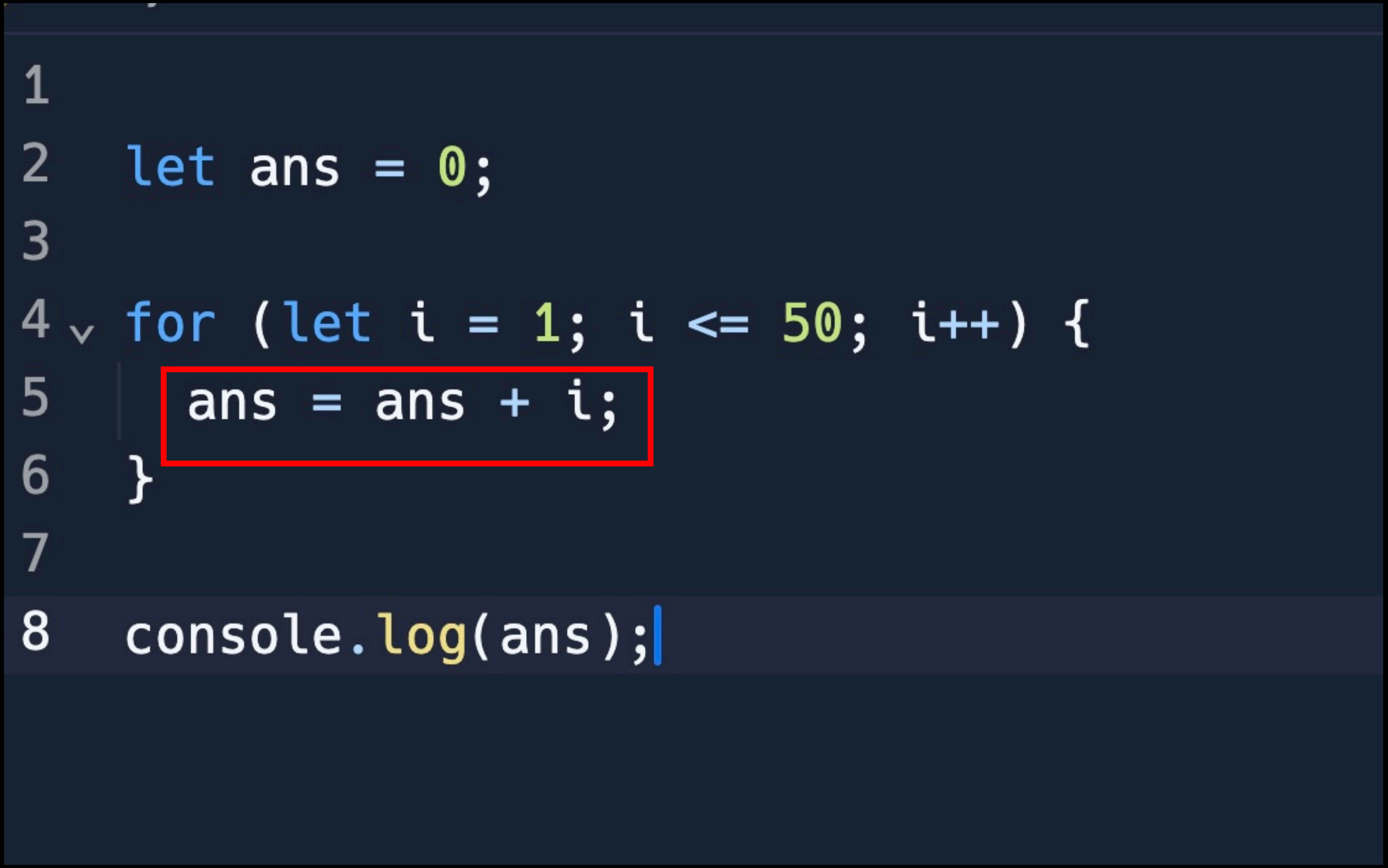
```
let ans = 0;

for (let i = 1; i <= 50; i++) {
  ans = ans + i;
}

console.log(ans);
```

ans = 3

I = 3

Loops

Better way - For loops



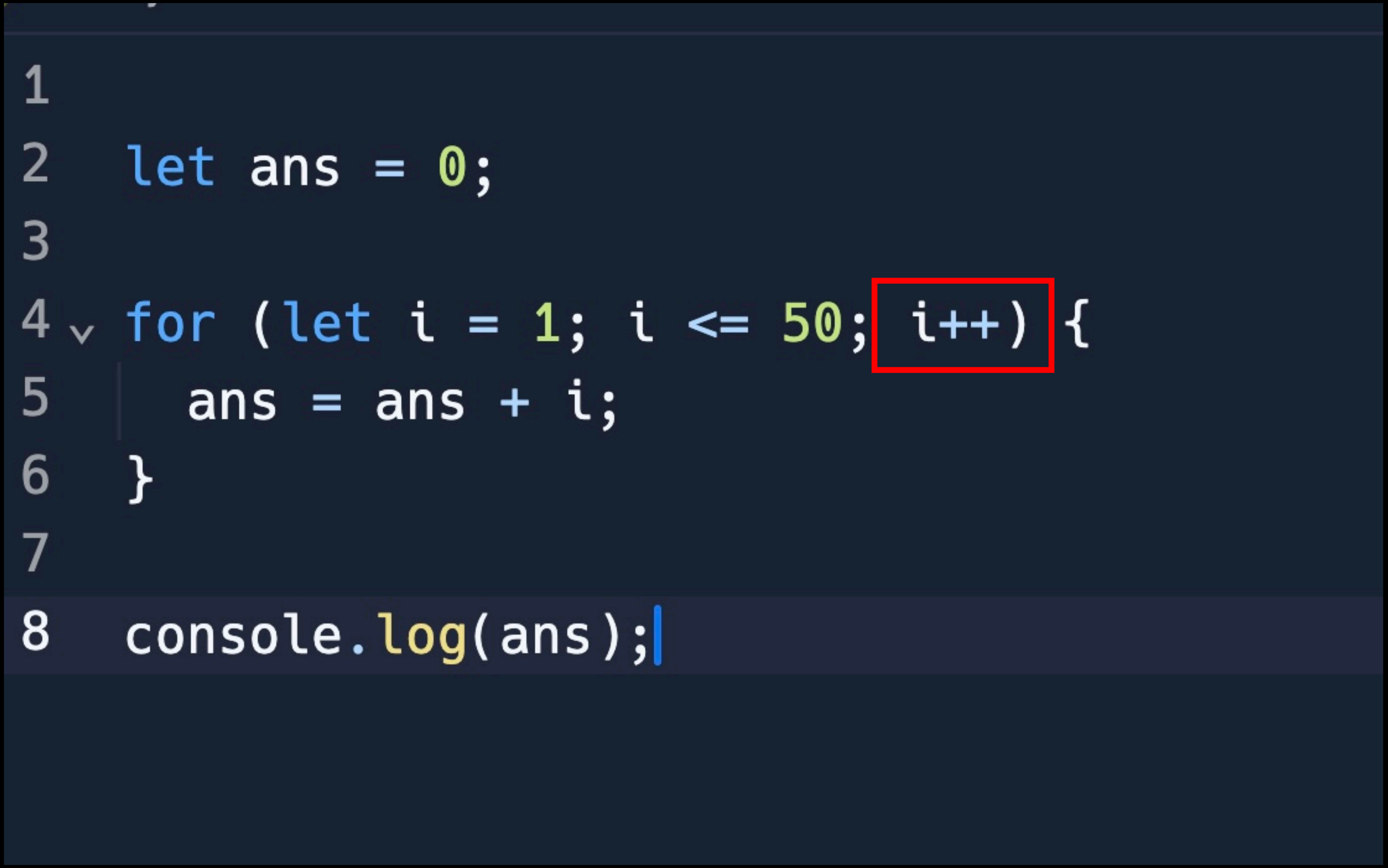
```

let ans = 0;

for (let i = 1; i <= 50; i++) {
  ans = ans + i;
}

console.log(ans);
```

ans = 6

I = 3

Loops

Better way - For loops



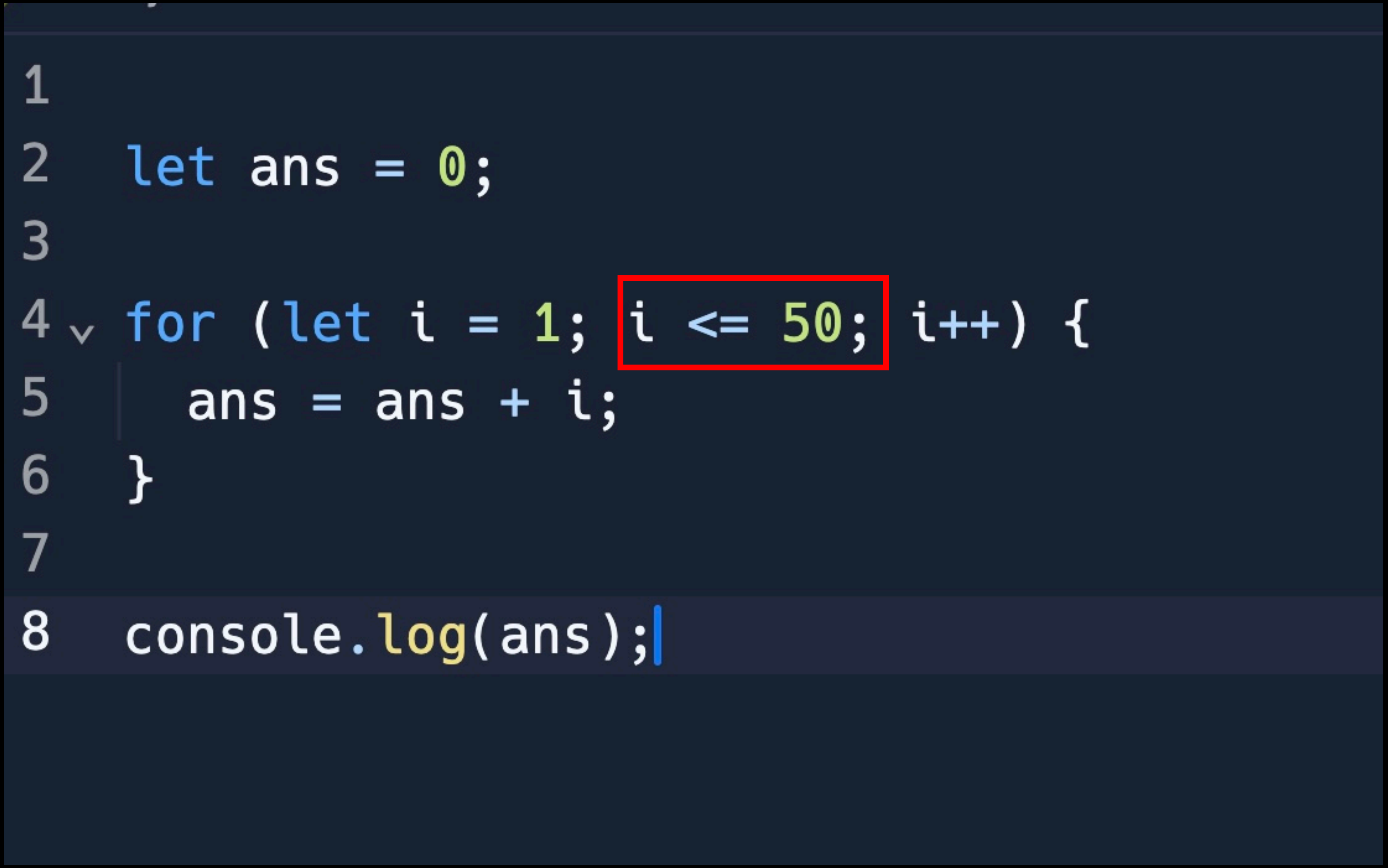
```
let ans = 0;

for (let i = 1; i <= 50; i++) {
  ans = ans + i;
}

console.log(ans);
```

ans = 6

I = 4

Loops

Better way - For loops



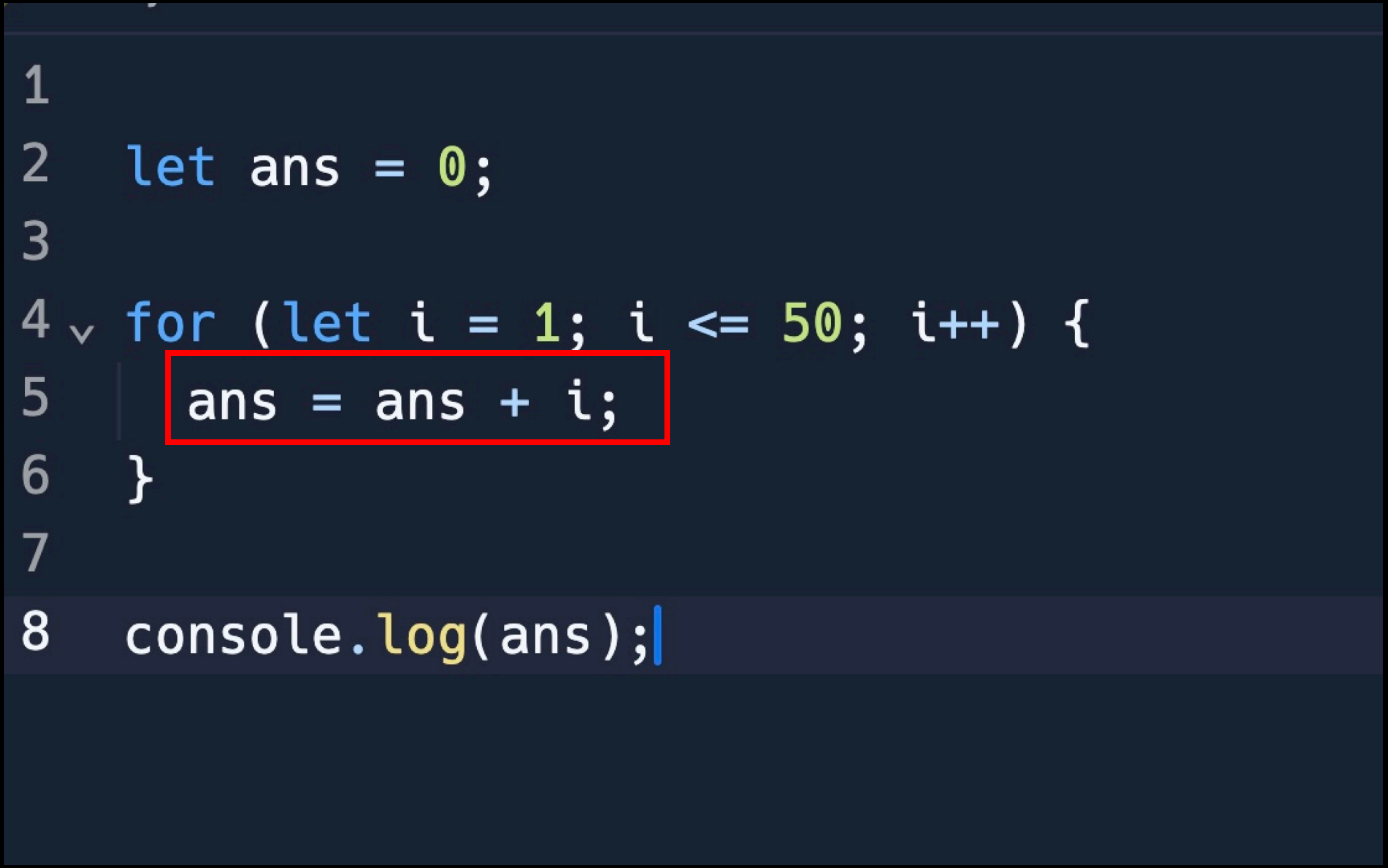
```

let ans = 0;

for (let i = 1; i <= 50; i++) {
  ans = ans + i;
}

console.log(ans);
```

ans = 6

I = 4

Loops

Better way - For loops



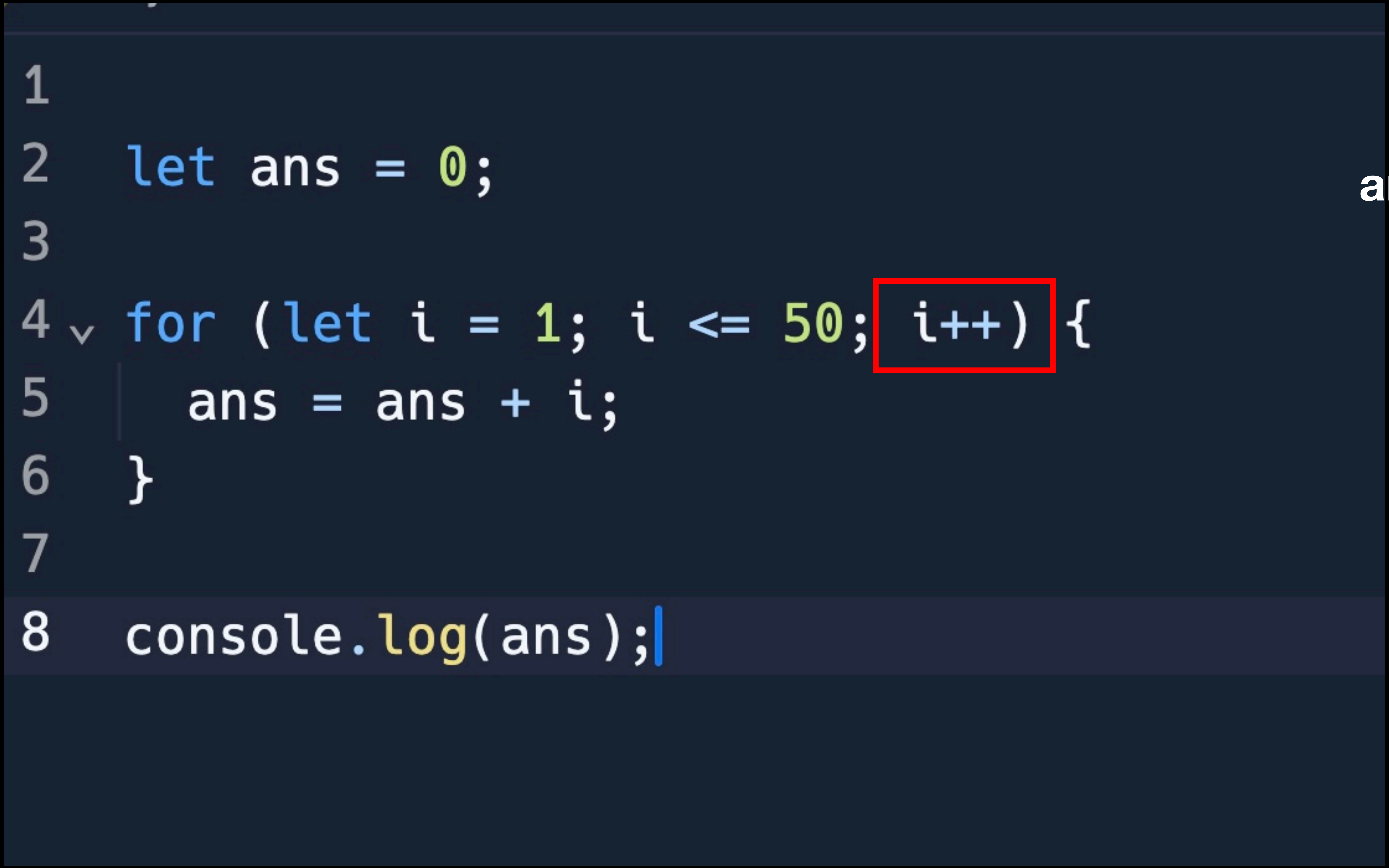
```
let ans = 0;

for (let i = 1; i <= 50; i++) {
  ans = ans + i;
}

console.log(ans);
```

ans = 10

I = 4

Loops

Better way - For loops



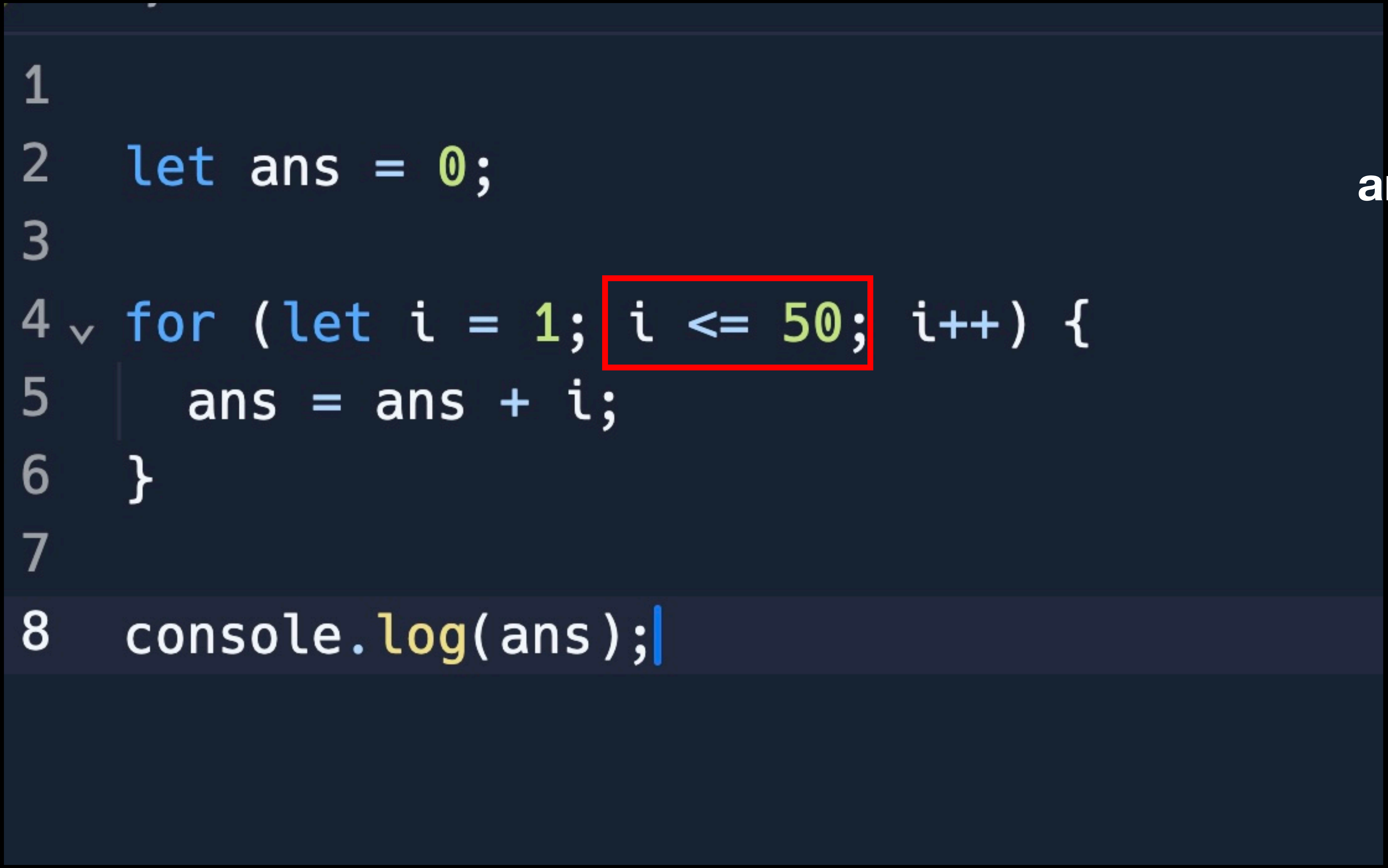
```
let ans = 0;

for (let i = 1; i <= 50; i++) {
  ans = ans + i;
}

console.log(ans);
```

ans = very big value

I = 51

Loops

Better way - For loops



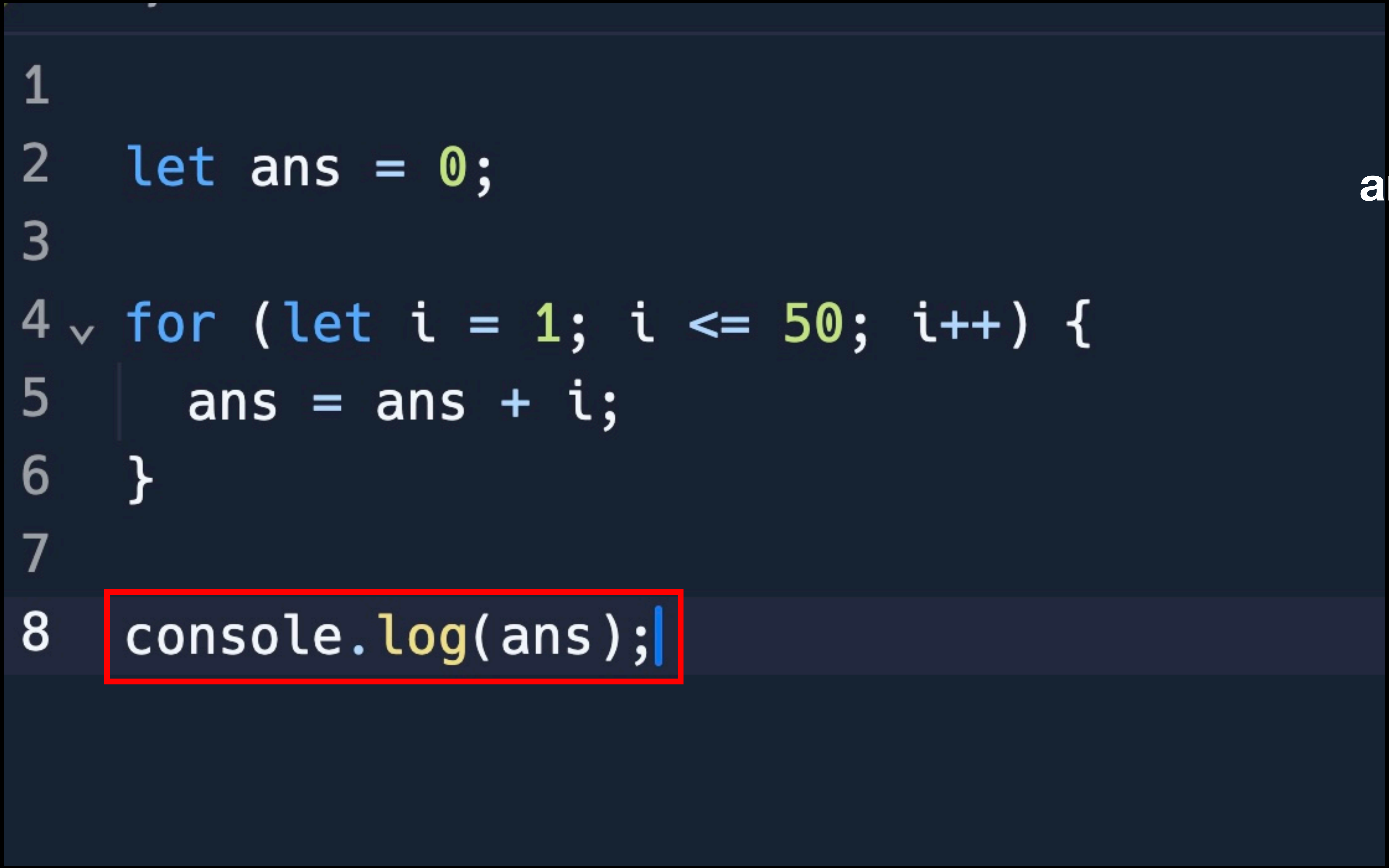
```javascript
let ans = 0;

for (let i = 1; i <= 50; i++) {
  ans = ans + i;
}

console.log(ans);
```

ans = very big value

I = 51

Loops

Better way - For loops



```
let ans = 0;

for (let i = 1; i <= 50; i++) {
  ans = ans + i;
}

console.log(ans);
```

ans = very big value

I = 51

Loops

Great way to visualise this - http://latentflip.com/loupe/

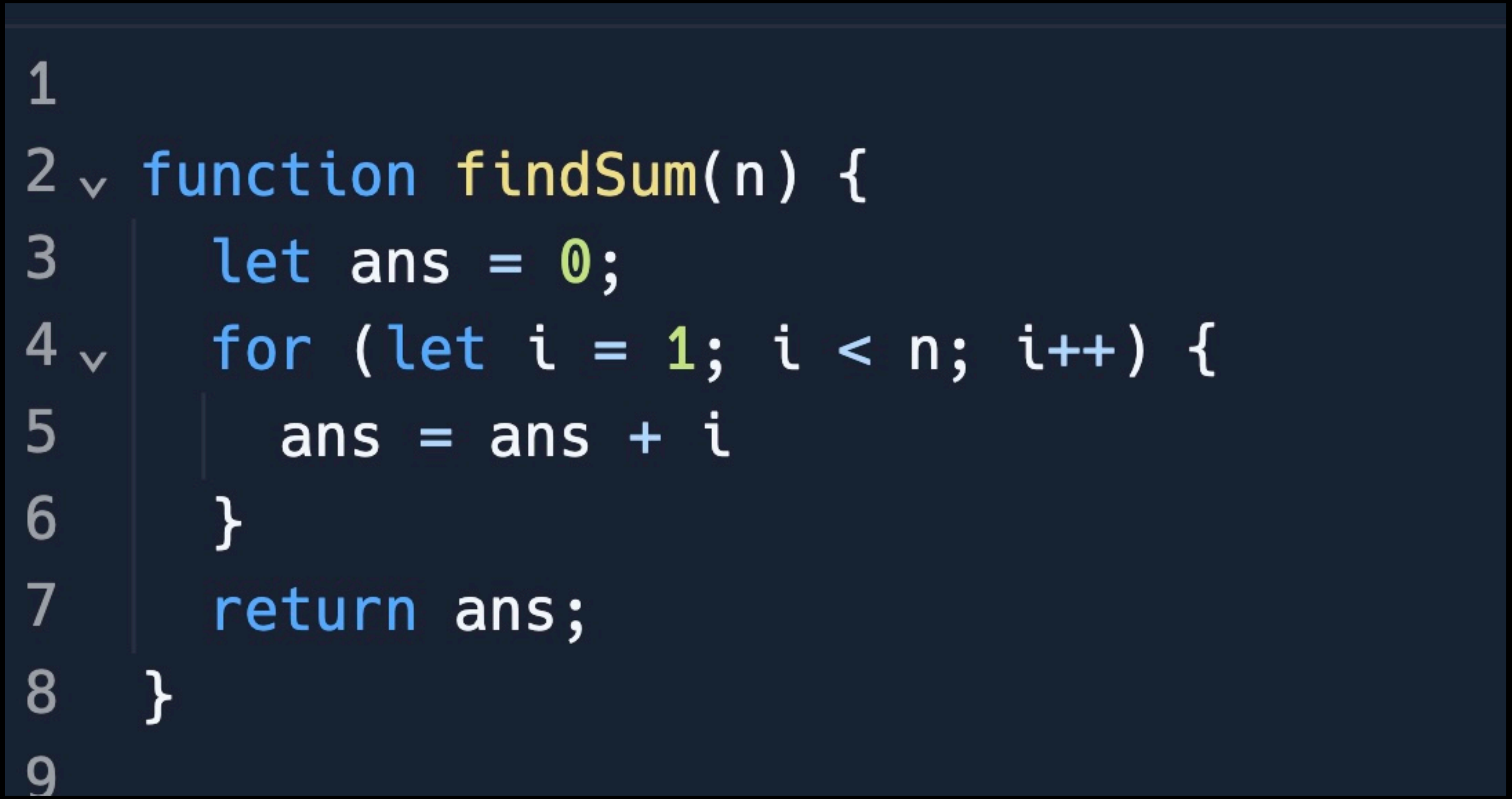
Functions

What is a function?

A function in JavaScript is a set of statements that performs a task or calculates a value
It should take some input and return an output where there is some obvious relationship between the input and the output.

Functions

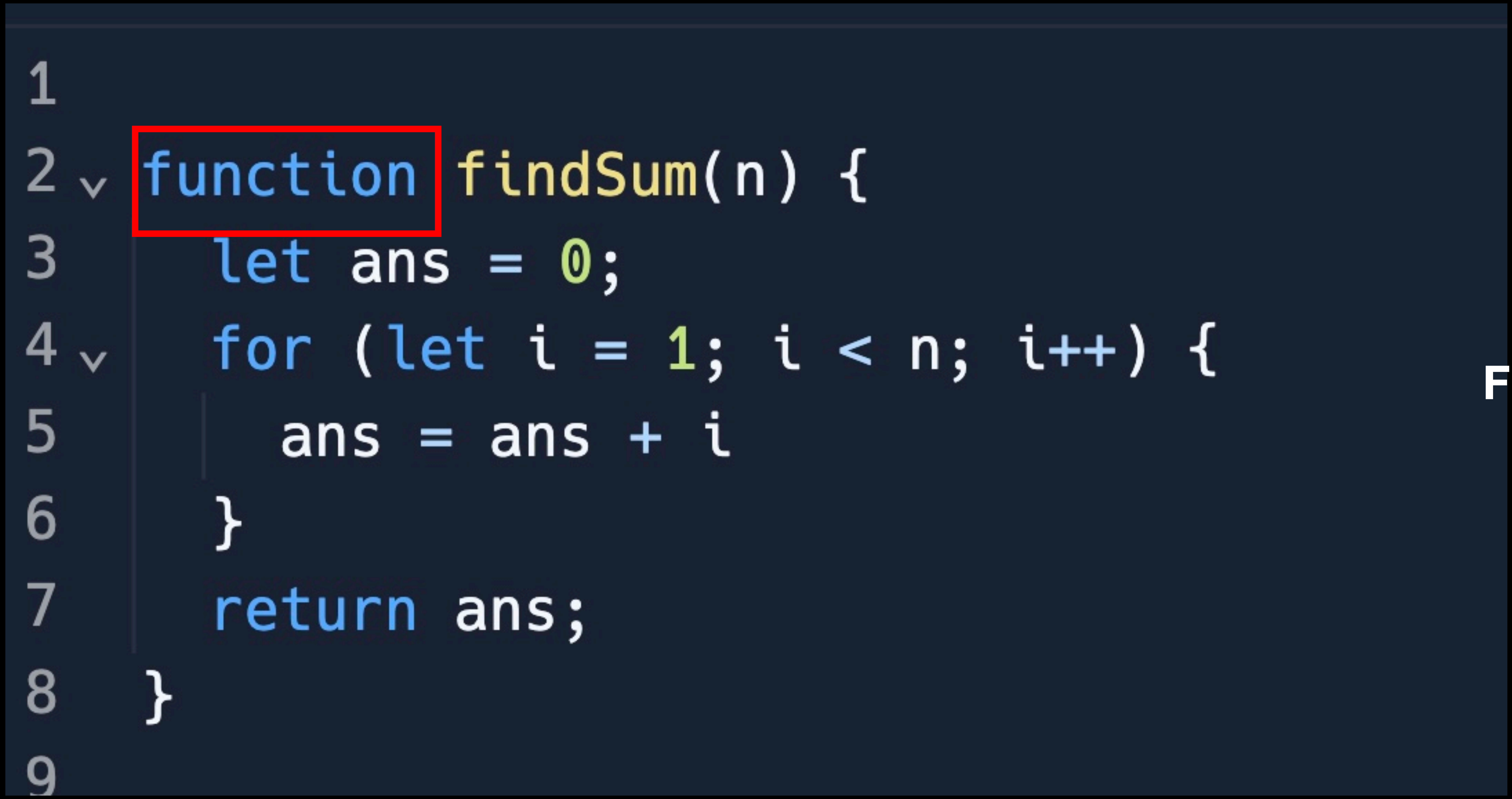
Syntax?



```javascript

function findSum(n) {
  let ans = 0;
  for (let i = 1; i < n; i++) {
    ans = ans + i
  }
  return ans;
}

```

Functions

Syntax?

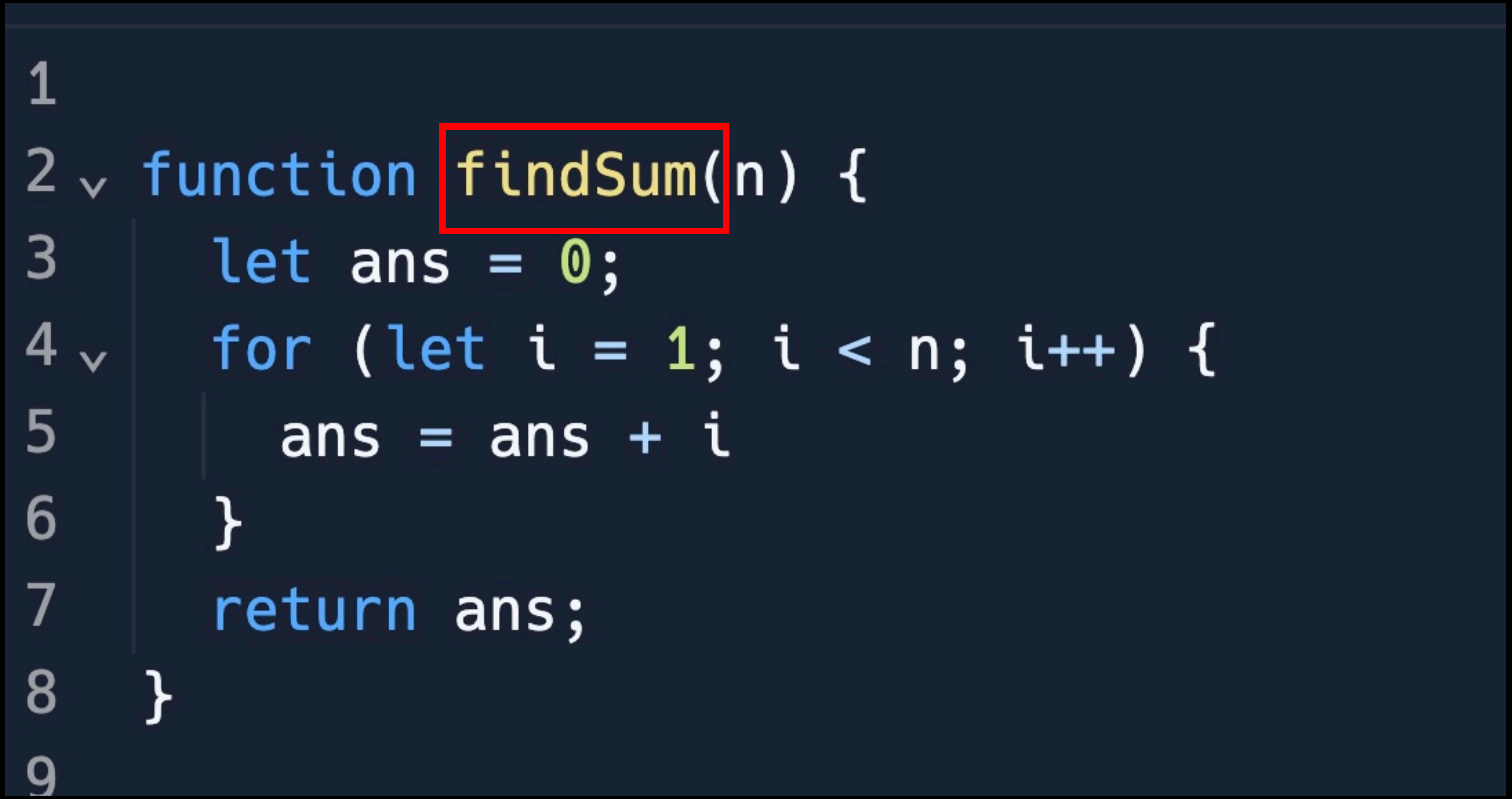


```
function findSum(n) {
  let ans = 0;
  for (let i = 1; i < n; i++) {
    ans = ans + i
  }
  return ans;
}
```

Function keyword

Functions

Syntax?

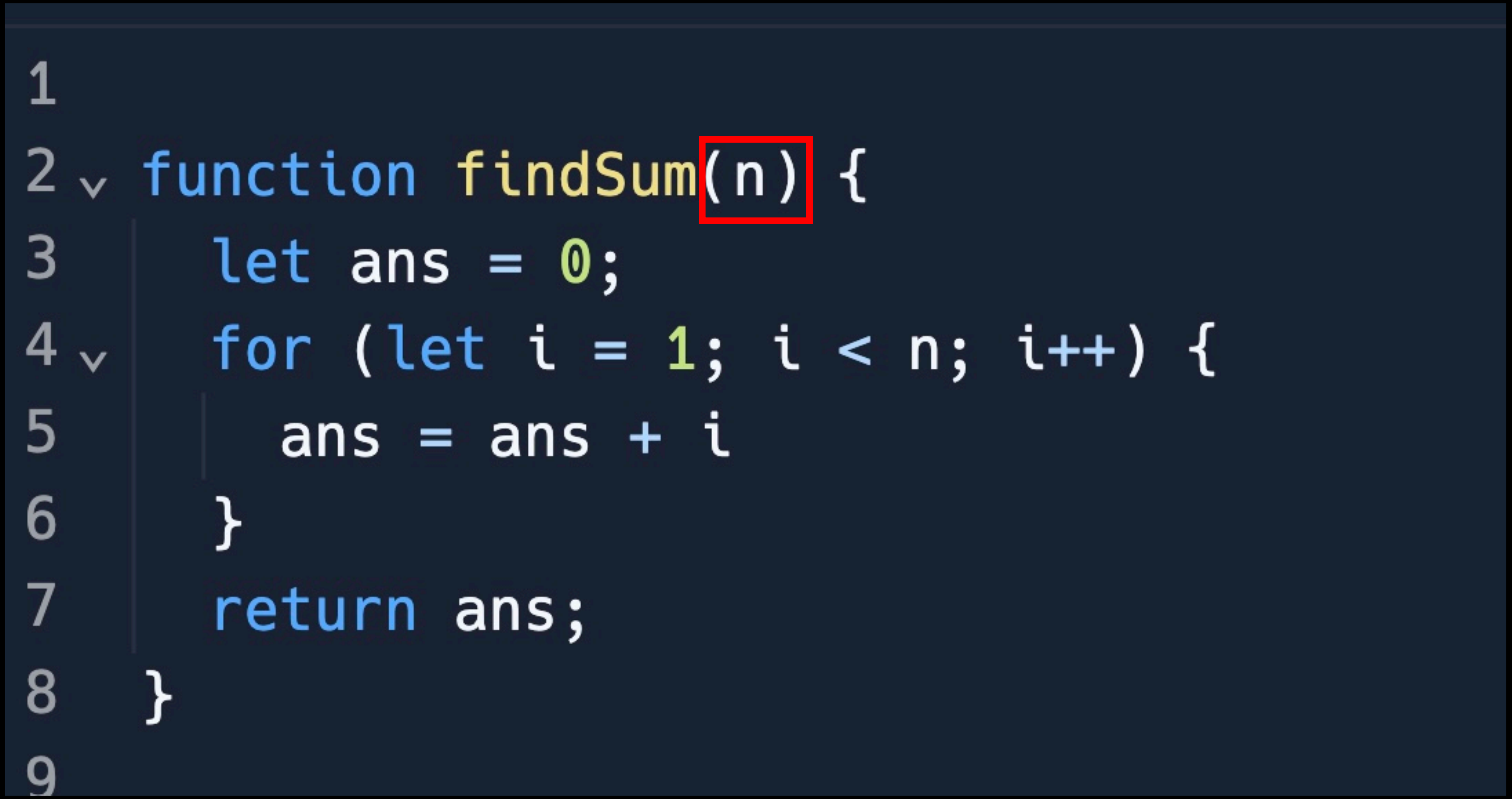


```

function findSum(n) {
  let ans = 0;
  for (let i = 1; i < n; i++) {
    ans = ans + i
  }
  return ans;
}

```

Name of fn

Functions

Syntax?

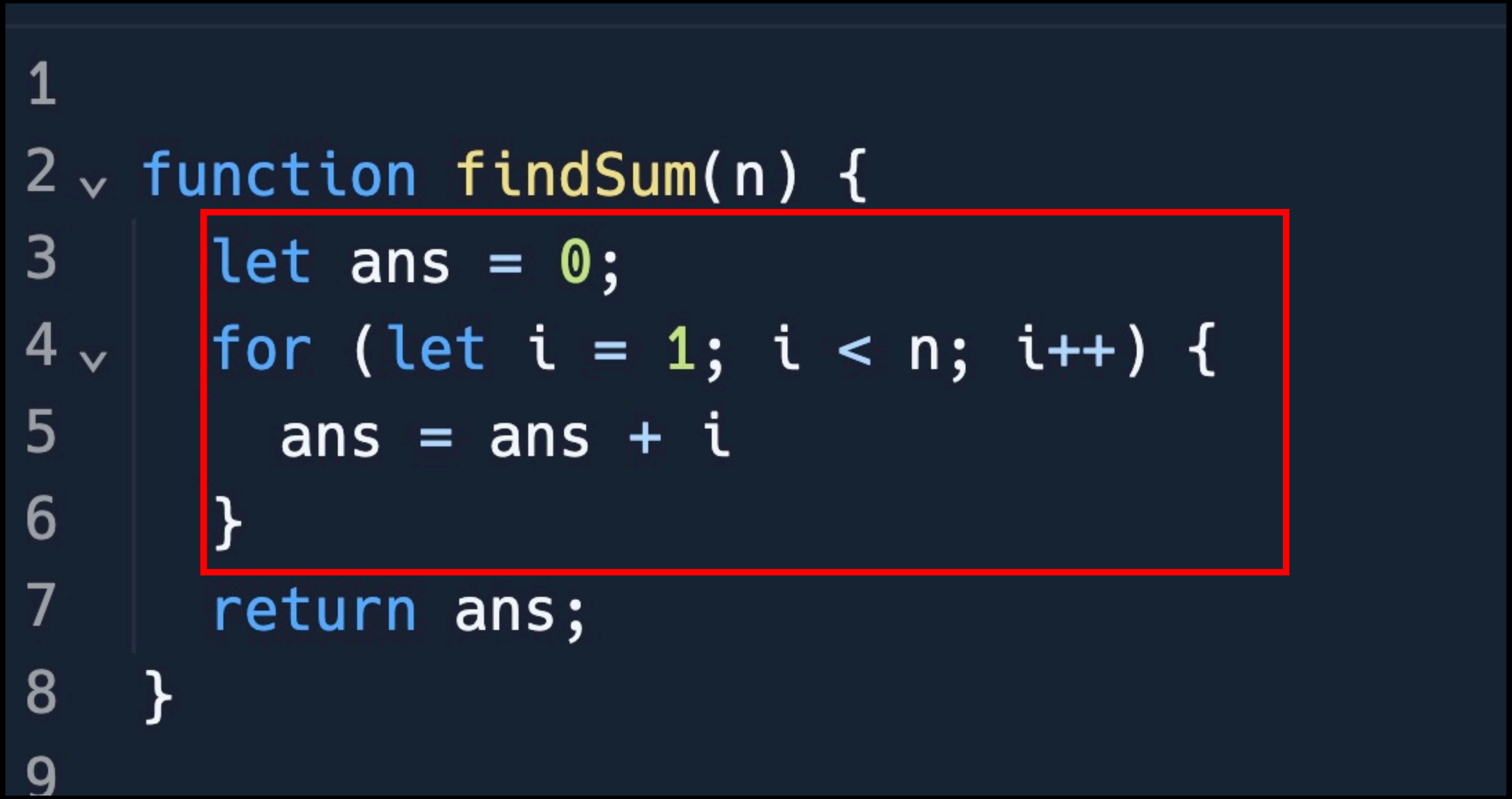


```
function findSum(n) {
  let ans = 0;
  for (let i = 1; i < n; i++) {
    ans = ans + i
  }
  return ans;
}
```

Arguments

Functions

Syntax?

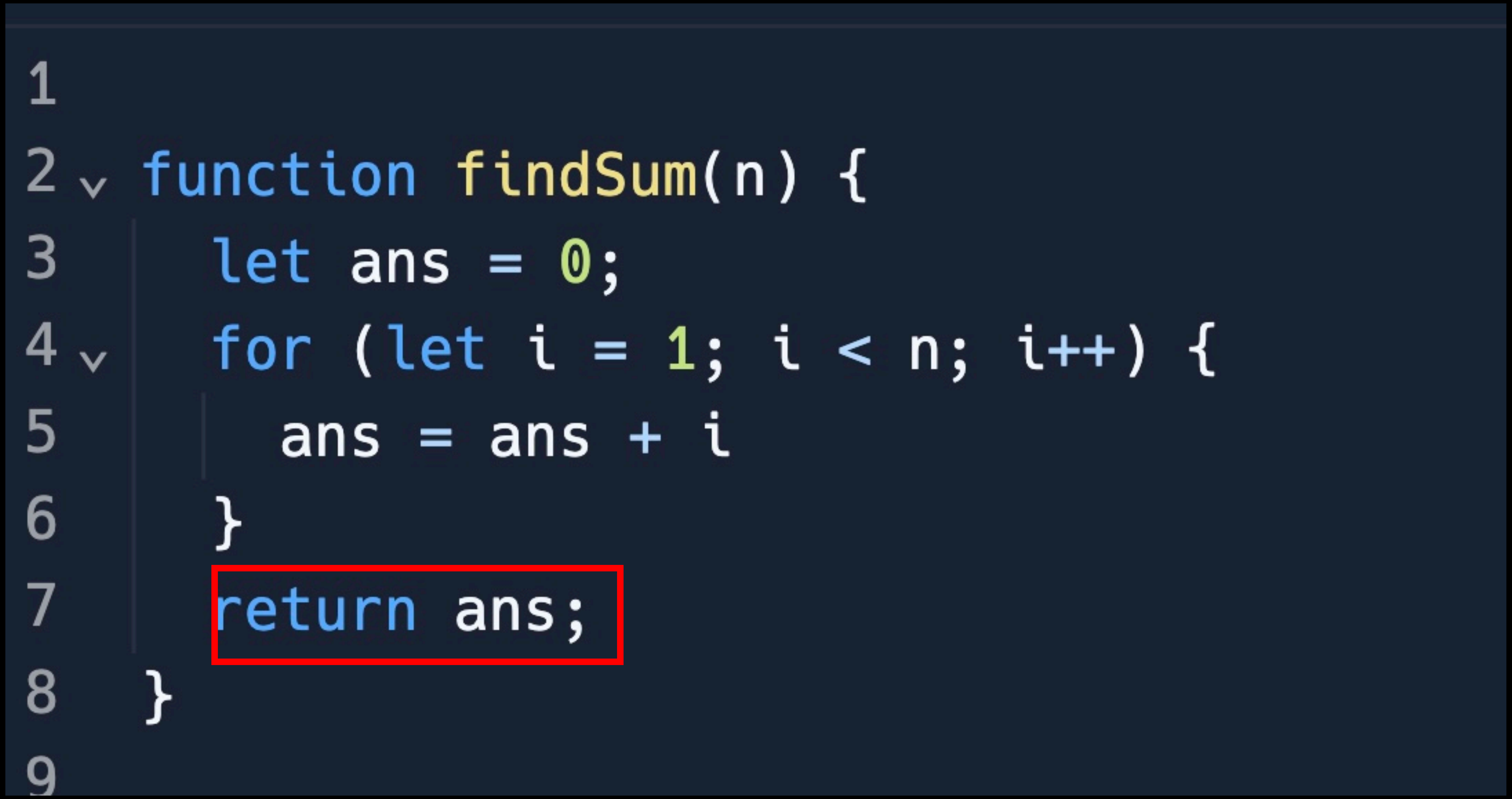


```
function findSum(n) {
  let ans = 0;
  for (let i = 1; i < n; i++) {
    ans = ans + i
  }
  return ans;
}
```

Function body

Functions

Syntax?

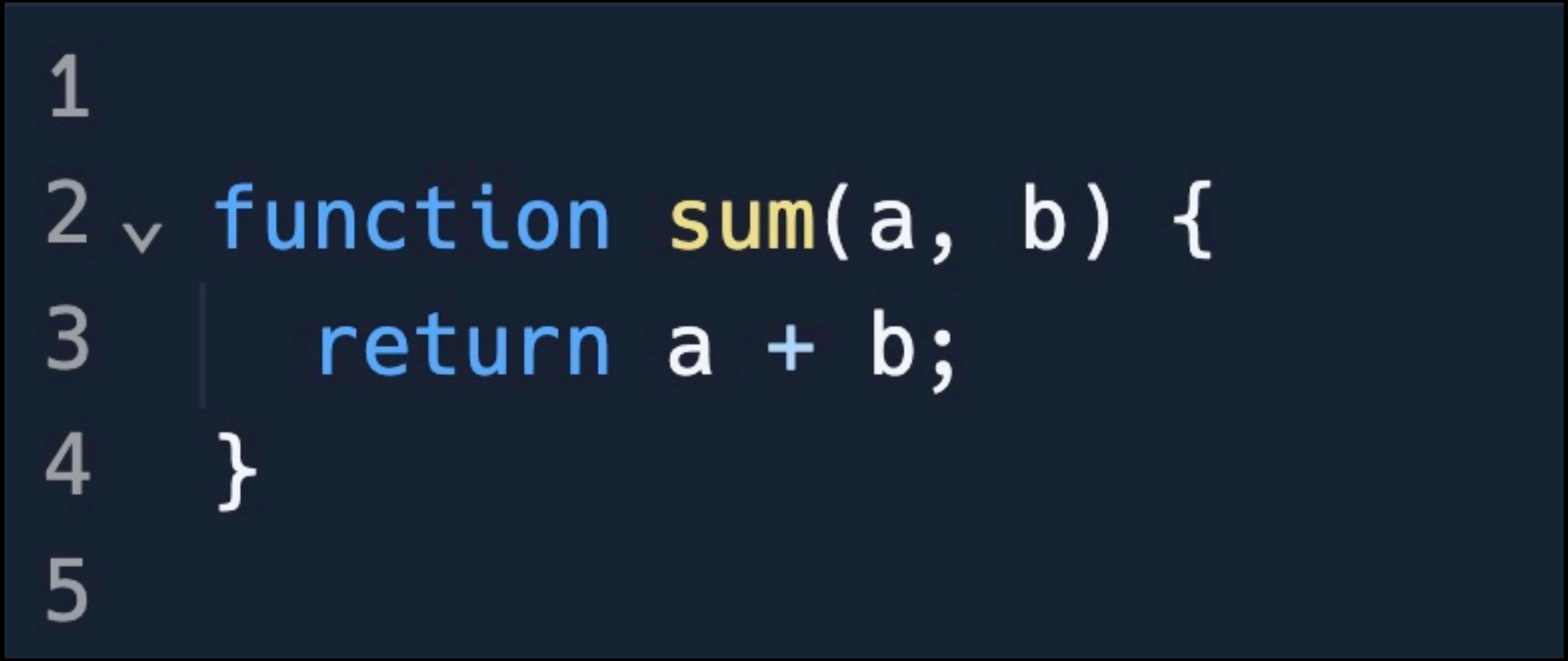


```

function findSum(n) {
  let ans = 0;
  for (let i = 1; i < n; i++) {
    ans = ans + i
  }
  return ans;
}

```

Return value

Functions

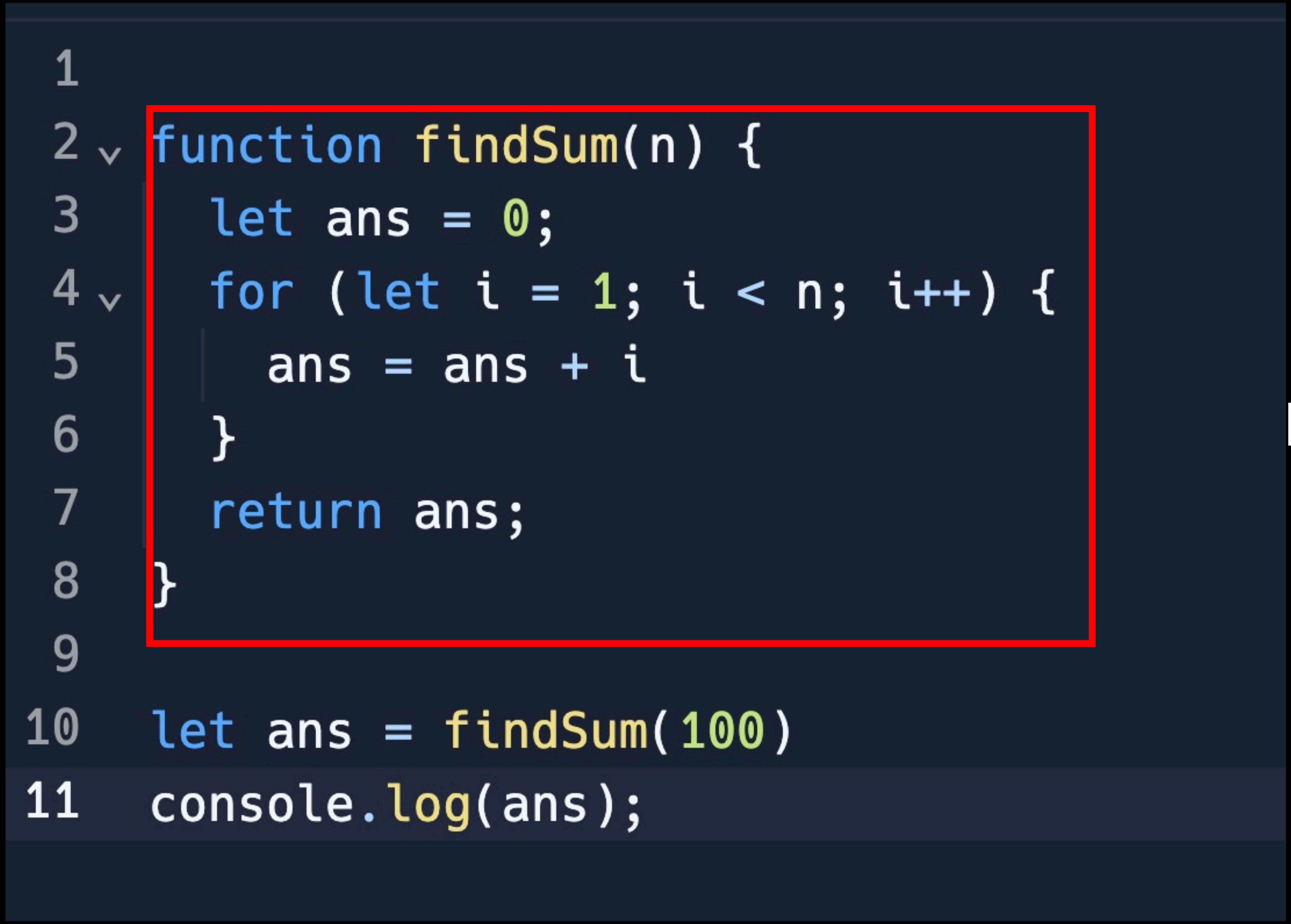
Another example



```

function sum(a, b) {
  return a + b;
}

```

Functions

How to call a fn?

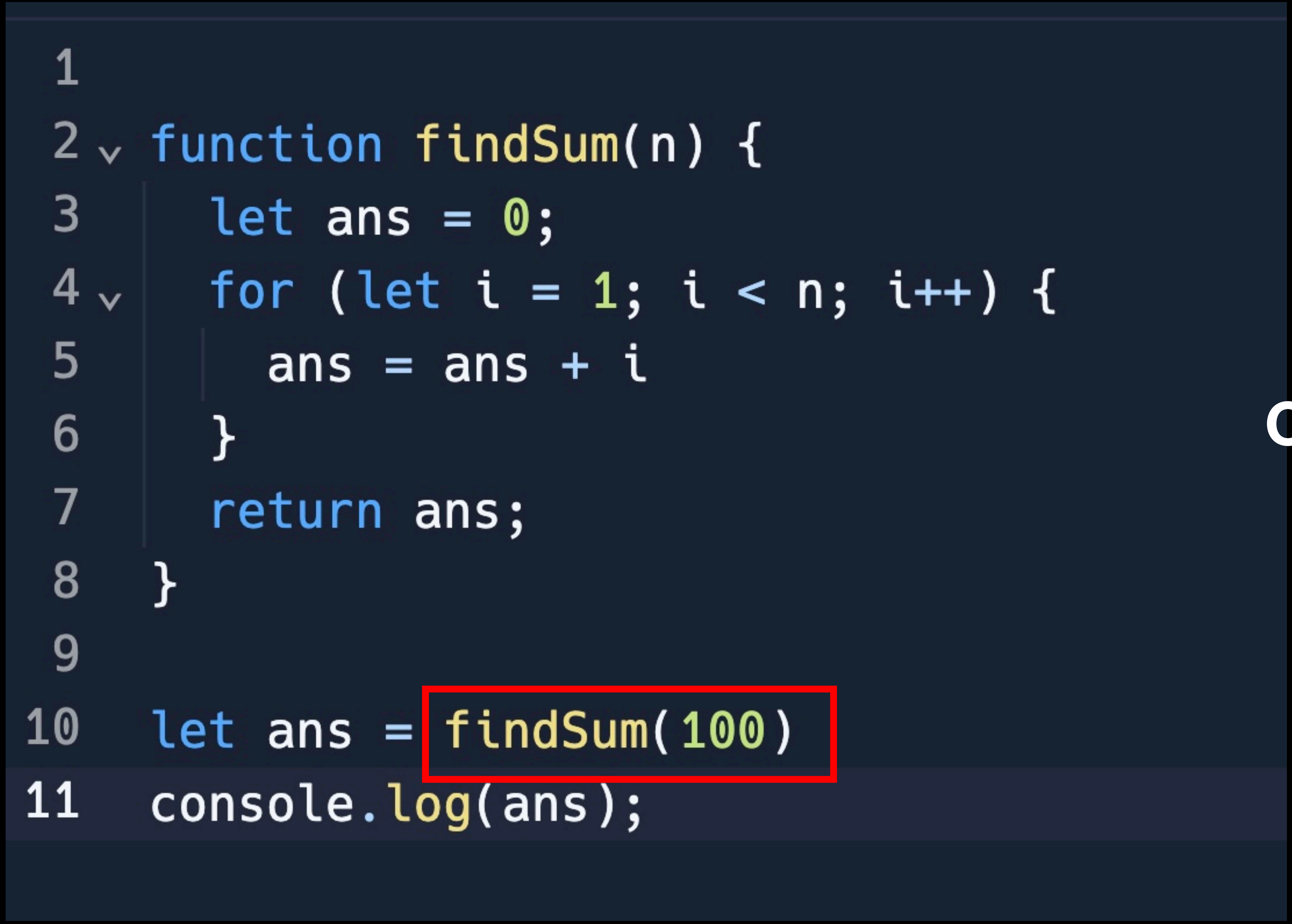


```
function findSum(n) {
  let ans = 0;
  for (let i = 1; i < n; i++) {
    ans = ans + i
  }
  return ans;
}

let ans = findSum(100)
console.log(ans);
```

Function body

Functions

How to call a fn?



```javascript

function findSum(n) {
  let ans = 0;
  for (let i = 1; i < n; i++) {
    ans = ans + i
  }
  return ans;
}

let ans = findSum(100)
console.log(ans);
```

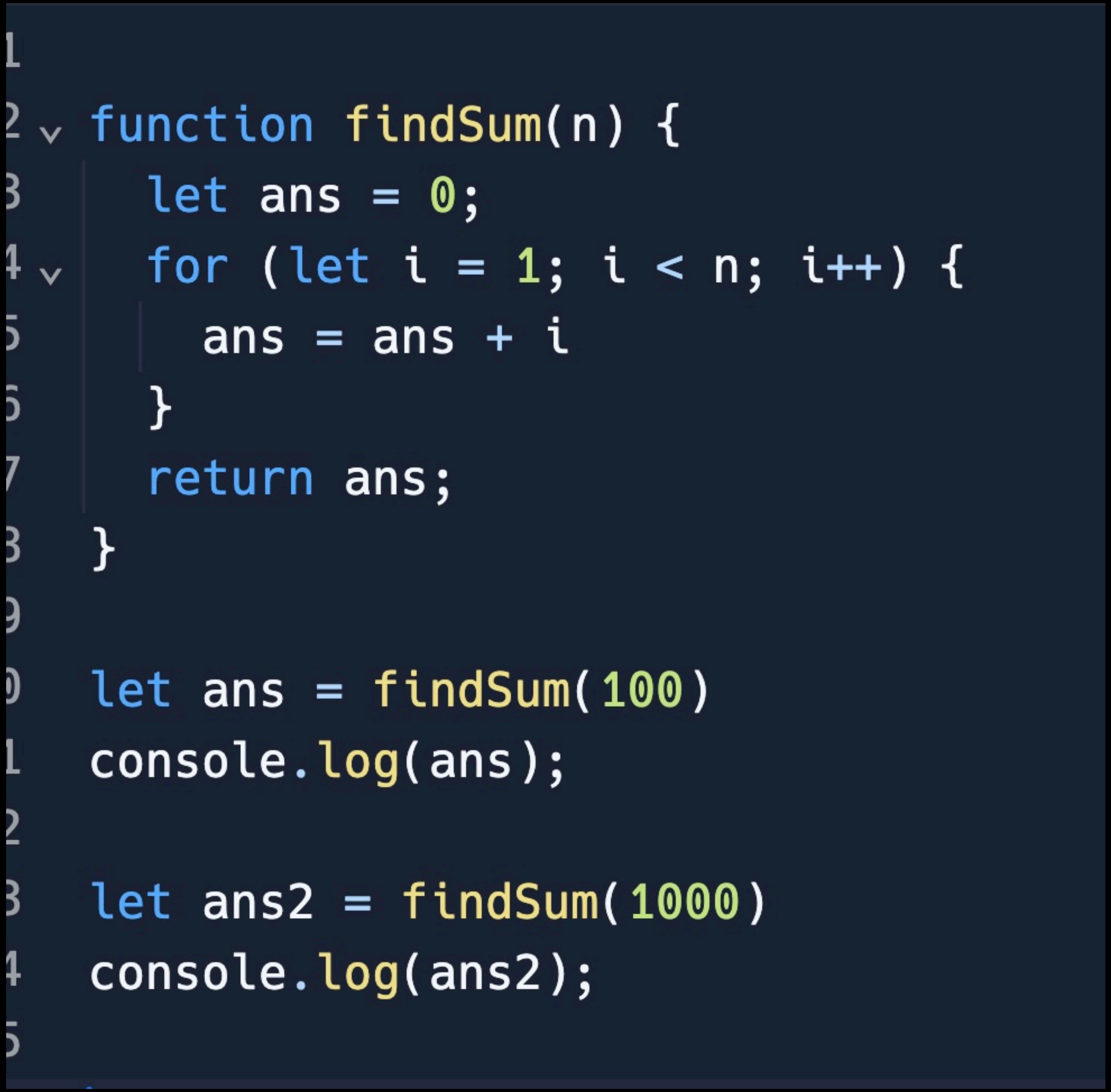
Calling function

Functions

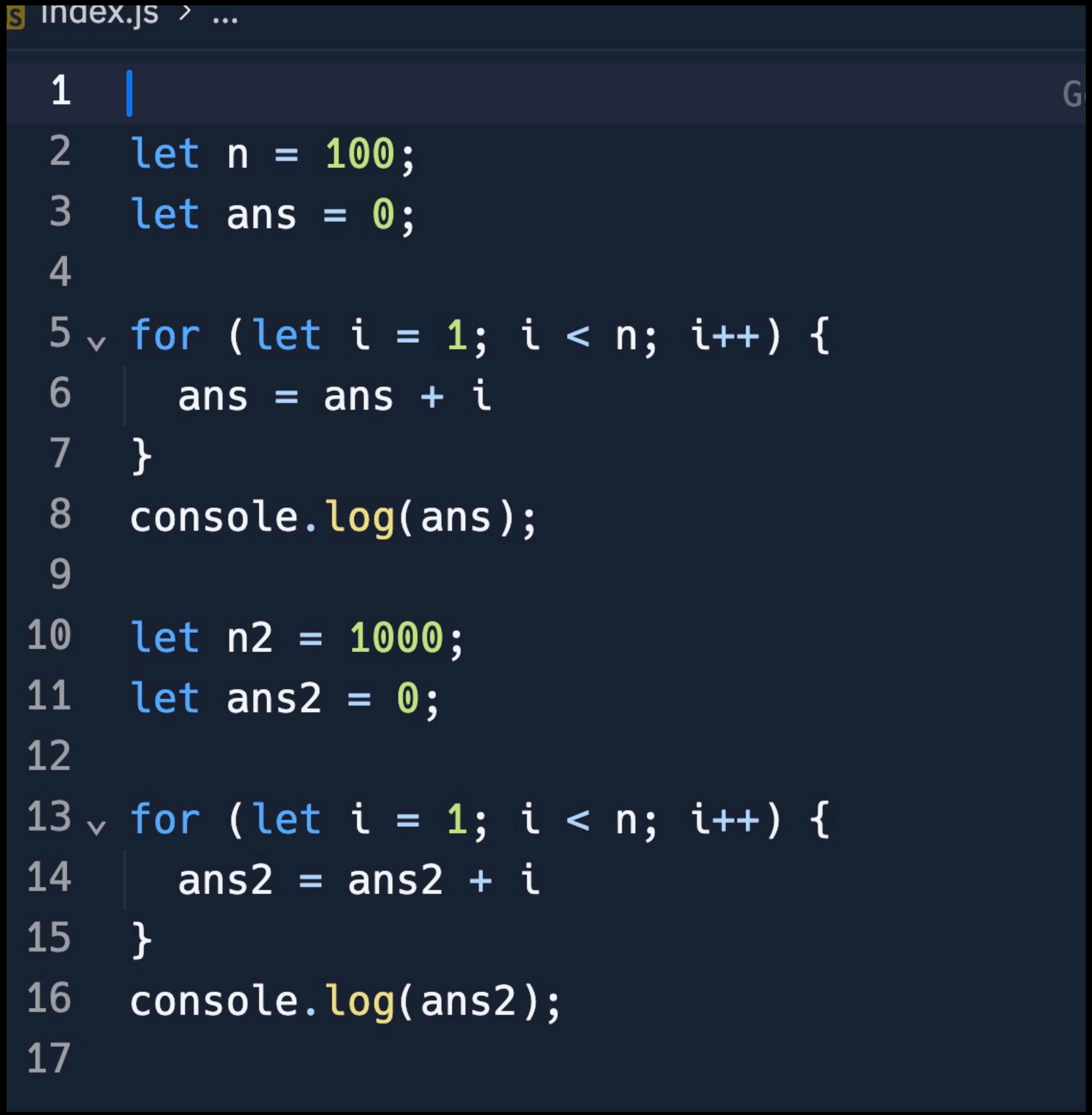
Why do we need functions?

Functions

Why do we need functions?



```
function findSum(n) {
  let ans = 0;
  for (let i = 1; i < n; i++) {
    ans = ans + i
  }
  return ans;
}

let ans = findSum(100)
console.log(ans);

let ans2 = findSum(1000)
console.log(ans2);
```

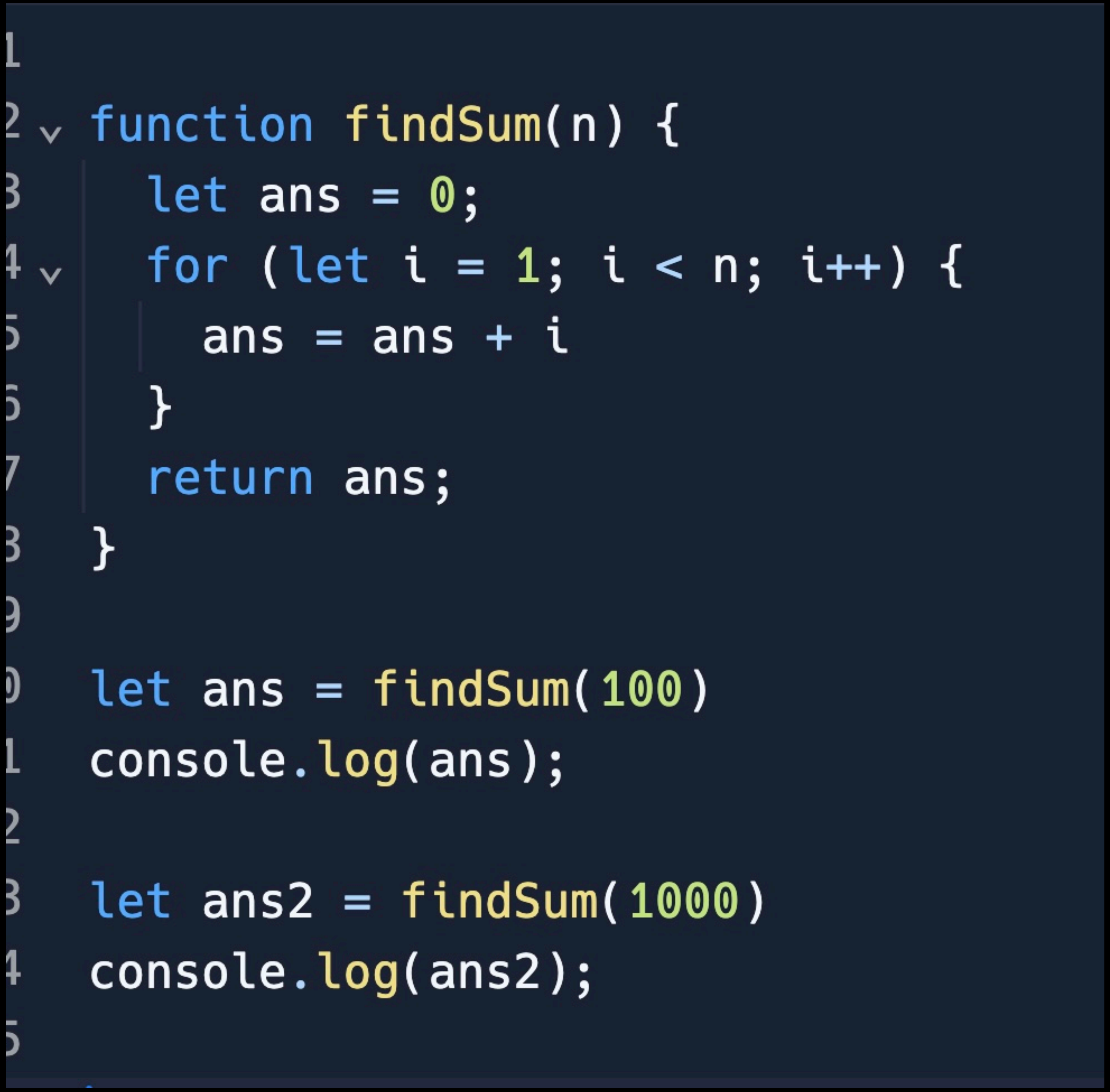


index.js › ...

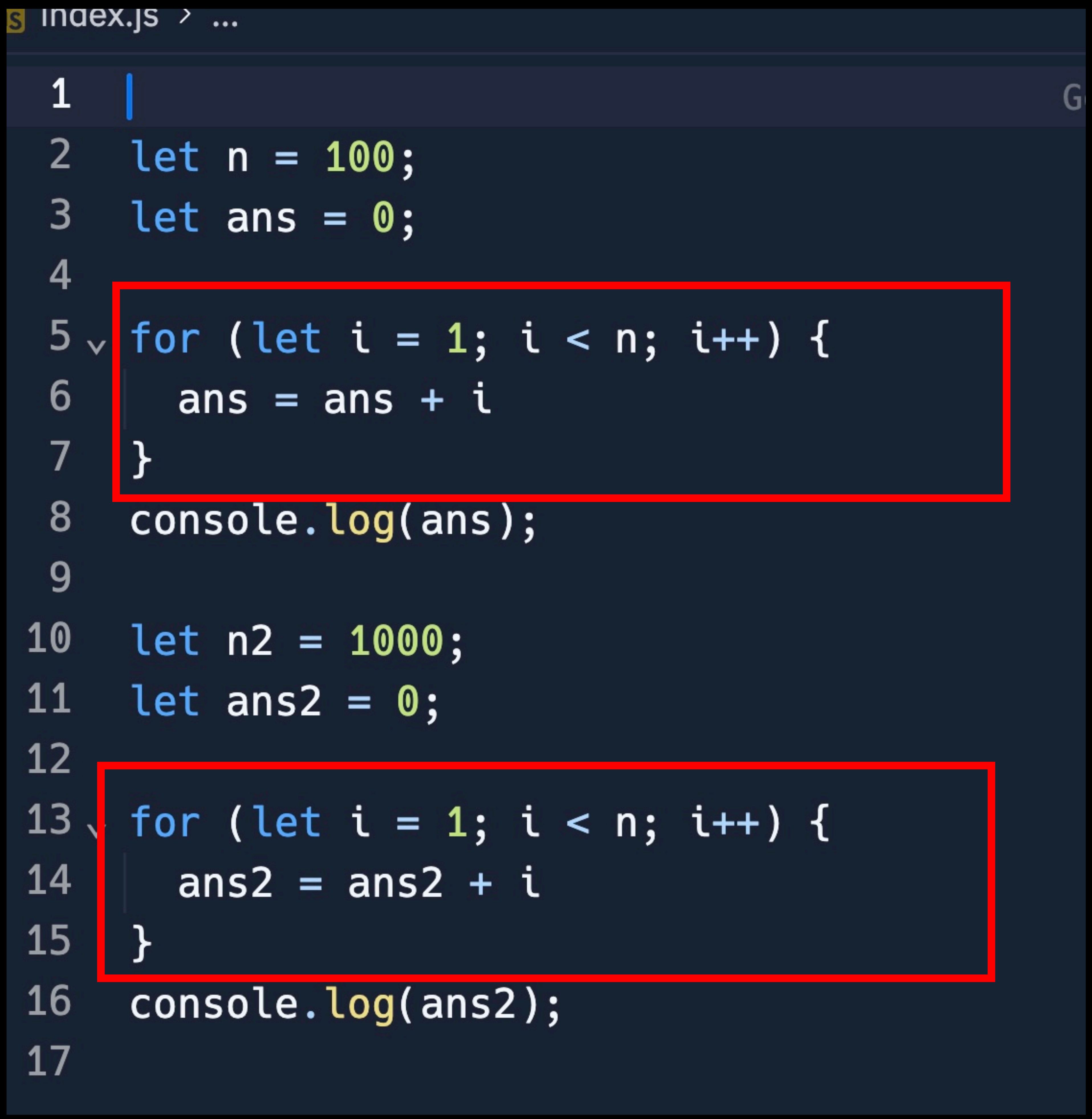
```

let n = 100;
let ans = 0;

for (let i = 1; i < n; i++) {
  ans = ans + i
}
console.log(ans);

let n2 = 1000;
let ans2 = 0;

for (let i = 1; i < n; i++) {
  ans2 = ans2 + i
}
console.log(ans2);

```

Functions

You are repeating yourself (DRY)



```
function findSum(n) {
  let ans = 0;
  for (let i = 1; i < n; i++) {
    ans = ans + i
  }
  return ans;
}

let ans = findSum(100)
console.log(ans);

let ans2 = findSum(1000)
console.log(ans2);
```



index.js › ...

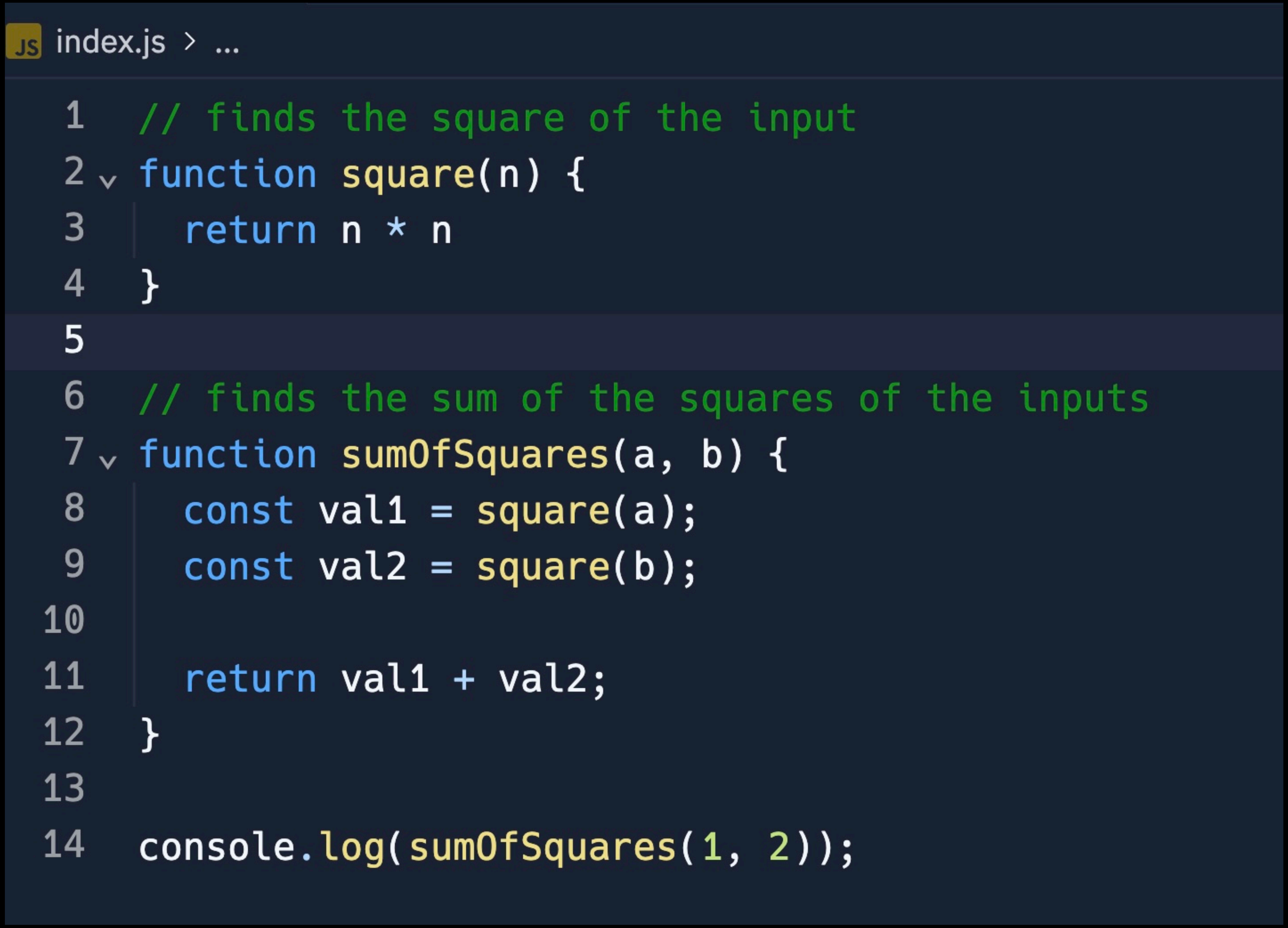
```

let n = 100;
let ans = 0;

for (let i = 1; i < n; i++) {
  ans = ans + i
}
console.log(ans);

let n2 = 1000;
let ans2 = 0;

for (let i = 1; i < n; i++) {
  ans2 = ans2 + i
}
console.log(ans2);

```

Callback Functions

Step 1 - Can you call one function inside another function?

Callback Functions

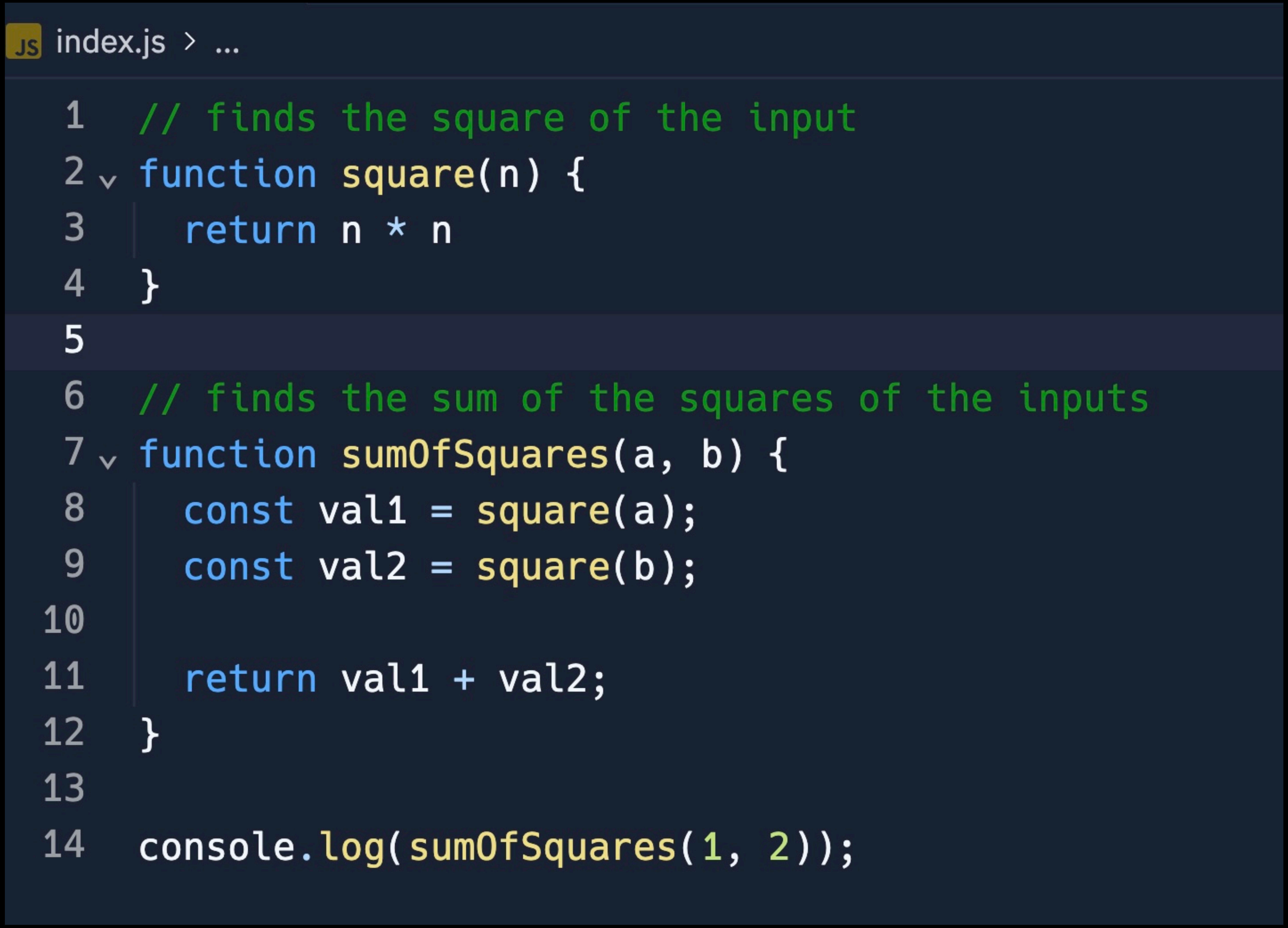
Step 1 - Can you call one function inside another function?
Yes



```javascript
// finds the square of the input
function square(n) {
  return n * n
}

// finds the sum of the squares of the inputs
function sumOfSquares(a, b) {
  const val1 = square(a);
  const val2 = square(b);

  return val1 + val2;
}

console.log(sumOfSquares(1, 2));
```

Callback Functions

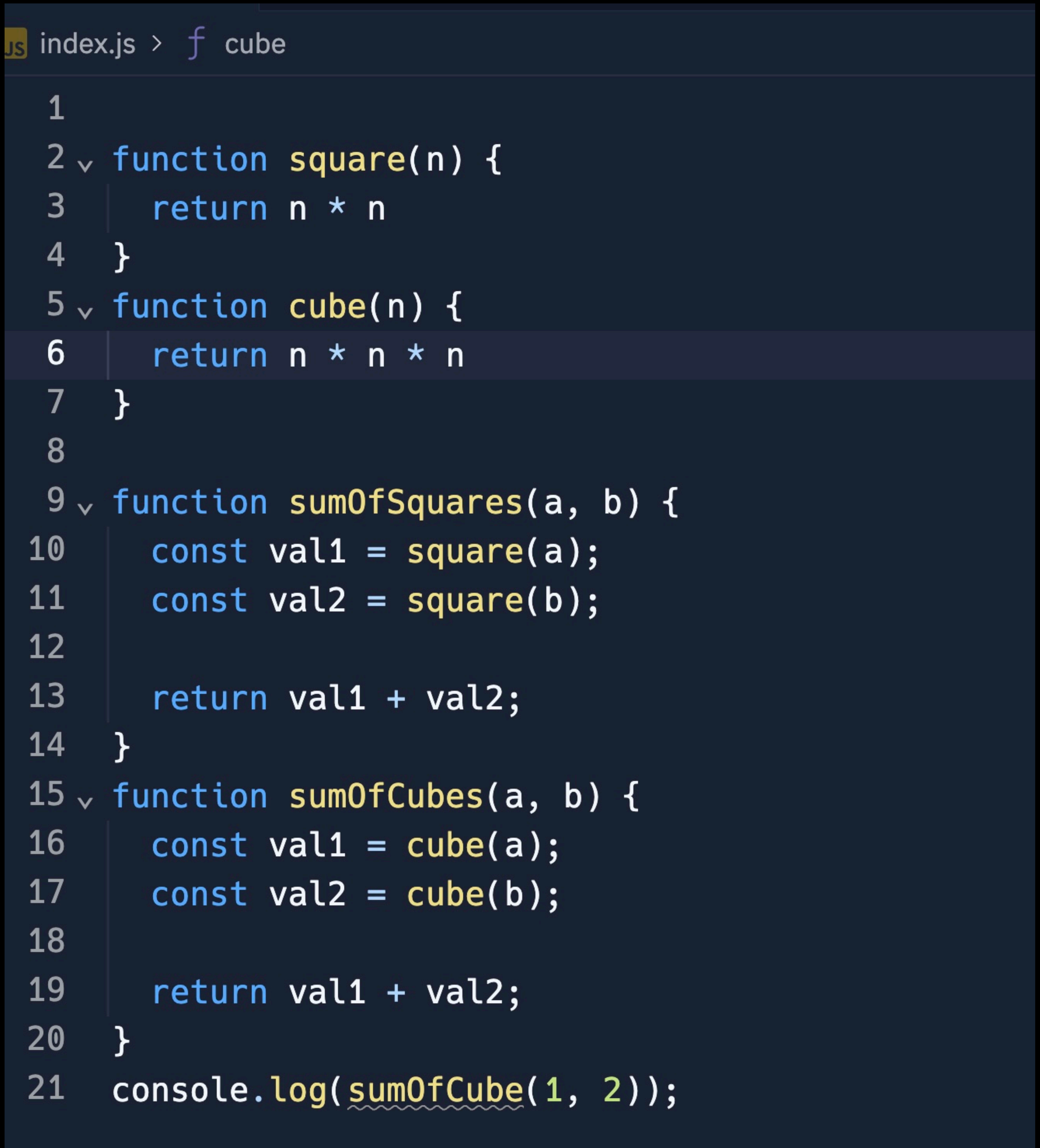
Step 1 - Can you call one function inside another function?
Yes



index.js > ...

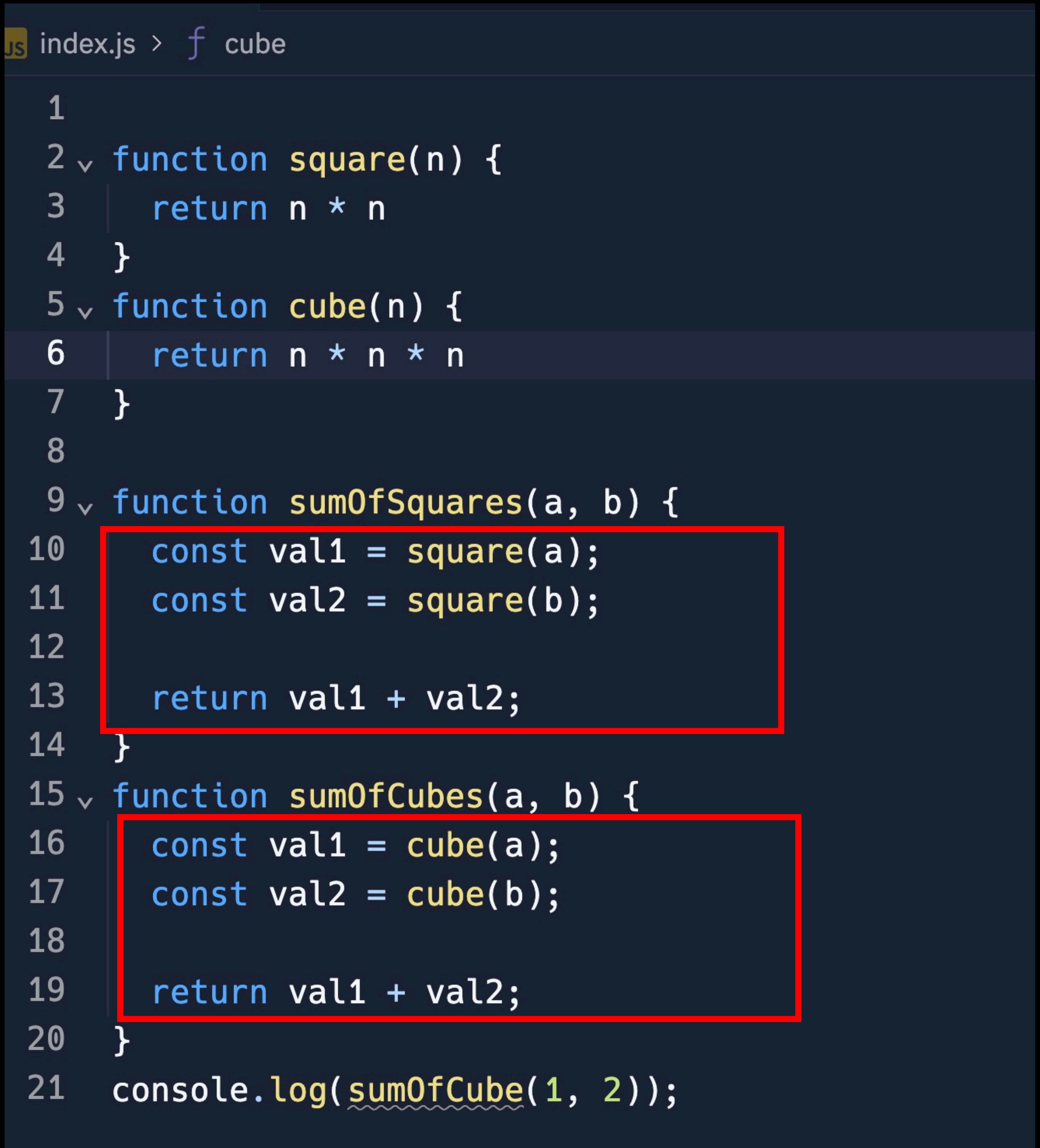
```javascript
// finds the square of the input
function square(n) {
  return n * n
}

// finds the sum of the squares of the inputs
function sumOfSquares(a, b) {
  const val1 = square(a);
  const val2 = square(b);

  return val1 + val2;
}

console.log(sumOfSquares(1, 2));
```

Callback Functions



```javascript

function square(n) {
  return n * n
}
function cube(n) {
  return n * n * n
}

function sumOfSquares(a, b) {
  const val1 = square(a);
  const val2 = square(b);

  return val1 + val2;
}
function sumOfCubes(a, b) {
  const val1 = cube(a);
  const val2 = cube(b);

  return val1 + val2;
}
console.log(sumOfCube(1, 2));
```

Callback Functions

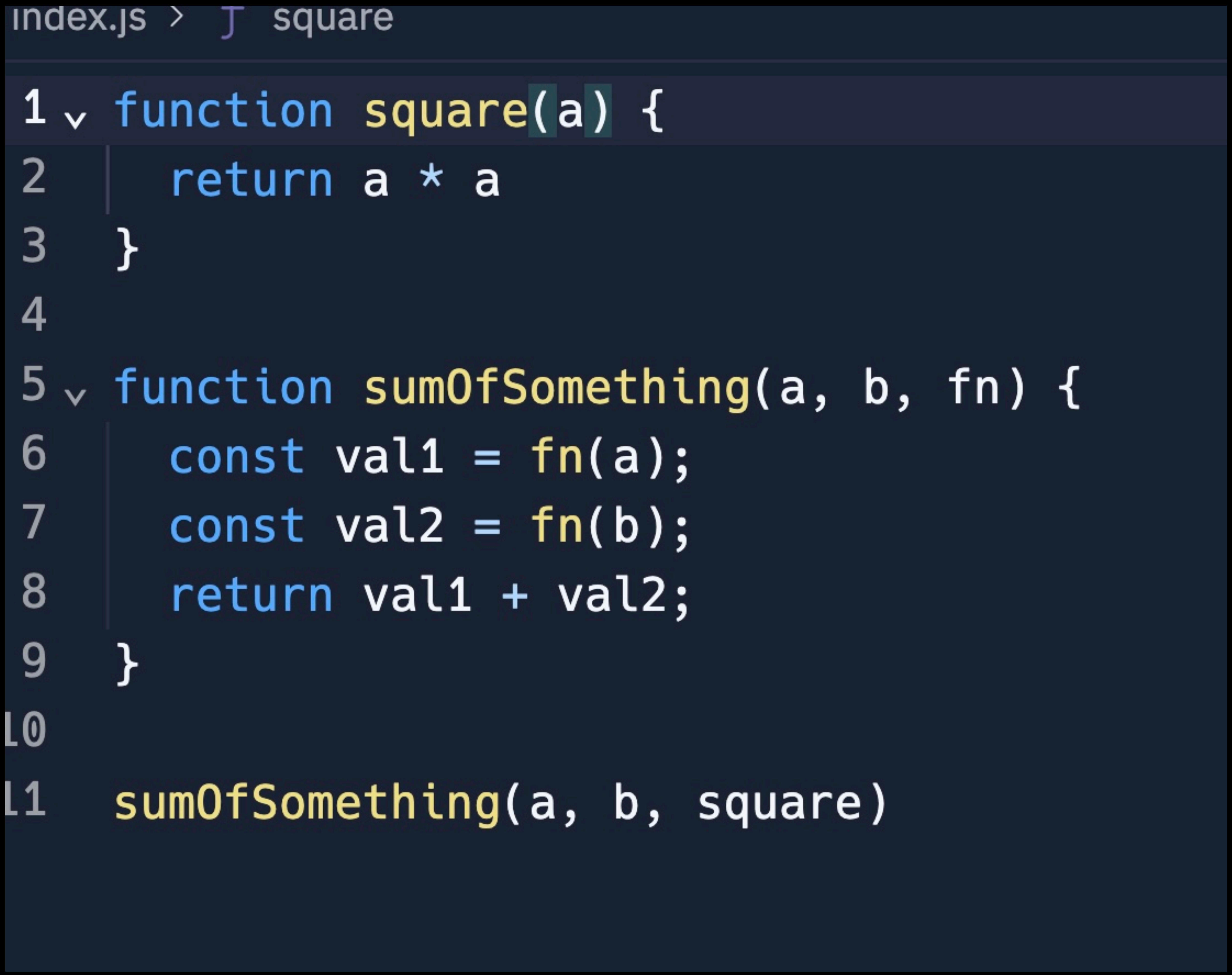


index.js > cube

```javascript

function square(n) {
  return n * n
}
function cube(n) {
  return n * n * n
}

function sumOfSquares(a, b) {
  const val1 = square(a);
  const val2 = square(b);

  return val1 + val2;
}
function sumOfCubes(a, b) {
  const val1 = cube(a);
  const val2 = cube(b);

  return val1 + val2;
}
console.log(sumOfCube(1, 2));
```

Is DRY being violated here?

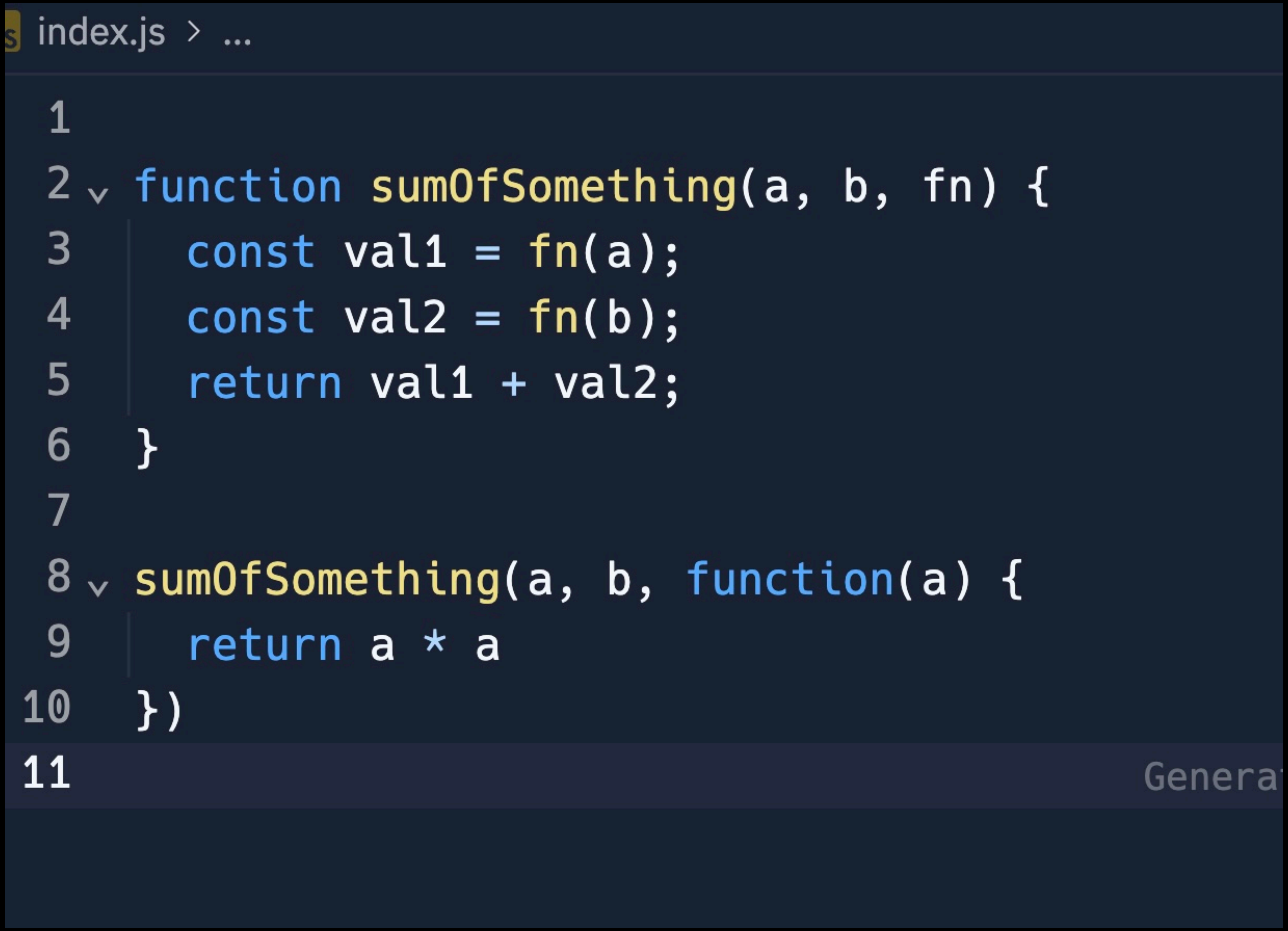
Callback Functions


index.js > square

```
function square(a) {
  return a * a
}

function sumOfSomething(a, b, fn) {
  const val1 = fn(a);
  const val2 = fn(b);
  return val1 + val2;
}

sumOfSomething(a, b, square)
```

Solution

Anonymous functions


index.js > ...

```
function sumOfSomething(a, b, fn) {
  const val1 = fn(a);
  const val2 = fn(b);
  return val1 + val2;
}

sumOfSomething(a, b, function(a) {
  return a * a
})
```