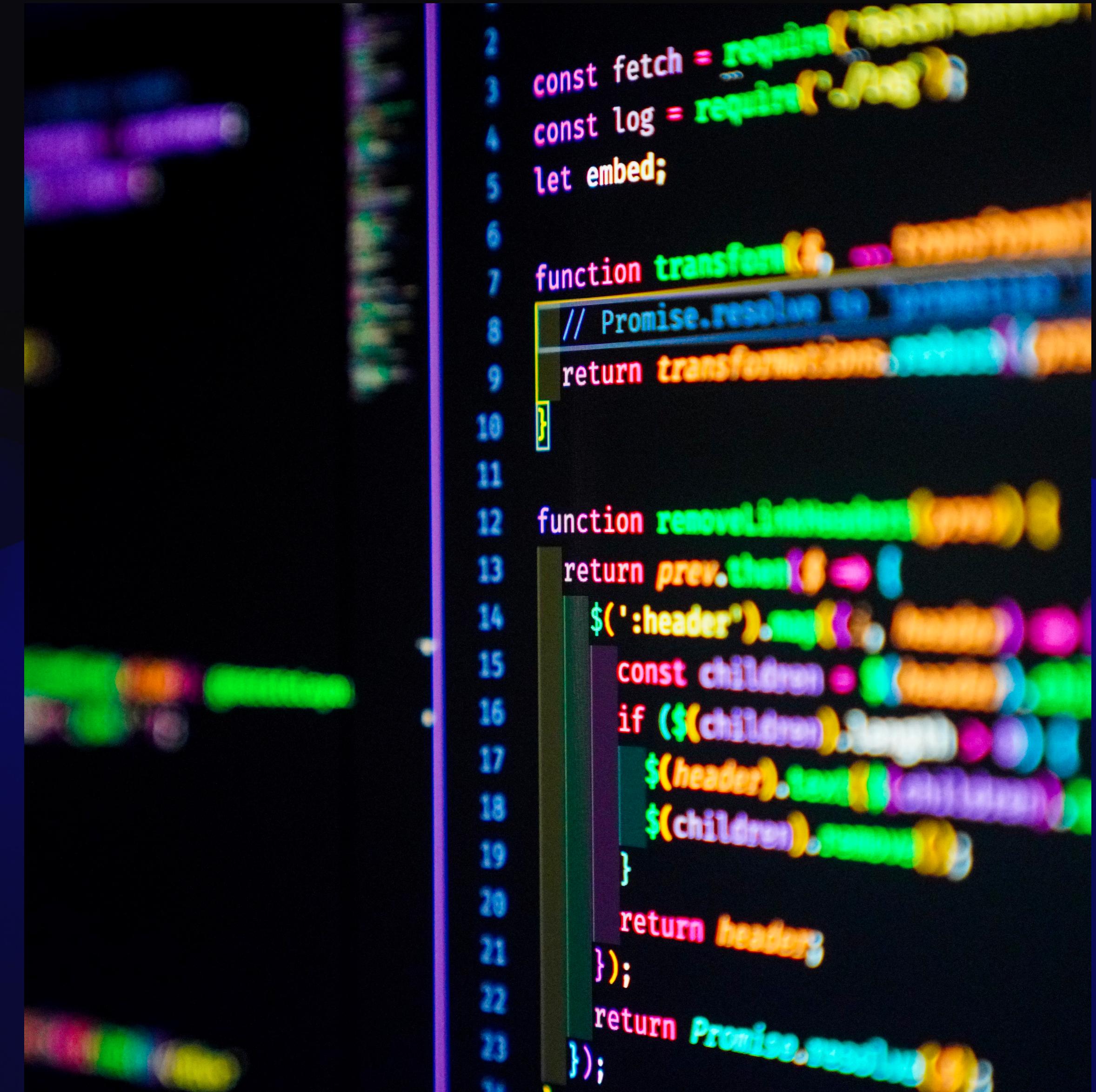


Javascript

Arrays & Controle de fluxo



```
2 const fetch = require('node-fetch');
3 const log = require('loglevel');
4 let embed;
5
6
7 function transform(...args) {
8   // Promise.resolve is required here
9   return transformPromise(...args);
10 }
11
12 function removeChildren(element) {
13   return prev.then($ => {
14     const children = $().children();
15     if ($().children().length === 0) {
16       $(element).append($());
17     } else {
18       $(element).append($().children());
19     }
20   });
21 }
22
23 return Promise.resolve();
24 }
```

Arrays

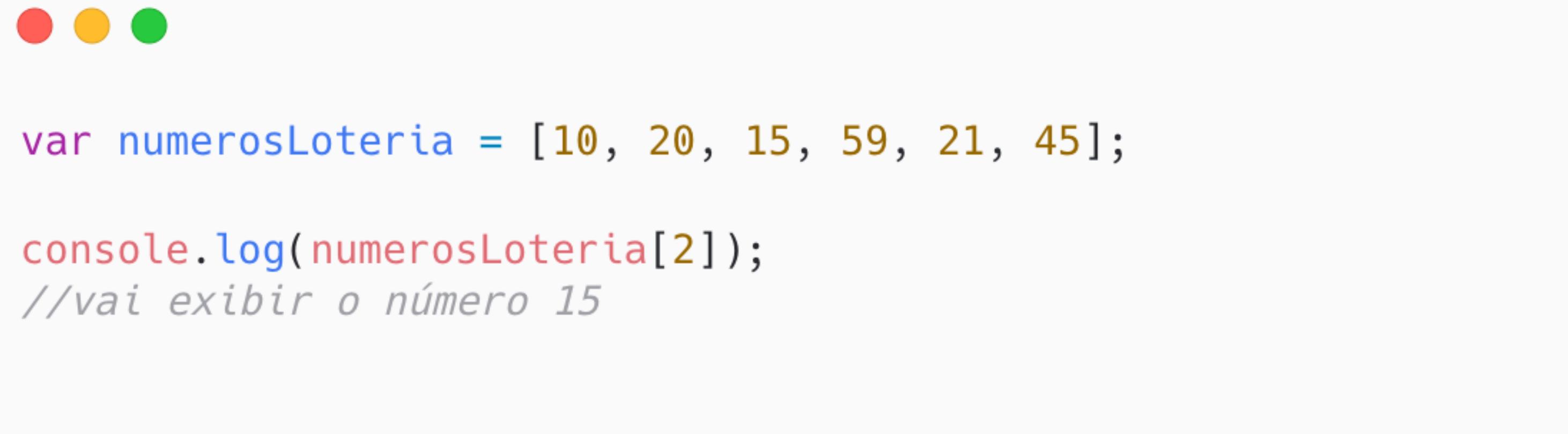
Estrutura de dados que permite armazena múltiplos valores em conjunto.



```
var numerosLoteria = [10, 20, 15, 59, 21, 45];  
  
console.log(numerosLoteria);
```

Arrays

Você pode acessar os valores da array usando colchetes []. A contagem dos valores de uma array SEMPRE começa em zero.



The image shows a screenshot of a browser's developer tools console. It features three colored circular icons (red, yellow, green) at the top left. The console itself has a white background and contains the following code:

```
var numerosLoteria = [10, 20, 15, 59, 21, 45];
console.log(numerosLoteria[2]);
//vai exibir o número 15
```

The code defines an array named 'numerosLoteria' with six elements: 10, 20, 15, 59, 21, and 45. The second line uses the `console.log` function to print the third element of the array, which is 15. A comment below the log statement explains that it will display the number 15.

Arrays

O javascript fornece diversas funções para manipular arrays.

https://developer.mozilla.org/pt-BR/docs/Web/JavaScript/Reference/Global_Objects/Array

```
var numerosLoteria = [10, 20, 15, 59, 21, 45];

//escreve a quantidade total de items no array
console.log("Total de items: ", numerosLoteria.length);

//adiciona um item no final da array
numerosLoteria.push(11);

//remove o ultimo item da array
numerosLoteria.pop();
```

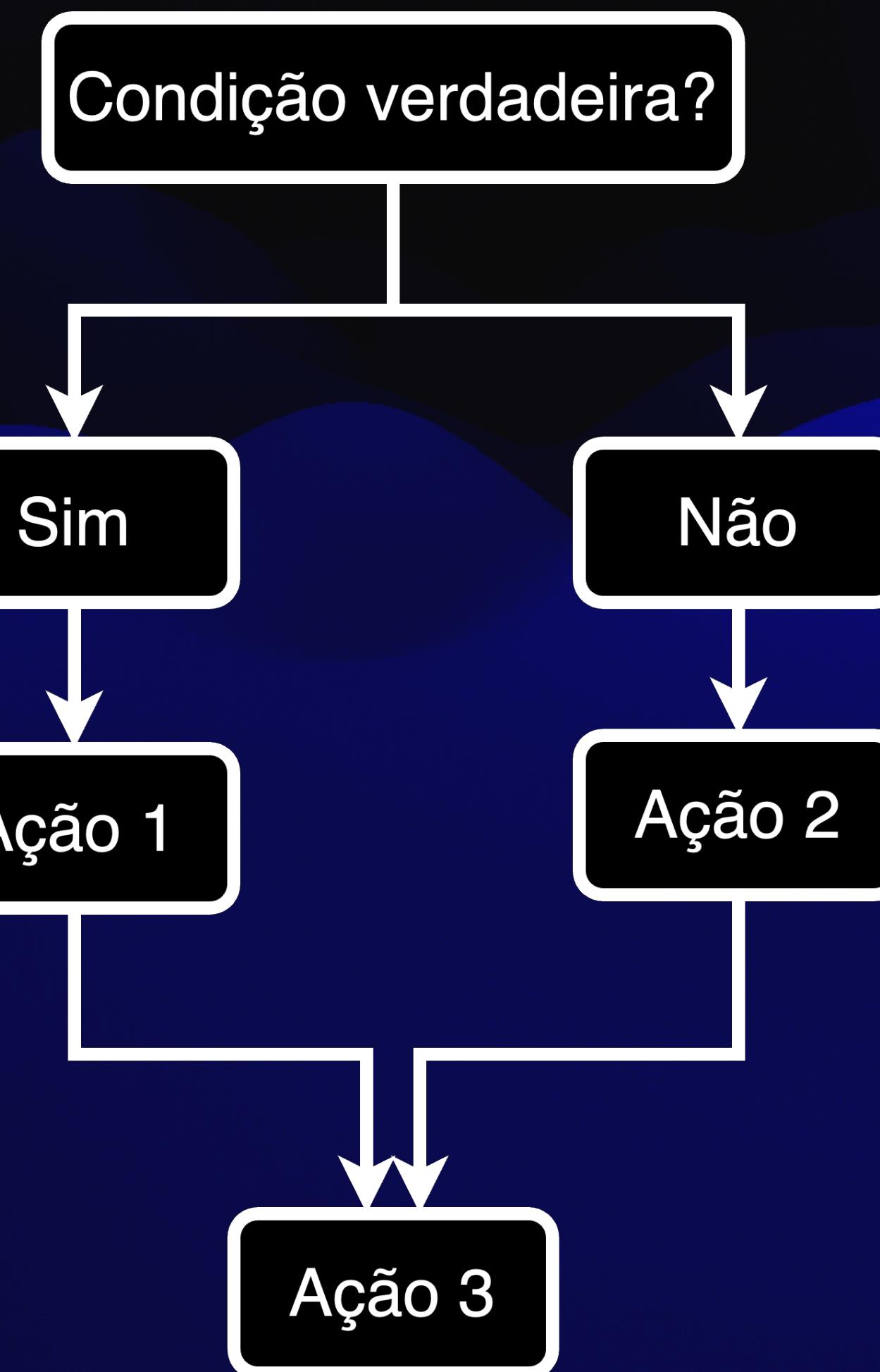
Controle de fluxo

Usamos estruturas como if, for, while etc para determinar o fluxo de um algoritmo.

Ao lado vemos um exemplo de fluxo if / else

```
if(/*condicao*){
    //acao 1
} else {
    //acao 2
}

//acao 3
```

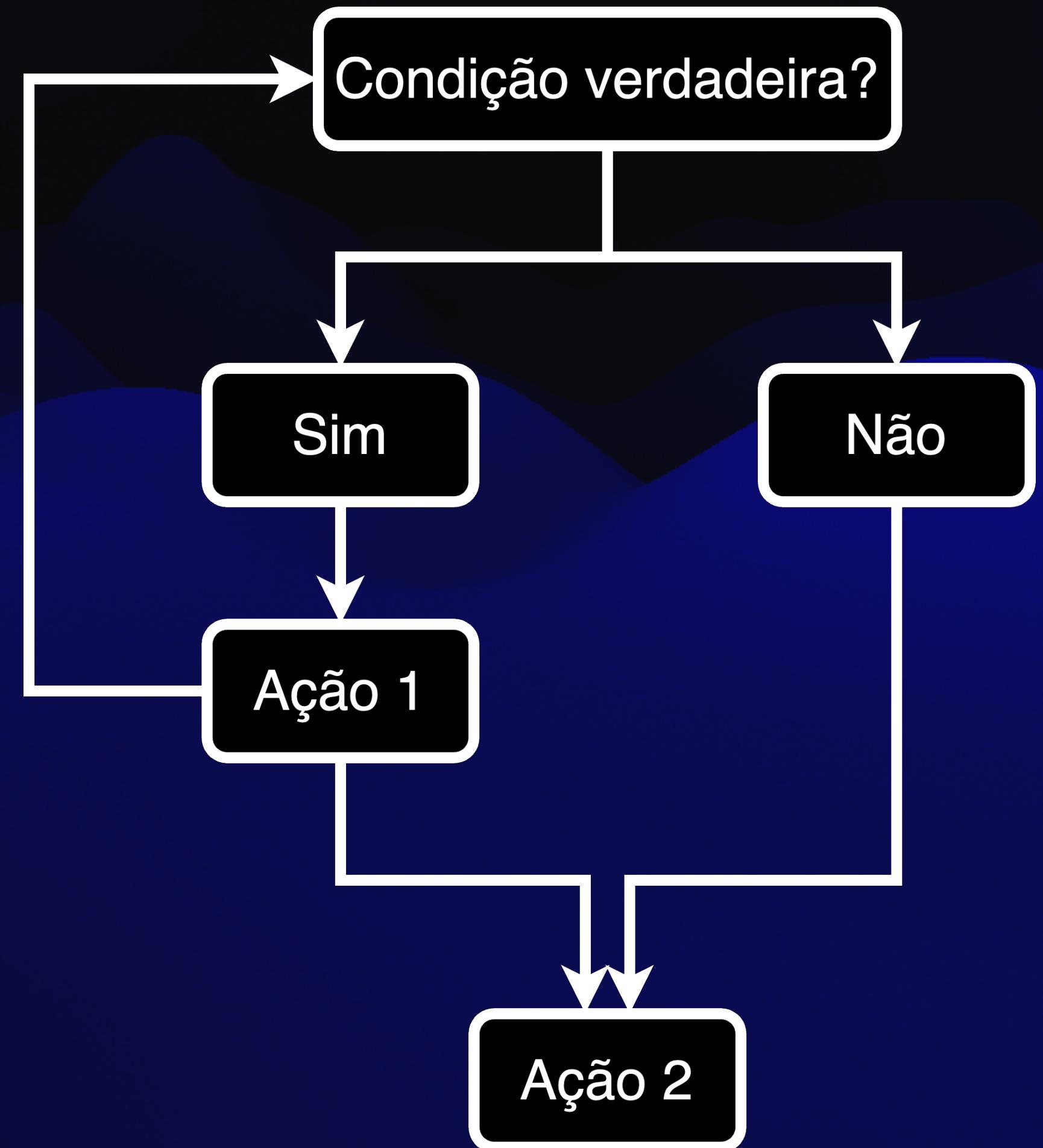


Controle de fluxo

Quando usamos um controle de fluxo com loop, como por exemplo for e while, as instruções serão repetidas até que a condição deixe de ser verdadeira.



```
while(/*condicao*\/){  
    //acao 1  
}  
  
//acao 2
```



While

Executa as instruções sem até que a condição não seja verdadeira.

```
var contador = 0;
var limite = 10;

while(contador <= limite){
    console.log(contador);
    contador++;
}
```