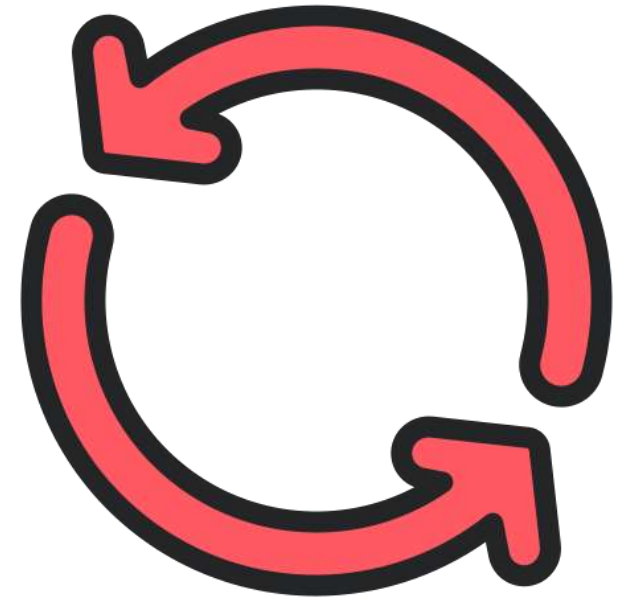


FrontEnd

Laços de repetição

São uma das principais características da programação, e está presente em grande parte das linguagens conhecidas hoje.

Permite que um bloco de código seja repetido diversas vezes até que uma condição seja satisfeita.



Laços de repetição

Eles são essenciais para a resolução de uma grande variedade de problemas de programação, como a repetição de uma ação, a iteração em uma coleção de itens ou a espera por uma condição ser satisfeita.

Em JavaScript, temos três estruturas principais de laços de repetição:

- FOR
- WHILE
- DO-WHILE.

FrontEnd - JavaScript

For

```
for ( let i = 1 ; i <= 10 ; i++){  
    console.log(i);  
}
```

Explicação: Este código inicia uma variável *i* em 1 e, enquanto *i* for menor ou igual a 10, imprime *i* no console e depois incrementa *i* em 1. O laço for é especialmente útil quando sabemos exatamente quantas vezes queremos que nosso código seja executado.

While

```
contador = 20
while (contador <= limite) { // Não entra no laço
  console.log(`${contador}ª iteração`)
  contador++
}
```

Explicação: Este código inicia uma variável contador em 20 e **enquanto uma condição seja verdadeira**, exibe o resultado no console e incrementa i em 1.

DO - While

```
let i = 1;  
do {  
    if (i % 2 !== 0){  
        console.log(i);  
    }  
    i++;  
} while (i <= 20);
```

Explicação: Este código inicia uma variável *i* em 1 e faz uma verificação, se o número é ímpar, então exibe o resultado no console e incrementa *i* em 1.

O Do While sempre vai executar pelo menos a 1 vez, e depois valida a condição do While

FrontEnd - JavaScript



FrontEnd - JavaScript

```
function getRandomArbitrary(min, max) {  
  let ganhou;  
  do {  
    let numero = Number.parseInt(Math.random() * (max - min) + min);  
    alert(numero)  
    ganhou = prompt("Houve ganhador? Se Sim digite 's', se não, digite 'n'");  
    if (ganhou === 'n') {  
      continue; // Continua para a próxima iteração do loop  
    } else {  
      return "Parabéns, houve um vencedor!";  
    }  
  } while (ganhou === 'n');  
}
```


FrontEnd - JavaScript

Exercícios

Neste exercício, você vai simular um jogo onde o usuário "pensa" em um número de 1 a 100 e o programa deve adivinhar o número, dando um número exato de chutes possíveis.

Neste exercício, você vai simular um jogo de "Pedra, papel e tesoura" contra o computador. Para simplificar, vamos supor que a escolha do usuário é sempre "pedra".

Neste exercício, você precisa imprimir os primeiros 10 números da sequência de Fibonacci. A sequência de Fibonacci é uma sequência de números onde cada número subsequente é a soma dos dois anteriores. Ela começa assim: 0, 1, 1, 2, 3, 5, 8, 13...

FrontEnd - JavaScript

Exercícios

Pedir um número ao usuário de 1 a 9, e imprima a tabuada desse valor.

Neste exercício, você deve calcular e imprimir a soma dos cubos dos primeiros 10 números naturais. Os números naturais são os números positivos começando em 1, então os primeiros 10 números naturais são 1, 2, 3, ..., 10, e os seus cubos são 1, 8, 27, ..., 1000.

Neste exercício, você precisa criar um programa que calcula o fatorial de um número. O fatorial de um número é o produto de todos os números inteiros positivos de 1 até o número. Por exemplo, o fatorial de 5 (representado por 5!) é $1 * 2 * 3 * 4 * 5 = 120$.