

## 4- Loops

Agora vamos falar sobre loops em JavaScript. Loops são ferramentas poderosas que permitem executar um bloco de código repetidamente sob determinadas condições. Vamos entender tudo sobre eles!

### O Que são Loops?

Em programação, um loop é uma estrutura que repete um bloco de código enquanto uma condição específica é atendida. No JavaScript, temos vários tipos de loops, cada um com suas peculiaridades e casos de uso.

### Por Que Usamos Loops?

Loops são usados para realizar tarefas repetitivas sem escrever o mesmo código várias vezes. Eles são fundamentais para operações como percorrer arrays, processar dados, ou até mesmo em lógicas de jogos.

### Tipos de Loops em JavaScript

Loop **for**: É o cara quando você sabe quantas vezes quer rodar algo. Tipo, "faz isso aqui 10 vezes".

```
for (let i = 0; i < 10; i++) {  
  console.log(i);  
}
```

Aqui, **i** começa em **0**, e o loop continua enquanto **i** for menor que **10**. A cada iteração, **i** é incrementado em **1**.

**Loop while:** Esse aqui é mais quando você tá naquela de "vou fazer isso até algo acontecer". Mas cuidado, se não botar um jeito de parar, vira festa sem hora pra acabar.

```
let i = 0; while (i < 10) {  
  console.log(i); i++;  
}
```

**Loop do...while:** Parecido com o while, mas a diferença é que ele executa pelo menos uma vez, mesmo que a condição já comece falsa.

```
let i = 0;  
do {  
  console.log(i); i++;  
} while (i < 10);
```

**Loop for...of:** Esse é pra quando você tem um array ou algo iterável e quer passar por cada elemento dele.

```
let nomes = ["Ana", "Bia", "Carlos"];  
for (let nome of nomes) {  
  console.log(nome); // Mostra cada nome  
}
```

## Controle de Loops

**break:** Usado para sair de um loop antes que ele atinja sua condição de parada natural.

**continue:** Pula a iteração atual e continua com a próxima.

```
for (let i = 0; i < 10; i++) {  
  if (i === 5) {  
    break; // Sai do loop quando i é 5  
  }  
  if (i % 2 === 0) {  
    continue; // Pula os números pares  
  }  
  
  console.log(i);  
}
```

## Exercícios com loops:

1. Criando um loop: Crie um loop **for** que imprima números de 0 a 10.
2. Loop com condição: Crie um loop **for** que imprima somente números ímpares de 0 a 20.
3. Usando **while**: Crie um loop **while** que imprima números de 0 a 10.
4. Interrompendo um loop: Crie um loop **for** que imprima números de 0 a 100, mas interrompa a execução quando chegar ao número 50.
5. Pulando uma iteração: Crie um loop **for** que imprima números de 0 a 20, mas pule a impressão do número 13.
6. Iterando um Array: Crie um array de nomes e use um loop **for...of** para imprimir cada nome do array.