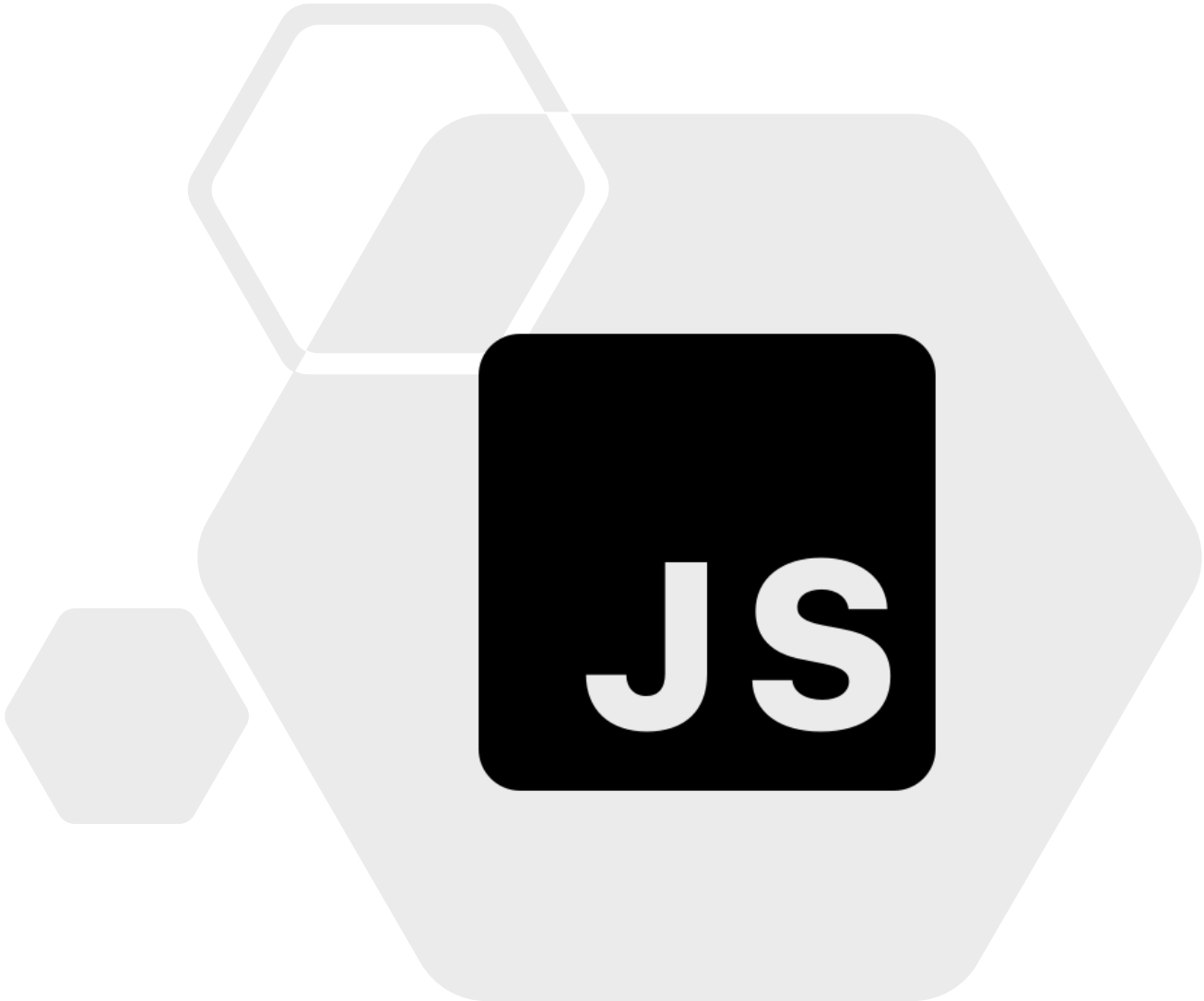
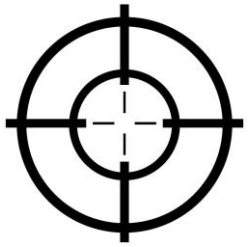
Estruturas de repetição



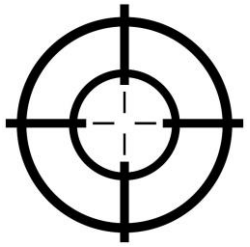
O que são estruturas de repetição?



Estruturas de repetição, ou loops, são usados quando um programa precisa processar repetidamente uma ou mais instruções até que alguma condição seja atendida, momento em que o loop termina.

Muitas tarefas de programação são repetitivas, tendo pouca variação de um item para o outro. O processo de executar a mesma tarefa repetidamente é chamado de iteração.

Quais as vantagens de usar uma estrutura de repetição?

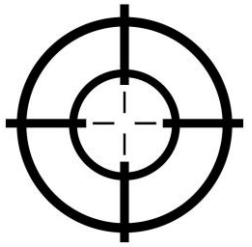


A repetição permite que o programador use variáveis de forma eficiente.

Pode-se estruturar instruções de programação para serem repetidas, desde que condições específicas sejam atendidas.

Importante! Classifica-se em dois tipos as estruturas de repetição, são loops de pré-teste ou pós-teste.

Loops de pós-teste...

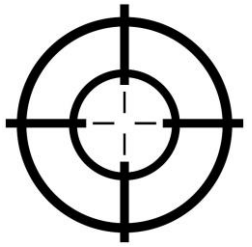


A condição de parada aparece no final do loop

```
let sorteio = (numeros, quantidadeSorteios) => {
    let numerosSorteados = []
    do {
        let indiceSorteado = sortear(numeros.length)

        let numeroSorteado = numeros[indiceSorteado]

        numerosSorteados.push(numeroSorteado)

        removerElemento(numeros, numeroSorteado)
    }
    while (numerosSorteados.length < quantidadeSorteios);

    return numerosSorteados
}

let numeros = [1, 2, 3, 4, 5, 6, 7, 8, 9, 0]
let resultado = sorteio(numeros, 3)

console.log(resultado)
```

```
let sortear = (possibilidades) => {
    let numeroSorteado = Math.floor(Math.random() * possibilidades);
    return numeroSorteado
}
```

Loops de pré-teste...



```
let removerElemento = (numeros, elemento) => {
    for (var indice = 0; indice < numeros.length; indice++) {
        if (numeros[indice] === elemento) {
            numeros.splice(indice, 1);
        }
    }
}
```

A condição de parada aparece no início do loop

```
let sorteio = (numeros, quantidadeSorteios) => {
    let numerosSorteados = []
    do {
        let indiceSorteado = sortear(numeros.length)

        let numeroSorteado = numeros[indiceSorteado]

        numerosSorteados.push(numeroSorteado)

        removerElemento(numeros, numeroSorteado)
    }
    while (numerosSorteados.length < quantidadeSorteios);
```

Tipos de loops “for”

Na linguagem JavaScript existem dois tipos especiais de loops “for”. O chamado de “for in” pode ser usado para iterar sobre as propriedades de um objeto.

```
let empresa = {
    razaoSocial: 'ABC LTDA',
    nomeFantasia: 'Mercado Online'
}

for (valor in empresa) {
    console.log(valor)
}
```

A saída do loop será:

```
razaoSocial
nomeFantasia
```

Tipos de loops “for”

As vezes deseja-se iterar sobre uma estrutura de dados e processar seus valores. Para isso usa-se o “for of”. A instrução “for of” do JavaScript percorre os valores de um objeto iterável.

```
let  estadosSudeste = new Set(['ES','SP','MG','RJ'])

for (let valores of estadosSudeste) {
    console.log(valores)
}
```

Importante

- Existem várias formas para aplicação de estruturas de repetição. Cabe então ao desenvolvedor treinar sua lógica. Isto permitirá o desenvolvimento de soluções com uma codificação coesa e limpa.

JavaScript