

# LOOPS EM JAVASCRIPT

DEFINIDOS ( ou seja quando eu SEI quantas interações irá fazer)

SINTAX:

```
for ( variável_contador ; condição; controle ; incremento ou decremento){  
    "O QUE VAI SER EXECUTADO A CADA INTERAÇÃO"  
}
```

Obs: enquanto for verdadeiro a condição ele incrementa ou decrementa

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

**i++** É A MESMA COISA DE **i=i+i** OU **i+=1**

## Para percorrer uma array

Normal:

Ler carros=[fusca, Monza, fox]

```
for ( let i=0 ; i < carros.length ; i++){
```

```
  console.log(i) // nesse caso o retorno será os índice (0,1,2)
```

```
  console.log([i]) // nesse caso o retorno será os índice (fusca, Monza, fox)}
```

**i++** É A MESMA COISA DE **i=i+i** OU **i+=1**

## FOR IN – Retrona os INDICES do Array

Ler carros=[fusca, Monza, fox]

```
for (i in carros){ console.log(i)} // retorna os indices
```

```
for (i in carros){ console.log(carro[i])} // retorna os elementos
```

Usado para percorrer um array

Observe o sintex normal e a sintex do for in

## FOR OF – Retrona os ELEMENTOS do Array

let cores=[turquoise, White, yellow];

```
for ( e of cores){
```

```
    console.log(e) // retorna os elementos ( turquoise, White, yellow)
```

```
}
```

**i++** É A MESMA COISA DE **i=i+i** OU **i+=1**

# LOOPS INDEFINIDOS

## (WHILE e DO-WHILE)

QUANDO NÃO TENHO CERTEZA DE QUANTOS INTERAÇÕES SERÃO FEITAS

Enquanto a condição for verdadeira que será testada no início executa as funções caso for false segue o programa

No while caso seja false nenhuma execução é feita já no Do-While será executada pelo menos 1 vez

### WHILE

```
let = 0
While(n < 10 ){
    n++
}
```

### DO-WHILE

```
let = 0
Do{
    }While(true ){
    n++
}
```

**i++** É A MESMA COISA DE **i=i+i** OU **i+=1**

# FUNÇÃO MAP

## PERCORRE COLEÇÕES OU ARRAY

Vai percorrer toda a coleção sem interrupção

Retorne três parâmetro os elementos os index e a própria array

Não é obrigado informar todos somente um

```
let COLACAO=[ CASA, CARRO, MESA]
```

```
colacao.map(elemento, index, array )=>{
```

```
return elemento
```

```
}
```

**i++** É A MESMA COISA DE **i=i+i** OU **i+=1**