



## Método reduce

Executa uma função para cada elemento do array, fornecendo uma variável com o valor acumulado dos elementos que já foram processados.

Ao final das execuções ele retornará um único valor.

Obs: Podemos dar um valor inicial para o acumulador, opcionalmente.

```
lista.reduce((acumulador, itemAtual) => {  
    return acumulador + itemAtual  
}, valorInicial)
```

```
lista.reduce((acumulador, itemAtual) => acumulador + itemAtual, valorInicial)
```



## Método reduce

Somar os valores de uma lista ao saldo inicial de 50.000

```
let lista = [1, 2, 3, 4]
let valorInicial = 50000

let soma = lista.reduce((acumulador, itemAtual) => {

  console.log(`Transformando o acumulador em ${acumulador} + ${itemAtual}`)
  return acumulador + itemAtual

}, valorInicial)

console.log(soma) // Retorna 50000 + 1 + 2 + 3 + 4 = 50010
```

# Método reduce



Pegar os números pares de um array e adicionar o dobro deles em um novo Array:

```
let lista = [1, 2, 3, 4, 5, 6, 7]
let valorInicial = []

let paresDobrados = lista.reduce((acumulador, item) => {
  console.log(`Executando o item ${item} com acumulador ${acumulador}`)
  if (item % 2 === 0) {
    console.log(`Adicionando item novo: ${item * 2}`)
    acumulador.push(item * 2)
  }
  // Sempre retornar algo para ser o acumulador
  return acumulador
}, valorInicial)
```



## Método reduce

Verificar o maior número

```
let numeros = [1, 2, 4, 3, 7, 8, 6, 4, 5]

let verificaMaior = numeros.reduce((valorMaximo, item) => {
  if (item > valorMaximo) {
    return item;
  } else {
    return valorMaximo;
  }
}, 0);

console.log(verificaMaior);
```