

```
//SPREAD / REST:
//O operador "..." recebe 2 nomes:
//    spread - traduzido literalmente para "espalhamento" - onde ele espalha um campo de iteráveis ou uma string sem que
//    seja necessário referenciar literalmente cada um dos elementos índice do array ou string.
//    rest - traduzido por "agrupar" - que é quando os valores de um iterável são juntados a outro iterável.

//Ele recebe esse nome justamente por que é justamente isso que ele faz, ele primeiro espalha os valores de um iterável dentro de
outro iterável, e no final ele agrupa os valores espalhados com os valores pré-existent em um novo iterável.

//spread/rest serve para trabalharmos com objetos, arrays e strings.

//Vejam alguns exemplos:

//APLICANDO O SPREAD/REST A UMA FUNÇÃO QUE PRECISA DE 3 ARGUMENTOS:
const soma = (x, y, z) => { //Note que a função soma precisa de 3 argumentos...
    return x + y + z
}
const num = [1, 2, 3, 4] //E temos um array atribuído a uma variável chamada "num" que contém 4 elementos...
console.log('1', soma(...num)) //Veja aqui o spread em ação, ele foi escrito antes do nome da variável que contém 4 elementos, como
a função soma só precisa de 3 elementos, o spread mostra somente os elementos necessários para a função, fazendo com que ela pegue os
3 primeiros elementos e ignore os restantes...

//APLICANDO SPREAD/REST A STRINGS:
//Quando usamos strings, o spread trata cada letra da string como se fosse um iterável de um array
const palavra = 'Olá tudo bem?' //Essa string tem 12 caracteres...
const mostraSo3 = (a, b, c) => { //Mas a função mostraSo3 vai pegar só os 3 primeiros caracteres...
    return a + b + c
}
console.log('\n2)', mostraSo3(...palavra)) //Passando nossa strings com o operador spread nossa string será transformada num array
onde cada caractere será um valor iterável...
```

```
//USANDO SPREAD/REST PARA JUNTAR VALORES DE 2 OBJECTS:
const funcionarios = {nome: 'Maria', salario: 2000}
const cloneFuncionarios = {ativo: true, ...funcionarios} //Perceba que desejamos criar um novo objeto que contém os valores de outro,
bastou apenas usar o spread/rest que capturou os valores do outro objeto, os espalhou no objeto novo e os juntou num objeto só...
console.log('\n3)', cloneFuncionarios)


//USANDO SPREAD/REST PARA AGRUPAR ARRAYS:
const grupoA = ['João', 'Pedro', 'Glória']
const grupoB = ['Maria', ...grupoA, 'Rafaela'] //Perceba que também conseguimos unir arrays, e ainda usamos o spread/rest entre 2
valores, e isso é respeitado quando imprimimos os valores...
console.log('\n4)', grupoB)
```

RESULTADO NO CONSOLE...

```
[Running] node "c:\Users\Almoxarifado\Documents\javascript\arquivos_das_aulas\090-Spread.js"
1) 6

2) Olá

3) { ativo: true, nome: 'Maria', salario: 2000 }

4) [ 'Maria', 'João', 'Pedro', 'Glória', 'Rafaela' ]

[Done] exited with code=0 in 0.115 seconds
```