

//ARRAYS:

//No Javascript o Array é uma estrutura totalmente dinâmica e heterogênea, ou seja, diferente de outras linguagens de programação, ele não é uma estrutura estática onde precisamos declarar o número de índices de um array. Ele pode aumentar ou diminuir em tempo de execução e ainda pode aceitar qualquer tipo de dado.

//Assim como os demais tipos primitivos em Javascript, Array também é uma função, e os objetos gerados a partir de um array são objetos de verdade. Por causa dessa característica podemos gerar arrays até por através de instancialização.

//No javascript não precisamos definir o tipo de dado de uma array, basta colocar uma variável ou constante declarada e atribuir valores sobre ela. Diferente de outras linguagens também, não precisamos definir um tamanho físico para o array no momento da declaração dele. Além disso, como o Javascript é uma linguagem fracamente tipada, podemos adicionar vários tipos de dados dentro de um array (Embora essa não seja uma boa prática).

//DECLARANDO UM ARRAY DE FORMA LITERAL:

```
const VALORES = [7.7, 8.9, 6.3, 9.2];
```

```
console.log(`1) ${VALORES[0]} ${VALORES[3]}`); //Perceba que os índices são referenciados sem problema algum...
```

//LENDO ÍNDICES NÃO DECLARADOS:

```
console.log(`\n2) ${VALORES[4]}`); //Perceba que o índice 4 não existe, mas o Javascript não apresenta erro, em vez disso, ele trata o valor como "undefined"...
```

//ATRIBUINDO VALORES A UM ARRAY:

VALORES[4] = 10.2; //Esse método é mais utilizado quando desejamos adicionar elementos a um determinado valor índice ou desejamos substituir um elemento já existente. Quando desejamos adicionar elementos ao final e um array é muito comum utilizarmos o método push()...

```
console.log(`\n3) ${VALORES[4]}`);
```

```
console.log(`\n4) ${VALORES}`); //Veja que foi incluso justamente na 4ª posição do Array...
```

//ATRIBUINDO VALOR A UMA POSSIÇÃO MUITO Á FRENTE:

VALORES[10] = 12.6; //Perceba que atribuímos um valor à 10ª posição de uma array, mas ele não tem outros valores da 4ª posição em diante...

```
console.log(`\n5) ${VALORES}`); //Ainda assim o Javascript não apresenta erro, em vez disso, ele gerará índices nulos até que cheguem ao índice de número 10...
```

```
//USANDO A FUNÇÃO LENGTH PARA LER A QUANTIDADE DE ELEMENTOS DE UM ARRAY...
console.log(`\n6) ${VALORES.length}`);

//ADICIONANDO DADOS POR ATRAVÉS DA FUNÇÃO PUSH():
VALORES.push({id: 3}, false, null, 'teste'); //Veja que usamos o push para adicionar valores ao final de uma array, e não apenas
isso, adicionamos valores de diferentes tipos de dado. Esse é o método mais utilizado quando desejamos adicionar elementos diretamente
ao final de um array...
console.log(`\n7) ${VALORES}`);

//RETIRANDO VALORES DO ARRAY E VISUALIZANDO O VALOR RETIRADO:
console.log(`\n8) ${VALORES.pop()}`); //Para isso usamos a função pop(), essa função retira o último valor do array e ainda imprime o
valor que foi retirado...
console.log(`\n9) ${VALORES}`); //Note que o valor foi realmente retirado...

//RETIRANDO VALORES DE UM ÍNDICE:
delete VALORES[0]; //Usando a propriedade delete podemos remover o valor de um índice, porém o índice ainda continuará lá...
delete VALORES[1];
delete VALORES[2];
delete VALORES[3];
delete VALORES[4];
console.log(`\n10) ${VALORES}`);

//TODO ARRAY EM JAVASCRIPT É DO TIPO OBJECT:
console.log(`\n11) ${typeof VALORES}`);

//ARRAY É UMA FUNÇÃO INTERNA DO JAVASCRIPT ASSIM COMO OS DEMAIS TIPOS PRIMITIVOS:
console.log(`\n12)`, typeof Array)

//PODEMOS INSTANCIAR NOVOS ARRAYS Á PARTIR DA FUNÇÃO ARRAY:
```

```

const arrayInstance = new Array(1, 2, 3, 4) //Acabamos de criar um array por instancialização e ainda por cima declaramos elementos literalmente. Embora essa seja uma forma de criar array, essa é uma forma não recomendada, o mais recomendado é o tipo de criação literal de array...
console.log('\n13)', arrayInstance)
console.log('\n14)', typeof arrayInstance) //Apesar de ser um array, ele não passa de um object, pois foi criado por instancialização...

//INDEXANDO ARRAY PELA NUMERAÇÃO Á PARTIR DO 1 EM VEZ DO 0:
const meses = ['janeiro', 'fevereiro', 'março', 'abril', 'maio', 'junho', 'julho', 'agosto', 'setembro', 'outubro', 'novembro', 'dezembro']

numero1 = 4 //Veja que os números estão atribuídos a uma variável...
numero2 = 5
numero3 = 6

console.log('\n15)', meses[--numero1]) //Em seguida usamos o operador de decremento pré-fixado, ele faz com que o elemento passado seja decrementado antes de ser operado, isso faz parecer que estamos usando o número original dos meses...
console.log(meses[--numero2])
console.log(meses[--numero3])

```

RESULTADO NO CONSOLE...

```

[Running] node "c:\Users\User\Documents\JAVASCRIPT\ARQUIVOS_DAS_AULAS\029-manipulandoArrays.js"
1) 7.7 9.2

2) undefined

3) 10.2

4) 7.7,8.9,6.3,9.2,10.2

5) 7.7,8.9,6.3,9.2,10.2,,,,,12.6

6) 11

```

7) 7.7,8.9,6.3,9.2,10.2,,,,,12.6,[object Object],false,,teste

8) teste

9) 7.7,8.9,6.3,9.2,10.2,,,,,12.6,[object Object],false,

10) ,,,,,,,12.6,[object Object],false,

11) object

12) function

13) [ 1, 2, 3, 4 ]

14) object

15) abril

maio

junho

[Done] exited with code=0 in 0.145 seconds