/\*\* Arrays: []

Uma lista de dados;

Uma coleção ordenada de elementos;

Armazenar vários valores em uma única variável.

Exemplos:

const nomes = ['Pedro', 'Tiago', 'Maria', 'Lucas'];

const notas = [10, 5, 8, 6, 4];

const misturado = [false, 123, 'Teste', undefined]

const listaDeLista = [['Teste', 123, true], [false, '423', 123], 1231];

OBS: uma string pode ser semelhante em alguns aspectos com um array;

Manipulação de Arrays:

OBS: os índices do array inicialmente começa com zero 0

┌─────────┬─────────┐

│ (index) │ Values │

├─────────┼─────────┤

│ 0 │ 'Pedro' │

│ 1 │ 'Tiago' │

│ 2 │ 'Maria' │

│ 3 │ 'Lucas' │

└─────────┴─────────┘

Exemplos:

// Pegar alguns dados do array:

const nomes = ['Pedro', 'Tiago', 'Maria', 'Lucas'];

nomes[0]; // 'Pedro'

notas[3]; // 6

nomes.length; // 4

nomes.indexOf('Tiago'); // Retorna o index de um elemento do array, se o elemento não for encontrado retorna -1

nomes.includes('Tiago'); // Verifica se um elemento existe na lista e retorna um valor booleano.

nomes.find((nome) => nome[0] === 'M'); // Retorna o primeiro elemento de acordo com a condição da busca, por exemplo, estou pegando o primeiro elemento da lista de nomes que começa com a letra 'M'

// Alterar valor

nomes[0] = 'João'; // Altera o valor do index 0 do array nomes

nomes.splice(inicio, quantidade, valor); // permite substituir x quantidades de elementos em qualquer parte da lista

// Adicionar valor:

nomes[4] = 'João'; // Adiciona um elemento no final da lista

nomes.push('João'); // adiciona um elemento no final da lista

nomes.unshift('João'); // adiciona um elemento no início da lista

// Remove valor:

nomes.pop(); // Remove o último elemento da lista

nomes.shift(); // Remove o primeiro elemento da lista

nomes.splice(inicio, quantidade); // permite remover x quantidades de elementos em qualquer parte da lista

delete nomes[2]; // remover um elemento específico, porém não reorganiza a lista, deixando o lugar do elemento removido como undefined

// Percorrer uma lista:

Para percorrermos umas lista existem várias formas:

Usando estrutura de loops:

const frutas = ['Maça', 'Pera', 'Banana'];

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

console.log(frutas[i]);

}

const frutas = ['Maça', 'Pera', 'Banana'];

for (let index in frutas) {

console.log(frutas[index]);

}

const frutas = ['Maça', 'Pera', 'Banana'];

for (let fruta of frutas) {

console.log(fruta);

}

Usando métodos:

frutas.forEach((fruta, index) => { // função anônima utilizando arrow function

console.log(fruta, index);

});

// Copiar e Combinar listas:

Tome cuidado no caso de você tentar copiar uma lista:

// Errado:

const nomesCopia = nomes; // Parece que está fazendo uma copia da lista 'nomes' para uma outra lista, porém isso é meio que fazer uma ligação, logo alterar uma lista altera a outra

nomesCopia[0] = 'João';

console.log(nomesCopia); // [ 'João', 'Tiago', 'Maria', 'Lucas' ]

console.log(nomes); // [ 'João', 'Tiago', 'Maria', 'Lucas' ]

// Certo:

Para copia uma lista da maneira correta, você precisa usar o operador spread '...',

é um recurso do JavaScript introduzido no ECMAScript 2015, permite expandir elementos de um array ou propriedades de um objeto.

const nomesCopia = [...nomes];

nomesCopia[0] = 'João';

console.log(nomesCopia); // [ 'João', 'Tiago', 'Maria', 'Lucas' ]

console.log(nomes); // [ 'Pedro', 'Tiago', 'Maria', 'Lucas' ]

Outra característica do spread é que podemos juntas listas:

const frutas = ['Maça', 'Pera', 'Banana'];

const verduras = ['Alface', 'Tomate', 'Cebola'];

const mercadorias = [...frutas, ...verduras];

console.log(mercadorias); // [ 'Maça', 'Pera', 'Banana', 'Alface', 'Tomate', 'Cebola' ]

Outra forma de combinar listas é usando o método 'concat', ele retorna uma nova lista com a combinação de todas as lista que você passar nesse método

const mercadorias = frutas.concat(verduras);

console.log(mercadorias); // [ 'Maça', 'Pera', 'Banana', 'Alface', 'Tomate', 'Cebola' ]

// Outros métodos de lista:

sort, reverse, map, filter, reduce