

Aula 1 – Introdução javascript

1. Escrever uma função que encontre o elemento mais frequente de um Array, assim como a quantidade de vezes que ele aparece (retorne um objeto seguindo o padrão dos exemplos abaixo). Lembre-se de testar as entradas inválidas:

Entrada: [3, 'a', 'a', 'a', 2, 3, 'a', 3, 'a', 2, 4, 9, 3]

Saída: {resp: 'a', freq: 5}

Entrada: [1, 2, 3, 'y', 'y', 4, 5, 'y', 6, 1, 2, 3, 4, 5, 6, 'a', 'a']

Saída: {resp: 'y', freq: 3}

Entrada: []

Saída: {resp: undefined, freq: 0}

Entrada: null ou undefined

Saída : undefined

Dicas: use o **for .. in** do javascript para iterar sobre os parâmetros de um object:

```
var ob = {'a': 1, 'b': 2}

for (var item in ob) {
  console.log(item + ' => ' + ob[item])
}

// saída:
// a => 1
// b => 2
```

Para checar se um object contém uma propriedade:

```
var ob = {'a': 1, 'b': 2}

console.log(ob.hasOwnProperty('a'))
console.log(ob.hasOwnProperty('b'))
console.log(ob.hasOwnProperty('c'))

// saída:
// false
// true
// false
```

2. Escrever uma função que remova as duplicatas em um Array (sem alterar a ordem em que aparecem pela primeira vez):

Entrada: [3, 'a', 'a', 'a', 2, 3, 'a', 3, 'a', 2, 4, 9, 3]

Saída: [3, 'a', 2, 4, 9]

Entrada: [1, 2, 3, 'y', 'y', 4, 5, 'y', 6, 1, 2, 3, 4, 5, 6, 'a', 'a']

Saída: [1, 2, 3, 'y', 4, 5, 6, 'a']

3. Escrever uma função que retorne um Array contendo a intersecção de dois Arrays (sem duplicatas):

Entrada: [3, 'a', 'a', 'a', 2, 3, 'a', 3, 'a', 2, 4, 9, 3]
[1, 2, 3, 'y', 'y', 4, 5, 'y', 6, 1, 2, 3, 4, 5, 6, 'a', 'a'].
Saída: [3, 'a', 2, 4]

4. Escrever uma função que receba uma string e retorne um Array contendo todas suas substrings (sem duplicatas e ordenada lexicograficamente).

Entrada: arara

Saída: ['a', 'ar', 'ara', 'arar', 'arara', 'r', 'ra', 'rar', 'rara']

Respostas:

1.

```
function maisFrequente(array) {  
    if (array == undefined || array == null)  
        return undefined  
    var freqs = {}  
    for (var i = 0; i < array.length; i++) {  
        var item = array[i];  
        if (freqs.hasOwnProperty(item))  
            freqs[item] ++;  
        else  
            freqs[item] = 1;  
    };  
    var freq = 0;  
    var resp = undefined;  
    for (var item in freqs) {  
        if (freqs[item] > freq) {  
            freq = freqs[item];  
            resp = item;  
        }  
    }  
    return {resp: resp, freq: freq}  
}
```

2.

```
function removeDuplicatas (array) {  
    var ob = {}, resp = [];  
    for (var i = 0; i < array.length; i++) {  
        item = array[i];  
        if (!ob.hasOwnProperty(item)) {  
            ob[item] = true;  
            resp.push(item);  
        }  
    }  
    return resp;  
}
```

3.

```
function intersecao(array1, array2) {  
    array1 = removeDuplicatas(array1);  
    var resp = [];  
    for (var i = 0; i < array1.length; i++) {  
        var item = array1[i];  
        if (array2.indexOf(item) !== -1)  
            resp.push(item);  
    }  
    return resp;  
}
```

4.

```
function substrings (string) {  
    var ob = {};  
    for (var i = 0; i < string.length; i++) {  
        for (var j = i+1; j < string.length + 1; j++) {  
            ob[string.substring(i, j)] = true;  
        }  
    }  
    return Object.keys(ob).sort();  
}
```