

Descrição da atividade: fazer os exercícios para praticar. Não precisa entregar as respostas.

Objetivo:

- Arrays unidimensional e bidimensional;
- Funções
- Estruturas de decisão e repetição;
- String.

Exercício 1: A função quebrar recebe uma string e retorna um array onde cada elemento é um caractere da string recebida como parâmetro.

Dica: use o método split da string (https://www.w3schools.com/jsref/jsref_split.asp).

```
function quebrar(nome) {  
    return _____;  
}
```

Exemplo de resultado:

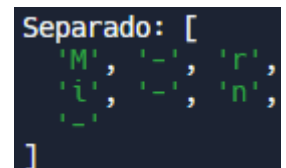
A screenshot of a terminal window showing the output of the function: "Separado: ['P', 'e', 'd', 'r', 'o']". The text is displayed in a monospaced font with syntax highlighting.

```
let r = quebrar("Pedro");  
console.log("Separado:", r);
```

Exercício 2: Alterar o corpo da função quebrar para substituir as letras a por traço.

```
function quebrar(nome) {  
    const v = _____;  
    for( let i = 0; i < v.length; i++ ) {  
        if( v[_____] == "a" ) {  
            v[_____] = _____;  
        }  
    }  
    return _____;  
}
```

Exemplo de resultado:

A screenshot of a terminal window showing the output of the function: "Separado: ['M', '-', 'r', 'i', '-', 'n', '-']". The text is displayed in a monospaced font with syntax highlighting.

```
let r = quebrar("Mariana");  
console.log("Separado:", r);
```

Exercício 3: Alterar o corpo da função quebrar do Exercício 2 para substituir qualquer vogal por traço.

Dica: use o operador ou || para comparar cada vogal.

Exercício 4: Alterar o corpo da função quebrar do Exercício 2 para retornar um array com a letras envolvidas pelas marcações HTML <div> </div>.

```
function quebrar(nome) {
```

Exemplo de resultado:

```
let v = _____;
for( let i = 0; i < v.length; i++ ){
    v[_____] = _____;
}
return _____;
}
```

```
let r = quebrar("Mariana");
console.log("Separado:", r);
```

```
Separado: [
  '<div>M</div>',
  '<div>a</div>',
  '<div>r</div>',
  '<div>i</div>',
  '<div>a</div>',
  '<div>n</div>',
  '<div>a</div>'
]
```

Exercício 5: A função criar recebe dois inteiros e retorna um array contendo os números no intervalo formado pelos parâmetros.

```
function criar(a,b){
    let v = []; //cria um array vazio
    for( let i = 0; _____ <= _____; i++, a++ ){
        v[_____] = _____;
    }
    return _____;
}
```

Exemplo de resultado:

```
Array: [ 5, 6, 7, 8, 9, 10 ]
```

```
let r = criar(5,10);
console.log("Array:", r);
```

Exercício 6: Alterar a função criar do Exercício 5 para os elementos serem strings.

Dica: somar cada número com a string vazia "".

Exemplo de resultado:

```
Array: [ '5', '6', '7', '8', '9', '10' ]
```

Exercício 7: A função somar recebe dois arrays de números inteiros e retorna um array com a soma de cada um dos elementos.

```
function somar(a,b){
    let v = [] //cria um array vazio
    for( let i = _____; _____ < _____; i++ ){
        v[_____] = _____;
    }
    return _____;
}
```

Exemplo de resultado:

```
Soma: [ 8, 13, 12, 6, 12, 11 ]
```

```
let a = [5,8,3,4,9,7];  
let b = [3,5,9,2,3,4];  
let r = somar(a,b);  
console.log("Soma:", r);
```

Exercício 8: Alterar a função somar do Exercício 7 para retornar o maior valor em cada par de valores entre os arrays recebidos como parâmetro.

Exemplo de resultado:

Maior: [5, 8, 9, 4, 9, 7]

Exercício 9: Alterar a função somar do Exercício 7 para retornar os arrays compondo um array bidimensional.

Dica: para criar um array basta colocar os elementos dentro dos colchetes.

Exemplo de resultado:

Juntar: [[5, 8, 3, 4, 9, 7], [3, 5, 9, 2, 3, 4]]

Exercício 10: Completar o código da função somar para retornar o somatório dos elementos do array bidimensional recebido como parâmetro.

```
function somar(m) {  
    let soma = _____;  
    for( let i = 0; i < m.length; i++ ){  
        for( let j = 0; j < m[i].length; j++ ){  
            _____;  
        }  
    }  
    return _____;  
}
```

```
let m = [  
    [5,8,3,4,9,7],  
    [3,5,9,2,3,4]  
]  
  
let r = somar(m);  
console.log("Somatório:", r);
```

Exemplo de resultado:

Somatório: 62