

Observação: Para fazer os exercícios você pode utilizar o Visual Studio Code ou a ferramenta online para JavaScript <https://repl.it/>.

Exercício 1 – Na linguagem JavaScript as funções podem ser escritas na notação de Arrow Function. Reescrever a função isPar usando a notação de Arrow Function. Restrições: você não poderá usar **parênteses**, **chaves** e o termo **return**.

```
function isPar(nro){  
  return nro%2 == 0? true : false  
}  
console.log( isPar(2) )  
console.log( isPar(3) )
```

Exercício 2 – O código a seguir verifica se cada elemento do array é par ou ímpar. Para fazer essa codificação utilizamos uma estrutura de repetição. Porém, o objeto Arrays da linguagem JavaScript possui a implementação do método forEach que é usado para iterar sobre os elementos do objeto Arrays. Reescrever o código a seguir substituindo a estrutura de repetição pelo método forEach.

```
function isPar(nro){  
  return nro%2 == 0? true : false  
}  
  
let nros = [2,3,4,5]  
for(let i = 0; i < nros.length; i++){  
  console.log(isPar(nros[i]))  
}
```

Exercício 3 – O código a seguir verifica se cada elemento do array é par ou ímpar e coloca o resultado no array **copia**. Para fazer essa codificação primeiramente copiamos todos os elementos do array **nros** para o array **copia** usando o operador spread (https://developer.mozilla.org/pt-BR/docs/Web/JavaScript/Reference/Operators/Spread_operator). Na sequência utilizamos uma estrutura de repetição para percorrer o array **nros** e alterar o array **copia**. Porém, o objeto Arrays possui a implementação do método map que é usado para iterar sobre os elementos do objeto Arrays e retornar um **novo** array. Reescrever o código a seguir substituindo a estrutura de repetição pelo método map. Observe que você terá problemas com a instrução **const copia**.

```
function isPar(nro){  
  return nro%2 == 0? true : false  
}  
  
const nros = [2,3,4,5]  
const copia = [...nros] //Spread Operator  
console.log(copia)  
for(let i = 0; i < nros.length; i++){  
  copia[i] = isPar(nros[i])  
}
```

```
console.log(copia)
```

Exercício 4 – O código a seguir soma os elementos do array e coloca na variável soma. O objeto Arrays possui a implementação do método reduce que é usado para iterar sobre os elementos do objeto Arrays e retornar um único valor. Reescrever o código a seguir substituindo a estrutura de repetição pelo método reduce.

O método reduce recebe como parâmetro uma função callback que será invocada para cada elemento do array. Reescrever o código 2 vezes: uma usando função anônima e outra usando Arrow Function.

```
const nros = [2,3,4,5]
let soma = 0
for(let i = 0; i < nros.length; i++){
    soma += nros[i]
}
console.log(soma)
```

Exercício 5 – Uma função pode ser criada e chamada anonimamente. O código a seguir cria uma função e chama ela. Esse recurso só deve ser utilizado quando a função só será chamada 1 vez. Reescrever o código a seguir usando Arrow Function.

```
(function(){
    console.log('Bom dia');
})();
```

Exercício 6 – Usando o recurso de criar e chamar uma função anonimamente. Codificar uma função que recebe os números 2 e 3, imprime no console a soma deles. Observação: escrever o código 2 vezes: uma usando função anônima e outra usando Arrow Function.