

DISCIPLINA: PROJETO FRONT-END WEB JAVASCRIPT
PROF. WILTON DE PAULA FILHO

PFJS - LISTA DE EXERCÍCIOS

ARRAY ITERATION METHODS

Utilize o conceito de funções de alta ordem da classe Array (forEach, map, reduce, filter, some e every) e o conceito de Arrow Function para resolver cada um dos problemas a seguir

1. Proponha uma solução utilizando JavaScript para imprimir no corpo de uma página web os valores de determinado vetor. Os valores do vetor devem ser impressos na mesma linha e separados por um traço. Sugestão: crie pelo menos três vetores com diversos valores numéricos e utilize-os para testar a função. Para este exercício irei apresentar a solução comentada.

Solução esperada:

```
let vetor1 = [10,4,8,7,1,20];  
let vetor2 = [0,41,13,7,1,20];  
let vetor3 = [2,3,4,1,8];  
  
let imprimir = (vet) => (vet.forEach(valor=>document.write(valor+" - ")));  
  
imprimir(vetor1);  
document.write("<br>");  
imprimir(vetor2);  
document.write("<br>");  
imprimir(vetor3);
```

Passo a passo para se chegar até a solução esperada:

Até você se acostumar, proponha a solução do problema utilizando o conceito de função declarativa. Veja a seguir como ficará a solução utilizando essa abordagem.

```
let vetor1 = [10,4,8,7,1,20];  
function imprimir(vetor) {  
    for (let i=0; i<vetor.length; i++)  
        document.write(vetor[i]+" -> ")  
}  
imprimir(vetor1);
```

Ótimo, para se chegar até a solução esperada vejamos o que fazer. Primeiro, eliminarei a estrutura de repetição “for” utilizando o método **forEach** da classe Array. Veja a seguir como ficará a solução:

```
1. let vetor1 = [10,4,8,7,1,20];  
2. function imprimir(vetor) {  
3.     vetor.forEach(mostrar);  
4. }  
5. function mostrar(valor) {  
6.     document.write(valor+" -> ")  
7. }  
8. imprimir(vetor1);
```

Para compreender a solução acima, utilizando o **forEach**, você terá que se lembrar da sintaxe da função **forEach**, ok? Caso não se lembre ou ainda esteja inseguro em utilizá-la, sugiro a leitura desses materiais (https://www.w3schools.com/jsref/jsref_foreach.asp, https://developer.mozilla.org/pt-BR/docs/Web/JavaScript/Reference/Global_Objects/Array/forEach). Lembre-se que o **forEach** precisa receber como parâmetro uma função. Foi por isso que na solução acima eu criei a função “mostrar()”. Veja no último quadro acima que eu passei ela como parâmetro para a função **forEach** (Linha 2). Bom, e de onde vem o parâmetro “valor” da função “mostrar()” (Linha 5)?. Quando uma função é passada como parâmetro para **forEach**, no momento da criação da função (Linha 5), você terá que inserir pelo menos um parâmetro para a função, que corresponderá ao valor corrente do vetor. No nosso exemplo, o parâmetro “valor” da função “mostrar()” será o elemento a ser iterado pela função. Lembre-se, se o vetor tiver 4 valores a função “mostrar()” será executada quatro vezes.

Vamos continuar? O próximo passo consistirá em utilizar a notação de **Arrow Function** para reduzir a função “mostrar()” (Linhas 5 à 7). Primeiro, farei o seguinte irei substituir a chamada da função “mostrar” (Linha 3) pela função “mostrar()” inteira (Linhas 5 à 7). Veja a seguir como ficará:

```
1. let vetor1 = [10,4,8,7,1,20];
2. function imprimir(vetor) {
3.     vetor.forEach(function mostrar(valor) {
4.         document.write(valor+" -> ")
5.     });
6. }
7. imprimir(vetor1);
```

Agora, em relação a esse último quadro, irei remover o termo “function” e o nome da função mostrar (Linha 3). Lembre-se que ao remover terei que colocar “=>” depois do parâmetro da função (veja que eu não fiz mais nada além disso). Veja como ficará:

```
let vetor1 = [10,4,8,7,1,20];
function imprimir(vetor) {
    vetor.forEach((valor)=>{document.write(valor+" -> ")});
}
imprimir(vetor1);
```

Continuando. Vamos utilizar o conceito de função literal? Isto é, vamos atribuir à função “imprimir()” a uma variável (let i = ...) ? É o que eu irei fazer, veja abaixo:

```
let vetor1 = [10,4,8,7,1,20];
let i = function imprimir(vetor) {
    vetor.forEach((valor)=>{document.write(valor+" -> ")});
}
i(vetor1);
```

Muito bem, vamos continuar, pois está ficando bonito demaaaaaais. Para finalizar, vamos utilizar o conceito de função anônima para remover o termo “function” e o nome da função imprimir,

ok? Você está lembrado que ao removê-los você terá que inserir o sinal “=>” depois do parâmetro? Ah bom, achei que tinha esquecido. Veja como ficará:

```
let vetor1 = [10,4,8,7,1,20];  
let i = (vetor) => {vetor.forEach((valor)=>{document.write(valor+" -> ")});  
}  
imprimir(vetor1);
```

Agora, volte para a primeira página deste documento e veja se a solução contida nesse último quadro está igual a solução esperada. Não? O que mudou? Tire suas próprias conclusões (Sussuro: “veja que algumas chaves foram removidas”). Só acho estranho uma função se chamar “i”, foi por isso que eu criei a minha variável (solução esperada) com o nome “imprimir”. Espero que você tenha entendido.

Bom, em relação aos passos anteriormente descritos, a questão é: “você terá que fazer isso toda vez?” Claro que não, com o tempo você vai se acostumar e vai sair escrevendo código utilizando apenas o conceito de **Arrow Function**, sem perceber (“Bom, eu espero!”). Bom, abaixo irei disponibilizar uma série de exercícios para você se exercitar. Bom trabalho!

- 2.** Proponha uma solução utilizando JavaScript para exibir no corpo de uma página web a soma dos valores armazenados em um vetor. A soma deverá ser exibida no corpo da página. Sugestão: crie pelo menos três vetores com diversos valores numéricos e utilize-os para testar a função.

```
let vetor1 = [10,4,8,7,1,20];  
let vetor2 = [0,41,13,7,1,20];  
let vetor3 = [2,3,4,1,8];  
let soma = (vetor) => (vetor.reduce((total,valor)=>total+valor));  
document.write(soma(vetor1)+"<br>"); // 55  
document.write(soma(vetor2)+"<br>"); // 82  
document.write(soma(vetor3)+"<br>"); // 18
```

3. Proponha uma solução utilizando JavaScript para exibir no corpo de uma página web a média aritmética simples dos valores armazenado em um vetor. A soma deverá ser exibida no corpo da página. Sugestão: crie pelo menos três vetores com diversos valores numéricos e utilize-os para testar a função.

```
let vetor1 = [10,4,8,7,1,20];  
let vetor2 = [0,41,13,7,1,20];  
let vetor3 = [2,3,4,1,8];  
let media = (vetor) => (vetor.reduce((total,valor)=>total+valor)/vetor.length).toFixed(2);  
document.write(media(vetor1)+"<br>"); // 8.33  
document.write(media(vetor2)+"<br>"); // 13.67  
document.write(media (vetor3)+"<br>"); // 3.60
```

OBS: Neste exercício, a função “toFixed(2)” foi utilizada para exibir apenas duas casas decimais. Sugiro a você remover essa função para ver o que acontecerá.

4. Proponha uma solução utilizando JavaScript para exibir no corpo de uma página web o maior valor armazenado em um vetor. Sugestão: crie pelo menos três vetores com diversos valores numéricos e utilize-os para testar a função.

```
let vetor1 = [-10,-4,-8,-78,1,20];  
let vetor2 = [0,41,13,7,1,20];  
let vetor3 = [2,3,4,1,8];  
let maior = (vetor) => (vetor.reduce((acumulador,valor)=>(valor>acumulador)?valor:acumulador));  
console.log(maior(vetor1)); // 20  
console.log(maior(vetor2)); // 41  
console.log(maior(vetor3)); // 8
```

5. Proponha uma solução utilizando JavaScript para exibir no corpo de uma página web os números ímpares armazenados em um vetor. Sugestão: crie pelo menos três vetores com diversos valores numéricos e utilize-os para testar a função.

```
let vetor1 = [10,4,8,7,1,20];
let vetor2 = [0,41,13,7,1,20];
let vetor3 = [2,3,4,1,8];
let impares = (vetor) => (vetor.filter(valor=>valor%2==1));
//let impares = (vetor) => (vetor.filter(valor=>valor%2));
document.write(impares(vetor1)+"<br>"); // 7,1
document.write(impares(vetor2)+"<br>"); // 41,13,7,1
document.write(impares(vetor3)+"<br>"); // 3,1
```

- 6.** Proponha uma solução utilizando JavaScript para informar (*true* ou *false*) no corpo de uma página web se os números armazenados em determinado vetor são todos pares. Sugestão: crie pelo menos três vetores com diversos valores numéricos e utilize-os para testar a função.

```
let vetor1 = [10,4,8,7,1,20];
let vetor2 = [0,41,13,7,1,20];
let vetor3 = [21,3,43,1,87];
let temPar = (vetor) => (vetor.some(valor=>valor%2==0));
//let temPar = (vetor) => (vetor.some(valor=>!valor%2));
document.write(temPar(vetor1)+"<br>"); // true
document.write(temPar(vetor2)+"<br>"); // true
document.write(temPar(vetor3)+"<br>"); // false
```

- 7.** Proponha uma solução utilizando JavaScript para informar (*true* ou *false*) no corpo de uma página web se TODAS as idades de pessoas armazenadas em determinado vetor são maiores de idade (≥ 18). Sugestão: crie pelo menos três vetores com diversos valores numéricos e utilize-os para testar a função.

```
let vetor1 = [10,4,8,7,1,20];
let vetor2 = [0,41,13,7,1,20];
let vetor3 = [21,33,43,19,87];
let todosMaiorIdade = (vetor) => (vetor.every(valor=>valor>=18));
document.write(todosMaiorIdade(vetor1)+"<br>"); // false
document.write(todosMaiorIdade(vetor2)+"<br>"); // false
document.write(todosMaiorIdade(vetor3)+"<br>"); // true
```

8. Proponha uma solução utilizando JavaScript para informar (*true* ou *false*) no corpo de uma página web se TODAS as idades de pessoas armazenadas em determinado vetor são maiores ou iguais a um valor (corte) informado pelo usuário. Sugestão: crie pelo menos três vetores com diversos valores numéricos e utilize-os para testar a função.

```
let vetor1 = [10,4,8,7,1,20];
let corte1 = 0; //parseInt(prompt("Digite um valor (corte)"));
let vetor2 = [0,41,13,7,1,20];
let corte2 = 10; //parseInt(prompt("Digite um valor (corte)"));
let vetor3 = [21,33,43,19,87];
let corte3 = 19; //parseInt(prompt("Digite um valor (corte)"));
let todosMaiorCorte = (vetor,corte) => (vetor.every(valor=>valor>=corte));
document.write(todosMaiorCorte(vetor1,corte1)+"<br>"); // true
document.write(todosMaiorCorte(vetor2,corte2)+"<br>"); // false
document.write(todosMaiorCorte(vetor3,corte3)+"<br>"); // true
```

9. Proponha uma solução utilizando JavaScript para informar no corpo de uma página web o número de elementos de um vetor que são maiores do que um valor de corte informado pelo usuário. Sugestão: crie pelo menos três vetores com diversos valores numéricos e utilize-os para testar a função.

```
let vetor1 = [10,4,8,7,1,20];
let corte1 = 5; //parseInt(prompt("Digite um valor (corte)"));
let vetor2 = [0,41,13,7,1,20];
let corte2 = 10; //parseInt(prompt("Digite um valor (corte)"));
let vetor3 = [21,33,43,19,87];
let corte3 = 50; //parseInt(prompt("Digite um valor (corte)"));
let contAcimaCorte = (vetor,corte) => (vetor.filter(valor=>valor>corte)).length;
document.write(contAcimaCorte(vetor1,corte1)+"<br>"); // 4
document.write(contAcimaCorte(vetor2,corte2)+"<br>"); // 3
document.write(contAcimaCorte(vetor3,corte3)+"<br>"); // 1
```

- 10.** Proponha uma solução utilizando JavaScript para, a partir da leitura de um vetor contendo valores numéricos e de um intervalo [limiteInferior, limiteSuperior], informar no corpo de uma página web todos os valores do vetor pertencentes ao intervalo. Sugestão: crie pelo menos três vetores com diversos valores numéricos e utilize-os para testar a função.

```
let vetor1 = [10,4,8,7,1,20];
let limInf = 3; //parseInt(prompt("Digite o valor do limite inferior:"));
let limSup = 10; //parseInt(prompt("Digite o valor do limite superior:"));
let valoresIntervalo = (vetor,limInf,limiteSup) => (vetor.filter(valor=>valor>=limInf&&valor<=limSup));
document.write(valoresIntervalo(vetor1,limInf,limSup)); // true
```

- 11.** Proponha uma solução utilizando JavaScript para somar os valores ímpares de determinado vetor. Sugestão: crie pelo menos três vetores com diversos valores numéricos e utilize-os para testar a função.

```
let vetor1 = [10,4,8,7,1,20];
let vetor2 = [0,41,13,7,1,20];
let vetor3 = [21,33,43,19,87];
let somaImpares = (vetor) => (vetor.filter(valor=>valor%2==1).reduce((acumulador,valor)=>acumulador+valor));
//let somaImpares = (vetor) => (vetor.filter(valor=>valor%2).reduce((acumulador,valor)=>acumulador+valor));
document.write(somaImpares(vetor1)+"<br>"); // 8
document.write(somaImpares(vetor2)+"<br>"); // 62
document.write(somaImpares(vetor3)+"<br>"); // 203
```

Neste exercício, a primeira iniciativa que eu tiver que fazer foi localizar os valores ímpares do vetor. Isso foi realizado a partir do seguinte código:

```
(vetor) => (vetor.filter(valor=>valor%2==1))
```


Se eu tivesse parado por aí, eu já teria como retorno um vetor contendo apenas os números ímpares. Bom, como ainda precisamos achar a soma dos ímpares, eu apliquei a função de alta ordem “reduce” ao resultado dessa função, que é um vetor. Por isso, tive que fazer o seguinte:

```
(vetor.filter(valor=>valor%2==1)).map....
```

Aí, basta lembrar quais são os parâmetros da função “map” (acumulador, valor, etc). Neste exemplo, utilizei apenas o “acumulador” e o “valor”, como parâmetros.

- 12.** Proponha uma solução utilizando JavaScript para exibir apenas o primeiro e o último nome de cada uma das várias pessoas armazenadas em um vetor.

```
let nomes = ["Wilton de Paula Filho", "Maria Jose da Silva", "ana maria"];
let imprimeNomesInicioFim = (vetor) => (
  vetor.map(valor => {
    let nome = valor.toLowerCase().split(' ');
    return nome[0].concat(" "+nome[nome.length-1]);
  }));
document.write(imprimeNomesInicioFim(nomes)); //wilton filho,maria silva,an
a maria
```

- 13.** Proponha uma solução utilizando JavaScript para informar se todas as palavras armazenadas em um vetor são palíndromos. Se forem, retorne *true* ou *false*, caso contrário.

```
let palavras1 = ["ama", "ana", "nawan", "arara"];
let palavras2 = ["ama", "ana", "nawan", "maria"];
let saoPalindromos = (vetor) => (vetor.every(valor=>valor==valor.split('').
reverse().join('')));
console.log(saoPalindromos(palavras1)); // true
console.log(saoPalindromos(palavras2)); // false
```

- 14.** Solicitar N preços de produtos de uma loja, bem como os nomes de cada um deles. Em seguida, ajustar os preços dos produtos conforme as regras abaixo:

- Preços entre R\$ 2,00 e R\$ 10,00 reais terão os preços acrescidos em 10%;

- Preços entre R\$ 11,00 e R\$ 100,00 reais terão os preços acrescidos em 15%;
- Preços acima de R\$ 100,00 reais terão os preços acrescidos em 20%;

A página deverá exibir em uma tabela os nomes de todos os produtos lidos, bem como os preços originais e os preços finais (após ajuste de preço).

```
let produtosOriginais = [  
  {produto:"bola",preco:5.00},  
  {produto:"camiseta",preco:45.50},  
  {produto:"tv",preco:800.00}  
];  
  
let ajustarPrecos = (listaProdutos) => (listaProdutos.map(valor=>(valor.pre  
co<=10)?valor.preco*1.1:(valor.preco<=100)?valor.preco*1.15:valor.preco*1.2  
)));  
  
produtosAjustados = ajustarPrecos(produtosOriginais);  
  
let imprimirPrecos = (produtosOriginais,produtosAjustados) => {  
  document.write("<table style='border:1px solid black'>");  
  document.write("<th>Produto</th><th>Preço original</th><th>Preço fina  
l (ajustado)</th>");  
  produtosOriginais.forEach((valor,indice)=>document.write("<tr><td>"+v  
alor.produto+"</td><td>"+valor.preco.toFixed(2)+"</td><td>"+produtosA  
justados[indice].toFixed(2)+"</td></tr>"));  
  document.write("</table>");  
}  
  
imprimirPrecos(produtosOriginais, produtosAjustados);
```