

STEP Javascript - Homework 26

Arrow functions e High Order Functions - introdução

No código que segue ao final do doc, desenvolvemos alguns exemplos-teste de como definir e usar arrow functions. Também falamos sobre a função map(), que é uma high order function.

As high order functions são funções de objeto e/ou listas (a função map() é uma função built-in de arrays) que geralmente recebem, como argumento de entrada, outra functions (e opcionalmente podem retornar uma outra function, o que costumamos chamar de *programação funcional*).

Exercício: Baseando-se no código apresentado, crie arrow functions que atuarão sobre a coleção de dados que segue:

```
var notas = [  
  {nome: 'João', nota: 8, sexo: 'M'},  
  {nome: 'Sara', nota: 12, sexo: 'F'},  
  {nome: 'Francisco', nota: 16, sexo: 'M'},  
  {nome: 'Paulo', nota: 2, sexo: 'M'},  
  {nome: 'Everton', nota: 4, sexo: 'M'},  
  {nome: 'Paula', nota: 18, sexo: 'F'},  
  {nome: 'Demétrio', nota: 5, sexo: 'M'},  
  {nome: 'Márcio', nota: 13, sexo: 'F'},  
  {nome: 'Carlos', nota: 15, sexo: 'F'},  
  {nome: 'Janaína', nota: 9, sexo: 'F'}  
]
```

As funções serão:

A média da classe

A média dos alunos de sexo masculino

A média dos alunos de sexo feminino

A maior nota entre os alunos de sexo masculino

A maior nota entre os alunos de sexo feminino

(obviamente, você também deve escrever código que usa essas funções, mostrando o resultado na página)

DICA: além do código apresentado ao final do doc, dê uma olhada no seguinte exemplo que seria aplicado a esta coleção de alunos:

```
let isMasculino = estudante => estudante.sexo === 'M'  
let getAlunosMasculino = notas => ( notas.filter(isMasculino) )
```

// IMPORTANTE: esta notação, não vista no código na sala, é a condensação completa
// de uma arrow function. A função filter() é uma high order function (com uma
// funcionalidade parecida com a da função map()) que usa a arrow function
// isMasculino() como argumento de entrada. Além disso, a função na segunda linha
// ela está inteiramente envolvida entre parentesis. Desenvolveremos esta forma na
próxima aula

NOTA: para entender melhor este exemplo, estude a função filter() no w3school e no nosso livro de referência. *Note também que a função acima pode fazer parte da resolução deste exercício* (porque ela sugere que você pode escrever pequenas arrow functions, e associá-las umas com as outras, decompondo seu código em várias pequenas funcionalidades (no exemplo acima, isMasculino foi projetada para ser usada por geAlunosMasculino..

=====
Entregue A PASTA (diretório) com os arquivos .html, .js e outros se houverem. Não entregue .txt, .docx, etc. Faça um zip da pasta e upload deste zip
=====

=== Codigo sobre Arrow Funcions e High Order Functions de sala de aula ===

Index.html

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <meta http-equiv="X-UA-Compatible" content="ie=edge">
  <script defer src="codigo.js"></script>
  <title>Document</title>
</head>
<body>
  <div id="tela"></div>

</body>
</html>
```

codigo.js

```
/* ARROW FUNCTION
    (arg1, arg2, .... argn) => { corpo da função; }
*/

// exemplo
//const adicionar = (n1, n2) => n1 + n2;
// ou
const adicionar = (n1, n2) => { return n1 + n2 }

// seria equivalente a
function somar (n1, n2) {
    return n1 + n2;
}

// exemplo comum de chamada de function tradicional
var adicionar2 = somar (3, 2);

// forma simples da chamda da arrow function
var x = adicionar(2,3)

// forma mais usada com listas
const dobro = n => n * 2;
//y = dobro(2)    // posso chamar assim

// HIGH ORDER FUNCTIONS NORMALMENTE SÃO UTILIZADAS COM ARROW FUNCTION
// as mais comuns: map, filter, sort, reduce, forEach
// usando em conjunção com high order function map
[1, 2, 3, 4].map(dobro)

const xxxx = element => {
    if (element = 'M') return true;
}

['M', 'M', 'F', 'M'].forEach(xxx);
```

```
// FORMA GERAL
// lista.funcaoDeMaisAltaOrdem(arrow function)

// usando em uma lista dentro de um objeto
a = {nome: 'dudu', idade: 14, preferencias : [1,2,3,5]}

var result = a.preferencias.map(dobro)
```