# *Os códigos devem estar comentados e em inglês*

# 1º exercício (DESAFIO)

Para testar seus conhecimentos com classes, crie uma classe com nome "Admin", essa classe deve extender uma segunda classe chamada "Usuario".

A classe usuário deve receber dois parâmetros no método construtor, e-mail e senha, e anotá-los em propriedades da classe. A classe "Admin" por sua vez não recebe parâmetros mas deve repassar os parâmetros de e-mail e senha à classe pai e marcar uma propriedade "admin" como true na classe.

Agora com suas classes formatadas, adicione um método na classe Usuario chamado isAdmin que retorna se o usuário é administrador ou não baseado na propriedade admin ser true ou não.

**const User1 = new Usuario('email@teste.com', 'senha123'); const Adm1 = new Admin('email@teste.com', 'senha123');**

**console.log(User1.isAdmin()) // false console.log(Adm1.isAdmin()) // true**

# 2º exercício

A partir do seguinte vetor e utilizando os métodos de array (map, reduce, filter e find):

**const usuarios = [**

**{name: "Felipe", age: 28, company: "AMF"},**

**{name: "Becker", age: 18, company: "AMF"},**

**{name: "Nunes", age: 38, company: "Other"}**

**];**

## Utilizando o map

Crie uma variável que contenha todas idades dos usuários: [28, 18, 38]

* 1. **Utilizando o filter**

Crie uma variáveis que tenha apenas os usuários que trabalham na AMF e com mais de 15 anos

* 1. **Utilizando o find**

Crie uma variável que procura por um usuário que trabalhe na empresa Other

## Unindo operações

Multiplique a idade de todos usuários por dois e depois realize um filtro nos usuários que possuem no máximo 50 anos

# 3º exercício

Converta as funções nos seguintes trechos de código em Arrow Functions:

**// 3.1**

**const arr = [1, 2, 3, 4, 5]; arr.map(function(item) {**

**return item + 10;**

**});**

**// 3.2**

**// Dica: Utilize uma constante pra function const user = {name: Felipe, age: 28};**

**function showAge(user) {return user.age;**

**}**

**showAge(user);**

**// 3.3**

**// Dica: Utilize uma constante pra function**

**const name = "Felipe"; const age = 28;**

**function showUser(name = 'Felipe', age = 28) { return { name, age };**

**}**

**showUser (name, age); showUser (name);**

**// 3.4**

**const promise = function() {**

**return new Promise(function(resolve, reject) {**

**return resolve();**

**})**

**}**

# 4º exercício

## Desestruturação simples

A partir do seguinte objeto:

**const company = { name: 'AMF', adress: {**

**city: 'Restinga Sêca', state: 'RS',**

**}**

**};**

Utilize a desestruturação para transformar as propriedades nome, cidade e estado em variáveis, no fim deve ser possível fazer o seguinte:

**console.log(name); // AMF console.log(city); // Restinga Sêca console.log(state); // RS**

## Desestruturação em parâmetros

Na seguinte função:

**function showInfo(user) {**

**return `${ user.name} has ${ user.age} years.`;**

**}**

**showInfo ({ name: 'Felipe', age: 28 })**

Utilize a desestruturação nos parâmetros da função para buscar o nome e idade do usuário separadamente e a função poder retornar apenas:

**return `${name} has ${age} years.`;**

# 5º Exercício

Utilizando o operador de rest/spread (...) realize as seguintes operações:

## Rest

A partir do array: const arr = [1, 2, 3, 4, 5, 6], defina uma variável x que recebe a primeira posição do vetor e outra variável y que recebe todo restante dos dados.

**console.log(x); // 1**

**console.log(y); // [2, 3, 4, 5, 6]**

Crie uma função que recebe inúmeros parâmetros e retorna a soma de todos eles:

**// function soma...**

**console.log(sum(1, 2, 3, 4, 5, 6)); // 21**

**console.log(sum (1, 2)); // 3**

## Spread

A partir do objeto e utilizando o operador spread:

**const user = { name: 'Felipe', age: 28, adress: {**

**city: 'Restinga Sêca', state: 'RS',**

**country: 'Brazil',**

**}**

**};**

Crie uma variável user2 que contenha todos os dados do usuário porém com nome Becker. Crie uma variável user3 que contenha todos os dados do usuário porém com cidade Santa Maria.

# 6º Exercício

Converta o seguinte trecho de código utilizando Template Literals:

**const user = 'Felipe'; const age = 28;**

**console.log(The ' + user + ' has ' + age + ' years);**

# 7º exercício

Utilize a sintaxe curta de objetos (Object Short Syntax) no seguinte objeto:

**const name = 'Felipe'; const age = 28;**

**const user = { name: name, age: age,**

**city: 'Restinga Sêca',**

**};**