Exercícios: Módulo 01

1º exercício

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 = [
  { nome: 'Diego', idade: 23, empresa: 'Rocketseat' },
  { nome: 'Gabriel', idade: 15, empresa: 'Rocketseat' },
  { nome: 'Lucas', idade: 30, empresa: 'Facebook' },
];
```

2.1 Utilizando o `map`

Crie uma variável que contenha todas idades dos usuários: `[23, 15, 30]`

2.2 Utilizando o `filter`

Crie uma variáveis que tenha apenas os usuários que trabalham na Rocketseat e com mais de 18 anos: `[{ nome: 'Diego', idade: 23, empresa: 'Rocketseat' }]`

2.3 Utilizando o `find`

Crie uma variável que procura por um usuário que trabalhe na empresa Google: `undefined`

2.4 Unindo operações

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

```
// Resultado:
[
  { nome: 'Diego', idade: 46, empresa: 'Rocketseat' },
  { nome: 'Gabriel', idade: 30, empresa: 'Rocketseat' },
]
```

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 usuario = { nome: 'Diego', idade: 23 };

function mostraIdade(usuario) {
  return usuario.idade;
}

mostraIdade(usuario);
```

```
// 3.3
// Dica: Utilize uma constante pra function

const nome = "Diego";
const idade = 23;

function mostraUsuario(nome = 'Diego', idade = 18) {
  return { nome, idade };
}

mostraUsuario(nome, idade);
mostraUsuario(nome);
```

```
// 3.4

const promise = function() {
  return new Promise(function(resolve, reject) {
```

```
    return resolve();
  })
}
```

4º exercício

4.1 Desestruturação simples

A partir do seguinte objeto:

```
const empresa = {
  nome: 'Rocketseat',
  endereco: {
    cidade: 'Rio do Sul',
    estado: 'SC',
  }
};
```

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

```
console.log(nome); // Rocketseat
console.log(cidade); // Rio do Sul
console.log(estado); // SC
```

4.2 Desestruturação em parâmetros

Na seguinte função:

```
function mostraInfo(usuario) {
  return `${usuario.nome} tem ${usuario.idade} anos.`;
}

mostraInfo({ nome: 'Diego', idade: 23 })
```

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

```
return `${nome} tem ${idade} anos.`;
```

5º Exercício

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

5.1 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(soma(1, 2, 3, 4, 5, 6)); // 21
console.log(soma(1, 2)); // 3
```

5.2 Spread

A partir do objeto e utilizando o operador spread:

```
const usuario = {
  nome: 'Diego',
  idade: 23,
  endereco: {
    cidade: 'Rio do Sul',
    uf: 'SC',
    pais: 'Brasil',
  }
};
```

Crie uma variável `usuario2` que contenha todos os dados do usuário porém com nome `Gabriel`.

Crie uma variável `usuario3` que contenha todos os dados do usuário porém com cidade `Lontras`.

6º Exercício

Converta o seguinte trecho de código utilizando Template Literals:

```
const usuario = 'Diego';
const idade = 23;

console.log('O usuário ' + usuario + ' possui ' + idade + ' anos');
```

7º exercício

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

```
const nome = 'Diego';
const idade = 23;

const usuario = {
  nome: nome,
  idade: idade,
  cidade: 'Rio do Sul',
};
```