



Geek University

Evolua seu lado geek!

www.geekuniversity.com.br

Exercícios JavaScript

1) Crie uma função que receba como parâmetro de entrada o objeto abaixo e imprima no console a string de saída, conforme:



```
// objeto de entrada da função
let endereco = {
  rua: "Rua da paz",
  numero: 42,
  bairro: "Cachoeira",
  cidade: "Campo Grande",
  uf: "MS"
};

// saída
O usuário mora em Campo Grande/MS, no bairro Cachoeira, na Rua da Paz, número 42.
```

Exercícios JavaScript

2) Crie uma função que receba dois inteiros como parâmetro de entrada e exiba todos os números pares entre estes dois números (inclusive), conforme esqueleto abaixo:



```
function pares_no_intervalo(num1, num2){  
    //implementação  
}  
  
// usando  
pares_no_intervalo(1, 42);
```

Exercícios JavaScript

3) Crie uma função que receba um array como entrada e verifique se neste array contém a habilidade “JavaScript”. Imprima a informação conforme cada caso, de acordo com o esqueleto abaixo:

```
function sabe_javascript(habilidades){  
    // implementação  
}  
  
// Exemplos  
  
//1  
habilidades1 = ["Python", "React", "JavaScript"];  
  
//2  
habilidades2 = ["Python", "Bash", "Linux"];  
  
// Execução  
sabe_javascript(habilidades1); // Sabe JavaScript  
  
sabe_javascript(habilidades2); // Não sabe JavaScript
```

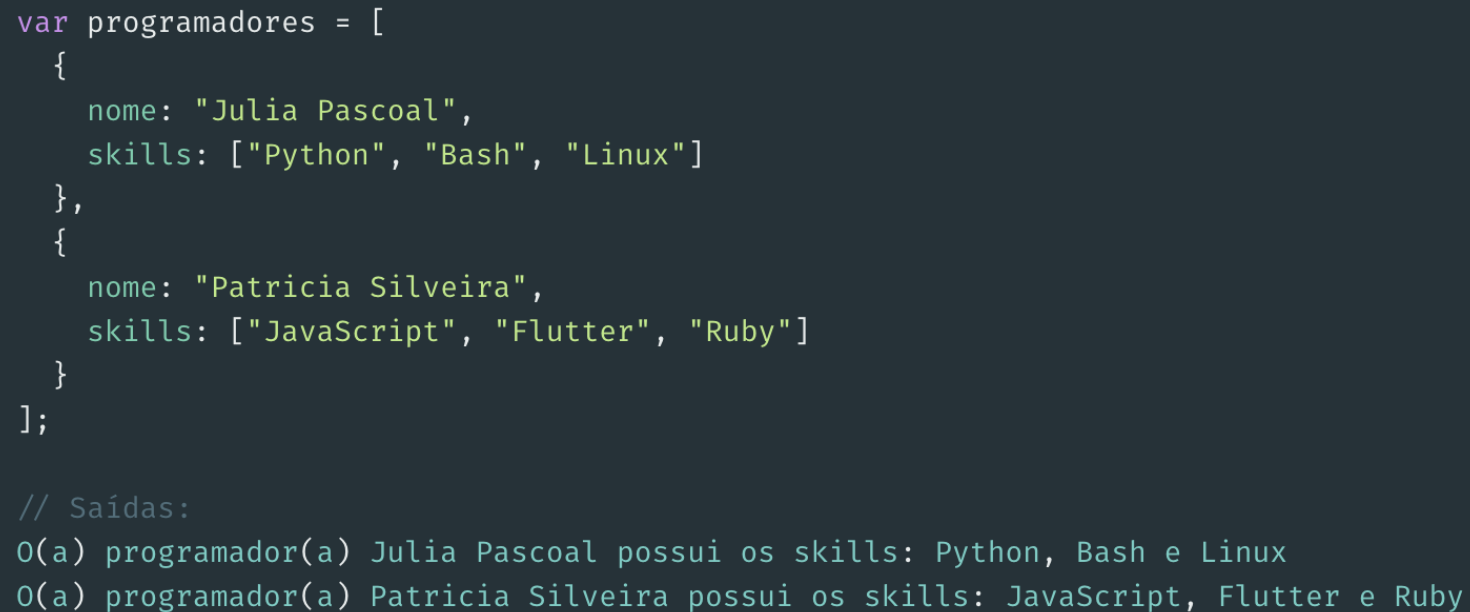
Exercícios JavaScript

4) Crie uma função que receba um inteiro como parâmetro de entrada sendo os anos de experiência de um programador e imprima qual é o tipo de programador de acordo com esta experiência em anos.

```
function nivel_de_programador(experiencia_em_anos){  
    // implementação  
}  
  
// Execução  
let experiencia = 4;  
nivel_de_programador(experiencia);  
  
/*  
    Entre 0 e 1 ano: Iniciante  
    Entre 1 e 3 anos: Intermediário  
    Entre 3 e 6 anos: Avançado  
    Igual ou maior 7 anos: Jedi Fuck Master  
*/
```

Exercícios JavaScript

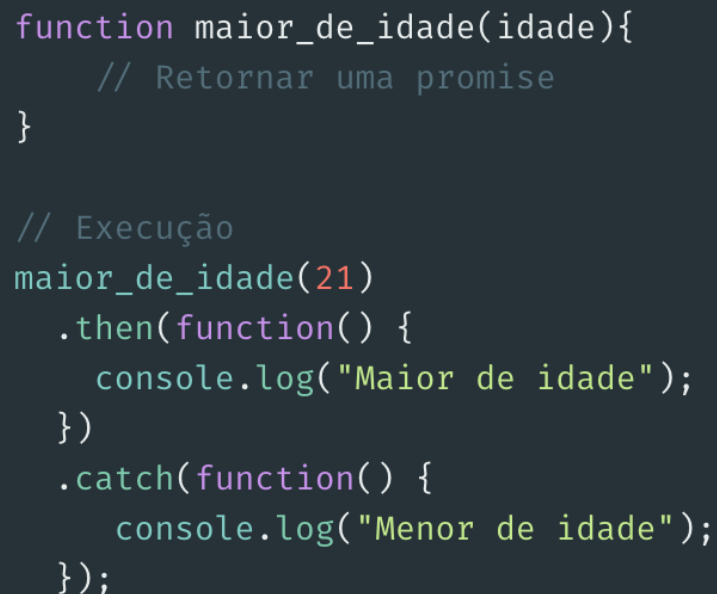
5) Crie uma função que receba uma lista de objetos e apresente as saídas conforme exemplo abaixo:



```
var programadores = [  
  {  
    nome: "Julia Pascoal",  
    skills: ["Python", "Bash", "Linux"]  
  },  
  {  
    nome: "Patricia Silveira",  
    skills: ["JavaScript", "Flutter", "Ruby"]  
  }  
];  
  
// Saídas:  
O(a) programador(a) Julia Pascoal possui os skills: Python, Bash e Linux  
O(a) programador(a) Patricia Silveira possui os skills: JavaScript, Flutter e Ruby
```

Exercícios JavaScript

6) Crie uma função que receba uma idade como parâmetro e retorne uma promise que após 3 segundos irá mostrar se o usuário é maior ou menor de idade, conforme:



```
function maior_de_idade(idade){  
    // Retornar uma promise  
}  
  
// Execução  
maior_de_idade(21)  
    .then(function() {  
        console.log("Maior de idade");  
    })  
    .catch(function() {  
        console.log("Menor de idade");  
    });
```

Exercícios JavaScript

7) Crie duas classes sendo Usuario, com atributos email e senha, e uma classe Admin que herda de usuário e tem um atributo boolean admin com valor true.

No construtor de Usuario, receba os parâmetros para setar nos atributos. No construtor de Admin, receba os parâmetros para repassar à classe pai.

Crie um método na classe Usuario que retorna um boolean informando se determinado usuário é admin ou não, conforme exemplo abaixo:



```
let usuario1 = new Usuario("joao@gmail.com", "senhasegura");  
let usuario2 = new Admin("maria@gmail.com", "senhasupersegura");  
  
console.log(usuario1.ehAdmin()); //false  
console.log(usuario2.ehAdmin()); //true
```


Exercícios JavaScript

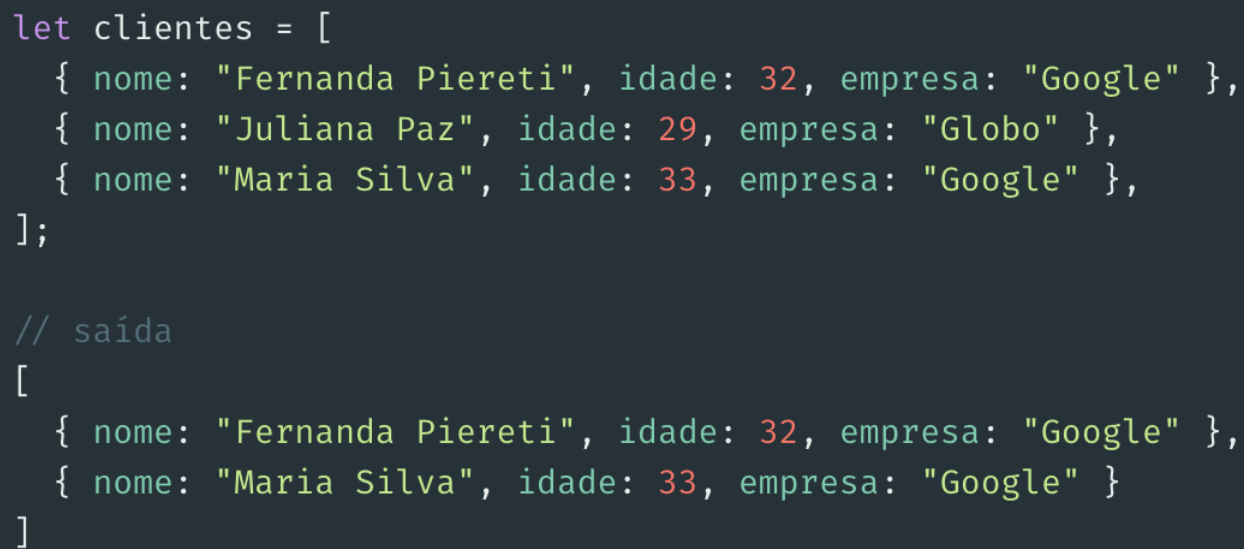
8) Dado o vetor abaixo, utilizando map, crie e apresente uma variável que contenha uma lista de todas as idades dos clientes presentes no vetor, conforme imagem:



```
let clientes = [  
  { nome: "Fernanda Piereti", idade: 32, empresa: "Google" },  
  { nome: "Juliana Paz", idade: 29, empresa: "Globo" },  
  { nome: "Maria Silva", idade: 33, empresa: "Google" },  
];  
  
// saída  
[32, 29, 33]
```

Exercícios JavaScript

9) Dado o vetor abaixo, utilizando filter, crie e apresente uma variável que contenha uma lista de todos os clientes que trabalham na empresa Google presentes no vetor, conforme imagem:



```
let clientes = [
  { nome: "Fernanda Piereti", idade: 32, empresa: "Google" },
  { nome: "Juliana Paz", idade: 29, empresa: "Globo" },
  { nome: "Maria Silva", idade: 33, empresa: "Google" },
];

// saída
[
  { nome: "Fernanda Piereti", idade: 32, empresa: "Google" },
  { nome: "Maria Silva", idade: 33, empresa: "Google" }
]
```

Exercícios JavaScript

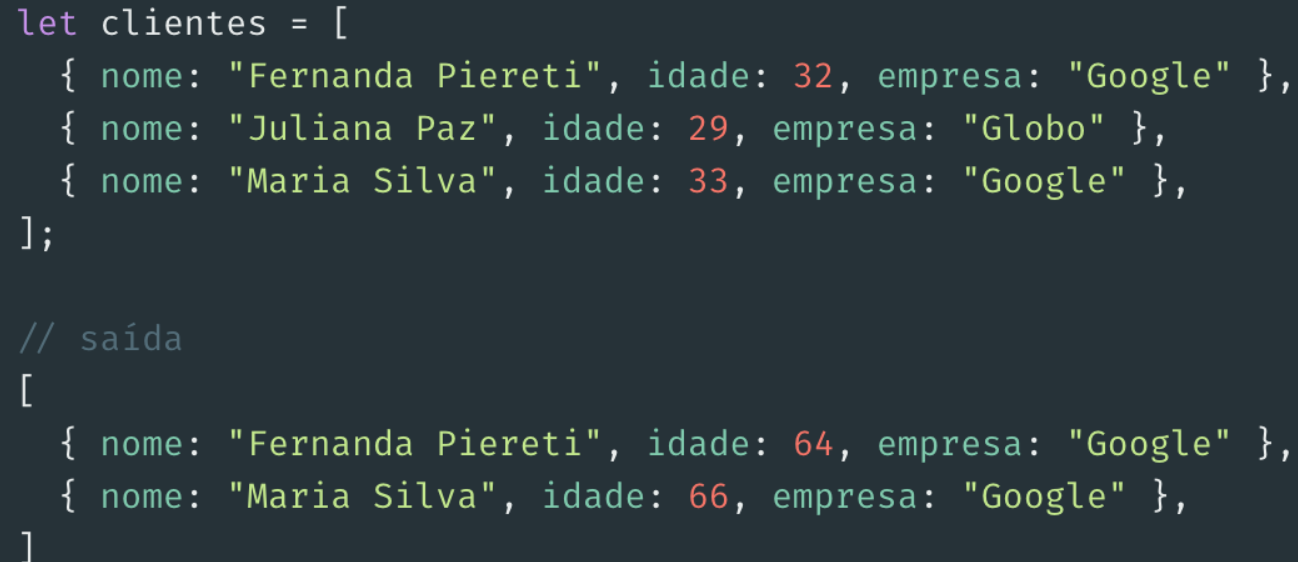
10) Dado o vetor abaixo, utilizando find, crie e apresente uma variável que contenha o usuário que trabalhe na empresa Udemy presentes no vetor, conforme imagem:



```
let clientes = [  
  { nome: "Fernanda Piereti", idade: 32, empresa: "Google" },  
  { nome: "Juliana Paz", idade: 29, empresa: "Globo" },  
  { nome: "Maria Silva", idade: 33, empresa: "Google" },  
];  
  
// saída  
undefined
```

Exercícios JavaScript

11) Dado o vetor abaixo, multiplique por 2 a idade de todos os clientes e então realize um filtro apresentando apenas os clientes que possuam mais de 60 anos:



```
let clientes = [  
  { nome: "Fernanda Piereti", idade: 32, empresa: "Google" },  
  { nome: "Juliana Paz", idade: 29, empresa: "Globo" },  
  { nome: "Maria Silva", idade: 33, empresa: "Google" },  
];  
  
// saída  
[  
  { nome: "Fernanda Piereti", idade: 64, empresa: "Google" },  
  { nome: "Maria Silva", idade: 66, empresa: "Google" },  
]
```

Exercícios JavaScript

12) Converta a função abaixo em uma Arrow Function:



```
let lista = [2, 4, 6, 8, 10];  
  
lista.map(function(valor) {  
    return valor + 3;  
});
```

Exercícios JavaScript

13) Converta a função abaixo em uma Arrow Function:



```
let usuario = { nome: "Fernanda Silva", idade: 45 };

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

mostrarIdade(usuario);
```

Exercícios JavaScript

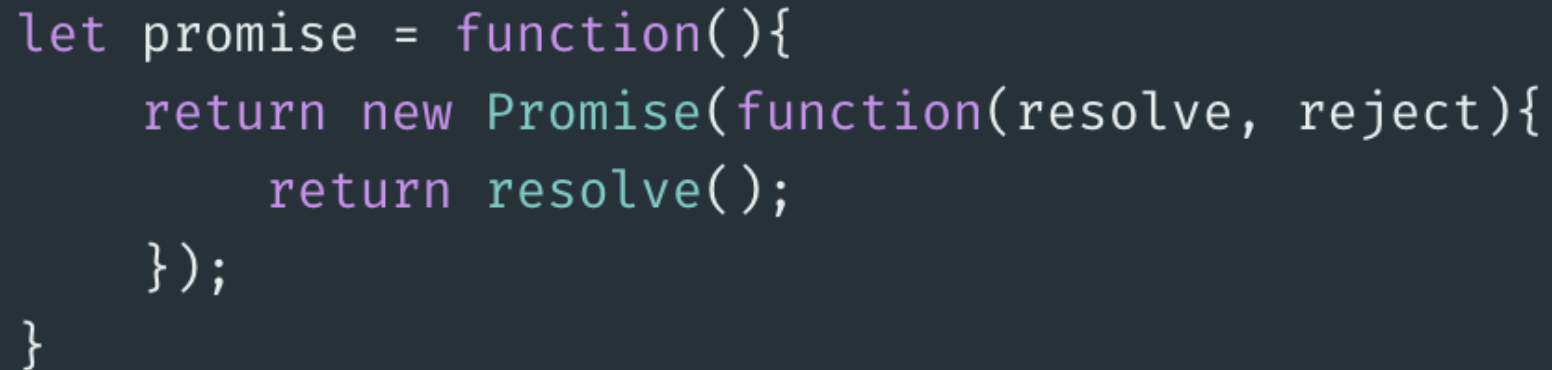
14) Converta a função abaixo em uma Arrow Function:



```
let nome = "Fernanda Lima";  
let idade = 34;  
  
function mostrarUsuario(nome = "Fernanda Lima", idade = 19){  
    return { nome, idade };  
}  
  
mostrarUsuario(nome, idade);  
mostrarUsuario(nome);
```

Exercícios JavaScript

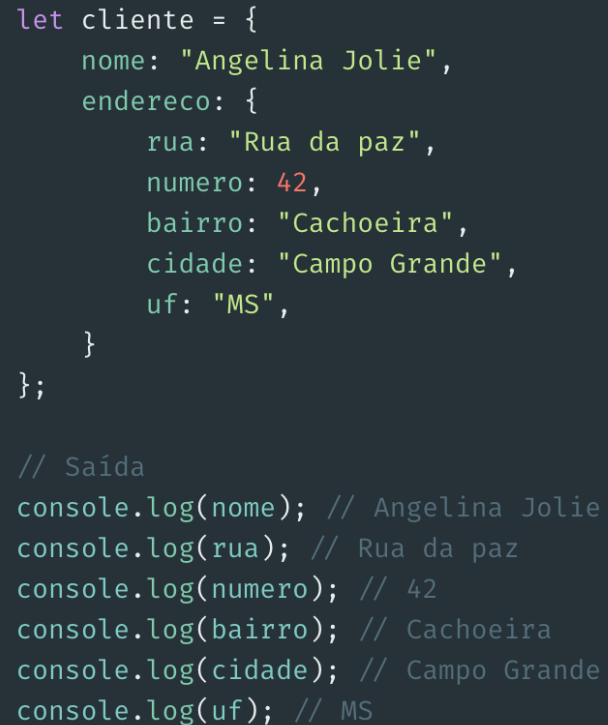
15) Converta a função abaixo em uma Arrow Function:



```
let promise = function(){  
    return new Promise(function(resolve, reject){  
        return resolve();  
    });  
}
```


Exercícios JavaScript


16) Utilize o conceito de destructuring para transformar em variáveis a partir do objeto abaixo:



```
let cliente = {  
  nome: "Angelina Jolie",  
  endereco: {  
    rua: "Rua da paz",  
    numero: 42,  
    bairro: "Cachoeira",  
    cidade: "Campo Grande",  
    uf: "MS",  
  }  
};  
  
// Saída  
console.log(nome); // Angelina Jolie  
console.log(rua); // Rua da paz  
console.log(numero); // 42  
console.log(bairro); // Cachoeira  
console.log(cidade); // Campo Grande  
console.log(uf); // MS
```

Exercícios JavaScript

17) Altera a função abaixo usando o conceito de destructuring para extrair os parâmetros necessários e apresentar o return já com as variáveis extraídas:



```
function mostrarInfo(usuario){  
    return `${usuario.nome} tem ${usuario.idade} anos.`;  
}  
  
mostrarInfo({ nome: "Angelina Jolie", idade: 46 });  
  
// O retorno da função deve ser  
return `${nome} tem ${idade} anos.`;
```

Exercícios JavaScript

18) Crie uma função que receba um array como parâmetro e usando os conceitos do operador Rest/Spread apresente uma variável 'x' com o primeiro elemento do array e a variável 'y' contendo o restante do array:



```
let lista = [1, 2, 3, 4, 5, 6];
```

```
apresentarDados(lista);
```

```
// Saída
```

```
x → 1
```

```
y → [2, 3, 4, 5, 6]
```

Exercícios JavaScript

19) Crie uma função que receba um parâmetro e usando os conceitos do operador Rest/Spread apresente a soma de todos os elementos:



```
somar(1, 2, 3, 4, 5, 6); // 21  
somar(42, 44); // 86
```

Exercícios JavaScript

20) Crie uma função que receba um parâmetro e usando os conceitos do operador Rest/Spread apresente a variável 'cliente1' que tenha todos os dados do parâmetro recebido (cliente) mas com o nome alterado para 'Felicity Jones', e a variável 'cliente2' com todos os dados de 'cliente' mas com o país alterado para 'EUA':



```
let cliente = {  
  nome: "Angelina Jolie",  
  idade: 48,  
  endereco: {  
    cidade: "Campo Grande",  
    uf: "MS",  
    pais: "Brasil",  
  }  
};
```

Exercícios JavaScript


21) Substitua a string do console.log para usar String Literals:



```
let nome = "Felicity Jones";  
let idade = 32;  
  
console.log("A " + nome + " tem " + idade + " anos");
```

Exercícios JavaScript

22) Faça uso do conceito de Object Short Syntax no objeto abaixo:



```
let nome = "Felicity Jones";  
let idade = 32;  
  
let cliente = {  
  nome: nome,  
  idade: idade,  
  cidade: "Campo Grande",  
};
```



Geek University

Evolua seu lado geek!

www.geekuniversity.com.br