

IOS – Instituto de
Oportunidade Social

Aula JS 10 - Classes e Funções



- > Classes
- > Métodos de uma classe
- > Mais um exemplo de classe
- > Lexical this no contexto de Funções

IOS – Instituto de
Oportunidade Social

Classe



Classes foram introduzidas no **JS** no **ECMAScript 2015**, mais conhecido como **ES6**, e elas são simplificações da linguagem para utilizarmos **herança baseadas nos protótipos**. Uma classe JavaScript **não é um objeto** e **sim um template para objetos JavaScript**. A sintaxe para **classes não introduz um novo modelo de herança de orientação a objetos em JavaScript**.

A sintaxe de uma classe em JavaScript é:

```
class NomeClasse {  
    constructor() { ... }  
}
```

Por exemplo, a classe a seguir tem o nome Carro e duas propriedades iniciais: nome e ano. Algumas literaturas chamam essas propriedades de atributos e/ou estados da classe.

```
class Carro {  
    constructor(nome, ano) {  
        this.nome = nome;  
        this.ano = ano;  
    }  
}
```

> Vamos Praticar

IOS – Instituto de
Oportunidade Social

Métodos de uma classe



Os métodos de uma classe são criados com a mesma sintaxe de um método de objetos. Além do constructor, você pode criar diversos protótipos de métodos na classe:

```
class NomeClasse {  
    constructor() { ... }  
    metodo_1() { ... }  
    metodo_2() { ... }  
    metodo_3() { ... }  
}
```


> Mais um exemplo de classe

Para fixar mais o conceito de classe, vamos para mais um exemplo.

> Vamos Praticar

IOS – Instituto de
Oportunidade Social

Lexical this no contexto de Funções



Podemos utilizar a palavra-chave `this` no contexto de funções. Em muitos casos, o valor `this` é determinado pela forma como a função é chamada. Por exemplo:

- > Em um método, o **`this`** faz referência ao próprio objeto.
- > Sozinho, o **`this`** faz referência ao objeto global.
- > Em uma função, o **`this`** faz referência ao objeto global.

- > Em uma função, no modo estrito, o **this** é **undefined**.
- > Em um evento, o **this** faz referência ao elemento que disparou o evento.
- > Em método como **call()** e **apply()** podem fazer referência do **this** para qualquer objeto.

> Vamos Praticar

IOS – Instituto de
Oportunidade Social

Conclusão



Procure sempre aprender e estudar mais. Seguem alguns links para você estudar e aprender mais:

Classes:

<https://developer.mozilla.org/pt-BR/docs/Web/JavaScript/Reference/Classes>

https://www.w3schools.com/js/js_classes.asp

this

<https://developer.mozilla.org/pt-BR/docs/Web/JavaScript/Reference/Operators/this>

https://www.w3schools.com/js/js_this.asp

Modo estrito:

https://developer.mozilla.org/pt-BR/docs/Web/JavaScript/Reference/Strict_mode

https://www.w3schools.com/js/js_strict.asp

Prototype:

https://developer.mozilla.org/pt-BR/docs/Learn/JavaScript/Objects/Object_prototypes

https://www.w3schools.com/js/js_object_prototypes.asp

IOS – Instituto de
Oportunidade Social

Exercícios



Crie uma classe chamada `contaBancaria`, no método constructor crie atributos `nome`, `idade`, `salário`, `sexo`, `agência`, `conta`, `numeroConta`.

Crie os métodos `contaPoupança`, `contaCorrente` e `contaEstudante`.

As regras para as contas são a seguinte

- Conta poupança com taxa de 1,5% ao mês.

- Conta corrente com taxa de 3,6% ao mês.
- Conta estudante com taxa de 1,2% ao mês.

Instancie 3 objetos um para cada conta da classe `contaBancaria` e imprima as seguintes informações:

Imprimir no console os dados do usuário, e o novo salário da aluna com o desconto aplicado