

IOS – Instituto de  
Oportunidade Social

## Aula JS 10 - Classes e Funções



- > Classes
- > Métodos de uma classe
- > Lexical this no contexto de Funções
- > Prototype de objetos

IOS – Instituto de  
Oportunidade Social

Classes



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() { ... }  
}
```

## Exemplo de 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() { ... }  
}
```

> Vamos Praticar



IOS – Instituto de  
Oportunidade Social

Lexical this no contexto de Funções



# 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

Prototype de objetos



O JS permite criar protótipos de objetos, que basicamente é permitir criar atributos ou métodos em objetos já criados anteriormente. E esses novos protótipos são herdados automaticamente.

> Vamos Praticar

IOS – Instituto de  
Oportunidade Social

Para Aprender mais



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](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](https://www.w3schools.com/js/js_this.asp)

## Modo estrito:

[https://developer.mozilla.org/pt-BR/docs/Web/JavaScript/Reference/Strict\\_mode](https://developer.mozilla.org/pt-BR/docs/Web/JavaScript/Reference/Strict_mode) [https://www.w3schools.com/js/js\\_strict.asp](https://www.w3schools.com/js/js_strict.asp)

## Prototype:

[https://developer.mozilla.org/pt-BR/docs/Learn/JavaScript/Objects/Object\\_prototypes](https://developer.mozilla.org/pt-BR/docs/Learn/JavaScript/Objects/Object_prototypes)  
[https://www.w3schools.com/js/js\\_object\\_prototypes.asp](https://www.w3schools.com/js/js_object_prototypes.asp)

IOS – Instituto de  
Oportunidade Social

Exercício





## Exercício 1

Crie uma classe chamada aluno com as seguintes informações

**id (number), nome (String), idade (number), desistente (boolean).**

cria três objetos da classe acima com as informações que você desejar.

apresente no console.log os três objetos