



# Classes

As classes foram introduzidas no ECMAScript 2015 e são simplificações da linguagem para as heranças baseadas nos protótipos. Não utiliza um novo modelo de herança de orientação a objetos em JavaScript, pois são formas mais simplificadas de criar objetos e lidar com herança.

## Declaração de classe

A declaração de classe cria uma nova classe com um determinado nome usando herança baseada em **protótipo**.

## Definição de Classes

### Class Expression

A **class expression** é uma maneira de definir uma classe no ECMAScript 2015. Semelhante às expressões de função, as expressões de classe podem ser nomeadas ou não. Se nomeadas, o nome da classe é local apenas para o corpo da classe.

## Sintaxe





Fonte: Autoral

## Descrição

Uma expressão de classe tem uma sintaxe semelhante a uma declaração de classe (instrução). Tal como acontece com as instruções de classe, o corpo de uma expressão de classe é executado no modo estrito.

Existem várias diferenças entre expressões de classe e instruções de classe. São elas:

- As expressões de classe podem omitir o nome da classe (“identificador de ligação”), o que não é possível com instruções de classe.
- Expressões de classe permitem que você redefina (declare novamente) classes sem lançar um `SyntaxError`. Este não é o caso com declarações de classe.

O método do construtor é opcional. As classes geradas com expressões de classe sempre responderão a `typeof` com o valor “função”.

## Uma expressão de classe simples

Esta é apenas uma expressão de classe anônima simples que você pode se referir usando a variável `Foo`.

Fonte: Autoral

## Expressões de classe nomeada



Se quiser fazer referência à classe atual dentro do corpo da classe, você pode criar uma expressão de classe nomeada. O nome só é visível dentro do escopo da própria expressão de classe.

### Atividade Extra

The Secret Life of Objects. Acesse o link se quiser saber um pouco mais sobre os objetos em POO e JavaScript.

Link: [https://eloquentjavascript.net/06\\_object.html](https://eloquentjavascript.net/06_object.html) (**Acesso em 21/07/2022**)

O vídeo a seguir mostra o funcionamento dos objetos em JavaScript de uma forma clara e simples.

Link: <https://www.youtube.com/watch?v=n5uiJr-v0KQ> (**Acesso em 21/07/2022**)

### Referência Bibliográfica

FLANAGAN, David. **JavaScript: O Guia Definitivo**. 6ª Ed. Porto Alegre: Bookman, 2013.

FREEMAN, Eric. **Use a cabeça!: programação JavaScript**. 1ª Ed. São Paulo: Alta Books, 2016.

### Atividade Prática 15 – Classes

**Título da Prática:** Criação e manipulação de classes.



**Objetivos:** Compreender a importância da utilização de Classes no desenvolvimento de JavaScript.

**Materiais, Métodos e Ferramentas:** Visual Studio Code ou Bloco de notas do Windows.

## Atividade Prática

### 1º Passo:

Leia atentamente o texto.


Uma classe JavaScript é um modelo para a criação de objetos. Uma classe encapsula dados e funções que manipulam dados.

Classes são bits de código que abrangem vários objetos, métodos e permitem a manipulação de suas variáveis e funções de membro. Dentro de cada linguagem, uma classe tem uma sintaxe diferente e o mesmo vale para o Javascript.

Ao contrário de outras linguagens de programação, como Java e C#, as classes ES6 são apenas funções especiais, ou seja, na linguagem de programação em JavaScript, uma classe é simplesmente uma variante de funções. Usando a funcionalidade de classe, pode-se definir o construtor e as funções de protótipo para os objetos membros em um bloco abrangente.

### 2º Passo:

Para criar uma classe em Javascript, algumas coisas devem ser observadas:

Uma classe é criada usando a palavra-chave **class**  como mostrado no exemplo abaixo:

Uma classe tem um construtor padrão que está vazio, mas também pode ser definido dentro do método da classe. Ao contrário de outras linguagens, para declarar o construtor, o Javascript possui uma palavra-chave específica chamada **constructor**.

O(s) parâmetro(s) fornecido(s) ao **constructor** são usados para definir elementos dentro da classe (membros da classe). O exemplo abaixo mostra como podemos aplicar esse princípio:

Vários métodos podem ser definidos em uma classe Javascript, como mostrado no exemplo abaixo:

Uma instância de uma classe é definida usando a palavra-chave **new**. Para ver como isso é feito, veja o código abaixo:

### 3º Passo:

Dado o código fonte de referência abaixo:



Figura 1 – Reprodução do código fonte.

Fonte: o próprio autor (2023)

Implemente a inclusão dos seguintes atributos para a class Carro:

- Ano
- KM
- Câmbio
- Combustível
- Final de placa
- Cor

Após incluir esses atributos, você deverá realizar um novo cadastro na íntegra (contendo todos os atributos criados) e deverá exibir na página HTML.

### **Comentários e conclusões:**

1. O que aprendeu com esta atividade?
2. Tudo funcionou como o esperado?

### 3. Quais foram as principais dificuldades?



#### **Gabarito Atividade Prática**

Segue o código fonte da atividade proposta:

Figura 2 – Reprodução do código fonte.

Fonte: o próprio autor (2023)

Figura 3 – Reprodução da execução do código fonte.

Fonte: o próprio autor (2023)

**Ir para exercício**