

# Introdução a objetos em JavaScript

Samuel Amaro

25 de abril de 2021

## Aula Protótipos de objetos - anotações da aula

Protótipos são o mecanismo pelo qual objetos JavaScript herdam recursos uns dos outros. Neste artigo, explicamos como as cadeias de protótipos funcionam e observamos como a propriedade `prototype` pode ser usada para adicionar métodos aos construtores existentes.

Protótipos são o mecanismo pelo qual os objetos JavaScript herdam as características de outros e como eles funcionam de uma forma diferente dos componentes de linguagens de programação clássicas. Neste artigo, vamos explorar essas diferenças, explicar como a cadeia de protótipos trabalha e perceber como um protótipo de propriedade pode ser usado para adicionar métodos a construtores existentes.

**Objetivo:** Entender protótipos de objetos JavaScript, como a cadeia de protótipos funciona, e como adicionar novos métodos à propriedade `prototype`.

## Uma linguagem baseada em protótipos ?

O JavaScript é frequentemente descrito como uma **linguagem baseada em protótipos** — para fornecer herança, os objetos podem ter um **objeto de protótipo**, que atua como um objeto de modelo do qual herda métodos e propriedades. O objeto de protótipo de um objeto também pode ter um objeto de protótipo, do qual herda métodos e propriedades, e assim por diante. Isso geralmente é chamado de **cadeia de protótipos** e explica por que objetos diferentes têm propriedades e métodos definidos em outros objetos disponíveis para eles.

Bem, para ser exato, as propriedades e os métodos são definidos na



Figura 1: Prototipação de objetos, propriedade prototype

propriedade **prototype** nas funções construtoras dos Objetos, não nas próprias instâncias do objeto.

Em JavaScript, é feito um link entre a instância do objeto e seu protótipo (sua propriedade **\_\_proto\_\_**, que é derivada da propriedade **prototype** no construtor), e as propriedades e os métodos são encontrados percorrendo a cadeia de protótipos.

**Nota:** É importante entender que há uma distinção entre o protótipo de um objeto (que está disponível por meio de `Object.getPrototypeOf(obj)`, ou por meio da propriedade **\_\_proto\_\_**) e a propriedade **prototype** em funções construtoras. O primeiro é a propriedade em cada instância e o último é a propriedade no construtor. Ou seja, `Object.getPrototypeOf(new Foobar())` refere-se ao mesmo objeto que `Foobar.prototype`.