

## ≡ Navegação de item

# POO baseada em Protótipos: o caso de JavaScript

JavaScript é uma linguagem muito popular pois ela pode ser executada dentro dos navegadores Web e são muito usadas para a geração de páginas Web bem dinâmicas (como essa página atual que você está lendo, por exemplo). É interessante observar que, apesar do nome, a linguagem não tem nada a ver com Java. Ela é derivada da linguagem ECMAScript [↗](#), o nome JavaScript foi adotado por questões de marketing apenas.

Antigamente, JavaScript tinha uma má reputação pois muita gente que não sabia programar direito (p.ex., com formação em outras áreas que não Computação) escrevia código em JavaScript dentro de páginas Web e esse código era muito sujo e mal escrito. Mas o problema não era com a linguagem; a linguagem em si é interessante. Nos últimos anos, JavaScript tem se tornado mais popular dentro da comunidade de Computação e muito código bem feito tem surgido. Hoje em dia, JavaScript também é usado para escrever código que é executado no servidor, não só no navegador Web. Essa tecnologia é baseada em um software livre chamado Node.js

O modelo de objetos em JavaScript é um pouco diferente do modelo de Java e Python que nós vimos até agora. Java e Python adotam o modelo mais comum baseado em classes, ou seja, objetos são criados a partir de classes.

Já JavaScript usa a ideia de protótipos, objetos são instanciados a partir de outros objetos prototípicos. Em JavaScript um objeto é usado como molde para a criação de novos objetos. Essa ideia se originou na linguagem Self [↗](#), uma variante de Smalltalk que apesar de ser bem OO, não tinha classes; um objeto era criado a partir da clonagem de outro objeto e não a partir de uma classe. O padrão de projeto **Prototype** do livro GoF (que iremos aprender na Semana 6) explica como esse mecanismo funciona.

Neste curso, nós não vamos entrar em detalhes da programação OO em JavaScript (afinal, já temos conteúdo demais no curso). Mas, se você tem curiosidade e quer aprender um pouco mais sobre como funciona esse tipo um pouco diferente de orientação a objetos, eu recomendo que você comece pela leitura do seguinte texto oferecido pelos nossos colegas da Fundação Mozilla:

Versão em português:

[https://developer.mozilla.org/pt-BR/docs/Aprender/JavaScript/Objetos/Object-oriented\\_JS](https://developer.mozilla.org/pt-BR/docs/Aprender/JavaScript/Objetos/Object-oriented_JS) [↗](#)

Versão em inglês:

[https://developer.mozilla.org/en-US/docs/Learn/JavaScript/Objects/Object-oriented\\_JS](https://developer.mozilla.org/en-US/docs/Learn/JavaScript/Objects/Object-oriented_JS) [↗](#)