**Herança?**

O termo "**Herança**" é muito usado para descrever que as características de um objeto são passadas para outro objeto. Assim como uma filha ou filho pode herdar características da sua mãe ou pai (geneticamente). Por isso estamos usando esse termo para descrever o comportamento desejado. Além disso, a maioria das literaturas, incluindo as documentações (como [MDN](https://developer.mozilla.org/pt-BR/docs/Aprender/JavaScript/Objetos/Heran%C3%A7a)) citam a palavra herança.

Vamos continuar usando o termo "**Herança**" até o final do curso, porém, Javascript não é uma linguagem que apresenta essa característica em específico, portanto não existe "**herança**" aqui. O que ocorre, na verdade, é "**delegação**". Esse seria o termo mais correto para descrever "**Herança**" em JS.

Quando fazemos algo como:

1. function Produto(nome, preco) {
2. this.nome = nome;
3. this.preco = preco;
4. }
6. Produto.prototype.aumento = function (quantia) {
7. this.preco += quantia;
8. };
10. function Camiseta(nome, preco, cor) {
11. Produto.call(this, nome, preco);
12. this.cor = cor;
13. }
14. Camiseta.prototype = Object.create(Produto.prototype);
15. Camiseta.prototype.constructor = Camiseta;

Mais especificamente no trecho:

1. Camiseta.prototype = Object.create(Produto.prototype);

Estamos, claramente, falando que o Camiseta.prototype é um novo objeto vazio que tem como protótipo Produto.prototype. Dessa forma, estamos adicionando Camiseta.prototype na cadeia de protótipos de Produto. Por consequência, tudo o que não existir em Camiseta nem em Camiseta.prototype será delegado para Produto.prototype (e assim por diante até chegar no topo da cadeia de protótipos). Por isso vejo o termo delegação como mais adequado para este comportamento.

Novamente, vamos continuar usando o termo "**Herança**", mas achei que seria interessante que você soubesse disso neste momento.