# Objetos JavaScript

[❮ Anterior](https://translate.googleusercontent.com/translate_c?depth=1&hl=pt-BR&ie=UTF8&prev=_t&rurl=translate.google.com.br&sl=en&sp=nmt4&tl=pt-BR&u=https://www.w3schools.com/js/js_functions.asp&usg=ALkJrhgtjcL0oxo79PlLoWLiEpHdnff8YQ)[Próximo ❯](https://translate.googleusercontent.com/translate_c?depth=1&hl=pt-BR&ie=UTF8&prev=_t&rurl=translate.google.com.br&sl=en&sp=nmt4&tl=pt-BR&u=https://www.w3schools.com/js/js_scope.asp&usg=ALkJrhipyRgd9kW2FwOT0R1Cl7ZkQbcUOQ)

## Objetos, propriedades e métodos de vida real

Na vida real, um carro é um **objeto** .

Um carro tem **propriedades** como peso e cor, e **métodos** como iniciar e parar:

|  |  |  |
| --- | --- | --- |
| **Objeto** | **Propriedades** | **Métodos** |
| https://www.w3schools.com/js/objectExplained.gif | car.name = Fiat   car.model = 500   carro. peso = 850 kg   car.color = branco | car.start ()   car.drive ()   car.brake ()   car.stop () |

Todos os carros têm as mesmas **propriedades** , mas os valores das propriedades diferem de carro para carro.

Todos os carros têm os mesmos **métodos** , mas os métodos são realizados em momentos diferentes.

## Objetos JavaScript

Você já aprendeu que as variáveis ​​JavaScript são contêineres para valores de dados.

Este código atribui um **valor simples** (Fiat) a uma **variável** chamada carro:

var car = "Fiat";

[Tente você mesmo "](https://translate.googleusercontent.com/translate_c?depth=1&hl=pt-BR&ie=UTF8&prev=_t&rurl=translate.google.com.br&sl=en&sp=nmt4&tl=pt-BR&u=https://www.w3schools.com/js/tryit.asp%3Ffilename%3Dtryjs_objects_variable&usg=ALkJrhjQL-3Pu7ROekMqXg0Uh1aYrpZYFQ)

Os objetos também são variáveis. Mas objetos podem conter muitos valores.

Este código atribui **muitos valores** (Fiat, 500, branco) a uma **variável** chamada carro:

var car = {type:"Fiat", model:"500", color:"white"};

[Tente você mesmo "](https://translate.googleusercontent.com/translate_c?depth=1&hl=pt-BR&ie=UTF8&prev=_t&rurl=translate.google.com.br&sl=en&sp=nmt4&tl=pt-BR&u=https://www.w3schools.com/js/tryit.asp%3Ffilename%3Dtryjs_objects_object&usg=ALkJrhhHDdm_c37wihdYbitow5nkLiI-mg)

Os valores são escritos como **nome:** pares de **valores** (nome e valor separados por dois pontos).

Objetos JavaScript são recipientes para **valores nomeados** .

## Propriedades do objeto

O nome: pares de valores (em objetos JavaScript) são chamados de **propriedades** .

var person = {firstName:"John", lastName:"Doe", age:50, eyeColor:"blue"};

|  |  |
| --- | --- |
| **Propriedade** | **Valor da propriedade** |
| primeiro nome | John |
| último nome | Corça |
| era | 50 |
| cor dos olhos | azul |

## Métodos de objeto

Os métodos são **ações** que podem ser realizadas em objetos.

Os métodos são armazenados em propriedades como **definições de função** .

|  |  |
| --- | --- |
| **Propriedade** | **Valor da propriedade** |
| primeiro nome | John |
| último nome | Corça |
| era | 50 |
| cor dos olhos | azul |
| nome completo | function () {return this.firstName + "" + this.lastName;} |

Objetos JavaScript são recipientes para valores nomeados chamados propriedades ou métodos.

## Definição do objeto

Você define (e cria) um objeto JavaScript com um objeto literal:

### Exemplo

var person = {firstName:"John", lastName:"Doe", age:50, eyeColor:"blue"};

[Tente você mesmo "](https://translate.googleusercontent.com/translate_c?depth=1&hl=pt-BR&ie=UTF8&prev=_t&rurl=translate.google.com.br&sl=en&sp=nmt4&tl=pt-BR&u=https://www.w3schools.com/js/tryit.asp%3Ffilename%3Dtryjs_objects_create_1&usg=ALkJrhj4KPZPyxBz56_KZEEFyYxYcuYg1w)

Espaços e quebras de linha não são importantes. Uma definição de objeto pode abranger várias linhas:

### Exemplo

var person = {  
    firstName:"John",  
    lastName:"Doe",  
    age:50,  
    eyeColor:"blue"  
};

[Tente você mesmo "](https://translate.googleusercontent.com/translate_c?depth=1&hl=pt-BR&ie=UTF8&prev=_t&rurl=translate.google.com.br&sl=en&sp=nmt4&tl=pt-BR&u=https://www.w3schools.com/js/tryit.asp%3Ffilename%3Dtryjs_objects_create_2&usg=ALkJrhjvA75-TmJB9-Nu4zG8bqrTHKkicw)

## Acessando propriedades do objeto

Você pode acessar as propriedades do objeto de duas maneiras:

*objectName.propertyName*

ou

*objectName["propertyName"]*

### Exemplo 1

person.lastName;

[Tente você mesmo "](https://translate.googleusercontent.com/translate_c?depth=1&hl=pt-BR&ie=UTF8&prev=_t&rurl=translate.google.com.br&sl=en&sp=nmt4&tl=pt-BR&u=https://www.w3schools.com/js/tryit.asp%3Ffilename%3Dtryjs_objects_properties_1&usg=ALkJrhjN3bLi8VandcwuboibPOdQrrokGw)

### Exemplo2

person["lastName"];

[Tente você mesmo "](https://translate.googleusercontent.com/translate_c?depth=1&hl=pt-BR&ie=UTF8&prev=_t&rurl=translate.google.com.br&sl=en&sp=nmt4&tl=pt-BR&u=https://www.w3schools.com/js/tryit.asp%3Ffilename%3Dtryjs_objects_properties_2&usg=ALkJrhi6oK18IkCfYQojSF7zulJ79LemyQ)

## Acessando Métodos de Objetos

Você acessa um método de objeto com a seguinte sintaxe:

*objectName.methodName()*

### Exemplo

name = person.fullName();

[Tente você mesmo "](https://translate.googleusercontent.com/translate_c?depth=1&hl=pt-BR&ie=UTF8&prev=_t&rurl=translate.google.com.br&sl=en&sp=nmt4&tl=pt-BR&u=https://www.w3schools.com/js/tryit.asp%3Ffilename%3Dtryjs_objects_method&usg=ALkJrhgo6q46vWZ-kGa2Ti2iCMAwVRA1zA)

Se você acessar o **método** fullName, sem (), ele retornará a **definição** da **função** :

### Exemplo

name = person.fullName;

[Tente você mesmo "](https://translate.googleusercontent.com/translate_c?depth=1&hl=pt-BR&ie=UTF8&prev=_t&rurl=translate.google.com.br&sl=en&sp=nmt4&tl=pt-BR&u=https://www.w3schools.com/js/tryit.asp%3Ffilename%3Dtryjs_objects_function&usg=ALkJrhgYCtw4ghsHUX6-urFhsAbJx87kOg)

Um método é, na verdade, uma definição de função armazenada como um valor de propriedade.

## Não declare strings, números e booleanos como objetos!

Quando uma variável JavaScript é declarada com a palavra-chave "novo", a variável é criada como um objeto:

var x = new String();        // Declares x as a String object  
var y = new Number();        // Declares y as a Number object  
var z = new Boolean();       // Declares z as a Boolean object

Evite String, Number e objetos booleanos. Eles complicam seu código e diminuem a velocidade de execução.