

W3schools

TutorialsReferencesExercises

HTMLCSSJAVASCRIPTSQLPYTHONPHPBOOTSTRAP

Português

Depuração JS

Guia de estilo JS

JS Best Practices

JS Erros

JS Performance

Palavras Reservadas JS

Objetos JS

Definições de Objeto

Propriedades do objeto

Métodos de Objeto

Exibição de objeto

Acessores de objeto

Construtores de objetos

Protótipos de objeto

Referência de Objeto

Mapa de Objetos ()

Conjunto de objetos ()

Funções JS

Log in

Paid Courses

Protótipos de objeto JavaScript

< Anterior

Próximo >

Todos os objetos JavaScript herdam propriedades e métodos de um protótipo.

No capítulo anterior, aprendemos como usar um **construtor de objeto** :

Exemplo

```
function Person(first, last, age, eyecolor) {
  this.firstName = first;
  this.lastName = last;
  this.age = age;
  this.eyeColor = eyecolor;
}

const myFather = new Person("John", "Doe", 50, "blue");
const myMother = new Person("Sally", "Rally", 48, "green");
```

Tente você mesmo "

Também aprendemos que **não** é possível adicionar uma nova propriedade a um construtor de objeto existente:

Exemplo

```
Person.nationality = "English";
```

Tente você mesmo "

Para adicionar uma nova propriedade a um construtor, você deve adicioná-la à função do construtor:

Exemplo

```
function Person(first, last, age, eyecolor) {
  this.firstName = first;
  this.lastName = last;
  this.age = age;
  this.eyeColor = eyecolor;
  this.nationality = "English";
}
```

Tente você mesmo "

Herança de protótipo

All JavaScript objects inherit properties and methods from a prototype:

- `Date` objects inherit from `Date.prototype`
- `Array` objects inherit from `Array.prototype`
- `Person` objects inherit from `Person.prototype`

The `Object.prototype` is on the top of the prototype inheritance chain:

`Date` objects, `Array` objects, and `Person` objects inherit from `Object.prototype`.

Adding Properties and Methods to Objects

Sometimes you want to add new properties (or methods) to all existing objects of a given type.

Sometimes you want to add new properties (or methods) to an object constructor.

Using the **prototype** Property

The JavaScript `prototype` property allows you to add new properties to object constructors:

Example

```
function Person(first, last, age, eyecolor) {
  this.firstName = first;
  this.lastName = last;
  this.age = age;
  this.eyeColor = eyecolor;
}

Person.prototype.nationality = "English";
```

Try it Yourself »

The JavaScript `prototype` property also allows you to add new methods to objects constructors:

Example

```
function Person(first, last, age, eyecolor) {
  this.firstName = first;
  this.lastName = last;
  this.age = age;
  this.eyeColor = eyecolor;
}

Person.prototype.name = function() {
  return this.firstName + " " + this.lastName;
};
```

Try it Yourself »

Modifique apenas seus **próprios** protótipos. Nunca modifique os protótipos de objetos JavaScript padrão.

< Anterior

Próximo >

PROPAGANDA

NIVEA

O MÊS DA Beleza

ESTÁ DE VOLTA!

SABE MAIS

Descontos de 20%

Sente-te linda

COLOR PICKER

COMO NÓS

Obtenha a certificação completando um curso hoje!

W3schools

3

CERTIFIED

iniciar

JOGO DE CÓDIGOS

Jogar um jogo

Reportar erro

Fórum

Cerca de

Comprar

Principais tutoriais

Tutorial HTML Tutorial
CSS Tutorial
JavaScript
Como fazer Tutorial
SQL Tutorial
Python Tutorial
W3.CSS Tutorial
Bootstrap Tutorial
PHP Tutorial
Java Tutorial
C ++ Tutorial
jQuery Tutorial

Referências principais

HTML Reference
CSS Reference
JavaScript Reference
SQL Reference
Python Reference
W3.CSS Reference
Bootstrap Reference
PHP Reference
HTML Colors
Java Reference
Angular Reference
jQuery Reference

Top Examples

HTML Examples
CSS Examples
JavaScript Examples
How To Examples
SQL Examples
Python Examples
W3.CSS Examples
Bootstrap Examples
PHP Examples
Java Examples
XML Examples
jQuery Examples

Web Courses

HTML Course
CSS Course
JavaScript Course
Front End Course
SQL Course
Python Course
PHP Course
jQuery Course
Java Course
C++ Course
C# Course
XML Course

Get Certified »

