# Protótipos de 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_object_constructors.asp&usg=ALkJrhj7PO6tylxFvmegawLFyeOJJAyzww)[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_function_definition.asp&usg=ALkJrhgFrOUXIbhiFb8cT5eBg5qP9fugJg)

Cada objeto JavaScript possui um protótipo. O protótipo também é um objeto.

Todos os objetos JavaScript herdam suas propriedades e métodos de seu protótipo.

## Protótipos de JavaScript

Todos os objetos JavaScript herdam as propriedades e os métodos do seu protótipo.

Objetos criados usando um objeto literal, ou com o novo Object (), herdam de um protótipo chamado Object.prototype.

Os objetos criados com nova Data () herdam o Date.prototype.

O Object.prototype está no topo da cadeia do protótipo.

Todos os objetos JavaScript (Date, Array, RegExp, Function, ....) herdam do Object.prototype.

## Criando um Protótipo

A maneira padrão de criar um protótipo de objeto é usar uma função de construtor de objetos:

### Exemplo

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

Com uma função de construtor, você pode usar a **nova** palavra-chave para criar novos objetos a partir do mesmo protótipo:

### Exemplo

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

[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_object_prototype1&usg=ALkJrhgTXP09CkA9keoehXTfyFToz32qcQ)

A função do construtor é o protótipo dos objetos Pessoa.   
É considerada uma boa prática para nomear a função do construtor com uma primeira letra maiúscula.

## Adicionando propriedades e métodos aos objetos

Às vezes, você deseja adicionar novas propriedades (ou métodos) a um objeto existente.

Às vezes, você deseja adicionar novas propriedades (ou métodos) a todos os objetos existentes de um determinado tipo.

Às vezes, você deseja adicionar novas propriedades (ou métodos) a um protótipo de objeto.

## Adicionando uma propriedade a um objeto

Adicionar uma nova propriedade a um objeto existente é fácil:

### Exemplo

myFather.nationality = "English";

[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_object_prototype2&usg=ALkJrhik8Z_P0tdsH5Z26uJWuQ6fe2kJBQ)

A propriedade será adicionada ao meu pai. Não para minha mãe. Não para objetos de outra pessoa.

## Adicionando um Método a um Objeto

Adicionando um novo método a um objeto existente também é fácil:

### Exemplo

myFather.name = function () {  
    return this.firstName + " " + this.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_object_prototype8&usg=ALkJrhhue_yzYjEKHXH_KXTDDlDiW_BVyQ)

O método será adicionado ao meu pai. Não para minha mãe.

## Adicionando propriedades a um protótipo

Você não pode adicionar uma nova propriedade a um protótipo da mesma maneira que você adiciona uma nova propriedade a um objeto existente, porque o protótipo não é um objeto existente.

### Exemplo

Person.nationality = "English";

[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_object_prototype3&usg=ALkJrhgHgZD0bgleMVgvQegTLjAW8uKmwg)

Para adicionar uma nova propriedade a um protótipo, você deve adicioná-la à função de 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 "](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_object_prototype4&usg=ALkJrhjzckoGGTF5GRWlmxlLfHZMRBKfuw)

As propriedades do protótipo podem ter valores de protótipo (valores padrão).

## Adicionando Métodos a um Protótipo

Sua função de construtor também pode definir métodos:

### Exemplo

function Person(first, last, age, eyecolor) {  
    this.firstName = first;  
    this.lastName = last;  
    this.age = age;  
    this.eyeColor = eyecolor;  
    this.name = function() {return this.firstName + " " + this.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_object_prototype7&usg=ALkJrhgr_eX3ie6TDkZN9JscAxf2_i4WCA)

A adição de métodos a um objeto é feita dentro da função do construtor:

### Exemplo

function Person(firstName, lastName, age, eyeColor) {  
    this.firstName = firstName;    
    this.lastName = lastName;  
    this.age = age;  
    this.eyeColor = eyeColor;  
    this.changeName = function (name) {  
        this.lastName = name;  
    };  
}

A função changeName () atribui o valor do nome à propriedade lastName da pessoa.

### Agora você pode tentar:

myMother.changeName("Doe");

[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_create_object3&usg=ALkJrhj-yJ-GkHd3NIeCJ5fn6Jz3UY0MwQ)

O JavaScript sabe para qual pessoa você está falando "substituindo" **isso** pela **minha Mãe** .

## Usando o ****protótipo de**** propriedade

A propriedade protótipo JavaScript permite que você adicione novas propriedades a um protótipo existente:

### Exemplo

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

[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_object_prototype5&usg=ALkJrhgZAgGFvM2syJJ7op1MqLGC9tz6qA)

A propriedade protótipo JavaScript também permite que você adicione novos métodos a um protótipo existente:

### Exemplo

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;  
};