



Start learning for free



Métodos de objeto JavaScript ES5

< Anterior

Próximo >

ECMAScript 5 adicionou muitos novos métodos de objeto ao JavaScript.

ES5 Novos Métodos de Objeto

```
// Adding or changing an object property
Object.defineProperty(object, property, descriptor)

// Adding or changing many object properties
Object.defineProperties(object, descriptors)

// Accessing Properties
Object.getOwnPropertyDescriptor(object, property)

// Returns all Properties as an array
Object.getOwnPropertyNames(object)

// Returns enumerable properties as an array
Object.keys(object)


// Accessing the prototype
Object.getPrototypeOf(object)

// Prevents adding properties to an object
Object.preventExtensions(object)
// Returns true if properties can be added to an object
Object.isExtensible(object)


// Prevents changes of object properties (not values)
Object.seal(object)
// Returns true if object is sealed
Object.isSealed(object)

// Prevents any changes to an object
Object.freeze(object)
// Returns true if object is frozen
Object.isFrozen(object)
```

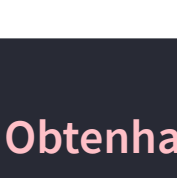
PROPAGANDA




COLOR PICKER



COMO NÓS



Obtenha a certificação completando um curso hoje!



iniciar

Alterar um valor de propriedade

Sintaxe

```
Object.defineProperty(object, property, {value : value})
```

Este exemplo altera o valor de uma propriedade:


Exemplo

```
const person = {
  firstName: "John",
  lastName: "Doe",
  language : "EN"
};

// Change a property
Object.defineProperty(person, "language", {value : "NO"});
```

Tente você mesmo "

JOGO DE CÓDIGOS



Jogar um jogo

Alteração de metadados

ES5 permite que os seguintes metadados de propriedade sejam alterados:

```
writable : true // Property value can be changed
enumerable : true // Property can be enumerated
configurable : true // Property can be reconfigured
```

```
writable : false // Property value can not be changed
enumerable : false // Property can be not enumerated
configurable : false // Property can be not reconfigured
```

ES5 permite que getters e setters sejam alterados:

```
// Defining a getter
get: function() { return language }
// Defining a setter
set: function(value) { language = value }
```

Este exemplo torna a linguagem somente leitura:

```
Object.defineProperty(person, "language", {writable:false});
```

Este exemplo torna a linguagem não enumerável:

```
Object.defineProperty(person, "language", {enumerable:false});
```

Listando todas as propriedades

Este exemplo lista todas as propriedades de um objeto:

Exemplo

```
const person = {
  firstName: "John",
  lastName: "Doe",
  language : "EN"
};

Object.defineProperty(person, "language", {enumerable:false});
Object.getOwnPropertyNames(person); // Returns an array of properties
```

Tente você mesmo "

Listando Propriedades Enumeráveis

Este exemplo lista apenas as propriedades enumeráveis de um objeto:

Exemplo

```
const person = {
  firstName: "John",
  lastName: "Doe",
  language : "EN"
};

Object.defineProperty(person, "language", {enumerable:false});
Object.keys(person); // Returns an array of enumerable properties
```

Tente você mesmo "

Adicionando uma Propriedade

Este exemplo adiciona uma nova propriedade a um objeto:

Exemplo

```
// Create an object:
const person = {
  firstName: "John",
  lastName: "Doe",
  language : "EN"
};

// Add a property
Object.defineProperty(person, "year", {value:"2008"});
```

Tente você mesmo "

Adicionando getters e setters

O `Object.defineProperty()` método também pode ser usado para adicionar Getters e Setters:

Exemplo

```
//Create an object
const person = {firstName:"John", lastName:"Doe"};

// Define a getter
Object.defineProperty(person, "fullName", {
  get: function () {return this.firstName + " " + this.lastName;}
});
```

Tente você mesmo "

Um exemplo de contador

Exemplo

```
// Define object
const obj = {counter:0};

// Define setters
Object.defineProperty(obj, "reset", {
  get : function () {this.counter = 0;}
});
Object.defineProperty(obj, "increment", {
  get : function () {this.counter++;}
});
Object.defineProperty(obj, "decrement", {
  get : function () {this.counter--;}
});
Object.defineProperty(obj, "add", {
  set : function (value) {this.counter += value;}
});
Object.defineProperty(obj, "subtract", {
  set : function (i) {this.counter -= i;}
});

// Play with the counter:
obj.reset;
obj.add = 5;
obj.subtract = 1;
obj.increment;
obj.decrement;
```

Tente você mesmo "

< Anterior

Próximo >