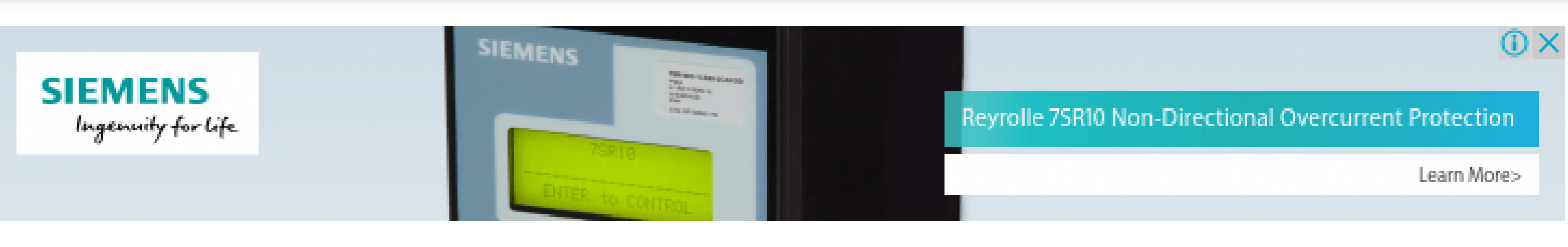
Classes JS
JS JSON
Depuração JS
Guia de estilo JS
JS Best Practices
JS Erros
JS Performance
Palavras Reservadas JS

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 ()



Acessadores de objeto JavaScript

‹ Anterior Próximo ›

Acessores de JavaScript (getters e setters)

ECMAScript 5 (ES5 2009) introduziu Getter e Setters.

Getters e setters permitem definir acessadores de objeto (propriedades computadas).

JavaScript Getter (a palavra-chave get)

Este exemplo usa uma `lang` propriedade para `get` o valor da `language` propriedade.

Exemplo

```
// Create an object:
const person = {
  firstName: "John",
  lastName: "Doe",
  language: "en",
  get lang() {
    return this.language;
  }
};

// Display data from the object using a getter:
document.getElementById("demo").innerHTML = person.lang;
```

Tente você mesmo "

Setter JavaScript (a palavra-chave definida)

Este exemplo usa uma `lang` propriedade para `set` o valor da `language` propriedade.

Exemplo

```
const person = {
  firstName: "John",
  lastName: "Doe",
  language: "",
  set lang(lang) {
    this.language = lang;
  }
};

// Set an object property using a setter:
person.lang = "en";

// Display data from the object:
document.getElementById("demo").innerHTML = person.language;
```

Tente você mesmo "

Função JavaScript ou getter?

Quais são as diferenças entre esses dois exemplos?

Exemplo 1

```
const person = {
  firstName: "John",
  lastName: "Doe",
  fullName: function() {
    return this.firstName + " " + this.lastName;
  }
};

// Display data from the object using a method:
document.getElementById("demo").innerHTML = person.fullName();
```

Tente você mesmo "

Exemplo 2

```
const person = {
  firstName: "John",
  lastName: "Doe",
  get fullName() {
    return this.firstName + " " + this.lastName;
  }
};

// Display data from the object using a getter:
document.getElementById("demo").innerHTML = person.fullName;
```

Tente você mesmo "

Exemplo 1 acessar fullName como uma função: person.fullName ().

Exemplo 2 acesse fullName como uma propriedade: person.fullName.

O segundo exemplo fornece uma sintaxe mais simples.

Qualidade de Dados

JavaScript pode garantir melhor qualidade de dados ao usar getters e setters.

Usar a `lang` propriedade, neste exemplo, retorna o valor da `language` propriedade em maiúsculas:

Exemplo

```
// Create an object:
const person = {
  firstName: "John",
  lastName: "Doe",
  language: "en",
  get lang() {
    return this.language.toUpperCase();
  }
};

// Display data from the object using a getter:
document.getElementById("demo").innerHTML = person.lang;
```

Tente você mesmo "

Usar a `lang` propriedade, neste exemplo, armazena um valor em maiúsculas na `language` propriedade:

Exemplo

```
const person = {
  firstName: "John",
  lastName: "Doe",
  language: "",
  set lang(lang) {
    this.language = lang.toUpperCase();
  }
};

// Set an object property using a setter:
person.lang = "en";

// Display data from the object:
document.getElementById("demo").innerHTML = person.language;
```

Tente você mesmo "

Por que usar getters e setters?

- Oferece uma sintaxe mais simples
- Ele permite sintaxe igual para propriedades e métodos
- Pode garantir uma melhor qualidade de dados
- É útil para fazer coisas nos bastidores

Object.defineProperty ()

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

Um exemplo de contador

```
// 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 (value) {this.counter -= value;}
});

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

Tente você mesmo "

‹ Anterior Próximo ›

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
Referência HTML Referência
CSS Referência
JavaScript Referência
SQL Referência
Python Referência
W3.CSS Referência
Bootstrap Referência
PHP
Cores HTML
Referência Java Referência
angular Referência
jQuery

Principais exemplos
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 »