SIMULADOS & TUTORIAIS

TUTORIAIS JAVASCRIPT

O que são objetos em JavaScript

JORGE EIDER

Copyright 2013 – Todos os Direitos Reservados
Jorge Eider F. da Silva

Proibida a reprodução deste documento no todo ou em parte por quaisquer meios, seja digital, eletrônico ou impresso sem a expressa autorização do autor por escrito. Os infratores serão punidos de acordo com a Lei.

O que são objetos em JavaScript

Introdução

Como já dissemos em um tutorial anterior, tudo em JavaScript é um objeto, uma String, um Number, uma Array, uma Function, etc. Além disso, o JavaScript permite também que você defina seus próprios objetos.

Os Objetos em JavaScript

O JavaScript dispõe de vários objetos pré-construídos, tais como String, Date, Array, entre outros. Um objeto é justamente um tipo especial de dado, com propriedades e métodos.

Como acessar as propriedades de um Objeto

Propriedades são valores associados aos objetos. A sintaxe correta para acessar a propriedade de um objeto é a seguinte:

```
nomeObjeto.nomePropriedade
```

Vejamos um exemplo:

```
var mensagem = "Olá, pessoal!";
var x = mensagem.length;
```

Após esse código ser executado, o valor de **x** será 12.

Como acessar os métodos de um Objeto

Métodos são as ações que podem ser executadas pelos objetos. Você pode chamar um método com a seguinte sixtaxe:

```
nomeObjeto.nomeMétodo()
```

Vejamos um exemplo:

```
var mensagem = "Olá, pessoal!";
var x = mensagem.toUpperCase();
```

Esse exemplo usa o método **toUpperCase()** do objeto **String** para converter o texto da variável mensagem para caracteres maiúsculos.

Após esse código ser executado, o valor de x será:

```
OLÁ, PESSOAL!
```

Criando objetos em JavaScript

Com JavaScript você pode definir e criar seus próprios objetos. Há duas maneiras diferentes de criar um novo objeto:

JORGE EIDER

1. Definir e criar uma instância direta de um objeto.
2. Usar uma função para definir um objeto, e em seguida criar novas instâncias desses objetos.

Criando uma instância direta

O exemplo a seguir cria uma nova instância de um objeto, e em seguida adiciona quatro propriedades a ele:

```
pessoa = new Object();
pessoa.primeiroNome = "Jorge";
pessoa.ultimoNome = "Silva";
pessoa.idade = 50;
pessoa.peso = "70";
```

Vejamos um exemplo prático:

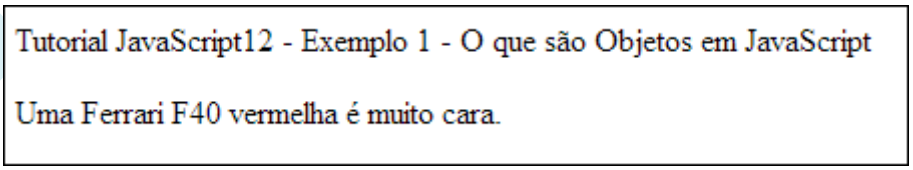
Exemplo 1

Nesse exemplo criamos um objeto e em seguida criamos uma nova instância e adicionamos três propriedades a ele. Para isso:

1. Digite o código abaixo no seu editor de texto e salve-o como: **js12-exemplo1.html**.

```
<!DOCTYPE html>
<html>
Tutorial JavaScript12 - Exemplo 1 - O que são Objetos em JavaScript
<head>
   <title>Tutorial JavaScript12 - Exemplo 1 - O que são Objetos em JavaS-
cript</title><p>
<script language="javascript">
var carro = new Object();
carro.modelo = "F40";
carro.fabricante = "Ferrari";
carro.cor = "vermelha";
document.write("Uma " + carro.fabricante + " " + carro.modelo + " " + carro.cor + "
é muito cara.");
</script>
</head>
<body>
</body>
</html>
```

2. O resultado desse código após executado será o seguinte:



Uma sintaxe alternativa para usar objetos literais podemos observar no exemplo a seguir:

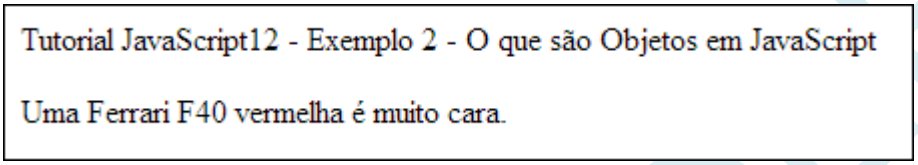
Exemplo 2

Nesse exemplo mostramos como criar uma alternativa para o mesmo caso anterior. Para isso:

1. Digite o código abaixo no seu editor de texto e salve-o como: **js12-exemplo2.html**.

```
<!DOCTYPE html>
<html>
<body>
<script>
carro={modelo:"F40", fabricante:"Ferrari", cor: "vermelha"}
document.write("Uma " + carro.fabricante + " " + carro.modelo + " " + carro.cor + "
é muito cara.");
</script>
</body>
</html>
```

2. O resultado desse código após executado será o mesmo do exemplo anterior, ou seja:



Usando um Objeto Construtor

O exemplo a seguir utiliza uma função para construir um objeto:

```
function carro(fabricante, modelo, cor)
{
this.fabricante = fabricante;
this.modelo = modelo;
this.cor = cor;
}
```

Vejamos essa função em um exemplo prático:

Exemplo 3

Nesse exemplo mostramos como criar um objeto a partir de uma função. Para isso:

1. Digite o código abaixo no seu editor de texto e salve-o como: **js12-exemplo3.html**.

```
<!DOCTYPE html>
<html>
<body>
<script>
function carro(fabricante, modelo, cor)
{
this.fabricante = fabricante;
this.modelo = modelo;
this.cor = cor;
}
meuCarro = new carro("F40", "Ferrari", "vermelha");
document.write("Uma " + meuCarro.fabricante + " " + meuCarro.modelo + " " +
meuCarro.cor + "  é muito cara.");
</script>
</body>
</html>
```

2. Da mesma forma que os exemplos anteriores, o resultado será o mesmo.

Criando instâncias de Objetos em JavaScript

Uma vez criado o objeto construtor, você poderá criar várias instâncias desse mesmo objeto, conforme mostram os exemplos a seguir:

```
var meuCarro = new carro("F40", "Ferrari", "vermelha");
var meuCarro = new carro("SE500", "Mercedes", "prata");
var meuCarro = new carro("Diablo", "Lamborghini", "amarela");
```

Adicionando propriedades aos Objetos em JavaScript

Você pode adicionar novas propriedades a um objeto existente simplesmente dando a elas um valor. Por exemplo, suponha que o objeto **carro** já exista, então você poderá criar novas propriedades para ele tais como:

```
carro.fabricante = "Mercedes";
carro.modelo = "SE500";
carro.cor = "prata";
carro.ano = "2012";
```

E depois criar uma variável para armazenar uma dessas propriedades, como por exemplo:

```
var x = carro.fabricante;
```

O resultado de **x** seria:

```
Mercedes
```

Adicionando métodos aos Objetos em JavaScript

Métodos são exatamente funções adicionadas aos objetos. A definição de método para um objeto é feita dentro da função construtora. Por exemplo:

```
function perssoa(primeitoNome, ultimoNome, idade)
{
this.primeitoNome = primeitoNome;
this.ultimoNome = ultimoNome;
this.idade = idade;
//
this.alteraNome = alteraNome;
function alteraNome (nome)
{
this.ultimoNome = nome;
}
}
```

A função **alteraNome** atribui o valor de **nome** à propriedade **ultimoNome** do objeto **pessoa**.

Vejamos essa função em um exemplo prático:

Exemplo 4

Nesse exemplo mostramos como adicionar um método **alteraNome**. Para isso:

1. Digite o código abaixo no seu editor de texto e salve-o como: **js12-exemplo4.html**.

```
<!DOCTYPE html>
<html>
<body>
<script>
function pessoa(primeiroNome, ultimoNome, idade)
{
this.primeiroNome = primeiroNome;
this.ultimoNome = ultimoNome;
this.idade = idade;
//
this.alteraNome = alteraNome;
function alteraNome (nome)
{
this.ultimoNome = nome;
}
}
minhaMae = new pessoa("Maria","Silva",70);
minhaMae.alteraNome("Isaura");
document.write(minhaMae.ultimoNome);
</script>
</body>
</html>
```

2. Esse código depois de executado terá o seguinte resultado:

```
Isaura
```

Classes em JavaScript

Apesar de JavaScript ser uma linguagem orientada a objetos, ela não usa classes. Em JavaScript você não define classes e cria objetos a partir dessas classes, como acontece na maioria das linguaagens orientadas a objetos. JavaScript é baseada em protótipos e não em classes. Por exemplo:

```
function perssoa(primeitoNome, ultimoNome, idade)
{
this.primeitoNome = primeitoNome;
this.ultimoNome = ultimoNome;
this.idade = idade;
//
this.alteraNome = alteraNome;
function alteraNome (nome)
```

A função **alteraNome** atribui o valor de **nome** à propriedade **ultimoNome** do objeto **pessoa**.

*****

Exercícios de fixação

1) Apesar de JavaScript ser uma linguagem orientada a objetos, ela não usa classes. Essa declaração é:

   a) Verdadeira
   b) Falsa

2) Qual das alternativas a seguir não é um objeto JavaScript?

   a) String
   b) Number
   c) Array
   d) Name

3) Métodos são exatamente funções adicionadas aos objetos. Essa declaração é:

   a) Verdadeira
   b) Falsa

4) Qual das declarações abaixo está sintaticamente incorreta na criação de um instância?

   a) var meuCarro = new carro("F40"; "Ferrari", "vermelha");
   b) var meuCarro = new carro["F40", "Ferrari", "vermelha"];
   c) var meuCarro = new carro{"F40", "Ferrari"; "vermelha"};
   d) var meuCarro = new carro("F40", "Ferrari", "vermelha");

5) A sintaxe correta para acessar a propriedade de um objeto é a seguinte:

   a) nomeObjeto.nomePropriedade[]
   b) nomeObjeto:nomePropriedade()
   c) nomeObjeto.nomePropriedade
   d) nomeObjeto.[nomePropriedade]

*****

JORGE EDER