JavaScript

IFRN – INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIAS E TECNOLOGIAS DO RIO GRANDE DO NORTE

Javascript

Objetos

Objeto é um tipo de dado constituído por uma coleção de dados, ou seja, é uma unidade que armazena dados formatados em pares nome/valor.

Segundo uma definição formal, podemos dizer que objeto é uma coleção não ordenada de propriedades e métodos constituída por pares nome/valor.

Os valores do par nome/valor podem ser tanto valores **primitivos** (números e strings) como outros **objetos**.

Javascript

Criando Objetos

Para criar um objeto vazio (sem propriedades), a sintaxe é:

```
var livro = new Object();
```

Identificamos a varável de nome **livro**, atribuída ao objeto a criar e o operador ***new*** seguido de uma função a qual denominamos **construtor** que inicia o objeto.

O construtor **Object()** é nativo da linguagem JavaScript e cria um objeto genérico vazio.

Javascript

Tabela 3 contendo alguns dados sobre um livro publicado pela editora Novatec.

1	Título	AJAX com Jquery
2	Autor	Maurício Samy Silva
3	Páginas	328
4	Preço	R$69,00
5	Frete	A calcular
6	Capítulo 1	Revisão do AJAX
7	Capitulo 2	Funções para requisições AJAX
8	Capitulo 3	Eventos e miscelânea
9	Capitulo 4	Requisições XML
10	Capitulo 5	Introdução ao formato JSON
11	Capitulo 6	Requisições JSON

Javascript

Sintaxe formal

Para definir uma propriedade do objeto livro denominada titulo, cujo valor seja a string "AJAX com jQuery", usamos a sintaxe mostada a seguir:

```
livro.titulo = "AJAX com jQuery";
```

Javascript

Observe a sintaxe para criar o objeto livro da tabela 3.:

```
var livro = new Object();
livro.titulo = "AJAX com jQuery";
livro.autor = "Maurício Samy Silva";
livro.paginas = 328;
livro.preco = "R$69,00";
livro.freteSedex = function (ceporigem, cepdestino, peso) {
    var valorFrete = "";
    //script de cálculo do frete aqui
    return valorFrete;
}
livro.capitulo1 = "Revisão do AJAX";
livro.capitulo2 = "Funções para requisições AJAX";
livro.capitulo3 = "Eventos e miscelanea";
livro.capitulo4 = "Requisições XML";
livro.capitulo5 = "Introdução ao formato JSON";
livro.capitulo6 = "Requisições JSON";
```

Javascript

Objetos

No exemplo anterior, criamos um objeto livro com dez propriedades e um método.

Com a sintaxe mostrada, podemos a qualquer momento acrescentar novas propriedades ou métodos ou alterar os já existentes. Para criar uma nova propriedade, basta declarar o seguinte:

livro.novaPropriedade = valor da nova propriedade;

Javascript

Recuperando valores de propriedades e métodos de um objeto

```
var nomeAutor = livro.autor;
var capituloCinco = livro.capitulo5;
var valorDoFrete = livro.freteSedex(59015410,69190120, 1060);

//Exibindo o resultado
Alert ("Autor: " + nomeAutor + "\nCap5: " + capituloCinco + "\nValor frete: " + valorDoFrete);
```

Há uma sintaxe alternativa para recuperar os valores de propriedades de objetos que usa colchetes ([]) no lugar do operador .(ponto) conforme mostrado a seguir:

```
var nomeAutor = livro[autor];
var capituloCinco = livro[capitulo5];
```

Como exercício. Faça um script JavaScript que, use o loop for/in para retornar todos os pares nome/valor do nosso objeto livro.

Javascript


```
<script type="text/javascript">
        var livro = new Object();
        livro.titulo="AJAX com jQuery";
        livro.autor = "Maurício Samy Silva";
        livro.paginas = 328;
        livro.preco = "R$69,00";
        livro.freteSedex = function (ceporigem, cepdestino,peso) {
            var valorFrete = "";
            //script paracalcular o frete aqui
            return valorFrete;
        };
        livro.capitulo1 = "Revisão do AJAX";
        livro.capitulo2 = "Funções pararequisições AJAX";
        livro.capitulo3 = "Eventos e miscelanea";
        livro.capitulo4 = "Requisições XML";
        livro.capitulo5 = "Introdução ao formato JSON";
        livro.capitulo6 = "Requisições JSON";

        //Exibindo objeto com for/in
        var pares = "";
        for (var prop in livro) {
            pares += prop + ": " + livro[prop] + "\n";
        };
        alert(pares);
</script>
```

Javascript

Como já foi dito, um objeto é um conjunto de dados, então pode conter outros objetos como dados, ou seja, objetos podem ser aninhados. Vamos exemplificar o conceito de objetos aninhados usando uma alternativa para criar nosso objeto livro.

Vamos supor que os capítulos do livro devam estar contidos em um objeto chamado capítulos. A definição do objeto capitulo a ser aninhada ao objeto livro se faz como mostrado a seguir:

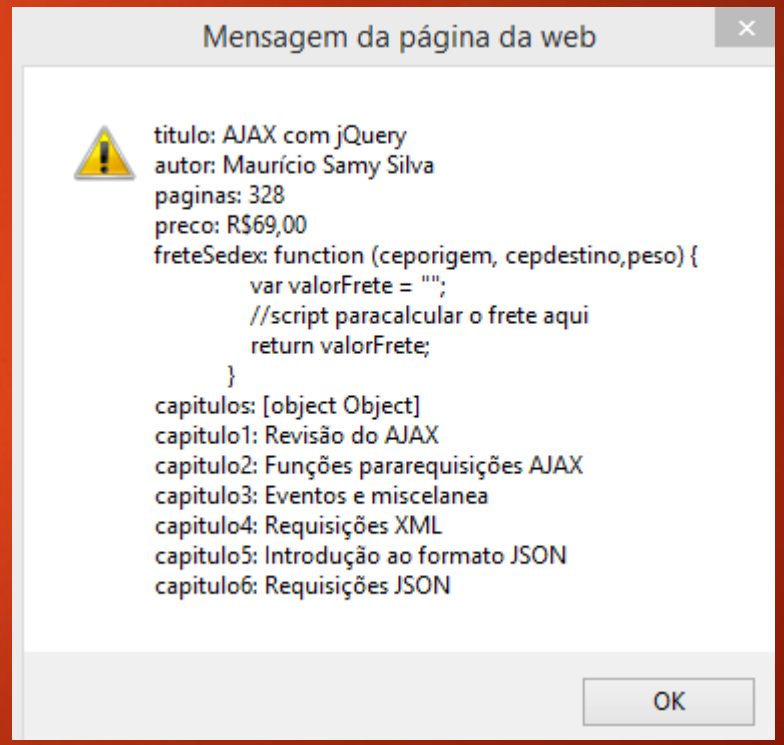
```
var livro.capítulos = new Object();
livro.capítulos.capitulo1 = "Revisão do AJAX";
livro.capítulos.capitulo2 = "Funções para requisições AJAX";
...
livro.capítulos.capitulo10 = "Requisições JSON";
```

Javascript

Exercício 1:

1. modifique o objeto livro, de forma que, os capítulos façam parte de um objeto capítulos.
2. Exiba em uma tela (alert()), todas os membros do livro, inclusive os capítulos.

resultado



Javascript


```
<script type="text/javascript">
        var livro = new Object();
        livro.titulo="AJAX com jQuery";
        livro.autor = "Maurício Samy Silva";
        livro.paginas = 328;
        livro.preco = "R$69,00";
        livro.freteSedex = function (ceporigem, cepdestino,peso) {
            var valorFrete = "";
            //script paracalcular o frete aqui
            return valorFrete;
        };
        livro.capitulos = new Object();
        livro.capitulos.capitulo1 = "Revisão do AJAX";
        livro.capitulos.capitulo2 = "Funções pararequisições AJAX";
        livro.capitulos.capitulo3 = "Eventos e miscelanea";
        livro.capitulos.capitulo4 = "Requisições XML";
        livro.capitulos.capitulo5 = "Introdução ao formato JSON";
        livro.capitulos.capitulo6 = "Requisições JSON";

        alert(msg);
...
```

Javascript

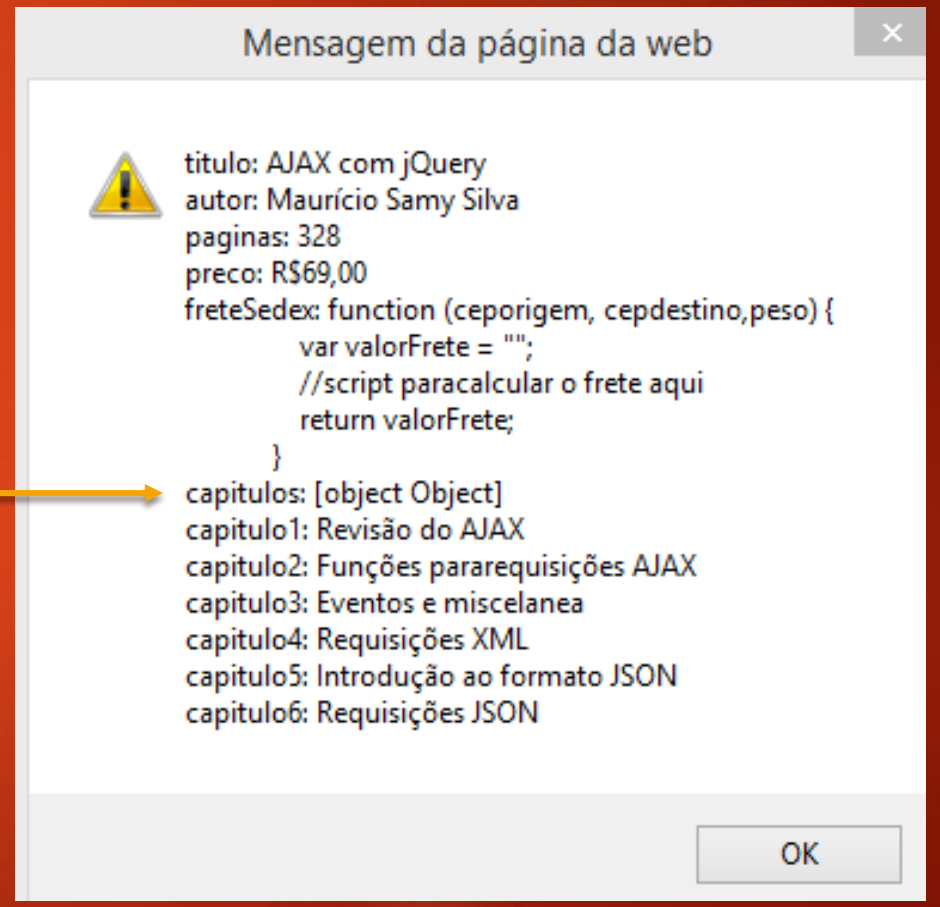

```
//Exibindo objeto com for/in
        var msg = "";
       for (var prop in livro) {
          msg += prop + ": " + livro[prop]+"\n";
          if (typeof livro[prop] == "object"){
             for (var cap in livro[prop]){
                msg += cap + ": " + livro[prop][cap] + "\n";
             }
          }
       }

        alert(msg);

</script>
```

Javascript

Exercício 2:

1. O que devo fazer para eliminar a linha 9 do resultado. Veja a figura abaixo.

Linha 9



Javascript

Basta colocar o if antes da varaível msg.

```
//Exibindo objeto com for/in
var msg = "";
for (var prop in livro) {
	if (typeof livro[prop] == "object"){
		for (var cap in livro[prop]){
			msg += cap + ": " + livro[prop][cap] + "\n";
		};
	} else {
			msg += prop + ": " + livro[prop]+"\n";
	};
 }

alert(msg);
```

Javascript

Construtor

Nos exemplos anteriores, criamos o objeto livro como o uso do operador ***new*** e de uma função construtora denominada ***Object()***.

Denomina-se função construtora ou simplesmente construtor uma função capaz de criar objetos. Não estamos limitados a criar objetos como o uso de construtores nativos, pois podemos criar nossos construtores pernsonalizados.

Javascript

Construtor

Vamos supor que em nosso script precisamos manipular cilindros e resolvemos criar um construtor para gerar cilindros.

```
1. function Cilindro(r,h) {
2.       this.raioBase = r;
3.       this.altura = h;
4. };
5. cilindroUm = new Cilindro(2,5);
6. alert("Raio da base: " +  cilindroUm.raioBase + "\nAltura: " + cilindroUm.altura);
```

Linha 1: função construtora Cilindro que adminite dois argumentos (raio=r, e altura = h)
Linha 2: a palavra-chave nativa da linguagem, this que faz referencia ao novo objeto criado
Linha 5: com o construtor Cilindro cria-se novo objeto, denominado cilindroUM, com raio=2, e altura=5

Javascript

Criar um construtor para cilindros com a finalidade de retornar os argumentos passados à função não tem valor prático algum. Vamos melhorar nosso construtor acrescentando alguns métodos.

Conhecendo-se o raio da base (r) e a altura (h) de um cilindro circular reto, é possível determinar as seguintes grandezas do cilindro:

1. Área da base: área do circulo de raio r: areaBase = πr^2
2. Área lateral: área da superfície lateral: areaLateral = $2\pi rh$
3. Área total: soma da área lateal com a área das bases: areaTotal = $2\pi rh + 2\pi r^2$
4. Volume: área da base vezes a altura: volume = $\pi r^2 h$

Como exercício, crie métodos para o construtor Cilindro capazes de retornar as grandezas listadas.

Javascript


```
<script type="text/javascript">
        function Cilindro (r,h) {
            this.raioBase = r;
            this.altura = h;
            this.areaBase = function calculaAreaBase() {
                aBase = Math.PI * Math.pow(this.raioBase,2);
                return aBase;
            };
            this.areaLateral = function calculaAreaLateral(){
                aLateral = 2 * Math.PI * this.raioBase * this.altura;
                return aLateral;
            };
            this.areaTotal = function calculaAreaTotal(){
                return 2 * aBase + aLateral;
            };
            this.volume = function calculaVoluma(){
                return aBase * this.altura;
            };
        }
        cilindroUm = new Cilindro(3,10);
            alert("Raio da base: " + cilindroUm.raioBase + "\nAltura: " + cilindroUm.altura +
                    "\nÁrea Base: " + cilindroUm.areaBase() + "\nÁrea Lateral: " + cilindroUm.areaLateral() +
                    "\nÁrea total: " + cilindroUm.areaTotal() + "\nVolume: " + cilindroUm.volume());
</script>
```

Javascript

Categorias de objetos

Os objetos da linguagem podem ser agrupados em três categorias, a saber:

- **Objetos nativos**: aqueles próprios da linguagem;
- **Objetos do ambiente de hospedagem**: aqueles próprios do dispositivo que interpreta a libguagem (um navegador gráfico, por exemplo);
- **Objetos customizados**: aqueles criados pelo desenvolvedor.