

Desenvolvimento Web - JavaScript

Profº Ed Wilson Jr

Incorporando JavaScript em HTML

Existem quatro maneiras de incorporar JavaScript em HTML. Vamos abordar as três primeiras, por ser mais comum:

- Entre as tags <script></script> dentro do código HTML.
- A partir de um arquivo externo.
- Em atributos de tratamento de eventos HTML.
- Em uma URL que use o protocolo especial javascript

O elemento <script>

O código JavaScript pode estar dentro do código HTML, entre as marcações <script></script>. Observe a sintaxe abaixo:

```
<script>
  // Seu código JavaScript aqui
</script>
```

Script em arquivo externo

A marcação <script> suporta um atributo src, que especifica a URL de um arquivo contendo código JavaScript. Por convenção, os arquivos JavaScript têm nomes que terminam com a extensão .js.

A sintaxe é:

```
<script src="nome e caminho do arquivo"></script>
```

Exemplo 01:

No projeto, em NetBeans, crie um novo arquivo window.html com o seguinte conteúdo:

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8">
  <title>Cursos - JavaScript Básico</title>
  <script>
```

```
</script>
</head>
<body>

  <p>Curso de JavaScript Básico - Objeto Window</p>

</body>
</html>
```

Interação com o usuário

Para exibir uma caixa de alerta, utilize o método `window.alert()`. Não é necessário usar a palavra `window` para utilizar as funções do objeto `Window`. Assim `.Alert()`, também irá funcionar. Inclua o código abaixo entre as tags `<script></script>` no código de exemplo criando anteriormente.

```
Var nome = "Treinamento JavaScript";
Alert("Esta é uma mensagem do curso " + nome );
```

Para incluir botões "OK" e "Cancelar" nas mensagens, utilize o método **Confirm()**. Ele retorna `true` se for clicado "OK" e `false`, caso contrário.

Exemplo 02:

```
var pergunta = confirm("Você está gostando do curso?");
if(pergunta == true) {
  alert("Ficamos imensamente felizes que você esteja gostando!");
} else {
  alert("Poxa! Em que podemos melhorar?");
}
```

Para interagir com o usuário, tem outro método interessante: a função `prompt()`. Com ela, é possível solicitar informações ao usuário.

Exemplo 03:

```
var nomeUsuario = prompt("Qual o seu nome?", "Nome");
if(nomeUsuario != "") {
  alert("Seu nome é: "+nomeUsuario+"!");
} else {
```

```
    alert("Pôxa, nem sei qual é seu nome =/");  
}
```

Temporizador

O método `setTimeout()` agenda a execução de uma função, decorrido o tempo especificado em milissegundos.

Exemplo 04:

```
setTimeout('bemVindo()',5000);  
  
function bemVindo(){  
    alert("Seja bem vindo!");  
}
```

Além, do `setTimeout()`, há outro método `setInterval()`. Ele funciona de forma bem parecida com `setTimeout()`. A diferença é que ele irá repetir o código a cada intervalo de tempo informado, ou até que o método `clearInterval()`, seja executado.

Exemplo 05:

```
var cont = 0;  
  
var t = setInterval('tw()',5000);  
  
function tw() {  
    cont++;  
    alert("O contador está em: "+cont);  
  
    if(cont == 5){  
        alert("Fim da contagem!");  
        clearInterval(t);  
    }  
}
```

Manipulando Janela

O método `Open()` pode ser usado para abrir uma nova janela do navegador.

Exemplo 06:

```
function abrirJanela(){
    open("http://www.treinaweb.com.br");
}
```

Para o teste será utilizado o evento onClick em um botão HTML.

```
<input type="button" value="Testes com Janelas" onclick="abrirJanela();" />
```

Introdução ao Objeto Document

Descrevendo-se o objeto Document de forma mais simples, pode-se dizer que ele serve para consultar ou modificar os elementos HTML da página carregada.

Para analisar o objeto Documento, crie o arquivo para teste chamado documento.html e defina para ele o seguinte código:

Documento.html

```
<!DOCTYPE html>

<html>

<head>
    <meta charset="utf-8">
    <title>Escola Adalgisa de Barros - JavaScript Básico</title>
    <script>
    </script>
</head>

<body>

<strong>Escola Adalgisa de Barros - Curso de JavaScript Básico - Objeto 'Document'.</strong>

<p><input type="button" value="Testes com o documento" onclick="manipularElementos();"
/></p>

<form name="frmCadastro" id="frmCadastro">

<fieldset>

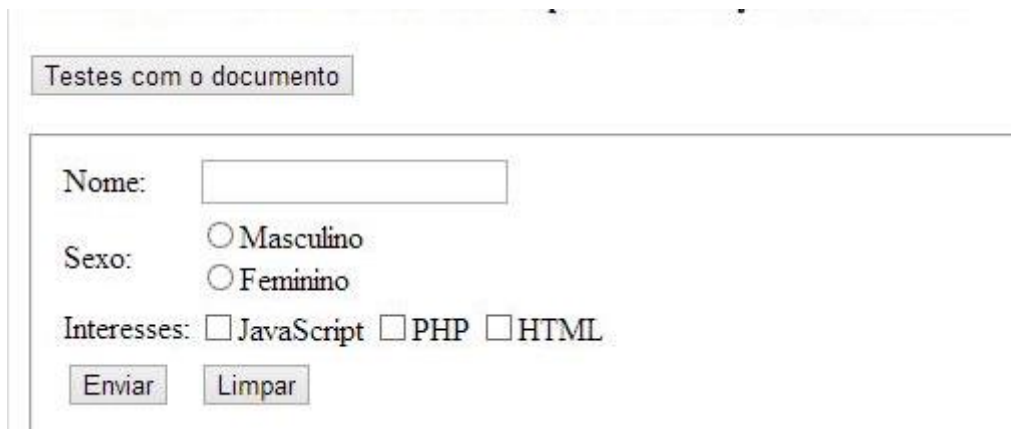
    <table>
```

```
<tr>
  <td>Nome:</td>
  <td><input type="text" name="nome" id="nome"
/></td> </tr>
<tr>
  <td>Sexo:</td>
  <td>
    <input type="radio" name="sexo" value="M">Masculino<br
    /> <input type="radio" name="sexo" value="F">Feminino
  </td>
<tr>
  <td>Interesses:</td>
  <td>
    <input type="checkbox" name="interesses" id="javascript" /><label
for="javascript">JavaScript</label>
    <input type="checkbox" name="interesses" id="php" /><label for="php">PHP</label>
    <input type="checkbox" name="interesses" id="html" /><label
for="html">HTML</label>
  </td>
<tr>
  <td>
    <input type="button" id="enviar" name="enviar"
value="Enviar"/> </td>
  <td>
    <input type="reset" id="limpar" name="limpar"
value="Limpar"/> </td>
</tr>
</table>
</fieldset>
</form>

</body>
```

</html>

salve o arquivo e abra-o no navegador.



Testes com o documento

Nome:

Sexo: Masculino
 Feminino

Interesses: JavaScript PHP HTML

Este é um formulário básico apenas para aplicar os testes.

Principais Propriedades

Forms[]

Esta propriedade retorna os formulários do documento HTML e os elementos que estão contidos nele. Para entender melhor, insira o código a seguir entre as tags <script></script> no código anterior.

```
function manipularElementos() {  
    alert(document.forms);  
}
```

Salve e execute novamente.

Foi mostrado um alerta informando que você está tentando exibir um objeto. Lembre-se que você está trabalhando com objetos e a maioria deles também possui propriedades que, em sua maioria, são comuns a todos os objetos. Utilize length para saber a quantidade de forms contidos no código HTML.

```
function manipularElementos() {  
    alert(document.forms.length);  
}
```

Salve e execute.

Outra informação importante, e que você deve lembrar, é sobre acesso aos forms. Ele funciona como um array e, como todos os arrays, possui um índice que começa com 0 (zero). Então, quando a propriedade length informa que há 1 form, ele está na posição 0 (zero). Então, para saber o nome de form de índice 0, use a propriedade name do form.

```
function manipularElementos() {  
    alert(document.forms[0].name);  
}
```

Salve e execute o projeto.

Title

Informa o título da página HTML atual.

```
function manipularElementos() {  
    alert(document.title);  
}
```

URL

Retorna o URL completa da página atual.

```
function manipularElementos() {  
    alert(document.URL);  
}
```

Principais Métodos

O método write() é utilizado para imprimir uma informação na tela da janela atual.

```
function manipularElementos() {  
    document.write("Inserindo texto no documento.");  
}
```

Writeln()

Funciona da mesma forma que write(). Porém insere uma quebra de linha.

```
function manipularElementos() {  
    document.write("<pre>");  
    document.write("Inserindo texto no documento com write().");  
    document.write("Linha 1.");  
}
```

```
document.write("Linha 2.");
document.write("Linha 3.");
document.write("</pre>");
document.writeln("<pre>");

document.writeln("Inserindo texto no documento com writeln().");
document.writeln("Linha 1.");

document.writeln("Linha 2.");
document.writeln("Linha 3.");
document.writeln("</pre>");
}
```

getElementById()

Este método encontra e retorna um objeto referente ao elemento com determinado ID. Por exemplo, se o objeto for um campo input, ele possui a propriedade value que retorna o seu valor.

```
function manipularElementos() {
    var nome = document.getElementById("nome");
    alert("O nome é: " + nome.value);
}
```

Com o método getElementById é possível realizar diversas operações com o elemento como, por exemplo, alterar sua formatação, alterar seu conteúdo, movê-lo pela tela e etc. Ele é útil para manipular elementos com JavaScript.

```
function manipularElementos() {
    document.getElementById("nome").style.color = "red";
    document.getElementById("enviar").style.backgroundColor = "green";
    document.getElementById("limpar").style.fontStyle = "italic";
}
```

Salve e atualize o navegador. Digite um nome e clique no botão **Teste com documento**. Veja o resultado.

getElementsByName()

getElementsByName() tem funcionalidade semelhante a getElementById(), com a diferença que ele encontra e retorna determinado elemento pelo Name.

```
function manipularElementos() {  
  
    var nome = document.getElementsByName("sexo");  
  
    alert("O total de elementos com o nome 'sexo' é: " + nome.length);  
  
}
```

Salve e atualize o navegador e teclle no botão **Testes com documento**.

Exemplo 2

```
function manipularElementos() {  
  
    // Seleciona os elementos e os armazena em variáveis  
    var nome = document.getElementById("nome");  
    var sexo = document.getElementsByName("sexo");  
  
    // Cria a variável que armazenará as informações e já a inicializa com o nome  
    var informacoes = "Seu nome: " + nome.value;  
  
    // Acrescenta o sexo  
    if( sexo[0].checked )  
        informacoes += "\nO seu sexo é: Masculino.";  
    else if( sexo[1].checked )  
        informacoes += "\nO seu sexo é: Feminino.";  
  
    // Exibe um alerta com as informações  
    obtidas alert(informacoes);  
  
}
```

Salve e atualize o navegador e teclle no botão **Testes com documento**.

Introdução ao tratamento de eventos

As propriedades de manipulação de eventos(handles) permite que os scripts especifique funções que devem ser chamadas de forma assíncrona, quando certos eventos ocorrem. Os mecanismos de tratamento de eventos permitem ao JavaScript alterar o comportamento de

janelas, de documentos e de elementos que compõem esses documentos. As propriedades de tratamento de evento têm nomes que começam com **on**.

O exemplo a seguir utiliza o evento **onClick**.

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8">
  <title>JavaScript Básico</title> <script>
    function executarAcao(acao) {
      if(acao === 'click') {
        alert("O botão Enviar foi clicado!"); }
      else if(acao === 'change') {
        alert("A caixa de texto foi modificada!");
      }
    }
  </script>
</head>
<body>
  <p><input type="text" onchange="executarAcao('change');" value="Modifique o Texto"
  /></p>
  <p><input type="button" onclick="executarAcao('click');" value="Enviar" /></p>
</body>
</html>
```

Salve e execute o arquivo.

