

Formulários

Em geral

Com o Javascript, os formulários Html tomam outra dimensão. Não esqueça que em Javascript, podemos acessar cada elemento de um formulário para, por exemplo, ler ou escrever um valor, uma escolha a qual poderá associar um procedimento de evento... Todos esses elementos irão reforçar as capacidades interativas da página.

Um formulário é um elemento Html declarado pela tags **<FORM></FORM>**. Um formulário contém um ou vários elementos que chamamos os campos (widgets). Esses campos são escrito por exemplo pela tag **<INPUT TYPE= ...>**.

Declaração de um formulário

A declaração de um formulário faz-se pelas tags **<FORM>** e **</FORM>**. Pode-se notar que em Javascript, o atributo **NAME="nome_do_formulário"** tem toda a sua importância para designar o caminho completo dos elementos. Em suma, os atributos **ACTION** e **METHOD** são facultativos pelo menos até não fazerem chamadas ao servidor.

Um erro clássico em Javascript é de esquecer de declarar o fim do formulário **</FORM>** depois de ter inserido um campo.

Controlador de linha de texto

A caixa de texto é o elemento de entrada/saída por padrão do Javascript. A sintaxe Html é **<INPUT TYPE="text" NAME="nome" SIZE=x MAXLENGTH=y>** para uma área de texto de uma só linha, de comprimento x e de comprimento máximo y.

O objeto texto tem 3 propriedades:

PROPRIEDADES	DESCRIÇÃO
name	indica o nome do campo pelo qual podemos acessar.
defaultvalue	indica o valor por padrão que será afixada na caixa de texto.
value	indica que o valor da caixa de texto. Seja ele escrito pelo usuário ou já ter um valor por padrão.

Ler um valor numa caixa de texto

Um exemplo que iremos utilizar:

Insira um valor e carregue no botão para controla-lo.

O script completo é assim:

```
<HTML>  
<HEAD>  
<SCRIPT LANGUAGE="javascript">
```

```

function controle(form1) {
var test = document.form1.input.value;
alert("Escreveste: " + test);
}
</SCRIPT>
</HEAD>
<BODY>
<FORM NAME="form1">
<INPUT TYPE="text" NAME="input" VALUE=""><BR>
<INPUT TYPE="button" NAME="botao" VALUE="Controlar" onClick="controle(form1)">
</FORM>
</BODY>
</HTML>

```

Quando clica-se no botão "controlar", o Javascript chama a função **controle()** na qual passamos o formulário em que o nome é **form1** como argumento. Esta função **controle()**, definida nas tags **<HEAD>**, toma sob a variável **test**, o valor da caixa de texto. Para acessar a este valor, escreve-se o caminho completo deste (ver o capítulo "**Um pouco de teoria objeto**"). No documento presente, temos o objeto formulário chamado form1 que contém o campo de texto nomeado input e que tem como propriedade o elemento de valor **value**. O que dá **document.form1.input.value**.

Escrever um valor na caixa de texto

Inserir um valor qualquer na caixa de texto de entrada. Clicar sobre o botão para inserir este valor na caixa de texto saída.

caixa de texto de entrada

caixa de texto de saída

Código:

```

<HTML>
<HEAD>
<SCRIPT LANGUAGE="javascript">
function escrever(form2) {
var testin =document. form2.input.value;
document.form2.output.value=testin
}
</SCRIPT>
</HEAD>
<BODY>
<FORM NAME="form2">
<INPUT TYPE="text" NAME="input" VALUE=""> caixa de texto de entrada<BR>
<INPUT TYPE="button" NAME="botao" VALUE="Escrever" onClick="escrever(form2)"><BR>
<INPUT TYPE="text" NAME="output" VALUE=""> caixa de texto de saída
</FORM>
</BODY>
</HTML>

```

Ao clicar no botão "Escrever", o Javascript chama a função **escrever()** na qual passa o formulário em que o nome é **form2** desta vez como argumento.

Esta função escrever() definida nas tags <HEAD> toma sob a variável **testin**, o valor da caixa de texto de entrada. Na instrução seguinte, dizemos ao Javascript que o valor da caixa de texto **output** contida no formulário nomeado **form2** é a da variável **testin**. Utiliza-se o caminho completo para chegar a propriedade valor do objeto desejado, seja em Javascript **document.form2.output.value**.

Os botões radio

Os botões radio são utilizados para escolher uma escolha, e só uma, entre um conjunto.

PROPRIEDADES	DESCRIÇÃO
name	indica o nome do campo. Todos os botões tem o mesmo nome.
index	o index ou o conjunto de botões radio começando por 0.
checked	indica o estado em curso do elemento radio
defaultchecked	indica o estado do botão seleccionado por padrão.
value	indica o valor do elemento radio.

Exemplo :

```
<HTML>
<HEAD>
<SCRIPT language="javascript">
function escolhaprop(form3) {
if (form3.escolha[0].checked) { alert("Escolheu a opção " + form3.escolha[0].value) };
if (form3.escolha[1].checked) { alert("Escolheu a opção " + form3.escolha[1].value) };
if (form3.escolha[2].checked) { alert("Escolheu a opção " + form3.escolha[2].value) };
}
</SCRIPT>
</HEAD>
<BODY>
Escolha :
<FORM NAME="form3">
<INPUT TYPE="radio" NAME="escolha" VALUE="1">Escolha número 1<BR>
<INPUT TYPE="radio" NAME="escolha" VALUE="2">Escolha número 2<BR>
<INPUT TYPE="radio" NAME="escolha" VALUE="3">Escolha número 3<BR>
<INPUT TYPE="button"NAME="but" VALUE="Qual é sua escolha ?"
onClick="escolhaprop(form3)">
</FORM>
</BODY>
</HTML>
```

Clique na sua escolha :

- Escolha número 1
- Escolha número 2
- Escolha número 3

No formulário nomeado **form3**, declara-se 3 botões radio. Note que utiliza-se o mesmo nome para

os três botões. Em seguida temos um botão que invoca a função escolhaprop().

Esta função testa qual o botão radio é checkado. Acessa-se os botões na forma de índice em relação ao nome dos botões radio que seja **escolha[0], escolha[1], escolha[2]**.

Testa-se a propriedade **checked** do botão por **if(form3.escolha[x].checked)**. Depois de escolher e carregar no botão "Qual a sua escolha", uma caixa de aviso aparece com a indicação do botão checkado. Esta mensagem toma o valor ligado a cada botão pelo caminho **form3.escolha[x].value**.

Checkbox

Os botões checkbox são úteis para escolher uma ou várias opções (lembre-se que os botões radio são utilizados para escolher só uma opção). Sua sintaxe e seu uso é bastante semelhante aos botões radio, exceto em relação ao atributo **name**.

PROPRIEDADES	DESCRIÇÃO
name	indica o nome do campo. Todos os botões tem o mesmo nome.
checked	indica o estado em curso do elemento radio
defaultchecked	indica o estado do botão seleccionado por padrão.
value	indica o valor do elemento radio.

Check as suas escolhas :
escolha os números 1,2 e 4 para obter uma boa resposta.

- Escolha número 1
- Escolha número 2
- Escolha número 3
- Escolha número 4

```
<HTML>
<HEAD>
<script language="javascript">
function resposta(form4) {
if ( (form4.check1.checked) == true && (form4.check2.checked) == true &&
(form4.check3.checked) == false && (form4.check4.checked) == true)
{ alert("É a boa respostas! ") }
else
{alert("Errado! continua a tentar.")}
}
</SCRIPT>
</HEAD>
<BODY>
```

Check as suas escolhas:

```
<FORM NAME="form4">
<INPUT TYPE="CHECKBOX" NAME="check1" VALUE="1">Escolha número 1<BR>
<INPUT TYPE="CHECKBOX" NAME="check2" VALUE="2">Escolha número 2<BR>
```

```

<INPUT TYPE="CHECKBOX" NAME="check3" VALUE="3">Escolha número 3<BR>
<INPUT TYPE="CHECKBOX" NAME="check4" VALUE="4">Escolha número 4<BR>
<INPUT TYPE="button"NAME="but" VALUE="Corrigir" onClick="resposta(form4)">
</FORM>
</BODY>
</HTML>

```

No formulário nomeado form4, declara-se 4 checkbox. Utilizar um nome diferente para as quatro checkbox. Em seguida temos um botão que irá invocar a função resposta(). Esta função teste quais botões que foram checkados. Para obter a resposta correta, tem de se checkar o 1,2 e 4. Acessa-se aos botões utilizando de cada vez os seus nomes. Testa-se a propriedade **checked** do botão por (**form4.nome_da_botão.checked**). Ao clicar no botão "Corrigir", no caso de positiva, uma caixa de aviso aparece que indica que está certo, em caso de negativo a caixa de aviso convida a tentar de novo.

Lista de selecção

Os botões da checkbox são utilizados para escolher uma ou várias opções. Sua syntax e seu uso é semelhante aos do botão radio exceto em relação ao atributo **name**.

PROPRIEDADES	DESCRIÇÃO
name	indica o nome da lista.
length	indica o número de elementos da lista.
selectedIndex	indica o lugar a partir do 0 do elemento da lista que foi seleccionado pelo usuário.
defaultselected	indica o elemento da lista seleccionado por padrão.É aquele que aparece então na lista em primeiro.

Um pequeno exemplo :

escolha um elemento da lista:

```

<HTML>
<HEAD>
<script language="javascript">
function lista(form5) {
alert("O elemento " + (form5.list.selectedIndex + 1));
}
</SCRIPT>
</HEAD>
<BODY>
Escolha um elemento: : <FORM NAME="form5">
<SELECT NAME="list">
<OPTION VALUE="1">Elemento 1
<OPTION VALUE="2">Elemento 2
<OPTION VALUE="3">Elemento 3
</SELECT>

```

```
<INPUT TYPE="button"NAME="b" VALUE="Qual é o elemento escolhido?"
onClick="lista(form5)"> </FORM>
</BODY>
</HTML>
```

No formulário nomeado **form5**, declara-se uma lista de selecção pelas tags **<SELECT></SELECT>**. Entre estas duas tags, declara-se os diferentes elementos da lista pela tag **<OPTION>**. Depois temos um botão que invoca a função **lista()**. Este função testa qual a opção foi seleccionada. o caminho completo do elemento seleccionado é **form5.nome_da_lista.selectedIndex**. Como o **index** comece em 0, acrescenta-se 1 para encontrar a linha elemento.

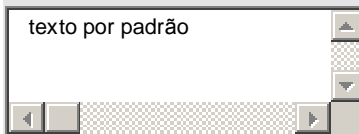
O campo textarea

O objeto textarea é uma caixa de texto de várias linhas.

A sintaxe Html é :

```
<FORM>
<TEXTAREA NAME="nome" ROWS=x COLS=y>
texto por padrão
</TEXTAREA>
</FORM>
```

onde **ROWS=x** representa o número de linhas e **COLS=y** representa o número de colunas.



O objeto textarea possui várias propriedades :

PROPRIEDADES	DESCRIÇÃO
name	indica o nome do campo pelo qual pode-se acessar.
defaultvalue	indica o valor por padrão que será afixado na caixa do texto.
value	indica o valor a decorrer na caixa de texto. Seja ela escrito pelo usuário ou se este não escreveu nada, o valor por padrão.

A estas propriedades, deve-se acrescentar o onFocus(), onBlur(), onSelect() e onChange().

Em Javascript, utiliza-se **\r\n** para quebra de linha.

Como por exemplo na expressão `document.Form.Text.value = 'Check\r\nthis\r\nout'`.

O campo Reset

O campo **Reset** permite de anular as modificações efetuadas aos campos de um formulário e de restaurar os valores por padrão.

A sintaxe Html é:

```
<INPUT TYPE="reset" NAME="nome" VALUE="texto">
```

Onde **VALUE** é o texto do botão.

Um único método associado ao campo **Reset**, é o método **onClick()**. O que pode servir, por exemplo, para escrever um outro valor do que o padrão.

O campo Submit

O campo tem a tarefa específica de transmitir todas as informações contidas no formulário a URL designada no atributo **ACTION** da tag **<FORM>**.

A sintaxe Html é:

```
<INPUT TYPE="submit" NAME="nome" VALUE="texto">
```

Onde **VALUE** é o texto do botão.

Um único método associado ao campo **Submit**, é o método **onClick()**.

O campo Hidden (oculto)

O campo **Hidden** permite entrar no script dos elementos (geralmente dados) que não aparecem na tela. Estes elementos estão ocultos. Daí seu nome.

A sintaxe Html é:

```
<INPUT TYPE="hidden" NAME="nome" VALUE="os dados ocultos ">
```

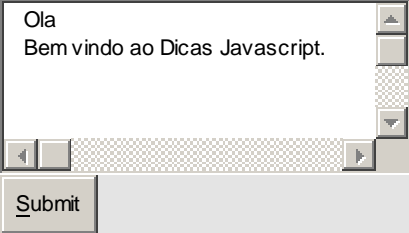
Envio de um formulário por E-mail

Javascript, não permite escrever num arquivo. Depois, o campo **Submit** é sobretudo destinado aos CGI. A solução é o envio do formulário via correio electrónico.

A sintaxe é :

```
<FORM METHOD="post" ACTION="mailto:alguem@hotmail.com">  
<INPUT TYPE=text NAME="nome"><br>  
<TEXTAREA NAME="mensagem" ROWS=2 COLS=35>  
</TEXTAREA><br>  
<INPUT TYPE=submit VALUE="Submit">  
</FORM>
```

O que dá:



Ola
Bem vindo ao Dicas Javascript.

Submit

E recebemos no e-mail, algo assim:

nome=Alguem&mensagem=Ola%0D%0ABem+vindo+ao+Dicas+Javascript.&submit=Submit Onde encontramos os campos **nome=** e **mensagem=** separados pelo símbolo **&**, os espaços são substituídos por **+** , e **%0D%0A** corresponde a uma quebra de linha.