Um pouco de tudo

Caixas de Diálogo ou de Mensagem

Javascript coloca a sua disposição 3 caixas de mensagens

- alert()
- prompt()
- confirm()

São os três métodos do objeto window.

O método alert()

O método **alert()** já lhe deve ser familiar, visto que já o utilizamos frequentemente ao longo do tutorial.

O método **alert()** escreve uma caixa de diálogo na qual é reproduzido o valor (variável e/ou cadeia de caracteres) do argumento que lhe foi fornecido. Esta caixa bloqueia o programa até que o usuário clicar em "OK".

Sua sintaxe é:

```
alert(variável);
alert("cadeia de caracteres");
alert(variável + "cadeia de caracteres");
```



Para escrever em várias linhas, deves usar: \n.

O método prompt()

No mesmo estilo do que o método alert(), o Javascript dispõe de uma outra caixa de diálogo, no caso presente é chamada caixa de convite, que é composto por um campo que contém uma entrada a completar pelo usuário. Esta entrada contém um valor por padrão.

A sintaxe é:

prompt("texto da caixa de convite","valor por padrão");



Ao clicar no OK, o método reenvia o valor escrito pelo usuário ou a resposta proposta por padrão. Se usuário clica em Cancel, o valor nulo é então reenviado.

Prompt() é porvez utilizado para incrementar dados fornecidos pelo usuário.

O método confirm()

Este método escreve 2 botões o "OK" e "Cancel". Ao clicar no OK, confirm() reenvia o valor true e obviamente false caso clicar em Cancel. Esta método é utilizado sobretudo para confirmar uma opção.



A sintaxe deste exemplo:

confirm("Deseja continuar ?")

Temporizador

Javascript dispõe de um temporizador (ou mais precisamente uma contagem decrescente) que permita invocar uma função após um derterminado tempo. A sintaxe do temporizador é :

nome_do_contador = setTimeout("função_invocada()", tempo em milisegundos)

Assim, setTimeout("arrancar()",5000) vai chamar a função arrancar() após 5 segundos.Para parar o temporizador antes do fim da contagem, temos :

clearTimeout(nome_do_contador)

Tomamos o seguinte exemplo (de pouca utilidade mas didático):Ao clicar em "Iniciar", ativa-se o temporizador, que após 3 segundos chama uma caixa alerte(). Pode-se parar o processo prematuramente clicando em "Stop":

Iniciar Stop

<HTML>

<HEAD>

</HEAD>

<BODY>

<script language="Javascript">

<!--

function iniciar()

```
{
cont=setTimeout("alert('Está um pouco devagar seus reflexos !')", 3000)
}
// -->
</script>
<a href="#" onClick="iniciar()">Iniciar</a>
&nbsp;&nbsp;&nbsp;&nbsp;
<a href="#" onClick="clearTimeout(cont)">Stop</a>
</HTML>
```

Com a instrução **cont=setTimeout("alert('Está um pouco devagar seus reflexos !')",3000),** inicializa-se um temporizador, chamado pelo método **setTimeout()**. Este temporizador vai invocar a função iniciar() após 2000 milisegundos seja 2 segundos. Com **clearTimeout(cont)**, pode-se parar o temporizador antes dele acabar a contagem e invocar a função.

O uso de This

Para designar o objeto, o Javascript usa a palavra chave **this**. Esta palavra é frequentemente utilizada substituindo o caminho completo do objeto num formulário. Aqui há um exemplo:

O formulário:

```
<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="botão" VALUE="Qual é a sua escolha ?" onClick="escolhaprop(form3)">

</FORM>
```

Em vez de usar escolhaprop(**form3**), pode-se utilisar escolhaprop(**this.form**) e evitar assim toda confusão com os outros nomes dos formulários. Neste exemplo, **this.form** refera-se ao formulário form3 completo. Enquanto que, **escolhaprop(this)** iria referir-se só ao elemento tipo botão do form3.

Para completar, this é utilizado também para criar uma ou várias propriedades de um objeto. Assim, para criar um objeto livro com as propriedades autor, editor e preço esta operação pode ser efetuada com a ajuda de uma função:

```
function livro(autor, editor, preço) {
this.autor = autor;
this.editor = editor;
this. preco = preço;
}
```