

1 Window

1.1 Sumario

- 1 Introducción
- 2 Propiedades do obxecto *Window*
- 3 Métodos do obxecto *Window*
- 4 Manipulación de obxectos *Window*
 - ◆ 4.1 Abrir fiestras
 - ◆ 4.2 Cerrar fiestras

1.2 Introducción

A principal tarefa dun explorador Web é mostrar documentos HTML nunha fiestra. En JavaScript o obxecto *Document* (Documento) representa un documento HTML e o obxecto *Window* (Fiestra) representa a fiestra (ou marco) do explorador que mostra o documento. Aínda que ambos obxectos son importantes para o JavaScript na parte do cliente, o obxecto *Window* é máis importante por unha razón substancial: o obxecto *Window* é o obxecto global na programación na parte do cliente.

1.3 Propiedades do obxecto *Window*

Os obxectos de tipo *window* teñen as seguintes **propiedades**:

- **closed**: É un valor booleano que indica se a fiestra está ou non pechada.
- **defaultStatus**: Esta propiedade contén a mensaxe por defecto que aparece na barra de estado do navegador.
- **frames**: É un **array** que representa os obxectos **frame** que se atopan no obxecto **window** actual.
- **history**: É un **array** que representa as URL almacenadas no historial do obxecto **window** actual.
- **length**: É un valor enteiro que indica o número de **frames** do obxecto **window** actual.
- **name**: Contén o nome do obxecto **window** actual, o do **frame** actual.
- **opener**: Contén a referencia ó obxecto **window** que abreu o obxecto **window** actual (se se empregou para iso o método **open()**).
- **parent**: Contén a referencia ó obxecto **window** que contén o **frameset**.
- **self**: Nome alternativo do obxecto **window** actual.
- **status**: Determina a mensaxe que aparece na barra de estado do navegador.
- **top**: Nome alternativo para a fiestra de nivel máis superior.
- **window**: Nome alternativo do obxecto **window** actual.

1.4 Métodos do obxecto *Window*

Os obxectos de tipo *Window* teñen os seguintes **métodos**:

- **alert(mensaxe)**: Mostra a mensaxe nunha caixa de diálogo.
- **blur()**: Elimina o foco do obxecto **window** actual.
- **clearInterval(id)**: Elimina o intervalo referenciado por **id**.
- **clearTimeout(nome)**: Cancela o intervalo especificado por nome.
- **close()**: Pecha o obxecto **window** actual.
- **eval()**: Avalía a cadea como unha sentencia, en referencia ó obxecto **window**.
- **focus()**: Captura o foco sobre o obxecto **window** actual.
- **moveB(x,y)**: Move o obxecto **window** actual ó número de pixels especificados por x e y.
- **moveTo(x,y)**: Move o obxecto **window** actual ás coordenadas especificadas por x e y.
- **open(url, nome, características)**: Abre a url especificada na fiestra chamada **nome**. Se non existe tal fiestra, a url abrírase nunha nova fiestra coas características especificadas. Entre as características opcionais que definen unha nova fiestra podemos atopar as seguintes:
 - **toolbar** = [yes,no,1,0]: Indica se a nova fiestra deberá ter barra de ferramentas ou non.
 - **location** = [yes,no,1,0]: Indica se a nova fiestra deberá ter campo de localización ou non.
 - **directories** = [yes,no,1,0]: Indica se a nova fiestra deberá ter botóns de dirección ou non.
 - **status** = [yes,no,1,0]: Indica se a nova fiestra deberá ter barra de estado ou non.
 - **menubar** = [yes,no,1,0]: Indica se a nova fiestra deberá ter barra de menús ou non.
 - **scrollbars** = [yes,no,1,0]: Indica se a nova fiestra deberá ter barras de desprazamento ou non.
 - **resizable** = [yes,no,1,0]: Indica se a nova fiestra poderá cambiar de tamaño ou non.

- `width = pixels`: Indica o ancho da fiestra cliente en pixels.
 - `height = pixels`: Indica o alto da fiestra cliente en pixels.
 - `outerWidth = pixels`: Indica o ancho total da fiestra en pixels.
 - `outerHeight = pixels`: Indica o alto total da fiestra en pixels.
 - `left = pixels`: Indica a distancia en pixels dende o lado esquerdo da pantalla á que se deberá colocar a fiestra.
 - `top = pixels`: Indica a distancia en pixels dende o lado superior da pantalla á que se deberá colocar a fiestra.
 - `alwaysRaised = [yes,no,1,0]`: Indica se a fiestra deberá permanecer sempre en fronte de calquera outra.
 - `z-lock = [yes,no,1,0]`: Indica se se debe bloquear ou non a fiestra no seu z-order actual.
- `prompt(mensaxe, resposta)`: Mostra a mensaxe nunha caixa de diálogo que contén unha caixa de texto que permite ó usuario introducir información. En dita caixa aparece inicialmente a resposta por defecto, se é que esta existe. A resposta devólvese como unha cadea de caracteres.
 - `resizeBy(x,y)`: Axusta o tamaño do obxecto `window` actual movendo a súa esquina inferior dereita, o número de pixels especificados por `x` e `y`.
 - `resizeTo(ancho,alto)`: Axusta o tamaño do obxecto `window` actual cambiando os valores `outerWidth` e `outerHeight` ós valores especificados en ancho e alto.

1.5 Manipulación de obxectos *Window*

O obxecto *window* define diversos métodos que permiten un control total da propia fiestra. Veremos como abrir e pechar fiestras, controlar a posición e tamaño, solicitar e abandonar o enfoque do teclado e desprazar o contido dunha fiestra.

1.5.1 Abrir fiestras

Podemos abrir unha nova fiestra do explorador có método `open()` do obxecto *Window*. `Window.open()` acepta catro argumentos opcionais e devolve un obxecto *Window* que representa a fiestra que se acaba de abrir.

- **O primeiro argumento para `open()`** é a URL do documento a mostrar na nova fiestra. Se se omite este argumento, a fiestra estará baleira.
- **O segundo argumento de `open()`** é o nome da fiestra. Este nome pode ser útil como valor do atributo `target` dunha etiqueta `<form>` ou `<a>`. Se especificamos o nome dunha fiestra xa existente, `open()` devolverá unha referencia a esta fiestra xa aberta en lugar de abrir unha nova.
- **O terceiro argumento opcional de `open()`** é unha lista de opcións que especifican o tamaño da fiestra e as decoracións da GUI. Omitindo este argumento, a nova fiestra terá un tamaño predeterminado e un conxunto de opcións estándar.

Exemplo: Abrir unha fiestra de explorador pequena, redimensionable, cunha barra de estado pero sen barra de menús, barra de ferramentas nin barra de localización:

```
var w = window.open("pequena.html", "pequena", "width=400, height=350, status=yes, resizable=yes");
```

- **O cuarto argumento de `open()`** é útil só cando o segundo argumento chama a unha fiestra xa existente. Este argumento especifica se o URL especificado como primeiro argumento debe re emprazar a entrada actual no historial de exploración da fiestra (`true`) ou non (`false`), que é o comportamento predeterminado.

O valor devolto do método `open()` é o obxecto *Window* que representa a fiestra que se acaba de crear.

1.5.2 Cerrar fiestras

Do mesmo xeito que o método `open()` abre unha nova fiestra, o método `close()` a pecha. Se creamos un obxecto *Window* denominado `w`, poderemos cerralo con:

```
w.close();
```