



Co-funded by the
Erasmus+ Programme
of the European Union



Materiais de Formação em JavaScript

Subcapítulo 3 – JS BOM

WP3: Materiais de Formação do Code4SP

Elaborado por:



CITIZENS
IN POWER



Center for Social
Innovation



ZAUG
gGmbH



Co-funded by the
Erasmus+ Programme
of the European Union





Subcapítulo 3 – JS BOM

Co-funded by the
Erasmus+ Programme
of the European Union





O Modelo de Objeto do Browser (BOM)

- O Modelo do Objeto do Browser (BOM) permite que o JavaScript “fale” para o browser.
- Não existem standards oficiais para o Modelo de Objeto do Browser (BOM).
- Tendo em conta que os browsers modernos já implementaram (praticamente) os mesmos métodos e propriedades para a interatividade do JavaScript, este é muitas vezes referido como métodos e propriedades do BOM.





O Objeto da Janela

- O objeto da janela é suportado por todos os browsers. Representa a janela do browser.
- Todos os objetos, funções e variáveis globais de JavaScript tornam-se, automaticamente, membros de um objeto da janela. As variáveis globais são propriedades do objeto da janela.
- As funções globais são métodos de um objeto da janela.

Até o objeto de um documento (HTML DOM) é uma propriedade do objeto da janela:

```
window.document.getElementById("header");
```





O tamanho da janela

Duas propriedades podem ser usadas para determinar o tamanho da janela do browser. Ambas propriedades retomam o tamanho em pixéis:

- `window.innerHeight` – o tamanho interior da altura da janela do browser (em pixéis)
- `window.innerWidth` - o tamanho interior da largura da janela do browser (em pixéis)

A janela do navegador (a janela de visualização do navegador) NÃO inclui barras de ferramentas e barras de *scroll*.

```
let w = window.innerWidth;  
let h = window.innerHeight;
```





Ecrã de Janela em JavaScript

O objeto `window.screen` pode ser escrito sem o prefixo “window”.

Propriedades:

- `screen.width`
- `screen.height`
- `screen.availWidth`
- `screen.availHeight`
- `screen.colorDepth`
- `screen.pixelDepth`





Localização da Janela em JavaScript

O objeto `window.location` pode ser usado para obter um endereço de página atual (URL) e redirecionar o browser para uma nova página. O objeto `window.location` pode ser escrito sem o prefixo “`window`”. Alguns exemplos:

- `window.location.href` devolve o *href* (URL) da página atual
- `window.location.hostname` devolve o nome de domínio ao anfitrião da página
- `window.location.pathname` devolve o caminho e o filename da página atual
- `window.location.protocol` devolve o protocolo web usado (`http:` ou `https:`)
- `window.location.assign()` carrega um novo documento





Histórico da Janela

O objeto `window.history` pode ser escrito sem o prefixo “window”. Para proteger a privacidade dos utilizadores, existem limitações sobre como o JavaScript pode aceder a este objeto. Alguns métodos:

- `history.back()` - O mesmo que carregar para trás no browser
- `history.forward()` - O mesmo que carregar para a frente no browser





Histórico da Janela para Trás

O método `history.back()` carrega o URL anterior na lista do histórico. É o mesmo que carregar no botão de retroceder do browser.

Example

Create a back button on a page:

```
<html>
<head>
<script>
function goBack() {
  window.history.back()
}
</script>
</head>
<body>

<input type="button" value="Back" onclick="goBack()">

</body>
</html>
```

The output of the code above will be:

Back





Janela de Navegação

O objeto `window.navigator` contém informação sobre o browser do visitante. O objeto `window.navigator` pode ser escrito sem o prefixo “`window`”. Alguns exemplos:

- `navigator.appName`
- `navigator.appCodeName`
- `navigator.platform`

```
document.getElementById("demo").innerHTML =  
"navigator.appName is " + navigator.appName;
```





Caixas Popup em JavaScript

Caixa de Alerta

Uma caixa de alerta é muitas vezes usada para garantir que a informação chega ao utilizador.

Quando uma caixa de alerta aparece, o utilizador terá que carregar “OK” para continuar. O método `window.alert()` pode ser escrito sem o prefixo “window”.

```
alert("I am an alert box!");
```





Caixa de Confirmação

A caixa de confirmação é, por norma, usada no caso de querer que o utilizador verifique ou aceite algo. Quando uma caixa de confirmação aparece, o utilizador terá que carregar “OK” ou “Cancelar” para continuar. Se o utilizador carregar em “OK”, a caixa retoma como verdadeira. Se o utilizador carregar em “Cancelar”, a caixa retoma falsa

O método `window.confirm()` pode ser escrito sem o prefixo “window”.

```
if (confirm("Press a button!")) {  
    txt = "You pressed OK!";  
} else {  
    txt = "You pressed Cancel!";  
}
```





Caixas de Solicitação

A caixa de solicitação é, por norma, usada se quiser que o utilizador coloque um valor antes de entrar na página.

Quando uma caixa de solicitação aparece, o utilizador tem de carregar em “OK” ou em “Cancelar” de forma a proceder depois de introduzir um valor.

Se o utilizador carregar em “OK”, a caixa retoma um valor de entrada. Se o utilizador escolher “Cancelar”, a caixa retoma falsa.

```
let person = prompt("Please enter your name", "Harry Potter");
let text;
if (person == null || person == "") {
  text = "User cancelled the prompt.";
} else {
  text = "Hello " + person + "! How are you today?";
}
```





Eventos de Tempo

O objeto da janela permite a execução de um código em intervalos de tempo específicos. Estes intervalos de tempo são chamados “eventos de tempo”. Os dois métodos essenciais para usar com o JavaScript são:

- `setTimeout(function, milliseconds)` Executa uma função, depois espera um número específico de milissegundos.
- `setInterval(function, milliseconds)` O mesmo que o `setTimeout()`, mas repete a execução da função continuamente.

O `setTimeout()` e o `setInterval()` são ambos métodos do Objeto da Janela HTML em DOM.





Cookies do JavaScript

Os cookies são dados, armazenados em pequenos ficheiros de texto, no seu computador. Quando um servidor web envia uma página web para um browser, a ligação é desligada, e o servidor esquece tudo sobre o utilizador. Os cookies foram inventados para resolver o problema "como lembrar informações sobre o utilizador":

- Quando um utilizador visita uma página web, o seu nome pode ser armazenado num cookie.
- Da próxima vez que o utilizador visitar a página, o cookie "lembra-se" do seu nome.

Os cookies são salvos em elementos de valor nominal, como o ex.:

```
username = John Doe
```





VAMOS PRATICAR!

Abra o link abaixo de forma a praticar alguns dos conceitos aprendidos neste subcapítulo:

<https://www.w3resource.com/javascript-exercises/javascript-object-exercises.php>





OBRIGADO!

Co-funded by the
Erasmus+ Programme
of the European Union

