
Os Objetos `document` e `navigator`

Nessa aula, **finalizando** o estudo dos objetos em JavaScript, conheceremos os objetos `document` e `navigator`.

Veremos também como gerar páginas **dinâmicas e dependentes do navegador**.

Objetivos da aula:

Nessa aula você aprenderá:

- as características e métodos do objeto `document`
 - como gerar **páginas dinamicamente**
 - as características e métodos do objeto `navigator`
-

O objeto `document`

- É um dos objetos mais usados na forma *client side* de JavaScript, pois permite **transformar os documentos HTML estáticos em programas interativos**;
- É através do objeto `document` que é possível **acessar e modificar o conteúdo do documento atual mostrado pelo navegador**.

Como modificar **propriedades**

- Cada objeto `window` tem uma propriedade `document`. Esta propriedade se refere ao objeto `document` que representa o documento HTML mostrado na janela.
 - As propriedades deste objeto ao serem modificadas modificam o documento e a forma como a página é mostrada.
 - Algumas propriedades deste objeto são *arrays* que representam **formulários**, **imagens**, **links**, **âncoras** e **applets** do documento.
-

alterando a cor de fundo

```
<html><head><title>MUDANDO a Cor</title>
<script language="javascript">
function mudaCor ( cor )
    {
        cor = " light " + cor ;
        document.backgroundColor = cor ;    }
</script>
</head><body><font size=7><p>Mudando a Cor</p>
<a href = "javascript: mudaCor ( 'pink' ) " > Rosa </a> -
<a href = "javascript: mudaCor ( 'green' ) " > Verde </a> -
<a href = "javascript: mudaCor ( 'blue' ) " > Azul </a>
</body> </html>
```

alterando a cor de fundo

```
<html><head><title>MUDANDO a Cor</title>
<script language="javascript">
function mudaCor ( cor )
    {
        cor = " light " + cor ;
        document.bgColor = cor ;    }
</script>
</head><body><font size=7><p>Mudando a Cor</p>
<a href = "javascript: mudaCor ( 'pink' ) " > Rosa </a> -
<a href = "javascript: mudaCor ( 'green' ) " > Verde </a> -
<a href = "javascript: mudaCor ( 'blue' ) " > Azul </a>
</body> </html>
```

Como modificar **propriedades na forma de *array***

- Por exemplo o *array* `images []` contém os objetos que representam as imagens do documento.
 - Cada tag `` em um documento HTML cria um elemento numerado no *array* `images []`.
 - Usar o número do elemento no ***array*** permitem trocar as imagens.
 - Idem para as tags `<a name>`, `<a href>`, `<form>`, `<applet>` que criam um elemento nos ***arrays***: `anchors[]`, `links[]` , `forms[]`, `applets[]`.
-

Modificar propriedades por **NAME**

- Se alguma *tag* tiver a propriedade **NAME**, o valor deste atributo pode ser usado como um nome de uma propriedade do objeto `document`.
- Por exemplo, suponha que no exemplo de **Hierarquia** da aula passada o usássemos :

```
..... <FORM NAME = " f1 " >.....
```

```
    novo = document . f1 . elements [2] . value
```

```
.... document . f1 . elements [3] . value = " Trocou ....
```

seriam usados ao invés de :

```
    novo = document . forms [ 0 ] . elements [ 2 ] . value ;
```

```
    document . forms [ 0 ] . elements [ 3 ] . value = " Trocou ...
```

Alterando dinamicamente o conteúdo

- Os métodos `clear`, `write` e `writeln` permitem alterar dinamicamente textos HTML nos documentos que estão sendo interpretados.
- Os métodos `close` e `open` permitem criar documentos inteiramente novos.
- Não esquecer o fechamento porque o *browser* pode **armazenar temporariamente** o HTML que você está escrevendo e só mostrar esta tela de saída quando o método `close()` for chamado.

Objeto navigator

- A forma como este objeto está implementado varia entre os *browsers* Explorer e Netscape, mas propriedades permitem a criação de documentos adaptáveis a cada tipo de navegador.
 - Como há uma única instância deste objeto **não é necessário** se referir a ele como:
`window.navigator`.
 - As propriedades deste objeto contêm informações **apenas para leitura**, isto é, **não modificáveis**, sobre o *browser* ou navegador em uso.
-

Páginas Dependentes do Navegador

- Devido às diferenças de implementação entre os principais tipos de navegadores, algumas vezes é necessário distinguir qual o navegador é usado.
 - Uma forma de fazer isso é através do objeto `navigator`.
 - A técnica de criar código adaptável ao tipo do *browser* consiste em colocar o código dependente do tipo abaixo de um desvio condicional, onde se testa o valor da propriedade `appName`.
 - Como JavaScript é interpretada, o navegador só acusa o erro de não reconhecer um código quando o executa o que é evitado pelo desvio.
-

Identificando o navegador

```
function ident( )  
{`ns = (navigator.appName.indexOf("Netscape") != -1);  
  ie = (navigator.appName.indexOf("Explorer") != -1);  
  document.write("Seu navegador é: ");  
  if (ns) document.write("Netscape");  
    else if (ie) document.write("Explorer");  
      else document.write("desconhecido"); }  
  
.....  
<script language="javascript"> ident( ); </script>
```

finalizando....

Nesta aula você viu detalhes especialmente úteis para criar páginas com **maior dinamismo** e que podem ser **geradas automaticamente**. Na próxima aula **continuaremos neste assunto**, aprendendo como fazê-las **interagir com o usuário**, o que permitirá a criação de páginas muito versáteis.

Faça uma **auto-avaliação geral de objetos**. Se algum ponto não ficou muito claro, releia-o antes da próxima aula, na qual você aprenderá as coisas mais importantes para uma programação dinâmica: como **usar os eventos!**
