

Programação - Dynamic HTML

DHTML no Netscape 4.0

Ao contrário do Internet Explorer, a implementação DHTML apresentada no [Netscape Navigator 4.0](#), sendo anterior, cobre apenas três áreas principais algo distintas. Essas áreas são:

- **Style Sheets**

O Navigator suporta duas sintaxes distintas para definir **Style Sheets**: a sintaxe CSS (Cascading Style Sheets) e uma segunda sintaxe escrita em **JavaScript** e que utiliza o modelo de objectos do documento.

- **Posicionamento de Conteúdo**

Esta funcionalidade permite definir a posição de blocos de HTML, de forma que o autor da página passa a ter um controlo absoluto do que fica aonde na página. Depois, recorrendo a scripts desenvolvidas em JavaScript, é possível modificar o layout dinamicamente, fazendo os elementos aparecerem, desaparecerem ou mudarem de lugar.

- **"Downloadable Fonts"**

Finalmente, o Navigator permite também associar fonts específicas a uma página Web. Dessa forma, a página será sempre visualizada pelo utilizador da forma mais conveniente. Ao browser caberá a responsabilidade de carregar as fonts necessárias para o computador do utilizador.

Analisemos então estas três áreas com mais algum pormenor.

Nota: Os exemplos apresentados nesta página apenas funcionarão correctamente com o [Netscape Navigator 4.0](#) ou superior.

Style Sheets

Tal como o Internet Explorer, o Netscape Navigator suporta a definição de style sheets utilizando a sintaxe CSS. Assim, qualquer página desenvolvida utilizando este tipo de style sheets, será convenientemente visualizada utilizando o Navigator 4.0.

No entanto, a grande novidade na última versão deste browser é a possibilidade de definir style sheets utilizando JavaScript. Deste modo, é possível criar estilos que tiram partido do "document object model". Neste modelo, uma página HTML é considerada como sendo um objecto, com propriedades que podem ser acedidas e alteradas. Cada uma dessas propriedades pode ser, por sua vez, um objecto com outras propriedades.

Assim, por exemplo, pode-se criar uma style sheet JavaScript, que altera a definição das tags P e H1, da seguinte forma:

```
<STYLE TYPE="text/javascript">
tags.P.fontSize = "18pt";
tags.P.marginLeft = "20pt";
tags.H1.color = "blue";
</STYLE>
```

Consideremos, como exemplo, a mesma style sheet definida nas duas sintaxes (CSS e JavaScript), de forma a comparar as diferenças:

```
<STYLE TYPE=text/css>
H4 {text-decoration:underline; color:green;}
H5 {text-transform:uppercase; color:red;}
```

```
BLOCKQUOTE {color:blue; font-style:italic; line-height:1.5; text-indent:5%;}  
</STYLE>
```

```
<STYLE TYPE=text/javascript>  
with (tags.H4) {  
  textDecoration="underline";  
  color="green";}  
with (tags.H5) {  
  textTransform="uppercase";  
  color="red";}  
with (tags.BLOCKQUOTE) {  
  color="blue";  
  fontStyle="italic";  
  lineHeight=1.5;  
  textIndent="5%";}  
</STYLE>
```

Eis o resultado da aplicação da style sheet JavaScript:

Heading 4

Heading 5

Este bloco de texto está inserido entre as tags <BLOCKQUOTE> e </BLOCKQUOTE>. Note-se o efeito do início do parágrafo começar mais à direita.

De resto, a sintaxe JavaScript aplica todos os conceitos (definição de classes, IDs, herança, etc) da mesma forma que a sintaxe CSS, apenas com algumas diferenças de notação. Por exemplo, a definição de classes e identificadores (IDs) deve ser feita recorrendo, respectivamente, às propriedades **CLASSES** e **IDS** do documento:

```
<STYLE TYPE=text/javascript>  
classes.DARK.all.color = "black";  
classes.RED1.P.color = "red";  
classes.RED1.P.fontWeight = "bold";  
ids.BLUE1.color = "blue";  
</STYLE>
```

Posicionamento de Conteúdo

O Netscape Navigator 4.0 introduz novas funcionalidades que permitem definir blocos de HTML que podem ser precisamente posicionados na página Web.

Utilizando código JavaScript, esses blocos - ou **Layers** - podem ser modificados das mais variadas formas: movendo-os, escondendo-os, modificando a sua cor de fundo, etc. Mais: o Navigator permite inclusivé modificar o seu conteúdo ou criar novos layers.

Basicamente, existem duas formas distintas de posicionar dinamicamente layers HTML:

- **Definindo estilos com a propriedade POSITION**

Neste caso, a style sheet para definir um bloco posicionado de HTML deve sempre incluir a propriedade **POSITION**. De resto devem utilizar-se também as propriedades **TOP** e **LEFT** para definir a posição do layer.

```
<STYLE TYPE="text/css">  
#layer1 {position:absolute; top:20px; left:5px; width:200px; background-color:#cc00ee;}  
</STYLE>
```

Neste exemplo, o código HTML para incluir o layer definido no bloco STYLE, poderia ser o seguinte:

```
<DIV ID="layer1">
  <H1>Layer 1</H1>
  <P>Conteúdo do Layer 1</P>
</DIV>
```

- **Usando a tag LAYER**

Neste caso, o mesmo layer do exemplo anterior seria definido da seguinte forma:

```
<LAYER ID="layer1" TOP=20pt LEFT=5pt WIDTH=200 BACKGROUND="#cc00ee">
  <H1>Layer 1</H1>
  <P>Conteúdo do Layer 1</P>
</LAYER>
```

Nota: Nesta altura a tag LAYER é específica do Netscape Navigator 4.0. Outros browser podem não suportar este elemento correctamente.

Na verdade, a grande força do posicionamento de layers no Netscape Navigator está intimamente relacionada com a facilidade de alterar as propriedades desses layers utilizando código JavaScript.

A título de exemplo, a operação de esconder um determinado layer (chamado, por exemplo, "layer1") é tão simples como incluir a seguinte linha numa script JavaScript:

```
<SCRIPT LANGUAGE="JavaScript">
...
document.layers["layer1"].visibility = "hide";
...
</SCRIPT>
```

"Downloadable Fonts"

Os melhoramentos do tratamento de fonts no Navigator 4.0 incluem a possibilidade de associar "downloadable fonts" a um documento. Desta forma, o autor fica com a inteira liberdade de utilizar qualquer font que possa melhorar a qualidade do design das suas páginas, sem a preocupação de saber se os utilizadores terão, ou não, essa font instalada no seu computador.

Para isso a font deve ser armazenada, em conjunto com a página em si, num "font definition file" no servidor. Assim que o documento for acedido, esse ficheiro será carregado pelo browser tal como acontece com qualquer imagem. E o ficheiro ficará no sistema cliente apenas durante a visualização da página associada.

Resta apresentar a forma de associar uma font a um documento.

O primeiro passo consiste em criar a "font definition file". Para tal é necessário recorrer a uma ferramenta própria, como o [HexMac Typograph](#) ou o **Font Composer Plugin**.

A associação entre esse ficheiro e o documento HTML, pode ser feita de duas formas:

```
<STYLE TYPE="text/css">
@fontdef url(http://server.com/fonts/sample.pfr)
</STYLE>

<LINK REL=FONTDEF SRC="http://server.com/fonts/sample.pfr">
```

Finalmente, depois de associar as fonts ao documento, estas podem ser utilizadas no documento exactamente da mesma forma que se utiliza qualquer outra font. Utilizando o atributo **FACE** da tag **FONT** ou o atributo **FONT-FAMILY** na definição de style sheets.

Fontes de Informação

Aqui ficam também alguns links para sites sobre o suporte DHTML no Netscape Navigator:

- [Dynamic HTML in Netscape Communicator](#)
- [JavaScript Authoring Guide](#)
- [Project Cool Developer Zone](#)
- [Netscape](#)

De seguida apresentaremos dois exemplos práticos da aplicação dos conceitos discutidos quer no I.E. 4.0 como no Navigator 4.0.

[Intro](#) | [Anterior](#) | [Seguinte](#)