

1 Introducción a (X)HTML

1.1 Sumario

- 1 Introducción
- 2 O W3C
- 3 O papel de HTML
- 4 Principios básicos do marcado
- 5 Elementos
 - ◆ 5.1 Atributos
 - ◆ 5.2 Elementos anidados
- 6 Información que ignoran os navegadores
- 7 Protexer Scripts
- 8 Ferramentas de creación de páxinas Web
 - ◆ 8.1 Editores HTML/XHTML
 - ◆ 8.2 Ferramentas de autoría WYSIWYG
- 9 Boas prácticas de autoría

1.2 Introducción

HTML (*Hypertext Markup Language*, **Linguaxe de Marcado de Hipertexto**) é a linguaxe de marcado empregado para converter documentos de texto en páxinas Web. HTML permite ós autores identificar elementos que estruturan os documentos, como cabeceiras, parágrafos, listas, etc. Outros elementos actúan como mecanismos para a adición de vínculos de hipertexto, formularios interactivos e son e vídeo nas páxinas Web.

HTML evolucionou moito dende a súa creación en 1991 (por **Tim Berners-Lee**) como un modo sinxelo de indicar o significado e estrutura dos documentos de hipertexto. Non pasou moito tempo ata que se engadiron novos elementos para concibir HTML como unha ferramenta de deseño visual.

XHTML (*eXtensible Hypertext Markup Language* - Linguaxe Extensible de Marcado de Hipertexto), é unha reformulación de HTML en **XML** (*eXtensible Markup Language* - Linguaxe de Marcas Extensible). Noutras palabras, emprega o mesmo vocabulario que HTML pero con regras sintácticas tomadas de XML, moito mais estritas.

1.3 O W3C

Advertindo a necesidade de poñer orde no desenvolvemento de HTML, Berners-Lee fundou en 1994 o **World Wide Web Consortium** (**W3C**). O **W3C** supervisa o HTML e tecnoloxías Web relacionadas, dende 1995 sacou á luz versións actualizadas e estandarizadas de HTML en publicacións coñecidas como "Recomendacións". Os estándares actuais son HTML 5 (2014) e **XHTML 1.1** (2001).

1.4 O papel de HTML

O documento HTML con marcado denomínase a **capa estrutural** dunha páxina Web. É a base sobre a que se aplican a **capa de presentación** (instrucións que indican como se mostran en pantalla os elementos), para o que temos o **CSS**, e a **capa de comportamento** (programación e interactividade), que se programará con linguaxes de script de servidor (**PHP**, **ASP**, **Perl**,...) ou linguaxes de script de cliente (**JavaScript**, **VBScript**, **JScript**, **ECMAScript**,...).

Por tanto, o obxectivo fundamental de HTML é describir o contido da páxina e establecer unha estrutura do documento.

1.5 Principios básicos do marcado

Un documento HTML ou XHTML é un documento ASCII (texto) ou Unicode (por exemplo UTF-8) marcado con etiquetas que indican elementos e outras declaracións necesarias.

1.6 Elementos

Os elementos van indicados na fonte do texto coa inserción de etiquetas HTML. A maioría dos elementos seguen esta sintaxe:

```
<nome_elemento>contido</nome_elemento>
```

O nome do elemento aparece na etiqueta de apertura e na de peche, esa segunda vez precedido por unha barra invertida (/). A etiqueta de peche "apaga" o elemento. Nada do que está recollido entre < e > é mostrado polo navegador.

Nota: En XHTML os nomes de atributos e de elementos deben ir en minúsculas. HTML non distingue entre maiúsculas e minúsculas pero si é conveniente.

Nota: En XHTML sempre son necesarias as etiquetas de peche.

Algúns elementos non teñen contido porque se empregan para proporcionar unha directiva sinxela. Estes elementos son elementos "baleiros". Estes elementos poden verse na seguinte táboa:

Elementos baleiros

<area />	<frame />	<link />
<base />	<hr />	<meta />
<basefont /> (obsoleto)		<param />
 	<input />	<col />
<isindex /> (obsoleto)		

1.6.1 Atributos

Un atributo permite modificar a acción dun elemento, é dicir, determinan unha propiedade dunha etiqueta. Os atributos indícanse na etiqueta de apertura nunca na de peche. A orde de aparición dos atributos non ten importancia.

A sintaxe dun elemento con atributos é así:

```
<elemento atributo="valor">contido</elemento>
```

XHTML precisa que todos os valores de atributo estean recollidos entre comiñas. Poden empregarse comiñas sinxelas ou dobres.

1.6.2 Elementos aniñados

Os elementos HTML poden conter outros elementos. Son os chamados elementos aniñados e para que sexan correctos, todo o elemento debe estar nas etiquetas de apertura e peche do elemento que o contén (o pai). Este é un dos criterios dun documento ben formado (un requisito XHTML).

Un erro típico ao aniñar elementos é pechar o elemento pai antes de pechar o elemento que contén (o seu fillo).

1.7 Información que ignoran os navegadores

Pode haber información dentro dun documento HTML que é ignorada polo navegador. Vexamos algúns exemplos:

- **Salto de liña:** Nos documentos HTML os saltos de liña trátanse como espazos.
- **Espazos múltiples e tabulacións:** Cando un navegador atopa mais dun espazo en branco consecutivo nun documento HTML móstrao como un espazo único. Poden engadirse espazos adicionais no fluxo de texto empregando o código ` `; para inserir un *Non Blank Space*. As tabulacións nos documentos son mellor evitalas.
- **Elementos baleiros:** Trátanse como se fosen un salto de parágrafo. A maioría dos navegadores mostran varios elementos `
` como varios saltos de liña.
- **Elementos non recoñecidos:** Un navegador ignora os elementos que non comprende ou que están mal especificados. Moitas veces eses elementos móstranse como se fosen texto normal.
- **Texto en comentarios:** Os navegadores non mostran en pantalla o texto que estea entre os elementos `<!--` e `-->`. Por exemplo:

Na seguinte ligazón hai varios exemplos disto: http://www.w3schools.com/html/tryit.asp?filename=tryhtml_poem

1.8 Protexer Scripts

Cando se introduce un *script* nun documento XHTML e non queremos que dea un erro cando o lea un navegador vello o mellor é recollelo como se fose un comentario de XHTML. Por exemplo:

```
<script language="JavaScript">  
  
</script>
```

1.9 Ferramentas de creación de páxinas Web

Os documentos HTML son arquivos de texto, así que se pode empregar calquera editor de texto para escribilos. Afortunadamente hai moitas ferramentas que fan mais rápida e eficaz a xeración de documentos HTML. Estes programas divídense en dúas categorías principais: editores HTML e ferramentas de autoría Web **WYSIWYG**, *What You See Is What You Get* ("o que ves é o que obtés").

1.9.1 Editores HTML/XHTML

Son ferramentas de edición de texto deseñadas especialmente para escribir HTML. Necesítase saber como escribir "a man" HTML, pero teñen ferramentas especiais para tarefas repetitivas (crear táboas, aplicar estilos, ...).

Hai moitos editores e moitos deles libres. Para atopar algún podes facer unha busca coas palabras "free HTML editor".

1.9.2 Ferramentas de autoría WYSIWYG

Os editores HTML WYSIWYG teñen interfaces gráficas que fan que escribir HTML sexa mais parecido ao uso dun procesador de textos ou dun programa de deseño de páxinas Web. Estas ferramentas teñen varias vantaxes como poden ser:

- Aforran tempo fronte á escritura a man.
- Non precisan que se coñeza a linguaxe HTML, incluso poden permitir aprender HTML pois se pode deseñar unha páxina como se queira e ver o código resultante.
- Son boas para facer probas e prototipos.
- Son un bo punto de arranque para crear follas de estilo, introducir scripts nas páxinas e outras funcións.

Hai que ter en conta que unha ferramenta WYSIWYG non nos libera da aprendizaxe de HTML. En moitos casos hai que axustar o código HTML a man. Algunhas das ferramentas mais populares son:

- Adobe Dreamweaver (privativa)
- Adobe GoLive (privativa)
- Microsoft FrontPage (privativa e pouco estándar)
- KompoZer/Nvu (libre e multiplataforma)

1.10 Boas prácticas de autoría

Algunhas pautas para escribir "bos" documentos HTML son as seguintes:

- Seleccionar elementos que describan con precisión e con significado o contido
- Non seleccionar elementos HTML extra en función de como se visualiza o resultado no navegador, para iso están as follas de estilos
- Evitar o uso de elementos e atributos obsoletos
- Escribir documentos válidos e fieis aos estándares
- **Validar o HTML (W3C)**. Validar un documento HTML implica comprobar que segue as restricións dun determinado DTD. Existen varios DTD definidas polo W3C:
 - **HTML 4.01 Strict**. Este DTD contén todos os elementos HTML e atributos, pero **non inclúe** elementos de formato ou obsoletos (deprecated) como font. Non se permiten marcos:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN" "http://www.w3.org/TR/html4/strict.dtd">
```

- **HTML 4.01 Transitional.** Contén todos os elementos HTML e atributos, incluíndo elementos para a formato e obsoletos como font. Non se permiten marcos.

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">
```

- **HTML 4.01 Frameset.** Este DTD é igual que o HTML 4.01 Transitional, pero permite marcos.

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Frameset//EN" "http://www.w3.org/TR/html4/frameset.dtd">
```

Para o HTML 5 non é necesario empregar ningún DTD e indicárase así:

```
<!DOCTYPE html>
```

Para o XHTML temos dous DTD's

- **XHTML 1.0 Strict.** Contén todos os elementos HTML e atributos, pero non inclúe elementos de presentación ou obsoletos. Os marcos tampouco están permitidos

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
```

- **XHTML 1.0 Transitional.** Contén todos os elementos HTML e atributos, incluíndo tamén elementos de presentación ou obsoletos. Os marcos tampouco están permitidos

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
```

- Evitar o uso de espazos de caracteres e retornos extra
- Utilizar comentarios para sinalizar seccións e poder atopalas rapidamente
- Seguir as convencións de nomeado de ficheiros:
 - 1.- Empregar a extensión de documento .html ou .htm.
 - 2.- Evitar espazos e caracteres especiais como ?, %, #, etc. nos nomes de ficheiros.
 - 3.- Empregar letras minúsculas para os nomes de ficheiros.
 - 4.- Empregar nomes curtos nos nomes de ficheiros.

--Vieites 13 outubro 2010