Unidad 1

Introducción a los Lenguajes de Marcas

"Me lo contaron y lo olvidé;
Lo vi y lo entendí;
Lo hice y lo aprendí"

Confucio

Un lenguaje de marcas es una forma de codificar un documento que, junto con el texto, incorpora etiquetas (o marcas) que contienen información adicional acerca de la estructura del texto o su presentación. Tal vez la forma más primitiva que hemos usado de lenguaje de marcas fuera un dictado en el que la persona que dicta nos va dando notas acerca de lo que tenemos que poner en negritas, cursivas, etc. En el mundo de los ordenadores llevamos mucho tiempo usando lenguajes de marcas. Wordstar, uno de los primeros procesadores de texto que existieron para el mundo de los PC o Latex, el programa de autoedición favorito para edición profesional de textos, son dos ejemplos de ello:

Wordstar

La lluvia en ^BSevilla^S es una ^Ymaravilla^S.

LATEX

La lluvia en \textbf{Sevilla} es una \textit{maravilla}.

Resultado:

La lluvia en **Sevilla** es una *maravilla*.

Es común en muchos ámbitos confundir un lenguaje de marcas con un lenguaje de programación. Pero no: se trata de cosas diferentes. Hay, fundamentalmente, tres carencias de los lenguajes de marcas que los distinguen:

- No tienen funciones aritméticas
- No tienen variables
- No tienen estructuras de control

Las características principales de un lenguaje de marca son las siguientes:

- Se usan siempre sobre texto plano.
- Las marcas se entremezclan con el contenido del documento aunque, en general, es fácil distinguir unas del otro.
- Su procesamiento es muy sencillo.
- Son muy flexibles.

Historia de los lenguajes de marcas

La primera referencia que se tiene de un lenguaje de marcas como tal está aún alejado de la informática. Se trata de la práctica de los empleados de imprenta de anotar marcas con instrucciones en los márgenes de las pruebas de impresión:

cap	set in CAPITALS	set nato as NATO
sm cap or s.c.	set in SMALL CAPITALS	set signal as SIGNAL
lc	set in lowercase	set South as south
ital	set in *italic*	set oeuvre as *oeuvre*
rom	set in roman	set *mensch* as mensch
bf	set in **boldface**	set important as **important**

En el mundo de la informática el "padre" de los lenguajes de marcas más importantes de la actualidad surge en los años 80. Se llama SGML (Standard Generalized Markup Language) y define unas reglas básicas para el etiquetado de documentos mediante marcas. SGML no es en realidad un lenguaje sino un metalenguaje, es decir, un lenguaje creado con la finalidad de definir otros lenguajes a partir de él. RTF (Rich Text Format), HTML (HyperText Markup Language) o XML (Extensible Markup Language) derivan de SGML.

Algunos lenguajes de marcas (y otros que no son)

HTML es, sin duda, el lenguaje de marcas más usado y la base de las páginas web. Su primera versión, que data de los años 90, describía sólo 22 elementos diferentes. La versión actual es la 5 aprobada finalmente en octubre de 2014 tras un largo desarrollo.

XML se trata también de otro metalenguaje. Parte de un subconjunto de SGML y añade algunas restricciones nuevas de forma que los lenguajes derivados a partir de XML resulten más sencillos y fáciles de interpretar que los derivados directamente de SGML.

XHTML (Extensible HTML) es equivalente a HTML pero deriva de XML. La versión actual es la 5 y se ha desarrollado en paralelo a HTML. Las diferencias entre HTML 5 y XHTML 5 son nímias.

CSS (Cascading Style Sheets) es un lenguaje usado para definir la presentación de un documento en HTML o XHTML. **No se trata en realidad de un lenguaje de marcas**, pero se encuentra indisolublemente unido a estos dos. La versión actual es la 3.1. CSS 3.1 es modular y sus diferentes módulos se encuentran también en fases diferentes de desarrollo. Los primeros se aprobaron en 1999. En otros se sigue aún trabajándo

XSL (Extensible Stylesheet Language) describre la forma en que debería de mostrarse la información contenida en un documento con formato XML.

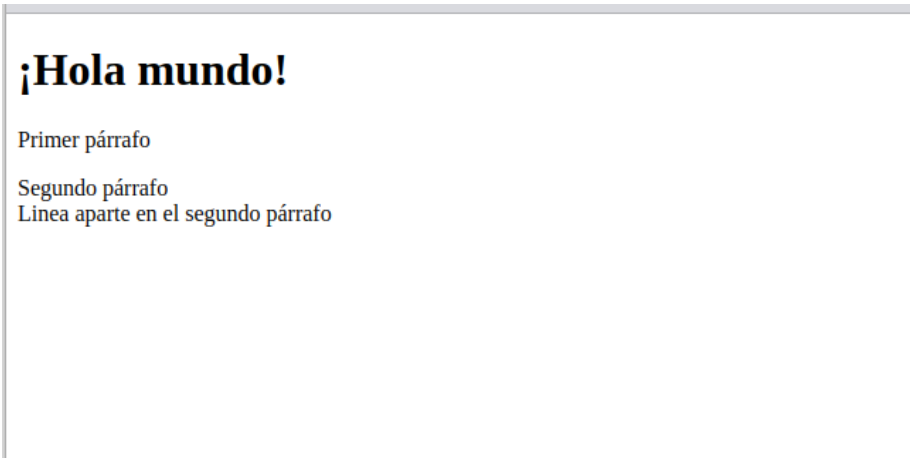
SGML está definido como una norma ISO mientras que el resto de los lenguajes mencionados están definidos por la W3C (Worl Wide Web Consortium).

Componentes de un lenguaje de marcas

Veamos un primer ejemplo muy simple de HTML y analicemos los diferentes componentes que pueden aparecer en un lenguaje de marcas y algunas de las características particulares de HTML:

```
<!DOCTYPE html>
<html lang="es">
	<head>
		<meta charset="UTF-8" />
		<title>HTML</title>
	</head>
	<body>
		<h1>¡Hola mundo!</h1>
		<p>Primer párrafo</p>
		<p>Segundo párrafo<br/>
		Linea aparte en el segundo párrafo</p>
	</body>
</html>
```

Se trata de un ejemplo reducido al mínimo, pero nos vale como primer contacto. Veamos como se vería en un navegador:



La nomenclatura que usaremos es la siguiente:

- **Elementos:** Constan de una etiqueta de inicio, una de fin y todo lo que haya en medio. Los elementos constan, a su vez, de tres elementos: etiquetas, atributos y contenido. Un ejemplo de elemento sería este:

  ```
  <h1>¡Hola mundo!</h1>
  ```

- **Etiquetas o tags:** Son las marcas propiamente dichas y habitualmente van entre corchetes quebrados <> Las hay de inicio y de fin aunque, en algunos casos, ambas pueden coincidir en una sóla partícula con una sintaxis especial cuando el elemento no tiene contenido:

  ```
  <title> ... </title>
  <br/>
  ```

- **Contenido:** Es el texto base informativo del documento. Por ejemplo, en el anterior elemento con las etiquetas h1 el contenido sería el texto que luego aparecerá como titular. El contenido de un elemento puede ser sólo un texto o estar formado a su vez por otros elementos.

- **Atributos:** Es una pareja compuesta por un nombre y un valor que se encuentra dentro de una etiqueta de inicio e identifica alguna propiedad asociadas al elemento.

  ```
  lang="es"
  ```

  En realidad, en HTML convencional es posible encontrar atributos sin valor, pero en XHTML, que será nuestro preferido, esto no es válido como veremos a continuación.

Diferencias entre HTML y XHTML

Como hemos dicho hace un momento, XHTML introduce ciertas restricciones a HTML para hacer que el lenguaje resultante sea más sencillo y fácil de interpretar. Estas diferencias son las siguientes:

- Los elementos deben de cerrarse siempre. En HTML normal es perfectamente válido, por ejemplo, empezar un párrafo con la etiqueta y, sin poner la marca de fin de párrafo comenzar un segundo párrafo de nuevo con . Puesto que no podemos anidar dos párrafos uno dentro de otro el intérprete debe de reconocer que al empezar el segundo párrafo debería antes de terminar el primero. En XHTML hay que cerrar el primero explicitamente o en caso contrario tendremos un error de validación:

  ```
  HTML: <p>Primer párrafo <p>Segundo párrafo
  XHTML: <p>Primer párrafo</p><p>Segundo párrafo</p>
  ```

- Los elementos sin contenido deben de "cerrarse" siempre usando una etiqueta especial que realiza el autocierre en la misma etiqueta de inicio:

  ```
  HTML: <br>
  XHTML: <br/>
  ```

- Los elementos anidados no deben solaparse. En HTML está permitido pero en XHTML es incorrecto:

```
HTML: <em><strong>Texto</em></strong>
XHTML: <em><strong>Texto</strong></em>
```

- Los valores de los atributos deben de ir siempre entre comillas simples o dobles:

```
HTML: <html lang=es>
XHTML: <html lang="es">
```

- Los nombres de etiquetas y atributos deben de ir siempre en minúsculas

```
HTML: <HTML LANG="es">
XHTML: <html lang="es">
```

- No está permitido usar un atributo sin valor (minimización de atributos)

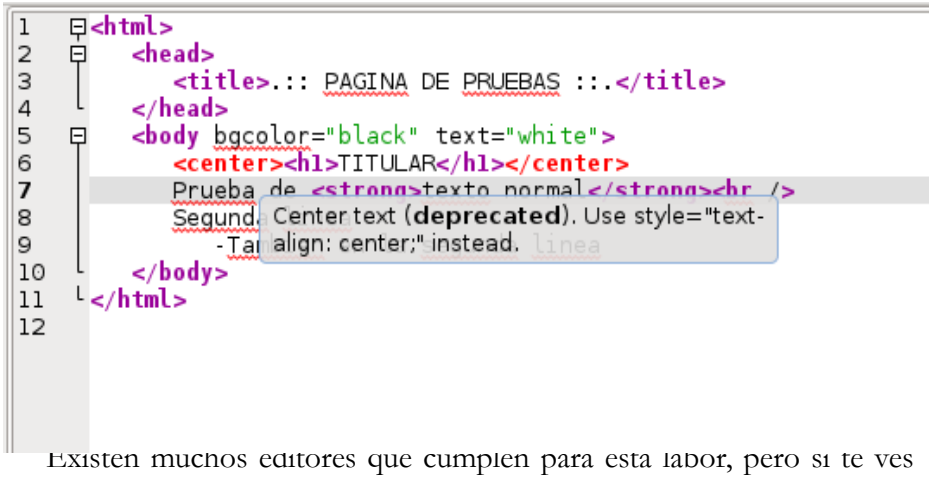
```
HTML: <textarea readonly>
XHTML: <textarea readonly="yes">
```

- Los atributos y etiquetas desaprobados o desaconsejados (deprecated) en HTML no son válidos en XHTML. Los veremos más adelante.

Una aclaración en todo esto: un fichero que valida correctamente como XHTML siempre validará como HTML pero no a la inversa, es decir, si escribimos de acuerdo a la norma marcada por XHTML también conseguimos ficheros válidos para HTML con una sintaxis más clara, legible y menos sujeta a errores de interpretación. Por eso será nuestra elección a lo largo de todo este manual.

Editores de texto enriquecidos

A la hora de trabajar con lenguajes de marcas, la elección de un editor de texto apropiado con ayuda contextual es muy importante:



Existen muchos editores que cumplen para esta labor, pero si te ves perdido a la hora de empezar puedes probar con uno de estos:

- Bluefish (multiplataforma): https://bluefish.openoffice.nl/
- Notepad++ (sólo Windows): https://notepad-plus-plus.org/
- Visual Studio Code (multiplataforma): https://code.visualstudio.com/
- Notepadqq (sólo Linux): https://notepadqq.com/s/
- Brackets (sólo Linux): https://brackets.io/

Navegadores

En el caso del HTML y el XHTML, el navegador web funciona como visor o intérprete del lenguaje de marcas y su respeto por los estándares es fundamental. Muchas veces en el pasado se ha utilizado la posición de supremacía de uno de ellos para desviarse del estándar y perjudicar a la competencia. Afortunadamente estas prácticas parecen abandonadas hoy

en día. Tenemos varios recursos para comprobar la validez de un fichero escrito en HTML o la forma en que el navegador implementa los estándares:

Validadores de HTML y CSS:

- http://validator.w3.org/
- http://jigsaw.w3.org/css-validator/

Tests de cumplimiento de estándares:

- http://acid2.acidtests.org/
- http://acid3.acidtests.org/
- http://www.css3.info/selectors-test/

Importante: *Para validar como XHTML en lugar de como HTML debemos de hacer dos cosas:*
1. Incluir el siguiente atributo en la etiqueta html:
xmlns="http://www.w3.org/1999/xhtml"
2. Nuestro fichero debe de tener la extensión .xhtml en lugar de .html

HTML4 vs HTML5

HTML5 (y CSS3) son ya una realidad. Prácticamente todas las versiones de los navegadores los implementan casi en su totalidad. Su uso simplifica y mejora el diseño de la web

Las principales novedades de HTML5 y CSS3 frente a sus anteriores versiones son las siguientes:

- Ya no se habla de páginas web sino de aplicaciones web. Esto quiere reforzar el cambio en la filosofía que se persigue con esta nueva versión del lenguaje.

- La separación entre presentación y contenido se ve reforzada. En HTML5 todo lo relativo al diseño irá en los CSS. HTML5 sin CSS es en blanco y negro y no se debería siquiera elegir un tipo de letra diferente al que el navegador nos muestra por defecto.

- Existe una gran mejora en todo lo relativo al manejo de formularios

- Se refuerza la importancia de introducir contenido semántico

- HTML5 se encuentra plenamente integrado con Javascript. De hecho, se encuentra ligado de forma casi indisoluble a una amplia colección de API's de Javascript que le proporcionan soporte para diseño 2D y 3D, geolocalización, arrastrar y soltar, multimedia, etc.