

■ Tema 2a

■ Lenguajes de marcado: HTML y XML



Contenidos

■ Parte I: HTML

- Conceptos básicos
- Texto
- Enlaces
- Imágenes
- Formularios
- Tablas

■ Parte II: Introducción rápida a XML



2.1

Conceptos básicos sobre HTML



Características fundamentales

Lenguaje de marcado (etiquetas)

`<p>Como mucha gente sabe, <tt> HTML </tt> es un lenguaje de marcado </p>`

Texto puro (UNICODE)

Enfatiza **contenido** frente a **estilo**

Sintaxis poco estricta



Léxico

■ No distingue mayúsculas/minúsculas (XHTML sí)

■ No significativos espacios en blanco, retornos de carro, tabuladores

■ Comentarios

```
<!-- Esto es un comentario -->
```

■ Entidades: símbolos accesibles mediante un código

< (<) & (&) 水 (el ideograma chino para *agua*)



Tecnologías  Web

Etiquetas

■ **Etiqueta**

```
<h1>
```

■ **Elemento:** etiq. apertura [+ contenido] [+ etiq. cierre]

```
<h1> Bienvenidos a mi página </h1>
```

```
<p> Párrafo
```

```
<br>
```

■ **Atributos:** pares nombre="valor"

```
<h1 id="titulo"> Bienvenidos a mi página </h1>
```



Tecnologías  Web

Versiones del HTML

- Definidas por el W3C (*World Wide Web Consortium*)

<http://www.w3.org>

- Actual: HTML 4.01

- XHTML: HTML 4.01 + sintaxis XML

- Todas las etiquetas se abren y cierran
- Distinción mayúsculas/minúsculas
- ...



Tecnologías  Web

Estructura de un documento HTML

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

Declaración de tipo de documento

```
<html>
```

```
<head>
  <title>
    Mi primer documento HTML
  </title>
```

Cabecera (HEAD)

- Título página (TITLE)
- Metainformación (META)
- Estilos (STYLE)

```
</head>
```

```
<body>
  <p>¡Hola mundo!
```

Cuerpo (BODY o FRAMESET)

```
</body>
```

```
</html>
```



Tecnologías  Web

2.2

Texto



Texto básico

Los espacios, retornos de carro y tabuladores no son significativos

En `<p>` contra de lo que pudiera parecer, todo esto aparecerá dentro del mismo párrafo en HTML, y todo seguido

Párrafos: `<p>`

`<p>` Esto es un párrafo `</p>`
`<p>` Esto otro `<p>` y esto un tercero

Salto de línea `
`

`<p>` ¿No es verdad, ángel de amor `
` que en esta apartada Orilla `
` más pura la luna brilla `
` y se respira mejor? `</p>`



Texto: estilos basados en contenido

■ Indican **estructura** más que **estilo**

Ejemplo: un elemento *éfnasis* (``) simplemente debe destacarse del resto del texto

■ **Encabezados:** definen secciones

`<h1>...</h1>`,.. `<h6>...</h6>`

```
<h1> Introducción al HTML </h1>
```

```
<h2> Etiquetas y atributos </h2>
```

```
<h2> Estructura de un documento </h2>
```



Texto: estilos basados en contenido (II)

■ **Énfasis** ``. Más énfasis ``

En XHTML los atributos `` deben `` ir `` entre comillas ``

■ **Código fuente:** `<code>`, `<var>`

La función `<code>alert(<var>mensaje</var>)`
`</code>` muestra un mensaje en pantalla

La función `alert(mensaje)` muestra un mensaje en pantalla



Texto: estilos físicos

- Indican formato, aunque pueden redefinirse con CSS
- Negrita ``, itálica `<i>`, grande `<big>`, subíndices `<sub>`, superíndices `<sup>`,...

`` Negrita ``, `<i>` itálica `</i>`, H`₂`
Negrita , *itálica* , H₂O



Listas

- Cada elemento se define con un ``

- Listas con viñetas `` La lista de la compra:

La lista de la compra:

```
<ul>
<li> Patatas </li>
<li> 1 botella de Smirnoff </li>
</ul>
```

- Patatas
- 1 botella de Smirnoff

- Listas numeradas `` §HTML:

HTML:

```
<ol>
<li> Conceptos básicos </li>
<ol>
<li> Sintaxis básica
<li> Estructura de un documento </li>
</ol>
<li> Texto </li>
</ol>
```

- Conceptos básicos
 - Sintaxis básica
 - Estructura de un documento
- Texto



2.3

Enlaces



Enlaces: etiqueta <A>

Define un enlace o destino de enlace

Atributos

- href: URL a la que se enlaza (absoluta o relativa)
- name o id: para crear un **destino** de salto

```
Aquí tienes información sobre mi <a href="#experiencia">
experiencia laboral </a>, estudios, <a
href="aficiones.html"> aficiones </a>, ...
```

...

```
Estudié en la <a href="http://www.ua.es"> Universidad de
Alicante </a>
```

...

```
<a name="experiencia"> Experiencia laboral </a>
```

```
Tres años como programador en Java en...
```



2.4

Imágenes



Imágenes: etiqueta

Atributos

- src: URL del fichero que contiene la imagen
- alt: texto alternativo
- width, height: ancho y alto
- border: ancho del borde (*útil en enlaces*)

Este soy yo: ``

```
<a href="http://come.to/TW">  </a>
```



2.5

Formularios



Formularios: <form>

Atributos:

- **action**: URL del programa que va a tratar el formulario
- **method**: cómo se enviarán los datos (GET o POST)
- **enctype**: cómo se codifican los datos antes de enviarlos (no es lo mismo que *encriptar*)

```
<form action="login.jsp" method="get">  
  Login: <input type="text" name="login"> <br>  
  Password: <input type="password"  
  name="password"> <br>  
  <input type="submit" value="entrar">  
</form>
```



Envío de los datos

```
<form action="login.jsp" method="get">  
  Login: <input type="text" name="login" value="guest"> <br>  
  Password: <input type="password" name="password"> <br>  
  <input type="submit" value="entrar">  
</form>
```

- Los datos se envían como una secuencia de pares **dato**(name) y **valor**(value)



GET vs. POST

- GET envía los datos en la URL de la petición HTTP
`login.jsp?login=pepe&password=hola%20mami&papí`

- Los datos se ven en la barra del navegador
- La longitud no puede ser muy larga
- Permite llamar a parámetros mediante enlaces

- POST envía los datos en el cuerpo de la petición



Controles de formulario

- La mayor parte se definen con `<input>`
- El atributo `<type>` indica el tipo de control
- Controles básicos
 - Cuadro de texto de 1 línea: `type="text"`
 - Cuadro de *password*: `type="password"`
 - No se muestra en pantalla: `type="hidden"` (para cuando se necesita enviar un parámetro fijo con el que no queremos "marear" al usuario)
 - Botones de acción
 - Enviar formulario: `type="submit"`
 - Borrar controles: `type="reset"`



Controles (II): Casillas de verificación

- `type="checkbox"`
- Todos los del mismo grupo tienen el mismo **name**

```
<form action="user.jsp" method="get">
  Describa aficiones: <br>
  <input type="checkbox" name="aficion" value="cine" checked> Cine
  <input type="checkbox" name="aficion" value="alcohol"> Alcohol y
  otros
  <input type="submit" value="Enviar">
</form>
```

Describa aficiones:
 Cine Alcohol y otros

`user.jsp?aficion=cine`



Controles (III): Botones de radio

■ `type="radio"`

■ Todos los del mismo grupo tienen el mismo **name**

```
<form action="user.jsp" method="get">
  <input type="radio" name="voto" value="enrique" checked> Enrique
  Iglesias
  <input type="radio" name="voto" value="manolo"> Manolo Kabezabolo
  <input type="submit" value="¡Vota!">
</form>
```

Enrique Iglesias Manolo Kabezabolo

<user.jsp?voto=enrique>



Tecnologías  Web

Controles (IV): cuadros de lista

■ `<select>`: cuadro de lista o lista desplegable

– Atributos

- **multiple**: si está presente, se puede seleccionar más de una opción
- **size=tamaño**: número de opciones simultáneas en pantalla (si tamaño=1, **lista desplegable**)

■ `<option>`: define cada opción

– Atributos

- **value**: si no está presente, el *value* es el contenido de `<option>`
- **selected**: seleccionada por defecto



Tecnologías  Web

Controles (IVb) cuadros de lista

```
<form action="user.jsp" method="get">
  <select name="edad" size="1">
    <option value="1"> menor de 18 </option>
    <option value="2" selected> 18-25 </option>
    <option value="3"> mayor de 25 </option>
  </select>
  <input type="submit" value="Enviar">
</form>
```



A screenshot of a web form. It features a dropdown menu with three options: 'menor de 18', '18-25', and 'mayor de 25'. The '18-25' option is selected. To the right of the dropdown is a button labeled 'Enviar'.

<user.jsp?edad=2>



Tecnologías  Web

2.6

Tablas




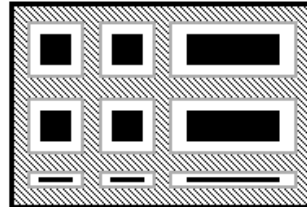
Tecnologías  Web


Tablas: <table>

Algunos atributos:


- **width**: ancho en *pixeles* o en *porcentaje* de ventana (%)
- **border**: ancho del borde
- **cellspacing**: espacio entre celdas
- **cellpadding**: espacio entre limite de celda y texto en ella

Borde de la tabla 



Cellspacing 

Cellpadding 

Contenido de la celda 



Tecnologías  Web

Tablas (II)

▮ <tr>: define una fila de datos

▮ <td>: define una celda de datos

- Algunos atributos
 - **width**: ancho en *pixeles* o en *porcentaje* de tabla (%)
 - **height**: alto mínimo
 - **rowspan**: filas abarcadas por la celda
 - **colspan**: columnas abarcadas por la celda



Tecnologías  Web

Tablas (III)

```
<table border="3" cellspacing="10" cellpadding="5">
<tr>
  <td colspan="2">Temas</td>
  <td> Sesiones </td>
</tr>
<tr>
  <td> 1 </td>
  <td> HTTP </td>
  <td> 1 </td>
</tr>
<tr>
  <td> 2 </td>
  <td> Applets Java </td>
  <td> 3 </td>
</tr>
</table>
```

	Temas	Sesiones
1	HTTP	1
2	Applets Java	3



2.7

Una introducción acelerada a XML



HTML

■ Un lenguaje de marcado, con sintaxis y semántica

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0  
Transitional//EN">  
  
<html>  
<head>  
  <title>Cutrepagina HTML</title>  
</head>  
<body>  
<h1> Bienvenido a mi página </h1>  
<p>  
En esta <b> maravillosa </b> página web vas a encontrar...  
</p>  
</body>  
</html>
```



Un problema del HTML

■ Es demasiado genérico ⇒ las etiquetas no tienen un significado específico para codificar **nuestra** información

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">  
<html>  
<head></head>  
<body>  
<h1>Octubre 2006 </h1>  
<p>  
  O sea <br>  
  Enrique Iglesias  
</p>  
<p>  
  Programación de aplicaciones web <br>  
  John McCarra <br> Frank Webini  
</p>  
</body>  
</html>
```



XML

- Una forma de definir lenguajes de marcado, con sintaxis. La semántica la da el usuario

```
<?xml version="1.0" encoding="ISO-8859-1" standalone="no" ?>
<!DOCTYPE novedades SYSTEM "novedades.dtd">
<novedades>
<fecha>Octubre 2006 </fecha>
<disco genero="pop">
  <titulo>O sea</titulo>
  <interprete>Enrique Iglesias</interprete>
</disco>
<libro>
  <titulo>Programación de aplicaciones web</titulo>
  <autor>John McCarra</autor><autor>Frank Webini</autor>
</libro>
</novedades>
```



XML “correcto”

- Bien formado

- 1ª línea declaración de doc. xml (opcional)
`<? xml version="1.0" encoding="ISO-8859-1" ?>`
- 1 sola etiqueta raíz
- Se distingue minúsculas-mayúsculas
- Las etiquetas deben abrirse y cerrarse
`<interprete> Enrique Iglesias </interprete>`
`<top10/>`
- Valores de atributos siempre entre comillas
`<top10 puesto="5"/>`

- Válido

- Cumple con una determinada gramática
- Las gramáticas XML se pueden definir con lo que se llama un DTD (Document Type Definition)



Ejemplo DTD

```
<!ELEMENT novedades (fecha,(disco|libro)*)>
<!ELEMENT fecha (#PCDATA)>
<!ATTLIST disco genero (rock|pop|melodica|clasica|dance)
#REQUIRED >
<!ELEMENT disco (titulo,interprete?,direccion)>
<!ELEMENT libro (titulo,autor+,direccion)>
<!ELEMENT titulo (#PCDATA)>
<!ELEMENT genero (#PCDATA)>
<!ELEMENT interprete (#PCDATA)>
<!ELEMENT autor (#PCDATA)>
<!ELEMENT direccion (#PCDATA)>
```



XHTML

Reformular el HTML siguiendo sintaxis XML

```
<?xml version="1.0" encoding=" ISO-8859-1"?>
<!DOCTYPE html
PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<title>Mi primera página XHTML</title>
</head>
<body>
  Hola mundo <br/ >
</body>
```



XML para sitios web

- **Opción 1: Las páginas son documentos XML**
 - Como el XML no tiene apariencia visual, hay que dársela con CSS

```
disco {color:red; font-size:15px}
```
- **Opción 2: se parte de documentos XML pero éstos se transforman al formato apropiado (HTML o variantes)**
 - Necesitamos un programa que realice la transformación y un lenguaje para especificar formalmente cómo hacerla.
 - CSS se sigue utilizando para dar estilo al HTML resultante
- **Opción 3: redefinir HTML en términos de XML**
 - Las etiquetas no serán nuestras, pero al menos tendremos un lenguaje más manejable que HTML

