

## Programación en Internet

### Tema 2. XML y XHTML

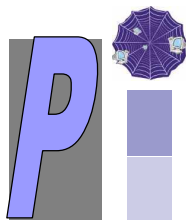
#### Contenido

1. La evolución: HTML, XML y XHTML
2. Fundamentos de XML
3. XHTML
  - 3.1 Estructura documento
  - 3.2 Enlaces
  - 3.3 Imágenes
  - 3.4 Viñetas
  - 3.5 Tablas
  - 3.6 Formularios
  - 3.7 Frames
4. Formato y Estilo: CSS
5. Transformación de XML a XHTML: XSLT
6. Diferencias entre HTML y XHTML



Universidad  
de Huelva

DEPARTAMENTO DE INGENIERÍA ELECTRÓNICA,  
SISTEMAS INFORMÁTICOS Y AUTOMÁTICA



## Programación en Internet

### Tema 2. XML y XHTML

#### Contenido

1. La evolución: HTML, XML y XHTML
2. Fundamentos de XML
3. XHTML
  - 3.1 Estructura documento
  - 3.2 Enlaces
  - 3.3 Imágenes
  - 3.4 Viñetas
  - 3.5 Tablas
  - 3.6 Formularios
  - 3.7 Frames
4. Formato y Estilo: CSS
5. Transformación de XML a XHTML: XSLT
6. Diferencias entre HTML y XHTML



Universidad  
de Huelva

DEPARTAMENTO DE INGENIERÍA ELECTRÓNICA,  
SISTEMAS INFORMÁTICOS Y AUTOMÁTICA

## La Evolución: HTML, XML y XHTML

### ■ HTML + XML = XHTML

- El HTML (HyperText Markup Language) es un lenguaje de marcas basado en SGML (Standard Generalized Markup Language).
  - Versiones: HTML 1.0, HTML 2.0, HTML 3.0, HTML 3.2, HTML 4.0
- XML (Extensible Markup Language) es un metalenguaje de marcas surgido como una formulación restringida de SGML, en la que se han eliminado muchos de los rasgos más complejos, pero conservando sus principales cualidades.
  - Versiones: XML 1.0 y XML 1.1
- XHTML (Extensible HyperText Markup Language) es la reformulación de HTML 4.x basada en XML.
  - Versiones: XHTML 1.0, XHTML 1.1 y XHTML 2.0

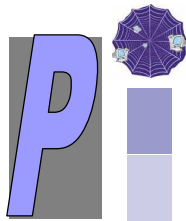
### ■ ¿ Qué es un lenguaje de marcas ?

Es un lenguaje que utiliza etiquetas para expresar que representan los elementos de un documento y como se presentarán en el navegador

```
<b> texto afectado por la marca b </b>
```

```
<a href="http://www.uhu.es"> Pincha Aquí </a>
```

```
<hr />
```



## Programación en Internet

### Tema 2. XML y XHTML

#### Contenido

1. La evolución: HTML, XML y XHTML
2. **Fundamentos de XML**
3. XHTML
  - 3.1 Estructura documento
  - 3.2 Enlaces
  - 3.3 Imágenes
  - 3.4 Viñetas
  - 3.5 Tablas
  - 3.6 Formularios
  - 3.7 Frames
4. Formato y Estilo: CSS
5. Transformación de XML a XHTML: XSLT
6. Diferencias entre HTML y XHTML



Universidad  
de Huelva

DEPARTAMENTO DE INGENIERÍA ELECTRÓNICA,  
SISTEMAS INFORMÁTICOS Y AUTOMÁTICA

## Fundamentos de XML

- XML (eXtensible Markup Language), o el lenguaje de marcas extensible, es un metalenguaje que define un estándar para la creación de lenguajes de marcas como XHTML, MathXML, VoiceXML, etc.
- El diseño de un nuevo lenguaje XML no es una tarea trivial y requiere un buen análisis del problema concreto a resolver.
- Para procesar documentos XML es necesario una aplicación que disponga de un XML parser. Muchos lenguajes de programación (como Java, C++, Python) disponen de XML parser.
- La definición de un nuevo lenguaje debe cumplir unas reglas básicas para “cumplir” con la especificación XML:
  - Debe estar asociado a un único objeto raíz
  - Debe existir etiquetas de inicio y fin para cada elemento del lenguajes  
Si la etiqueta no afecta a otro texto sería: <etiqueta />
  - Es sensible a mayúsculas y minúsculas
  - No se pueden intercalar etiquetas : <libro><paginas>XML</libro>12</paginas>
  - Los atributos de las etiquetas deben estar enrecomillados
  - Los comentarios van encerrados entre : <!-- comentario -->

## Fundamentos de XML

### ■ Ejemplo documento XML

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!-- Mi Biblioteca -->

<biblioteca>
<libro cod="0001">
  <titulo>XML en 24 horas</titulo>
  <autor>
    <nombre>Jonh Palmer</nombre>
    <apellido> Brian </apellido>
  </autor>
  <estado prestado="no"/>
</libro>

<libro cod="0002">
  <titulo>XML en 48 horas</titulo>
  <autor>
    <nombre>Juan Ramón</nombre>
    <apellido>López</apellido>
    <apellido>Prats</apellido>
  </autor>
  <isbn>8-12345-09</isbn>
  <estado prestado="si"/>
</libro>
</biblioteca>
```

## Fundamentos de XML

- DTD (Document Type Definitions) – Definición de tipos de documentos
  - Aunque se pueden crear documentos XML sin haber diseñado formalmente su sintaxis, no es aconsejable seguir esta filosofía. Lo ideal es establecer la sintaxis del lenguaje mediante un DTD, ya que esto permitirá la validación, de los documentos XML que creemos de ese lenguaje, de forma automática.
  - El DTD o sintaxis del lenguaje puede ser:
    - (1) Incluido en el propio documento
    - (2) Un fichero externo (.dtd)
    - (3) Público

```
<?xml version="1.0" ...>
<!DOCTYPE biblioteca SYSTEM
  "biblioteca.dtd" >
...
```

(2)

```
<?xml version="1.0" ... >
<!DOCTYPE biblioteca [
  aquí la sintaxis ] >
...
(1)
```

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE biblioteca PUBLIC "identificador public"
  "dirección URL" >
...
(3)
```

## Fundamentos de XML

- DTD (Document Type Definitions) – Definición de tipos de documentos
  - Ejemplo de documento XML Público: Documento XHTML

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE biblioteca PUBLIC "identificador public"
  "dirección URL" >
...
```

```
?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
  "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
...
```

**Ejemplo documento XHTML**

## Fundamentos de XML

### ■ DTD (Document Type Definitions) – Definición de tipos de documentos Creación de un DTD para un lenguaje XML

#### □ Componentes de un lenguaje XML (DTD)

- Elementos: Cada una de las partes del documento XML.

p.e. libro, título, autor, etc

**<!ELEMENT libro (autor+, título, isbn?, estado)>**

- Atributos: Propiedades de los elementos.

p.e. código libro, prestado de estado

**<!ATTLIST estado prestado (Si | No) "No">**

- Entidades: Permiten establecer acciones especiales en un documento XML.

p.e. caracteres especiales &lt; &acute;

**<!ENTITY dtd "Document Type Definitions">**

- Notaciones: Determinan como procesar un tipo de datos binarios (no XML).

**<!NOTATION gif SYSTEM "C:/windows/viewer.exe">**

## Fundamentos de XML

### ■ DTD (Document Type Definitions) – Definición de tipos de documentos Ejemplo de DTD para el documento XML anterior

**<!ELEMENT biblioteca (libro+)>**

**<!ELEMENT libro (autor+, título, isbn?, estado)>**

**<!ELEMENT autor (nombre, apellido) >**

**<!ELEMENT título ( #PCDATA ) >**

**<!ELEMENT isbn ( #PCDATA ) >**

**<!ELEMENT estado EMPTY >**

**<!ELEMENT nombre ( #PCDATA ) >**

**<!ELEMENT apellido ( #PCDATA ) >**

**<!ATTLIST libro cod ID #REQUIRED  
relacionados IDREFS #IMPLIED>**

**<!ATTLIST estado prestado (Si | No) "No">**

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!-- Mi Biblioteca -->
<biblioteca>
  <libro cod="0001">
    <título>XML en 24 horas</título>
    <autor>
      <nombre>Jonh Palmer</nombre>
      <apellido> Brian </apellido>
    </autor>
    <estado prestado="no"/>
  </libro>
  <libro cod="0002">
    <título>XML en 48 horas</título>
    <autor>
      <nombre>Juan Ramón</nombre>
      <apellido>López</apellido>
      <apellido>Prats</apellido>
    </autor>
    <isbn>8-12345-09</isbn>
    <estado prestado="si"/>
  </libro>
</biblioteca>
```

## Fundamentos de XML

### ■ DTD (Document Type Definitions) – Definición de tipos de documentos

#### Elementos

`<!ELEMENT nombre ( modelo_contenido )>`

donde *modelo\_contenido* establece la composición de un elemento:

- Lista de elementos que lo compone separados por comas. Estos pueden incluir:
  - + . Incluye una o más ocurrencias de ese elemento
  - \*. Incluye 0 o más ocurrencias del elemento
  - ? . El elemento es opcional
- #PCDATA. Indica que el elemento está compuesto por datos
- EMPTY. Indica que el elemento no contiene nada.
- ANY. Indica que contiene cualquier valor.
- | : Alternativa de contenido

Ejemplos:            `<!ELEMENT a (b, (c | d)*, e+, f?) >`  
                      `<!ELEMENT libro (autor+, titulo, isbn?, estado)>`  
                      `<!ELEMENT autor (nombre, apellido+) >`  
                      `<!ELEMENT titulo ( #PCDATA ) >`  
                      `<!ELEMENT estado EMPTY >`

## Fundamentos de XML

### ■ DTD (Document Type Definitions) – Definición de tipos de documentos

#### Atributos

`<!ATTLIST nombre_elem n_atr1 tipo_atr1 "val_atr1" ... >`

*tipo\_atr*. Puede tomar los valores:

- CDATA. El atributo es una cadena de caracteres cualquiera.
- NMTOKEN (o NMTOKENS). El atributo toma una palabra (o varias) como valor. El valor debe ser un nombre clave (no repetido).
- IDREF (o IDREFS). El valor (o valores) indica su relación con el ID de otro elemento.
- Lista de nombres. El atributo toma uno de los valores.

*val\_atr*. Puede tomar los valores

- #REQUIRED. El atributo debe tener un valor específico.
- #IMPLIED. El valor del atributo no es requerido.
- "valor". Valor por defecto para el atributo.
- #FIXED "value". Establece el valor que debe tomar el atributo.

Ejemplos:            `<!ATTLIST libro cod ID #REQUIRED relac IDREFS #IMPLIED>`  
                      `<!ATTLIST estado prestado (Si | No) "No">`

## Fundamentos de XML

### ■ DTD (Document Type Definitions) – Definición de tipos de documentos

#### Ejemplo

```
<!-- fichero biblioteca.dtd -->
```

```
<!ELEMENT biblioteca (libro+)>
<!ELEMENT libro (autor+, titulo, isbn?, estado)>
<!ELEMENT autor (nombre, apellido+) >
<!ELEMENT titulo ( #PCDATA ) >
<!ELEMENT isbn ( #PCDATA ) >
<!ELEMENT estado EMPTY >
<!ELEMENT nombre ( #PCDATA ) >
<!ELEMENT apellido ( #PCDATA ) >
<!ATTLIST libro cod ID #REQUIRED
relacionados IDREFS #IMPLIED>
<!ATTLIST estado prestado (Si | No) "No">
```

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE biblioteca SYSTEM
"biblioteca.dtd" >
```

```
<biblioteca>
...
<libro cod="0002" relac="0001">
<titulo>XML en 48 horas</titulo>
<autor>
<nombre>Juan Ramón</nombre>
<apellido>López</apellidos>
<apellido>Prats</apellidos>
</autor>
<isbn>8-12345-09</isbn>
<estado prestado="Si"/>
</libro>
...
</biblioteca>
```

## Fundamentos de XML

### ■ DTD (Document Type Definitions) – Definición de tipos de documentos

#### Ejercicio 1. Crear el DTD para este documento XML

```
<buzon>
<mensaje prioridad="urgente">
<de>Juan</de>
<a>Pepe</a>
<texto idioma="es">
Estimado amigo: ....
</texto>
</mensaje>
<mensaje>
<de>Joss</de>
<a>Pepe</a>
<texto idioma="en">
Dear colleague: ....
</texto>
</mensaje>
...
</buzon>
```

```
<!ELEMENT buzon (mensaje+)>
<!ELEMENT mensaje (de, a, texto)>
<!ATTLIST mensaje prioridad (normal | urgente) "normal">
<!ELEMENT de ( #PCDATA ) >
<!ELEMENT a ( #PCDATA ) >
<!ELEMENT texto ( #PCDATA ) >
<!ATTLIST texto idioma CDATA #REQUIRED>
```

## Fundamentos de XML

### ■ DTD (Document Type Definitions) – Definición de tipos de documentos

#### Ejercicio 2. Crear documento XML para el siguiente DTD

```
<?xml version="1.0" encoding="UTF-8"?>
<!ELEMENT agenda (persona+)>
<!ELEMENT persona (nombre, email*, telefono, relacion?)>
<!ATTLIST persona id ID #REQUIRED>
<!ATTLIST persona sexo (hombre | mujer) #IMPLIED>
<!ELEMENT nombre (#PCDATA)>
<!ELEMENT email (#PCDATA)>
<!ELEMENT telefono (#PCDATA)>
<!ELEMENT relacion EMPTY>
<!ATTLIST relacion amigo-de IDREFS #IMPLIED>
```

Conociendo que los contactos son

Juan Ruiz. juan@empresa.com juan@particular.net 555001002 amigo de Eva  
Eva Gómez (email desconocido) 555002003 (amigo no conocido)  
Manuel López manuel@empresa.net 555444000 amigos Eva y Juan

## Fundamentos de XML

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE agenda SYSTEM "agenda.dtd">
<agenda>
  <persona ID="juan" sexo="hombre">
    <nombre>Juan Ruiz</nombre>
    <email>juan@empresa.com</email>
    <email>juan@particular.net</email>
    <telefono>555001002</telefono>
    <relación amigo-de="eva" />
  </persona>
  <persona ID="eva" sexo="mujer">
    <nombre>Eva </nombre>
    <telefono>555002003</telefono>
  </persona>
  <persona ID="manu" sexo="hombre">
    <nombre>Manuel López</nombre>
    <email>manuel@empresa.net</email>
    <telefono>555444000</telefono>
    <relación amigo-de="eva juan" />
  </persona>
</agenda>
```

## Fundamentos de XML

- DTD (Document Type Definitions) – Definición de tipos de documentos

Ejercicio 3. Crear un DTD para un recetario de cocina, donde

Cada receta tiene:

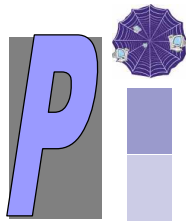
- Un atributo obligatorio que puede tomar los valores: 1plato, 2plato, postre
- Un nombre (texto)
- Varios ingredientes
- Varios pasos para su elaboración

Cada ingrediente contiene:

- Atributo con la cantidad
- Atributo con el nombre

Cada paso contiene:

- Un atributo identificador que establece el orden
- Un atributo que referencia otros pasos del cual depende éste
- Texto con la explicación



## Programación en Internet

### Tema 2. XML y XHTML

#### Contenido

1. La evolución: HTML, XML y XHTML
2. Fundamentos de XML
3. **XHTML**
  - 3.1 Estructura documento
  - 3.2 Enlaces
  - 3.3 Imágenes
  - 3.4 Viñetas
  - 3.5 Tablas
  - 3.6 Formularios
  - 3.7 Frames
4. Formato y Estilo: CSS
5. Transformación de XML a XHTML: XSLT
6. Diferencias entre HTML y XHTML



Universidad  
de Huelva

DEPARTAMENTO DE INGENIERÍA ELECTRÓNICA,  
SISTEMAS INFORMÁTICOS Y AUTOMÁTICA

## XHTML

---

- XHTML (*eXtensible HyperText Markup Language*) es una nueva versión del tradicional HTML, pero compatible con XML.
- XHTML es un lenguaje de marcas para la creación de páginas web.
- ¿Por qué usar XHTML?
  - Extensible (facilidad para ser ampliado)  
Los documentos están bien formados (los elementos anidados adecuadamente). Esto facilita el desarrollo y la integración de nuevos elementos.
  - Portabilidad  
Es una tecnología estándar propuesta por el W3C, junto con las hojas de estilo (CSS), para el desarrollo de páginas web. Por ello, todos los navegadores cumplirán estos estándares.

## XHTML – Estructura de un documento

---

- Elementos necesarios

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

<html>

<head>
<title> Aquí va el título de la página </title>
</head>

<body>
  <p> Aquí va el contenido de la página web </p>
  <p> en párrafos con texto, tablas, imágenes, formularios, etc. </p>
</body>

</html>
```

## XHTML – Estructura de un documento

### ■ Elementos necesarios

```
<?xml version="1.0" encoding="Codificación"?>
```

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

Establece el juego de caracteres utilizado para elaborar el documento XHTML.

Por defecto: UTF-8

Otros posibles valores

- UTF 16
- US-ASCII
- ISO-8859-1

```
</html>
```

## XHTML – Estructura de un documento

### ■ Elementos necesarios

```
<?xml version="1.0" encoding="Codificación"?>
```

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

Establece el tipo de documento, especificando el DTD que contiene las reglas para crear documentos bien formados.

En XHTML 1.0 existen tres tipos:

**XHTML 1.0 Strict.** Documentos con sintaxis estricta para XHTML 1.0

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

**XHTML 1.0 Transitional.** Documentos que utilizan etiquetas obsoletas

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

**XHTML 1.0 Frameset.** Documentos que utilizan Frames para organizar la web.

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

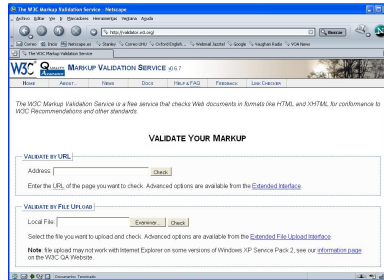
## XHTML – Estructura de un documento

### ■ Elementos necesarios

```
<?xml version="1.0" encoding="Codificación"?>  
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"  
http://www.w3.org/TR/xhtml1/DTD/xhtml1transitional.dtd">
```

Validación de documentos bien formados:

<http://validator.w3.org/>



## XHTML – Estructura de un documento

### ■ Elementos necesarios

```
1: <?xml version="1.0" encoding="Codificación"?>  
2: <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"  
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
```

3: <html>

4: <head>

5: <title> Aquí va el título de

6: </head>

7: <body>

8: <p> Aquí va el contenido

<p> en párrafos con text

10: </body>

11: </html>

Elemento raíz.

Se puede indicar el espacio de nombres de un documento XHTML y el idioma.

```
<html xmlns="http://www.w3.org/1999/xhtml"  
xml:lang="es" lang="es">
```

Otros idiomas: en (inglés), fr (frances), etc.

## XHTML – Estructura de un documento

### ■ Elementos necesarios

```
<?xml version="1.0"
<!DOCTYPE html PUBLIC
"html"
">
<html>
  <head>
    <title> Aquí va el título de la página </title>
  </head>
  <body>
    <p> Aquí va el
    <p> en párrafos
  </body>
</html>
```

La cabecera `<head> ... </head>`

Es obligatorio un elemento *title* que establece el título que mostrará el navegador cuando se visualice la página.

Aunque no obligatorio para XHTML 1.0, pero si en la versión 1.1 y sucesivas, la cabecera debe incluir un elemento para indicar el fichero con el formato (hoja de estilo) del documento.

```
<link href="fichero.css" rel="stylesheet" type="text/css" />
```

Más adelante profundizaremos en este elemento.

## XHTML – Estructura de un documento

### ■ Elementos necesarios

```
<?xml
<!DOCTYPE
<html
<head
<title
</head
<body>
  <p> Aquí va el contenido de la página web </p>
  <p> en párrafos con texto, tablas, imágenes, formularios, etc. </p>
</body>
</html>
```

El cuerpo o contenido del documento `<body>...</body>`

Contendrá la información del documento, como texto, imágenes, tablas, listas, formularios, sonido, video, etc.).

Los elementos más simples son:

- Cabeceras o títulos: `<h1> ... </h1> ... <h6>...</h6>`
- Párrafos: `<p> ... </p>`
- Saltos de línea: `<br />`
- Barra horizontal: `<hr />`
- Comentarios `<!-- comentario -->`
- Negrilla `<b> ... </b>`
- Cursiva `<i> ... </i>`

## XHTML – Estructura de un documento

### ■ Ejemplo1.html

```
<?xml version="1.0" encoding="iso-8859-1"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html>
<head>
<title> Mi primera Página Web </title>
</head>
<body>
  <!-- Aquí comienza el documento -->
  <h1> Página web de Prueba</h1>
  <h2> Creada para Prog. en Internet </h2>
  <p> Analizaremos los <b>elementos básicos</b> de un documento </p>
  <p> El segundo párrafo tiene una <br />
    y dos líneas </p>
  <hr />
  <p>Página creada por <i>elAutor © 2005</i></p>
  <!-- Aquí finaliza el documento -->
</body>
</html>
```

## XHTML – Estructura de un documento

### ■ Ejemplo1.html



## XHTML – Enlaces

---

- Los enlaces (hipertexto) son sin lugar a dudas la esencia de la web
- Los enlaces se realizan con la etiqueta `<a ...> texto_o_imagen</a>`
- Posibilidades
  - Enlazar dentro del propio documento
    - Requiere la declaración de marcadores o anclas dentro del documento
    - Declaración de marcador: `<a id="nombre" />`
    - Enlace: `<a href="#nombre">Texto_o_Imagen</a>`
  - Enlazar entre los documentos de una misma página web  
`<a href="ruta/documento.html">Texto_o_Imagen</a>`
  - Enlazar con otros documentos de otros sitios web.  
`<a href=http://direccion.url.al/documento.html>Texto_o_Imagen</a>`
  - Enlazar con marcadores de otros documentos de otros sitios web.
    - Requiere la declaración de marcadores en el documento destino
    - `<a href=http://direccion.url.al/documento.html#marcador>Texto_o_Imagen</a>`
  - Enlazar con una dirección de correo  
`<a href="mailto:pepe@micorreo.net">Texto_o_Imagen</a>`

## XHTML – Enlaces

---

- Otros atributos de la etiqueta `<a>`
    - `title="texto de título"`  
Establece un título que será mostrado al pasar el puntero del ratón por el enlace.  
  
Ejemplo  
`<a href="http://www.uhu.es" title="Visita la UHU">Universidad de Huelva</a>`
    - `target="destino"`  
Indica donde se abrirá el nuevo documento. Si se omite se abrirá por defecto en la misma ventana.  
  
Valores posibles
      - `_blank` El documento se abre en una nueva ventana.
      - `_self` El documento se abrirá en el mismo frame.
      - `_parent` El documento se abrirá en el frame padre.
      - `_top` El documento se abrirá en la misma ventana.
- Quando se utilizan Frame es posible redirigir la salida a uno concreto.

## XHTML – Enlaces

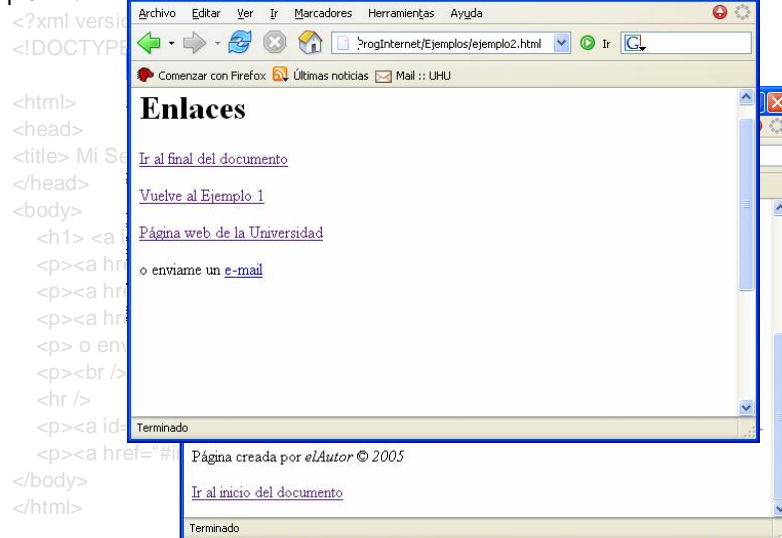
### ■ Ejemplo2.html

```
<?xml version="1.0" encoding="iso-8859-1"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">


<html>
<head>
<title> Mi Segunda Página Web </title>
</head>
<body>
  <h1> <a id="inicio" />Enlaces</h1>
  <p><a href="#final" title="bajar">Ir al final del documento</a></p>
  <p><a href="Ejemplo1.html">Vuelve al Ejemplo 1</a></p>
  <p><a href="http://www.uhu.es">Página web de la Universidad</a></p>
  <p> o enviame un<a href="mailto:elAutor@correo.es">e-mail</a></p>
  <p><br /><br /><br /><br /><br /><br /><br /><br /><br /></p>
  <hr />
  <p><a id="final" />Página creada por <i>elAutor © 2005</i></p>
  <p><a href="#inicio" title="subir">Ir al inicio del documento</a></p>
</body>
</html>
```

## XHTML – Enlaces

### ■ Ejemplo2.html



## XHTML – Imágenes

- La inclusión de imágenes hace más atractiva una web, sin embargo también la hace más “pesada”.
  - Posibles formatos:
    - GIF. 256 colores. Es posible incluir GIF's animados
    - JPG. Miles de colores. Utilizada para fotos con calidad
    - PNG. Poco utilizada, aunque es un formato con muchas posibilidades
  - Editores gráficos
    - GIMP (the GNU Image Manipulation Program) <http://www.gimp.org/>
    - Paint Shop Pro. <http://www.core.com>
    - Adobe Photoshop. <http://www.adobe.com>
  - Ms Gif Animator. [http://www.jhepple.com/gif\\_animator.htm](http://www.jhepple.com/gif_animator.htm)
  - Animation Shop. <http://www.core.com>
- 
- Recursos
  - En Internet existen miles de recursos para páginas web.

## XHTML – Imágenes

- Insertar una imagen: etiqueta `<img ... />`

### Atributos:

src: ruta al fichero con la imagen

alt: texto alternativo en caso de cargar la imagen

width; ancho en pixel o en porcentaje (%) de la imagen al ser presentada

height: alto en pixel o en porcentaje (%) de la imagen en la presentación

### Ejemplos

```

```

```

```

- Incluir una imagen como enlace.

```
<a href="http://www.uhu.es" title="UHU">
```

```

```

```
</a>
```

## XHTML – Viñetas

### ■ Existen dos tipos de viñetas:

- Listas: `<ul> ... </ul>`
  - Elementos `<li> ... </li>`
- Listas enumeradas: `<ol> ... </ol>`
  - Elementos `<li> ... </li>`
- Listas de definición: `<dl> ... </dl>`
  - Elementos: Término: `<dt> ... </dt>`  
Definición: `<dd> ... </dd>`

#### Ejemplos

```
<p>Lista</p>
<ul> <li> uno </li> <li> dos </li> <li> tres </li> </ul>
<p>Lista ordenada</p>
<ol> <li>Primero</li> <li> Segundo </li> <li> Tercero </li> </ol>
<p>Lista definición</p>
<dl> <dt> Uno </dt> <dd> El primero </dd>
      <dt> Dos </dt> <dd> El segundo </dd>
      <dt> Tres </dt> <dd> El Tercero </dd>
</dl>
```

#### Lista

- ◆ uno
- ◆ dos
- ◆ tres

#### Lista ordenada

1. Primero
2. Segundo
3. Tercero

#### Lista definición

Uno  
El primero

Dos  
El segundo

Tres  
El Tercero

## XHTML – Viñetas

### Ejemplo: Lista anidada

```
<dl>
<dt>Apache</dt> <dd>Servidor Web
  <ol> <li>Instalación y Configuración
    <ul> <li>Buscar </li>
        <li>instalar </li>
        <li>configurar </li>
    </ul> </li>
    <li>Utilización
      <ul><li>Iniciar </li>
        <li>crear pág inicial </li>
        <li>navegador </li>
      </ul> </li>
  </ol> </dd>
<dt>Tomcat</dt>
  <dd>Contenedor de Servlets</dd>
</dl>
```

```
Apache
  Servidor Web
    1. Instalación y Configuración
      ◇ Buscar
      ◇ instalar
      ◇ configurar
    2. Utilización
      ◇ Iniciar
      ◇ crear pág inicial
      ◇ navegador
Tomcat
  Contenedor de Servlets
```

## XHTML – Tablas

- Las tablas permiten presentar información de forma estructurada
- Elementos básicos

`<table> ... </table>`: Define una tabla

`<caption> ... </caption>`: Asigna un título a una tabla

`<tr> ... </tr>`: Incluye una fila en una tabla

`<td> ... </td>`: Incluye una celda en una tabla

`<th> ... </th>`: Incluye celda de cabecera

- Ejemplo

```
<table border = "1">
```

```
<tr>
```

```
<td>Cell A</td>
```

```
<td>Cell B</td>
```

```
</tr>
```

```
</table>
```

Cell A	Cell B
--------	--------

## XHTML – Tablas

- Ejemplo

```
<table border="1">
```

```
<caption>Artículos en Stock</caption>
```

```
<tr>
```

```
<th>Código</th>
```

```
<th>Descripción</th>
```

```
</tr>
```

```
<tr>
```

```
<td>PC-123001</td>
```

```
<td>Pentium 4. 80HDD</td>
```

```
</tr>
```

```
<tr>
```

```
<td>PT-000101</td>
```

```
<td>Portátil Centrino 1.4. 40HDD</td>
```

```
</tr>
```

```
</table>
```

Artículos en Stock

Código	Descripción
PC-123001	Pentium 4. 80HDD
PT-000101	Portátil Centrino 1.4. 40HDD

## XHTML – Formularios

---

- Permiten recoger información introducida por el usuario para enviarla al servidor.

`<form ... > ... </form>` : Define un formulario.

### Atributos

action: URL donde se envían los datos.    *action="comprar.jsp"*  
method: Tipo de petición (get o post).    *method="get"*

### Contenido

`<input ... />` : Incluye elementos en el formulario

`<label> ... </label>` : Asigna etiquetas a los elementos

`<textarea ... > ... </textarea>` : incluye areas de texto

`<select> ... </select>` : Incluye casillas de verificación

## XHTML – Formularios - Contenido

---

- Etiqueta `<input ... />`

### Atributos

type: Tipo de elemento a incluir (campo texto, botón, password, etc)  
name: Nombre  
id: Identificador  
value: Valor por defecto. Etiqueta para los botones  
size: Tamaño

- Campo de texto y contraseña con etiquetas

`<label for="cliente">Nombre:</label>`

`<input type="text" id="cliente" name="cliente" size="20" /> <br />`

`<label for="clave">Clave:</label>`

`<input type="password" name="clave" id="clave" size="20" /> <br />`

- Botones de para enviar y restablecer

`<input type="submit" value="Enviar" />`

`<input type="reset" value="Borrar" />`

## XHTML – Formularios - Contenido

### ■ Ejemplo

```
<form action="validar.jsp" method="post">
  <label for="cliente">Nombre:</label>
  <input type="text" id="cliente" name="cliente" size="20" /> <br />
  <label for="clave">Clave:</label>
  <input type="password" name="clave" id="clave" size="20" /> <br />

  <input type="submit" value="Enviar" />
  <input type="reset" value="Borrar" />
</form>
```

Cliente:

Clave :

## XHTML – Formularios - Contenido

### Componentes de un Formulario mediante ejemplos

#### ■ Área de texto

```
<p> <label for="comentario">Comentario:</label><br />
<textarea name="comentario" id="comentario" cols="30" rows="5">
Texto por defecto
</textarea> </p>
```

#### ■ Casillas de verificación

```
<p> Selecciona los lenguajes de programación que conoces: <br />
<input type="checkbox" name="java" id="java" value="si" checked="checked" />
<label for="java">Java</label> <br />
<input type="checkbox" name="C++" id="C++" value="si" />
<label for="C++">C++</label><br />
<input type="checkbox" name="PHP" id="PHP" value="si" />
<label for="PHP">PHP</label><br />
</p>
```

## XHTML – Formularios - Contenido

### ■ Botones de selección

```
<p> Selecciona el lenguaje de programación que más te gusta: <br />
<input type="radio" name="leng" id="Java" value="Java" checked="checked" />
<label for="java">Java</label> <br />
<input type="radio" name="leng" id="C++" value="C++" />
<label for="C++">C++</label><br />
<input type="radio" name="leng" id="PHP" value="PHP" />
<label for="PHP">PHP</label><br />
</p>
```

### ■ Listas de Selección

```
<p>
<label for="leng">Elige lenguaje preferido:</label><br />
<select name="leng" id="leng">
  <option value="Java">Java</option>
  <option value="C++">C++</option>
  <option value="PHP" selected="selected">PHP</option>
</select> </p>
```

## XHTML – Marcos (Frames)

- Permite dividir la ventana del navegador en varias partes y presentar varios documentos XHTML.

### ■ Estructura

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Frameset//EN"
  "http://www.w3.org/TR/xhtml1/DTD/xhtml1-frameset.dtd">

<html>

<head>
<title> Título de la página </title>
</head>

<frameset ... >
  <!-- Aquí se definen los marcos (<frame ... />) o nuevos frameset -->
</frameset>

</html>
```

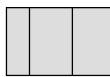
## XHTML – Marcos (Frames)

- Etiqueta `<frameset ... > ... </frameset>` : define un entorno de marcos

### Atributos: cols ó rows

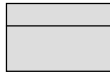
Define, separadas por comas, el número y el tamaño (en pixels, porcentajes o \*) de las columnas (cols) y las filas (rows).

### Ejemplos



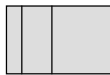
```
<frameset cols="20%,40%,40%"> ... </frameset>
```

Define tres marcos en columnas, el primero ocupará el 20% y los otros dos el 40% cada uno.



```
<frameset rows="100,*"> ... </frameset>
```

Define dos marcos en filas, el primero ocupa 100 pixels y el otro el resto.



```
<frameset cols="10%,*,2*"> ... </frameset>
```

Define tres marcos (columnas), el primero ocupa el 10% y los otros dos se reparten el resto, tomando un tercio el primero y el resto el último.

## XHTML – Marcos (Frames)

- Etiqueta `<frame ... />` . Establece el contenido y las propiedades de un marco

Atributos	Valor	Objetivo
src	URL	Define la URL (fichero) que mostrará el marco
frameborder	0 ó 1	Establece el borde del marco (0: sin borde)
name	<i>nombre</i>	Define un nombre único para el marco
noresize	noresize	Impide que el marco pueda ser redimensionado
scrolling	yes/no/auto	Establece la posibilidad de incluir barras de desplazamiento

### Ejemplos

```
<frame src="menu.html" name="Menu" noresize="noresize" scrolling="no" />
```

```
<frame src="ppal.htm" name="Ppal" scrolling="auto" />
```

```
<frame src="cualquier_url.htm" =frameborder="0" />
```

### Mostrar un enlace en un marco

```
<p> <a href="direccion_enlace.html" target="nombre_marco">Pulsa aquí</a></p>
```

## XHTML – Marcos (Frames)

### ■ Ejemplo

```
<frameset cols="20%,*">
  <frame src="menu.htm" />
  <frame src="ppal.htm" name="ppal" />
</frameset>
```

Menú	Ventana Ppal
<a href="#">Doc 1</a>	Muestra contenidos
<a href="#">Doc 2</a>	
<a href="#">Doc 3</a>	

#### Fichero menu.htm

```
...
<p><a href="documento1.html" target="ppal">Doc 1</a></p>
<p><a href="documento2.html" target="ppal">Doc 2</a></p>
<p><a href="documento3.html" target="ppal">Doc 3</a></p>
...
```

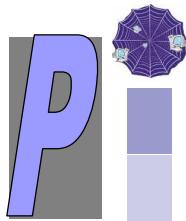
#### Fichero ppal.htm

```
<h3>Ventana Ppal</h3>
<p>Muestra contenidos</p>
```

#### Fichero documento1.html

```
<h3>Documento 1</h3>
<p>Este es el contenido <br />
del documento 1</p>
```

Menú	Documento 1
<a href="#">Doc 1</a>	Este es el contenido del documento 1
<a href="#">Doc 2</a>	
<a href="#">Doc 3</a>	



## Programación en Internet

### Tema 2. XML y XHTML

#### Contenido

1. La evolución: HTML, XML y XHTML
2. Fundamentos de XML
3. XHTML
  - 3.1 Estructura documento
  - 3.2 Enlaces
  - 3.3 Imágenes
  - 3.4 Viñetas
  - 3.5 Tablas
  - 3.6 Formularios
  - 3.7 Frames
4. **Formato y Estilo: CSS**
5. Transformación de XML a XHTML: XSLT
6. Diferencias entre HTML y XHTML





## CSS – Formato y Estilo

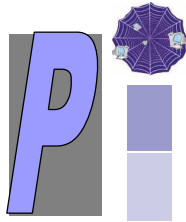
### ■ Principales propiedades

- Fondo
  - background : [color] [imagen]
  - background-color : [color]
  - background-image : [url(fichero.gif)]
  
- Texto
  - color : [color]
  - text-align : [left / right / center / justify ]
  - text-decoration : [ none / underline / overline / line-through / blink ]
  - text-transform : [ none / capitalize / uppercase / lowercase ]
  
- Tipo de letra (Font)
  - font-family : {fuente} tamaño / % ]
  - font-size : [xx-small / x-small / small / medium / large / x-large / xx-large / smaller / larger / ↓
  - font-style : [ normal / italic / oblique ]
  - font-weight : [ normal / bold / bolder / lighter / 100 – 900 ]

## CSS – Formato y Estilo

### ■ Principales propiedades

- Bordes
  - border : *border\_width border\_style border\_color*
  - border-width : [ thin / medium / thick / *valor\_deseado* ] inset / outset]
  - border-style : [ none / hidden / dotted / dashed / solid / double / groove / ridge ↓
  - border-color : [ *color* ]
  
- Márgenes
  - margin : *margin-top margin-right margin-bottom margin-left*
  - margin-bottom : [ auto / *valor-deseado* / % ]
  - margin-left : [ auto / *valor-deseado* / % ]
  - margin-right : [ auto / *valor-deseado* / % ]
  - margin-top : [ auto / *valor-deseado* / % ]
  
- Listas
  - list-style-type : [ none / disc / circle / square / decimal / decimal-leading-zero /  
lower-roman / upper-roman / lower-alpha / upper-alpha /  
lower-greek / lower-latin / upper-latin / hebrew / armenian /  
georgian / cjk-ideographic / hiragana / katakana / hiragana-iroha  
katakana-iroha ]



# Programación en Internet

## Tema 2. XML y XHTML

### Contenido

1. La evolución: HTML, XML y XHTML
2. Fundamentos de XML
3. XHTML
  - 3.1 Estructura documento
  - 3.2 Enlaces
  - 3.3 Imágenes
  - 3.4 Viñetas
  - 3.5 Tablas
  - 3.6 Formularios
  - 3.7 Frames
4. Formato y Estilo: CSS
5. **Transformación de XML a XHTML: XSLT**
6. Diferencias entre HTML y XHTML



Universidad  
de Huelva

DEPARTAMENTO DE INGENIERÍA ELECTRÓNICA,  
SISTEMAS INFORMÁTICOS Y AUTOMÁTICA

### XSLT – Transformación de XML a XHTML

- XSLT es un lenguaje de transformación de documentos XML, que junto con XPath (rutas al documento XML) y XSL-FO (lenguaje de formato para XML) conforman XSL (eXtensible Stylesheet Language) o lenguajes de estilo para XML.
- XSLT (XSL Transformations) es el más importante de los componentes del estándar XSL. Su objetivo es transformar documentos XML es otro tipo de documentos. Nuestro interés está en transformar XML a XHTML.
- XSLT utiliza XPath para recorrer el documento XML.
- Para la transformación del documento se crea un fichero de hojas de estilos (.xsl) y el documento XML que se desea transformar se le incorpora este fichero de estilos.

#### Documento XML

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!-- Mi Biblioteca -->
<biblioteca>
<libro cod="0001" >
...
```

#### Documento XML con XSLT

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<?xml-stylesheet type="text/xsl" href="fichero.xsl"?>
<!-- Mi Biblioteca -->
<biblioteca>
  <libro cod="0001" >
  ...
```

## XSLT – Transformación de XML a XHTML

### ■ Fichero .xsl

#### □ Estructura

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<xsl:stylesheet version="1.0"
  xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
  ...
</xsl:stylesheet>
```

### ■ Existen dos formas de transformar el documento XML a XHTML

#### □ Iterativa

- Se crea un *template* desde el punto del documento desde donde se desea iniciar la transformación
- Se aplican etiquetas que permiten recorrer los elementos del documento

#### □ Basada en *templates*.

- Se crean *templates* para cada elemento que se desea presentar.
- El *template* dará formato al contenido del documento XML o indicará los *templates* de los que depende.

## XSLT – Transformación de XML a XHTML

### ■ Contenido fichero .xsl

□ `<xsl:template match="XPath_EltoXML"> ... </xsl:template>`

□ `<xsl:value-of select="XPath_EltoXML" />`

□ `<xsl:for-each select="EltoXML"> ... </xsl:for-each>`

□ `<xsl:sort select="EltoXML" />`

□ `<xsl:if test="EltoXM op valor"> ... </xsl:if>`, donde *op* =, !=, &lt; ó &gt;

□ `<xsl:choose>`  
`<xsl:when test="EltoXM op valor"> ... </xsl:when>`  
...  
`<xsl:otherwise> .... </xsl:otherwise>`  
`</xsl:choose>`

□ `<xsl:apply-templates [select="XPath_XML"] />`

## XSLT – Transformación de XML a XHTML - Ejemplo

... <!-- Cabecera suprimida -->

```
<xsl:template match="/">
<html>
<body>
<h2>Mi Biblioteca Personal</h2>
<table border="1">
<tr>
<th align="left">Cod.Libro</th>
<th align="left">Titulo</th>
<th align="left">Autor</th>
<th align="center">Prestado</th>
</tr>

<xsl:for-each select="biblioteca/libro">
<tr>
<td><xsl:value-of select="@cod"/></td>
<td><xsl:value-of select="titulo"/></td>
<td>
<xsl:for-each select="autor/apellido">
<xsl:value-of select="."/>
</xsl:for-each>
,<xsl:value-of select="autor/nombre"/>
</td>
```

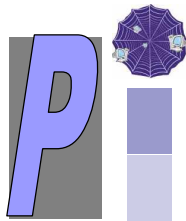
... <!-- continuación -->

```
<td>
<xsl:for-each select="estado">
<xsl:choose>
<xsl:when test="@prestado='si'" > Uff!
</xsl:when>
<xsl:when test="@prestado='no'" > Bien
</xsl:when>
</xsl:choose>
</xsl:for-each>

<xsl:for-each select="isbn">
(isbn)
</xsl:for-each>
```

### Mi Biblioteca Personal

Cod.Libro	Titulo	Autor	Prestado
0001	XML en 24 horas	Brian , Jonh Palmer	Bien
0002	XML en 48 horas	López, Prats, Juan Ramón	Uff! (isbn)



## Programación en Internet

### Tema 2. XML y XHTML

#### Contenido

1. La evolución: HTML, XML y XHTML
2. Fundamentos de XML
3. XHTML
  - 3.1 Estructura documento
  - 3.2 Enlaces
  - 3.3 Imágenes
  - 3.4 Viñetas
  - 3.5 Tablas
  - 3.6 Formularios
  - 3.7 Frames
4. Formato y Estilo: CSS
5. Transformación de XML a XHTML: XSLT
6. **Diferencias entre HTML y XHTML**



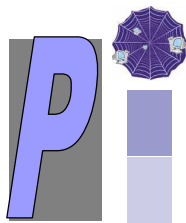
Universidad  
de Huelva

DEPARTAMENTO DE INGENIERÍA ELECTRÓNICA,  
SISTEMAS INFORMÁTICOS Y AUTOMÁTICA

## XHTML vs HTML

### ■ Principales diferencias entre XHTML y HTML

- Los documentos deben especificar el tipo de documento: DOCTYPE
- Todas las etiquetas deben ser en minúsculas
- Todas las etiquetas deben ser cerradas
- Los valores de los atributos deben estar entre comillas
- El atributo *id* reemplaza al atributo *name*.
- Las etiquetas no deben ser anidadas
- El documento debe contener sólo el contenido, el formato debe ser especificado independiente (css)
- Las tablas no deben usarse para establecer formato.



## Tema 2. XML y XHTML

### Resumen

1. La evolución: HTML, XML y XHTML
2. Fundamentos de XML
3. XHTML
  - 3.1 Estructura documento
  - 3.2 Enlaces
  - 3.3 Imágenes
  - 3.4 Viñetas
  - 3.5 Tablas
  - 3.6 Formularios
  - 3.7 Frames
4. Formato y Estilo: CSS
5. Transformación de XML a XHTML: XSLT
6. Diferencias entre HTML y XHTML

### Referencias

Libro: Capítulos 4-6 y 20. Internet & WWW: How to Program.  
Capítulos 2 y 10, Programación de Aplicaciones Web.  
Part I, Capítulo 1-5. Core Web Programming.  
HTML & XHTML: The Definitive Guide  
W3 Schools, <http://www.w3schools.com>  
Tutoriales y Manuales de Referencia de XML, XHTML y CSS



Universidad  
de Huelva

DEPARTAMENTO DE INGENIERÍA ELECTRÓNICA,  
SISTEMAS INFORMÁTICOS Y AUTOMÁTICA

