

## 2

## Presentación de contenido Parte I: HTML

**Facultad de Informática  
Universidad Complutense de Madrid**

Material elaborado por Juan Pavón Mestras, con modificaciones de  
Pablo Moreno Ger, Manuel Freire Morán, Raquel Hervás Ballesteros  
e Iván Martínez Ortiz

# HTML, XHTML, HTML5



# ¿Qué es HTML?

HTML (HyperText Markup Language): Lenguaje con el que se definen páginas Web

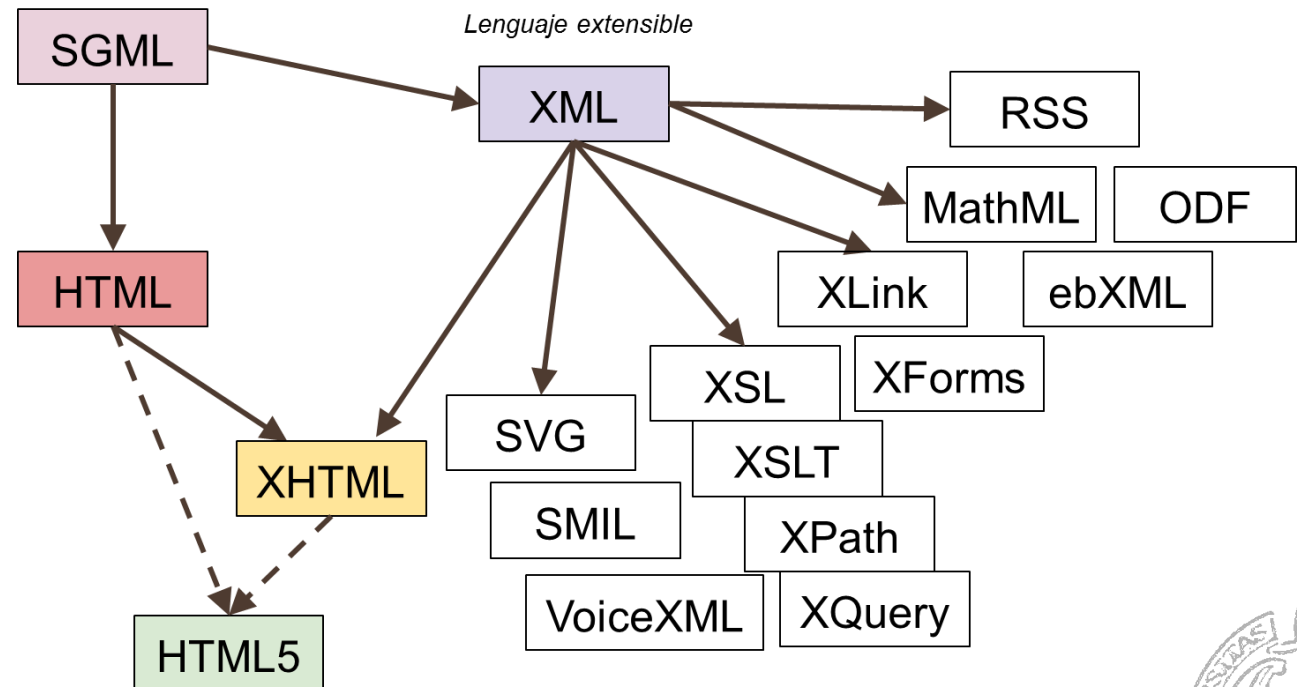
- ✓ Permite describir el contenido de una página, incluyendo texto y otros elementos
- ✓ Una página HTML consta de texto y marcas especiales que permiten indicar algún tratamiento especial
- ✓ Las marcas se indican en formato <marca>...</marca>

```
<html>
  <head>
    <title>Título de la página</title>
  </head>
  <body>
    <h1>Título de sección</h1>
    <p>Texto...</p>
    <p>Texto <b>en negrita</b></p>
  </body>
</html>
```



## HTML: Lenguaje original

- ✓ Basado en SGML (Standard Generalized Markup Language)
- ✓ Es casi un lenguaje XML, pero tiene elementos no compatibles con XML



## Evolución de HTML: Separación contenido-formato de presentación

- ✓ La gestión del estilo de todo el sitio web está localizada en un punto concreto: las hojas de estilo (CSS)
  - Es posible modificar todo el estilo del sitio (manteniendo su coherencia) sin editar todas sus páginas
- ✓ Se pueden tener varios estilos definidos para el mismo sitio web
  - Los usuarios de un sitio web pueden seleccionar el conjunto de colores y estilos (tema) del sitio web
- ✓ Soporte a la navegación desde múltiples dispositivos, adaptando las páginas a sus características (tamaño de pantalla, etc.)



## HTML 4.01 (1999)

### ✓ Dos modos:

- Strict: no permite marcado de presentación (center, font, align, bgcolor, ...)
- Transitional: permite marcado de presentación

### ✓ Framesets:

- Primera incorporación al estándar de la etiqueta <frameset> (aunque ya se usaba comúnmente en Netscape)
- Y última (desaparecen en HTML5)



## XHTML (2000): versión basada en XML de HTML 4.01

- ✓ Compatibilidad con otros documentos XML
  - Combinar con otros documentos en lenguajes basados en XML
  - Usar herramientas (XSLT)
  
- ✓ XHTML es más estricto que HTML
  - Exige que los documentos estén bien formados (requisito de XML)
    - Por ejemplo: todos los elementos tienen que estar bien anidados y tener siempre la etiqueta de cierre
  - Menos recursos para procesar los documentos
    - Ventaja para hacer navegadores más eficientes



## HTML5: Evolución de XHTML y HTML 4.01

- ✓ Estándar oficial desde octubre de 2014
  - XHTML 2.0 ha sido abandonado
  - Desarrollado por el WHATWG, adoptado por el W3C
    - Grupo de trabajo informal formado por individuos y empresas (Mozilla, Opera, Apple, Google entre ellas) y dedicado al desarrollo del HTML
- ✓ Un paso más
  - Construye sobre 'strict'; acaba con 'frameset' y 'transitional'
  - Añaden cosas y recomiendan no usar otras, pero quitan poco o nada
- ✓ HTML5 vs. XHTML
  - La recomendación vigente es codificar en HTML5 (la transición hacia navegadores optimizados para XML puro no ha funcionado).





En este curso vamos a trabajar directamente siguiendo la especificación HTML5



## HTML(5): Conceptos generales



La idea básica de los lenguajes de marcado es estructurar el contenido mediante marcas o etiquetas (tags)

- ✓ Con cada etiqueta se pueden asociar varios atributos
- ✓ Así se puede especificar distintas características de formato, de tipo de información, etc. que pueda ser procesada por un programa (por ejemplo, por un navegador)

Ejemplos:

`<oscuro>Este texto va en negrita</oscuro>`

- ✓ **Este texto va en negrita**

Esta `<importante prioridad="alta">palabra</importante>` es fundamental

- ✓ Esta *palabra* es fundamental



## Estructura general

- ✓ Todo documento HTML comienza por la etiqueta `<html>` y acaba con `</html>`
- ✓ Todo documento HTML tiene dos partes: Cabecera y Cuerpo

<b>Cabecera</b>	<pre>&lt;html&gt; &lt;head&gt;   &lt;title&gt;Título de la página&lt;/title&gt; &lt;/head&gt;</pre>
<b>Cuerpo</b>	<pre>&lt;body&gt;   &lt;h1&gt;Título de sección&lt;/h1&gt;   &lt;p&gt;Texto...&lt;/p&gt;   &lt;p&gt;Texto &lt;b&gt;en negrita&lt;/b&gt;&lt;/p&gt; &lt;/body&gt;</pre>
	<pre>&lt;/html&gt;</pre>



## El elemento <html>

- ✓ Puede servir para indicar el idioma del texto
  - `<html lang="en">`
  - `<html lang="en-GB">`
  - `<html lang="es">`
- ✓ Esto es importante para mejorar la accesibilidad del documento
- ✓ Si en alguna parte del documento se utiliza otro idioma se puede indicar en el elemento contenedor correspondiente
  - `<p lang="fr">Ceci est un paragraphe.</p>`
  - `<p>Buenos días en francés: <span lang="fr"> Bonjour</span></p>`



# Documento HTML

Declaración que indica que el documento es HTML5

```
<!DOCTYPE html>
```

```
<html>
```

```
  <head>
```

```
    <title>Título del documento</title>
```

```
  </head>
```

```
  <body>
```

```
    <p>Ejemplo de documento HTML5</p>
```

```
  </body>
```

```
</html>
```

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



## Etiquetas emparejadas y sin emparejar:

```
<h1>Introducción</h1>  
<p>Esto es um párrafo</p>  
<hr />  
<link href="estilos.css" rel="stylesheet" type="text/css" />
```

HTML permite que las etiquetas sin emparejar no lleven cierre (/)

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

En HTML las etiquetas se pueden escribir en mayúsculas o minúsculas, indistintamente

- ✓ Aunque se recomienda escribir las etiquetas en minúsculas

```
<h1>Título <Em>1</eM></H1>
```



Un elemento puede tener una etiqueta con atributos, separados por espacios, y que se especifican como pares atributo="valor"

```
<etiqueta atributo="valor"> texto </etiqueta>
```

- ✓ En XHTML el valor siempre tiene que ir entre comillas
- ✓ En HTML no es obligatorio, pero muy recomendado.

Los atributos permiten añadir información adicional

- ✓ Por ejemplo, un hiper-enlace:

```
<a href="http://www.ucm.es/">Universidad Complutense de Madrid</a>
```





Se pueden asociar a cualquier etiqueta

- ✓ `id = "texto"`
  - Asigna un identificador único al elemento dentro de la página
- ✓ `class = "texto"`
  - Asocia una clase CSS que define un conjunto de estilos de formato para el elemento
- ✓ `style = "texto"`
  - Establece de forma directa los estilos CSS del elemento
- ✓ `title = "texto"`
  - Asigna un título a un elemento
  - Es útil para mejorar la accesibilidad ya que los navegadores lo muestran cuando el usuario pasa el ratón por encima del elemento
- ✓ `contenteditable = "true | false"`
  - Indica si el contenido de un elemento puede ser editado

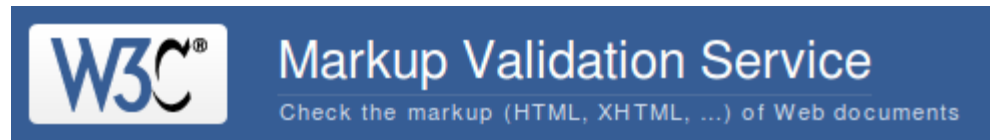


## Conviene validar el código HTML5

- ✓ Elementos obligatorios (como <title>)
- ✓ Etiquetas abiertas que no se cierran
- ✓ Etiquetas mal anidadas
- ✓ Atributos necesarios para elementos (como un <img> sin src)
- ✓ Elementos en lugares incorrectos (como texto en la cabecera)

## Validador oficial de W3C

- ✓ <http://validator.w3.org>



También se debe comprobar qué características funcionan en los distintos navegadores

- ✓ <http://caniuse.com>



## HTML: Elementos en la cabecera



## <head>

- ✓ Delimita la cabecera del documento
- ✓ En la cabecera se describe información del documento (título, meta-información, scripts, estilos)

## <title>

- ✓ Indica el título del documento
- ✓ Es “obligatoria” y tiene que aparecer una y sólo una vez en el documento
- ✓ El navegador lo visualiza en la barra de título de su ventana (o en la pestaña)



## <link />

- ✓ Enlace a otros archivos (generalmente hojas de estilo)

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

- ✓ Separación del contenido y el estilo

- Se puede reutilizar el fichero de estilo para todas las páginas de un sitio
- Código HTML más claro

- ✓ Se pueden definir distintos estilos según el propósito

- Un estilo para visualizar en el navegador y otro para imprimir la página

```
<link rel="stylesheet" media="screen" href="estilo.css" >  
<link rel="stylesheet" media="print" href="impresora.css" >
```



## Metadatos (<meta />)

- ✓ Meta-información sobre el documento (información procesable automáticamente por programas que analicen la página)
- ✓ Normalmente se usan los atributos *name* (para definir un tipo de metadato) y *content* (para definir el valor)
- ✓ Atributos habituales:

```
<meta name="author" content="Raquel" />  
<meta name="robots" content="index, follow" />  
<meta name="keywords" content="aplicaciones web, html" />  
<meta name="description" content="Curso sobre diseño web" />
```



## <script>

- ✓ Permite incluir código de script
- ✓ Aunque por defecto se espera JavaScript, conviene especificarlo

## <noscript>

- ✓ Indica un mensaje a mostrar en navegadores que no pueden procesar <script>

```
<script type="text/javascript">
  //Código JavaScript
</script>
<noscript>
  <p>Esta página requiere el uso de JavaScript. Por favor
  compruebe la configuración de su navegador.</p>
</noscript>
```

**NOTA:** El atributo type ya no es obligatorio, ni recomendable para Javascript



## <script>

- ✓ También puede referirse a un fichero externo

```
<script src="http://www.ejemplo.com/js/miscript.js"></script>
```





## <style>

- ✓ Permite definir propiedades de estilos que se aplicarán a lo largo del documento
- ✓ En lugar de esto generalmente se indica la hoja de estilo que se va a aplicar (con <link>)
- ✓ Veremos la sintaxis detallada más adelante

```
<style type="text/css">
p {
  font-family: Arial;
  font-size: 10pt;
  color: rgb(0,0,128);
  text-indent: 15px;
  text-align: justify;
  margin-left: 10px;
}
</style>
```



## HTML: Elementos para texto



El texto de un documento debe estar contenido por alguna de las etiquetas de una de las tres categorías siguientes:

- ✓ Etiquetas de bloque
  - Pueden aparecer directamente dentro de `<body>`
  - No se deben anidar
  - `<p>` `<pre>` `<h1>` `<h2>` `<h3>` `<h4>` `<h5>` `<h6>` `<address>`
- ✓ Etiquetas *inline*:
  - Se usan dentro de los bloques
  - No deben aparecer fuera de los bloques
  - Afectan a una parte de texto dentro de un bloque
  - `<br />` `<abbr>` `<cite>` `<code>` `<em>` `<kbd>` `<strong>` `<sub>` `<sup>`
- ✓ Etiquetas de contenedores de bloques:
  - Sirven para estructurar el texto y definir agrupaciones de bloques
  - Pueden contener etiquetas de bloque u otros contenedores anidados
  - `<body>`
  - `<blockquote>`, `<div>`, `<article>`, `<header>`, `<section>`, `<aside>`, `<nav>`, `<footer>`



## HTML: Etiquetas de bloque



## <p>

- ✓ Contiene el texto de un párrafo
- ✓ El navegador no muestra los espacios en blanco ni los saltos de línea dentro de un párrafo

```
<p>Esto es un párrafo. El contenido aparecerá partido en varias  
líneas según el tamaño de la ventana, pero es independiente de  
los  
saltos de línea o del uso de      espacios adicionales</p>
```

## <pre>

- ✓ Texto preformateado: igual que <p> pero sí se tienen en cuenta espacios en blanco y líneas en blanco
- ✓ Se usa un tipo de letra de ancho fijo



<h1> <h2> <h3> <h4> <h5> <h6>

- ✓ Encabezado (de nivel 1 a 6)

```
<h1>Sección 1</h1>
<h2>Sección 1.1</h2>
<p>Texto normal.</p>
<h3>Sección 1.1.1</h3>
<p>Texto normal.</p>
```

<address>

- ✓ Información de contacto del autor de la página

```
<address>Autor: Juan Pavón (UCM) <br />
Última modificación: 10 de enero de 2013
</address>
```



## HTML: Etiquetas *inline*



`<br />`

- ✓ Salto de línea
- ✓ **NO** se debe usar

`&nbsp;`

- ✓ Espacio en blanco
- ✓ Para forzar espacios que si no ignoraría el navegador
  - Poco aconsejable, mejor usar `<pre>`
- ✓ Para poner un blanco en el que el navegador no rompa las líneas
- ✓ **NO** se debe usar

`<hr />`

- ✓ Línea horizontal (horizontal rule)
- ✓ Se usa cada vez menos ya que con CSS se pueden crear mejor los bordes





## Importancia del texto

### ✓ <em>

- Para resaltar una porción de texto dándole énfasis (por defecto aparecerá en cursiva)

### ✓ <strong>

- Mayor énfasis que con <em> (por defecto aparecerá en negrita)

### ✓ En desuso

- Antes se usaba <i> para cursiva y <b> para negrita
- Ahora tienen un nuevo significado (voz alternativa o atención especial)



## Tipo de información (relativamente poco comunes)

### ✓ <abbr>

- Abreviatura

```
<abbr title="etcétera">etc.</abbr>  
<abbr title="Universidad Complutense de Madrid">UCM</abbr>
```

### ✓ <cite>

- Para incluir una referencia o cita (por defecto aparecerá en cursiva)

```
<p>Como dice el refrán, <cite>A la tercera va la  
vencida</cite>. </p>
```

### ✓ <dfn>

- Definición de un término (por defecto aparecerá en cursiva)



## Tipo de información

### ✓ <code>

- Parecido a <pre>, pero para usar en mitad de un bloque
- Fragmento de código de programa (por defecto aparecerá en letra monospace)
- Los saltos de línea y espacios dentro de <code> se ignoran

```
<p>La función se escribe poniendo <code>main() { printf  
("Hola, mundo"); }</code> en el archivo fuente.</p>
```



## Para indicar edición del texto

### ✓ <del>

- Para mostrar que se elimina un texto (por defecto aparecerá tachado)

### ✓ <ins>

- Para mostrar que se ha insertado un texto (por defecto aparecerá subrayado)

```
<p>La nota final es <del>suspenseo</del>  
<ins datetime="20130625">aprobado</ins>.</p>
```

- La nota final es ~~suspenseo~~ aprobado.



## Modificación de aspecto

- ✓ Las siguientes etiquetas existían en HTML, pero han sido eliminadas de HTML5:
  - `<big>` para etiquetar texto de mayor tamaño
  - `<tt>` para etiquetar texto de fuente de espaciado fijo
- ✓ Las siguientes etiquetas se usan habitualmente
  - `<sub>` para identificar texto en subíndice: `H<sub>2</sub>O`  
→  $H_2O$
  - `<sup>` para identificar texto en superíndice: `x<sup>n</sup>`  
→  $x^n$



# Agrupación de texto

## <span>

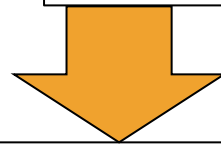
- ✓ Se usa para dar formato con la hoja de estilo a un grupo de elementos en línea seguidos dentro de un mismo bloque (por ejemplo, varias palabras seguidas dentro de un párrafo)
- ✓ Se suele asociar a clases definidas en un fichero de estilos (.css)

### Fichero .css

```
span.feroz {  
  font-style: italic;  
  text-decoration: underline;  
}  
span.rojo {  
  color: red;  
}
```

### Fichero .html (incluye al .css)

```
<p>Andaba sola  
<span class="rojo">Caperucita  
Roja</span>  
y se encontró  
<span class="feroz">al lobo  
feroz</span></p>
```



Andaba sola **Caperucita Roja** y se encontró al lobo feroz



## HTML: Etiquetas de contenedores



## <blockquote>

- ✓ Para incluir una cita larga, que puede contener varios párrafos u otras etiquetas
- ✓ Se suele mostrar el texto dentro de esta etiqueta con márgenes a izquierda y derecha
  - Lo mejor es definir el formato deseado en la hoja de estilo
- ✓ Se puede indicar el origen de la cita con el atributo cite

```
<p>El candidato a la presidencia fue locuaz en su discurso:</p>  
<blockquote cite="http://www.buenasnoticias.org/entrevista12.html">  
<p>Prometo que voy a respetar los servicios sociales, naturalmente.  
Miente quien infunda alguna sospecha al respecto. </p>  
<p>Y bla bla bla...</p>  
</blockquote>
```





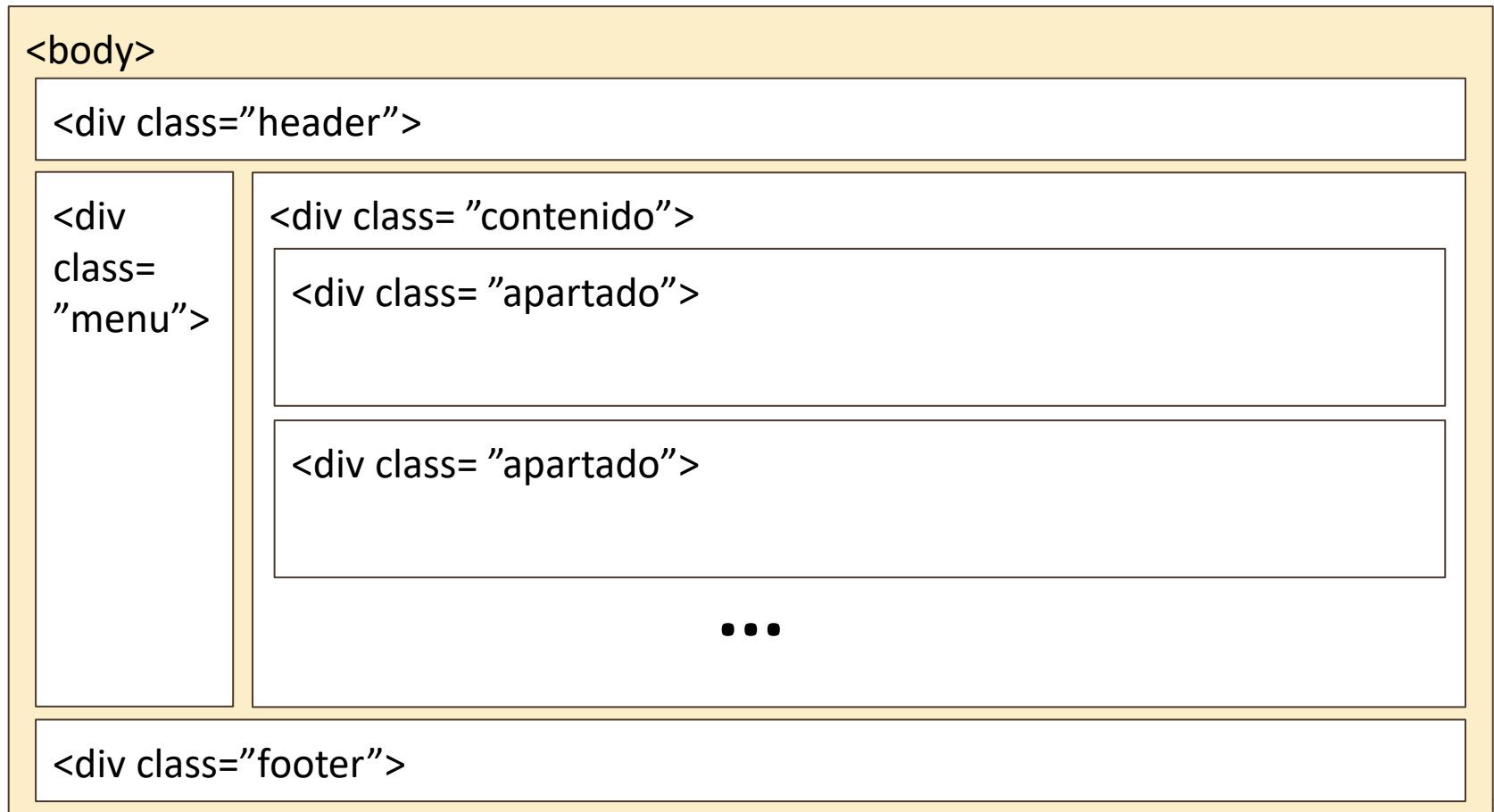
## <div>

- ✓ División: mecanismo más importante para agrupar diversos elementos de bloque (párrafos, encabezados, listas, tablas, divisiones, etc.)
- ✓ El formato hay que definirlo en una hoja de estilo
- ✓ Una división no puede insertarse dentro de una etiqueta *inline* (<strong>,<em>, etc.) o de un bloque de texto (<p>, <h1>, etc.)
  - Pero sí puede insertarse dentro de otra división <div>

Con CSS se puede luego definir la posición de los distintos elementos



Tradicionalmente se hacía con elementos **<div>**

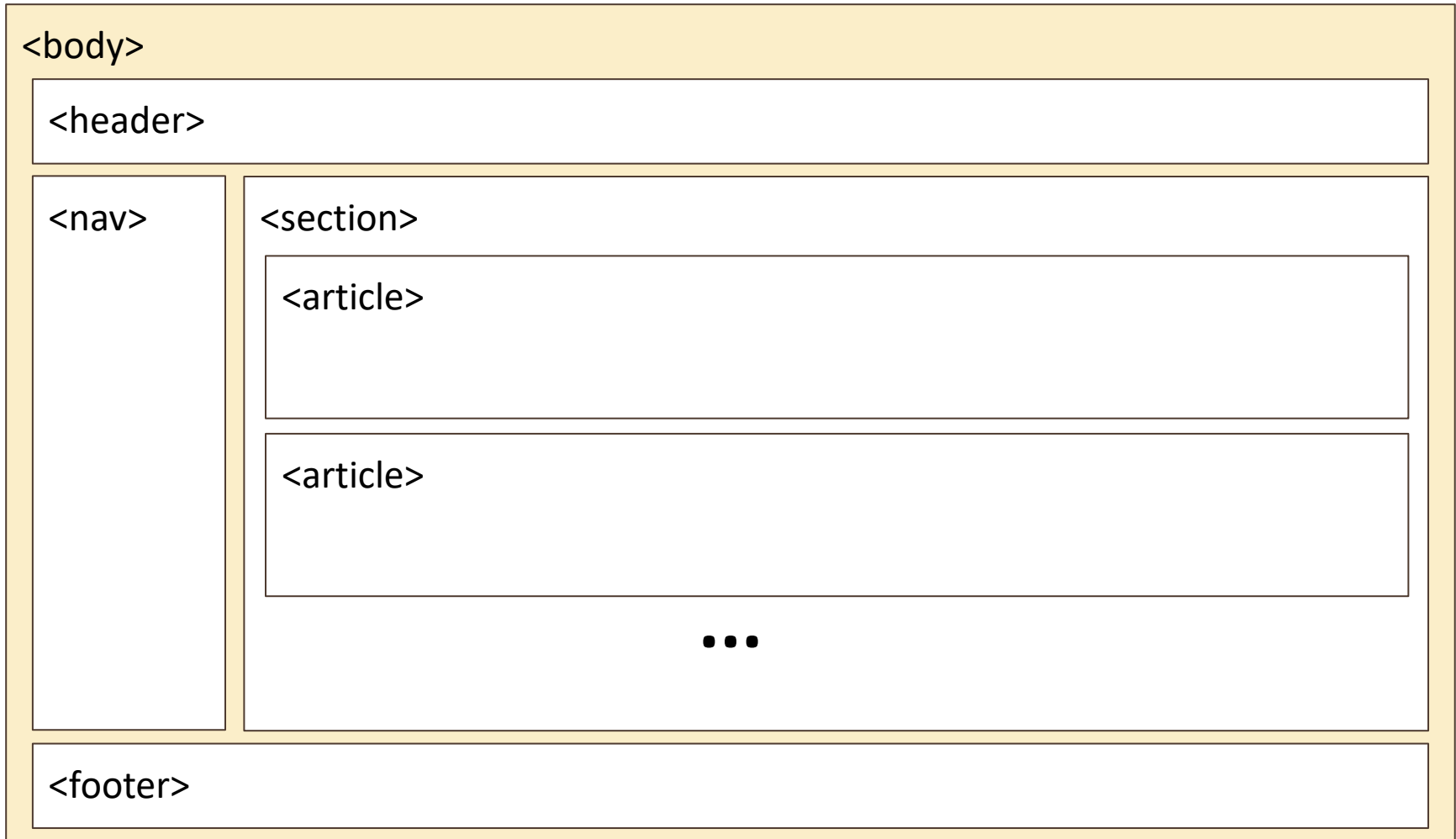


## Estructuras semánticas (sólo HTML5)

- ✓ Etiquetas similares a `<div>` pero con significados más específicos:
  - `<header>`, `<nav>`, `<section>`, `<article>`, `<aside>`, `<footer>`, `<details>`, `<main>` ...
- ✓ Facilitan el procesamiento semántico (búsquedas, procesamiento del contenido)
- ✓ Clarifica la estructura del documento con elementos estándar comunes
- ✓ Se pueden anidar dentro de otros elementos contenedores



# Estructura de la página



## HTML: Apariencia y navegación



# Imágenes

``

- ✓ Descripción textual para navegadores 'modo texto', mientras se carga la imagen, y como texto 'rollover'

Formatos recomendables:

- ✓ .svg: para imágenes, cliparts, y esquemas (escalan a cualquier tamaño de dispositivo).
- ✓ .png : para imágenes con pocos colores donde los detalles pequeños son importantes; ej.: capturas de pantalla
- ✓ .jpg: para imágenes con muchos colores donde los detalles finos son menos salientes; ej.: fotos

Ahora ya podemos utilizar imágenes adaptadas a los dispositivos

- ✓ Usando `<img srcset size>` ó `<picture>`
- ✓ Otra explicación detallada

Code 0x7F c

chart; it is c

Code 0x7F corres  
chart; it is covered in the previous section's

Binary	Oct	Dec	Hex	Glyph
010 0000	040	32	20	☐
010 0001	041	33	21	!
010 0010	042	34	22	"
010 0011	043	35	23	#

Code 0x7F c

chart; it is c

Code 0x7F corresponds to th  
chart; it is covered in the previous section's

Binary	Oct	Dec	Hex	Glyph
010 0000	040	32	20	☐
010 0001	041	33	21	!
010 0010	042	34	22	"
010 0011	043	35	23	#

.png – 6kb

.jpg – 12kb  
(calidad al 50%)



.jpg – 32kb  
(calidad al 50%)

.png – 428kb

`<a href="destino">texto del enlace</a>`

✓ “a” de *anchor* (ancla)

Campo “destino”:

- ✓ Dirección absoluta: `"http://www.ucm.es"`
- ✓ Página local absoluta: `"file:///c:/mifichero.txt"`
- ✓ Dirección relativa: `"foros/general.php"`
- ✓ Correo electrónico: `"mailto:raquelhb@fdi.ucm.es"`



## Contenido

- ✓ Generalmente texto que aparece subrayado en azul
- ✓ Puede ser una imagen

```
<a href="http://www.ucm.es/">  
    
</a>
```

## Dónde se abre el enlace:

- ✓ En la misma ventana: target="\_self" (opción por defecto)
- ✓ En otra ventana (o pestaña): target="\_blank"





En lugar de a una página, los enlaces pueden apuntar a un elemento concreto

- ✓ Primero, definimos un identificador

```
<h1 id="apartado2">...</h1>
```

- ✓ Enlace para ir al apartado desde el mismo documento

```
<a href="#apartado2">Ir al apartado 2</a>
```

- ✓ Enlace desde otro documento distinto

```
<a href="pagina.html#apartado2">Ir al apartado 2</a>
```



## Tres tipos de listas:

- ✓ Listas numeradas (`<ol> ... </ol>`)
  - `<li> ... </li>` delimitan cada elemento de la lista
- ✓ Listas no ordenadas (`<ul> ... </ul>`)
  - `<li> ... </li>` delimitan cada elemento de la lista
- ✓ Listas de definición (`<dl> ... </dl>`)
  - `<dt> ... </dt>` delimitan los términos
  - `<dd> ... </dd>` delimitan las definiciones



## Las listas se pueden anidar

```
<ol>
  <li>Primer elemento </li>
  <li>Segundo elemento </li>
  <ul>
    <li>Elemento de lista desordenada anidada</li>
  </ul>
  <li>Definiciones: </li>
  <dl>
    <dt>Término</dt>
    <dd>Definición del término 1</dd>
  </dl>
</ol>
```

1. Primer elemento
2. Segundo elemento
  - Elemento de lista desordenada anidada
3. Definiciones:
  - Término
  - Definición del término 1

Escribe el código necesario para generar las siguientes listas anidadas:

## Ejercicio de listas anidadas

- Introducción
- Edición de páginas Web
  - [HTML y XHTML](#)
  - [CSS](#)
  - [HTML5](#)
- Programación de páginas web
  1. JavaScript
  2. CGI
  3. PHP
  4. J2EE
- Profesor

*Email*

**nombre@direccion.com**

*Teléfono*

91 123 4567

*Tutorías*

Los martes de 10 a 12



Permiten presentar información tabular, en filas y columnas, con cabeceras

- ✓ Cada elemento de la tabla puede ser simple o a su vez ser otra agrupación de filas y de columnas, cabeceras y pies de tabla, subdivisiones, cabeceras múltiples y otros elementos complejos

Como permiten un control muy detallado, a veces se usan para organizar la estructura general de una página web.

**Esto es algo que no se recomienda en absoluto**



# Tablas – Componentes

Leyenda de la tabla  
<caption>

*Calificaciones*

Cabecera de columna  
<th>

Cabecera de la tabla  
<thead>

Cuerpo de la tabla  
<tbody>

Pie de la tabla  
<tfoot>

Fila  
<tr>

Cabecera de fila  
<th>

celda  
<td>

Alumno	Práctica	Trabajo	Final
Álvarez Gómez, Javier	8	8	NT (8)
Gutiérrez Rodríguez, Clara	8	10	SB (9)
Rodríguez Hernández, Pedro	8	6	NT (7)
Revisión de exámenes: martes 18 a las 12h			



## <table>

- ✓ Define una tabla
- ✓ Anteriormente, se configuraba con atributos:
  - Border, width, cellpadding, cellspacing, ...
  - Todos los atributos de las tablas desaparecen en HTML5
    - Mejor usar CSS

## <caption>

- ✓ Leyenda de la tabla: texto opcional que se muestra fuera de la tabla (por defecto, arriba)
- ✓ Se suele poner justo después de <table>
- ✓ No puede incluir párrafos ni otros elementos de bloque, aunque sí etiquetas en línea (<strong>, <em>, etc.)



Una tabla sencilla se define con

- ✓ La etiqueta `<table>`
- ✓ A continuación se definen las filas, con `<tr>` (table row)
- ✓ Y para cada fila, los elementos con `<td>` (table data cell)
  - Algunas celdas se usan como cabeceras de fila o columna:  
`<th>` (table header)

Se pueden agrupar celdas

- ✓ En una fila con el atributo `colspan`
- ✓ En una columna con el atributo `rowspan`





# Tablas sencillas

```
<table>
<caption>Fusión de filas y columnas</caption>
<tr>
  <th colspan=3>Números</th>
</tr>
<tr>
  <th>Nombre</th>
  <th>Valor</th>
  <th>Idioma</th>
</tr>
<tr>
  <td>Uno</td>
  <td>1</td>
  <td rowspan="2">Español</td>
</tr>
<tr>
  <td>Dos</td>
  <td>2</td>
</tr>
</table>
```

Fusión de filas y columnas

Números		
Nombre	Valor	Idioma
Uno	1	Español
Dos	2	



## Acentos y letras especiales:

- ✓ Lo mejor es declarar el uso de caracteres UTF-8 para que se vean bien los acentos y letras como la ñ
  - Hoy día la mayoría de los navegadores interpretan bien UTF-8
- ✓ En ocasiones, sin embargo, algunas herramientas no lo interpretan bien, y convierten el texto a ISO-8859 y habría problemas
- ✓ Se pueden sustituir por símbolos universales (cualquier navegador sabrá interpretarlos)

&ntilde;	ñ	&Ntilde; Ñ	Ñ
&aacute;	á	&Aacute; Á	Á
&eacute;	é	&Eacute; É	É
&iacute;	í	&Iacute; Í	Í
&oacute;	ó	&Oacute; Ó	Ó
&uacute;	ú	&Uacute; Ú	Ú
&uuml;	ü	&Uuml; Ü	Ü
&euro;	€		



## Caracteres especiales

- ✓ Hay muchos símbolos que no se pueden poner “sin más” en un documento HTML
  - &lt; <
  - &gt; >
  - &amp; &
  - &quot; "
  - &nbsp; (espacio en blanco)
  - &apos; '
  - &ndash; –

Lista de todos los 256 caracteres especiales en HTML:

- ✓ [http://en.wikipedia.org/wiki/List\\_of\\_XML\\_and\\_HTML\\_character\\_entity\\_references](http://en.wikipedia.org/wiki/List_of_XML_and_HTML_character_entity_references)



Texto que ignora el navegador

```
<!-- texto del comentario (una o más líneas) -->
```

Los comentarios pueden estar insertados en cualquier lugar de la página web. Normalmente se usan para:

- ✓ Marcar el comienzo y el final de las secciones de las páginas

```
<!-- Inicio de las noticias -->
```

```
<div id="noticias"> ... </div>
```

```
<!-- Fin de las noticias -->
```

- ✓ Incluir notas para otros diseñadores

```
<!-- Esto se puede mejorar -->
```

- ✓ Incluir explicaciones sobre el código de la página

```
<!--
```

```
Script para identificar las preferencias del usuario
```

```
-->
```



## HTML: Formularios



<form>

Conjunto de controles que permiten al usuario interactuar

- ✓ Generalmente para introducir datos y enviarlos al servidor web
- ✓ El navegador envía únicamente los datos de los controles contenidos en el formulario
- ✓ En una misma página puede haber varios formularios que envíen datos al mismo o a diferentes servidores



## Ejemplo:

```
<form action="http://www.miweb.com/procesaform.php" method="post">  
Escribe tu nombre:  
<input type="text" name="nombre" value="" />  
<br/>  
<input type="submit" value="Enviar" />  
</form>
```

### Formulario muy sencillo

Escribe tu nombre:



## Atributos de <form>

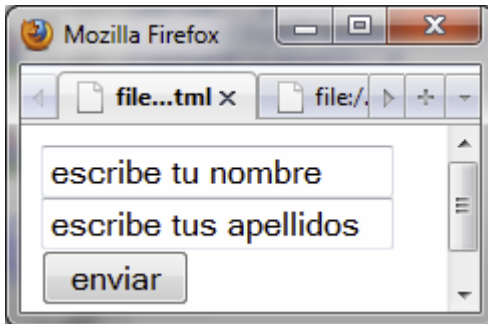
- ✓ **action="URL":** aplicación del servidor que procesará los datos remitidos (por ejemplo, un script de PHP)
- ✓ **method:** método HTTP para enviar los datos al servidor
  - **GET:** como añadido a la dirección indicada en el atributo **action**
    - Limitado a 500 bytes
    - Los datos enviados se añaden al final de la URL de la página y por tanto se ven en la barra del navegador
  - **POST:** en forma separada
    - Puede enviar más información
    - Permite enviar ficheros adjuntos
    - Los datos enviados no se ven en la barra del navegador
    - Se suele usar cuando se envía información que puede modificar el servidor





# Formularios

- ✓ POST vs GET: ¿estás cambiando algo de forma irreversible?  
Un botón de 'refrescar listado' puede enviar 'GET';  
Uno de 'borrar archivo' (irreversible) sólo debe enviar 'POST'.



A screenshot of a Mozilla Firefox browser window. The address bar shows 'file:///...'. The page content includes two text input fields: 'escribe tu nombre' and 'escribe tus apellidos', followed by an 'enviar' button.



POST /pagina-destino HTTP/1.1

... resto de cabeceras ...

Connection: keep-alive

Content-Type: application/x-www-form-urlencoded

Content-Length: 56

nombre=escribe+tu+nombre&apellidos=escribe+tus+apellidos



GET /pagina-destino?nombre=escribe+tu+nombre&apellidos=escribe+tus+apellidos HTTP/1.1  
... resto de cabeceras ...



Dentro de un formulario puede haber:

- ✓ Cualquier elemento típico de una página web
  - Párrafos, imágenes, divisiones, listas, tablas, etc.
- ✓ Controles de formularios
  - `<input />`, `<button>`, `<select>` y `<option>`, `<optgroup>`, `<textarea>`
- ✓ Estructura de formularios
  - `<fieldset>` y `<legend>`
- ✓ Información para accesibilidad
  - `<label>` permite mejorar la accesibilidad de los controles
- ✓ Controles avanzados (sólo HTML5)
  - `<datalist>`, `<keygen>`, ...
- ✓ Revisar [MDN Web Forms!](#)



## <input />

- ✓ type = "text | password | checkbox | radio | submit | reset | file | hidden | image | button" - Indica el tipo de control que se incluye en el formulario
- ✓ name = "texto" - Nombre del control (para que el servidor pueda procesar el formulario)
- ✓ value = "texto" - Valor inicial del control
- ✓ size - Tamaño inicial del control (en píxeles, salvo para campos de texto y de password que se refiere al número de caracteres)
- ✓ disabled - El control aparece deshabilitado y su valor no se envía al servidor junto con el resto de datos
- ✓ readonly - El contenido del control no se puede modificar
- ✓ alt = "texto" - Descripción del control

Y muchos más...



## Cuadro de texto

Nombre <br/>

Nombre

```
<input type="text" name="nombre" value="" />
```

Se enviará al servidor cuando se pulse un botón de enviar

- ✓ El nombre asignado en **name** tiene que concordar con el que se use en la aplicación en el servidor
- ✓ **value** permite establecer un valor inicial en el cuadro de texto

## Contraseñas

Contraseña <br/>

Contraseña

```
<input type="password" name="contrasena" value="" />
```

- ✓ Igual que el cuadro de texto, pero el valor introducido no se ve



## Cuadro de texto de varias líneas

Nombre <br/>

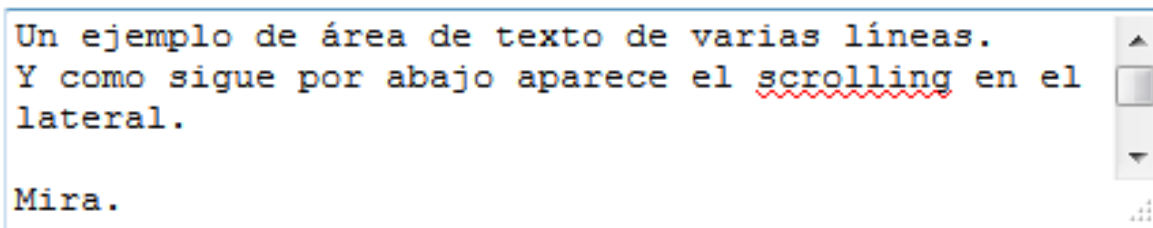
```
<textarea name="nombre" rows="4" cols="50">
```

Contenido inicial del cuadro de texto

```
</textarea>
```

### ✓ Atributos

- rows: número de filas visibles (sale una barra de desplazamiento si se introducen hay más)
- cols: anchura en caracteres

A screenshot of a web browser showing a text area. The text area contains the text: "Un ejemplo de área de texto de varias líneas. Y como sigue por abajo aparece el scrolling en el lateral. Mira." The word "scrolling" is underlined with a red wavy line. To the right of the text area is a vertical scrollbar with a grey slider and up/down arrows, indicating that the text area is scrollable.

## Botón de envío de formulario

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



- ✓ El navegador se encarga de enviar automáticamente los datos cuando el usuario pincha el botón

## Botón de reseteo de formulario

```
<input type="reset" name="borrar" value="Borrar  
formulario" />
```



- ✓ El navegador borra toda la información introducida y muestra el formulario en su estado original

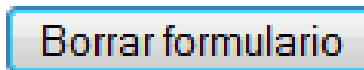


## Botones en general: <button>

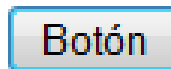
```
<button type="submit">Enviar</button>
```



```
<button type="reset">Borrar formulario</button>
```



```
<button type="button">Botón</button>
```



Ambos tienen tres tipos:

- ✓ submit: botón de envío; envía el formulario a su destino
- ✓ reset: pone todos los campos a sus valores iniciales
- ✓ button: no hace nada; útil en combinación con JavaScript

## Mejor Button

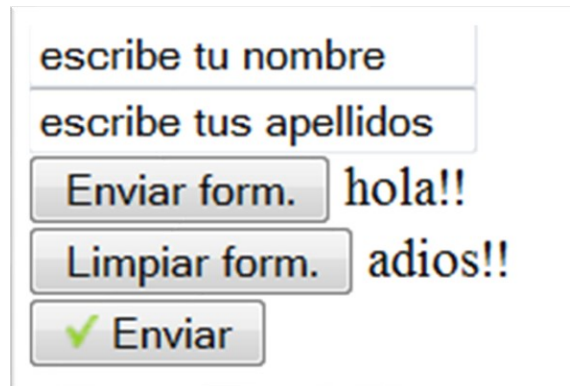
- ✓ Puede contener HTML dentro
- ✓ Se puede personalizar su estilo más fácilmente
- ✓ Pero no funciona en IE6 (y anteriores)





# ¿Input o Button?

```
<form action="http://example.com/prueba.php" method="POST">
  <input type="text" name="nombre" value="escribe tu nombre"/>
  <input type="text" name="apellidos" value="escribe tus
apellidos"/>
  <input type="submit" name="submit1" value="Enviar
form.">Hola!!</input>
  <input type="reset" name="ignorado" value="Limpiar
form.">adios!!</input>
  <button type="submit" name="submit2" value="Enviar form.">
     Enviar
  </button>
</form>
```



## Casillas de verificación (checkbox)

```
Lenguajes de programación: <br/>
<input name="java" type="checkbox" value="Java" checked/> Java
<input name="cplusplus" type="checkbox" value="Cplusplus"/> C++
<input name="csharp" type="checkbox" value="Csharp"/> C#
<input name="otros" type="checkbox" value="Otros"/> Otros
```

**checked** indica si la casilla está activada por defecto

Lenguajes de programación:

☒ Java ☐ C++ ☐ C# ☐ Otros



## Radiobutton

Sexo <br/>

```
<input type="radio" name="sexo" value="hombre" checked/> Hombre
```

```
<input type="radio" name="sexo" value="mujer" /> Mujer
```

Sexo

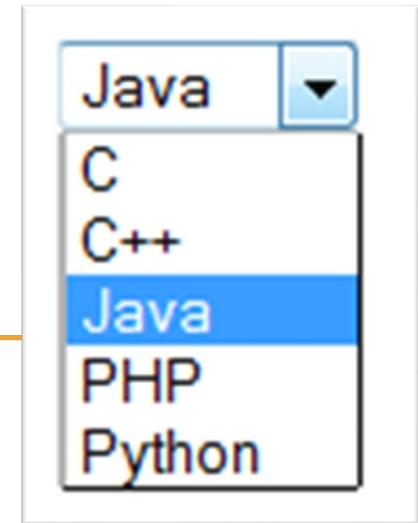
☒ Hombre ☐ Mujer

Hay que usar el mismo “name” para formar un grupo de selección exclusiva



## Listas de selección

```
<select name="lenguajes">
  <option value="c">C</option>
  <option value="cplusplus">C++</option>
  <option value="java" selected>Java</option>
  <option value="php">PHP</option>
  <option value="python">Python</option>
</select>
```



### ✓ Atributos de option:

- value determina el valor que se envía al servidor
- selected permite definir la opción por defecto



## Incluir un fichero

- ✓ El atributo enctype en la etiqueta <form> del formulario tiene que ser multipart/form-data
- ✓ Sólo vale si el método es post

```
<form name="fichero" action="procesa_fichero.php"
  method="post" enctype="multipart/form-data">
  Fichero: <input type="file" name="archivo" />
  <input type="submit" value="Enviar">
</form>
```

Fichero:



## Agrupación de elementos

- ✓ Permite ver mejor las partes de un formulario agrupando elementos relacionados
- ✓ `<legend>` es el título que se visualiza con el grupo

```
<fieldset>  
  <legend>Información personal:</legend>  
  Nombre: <input type="text" size="50"><br>  
  E-mail: <input type="text" size="50"><br>  
  Ciudad: <input type="text" size="20">  
</fieldset>
```

Información personal:

Nombre:

E-mail:

Ciudad:



## Licencia Creative Commons

✓ Este documento tiene establecidas las siguientes condiciones:

- ① Reconocimiento (*Attribution*):  
En cualquier explotación de la obra autorizada por la licencia hará falta reconocer la autoría.
- Ⓜ No comercial (*Non commercial*):  
La explotación de la obra queda limitada a usos no comerciales.
- Ⓢ Compartir igual (*Share alike*):  
La explotación autorizada incluye la creación de obras derivadas siempre que mantengan la misma licencia al ser divulgadas.

✓ Material elaborado por Juan Pavón Mestras, con modificaciones de Pablo Moreno Ger, Manuel Freire Morán y Raquel Hervás Ballesteros.

