

# Tema 3:

## XHTML y CSS

---

1. Introducción.
2. Estructura básica: head. body.
3. Organización del texto.
4. Elementos hipertexto-hipermedia.
5. Listas.
6. Tablas.
7. Frames.
8. Capas.
9. Formularios.
10. Hojas de estilo.

## 1. Introducción.

# La Web

---

- World Wide Web (abreviado: Web)
  - Universo de información interconectada, accesible a través de Internet.
- Propuesta por Tim Berners-Lee (1989).
- Ha tenido mayor difusión que otros servicios contemporáneas (Archie, Gopher, WAIS), gracias sobre todo a uno de sus elementos: el HTML.
  - Hipertexto.
  - Hipermedia.
- En la actualidad hay más de **un trillón** de URLs accesibles al público, repartidas entre más de **185 millones** de servidores.
- .

## 1. Introducción.

# Lenguajes de Marcas

---

- **Uso:**
  - Proveen una representación estándar de los documentos hipertexto en formato ASCII.
  - Permiten formatear texto, integrar imágenes, referenciar otros documentos, etc.
- **Algunos tipos:**
  - **HTML (HyperText Markup Language)**
    - Formatea páginas web para su visualización “estándar” en PC.
    - Sintaxis relajada.
  - **XML (eXtensible Markup Language)**
    - Metalenguaje. Define la estructura y contenido de documentos.
    - Sintaxis rígida.
  - **XHTML (XML HTML)**
    - Reformulación del HTML en XML.
  - **WML (Wireless Markup Language)**
    - Para la visualización de páginas WAP en dispositivos inalámbricos.

## 1. Introducción.

# Versiones de HTML y XHTML

---

- HTML 1.0 (1992) : Contenedor principal <html>. Secciones principales <head> y <body>. Operadores lógicos, listas, enlaces, imágenes, cabeceras y conj. caracteres.
- HTML 2.0 (1995): Formularios (CGIs).
- HTML 3.0 (1996): A partir de las propuestas de Netscape. Demasiado extenso.
- HTML 3.2 (1997): Creado por W3C. Tablas, applets, etc
- HTML 4.01 (1999): Frames, CSS, scripts.
  - <http://www.w3.org/TR/html401>
- XHTML 1.0 (2000): Sintaxis XML.
  - <http://www.w3.org/TR/xhtml1>
- XHTML 2.0: Siguiete versión (borrador).
- HTML 5 (borrador). Incorpora nuevos elementos : section, article, aside, header, footer, nav, dialog, figure, audio, video, embed, m, meter, time, canvas, command, datagrid, details, datalist, event-source, output, progress

## 1. Introducción.

# Elementos de los documentos HTML y XHTML (I)

### ■ Texto.

- Los espacios, retornos de carro y tabuladores no son significativos

La siguiente                      secuencia  
aparecerá                      en la misma línea, sin  
espacios ni                      tabuladores.

### ■ Caracteres especiales.

- Valores numéricos (Codificación aplicada: iso-latín-1)

- `&#D.`                      Pe: `la ñ es &#241`

- `&#xH.`                      Pe: `&#xF1`

- Valores simbólicos:

- `acute.`                      Pe: `á - &aacute;`

- `grave.`                      Pe: `à - &agrave;`

- `tilde.`                      Pe: `ñ - &ntilde;`

- `cedil.`                      Pe: `ç - &ccedil;` `Ç - &Ccedil;`

- `Otros..`                      Pe: `< - &lt;`; `> - &gt;`; `& - &amp;`;

## 1. Introducción.

# Elementos de los documentos HTML y XHTML (II)

- Directivas (comandos o etiquetas).
  - Interpretar el documento HTML en el navegador.
  - Estructura: <directiva atributo="valor" ...>.
  - Tipos:
    - Vacías (o simples). Pe: <br /> (XHTML) o <BR> (HTML)
    - Delimitadoras (o dobles) Pe: <b> ..... </b>
  - Atributos: Modifican el significado o el comportamiento normal de la directiva.  
Pe: <h1 align="center"> ..... </h1>
    - Atributos comunes: class, id, lang, style.
- Comentarios (es una directiva): <!-- ..... -->

## 1. Introducción.

# Diferencias entre HTML y XHTML (I)

- La sintaxis de XHTML es mucho más rígida
  - Las etiquetas y los atributos siempre se ponen en minúsculas.
    - Ej. HTML: `<BODY BgColor="#ff00ff">`
    - Ej. XHTML: `<body bgcolor="#ff00ff">`
  - Los valores de los atributos van entrecomillados.
    - Ej. HTML: `<TABLE BORDER=0 align="center">`
    - Ej. XHTML: `<table border="0" align="center">`
  - No se pueden dejar instrucciones por cerrar
    - Ej. HTML: `<P>Un párrafo.`  
`<P>Otro párrafo.`
    - Ej. XHTML: `<p>Un párrafo</p>`  
`<p>Otro párrafo</p>`
  - Deben cerrarse correctamente.
    - Ej. HTML: `<p>Ejemplo<b>caracteres en negrita</p></b>`
    - Ej. XHTML: `<p>Ejemplo<b>caracteres negrita</b></p>`

## 1. Introducción.

# Diferencias entre HTML y XHTML (II)

- XHTML añade nuevos elementos (XML).
  - Al igual que en cualquier otro documento XML, la primera línea del documento indica la versión de XML empleada.
    - `<?xml version="1.0" encoding="UTF-8" ?>`
  - A continuación se indica el tipo de documento (DTD).
    - `<!DOCTYPE html PUBLIC`  
    `"- //W3C//DTD XHTML 1.0 Strict//EN"`  
    `"http://www.w3c.org/TR/xhtml1/DTD/strict.dtd" >`
  - Y a continuación se define el espacio de nombres con el atributo `xmlns` de la directiva `<html>`
    - `<html xmlns="http://www.w3c.org/TR/xhtml1" .. >`
  - Es obligatorio cerrar todas las directivas, incluso las vacías.
    - Ej. HTML: ``
    - Ej. XHTML: ``



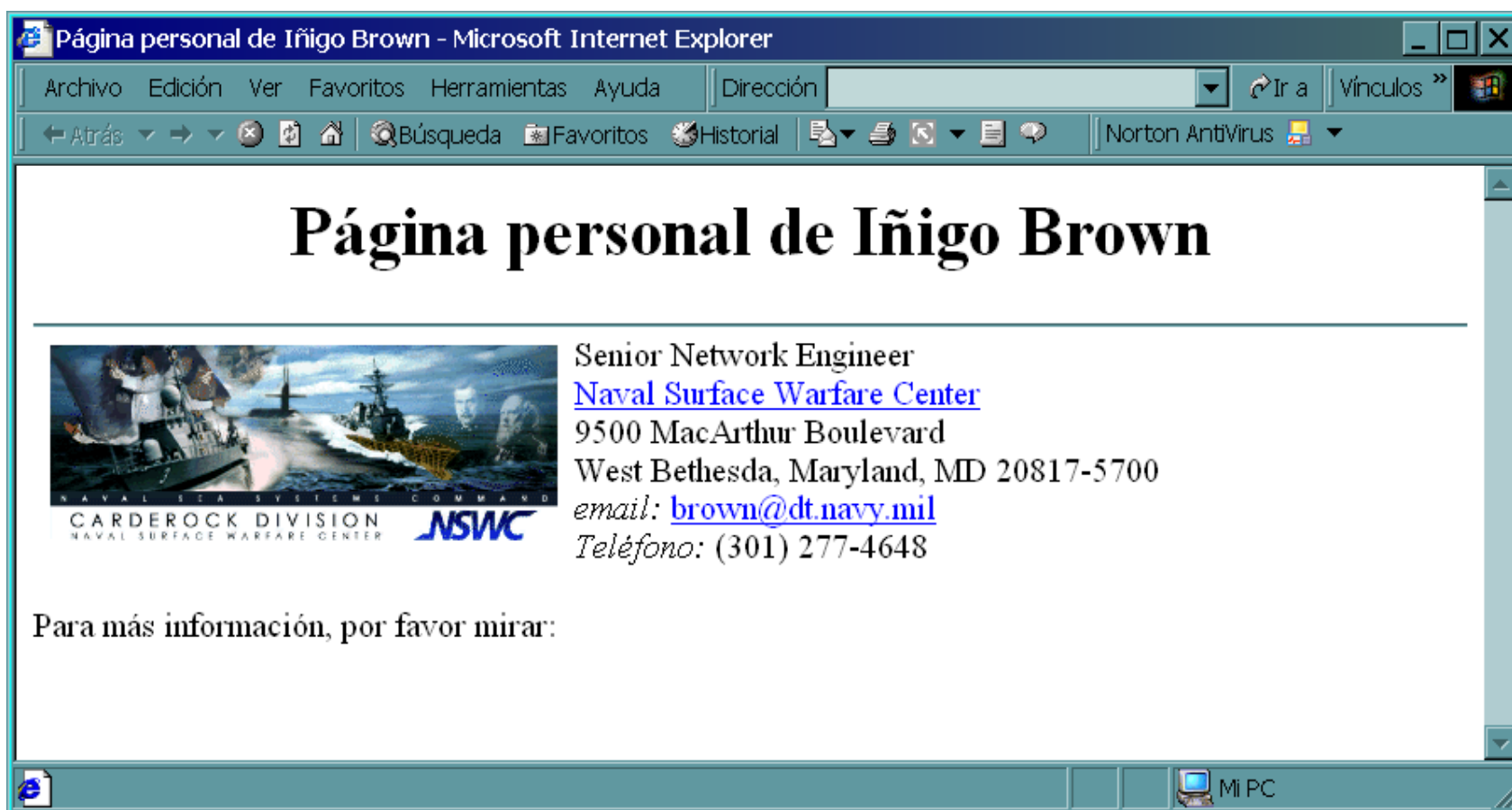
## 1. Introducción.

# Ejemplo XHTML (I)

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="es" lang="es">
  <head>
    <title>P&aacute;gina personal de I&ntilde;igo Brown</title>
  </head>
  <body>
    <h1 align="center">P&aacute;gina personal de I&#241;igo Brown</h1>
    <hr/>
    
    Senior Network Engineer <br />
    <a href="http://www.dt.navy.mil/"> Naval Surface Warfare Center
  </a><br/>
    9500 MacArthur Boulevard <br />
    West Bethesda, Maryland, MD 20817-5700<br />
    <i>email:</i>
    <a href="mailto:brown@dt.navy.mil">brown@dt.navy.mil</a><br />
    <i>Tel&eacute;fono:</i> (301) 277-4648 <br clear="all" />
    <p>Para m&aacute;s informaci&oacute;n, por favor mirar:</p>
    <!-- El resto de la pagina ha sido borrado -->
  </body>
</html>
```

# 1. Introducción.

## Ejemplo XHTML (II)



## 2. Estructura básica de los documentos XHTML.

### Partes del documento XHTML

- Estructura = primeros elementos + cabecera + cuerpo.

- Ejemplo:

```
<?xml version="1.0"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="es"
lang="es">
```

```
<head>
  <title>
    Bienvenido a mi p&aacute;gina
  </title>
</head>
```

```
<body>
  <h1>
    Bienvenido al XHTML!
  </h1>
</body>
```

```
</html>
```



## 2. Estructura básica de los documentos XHTML.

### Primeros elementos del documento

- **Version de xml:** `<?xml version="1.0"?>`  
`<?xml version="1.0" encoding="UTF-8"?>`
- **Tipo de documento.** Referencia al DTD (Data Type Document)
  - **Strict DTD:** Estricta, sin marcado presentacional y sin frames (para usar con CSS)  
`<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"`  
`"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">`
  - **Transitional DTD:** Incluye todos los elementos menos frames.  
`<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"`  
`"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">`
  - **Frameset DTD:** Todos los componentes estándares y frames.  
`<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Frameset//EN"`  
`"http://www.w3.org/TR/xhtml1/DTD/xhtml1-frameset.dtd">`
- **Elemento documento (<html> ... </html>)**
  - **Atributos:** `xmlns` (obligatorio, especifica el espacio de nombres que se va a utilizar), `xml:lang` y `lang`.  
`<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="es"`  
`lang="es">`

## 2. Estructura básica de los documentos XHTML.

### Cabecera del documento (I)

---

- Incluida dentro de la directiva `<head> .. </head>`  
Contiene información sobre el propio documento, así como la definición de reglas de estilo y código script utilizado posteriormente dentro del cuerpo del documento.

- **Título del documento** (`<title> .. </title>`).

- Texto que aparece en la barra del browser.
- Obligatorio.

- `<meta />` Metainformación sobre el documento.

#### Atributos:

- `name=“..”` tipo de información a exportar.
- `content=“..”` contenido de la información (obligatorio)
- `http-equiv=“..”` comando HTTP.

*P.E:*

```
<meta name="description" content="Free Web tutorials on HTML,  
CSS, XML, and XHTML" />
```

```
<meta http-equiv="refresh" content="5; URL=http://www.uv.es/" />
```

## 2. Estructura básica de los documentos XHTML.

### Cabecera del documento (II)

---

- **<link />** Relaciones con otros documentos.

#### Atributos:

- **href**: Documento relacionado con el actual. Se indica una URL.
- **rel**: Tipo de relación entre el documento referenciado y el actual (Posibilidades: appendix, bookmark, chapter, contents, copyright, glossary, help, home, index, next, section, start, **stylesheet**, subsection)
- **type**: Código MIME del documento relacionado.

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

- **<script> ... </script>** Define Scripts (JavaScript).

#### Atributos:

- **Language**: Lenguaje de script (es obligatorio si no se utiliza src). Pe: javascript, vbscript.
- **src**: dirección URL del fichero con el script.
- **type**: Tipo MIME (obligatorio).

```
<script type="text/javascript" src="script.js"> </script>
```

- **<style> .. </style>** Reglas de estilo (Hojas de estilo).

```
<style type="text/css"> h1{color:red} </style>
```

## 2. Estructura básica de los documentos XHTML.

### Cabecera del documento (III)

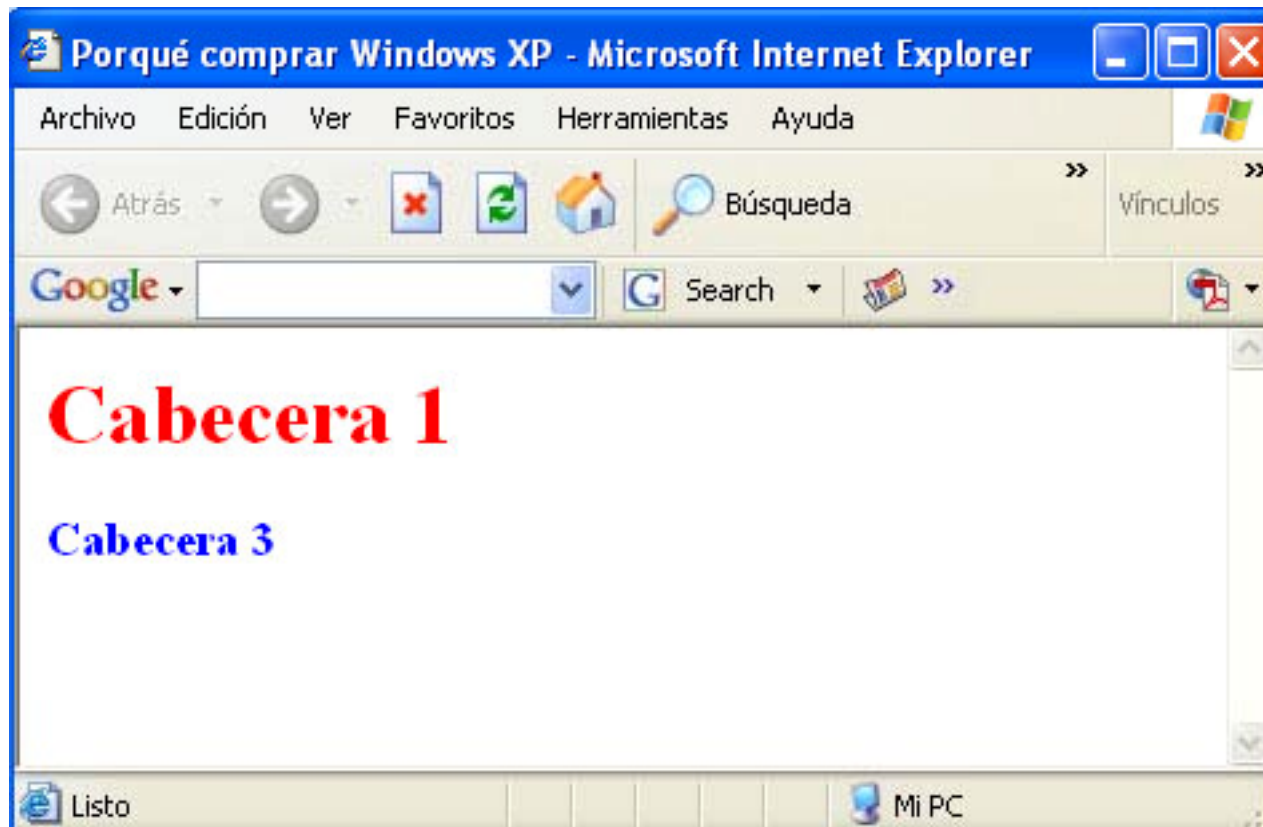
Ejemplo:

```
....
<head>
  <title> Porqué comprar Windows XP </title>
  <meta name="author" content="Bill Gates" />
  <meta name="keywords" content="Windows, OS, Sistema Operativo" />
  <meta name="description" content="Ventajas del Windows XP" />
  <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
  <script type = "text/javascript" language="javascript">
    <!--
      // Aquí vendría el código JavaScript
    ...
  //-->
</script>
<style type="text/css">
h1 {color: red}
h3 {color: blue}
</style>
</head>
<body>
  <h1>Cabecera 1</h1>
  <h3>Cabecera 3</h3>
</body>
....
```

## 2. Estructura básica de los documentos XHTML.

### Cabecera del documento (IV)

---





## 2. Estructura básica de los documentos XHTML.

### Cuerpo del documento

---

- Contiene la información del documento, así como distintas directivas XHTML que indican como interpretar las distintas partes del mismo.
- El cuerpo va incluido dentro de la directiva:  
`<body> . . </body>`
  - La directiva `<body>` incluye una serie de atributos relacionados con la **presentación** del documento (bgcolor, text, link, etc.). Estos atributos están en desuso y se recomienda utilizar en su lugar las hojas de estilo.

### 3. Formato del texto.

## División del texto (I)

---

- `<br />` Salto de línea forzado.
- `<p> .. </p>` Párrafo.

**Atributos:** `align`. Alineación de párrafos.

- Centrados: `<p align="center"> .. </p>`
- A izquierda: `<p align="left"> .. </p>`
- A derecha: `<p align="right"> .. </p>`
- Justificado: `<p align="justify"> .. </p>`

Por defecto aparecen a la izquierda y sin justificar.

Los párrafos vacíos se resumen. Parecería que no están. Para evitarlo se puede utilizar espacios en blanco como contenidos: `&nbsp;`;

- `<div> .. </div>`
  - División del documento en partes.
  - **Atributos:** `align`
    - Con la ayuda de este atributo se puede alinear elementos genéricos (no sólo párrafos).

### 3. Formato del texto.

## División del texto (II)

`<body>`

Esta y la siguiente  
son la misma línea.

`<hr />`

Esta y la siguiente `<br />`  
son líneas distintas.

`<hr />`

Este y el siguiente

`<p>`son párrafos distintos.`</p>`

`<p></p>`

`<p></p>`

`<p></p>`

`<p>` Otro párrafo. `</p>`

`<p> &nbsp; </p>`

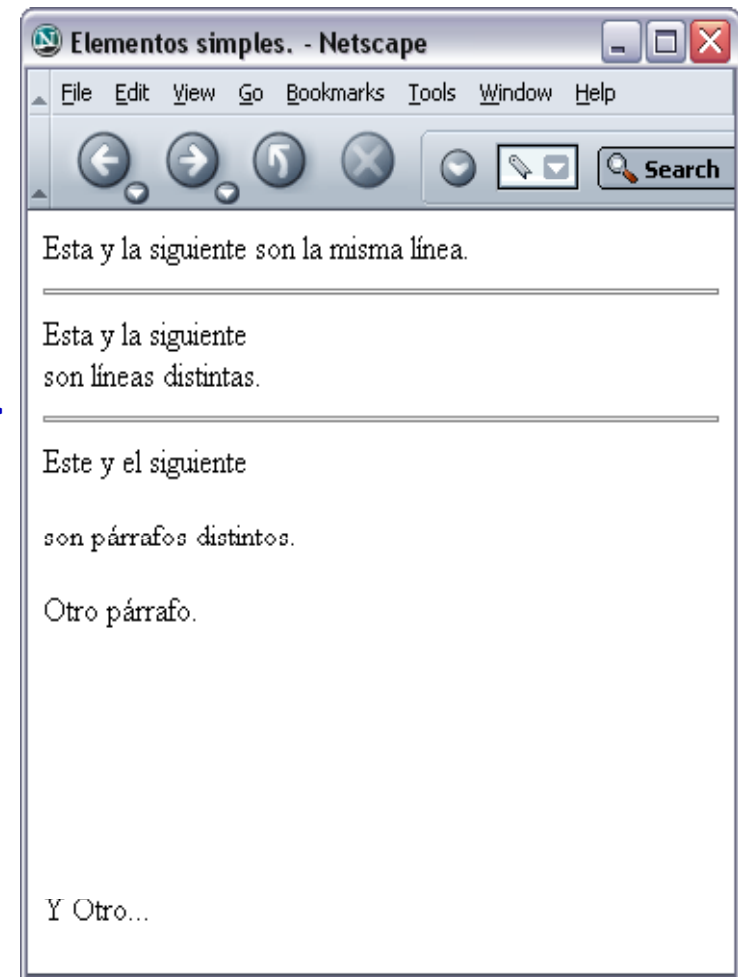
`<p> &nbsp; </p>`

`<p> &nbsp; </p>`

`<p>` Y Otro... `</p>`

`</body>`

`</html>`



### 3. Formato del texto.

## División del texto (III)

---

....

```
<body>
```

```
  <p>Los bloques de textos se alinean normalmente a la  
  izquierda.</p>
```

```
  <p align="center">Con el atributo <p align="center"> se  
    alinean en el centro.
```

```
</p>
```

```
  <p align="right"> Con <p align="right"> se alinea a la  
  derecha
```

```
</p>
```

```
  <p align="justify"> Con la ayuda del atributo <p  
  align="justify"> se consigue justificar el bloque de texto de  
  manera que ocupe toda la l&iacute;nea de la ventana, sin dejar  
  huecos a izquierda o a derecha.
```

```
</p>
```

```
  <div align="center"> Con "div" pueden obtenerse resultados  
  similares </div>
```

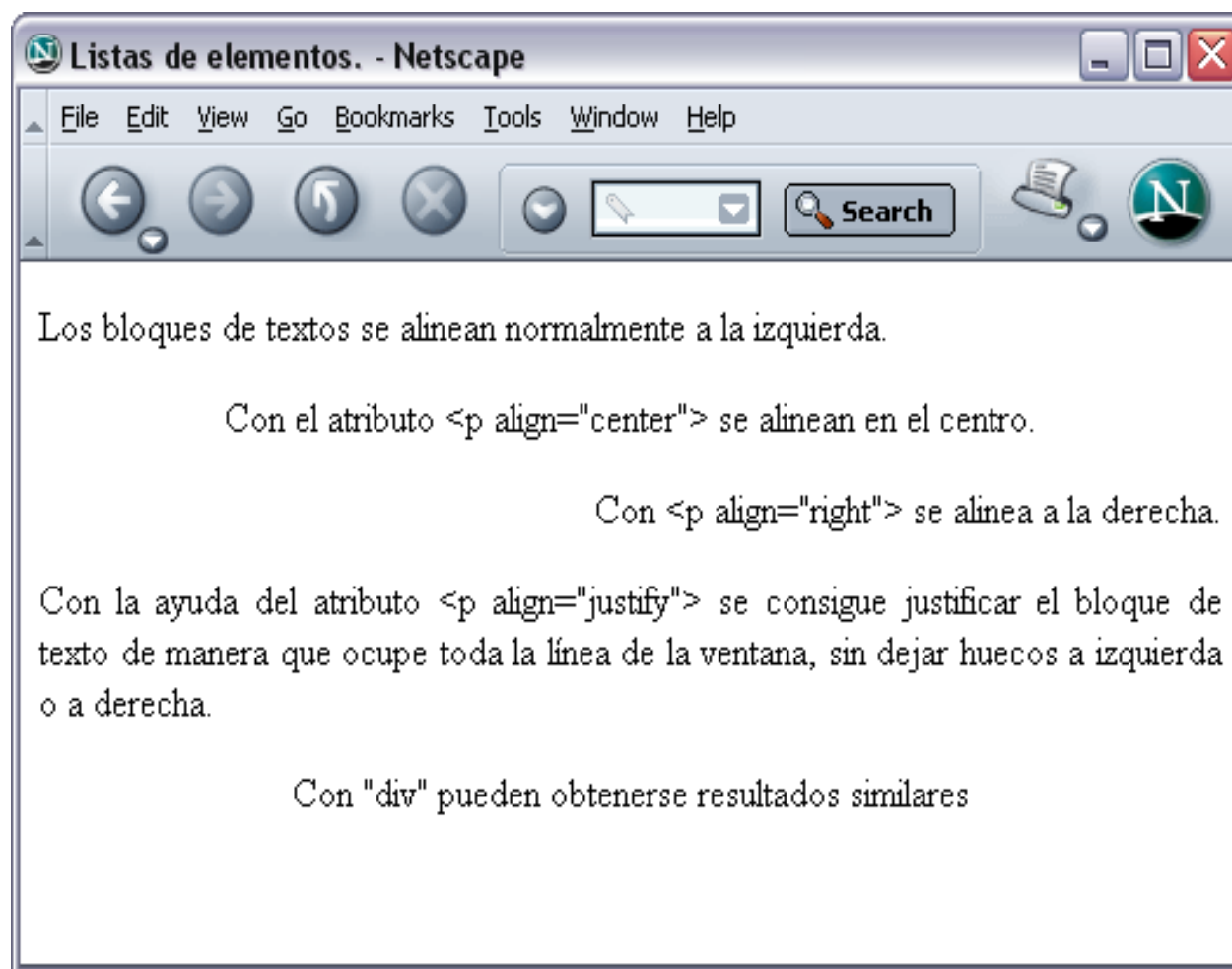
```
</body>
```

```
</html>
```

### 3. Formato del texto.

## División del texto (IV)

---



### 3. Formato del texto.

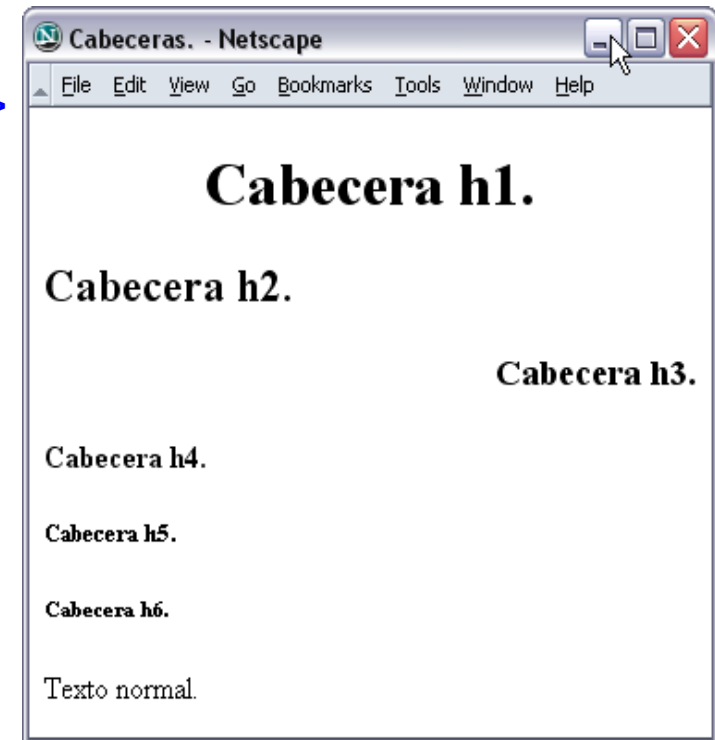
## Cabeceras

- `<hX> .. </hX>` (donde X=1..6): Varios niveles de título. Modifican el tamaño y apariencia de la letra.

**Atributos:** `align`, con los valores “left”, “right” y “center”.

.....

```
<body>
  <h1 align="center"> Cabecera h1.</h1>
  <h2> Cabecera h2.</h2>
  <h3 align="right"> Cabecera h3.</h3>
  <h4> Cabecera h4.</h4>
  <h5> Cabecera h5.</h5>
  <h6> Cabecera h6.</h6>
  Texto normal.
</body>
</html>
```



### 3. Formato del texto.

## Estilos del texto (I)

---

Se distinguen normalmente entre estilos físicos y lógicos.

- **Físicos:** heredados de los procesadores de texto.
  - `<b> .. </b>`: Negrita.
  - `<i> .. </i>`: Cursiva o itálica.
  - `<u> .. </u>`: Subrayado.
  - `<strike> .. </strike>` (o `<s> .. </s>`): Tachado.
  - `<sub> .. </sub>`: Subíndice.
  - `<sup> .. </sup>`: Superíndice.
- **Lógicos:** su apariencia depende del navegador y de las preferencias del usuario.
  - `<strong>` : Marcado fuerte (suele identificarse con negrita)
  - `<em>` : Enfatizado (suele corresponderse con el tipo cursiva)
  - `<span>` : Utilizado en las hojas de estilo para cambiar las propiedades del texto encerrado. (Atributos: class, style)

### 3. Formato del texto.

## Estilos del texto (II)

....

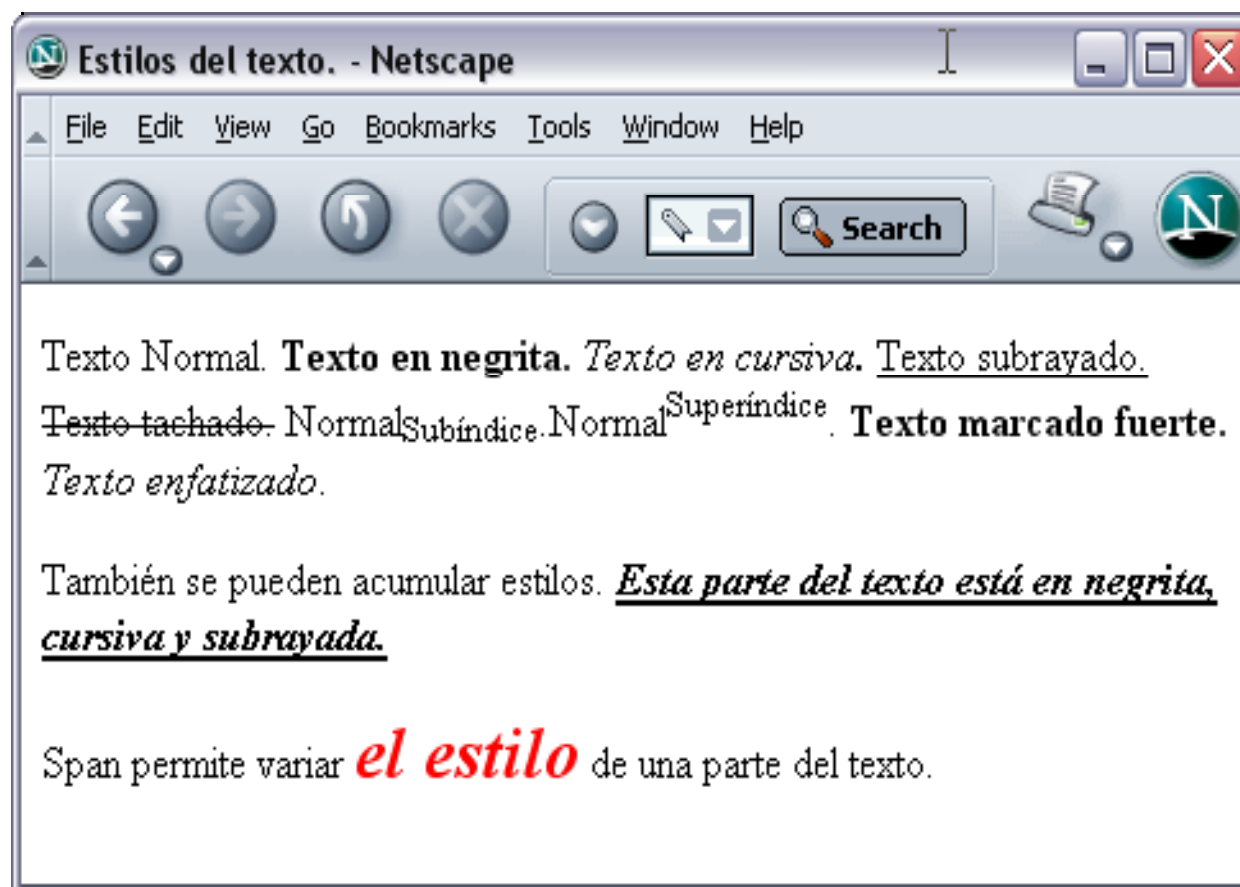
```
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="es" lang="es">
<head>
  <title>Estilos del texto.</title>
  <style type="text/css">
    span {color:red; font-weight:bold; font-style:italic; font-size:20pt}
  </style>
</head>
<body>
  <p>Texto Normal. <b>Texto en negrita.</b> <i>Texto en cursiva.</i>
    <u>Texto subrayado.</u> <s>Texto tachado.</s>
    Normal<sub>Subíndice</sub>.Normal<sup>Superíndice</sup>.
    <strong>Texto marcado fuerte.</strong> <em>Texto enfatizado</em>.
  </p>
  <p>
    También se pueden acumular estilos. <b><i><u> Esta parte
    del texto está en negrita, cursiva y subrayada.</u></i></b>
  </p>
  <p>Span permite variar <span>el estilo</span> de una parte del texto.</p>
</body>
</html>
```



### 3. Formato del texto.

## Estilos del texto (III)

---



### 3. Formato del texto.

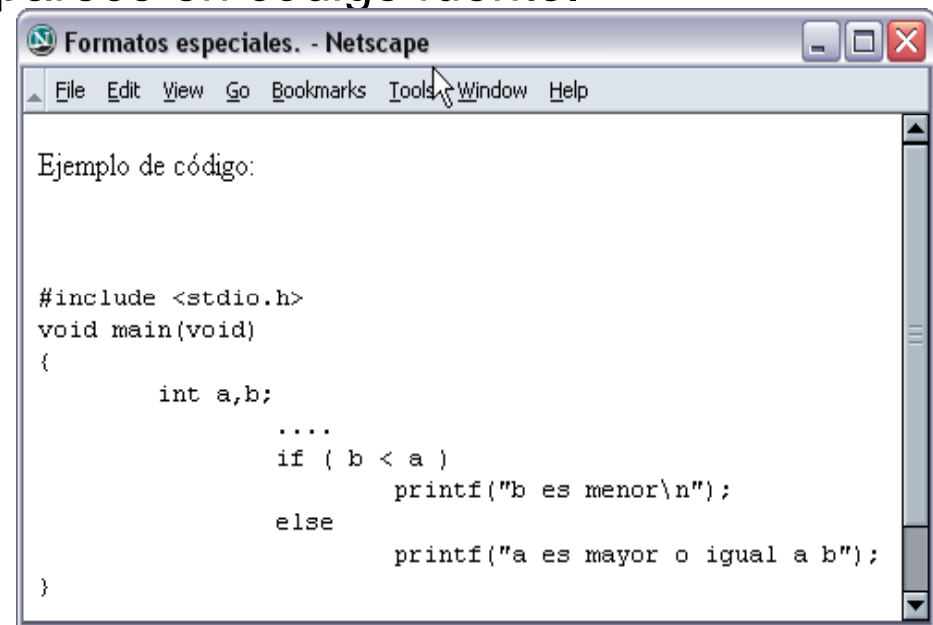
## Formatos especiales

- **<pre> ..</pre>**

- Muestra el texto tal como aparece en código fuente.

**Atributos: width.**

```
....  
<body>  
  <p>Ejemplo de código:</p>  
  <pre>  
    <![CDATA[  
#include <stdio.h>  
void main(void)  
{  
    int a,b;  
    ....  
    if ( b < a )  
        printf("b es menor\n");  
    else  
        printf("a es mayor o igual a b");  
}  
    ]]>  
  </pre>  
</body>
```



## 4. Elementos hipertexto-hipermedia.

### Enlaces hipertexto (I)

---

- Son referencias a otros documentos o a otros recursos web. Permiten saltar de un documento a otro (navegación).
- `<a> .. </a>`
  - Directiva responsable de los enlaces.

#### Atributos:

- **href**: indica la dirección URL de la página o recurso que ha de ser descargado en el navegador. Opcionalmente se puede indicar la sección dentro de la página (#).
- **name** (id): nombre de una nueva sección. Indica el punto de comienzo de la nueva sección. Permite posteriormente hacer referencia a ella con nuevos enlaces (href).
- **target**: frame donde se ubicará el nuevo documento.

## 4. Elementos hipertexto-hipermedia.

### Enlaces hipertexto (II)

---

....

<body>

<a name="inicio\_pagina" id="inicio\_pagina" >

<h1> Enlaces </h1></a>

<p>Con la siguiente p gina se pretende demostrar los diferentes usos de la directiva <A>.

</p>

.....

<p> Se puede encontrar m s informaci n en la p gina

<a href="http://www.w3.org/MarkUp/index.htm#xhtml">"XHTML"</a>

</p>

<p> Si quieres informaci n en castellano pincha

<a href="http://html.programacion.net/curso/">aqu ;</a>

</p>

<p> Para regresar al inicio de la p gina, activa el siguiente <a href="#inicio\_pagina">enlace</a>.

</p>

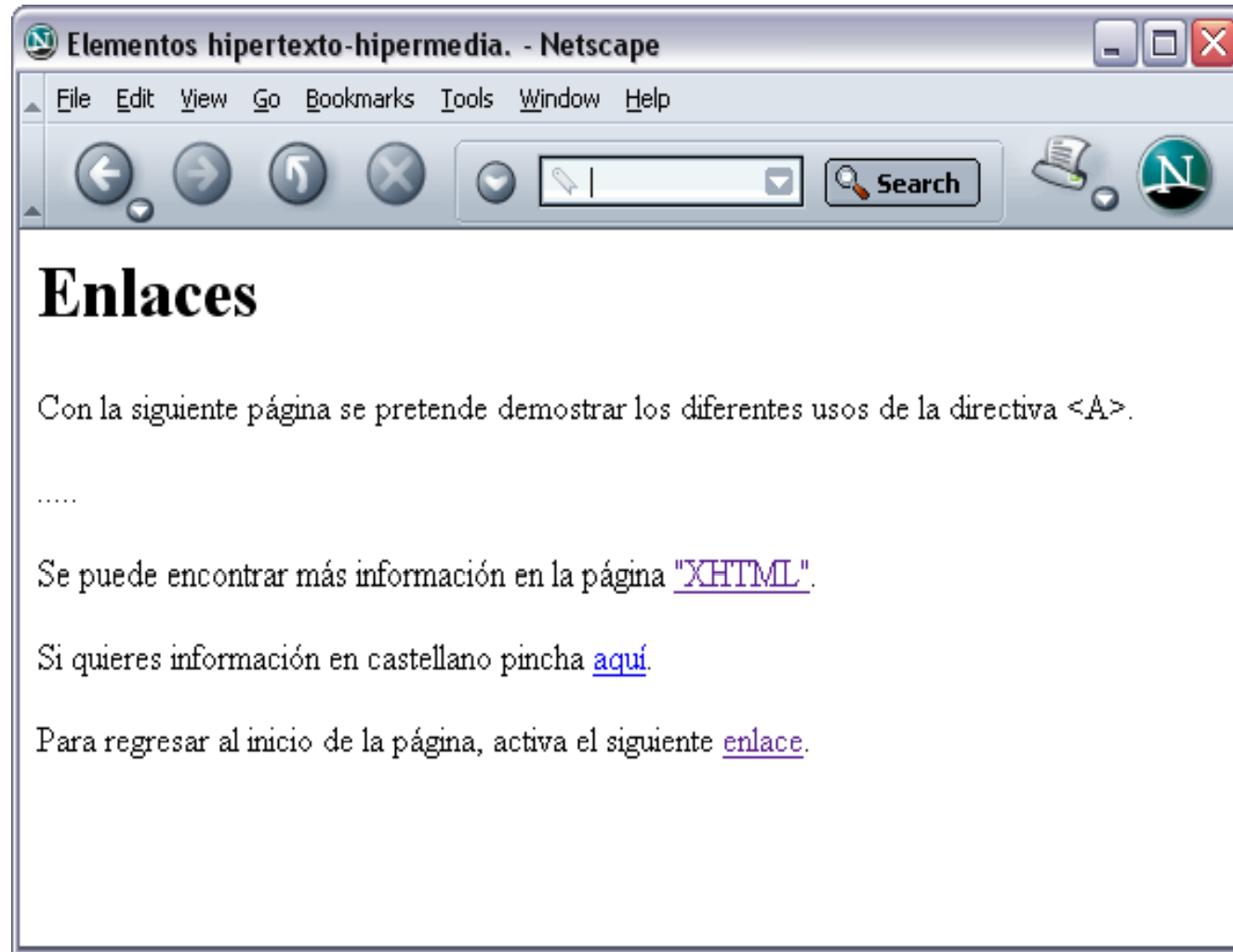
</body>

</html>

## 4. Elementos hipertexto-hipermedia.

### Enlaces hipertexto (III)

---



## 4. Elementos hipertexto-hipermedia.

### Imágenes (I)

---

- **<img />**: Inserta una imagen en el documento. La mayoría de los navegadores soportan los formatos GIF, JPEG y PNG.

#### Atributos:

- **src** (obligatorio): Localización de la imagen.
- **alt** (obligatorio): Texto alternativo. Para aquellos navegadores que no reconocen el formato de la imagen.
- **width** y **height**: Tamaño de la imagen en píxeles.
- **align**: Posición de la imagen. Los posibles valores son: "left", "right", "top", "bottom" y "middle".(<br clear="all" />).
- **hspace** y **vspace**: Número de píxeles en blanco que se dejan a la izquierda y derecha (hspace), y arriba y abajo (vspace) de la imagen.

## 4. Elementos hipertexto-hipermedia.

### Imágenes (II)

---

#### Imagen como enlace

```
<a href="www.uv.es" >
  
</a>
```

#### Atributos:

- **ismap**: Utiliza la imagen como un mapa de imagen en el lado del servidor, conectado a un aplicación (p.e. CGI) referenciada en un enlace que incluye a la imagen.
- **usemap**: definimos las regiones de la imagen en el propio documento html.

```

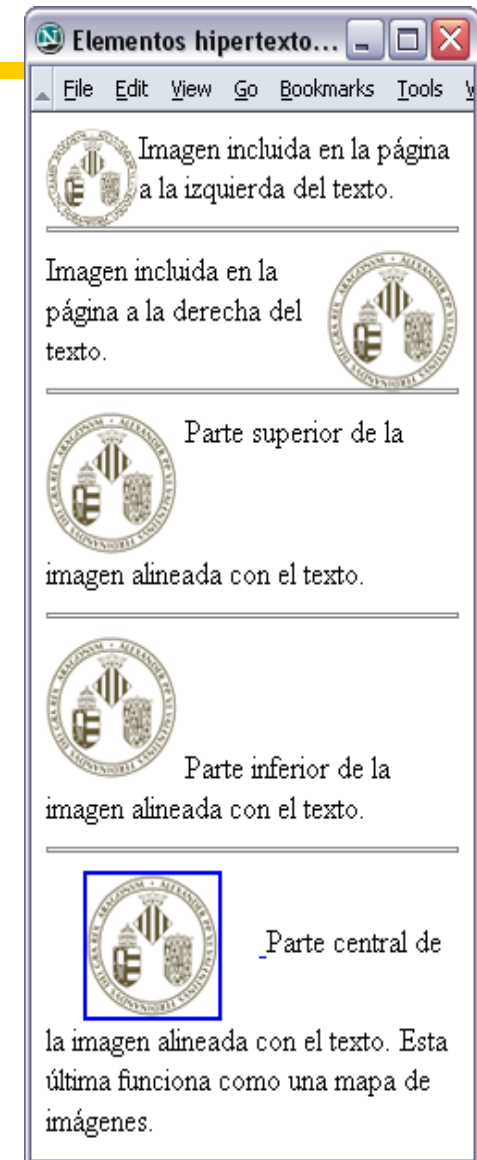

```

```
<map name="ejemplo">
  <area shape="circle" coords="67,58,53" href= "http://www.eis.uva.es" alt="p1" />
  <area shape="rect" coords="140,0,256,112" href= "http://www.uva.es" alt="p2" />
  <area shape="rect" coords="113,116,148,147" href="introHTML.html#indice" alt= "inicio" />
</map>
```

## 4. Elementos hipertexto-hipermedia.

### Imágenes (III)

```
....  
  
Imagen incluida en la página a la izquierda del  
texto.  
<hr />  
  
Imagen incluida en la página a la derecha del texto.  
<hr />  
  
Parte superior de la imagen alineada con el texto.  
<hr />  
  
Parte inferior de la imagen alineada con el texto.  
<hr />  
<a href="www.uv.es/cgi-bin/mapa.cgi">  
    
</a>  
Parte central de la imagen alineada con el texto.  
Esta última funciona como una mapa de  
imágenes.  
....
```





## 5. Listas de términos.

### Listas marcadas (I)

---

- Hay tres tipos de listas: marcadas, numeradas y de definiciones.
- Listas marcadas:
  - Listas de elementos donde a la izquierda de cada elemento aparece una marca.
  - Vienen delimitadas con `<ul> .. </ul>.`
  - Cada uno de sus elemento se delimita con `<li> .. </li>`

**Atributos:** `type`, con los valores: "disc", "circle" y "square".

```
<ul type="square">  
  <li>Coffee</li>  
  <li>Tea</li>  
</ul>
```

Si no se indica nada, por defecto se usa "disc".

Uno de los elementos de la lista puede ser a su vez una nueva lista.

## 5. Listas de términos.

# Listas marcadas (II)

....

<body>

Ejemplo de lista:

<ul>

<li> primer &iacute;tem </li>

<li> segundo &iacute;tem </li>

<li> tercer &iacute;tem

<ul>

<li> sub&iacute;tem 1 </li>

<li> sub&iacute;tem 2

<ul>

<li> subsub&iacute;tem 1 </li>

<li> subsub&iacute;tem 2 </li>

</ul></li>

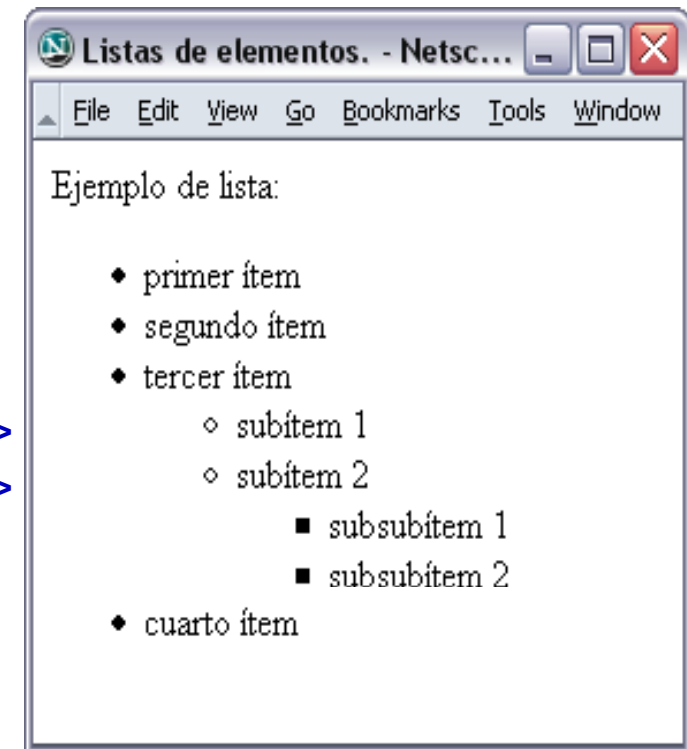
</ul></li>

<li> cuarto &iacute;tem </li>

</ul>

</body>

</html>



## 5. Listas de términos

### Listas numeradas (I)

---

- Listas numeradas.
  - Se delimitan con `<ol>... </ol>`
  - y sus elementos con `<li> ... </li>`

**Atributos:** `type`, `start` (índice inicial).

- `type = "1"` Números arábigos.
- `type = "I"` Números romanos (mayúsculas).
- `type = "i"` Números romanos (minúsculas).
- `type = "A"` Letras mayúsculas.
- `type = "a"` Letras minúsculas.

Por defecto todas las listas se numeran con números arábigos.

## 5. Listas de términos

### Listas numeradas (II)

.....

<body>

Ejemplo de lista:

<ol type="I">

<li>primer item </li>

<li>segundo item </li>

<li>tercer item

<ol type="A" start="4">

<li>subitem 1 </li>

<li>subitem 2

<ol>

<li>subsubitem 1 </li>

<li>subsubitem 2 </li>

</ol></li>

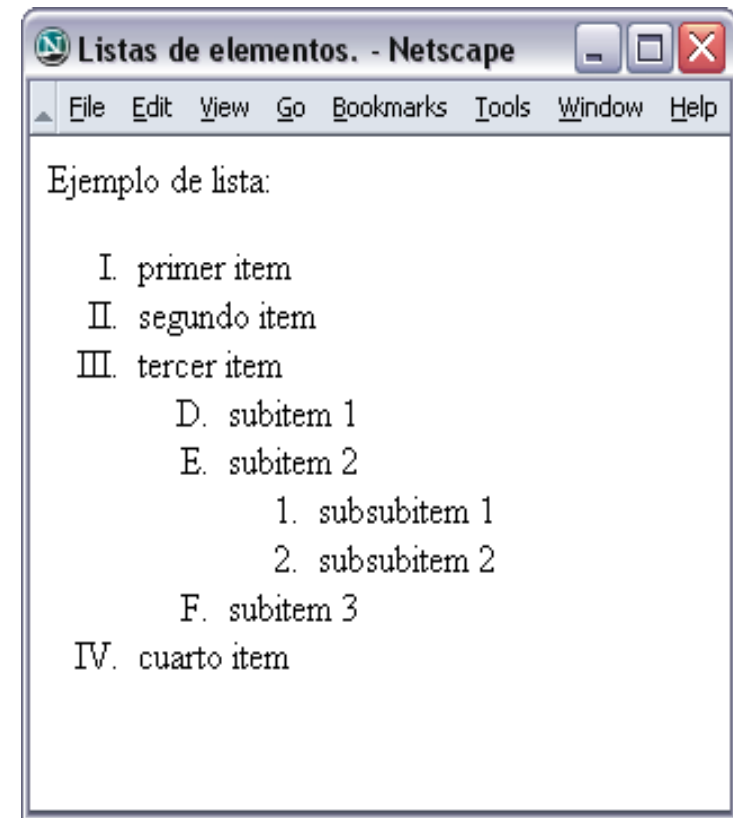
<li> subitem 3 </li>

</ol></li>

<li> cuarto item </li>

</ol>

</body>



## 5. Listas de términos

### Listas de definiciones

- Listas a dos niveles; en la primera línea el nombre, en las siguientes la definición.
  - Están delimitadas por `<dl> .. </dl>.`
  - El título por `<dt> ... </dt>` y
  - la definición por `<dd> .. </dd>.`

.....

```
<body>
```

Ejemplo de lista:

```
<dl>
```

```
  <dt>Primer término </dt>
```

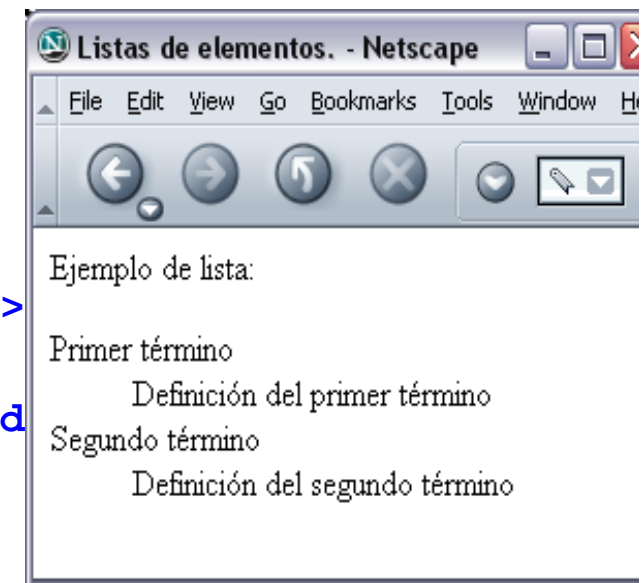
```
  <dd>Definición del primer término </dd>
```

```
  <dt>Segundo término </dt>
```

```
  <dd>Definición del segundo término </dd>
```

```
</dl>
```

```
</body>
```



## 6. Tablas.

### Definición de la tabla y de sus filas

---

- **<table> .. </table>**: Delimita la tabla de valores.

#### Atributos:

- **width**: Anchura de la tabla (medida en píxeles o en %).
  - **border**: Anchura, en píxeles, del borde exterior.
  - **cellspacing**: espacio medido en píxeles entre celdas adyacentes.
  - **cellpadding**: espacio, medido en píxeles, entre los datos y el borde de la celda.
  - **frame**: Tipo de borde exterior (box, void, above, bellow, hside, lhs, rhs, vsides).
- **<tr> .. </tr>**: Delimita una fila de la tabla.

#### Atributos:

- **align**: Alineación horizontal de los datos dentro de las celdas.
- **valign**: Alineación vertical de los datos en las celdas.
- **bgcolor**: Color de fondo de las celdas de la fila.

## 6. Tablas.

### Elementos de la tabla

---

- `<th> .. </th>`: Celda de cabecera.
- `<td> .. </td>`: Celda de datos.

**Atributos** (comunes a `th` y `td`):

- **colspan**: Define una entrada de datos que ocupa varias columnas.
- **rowspan**: Define una entrada de datos que ocupa múltiples filas.
- **align**: Alineación horizontal de los datos dentro de la celda (left, right, center y justify).
- **valign**: Alineación vertical de los datos en la celda (top, middle, bottom).
- **width** y **height**: ancho y alto de la celda, medido en píxeles.
- **bgcolor**: Color de fondo de la celda.

## 6. Tablas.

### Ejemplos (I)

.....

<body>

Ejemplo de una partida de tres en ralla:

```
<table border="2" cellspacing="3" cellpadding="5"
frame="void">
```

```
<tr>
```

```
<th bgcolor="red">O</th><th>O</th><th>X</th>
```

```
</tr>
```

```
<tr>
```

```
<th>X</th><th bgcolor="red">O</th><th>O</th>
```

```
</tr>
```

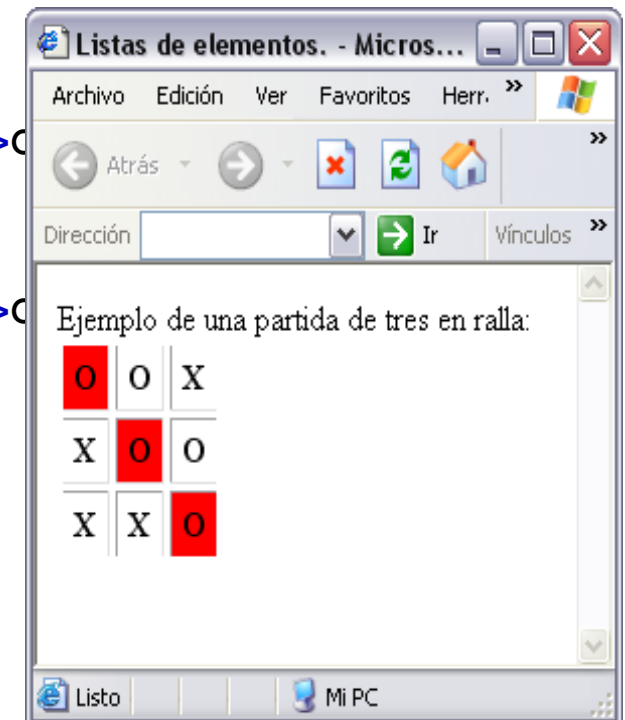
```
<tr>
```

```
<th>X</th><th>X</th><th bgcolor="red">O</th>
```

```
</tr>
```

```
</table>
```

</body>





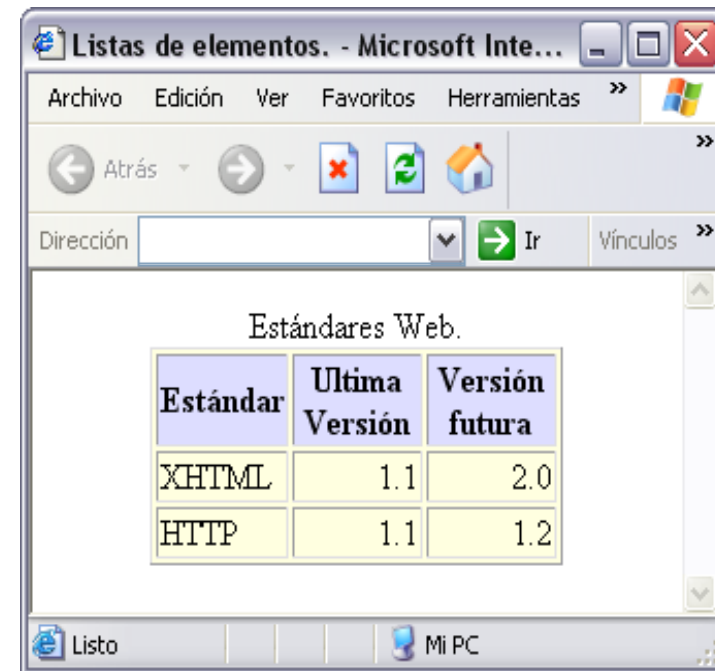
## 6. Tablas.

### Ejemplos (II)

....

```
<div align="center">
  Est&aacute;ndares Web.
  <table border="1" bgcolor="lightyellow" width="200">
    <tr bgcolor="#dddddff">
      <th>Est&aacute;ndar</th>
      <th>Ultima Versión</th>
      <th>Versión futura</th>
    </tr>
    <tr>
      <td>XHTML</td>
      <td align="right">1.1</td>
      <td align="right">2.0</td>
    </tr>
    <tr align="right">
      <td align="left">HTTP</td>
      <td>1.1</td>
      <td>1.2</td>
    </tr>
  </table>
</div>
```

....



## 6. Tablas.

### Ejemplos (III)

....

```
<div align="center">
```

```
Est&aacute;ndares Web.
```

```
<table border="1" bgcolor="lightyellow" width="200">
```

```
<tr bgcolor="#ddddff">
```

```
<th> Est&aacute;ndar</th>
```

```
<th colspan= "2"> Ultima Versi3n Versi3n futura</th>
```

```
</tr>
```

```
<tr>
```

```
<td rowspan="2">XHTML HTTP</td>
```

```
<td align="right">1.1</td>
```

```
<td align="right">2.0</td>
```

```
</tr>
```

```
<tr align="right">
```

```
<td>1.1</td>
```

```
<td>1.2</td>
```

```
</tr>
```

```
</table>
```

```
</div>
```

....

Est&aacute;ndares Web.

Est&aacute;ndar	Ultima Versi3n Versi3n futura	
XHTML	1.1	2.0
HTTP	1.1	1.2

## 7. Frames.

### División de la ventana en Frames (I)

---

- Con el ayuda de los frames se puede dividir la ventana del navegador en varias celdas rectangulares, cada una de ellas asociadas a un documento XHTML distinto.
- El tipo del documento debe ser “**Frameset DTD**”.
- **<frameset> ... </frameset>**  
Número de frames y tamaño de cada uno, en el que se divide, bien la ventana, o bien un frame anterior.

#### Atributos:

- **rows**: divide la ventana o un frame anterior, horizontalmente en nuevos frames. Los valores incluidos entre comillas, y separados por comas, indican el número de frames y su tamaño (pixels, %, \*).  
**P.e:** `<frameset rows="20%,*,400"> ... </frameset>`
- **cols**: idem que rows, pero esta vez la división es vertical.

## 7. Frames.

# División de la ventana en Frames (II)

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Frameset//EN"
    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-frameset.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="es" lang="es">
<head>
    <title>Frames</title>
</head>
<frameset rows="60%,40%">
    <frameset cols="*,*,*">
        <frame src="ejemplo1.html" />
        <frame src="ejemplo2.html" />
        <frame src="ejemplo3.html" />
    </frameset>
    <frameset cols="200,*">
        <frame src="ejemplo4.html" />
        <frame src="ejemplo5.html" />
    </frameset>
</frameset>
</html>
```



## 7. Frames.

### Contenido del Frame (I)

---

- **<frame />**

Indica el contenido de cada frame.

**Atributos:**

- **src**: Dirección URL del documento que ocupará el frame.
- **name**: nombre identificativo del frame. Se hará referencia a él dentro del atributo **target** de las directivas: <a>,<area>,<base> y <form>.
- **marginwidth**: márgenes a la izquierda y a la derecha.
- **marginheight**: márgenes arriba y abajo.
- **noresize**: bloquea la posibilidad de modificar el tamaño.
- **scrolling**: Determina si existe barra de scroll (auto, yes, no).

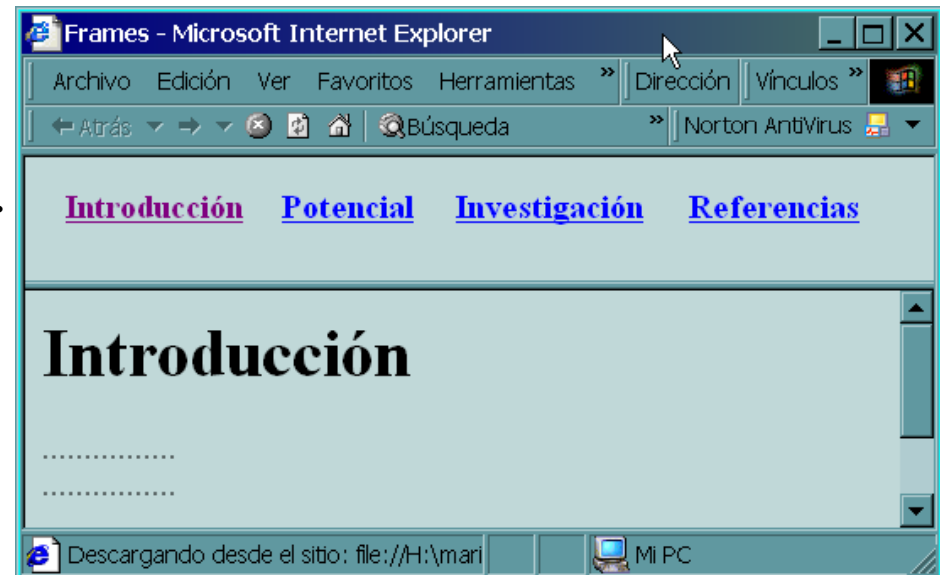
- **<noframe> ... </noframe>**

Texto alternativo que aparecerá en aquellos browsers sin soporte para frames.

## 7. Frames.

# Contenido del Frame (II)

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Frameset//EN"
    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-frameset.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="es" lang="es">
<head>
    <title>Frames</title>
</head>
<frameset rows="75,*">
    <frame src="menu.xml" name="Menu" />
    <frame src="introduccion.html" name="Principal" />
<noframes>
    <body>
        Esta página requiere
        soporte para frames
        para poder ser visualizada.
    </body>
</noframes>
</frameset>
</html>
```



## 7. Frames.

### Contenido del Frame (III)

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="es" lang="es">
  <head>
    <title>menu.xml</title>
  </head>
  <body>
    <table width="100%">
      <tr>
        <th><a href="introduccion.html" target="Principal">
          Introducción</a></th>
        <th><a href="potencial.html" target="Principal">
          Potencial</a></th>
        <th><a href="investigacion.xml" target="Principal">
          Investigación</a></th>
        <th><a href="referencias.xml" target="Principal">
          Referencias</a></th>
      </tr>
    </body>
  </html>
```

## 7. Frames.

### Frames Flotantes - iFrames

---

- Desventajas de los Frames
  - Tiempo de carga más lento, múltiples peticiones.
  - Problemas con los buscadores, el index.html está vacío de contenido.
- Solución: Emplear iFrames
  - Permite insertar una página web dentro de otra.
  - Se usa sobretodo para banners de publicidad.
  - Sigue manteniendo los problemas de usabilidad y accesibilidad de los Frames.



## 7. Frames.

### iFrames - Ejemplo

---

- Definiremos el iframe de la siguiente forma:

```
<iframe src="news.html" width="100%" height="160" align="center">  
    Texto alternativo para los usuarios que no ven iFrames.  
</iframe>
```

- Podemos aplicarles hojas de estilo para establecer un tamaño y posición adecuados.

```
<iframe id="miframe" name="miframe" frameborder="0"  
    framespacing="0" scrolling="auto" border="0"  
    style="position:absolute; left:726px; top:231px; width:216;  
    height:250; z-index:5">
```

```
    Texto alternativo para los usuarios que no ven iFrames.  
</iframe>
```

## 7. Frames.

### iFrames - Abrir url

---

- Para abrir una nueva url en un iframe se emplea el atributo target de los enlaces. De puede hacer referencia a urls locales y externas.

`<a href="pagina2.html" target="miframe"> Enlace </a>`

`<a href="http://www.uv.es" target="miframe"> Enlace </a>`

- `//iFrame`  
`<iframe id="miframe" name="miframe" >`  
`</iframe>`

## 8. Capas

# Capas

---

- Divide la página web en regiones sin necesidad de usar frames.
- `<div> ... </div>`
- Podremos desplazar de forma dinámica las capas por la página web, cambiar su contenido e incluso hacer que aparezcan y desaparezcan, pero para ello necesitamos emplear Javascript.
- Tiene multitud de parámetros:
  - id, class, title, style, dir, lang.
- Podemos aplicarles hojas de estilo lo que nos proporciona multitud de diseños diferentes y no fijos como en los frames.
- Tendremos en una página todas las divisiones, no necesitamos múltiples páginas.
- No tendremos los problemas de accesibilidad y usabilidad propios de los frames e iframes (impresión, bookmarks, ...).
- También podemos emplear las capas para definir tablas.

## 8. Capas

### Ejemplo

- La etiqueta a emplear es la siguiente:

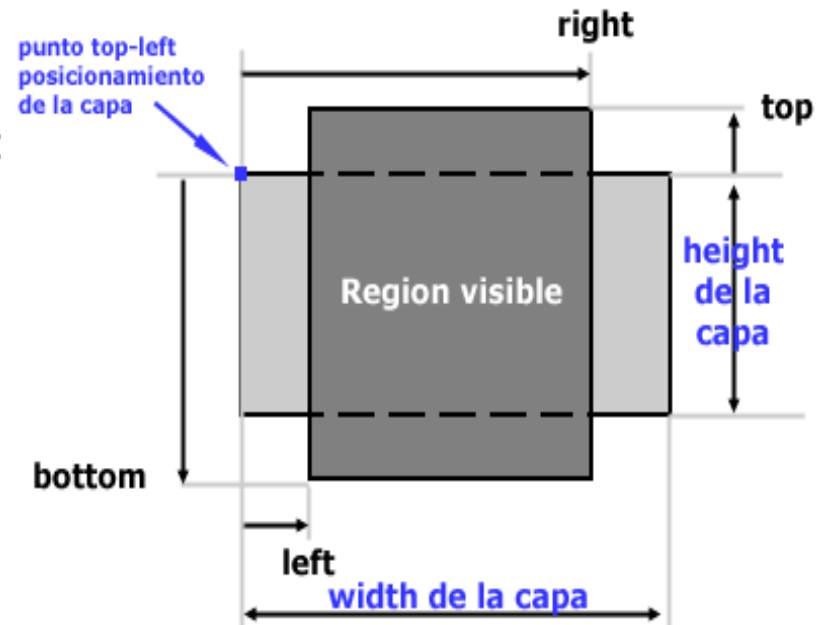
```
<div id="capa" >  
    Hola!  
</div>
```

- Podemos aplicar estilos.

```
<div style="clip: rect(0,158,148,15); height: 250px; width: 170px; left:  
    10px; top: 220px; position: absolute; visibility: visible; z-index:10;  
    font-size: 14pt; font-family: verdana; text-align: center; background-  
    color: #bbbbbb">
```

...

```
</div>
```



## 8. Capas

# Eventos

---

- Las capas pueden generar eventos, los cuales pueden ser gestionados a través de javascript.
  - onclick
  - ondblclick
  - onmousedown
  - onmouseup
  - onmouseover
  - onmousemove
  - onmouseout
  - onkeypress
  - onkeydown
  - onkeyup

## 8. Capas

# Javascript (Introducción I)

---

- Permite generar páginas dinámicas **en el lado del cliente**.
- Sensible a mayúsculas y minúsculas: `datos1` y `Datos1` son diferentes.
- No se define el tipo de las variables, va cambiando dependiendo del valor que se le asigna.
- Las variables se suelen definir con **var**.
  - **var datos1;**
- Si no se emplea **var** se consideran variables globales
  - **datos1;**
- Las variables definidas con **var** dentro de una función **serán locales a esa función**.
- Las variables definidas con **var** fuera de cualquier función **se consideran globales**.

## 8. Capas

# Javascript (Introducción II)

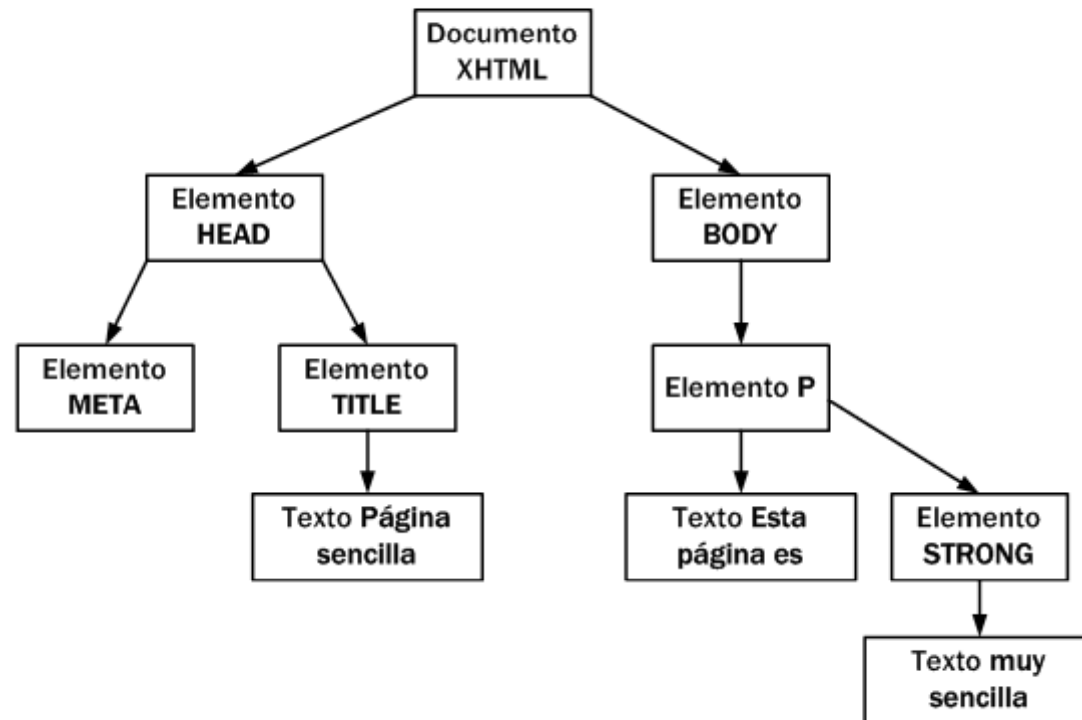
---

- Se pueden definir funciones, **function**.
  - Las funciones pueden devolver valores con **return**.
- Dispone de los operadores habituales: =, +, -, \*, /, ++, --, ==, ===, !=, !==, <, >, &&, ||, !, ...
- Existen algunos objetos definidos: String, Array, Math, Date, ...
- Existen las estructuras de control habituales: if, if ... else, for, while, switch, ...
- Existen los comentarios habituales: // Comentario de 1 línea  
/\* Comentario de bloque \*/

## 8. Capas

# Javascript (Introducción III)

- DOM Javascript





## 8. Capas

# Javascript (Introducción III)

### ■ Acceso a los elemento del DOM

Los principales nodos son los siguientes:

- Document: nodo raíz del que cuelgan todos los nodos
- Element: nodo que representa a cada una de las etiquetas XHTML.
- Attr: nodo que representa a los atributos de las etiquetas XHTML.
- Text: nodo que representa al texto encerrado por una etiqueta.
- Comment: nodo que representa los comentarios incluidos en la página XHTML.

### ■ Para acceder a los diferentes nodos se emplea la función `document.getElementsByTagName();`

```
... //Página XHTML
```

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

```
...
```

```
...//Javascript
```

```
var cabecera = document.getElementById("cabecera");
```

## 8. Capas

# Javascript (Introducción IV)

- Ejemplo básico

```
//Javascript
```

```
<!--
```

```
function HolaMundo()
```

```
{
```

```
    alert("Hola Mundo");
```

```
}
```

```
//-->
```

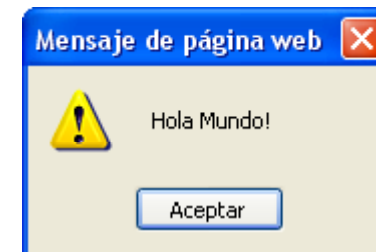
**Uso: //XHTML**

```
<body>
```

```
    <a onClick="holaMundo()"> Pulsa para ver el mensaje </a>
```

```
</body>
```

```
</html>
```



## 8. Capas

# Javascript y Capas(I)

---

- Atributos de las Capas:
  - style: almacena los estilos asociados a esa capa (CSS)
    - visibility (hidden|visible) : oculta o muestra la capa.
    - width: almacena la anchura de la capa.
    - height: almacena la altura de la capa.
  - innerHTML: almacena el contenido de la capa.

## 8. Capas

# Javascript y Capas(II)

---

- Mostrar y Ocultar Capas

- Atributo visibility;

```
//Javascript
```

```
<!--
```

```
function mostrar(capa)
```

```
{
```

```
    var obj = document.getElementById(capa);
```

```
    if (obj.style.visibility == "hidden")
```

```
        obj.style.visibility = "visible";
```

```
    else
```

```
        obj.style.visibility = "hidden";
```

```
}
```

```
//-->
```

**Uso: //XHTML**

```
    mostrar(nombreCapa);
```

## 8. Capas

# Javascript y Capas(III)

---

- Modificar tamaño capa
  - Atributos width y height;

//Javascript

<!--

function tamCapa(capa, ancho, alto)

{

var obj = document.getElementById(capa);

obj.style.width = ancho;

obj.style.height = alto;

}

//-->

**Uso: //XHTML**

mostrar(nombreCapa);

## 8. Capas

# Javascript y Capas (IV)

---

- Modificar el contenido de una capa
  - Atributo innerHTML;

<!--

```
function cambiarContenido(capa, texto)
{
    var obj = document.getElementById(capa);
    obj.innerHTML = texto;
}
```

//-->

**Uso:**

```
cambiarContenido(nombreCapa, texto);
```

## 8. Capas

# Javascript y Capas (V)

- Cargar url local en la capa.

```
//Javascript
```

```
<!--
```

```
function nuevaConexion()
```

```
{
```

```
    var xmlhttp=false;
```

```
    try {
```

```
        xmlhttp = new ActiveXObject("Msxml2.XMLHTTP");
```

```
    }
```

```
    catch (e)
```

```
    {
```

```
        try {
```

```
            xmlhttp = new ActiveXObject("Microsoft.XMLHTTP");
```

```
        }
```

```
        catch (E)
```

```
        {
```

```
            xmlhttp = false;
```

```
        }
```

```
    }
```

```
    if (!xmlhttp && typeof XMLHttpRequest!='undefined')
```

```
    {
```

```
        xmlhttp = new XMLHttpRequest();
```

```
    }
```

```
    return xmlhttp;
```

```
}
```

```
ARS - 2011 <!-->
```

XHTML

## 8. Capas

# Javascript y Capas (V)

- Cargar url local en la capa.

```
//Javascript
```

```
<!--
```

```
function Cargar(url,capa)
{
    var contenido = document.getElementById(capa);
    conexion = nuevaConexion();
    conexion.open("GET", url, true);
    conexion.onreadystatechange=function()
    {
        if(conexion.readyState == 4)
        {
            contenido.innerHTML = conexion.responseText;
        }
    }
    conexion.send(null);
}
```

```
//XHTML
```

```
<a onclick="Cargar('Ej0.html','capa1')">
    Enlace 1
```

```
</a>
```



## 8. Capas

# Javascript y Capas (VI)

- Abrir respuesta formulario en una capa.

```
//Javascript
<!--function CargarForm( url, capa, valores)
{
    var contenido = document.getElementById(capa);
    conexion = nuevaConexion();
    conexion.open("POST", url,true);
    conexion.onreadystatechange=function()
    {
        if(conexion.readyState == 4)
        {
            contenido.innerHTML = conexion.responseText;
        }
    }
    conexion.setRequestHeader('Content-Type','application/x-www-form-urlencoded');
    conexion.send(valores);
}
-->
```

## 8. Capas

# Javascript y Capas (VII)

- Abrir respuesta formulario en una capa.

//Javascript

```
<!-- function ProcesarForm(formulario, url, capa)
```

```
{
```

```
    valores="";
```

```
    for (i=0; i<formulario.elements.length;i++)
```

```
    {
```

```
        nombre = formulario.elements[i].name;
```

```
        if (nombre!="")
```

```
        {
```

```
            if ((formulario.elements[i].type == "radio") && (!formulario.elements[i].checked))
```

```
                ;
```

```
            else
```

```
            {
```

```
                valores += formulario.elements[i].name + "=";
```

```
                valores += formulario.elements[i].value + "&";
```

```
            }
```

```
        }
```

```
        CargarForm(url, capa, valores);
```

```
    } //-->
```

//XHTML

```
<form method="post" onsubmit="ProcesarForm(this, 'Registro','capa1'); return false">
```

```
    ..... //Elementos del formulario
```

```
</form>
```

## 9. Formularios.

### Estructura principal.

---

- **<form> ... </form>**

**Atributos:**

- **action:** acción a emprender. Generalmente hace referencia a la dirección url del cgi encargado de interpretar la entrada.
- **method:** método de transferencia: GET o POST.
  - GET:** Añade los datos (del formulario) al url fijado en action, separado por un ?. El programa-cgi recibe los datos a partir de la variable de entorno QUERY\_STRING (máx. 200 caracteres).
  - POST:** Envía los datos a parte, después de la cabecera HTTP. El programa cgi recibe los datos a partir de la entrada estándar.
- **enctype:** mecanismo para codificar el contenido del formulario, por defecto: "application/x-www-form-urlencoded"
- **name:** nombre identificativo del formulario (javascript).
- **target:** frame donde se visualiza el resultado

## 9. Formularios.

### Directiva Input (I)

---

- **<input />:**
  - La directiva input es la más usada y la que más opciones y funcionalidades ofrece.
  - Dependiendo del valor del atributo TYPE, se distinguen varias clases de entrada.
- **<input type="text" name=".." .. />:**
  - Es el valor por defecto de type. Corresponde a una entrada de datos formada por una línea de texto.

#### Atributos:

- **name:** nombre de la variable.
- **value:** valor inicial.
- **size:** anchura de la línea de texto.
- **maxlength:** Numero máximo de caracteres

## 9. Formularios.

### Directiva Input (II)

#### Ejemplo:

```
<form action="http://maquina/programa.cgi" method="GET">
```

```
Nombre: <input type="text" name="nom" value="Antonio" size="20"  
        maxlenght="30" /> <br />
```

```
Apellidos: <input type="text" name="ape" size="25" maxlenght="50" />  
<p> &nbsp; <p/>
```

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

```
</form>
```



## 9. Formularios.

### Directiva Input (III)

- `<input type="password" name=".." .. />`

Entrada de texto con los mismo atributos que TEXT. La única diferencia con TEXT es que se ocultan los caracteres de la entrada con asteriscos.

Ejemplo:

```
<form action="http://maquina/programa.cgi" method="POST">
```

```
<div align="center">
```

Nombre de usuario:

```
<input type="text" name="usuario"
```

```
value="Anonimo" size="20" maxlength="30"
```

Clave de acceso:

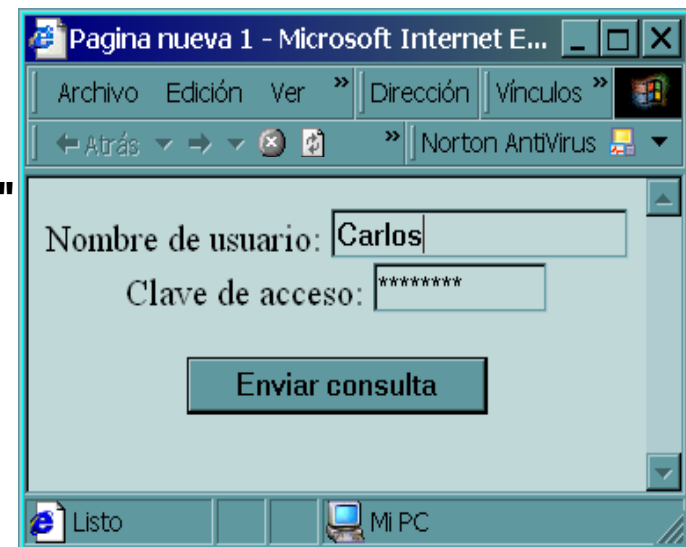
```
<input type="password" name="clave"
```

```
size="10" maxlength="10" />
```

```
<p><input type="submit" /></p>
```

```
</div>
```

```
</form>
```



## 9. Formularios.

### Directiva Input (IV)

---

- `<input type="checkbox" name=".." .. />`

Casilla de selección. Sólo caben dos posibles valores: seleccionada (value) o no seleccionada (cadena vacía).

#### Atributos:

- **name**: nombre de la variable.
- **value**: valor enviado cuando la casilla ha sido seleccionada.
- **checked**: inicialmente aparece seleccionada.

```
<input type="checkbox" name="correo" value="1"
      checked="checked">
```

Deseo recibir periódicamente  
información sobre nuevos productos.<br>

☒ Deseo recibir periódicamente inf

## 9. Formularios.

### Directiva Input (V)

---

- `<input type="radio" name=".." ../>`

Botones de radio. Una misma variable asociados a varios elementos, cada uno de ellos con un valor distinto.

#### Atributos:

- **name**: nombre de la variable.
- **value**: valor enviado cuando el botón está seleccionada.
- **checked="checked"**: botón inicialmente seleccionado.



## 9. Formularios.

### Directiva Input (VI)

```
<hr />
```

```
Envio:<br />
```

```
<input type="radio" name="correo" value="1" checked="checked" />
```

```
Correo urgente.<br />
```

```
<input type="radio" name="correo" value="2" />
```

```
Correo ordinario.
```

```
<hr />
```

```
Destino:<br />
```

```
<input type="radio" name="continente"
```

```
value="europa" checked="checked" /> Europa. <br />
```

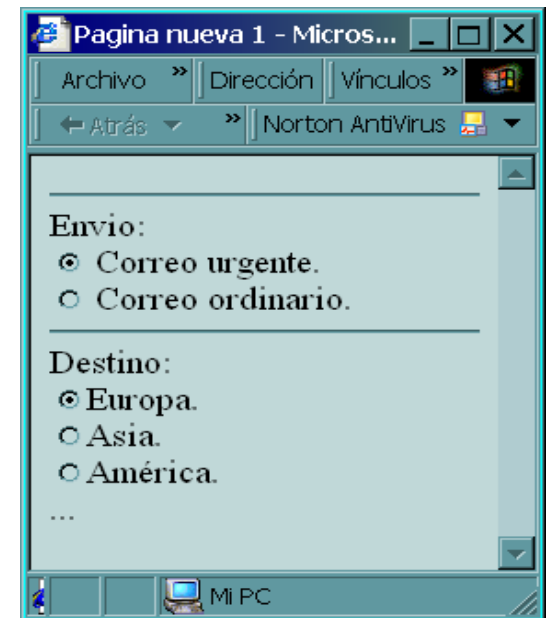
```
<input type="radio" name="continente"
```

```
value="asia" /> Asia. <br />
```

```
<input type="radio" name="continente" value="america">América.<br>
```

```
...
```

```
</form>
```



## 9. Formularios.

### Directiva Input (VII) - Botones

- **`<input type="reset">`**:
  - Inicializa todos los elementos del formulario (con sus valores por defecto).
  - **value**: Texto que aparecerá sobre el botón en lugar del mensaje "Reset".
- **`<input type="submit">`**:
  - Envía los datos del formulario (valores actuales).
  - **value**: Texto que aparecerá sobre el botón en lugar del mensaje "Submit".

...

```
<input type="reset" value="Vaciar el formulario" />  
<input type="submit" value="Enviar los datos" />
```

.....

Vaciar el formulario	Enviar los datos
----------------------	------------------

### Directiva Input (VIII) - Botones

- **`<input type="button">`**:
  - Botones javascript. Botones generales que activan llamadas a código javascript.

#### Atributos:

- **name**: Nombre del botón.
- **value**: Texto que aparece sobre el botón.
- **disabled**: Desactiva el botón.
- **onclick**, **ondblclick**, **onfocus**, **onblur**. Eventos reconocidos, utilizados para activar el código javascript correspondiente.

...

```
<input type="button" value="Comprueba Valores"  
      onclick="ValidForm()" />
```

...



## 9. Formularios.

### Área de texto (I)

---

- **<textarea> ... </textarea>:**
  - Variante de la entrada TEXT. Ahora la entrada del texto puede ocupar varias líneas. El texto incluido entre las directivas de apertura y cierre, es el texto que aparece por defecto en el área.

#### **Atributos:**

- **name:** Nombre de la variable.
- **rows** y **cols:** Tamaño del área de texto.

## 9. Formularios.

### Área de texto (II)

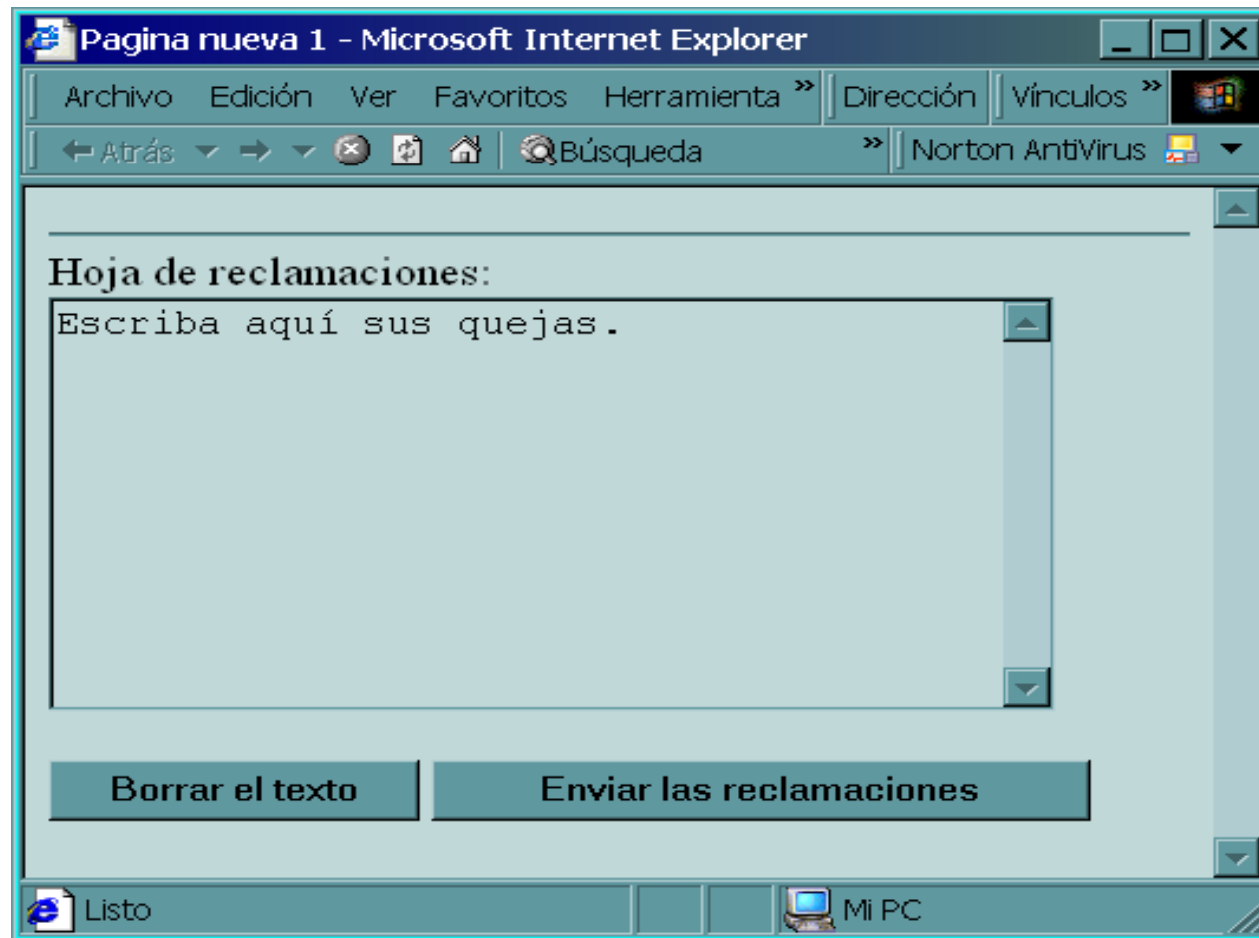
---

```
<form action="www.oficinadelconsumidor.org/reclamaciones.cgi"
  method="POST">
  <hr />
  Hoja de reclamaciones:<br />
  <textarea name="reclamacion" rows="10" cols="40">
    Escriba aquí sus quejas.
  </textarea> <br />
  <input type="reset" value="Borrar el texto" />
  <input type="submit" value="Enviar las reclamaciones" />
</form>
```

## 9. Formularios.

### Área de texto (III)

---



The screenshot shows a Microsoft Internet Explorer window titled "Pagina nueva 1 - Microsoft Internet Explorer". The menu bar includes "Archivo", "Edición", "Ver", "Favoritos", "Herramienta", "Dirección", and "Vínculos". The toolbar contains "Atrás", "Búsqueda", and "Norton AntiVirus". The main content area displays a form titled "Hoja de reclamaciones:". Below the title is a large text input field containing the placeholder text "Escriba aquí sus quejas.". At the bottom of the form are two buttons: "Borrar el texto" and "Enviar las reclamaciones". The status bar at the bottom shows "Listo" and "Mi PC".

## 9. Formularios.

### Combo Box y List Box (I)

---

- **<select> .. </select>:**
  - Sirve para generar las típicas combo box y list box.

#### Atributos:

- **name:** nombre de la variable.
- **size:** número de filas visibles.
- **multiple:** Indica que pueden ser seleccionadas varias entradas.

Para generar una Combo Box, la directiva **select** no debe incluir los atributos **size** ni **multiple**. En caso contrario, se genera una List Box.

- **<option value=".." [selected="selected"]>  
opcion1 </option>**

Cada una de las posibles opciones que aparecen dentro de la Combo Box o de la List Box.

## 9. Formularios.

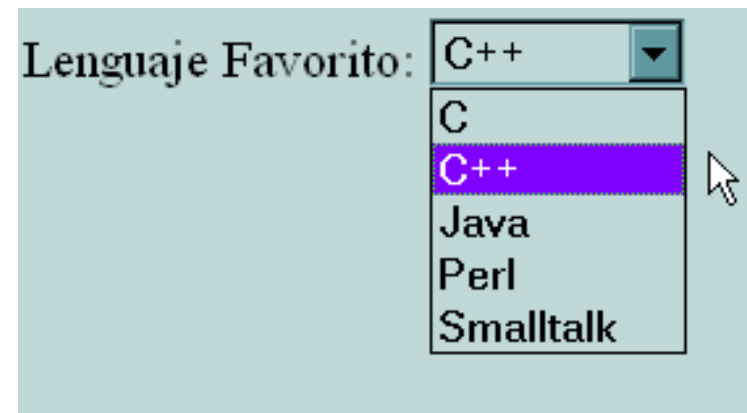
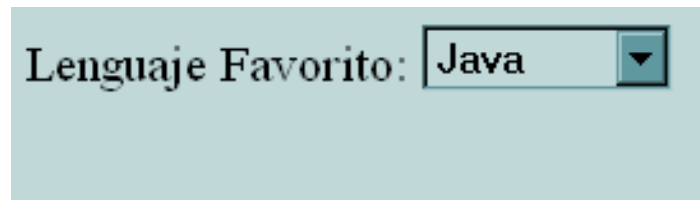
### Combo Box

...

Lenguaje Favorito:

```
<select name="leng_favo">  
  <option value="c" > C </option>  
  <option value="cpp"/> C++ </option>  
  <option value="java">  
    selected="selected" > Java </option>  
  <option value="perl" > Perl </option>  
  <option value="small" > Smalltalk </option>  
</select>
```

...





## 9. Formularios.

### List Box

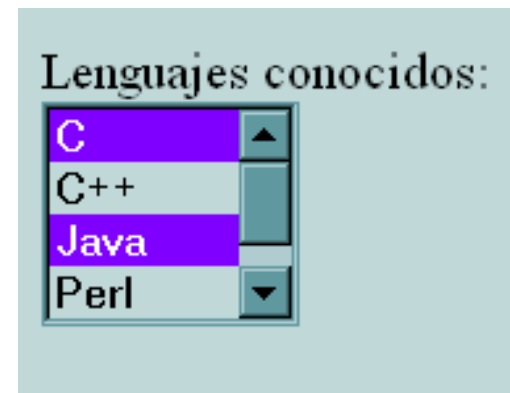
---

...

Lenguajes conocidos:<br>

```
<select name="leng_favo" size=4 multiple="multiple" >
  <option value="c" selected="selected" > C </option>
  <option value="cpp" > C++ </option>
  <option value="java" selected="selected" > Java </option>
  <option value="perl" > Perl </option>
  <option value="small" > Smalltalk </option>
</select>
```

...



## 10. Hojas de estilo.

### Introducción

---

- El XHTML dispone de directivas y atributos que permiten dar formato al documento.
  - Sin embargo, son poco potentes, engorrosos, y dificultan la lectura y compresión del código XHTML.
- Las Hojas de estilo son un conjunto de reglas de estilo definidas por el programador sobre elementos XHTML, que definen la apariencia final de esos elementos en la ventana del navegador.
- Beneficios de las hojas de estilo:
  - Potencian y dan mayor flexibilidad a la hora de definir el formato de elementos XHTML.
    - Permiten fijar fuentes de letra, tamaños, imágenes de fondo, etc.
  - **Permiten separar la apariencia del contenido del documento**
    - Hace el código más legible.
    - Permite compartir estilos entre múltiples documentos.

## 10. Hojas de estilo.

### Versiones

---

- CSS1 (1996).
  - Estándar universalmente aceptado. Soportado por los navegadores actuales.
  - <http://www.w3.org/TR/REC-CSS1>
- CSS2 (1998).
  - Especifica el soporte media para las hojas de estilo (browsers visuales, dispositivos auditivos, impresoras, dispositivos braille)
  - <http://www.w3.org/TR/REC-CSS21>
- CSS3 (en desarrollo).
  - Focalizado a la modularización de las especificaciones CSS.
  - <http://www.w3.org/Style/CSS/current-work>

## 10. Hojas de estilo.

# Sintaxis de las reglas de estilo

---

- Las hojas de estilo son una colección de reglas de estilo que se aplican sobre los elementos html/xhtml
  - Asocia una etiqueta con su aspecto

```
h1 {color:blue; font-size:25px;}
```

- Las reglas siguen la siguiente sintaxis:

- Forma general:

```
selector.clase {  propiedad1: valor1;
                  .....
                  propiedadN: valorN }
```

- Ejemplo:

```
h1 { text-align: center; font-style: italic }
h1.rojo {color: red} (CSS)
```

...

```
<h1> ... </h1> (XHTML)
```

```
<h1 class="rojo"> ... </h1>
```

## 10. Hojas de estilo.

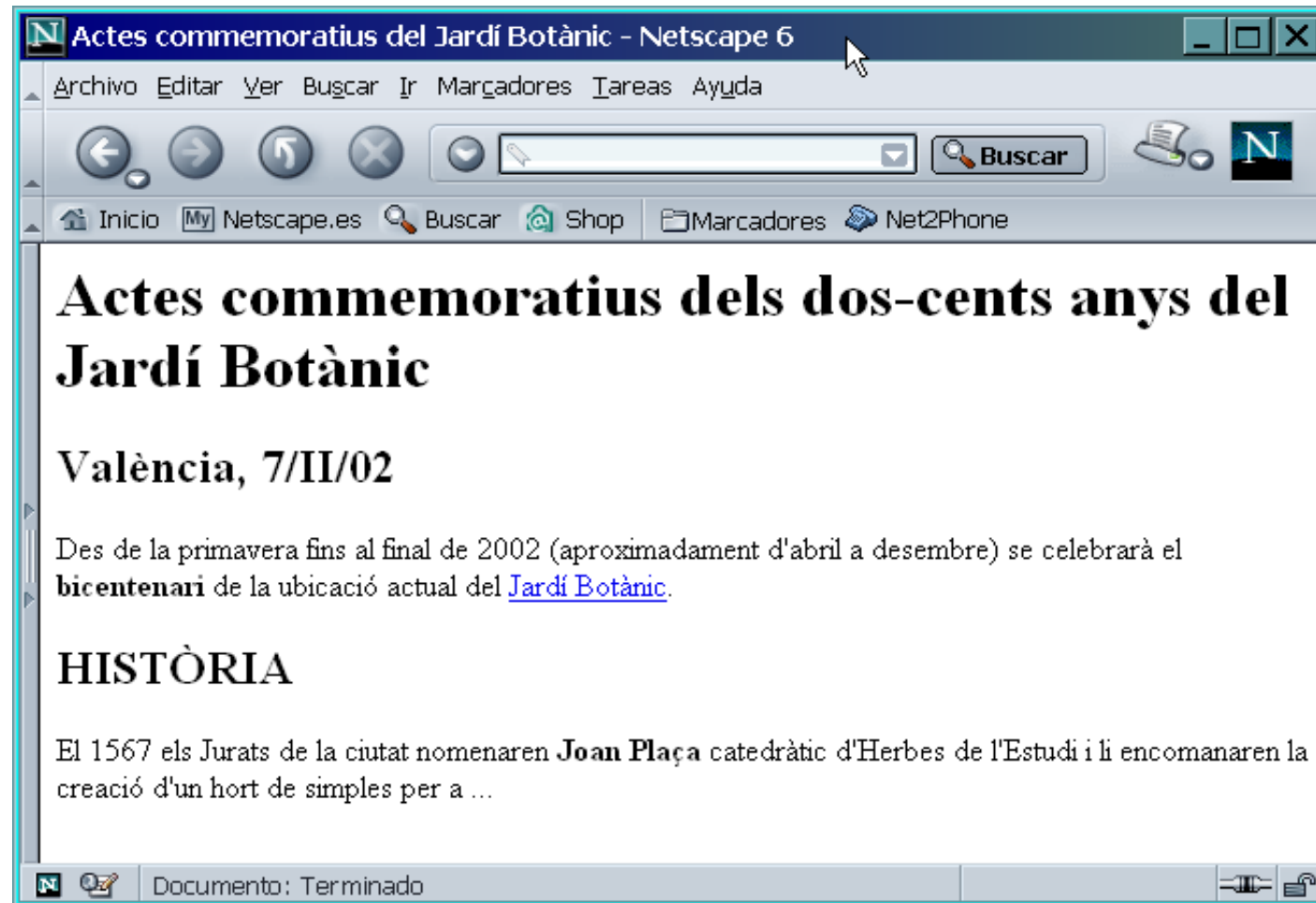
### Ejemplo sin CSS

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="es" lang="es">
<head>
    <title>Actes commemoratius del Jardí Botànic</title>
</head>
<body>
<h1>Actes commemoratius dels dos-cents anys del Jardí Botànic</h1>
<h2>València, 7/II/02</h2>
Des de la primavera fins al final de 2002 (aproximadament d'abril
    a desembre) se celebrarà el <strong>bicentenari</strong> de
    la ubicació actual del
<a href="http://www.uv.es/~uvalen/jardi.html">Jardí Botànic</a>.
<h2>HISTÒRIA</h2>
El 1567 els Jurats de la ciutat nomenaren <strong>Joan Plaça</strong>
    catedràtic d'Herbes de l'Estudi i li encomanaren la
    creació d'un hort de simples per a ...
</body>
</html>
```

## 10. Hojas de estilo.

### Ejemplo sin CSS (II)

---



## 10. Hojas de estilo.

### Ejemplo con CSS (I)

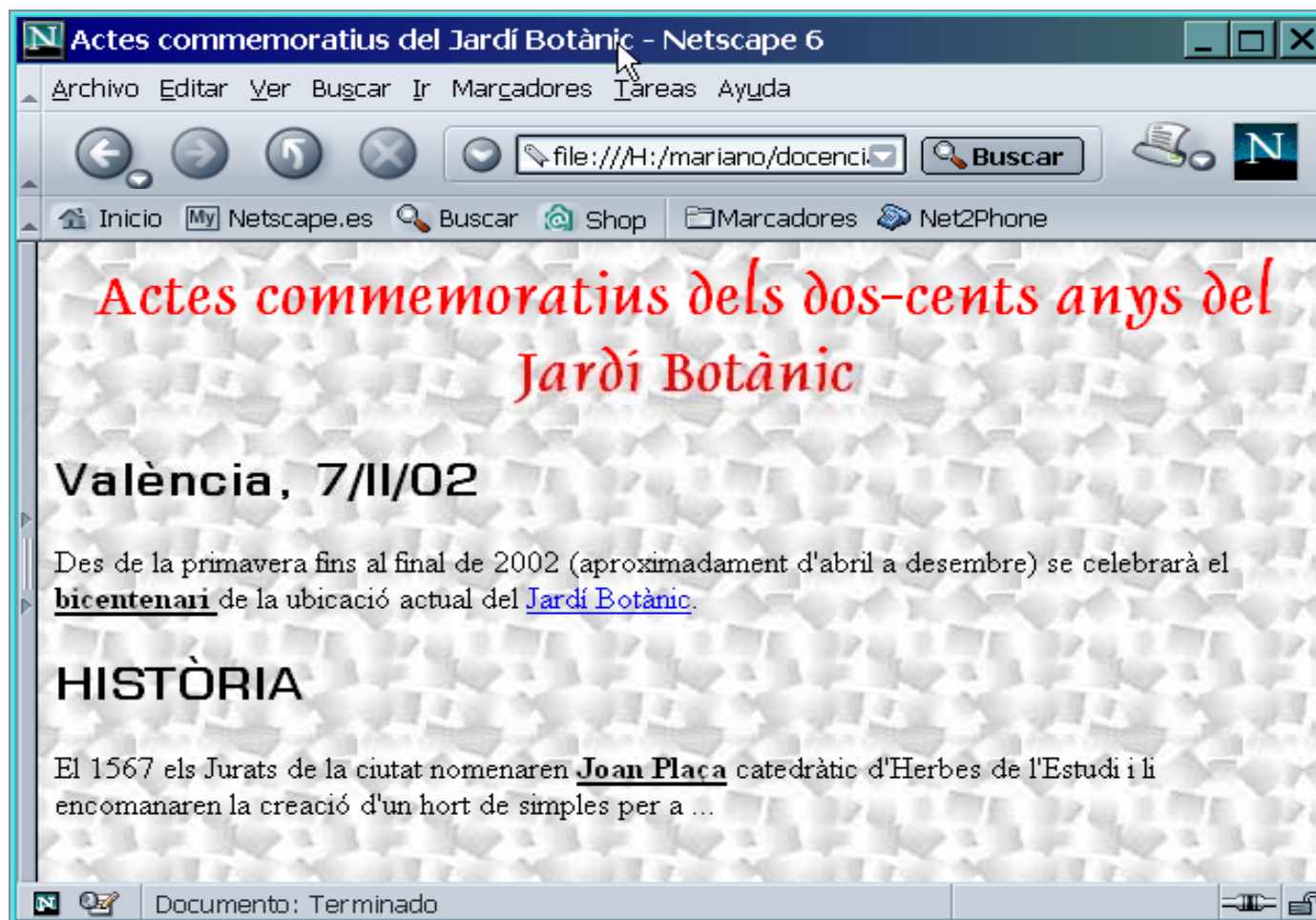
---

```
----- ejemplo.html -----  
...  
<head>  
  <title>Actes commemoratius del Jardí Botànic</title>  
  <link rel="stylesheet" href="estilo_ej.css" type="text/css"/>  
</head>  
<body>  
  .....  
</body>  
</html>  
----- estilo_ej.css -----  
body { background: URL(confetti-background.jpg) }  
h1 { text-align: center;  
    color: red;  
    font-family: Dauphin, "Book Antiqua", Arial}  
h2 { font-family: "Square 721 BT", Tahoma, Arial}  
strong { text-decoration: underline }
```

## 10. Hojas de estilo.

### Ejemplo con CSS (II)

---





## 10. Hojas de estilo.

### Selectores (I): Elementos XHTML

---

- Las reglas de estilo se definen utilizando la sintaxis:  
`selector.clase { propiedad: valor; ..;  
                  propiedad: valor }`
- Los selectores pueden ser: elementos XHTML o la pseudoclase enlace.
- Selectores formados por elementos XHTML:

```
body { color: blue }
p { text-align: center }
// También pueden agruparse
h1, h2, h3, h4, h5, h6 { color: red;
                        font-family: sans-serif }
// Los contextuales sólo se aplican en un
//contexto determinado
em { color: red }
h1 em { color: green }
```

## 10. Hojas de estilo.

### Selectores (II): Pseudoclase enlace

---

- El estándar CSS1 permite definir propiedades para cada uno de los tipos de enlaces:
  - `A:link` o `:link` - enlaces no visitados.
  - `A:visited` o `:visited` - enlaces visitados.
  - `A:active` o `:active` - enlaces activados.
  - `A:hover` o `:hover` - ratón sobre el enlace.

```
A:visited { color: red }
```

```
A:link img { border: solid green }
```

## 10. Hojas de estilo.

### Clases de estilo (I)

---

- Se puede crear una clase de estilo sobre un elemento XHTML, para que se aplique sólo a unas cuantas instancias de dicho elemento .
- Para definir una clase de estilo, se indica el elemento seguido del nombre de la clase:

```
p.miparrafo { margin-left: 1cm;  
               magin-right: 1cm;  
               font-style: italic }
```

- Para activar la clase definida, se utiliza el atributo `class`:

```
<p class="miparrafo">
```

```
Des de la primavera fins al .....
```

- La clase definida hereda también los estilos de los elementos genéricos (estilos de `<p>`, estilos de `body` ..)
- Se pueden activar varias clases de forma simultanea:  

```
<p class="miparrafo azul">
```

```
Des de la primavera fins al .....
```

## 10. Hojas de estilo.

### Clases de estilo (II)

---

- Para definir una clase de estilo global, se omite el nombre del elemento:

```
.azul { color: blue; font-weight: bold }
```

- Para utilizarla, simplemente se añade el atributo **class**, con el nombre de la clase, al elemento que se aplica:

```
<h2 class="azul">Cabecera azul</h2>
```

```
....
```

Este texto aparecería pintado con el color estándar, y `<span class="azul">` este otro en color azul. `</span>`

```
.....
```

## 10. Hojas de estilo.

### Aplicación de las reglas

- Las reglas se “heredan”, considerando el documento XHTML como una estructura jerárquica.

```
body {color:blue;}
p {font-size:15px;}
div {text-align: center;}
.....
<body>
...
<div>
...
  <p> Texto ... </p> //El texto aparecerá en azul y con
                      //tamaño de 15px
</div>
</body>
```

- Las reglas se aplican en Cascada.

```
p.rojo {color:red; font-size:25px;}
...
<p class="rojo"> Texto ... </p> //El texto aparecerá en
                                //rojo con tamaño 25px
```

## 10. Hojas de estilo.

### Aplicación de las reglas (II)

---

- Se pueden definir estilos anidados

```
p {color:blue;}           //Se aplica a todos los párrafos
div p {color: red;}       //Se aplica a los párrafos que se
                          //encuentren dentro de una capa (div)
```

- Ej. XHTML

```
<div>
  <p> Texto ... </p> //Texto rojo
</div>
...
<p> Texto ... </p> //Texto azul
```

## 10. Hojas de estilo.

### ¿Dónde colocar las reglas?: En línea

---

- Se utiliza el atributo **style** para cambiar directamente el estilo de cada elemento XHTML.

p.e.

```
<h2>Valencia, 7/II/02</h2>
<p style="margin-left: 1cm;
        magin-right: 1cm;
        font-style: italic">
    Des de la primavera fins al .....
    .....
</p>
```

## 10. Hojas de estilo.

### ¿Dónde colocar las reglas?: Estilo Importado

- En la cabecera del fichero XHTML se incluye un enlace al fichero de estilo:

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

- En el fichero de estilo se incluyen las reglas.

```
/* Ejemplo de un fichero de estilo */
body { background: URL(confetti-background.jpg) }
h1 { text-align: center;
     color: red;
     font-family: Dauphin, "Book Antiqua", Arial}
h2 { font-family: "Square 721 BT", Tahoma, Arial}
strong { text-decoration: underline }
```



## 10. Hojas de estilo.

### ¿Dónde colocar las reglas?: En la cabecera

- En la cabecera del fichero XHTML se incluye directamente las reglas de estilo.

```
<style type="text/css"> ... </style>
```

```
...  
<head>  
  <title>Actes commemoratius</title>  
  <style type="text/css">  
    <!--  
      h1 { text-align: center;  
          font-family: Arial}  
    -->  
  </style>  
</head>  
<body>  
  ....
```

## 10. Hojas de estilo.

### Propiedades

---

- Tipos de propiedades
  - Fuentes: tipos letra, tamaño, negrita, cursiva,...
  - Texto: justificación, espacio entre letras y/o palabras
  - Color: de texto, de fondo, imagen de fondo,...
  - Caja: bordes a los 4 lados
  - Otras.
- Unidades de medida
  - Palabras clave: **small**, **italic**, **Arial**, etc.
  - Unidades absolutas: **pt**(puntos), **cm**, **mm**, **in**(pulgadas)  
**px**(píxeles), etc.
  - Unidades relativas: **em**, **%**, etc. (resp. al tamaño por defecto)
  - URLs: **url(http://www.uv.es/logo.gif)**
  - Colores: **yellow**, **#ff0000**, **rgb(255,0,0)**

## 10. Hojas de estilo.

### Propiedades de las fuentes (I)

---

CSS agrupa las propiedades en categorías. A continuación se detalla las relacionadas con **las fuentes**:

- **font-weight**: Anchura relativa de la fuente.
  - **normal** | lighter | bold | bolder | 100 | 200 | ... | 900  
`h1 { font-weight: 200 }, h2 { font-weight: bold }`
- **font-style**: Apariencia de la fuente.
  - **normal** | italic | oblique  
`p { font-style: normal }, th { font-style: italic }`
- **font-size**: Tamaño (relativo o absoluto) de la fuente
  - pt, in, cm, mm | em, ex, px, % | xx-large | x-large | large | medium | small | x-small | xx-small | smaller | larger  
`strong {font-size: 150%}, p{font-size: 14pt}`
- **font-family**: fuente de letra.  
`h1 { font-family: "Book antique", Tahoma, Arial }`

## 10. Hojas de estilo.

# Propiedades de las fuentes (II)

....

```
<body>
```

```
<h1>Ejemplos de tipos de letra</h1>
```

En este primer ejemplo no se modifican el tipo de letra.

```
<h2>Cabecera 1</h2>
```

```
<h2>Cabecera 2</h2>
```

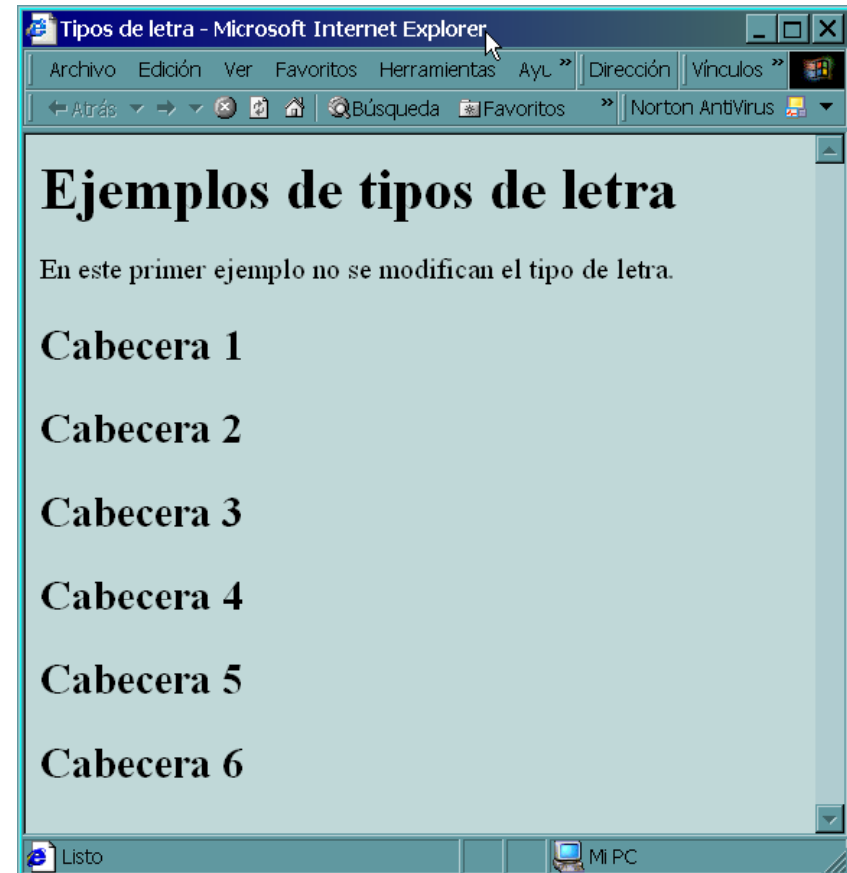
```
<h2>Cabecera 3</h2>
```

```
<h2>Cabecera 4</h2>
```

```
<h2>Cabecera 5</h2>
```

```
<h2>Cabecera 6</h2>
```

```
</body>
```



## 10. Hojas de estilo.

# Propiedades de las fuentes (III)

```
<h1>Ejemplos de tipos de letra</h1>
```

```
<p class="devinne"> En este segundo ejemplo han sido modificados  
los tipos de letra.</p>
```

```
<h2 class="Gothic">Cabecera 1</h2>
```

```
<h2 class="Decorated">Cabecera 2</h2>
```

```
<h2 class="Friz">Cabecera 3</h2>
```

```
<h2 class="Goudy">Cabecera 4</h2>
```

```
<h2 class="devinne">Cabecera 5</h2>
```

```
<h2 class="Firenze">Cabecera 6</h2>
```

```
-----  
h1 { text-align: center; font-family: "Dauphin" }
```

```
h2.Gothic { font-family: "Handel Gothic BT" }
```

```
h2.Decorated { font-family: "Decorated 035 BT" }
```

```
h2.Friz { font-family: "Friz Quadrata BT" }
```

```
.Goudy { font-family: "Goudy Handtooled BT" ; font-size: "300%" }
```

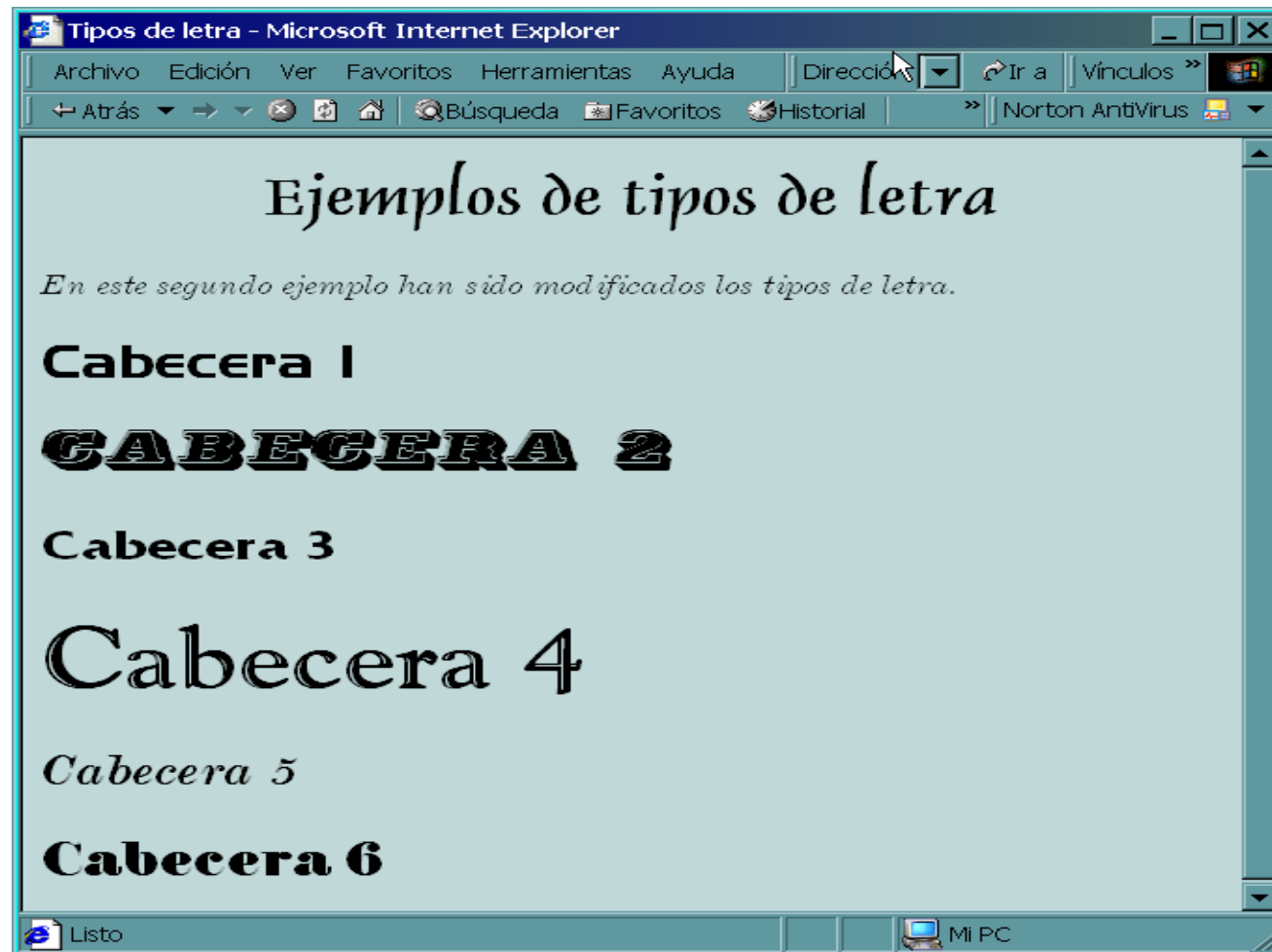
```
.devinne { font-family: "De Vinne BT"; font-style: italic }
```

```
.Firenze { font-family: "FirenzeITC-Normal"; font-weight:900 }
```

## 10. Hojas de estilo.

# Propiedades de las fuentes (IV)

---



## 10. Hojas de estilo.

### Propiedades del texto (I)

---

- **text-decoration**: añadidos al texto de un elemento.
  - **none** | underline | overline | line-through
  - ```
p { text-decoration: underline }
```
- **vertical-align**: alineación vertical del texto.
  - top | bottom | **baseline** | middle | sub | super
- **text-align**: alineación horizontal.
  - **left** | right | center | justify
- **text-indent**: indentación de la primera línea del texto.
  - +/- pt, in, cm, mm | +/- em, ex, px, %
  - ```
p { text-indent: -25px }
```
- **line-height**: distancia entre líneas del mismo párrafo.
  - **normal** | number | pt, in, cm, mm | em, ex, px, %

## 10. Hojas de estilo.

### Propiedades del texto (II)

---

```
//----- estilos.css -----  
body { background-image: URL(images/bond-paper.jpg) }  
p { margin-top: 5px }  
p.cabecera_pagina { text-align: right;  
                    text-decoration: underline;  
                    margin-right: 0.5in;  
                    font-family: Courier }  
  
p.rhead { text-align: right;  
          margin-right: 0.5in;  
          font-family: sans-serif }  
p.lhead { font-family: sans-serif }  
p.body { text-align: justify;  
         text-indent: 0.5in }  
p.foot { margin-left: 60%;  
        line-height: 300% }
```



## 10. Hojas de estilo.

### Propiedades del texto (III)

---

....

```
<head>
  <title>Propiedades del texto</title>
  <link rel="stylesheet" href="estilos.css" type="text/css"/>
</head>
<body>
<p class="cabecera_pagina"> Febrero, 2002 </p>
<p class="rhead">
  Antonio Marín <br />
  Macrosoft Corporation <br />
  Valencia, DP. 12345
</p>
<p class="lhead">
  Departamento de Comercio Electrónico <br />
  Valencia,
</p>
<p> Estimado señor,</p>
```

Continúa ...

## 10. Hojas de estilo.

### Propiedades del texto (IV)

---

Continuación de la anterior dispositiva...

```
<p class="body">
```

```
    Le escribo para informarle que, debido a las dificultades  
    financieras que atraviesa nuestra empresa, nos vemos  
    obligados a revocar el contrato establecido con su compañía.
```

```
</p>
```

```
<p class="body">
```

```
    Como puede imaginarse, las expectativas de beneficios que nos  
    habíamos marcado al inicio del año no se han cumplido.
```

```
    Dada el momento difícil que estamos atravesando, estamos  
    seguros de que podremos contar con su apoyo y comprensión.
```

```
</p>
```

```
<p class="foot">
```

```
    Sinceramente suyo, <br />
```

```
    Antonio Marín.
```

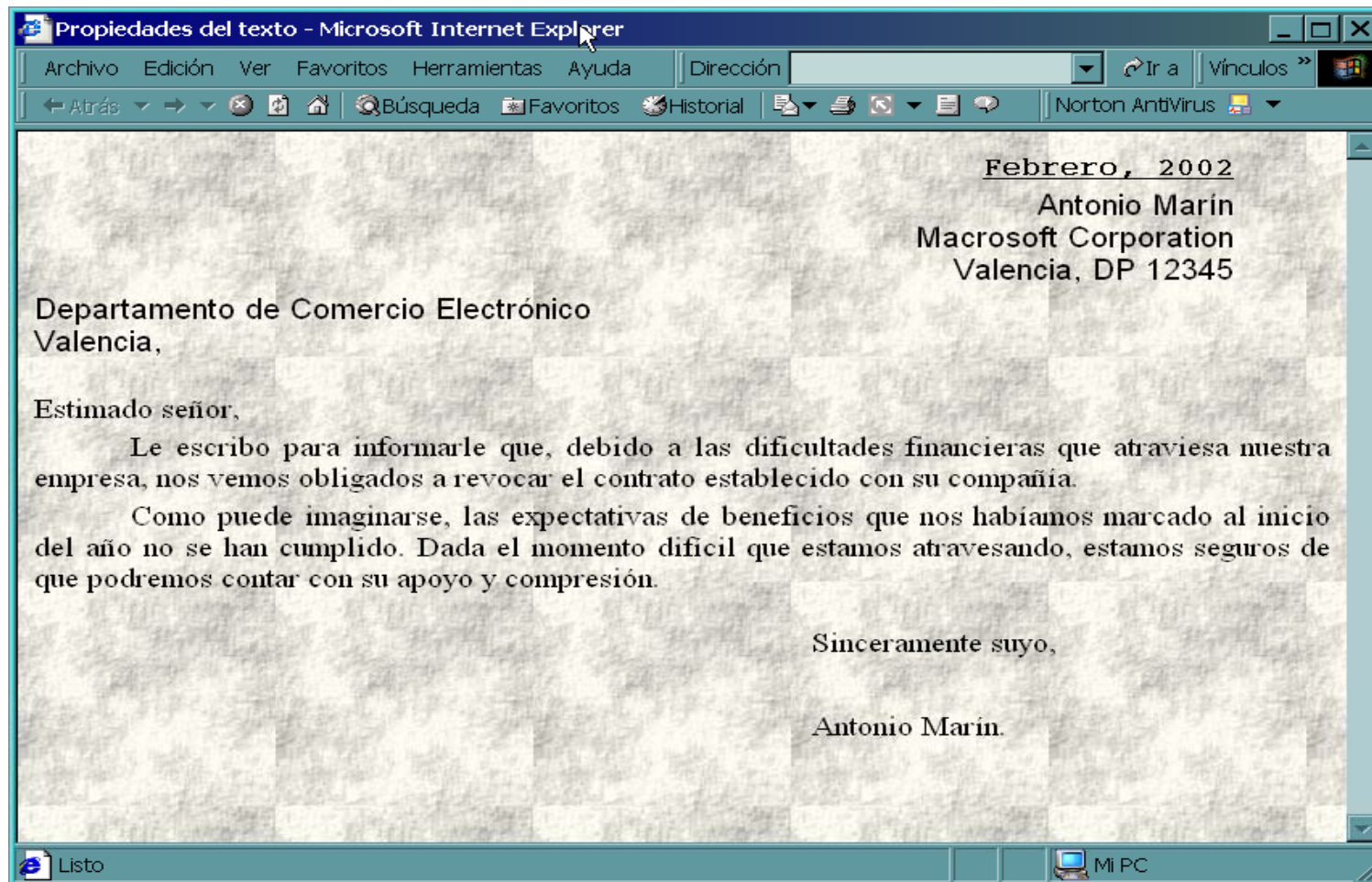
```
</p>
```

```
</body>
```

```
</html>
```

## 10. Hojas de estilo.

# Propiedades del texto (V)



## 10. Hojas de estilo.

### Prop. colores y fondos (I)

---

- **color**: color del texto.
  - nombre | #RRGGBB | rgb( rrr, ggg, bbb )  
`p {color: blue}, h1 {color: #00AABB},  
h3{color:rgb(255,0,0)}`
- **background-color**: color del fondo de la región.
  - nombre | #RRGGBB | rgb( rrr, ggg, bbb )
- **background-image**: imagen de fondo de la región.
  - none | url(nombre\_fichero)  
`h2 { background-image: url(Bluedrop.gif); }`
- **background-repeat**: modo de repetición de la imagen.
  - repeat | repeat-x | repeat-y | no-repeat  
`body { background-image: url(fondo.jpg);  
background-repeat: repeat-x }`
- **background**: combina las propiedades en una entrada.  
`p { background: url(fondo.jpg) repeat-x }`

## 10. Hojas de estilo.

### Prop. colores y fondos (II)

---

Ejemplo:

```
//----- fondos.css -----  
.banner { background: url(images/boards.jpg) repeat-x;  
          font-size: 50pt;  
          font-family: "Arial Rounded MT Bold" }  
.fondoazul { color: white; background-color: "#0000ff" }  
.fondoconfeti { background-image: url( confetti.jpg);  
               background-repeat: repeat; }
```

## 10. Hojas de estilo.

### Prop. colores y fondos (III)

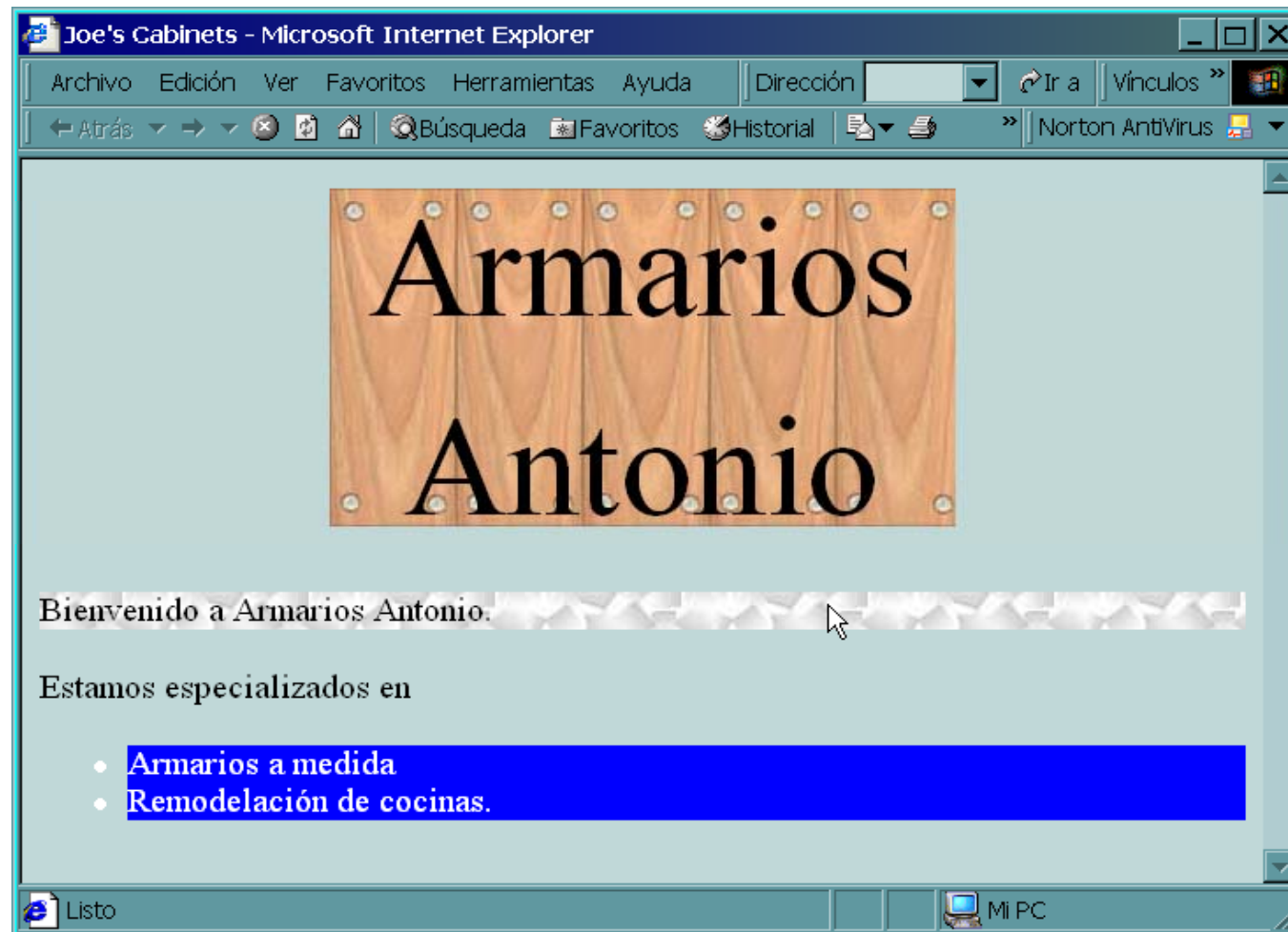
---

```
....
<head>
  <title>Ejemplo de fondos</title>
  <link rel="stylesheet" href="fondos.css" type="text/css"/>
</head>
<body>
<table align="center" width="360" height="199">
  <tr><td align="center" class="banner">Armarios <br />
    Antonio
  </td></tr>
</table>
<p class="fondoconfeti"> Bienvenido a Armarios Antonio. </p>
<p>Estamos especializados en </p>
<ul>
  <li class="fondoazul">Armarios a medida </li>
  <li class="fondoazul">Remodelación de cocinas. </li>
</ul>
</body>
</html>
```

## 10. Hojas de estilo.

### Prop. colores y fondos (IV)

---

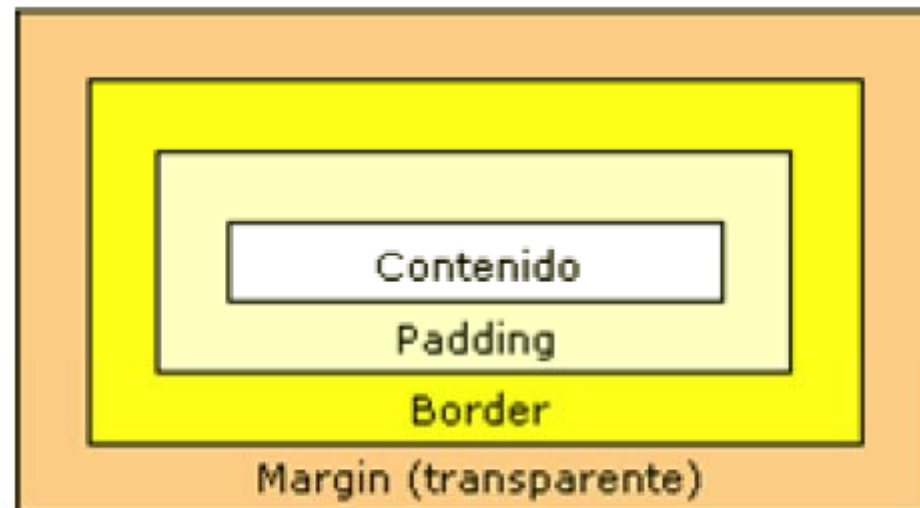


## 10. Hojas de estilo.

### Propiedades de la caja (I)

---

- CSS asume que todos los elementos están delimitados por una o más regiones rectangulares (caja).
- Los estilos pueden especificar márgenes, bordes y padding de la caja





## 10. Hojas de estilo.

## Propiedades de la caja (II)

- `margin-left`, `margin-right`, `margin-top`, `margin-bottom`.

auto | pt, in, cm, mm | em, ex, px, %

- **margin:** aúna todos los márgenes en una entrada.

P.e.: `p.margin {margin: 2cm 4cm 3cm 4cm}`

(arriba,derecha,abajo,izquierda)

- padding-left, padding-right, padding-top, padding-bottom

pt, in, cm, mm | em, ex, px, %

- **padding**: aúna las anteriores propiedades en una entrada.

P.e.: p.padding {padding: 1.5cm}

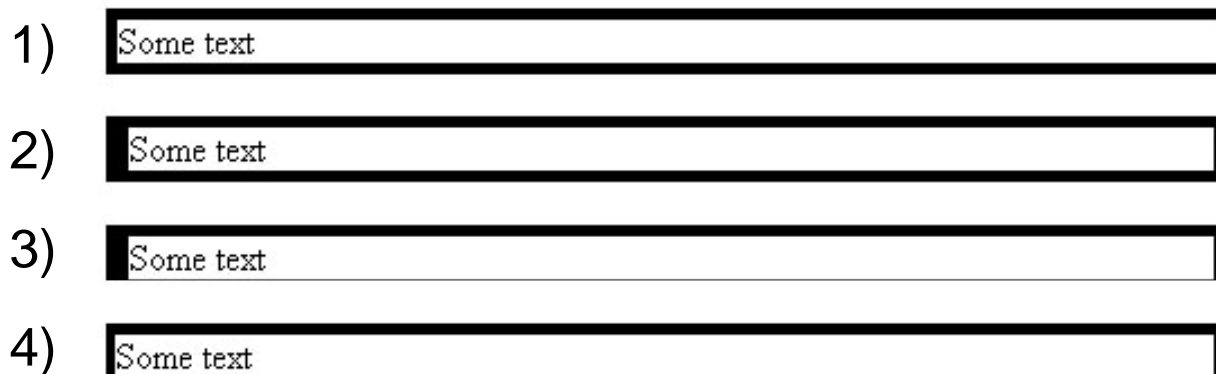
p.padding2 {padding: 0.5cm 2.5cm}

## 10. Hojas de estilo.

### Propiedades de la caja (III)

- **border-left-width, border-right-width, border-top-width, border-bottom-width**  
**none | thin | medium | thick | pt, in, cm, mm | em, ex, px**
- **border-width:** Fija el tamaño de los bordes en una entrada (arriba/derecha/abajo/izquierda)

P.e.:  
1) **border-width: 5px** (Todos los bordes iguales)  
2) **border-width: 5px 10px** (arriba/abajo, derecha/izq)  
3) **border-width: 5px 10px 1px** (arriba, der/izq, abajo)  
4) **border-width: 5px 10px 1px medium** (arr,der,ab,izq)



## 10. Hojas de estilo.

### Propiedades de la caja (IV)

---

- **border-color**: Fija el color del borde  
nombre-color | #RRGGBB | rgb(rrr, ggg, bbb)
- **border-style**: Estilo del borde.  
**none** | dotted | dashed | solid | double | groove | ridge | inset | outset
- **border-left, border-right, border-top, border-bottom**: Ancho, estilo y color en una entrada.
- **border**: Ancho, estilo y color de los 4 márgenes en una entrada.  
P.e.:            p { border: medium double rgb(250,0,255) }

## 10. Hojas de estilo.

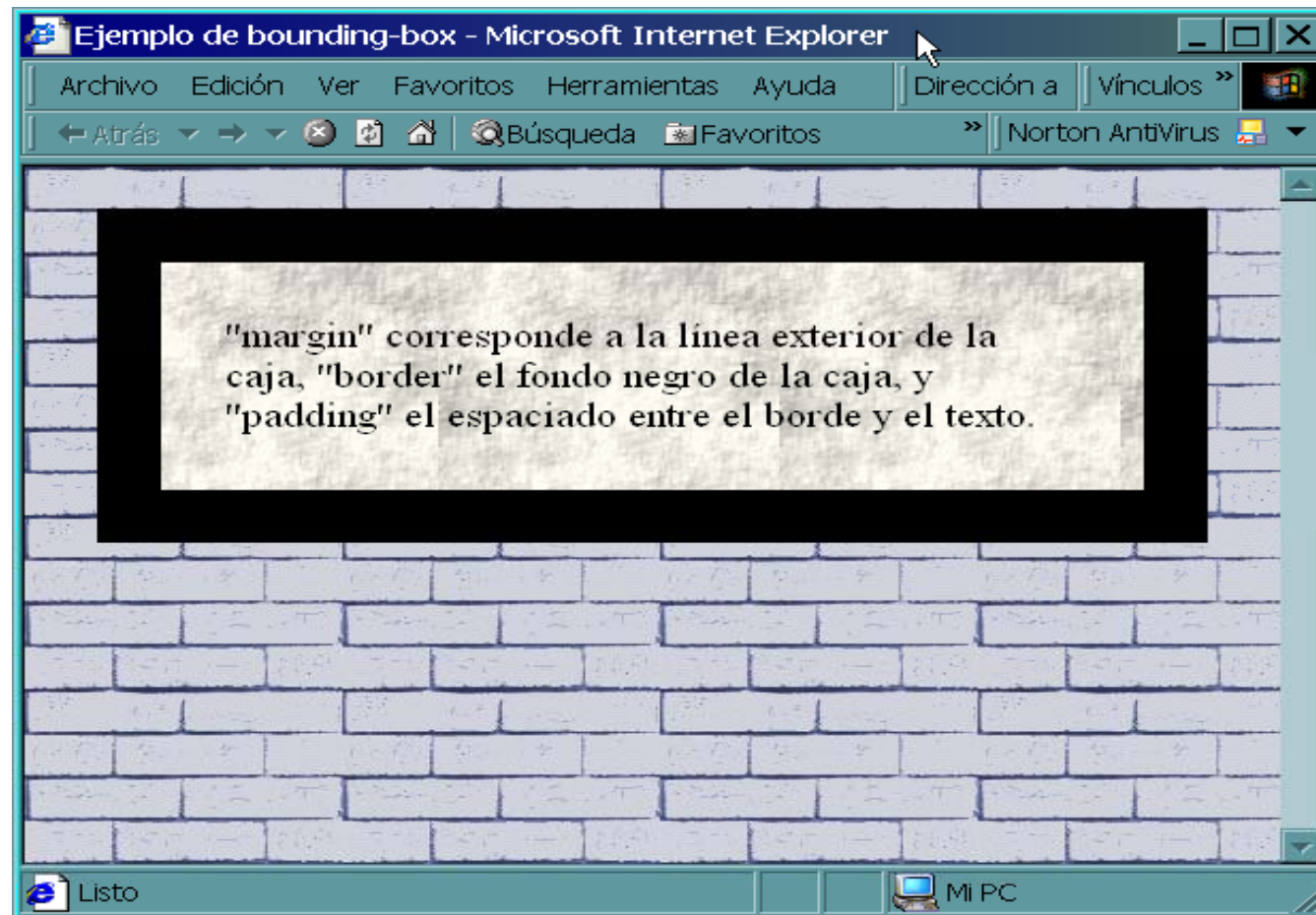
### Propiedades de la caja (V)

---

```
...
<head>
  <title> Ejemplo de bounding-box</title>
  <style type="text/css">
    <!--
    p { margin: 0.50 cm; background: url(images/bond-paper.jpg);
        border: 0.25in solid black; padding: 0.25in }
    body { background: url(images/bricks.jpg) }
    -->
  </style>
</head>
<body>
  <p> "margin" corresponde a la línea exterior de la caja,
      "border" el fondo negro de la caja, y "padding" el
      espaciado
      entre el borde y el texto.
  </p>
</body>
</html>
```

## 10. Hojas de estilo.

### Propiedades de la caja (VI)



## 10. Hojas de estilo.

### Imágenes y elementos flotantes (I)

---

- **width, height**: tamaño del elemento (normalmente una imagen).
  - **auto** | pt, in, cm, mm | em, ex, px

```
img.bala { width: 100px; height: 50px }
```

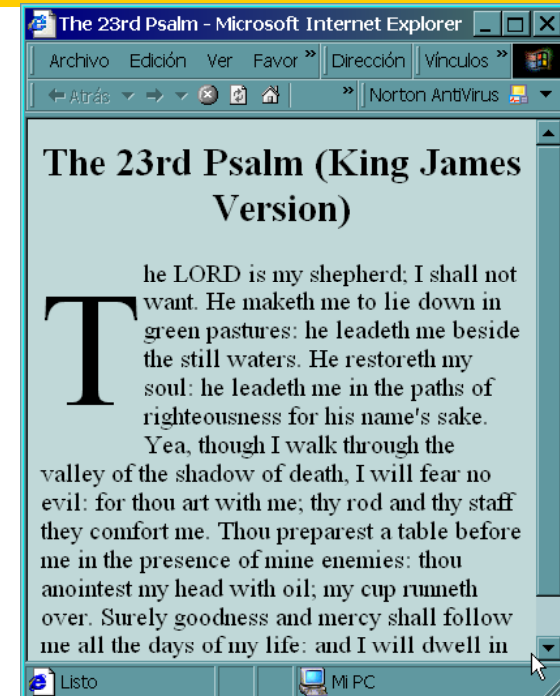
- **float**: indica como se sitúa el elemento flotante respecto al texto circundante.
  - **none** | left | right

```
img { float: left }
```

## 10. Hojas de estilo.

# Imágenes y elementos flotantes (II)

```
....  
<head>  
  <title>The 23rd Psalm</title>  
  <style type="text/css">  
    <!--  
    span { float: left;  
          font-size: 75pt }  
    -->  
  </style>  
</head>  
<body>  
  <h2 align="center">The 23rd Psalm (King James Version)</h2>  
  <span>T</span>he LORD is my shepherd; I shall not want.  
  ....  
</body>  
</html>
```



## 10. Hojas de estilo.

### Consejos de utilización

---

- Separar apariencia y contenido.
  - Evitar dependencias del documento con su apariencia.
  - Cuidadoso uso de style, separación en ficheros, uso de colores o longitudes absolutas.
- Cuidado con los navegadores:
  - Soporte a CSS no completo.
  - Probar en diferentes navegadores.
  - Otra opción: lenguajes script, JavaScript, VBScript
- Agrupar estilos:
  - Por elementos (contenedores) o por propiedades (colores, texto,...)
  - Facilita mantenimiento y modificación.