
Tema 2. I- Hojas de estilo CSS.

Programación Multimedia.

G.I.M.

Inmaculada Coma, Francisco Grimaldo



Resumen

- En este tema:
 - Recordaremos cómo crear estilos CSS e insertarlos en una página HTML.
 - Recordaremos los principales selectores CSS para dar formato a los contenidos HTML.
 - Veremos algunas opciones avanzadas de CSS e iremos avanzando algunas opciones de CSS3.

Hojas de estilo.CSS.

¿Qué es una hoja de estilo?

- Las hojas de estilo en cascada o CSS (*Cascading Style Sheets*) son un lenguaje utilizado para la presentación de un documento estructurado escrito en HTML, XHTML o XML.
- La idea que subyace en las hojas de estilo es separar la *estructura* de un documento de su *presentación*.
- Por ejemplo, podemos utilizar la etiqueta <H1> para definir una 'cabecera' de página, sin darle ninguna aspecto a la misma. Luego, en CSS definiremos el aspecto de dicha etiqueta (color, tipo de fuente, tamaño, o incluso volumen de un sintetizador de voz).
- Esto no permite compartir fácilmente los estilos entre múltiples documentos de un mismo sitio Web.
- Las reglas de estilo se aplican de forma jerárquica.

Hojas de estilo.CSS.

Sintaxis:

- Forma general:

```
selector { propiedad: valor }
```

O

```
selector { propiedad1: valor1;  
          propiedad2: valor2;  
          .....  
          propiedadN: valorN }
```

- Ejemplo:

```
H1 { text-align: center;  
     color: blue }
```

Inserción de estilos CSS en HTML

¿Cómo aplicar una hoja de estilo?

- Existen tres formas de aplicar estilos a un documento:
 - Una **hoja de estilo externa**. Almacenaremos en un fichero .css las reglas de estilo del documento. Es la forma más adecuada, separando el estilo completamente del documento.
 - Una **hoja de estilo interna**. Incrustamos dentro de la cabecera del documento HTML los estilos que vamos a aplicar.
 - Un **estilo en línea**. Insertamos un estilo directamente dentro de una etiqueta HTML.

Inserción de estilos CSS en HTML

Estilo importado de hoja externa.

- En la cabecera del fichero HTML 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 propiedades alteradas:

```
/* 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 }
```

Inserción de estilos CSS en HTML

Hoja de estilo interna

- En la cabecera del fichero HTML dentro de las etiquetas HEAD, se definen las propiedades de la hoja de estilo.

```
<head>
<title>Actes commemoratius del Jardí Botànic</title>
<style>
BODY { background-color:#00FF99;}
H1 { text-align: center;
     color: red;
     font-family: Dauphin, "Book Antiqua", Arial}
H2 { font-family: "Square 721 BT", Tahoma, Arial}
STRONG { text-decoration: underline }
</style>
</head>
```

Inserción de estilos CSS en HTML

Estilo definido en línea

- Se utiliza el atributo **STYLE** para cambiar directamente el estilo de cada elemento HTML.

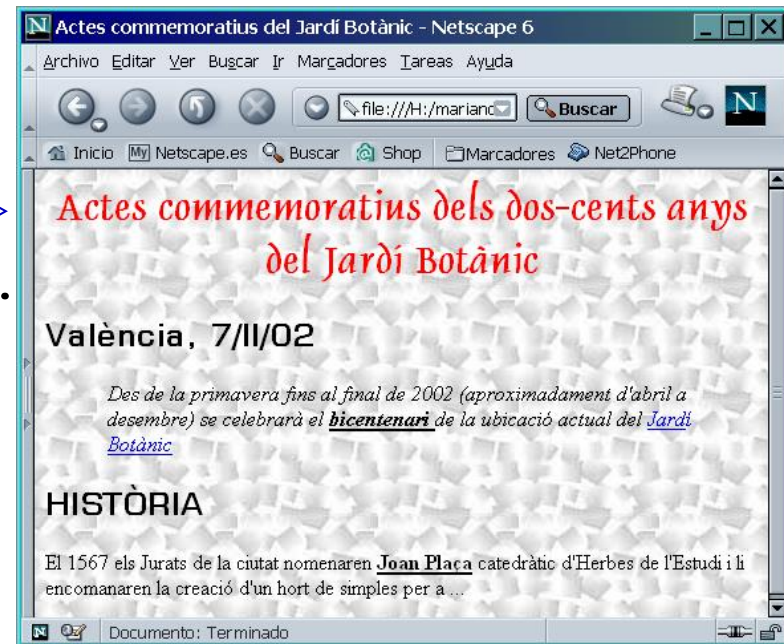
p.e.

```
<H2>València, 7/II/02</H2>
```

```
<p style="margin-left: 1cm;  
        magin-right: 1cm;  
        font-style: italic">
```

Des de la primavera fins al

.....



Selectores

- **Selectores formados por elementos HTML:**

```
BODY { color: blue }
H1, H2, H3, H4, H5, H6 { color: red;
                        font-family: sans-serif}
H1 EM { color: green }
```

- **Clases definidas por el usuario.**

```
//Clase que se aplica sobre 1 elemento HTML
P.miparrafo { margin-left: 1cm; magin-right: 1cm;
              font-style: italic }
<P class="miparrafo">
```

```
//Clase global
.azul { color: blue; font-weight: bold }
<H2 CLASS="azul">Cabecera azul</H2>
```

Selectores

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 }  
.bizarre :active { font-size: 300% }  
A:link IMG { border: solid green }
```

Selectores

- Además de los selectores vistos, CSS permite definir a qué etiquetas HTML afecta un estilo, de diferentes formas. Algunos selectores son nuevos en CSS3.
 - Por 'id' del elemento.
 - Por posición: etiquetas adyacentes, etiquetas anidadas.
 - Por eventos del usuario: click, mouseover..
 - Por atributos: si una etiqueta tiene determinados atributos.
- Selectores CSS3:
 - <http://www.w3.org/TR/css3-selectors/>
 - http://www.w3schools.com/cssref/css_selectors.asp

Selectores

- Selectores básicos: elemento HTML, clase o ID:

Tipo	Sintaxis CSS	HTML
Elemento	H1 {color:blue;}	<h1>
Nombre de clase	.azul {color:blue;}	<h1 class="azul">...
Identificador	#capa {display:block;}	<DIV id="capa">

Selectores

■ Por posición

Tipo	Sintaxis CSS
Anidados	tag1 tag2
Hijo	#capa > li {color:blue;}
Etiquetas adyacentes	h1 + p {color:blue;}
Posición en el árbol	li:first-child {color:blue;}

■ Selección de elementos dentro del árbol:

:first-child

:last-child

:nth-child (even | odd| {formula})

:after

:before

:first-letter :first-line

Selectores

■ Por atributos

Tipo	Sintaxis CSS
Atributo	tag[atributo]
Atributo con valor	tag[atributo="valor"]
Clase	tag.clase
ID	tag#id

■ Eventos ratón

Tipo	Sintaxis CSS
Hiperenlaces visitados o no visitados	tag:link tag:visited
Elemento al que se está haciendo click (active), ratón sobre el elemento, ratón que tiene el foco	tag:active tag:hover tag:focus

Selectores. Ejemplo.

Estilo:



```
div {
    width: 20 px
}
#parrafo{
    color: red;
}
.capa{
    width: 15 px;;
}
.capa .capa2{
    width: 30 px;
}
```

HTML

```
<p id="parrafo">
    Mi texto
</p>

<div class ="capa">
    <div class="capa2">
        Hola mundo
    </div>
</div>
```

Prioridades

- Las etiquetas se aplican en “cascada”, estableciéndose un orden de prioridades.
- Dependiendo de dónde se ha definido la etiqueta las prioridades son:
 1. Estilo inline (dentro de un elemento HTML). 
 2. Hoja de estilo interna (en la cabecera). 
 3. Hoja de estilo externa.
 4. Estilos por defecto del navegador.

Prioridades

- Cuando a un mismo elemento se le pueden aplicar varios estilos se calcula la especificidad de cada selector y se aplica el que tiene mayor valor.
- Para calcular la especificidad se hace:
 - a = contar 1 si la declaración viene de un atributo “**style**” inline.
 - b = contar el número de atributos **ID** en el selector.
 - c = contar el número de **pseudo-clases** y otros atributos en el selector.
 - d = contar el número de **elementos** en el selector.
- El resultado será un número “abcd” que definirá la prioridad.

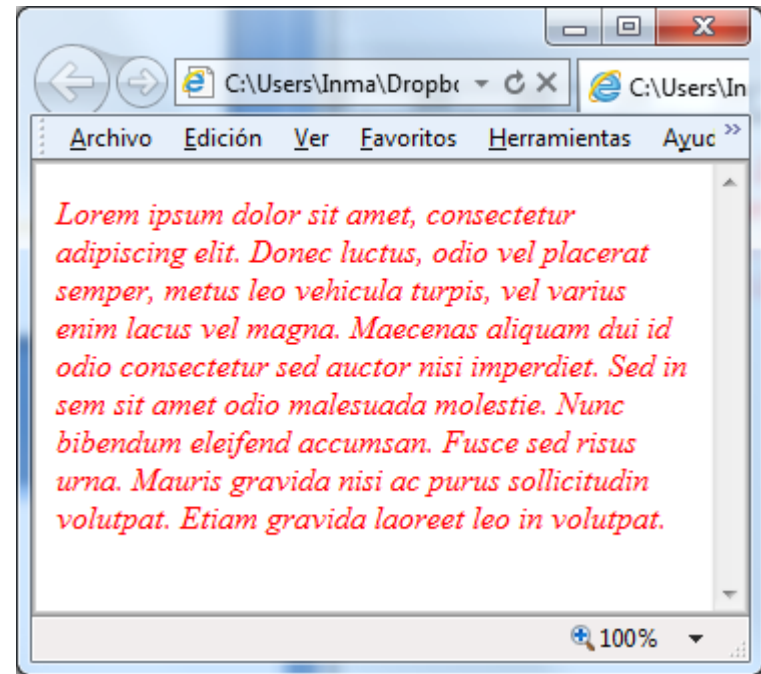
```
li.red.level { } /* a=0 b=0 c=2 d=1 -> specificity = 0,0,2,1 */  
ul ol+li { } /* a=0 b=0 c=0 d=3 -> specificity = 0,0,0,3 */
```

Prioridades

```
<HTML>
<HEAD>
  <STYLE type="text/css">
    #x97z
    {
      color: black;
      font-style: italic;
    }
  </STYLE>
</HEAD>
<BODY>
  <P ID="x97z" style="color:red"> ... </p>
</BODY>
</HTML>
```

```
#x34y {} /* a=0 b=1 c=0 d=0 -> specificity = 0,1,0,0 */
style="" /* a=1 b=0 c=0 d=0 -> specificity = 1,0,0,0 */
```

Como 1000 > 0100 -> se aplica el color rojo definido en style



Propiedades de las fuentes

- **font-weight:** Anchura relativa de la fuente.

- **nomal** | 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 }
```

Propiedades del texto

- **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, %

Colores y fondos.

- **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 | norepeat

```
BODY { background-image: url(fondo.jpg);  
background-repeat: repeat-x }
```

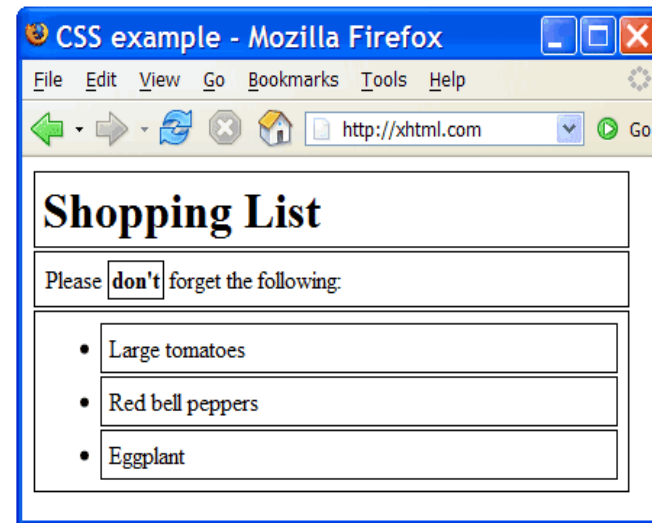
- **background:** combina las propiedades en una sólo entrada.

```
P { background: url(fondo.jpg) repeat-x }
```

Box model

- Cada uno de los elementos HTML que incluimos en nuestra página está delimitados por una región rectangular.

```
<!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="en">
  <head>
    <title>CSS Example</title>
  </head>
  <body>
    <h1>Shopping List</h1>
    <p>Please <strong>don't</strong> forget the following: </p>
    <ul>
      <li>Large tomatoes</li>
      <li>Red bell peppers</li>
      <li>Eggplant</li>
    </ul>
  </body>
</html>
```



Código HTML y áreas rectangulares asociadas a cada elemento

Box model

- Cada área rectangular está definida por 4 partes:
 - El contenido: *content*.
 - El borde: *border*. Este borde puede tener un determinado grosor.
 - El espacio entre el contenido y el borde: *padding*.
 - El espacio entre el borde y el exterior del rectángulo: *margin*.



Box model

- **border-width:** fija el tamaño de los bordes en una entrada
none | thin | medium | thick | pt, in, cm, mm | em, ex, px
- **border-left-width, border-right-width, border-top-width, border-bottom-width:** tamaño de los bordes de cada una de las líneas
- **border-color, border-left-color...:** fija el color del borde
nombre-color | #RRGGBB | rgb(rrr, ggg, bbb)
- **border-style, border-left-style...:** Estilo del borde.
none | dotted | dashed | solid | double | groove | ridge | inset | outset
- **border, border-left, border-right...:** fija el ancho, el estilo y de los márgenes.

Box model

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

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

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

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

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

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

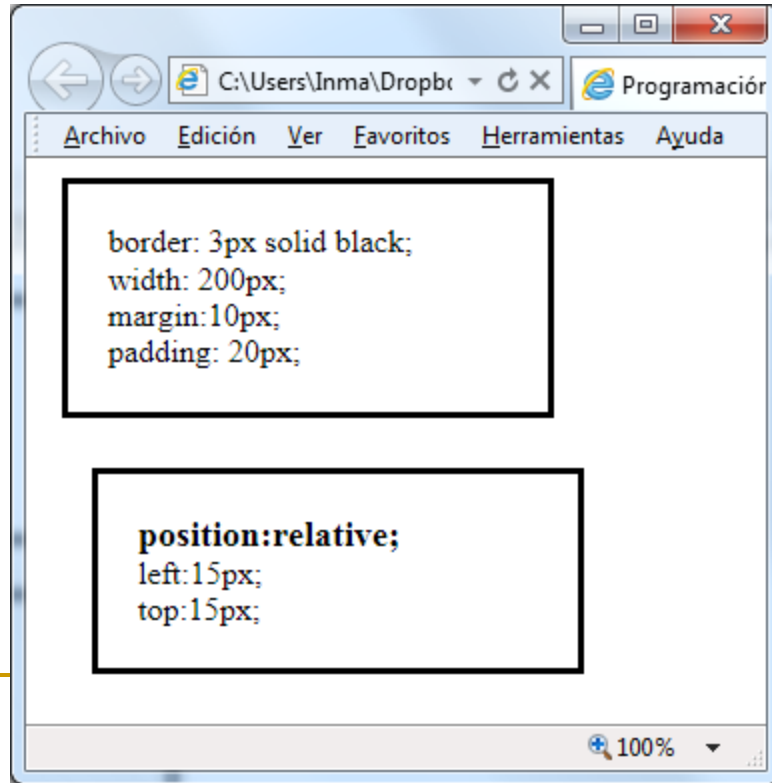
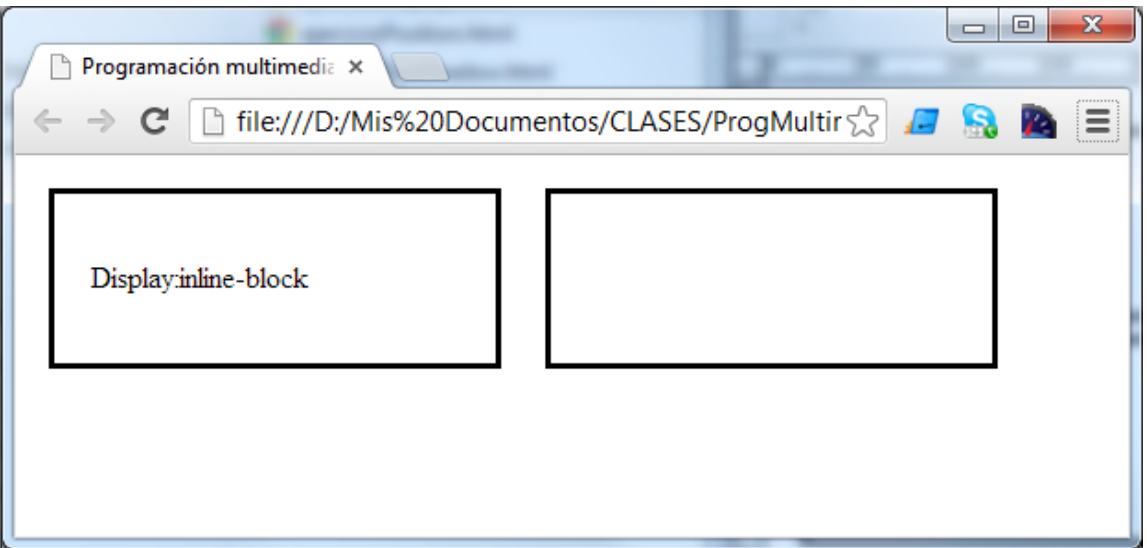
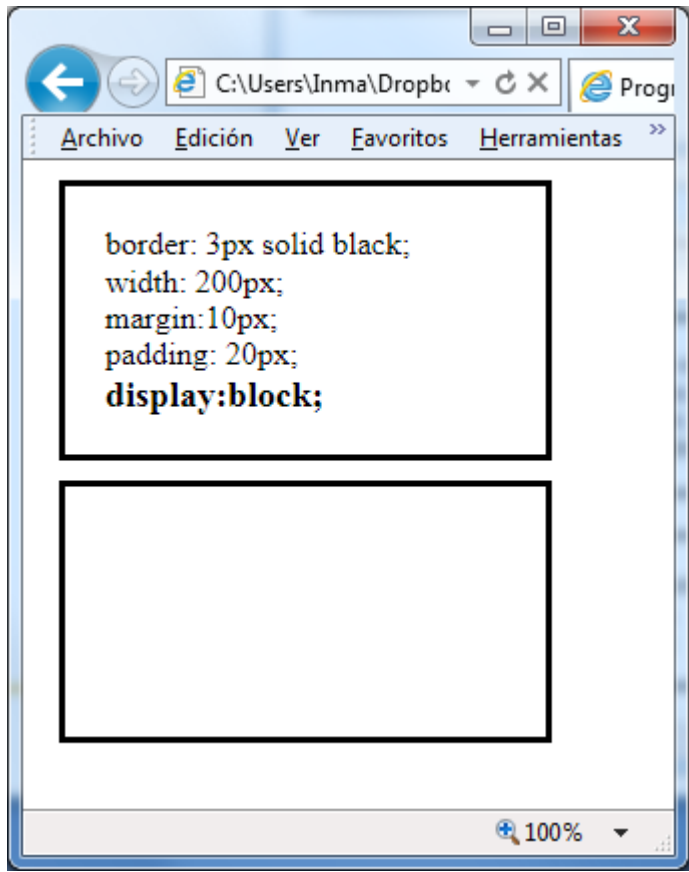
Dimensión, posición y visibilidad.

Para definir la dimensión y posición de imágenes o bloques creados con las etiquetas DIV y SPAN tenemos algunos selectores:

- **width, height:** tamaño del elemento
 - **auto** | pt, in, cm, mm | em, ex, px
- **position:** indica cómo se va a especificar la posición de un elemento: absoluta, relativa a elementos del entorno, fija.
 - **static** | fixed | relative | absolute
- **top, bottom, left, right:** píxeles de posición del objeto.
- **z-index:** orden de aparición de los elementos si se superponen (cuál se muestra encima).
- **visibility:** si es visible o no.
 - visible | hidden

Dimensión, posición y visibilidad.

- **display**: especifica cómo se agrupan las cajas de cada bloque en la página.
 - block | inline | inline-block | none
- **float**: indica como se sitúa el elemento flotante respecto al texto circundante.
 - none | left | right `IMG { float: left }`
- **clear**: indica en qué lados de un elemento no están permitidos los elementos flotantes
 - left | right | both
- **overflow**: especifica el comportamiento de un contenido si sobrepasa la caja que lo contiene
 - auto | hidden | scroll | **visible** | inherit
- **clip**: recorta un elemento



Bibliografía y referencias

- Cascading Style Sheets home page
 - <http://www.w3.org/Style/CSS/>
- Selectores CSS:
 - [http://www.w3schools.com/cssref/css_selectors.a
sp](http://www.w3schools.com/cssref/css_selectors.asp)