

Hojas de Estilos - CSS

Desarrollo de Aplicaciones Web
Departamento Informática y Sistemas
Universidad de Murcia
Curso 2014/15

Introducción

- **Motivación:**
 - Separar el **formato** del contenido y estructura de una página web.
- **Hoja de estilo:**
 - Conjunto de propiedades de formato más reglas de definición y aplicación.
- A partir de la versión 4 de HTML.
- Diferentes **tipos** de hojas de estilo:
 - **Estándar: CSS**, Hojas de Estilos en Cascada.
 - Versión bien soportada por los navegadores: **CSS 2.1**
- Son más potentes y cómodas que el formato en HTML.

Regla de Estilo

- **Regla de estilo:** componente de una hoja de estilo.
- Una regla está formada por:
 - **Selector:** indica las etiquetas a las que se aplica.
 - **Declaración:** conjunto de pares propiedad/valor.
- Ej.: `p {color: white; background-color: black;}`
- **Importante:** un error de sintaxis hace que la regla sea ignorada.

Tipos de selectores

- Basado en el **elemento (etiqueta)**:
 - `p` {color: white;}
 - `p, h1` {color: white;}
- Basado en el **contexto del elemento**:
 - `p strong` {color: blue; }
- Basado en la **clase o identificador** del elemento:
 - Clase: `span.rojo` {color: red;}
 - Id: `div#introduccion` {font-size: 14pt;}
- Basado en una **pseudoclase (estado)** de un elemento:
 - `a:hover` {color: pink;}

Selección basada en el elemento

- Aplica un estilo a todas las declaraciones de uno o varios tipos de etiquetas.

- Ejemplo:
 - “Aplica color de texto verde a todos lo párrafos”.
 - `p {color: green;}`

- Se pueden indicar varias etiquetas separadas por comas:
 - `p, h1 {color: green;}`

Selección por clase o ID

- Basado en la **clase** o **identificador** de **cualquier etiqueta**:
 - Clase: `span.rojo {color: red;}`
 - Identificador: `div#introduccion {font-size: 24pt;}`
- Se pueden aplicar a **toda una clase**, con independencia de la etiqueta:
 - `.rojo {color: red;}`
- Para identificadores, la etiqueta puede omitirse:
 - `#introduccion {font-size: 24pt;}`

Selección basada en el contexto

- Elemento que contiene a otro elemento, con independencia del nivel de profundidad
- Ejemplos:
 - “Establece el color de texto rojo a todas las declaraciones strong dentro de cualquier bloque div”
 - `div strong {color: red; }`
 - “Establece el color de texto rojo a todas las declaraciones strong dentro de cualquier párrafo del bloque div identificado como ‘introduccion’ ”
 - `div#introduccion p strong {color: red; }`

Selección según el estado (pseudoclase)

- Aplicable a **vínculos**: `<a>`
- Estados:
 - **link**: no visitado
 - **visited**: visitado
 - **focus**: cuando tiene el foco de control
 - **hover**: cuando el ratón está sobre el enlace
 - **active**: el momento en el que se pulsa
- Ej.:
 - `a:link {color:red;}`
- Un vínculo puede estar en **varios estados**
- **Para que no se anulen las propiedades entre sí, hay que declararlas en el orden anterior: link, visited, etc.**

Selectores - Otros

□ Pseudoelementos:

■ Primera línea:

□ `p:first-line {color:red;}`

■ Primera letra:

□ `p:first-letter {font-size:32pt;}`

□ Pueden combinarse varios tipos de selectores:

■ Contexto elemento (clase o ID) (pseudoclase o pseudoelemento)

■ `Ej: div#introduccion span.rojo {color: red;}`

■ `Ej: div#introduccion p:first-letter {color: green;}`

Declaración de estilos externa

- **Declaración externa:** en un fichero independiente con extensión **.css**
- **Mejor solución para compartir el estilo en todo un sitio web.**
- **Vincular la hoja de estilos** en la página:
 - En la **cabecera** del documento

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

- Se pueden enlazar varias hojas de estilo en una misma página web.

Declaración de estilos interna y local

- **Interna:** en la **cabecera** del documento

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

```
...
```

```
</style>
```

- **Local:** en cualquier etiqueta utilizando atributo **style**.

- Ej.: `<p style="color:red;"> ...</p>`

- NO es necesario especificar el selector. **Es implícito.**

- Tampoco se utilizan llaves { }

- Si hubiera que utilizar **comillas** en la declaración, deben ser sencillas: ' '. Ej. nombres de fuentes.

Tipos de Hojas de Estilo

- Hojas de estilo del **programador/diseñador**:
 - Las que definimos al editar una página web.
- Hoja de estilo del **navegador**:
 - Aplica estilos por defecto para aquellas propiedades para las que el programador no ha definido ningún formato.
- Hoja de estilos del **usuario**:
 - Definida por el usuario del navegador. Reemplaza a los estilos del programador y navegador.

Jerarquía de un documento HTML

- Las etiquetas dentro de una página web se organizan como un árbol.
- Concepto clave para aplicar propiedades y selectores de estilo en función del **contexto**.
- Algunas propiedades se heredan (!!).

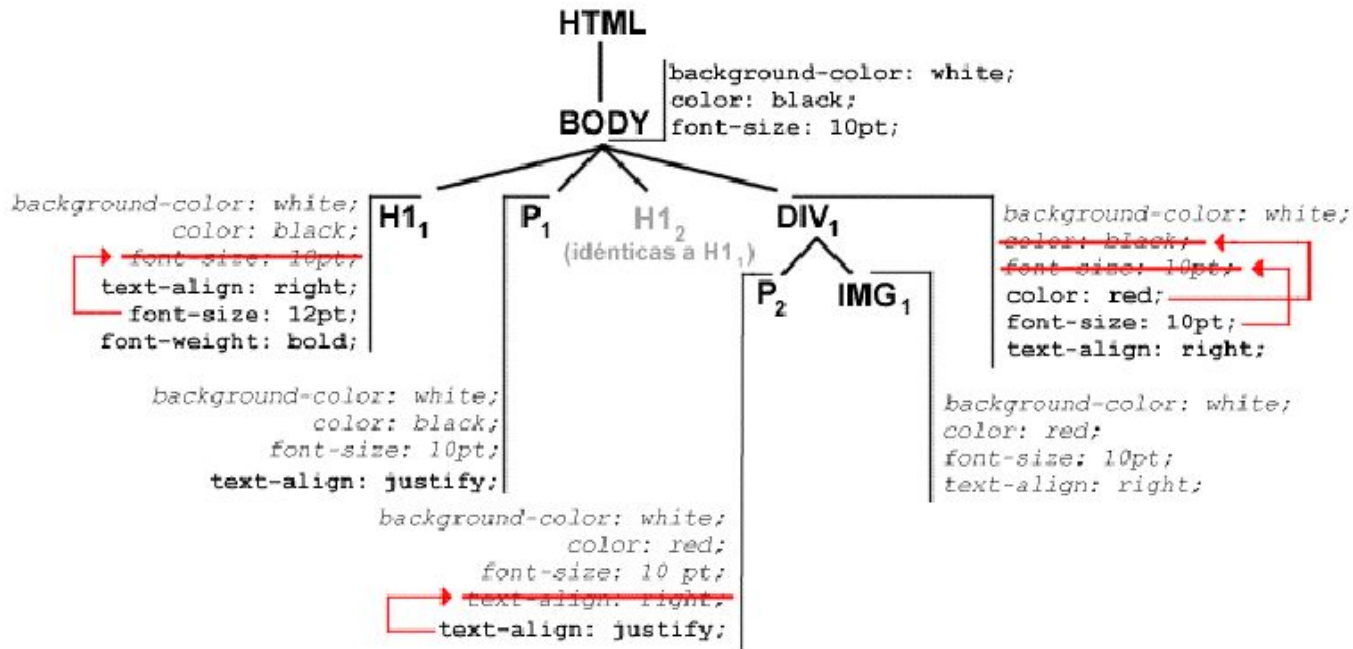
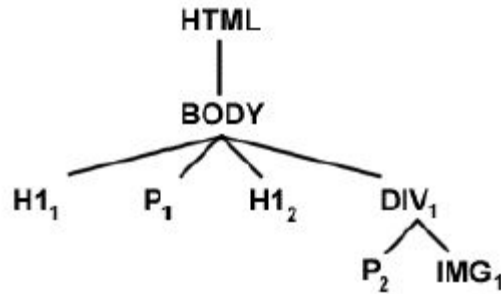
□ Ej.:

```
<div>
  <p> ...
    <strong> ... </strong>
  </p>
</div>
```

Herencia

- La herencia de propiedades de estilo permite que una propiedad llegue a la declaración de una etiqueta cuando no se ha definido explícitamente para ella.
- Esta característica se basa en el **principio de herencia**:
 - Una propiedad de estilo no definida para un elemento es heredada del elemento padre que lo contiene.
- La mayor parte de las propiedades de estilo se heredan, pero **no todas**.
- **Facilita que los ficheros con estilos sean más compactos.**
- **Ejemplo:** definir el color de la fuente en la etiqueta `<body>` hace que se herede en todo el documento.

Herencia



```

<STYLE TYPE=TEXT/CSS>
  BODY {
    background-color: white;
    color: black;
    font-size: 10pt;
  }

  H1 {
    text-align: right;
    font-size: 12pt;
    font-weight: bold;
  }

  P {text-align: justify;}

  DIV {
    color: red;
    font-size: 10pt;
    text-align: right;
  }
</STYLE>
  
```

Cascada

- A una declaración concreta de una etiqueta pueden llegar **estilos contradictorios**.
 - Ejemplo: color rojo y verde a la vez para un párrafo.
- El conflicto es resuelto en función de la **especificidad** de las reglas que generan el conflicto.
- **Orden de prioridad de las reglas:**
 - id > clase > elemento > heredadas

Cascada

- Ejemplo:
 - Sea una declaración `<div>` con id “introduccion”
 - Regla 1: `div { color: red; }`
 - Regla 2: `div#introduccion { color: blue; }`
 - El bloque `div` tendrá como color de texto el azul, ya que la regla 2 es más específica que la 1.

- Si persiste el conflicto, tiene prioridad:
 - La regla marcada como **!important**.
 - Ejemplo: `div#introduccion { color: black; !important}`

 - Si no hay reglas `!important`, **la última regla que se procesa: las reglas se procesan en el orden de declaración.**

Valores de propiedades

- **Medidas:** valor y unidad, **sin espacio**.
 - Ej. 14pt, 2cm, 12px, etc.
- **Medidas relativas:**
 - **em:** relativo al tamaño de la letra “M”. Ej.: 2em
 - Porcentajes. Ej. 100%
- **URL:** `url(fichero.jpg)`
 - URL relativas, **respecto a la hoja de estilo**.
- **Valores enumerados.**
 - Ej.: **border-style:** `none, dotted, dashed, ...`

Formato de fuentes

- **Tipo de fuente:** `font-family`
 - Se pueden indicar varias fuentes separadas por una coma y espacio.
 - Se aplica la primera disponible.
 - **Fuentes genéricas:** *monospace*, *serif*, *sans-serif*, *cursive*, *fantasy*
 - Ej.: `p {font-family: Verdana, Arial, sans-serif;}`

- **Fuente en cursiva:** `font-style`
 - `oblique` (oblicuo), `italic` (cursiva) y `normal`

Formato de fuentes

- Fuente en **negrita**: **font-weight**
 - **bold**: negrita intermedia
 - Valores relativos: **bolder**, **lighter**
 - Valor numérico (múltiplo de 100): entre 100 y 900
 - Valor **normal** para texto normal.

- **Tamaño de la fuente**: **font-size**
 - **Valor y unidad**: 12pt, 1.5em
 - **Relativo al padre**: 1.5em, 150%, larger, smaller.

Formato de texto

- **Interlineado:** `line-height`
 - Valor proporcional al tamaño de la fuente. Ej.: 1.5
 - Valor porcentual: 150% → equivale a 1.5

- **Todas las propiedades de texto juntas:** `font`
 - Orden de declaración: *font-style, font-weight, font-size/line-height, font-family*

 - Ej: `font: italic bold 16pt/150% Arial`

Color

- **Color del texto: `color`**
 - Valor enumerado: `red`, `blue`, `white`, ...
 - `#rrggbb` en hexadecimal, sin comillas
 - `rgb(r, g, b)`, del 0 al 255
 - `rgb(r%, g%, b%)`

- **Ejemplos:**
 - `p { color: red; }`
 - `p { color: #0033FF; }`
 - `p { color: rgb(233, 240, 35); }`

Fondo

- ❑ **Color de fondo: `background-color`**
 - Toma un valor de color o `transparent`.
- ❑ **Imagen de fondo: `background-image`**
 - Se indica la url de la imagen: Ej. `url(imagen.jpg)`
- ❑ **Repetición de la imagen de fondo: `background-repeat`**
 - `repeat, repeat-x, repeat-y, no-repeat`
- ❑ **Situación de la imagen cuando no se repite: `background-position`**
 - Ejemplos: `(bottom, left)` → abajo y a la izquierda, `(right, top)` → a la derecha y arriba, `(50%, 50%)` → al 50% en los dos ejes

Formato de párrafo

□ **Espaciado:**

- Entre **palabras**: `word-spacing`
- Entre **letras**: `letter-spacing`
- Expresado como valor absoluto o relativo.
- Ej.: `p { word-spacing: 0.5em; }`

□ **Sangría de primera línea: `text-indent`**

- Espacio en blanco al comienzo de la primera línea de un párrafo.
- Expresado como valor absoluto o relativo.
- *Sangría francesa*: valor negativo.
- Ej.: `p { text-indent: 1.5em; }`

Formato de párrafo

- **Espacio en blanco:** `white-space`
 - `pre`: se comporta como la etiqueta `<pre>`, sin modificar fuente
 - `nowrap`: evita que el navegador parta un fragmento de texto.

- **Alineación de texto:** `text-align`
 - `{left, right, center, justify}`

Formato especial de texto

- Efecto **versales**: `font-variant`
 - Muestra las minúsculas como mayúsculas pequeñas.
 - `{small-caps, normal}`

- **Decorar el texto**: `text-decoration`
 - `{underline, overline, line-through, none}`
 - Útil para quitarlo del formato predefinido. Ej. vínculos.

- **Cambio formato texto**: `text-transform`
 - `{capitalize, uppercase, lowercase, none}`

Listas

- **Viñetas y números:** `list-style-type`
 - Listas no ordenadas: `{disc, circle, square}`
 - Listas ordenadas: `{decimal, upper-alpha, lower-alpha, upper-roman, lower-roman}`
 - Quitar marcador: `none`.
- **Utilizar imágenes:** `list-style-image`
 - Imagen de la viñeta en listas no ordenadas: `url(imagen.jpg)`
 - Valor `none` para eliminarla
 - Se recomienda utilizar imágenes pequeñas.
 - Reemplaza la propiedad `list-style-type`
- **Posición del marcador:** `list-style-position`
 - `{inside, outside}`

Presentación con CSS

□ **Ventajas:**

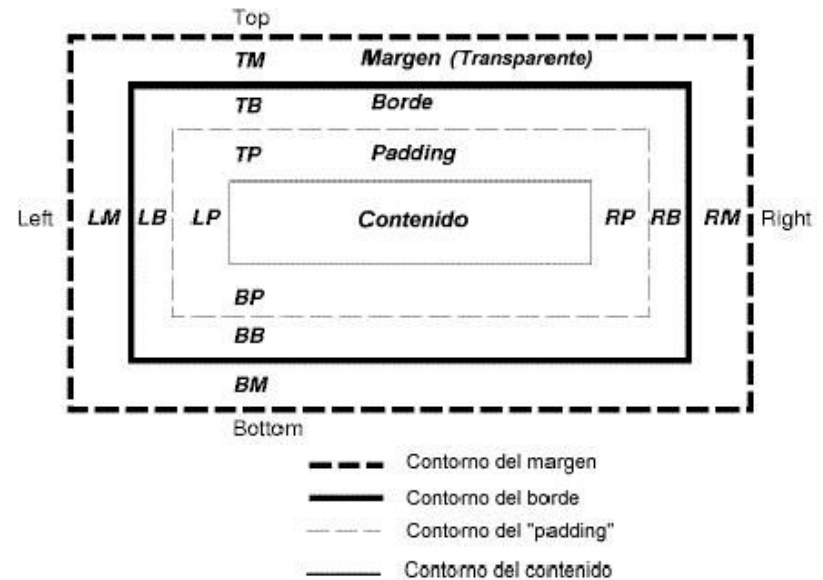
- Las páginas web son menos complejas, comparado con el uso de tablas.
- Aplicar la misma presentación a todo el sitio web.
- Los documentos HTML son más legibles y compactos.

□ **Inconvenientes:**

- Problema con navegadores antiguos.
- Navegadores actuales, a veces no interpretan correctamente las propiedades.

Modelo de Caja

- **TODOS** los elementos de una página web se encuentran en una caja invisible.
- **Caja:** área de contenido, relleno (*padding*), borde y margen.
- Con **propiedades de estilo** podemos controlar todas las propiedades de la caja.



Bordes

- **Tipo de borde:** `border-style`
 - `{none, dotted, dashed, solid, double, groove, ridge, inset, outset}`

- **Anchura:** `border-width`
 - Ej.: `p { border-width: 2px; }`
 - Valores predefinidos: `{thin, medium, thick}`

- **Color:** `border-color`
 - Para establecer el color del borde.
 - Por defecto, tiene el color del texto del elemento.

- **Aplicar propiedades a un solo lado:**
 - `border-top, border-left, border-right, border-bottom`
 - Ej.: `border-top-width: thin`

Relleno y márgenes

□ Relleno: **padding**

- Ej.: `p { padding: 2px; }`
- Aplicación a un solo lado: `padding-top`, etc.

□ Márgenes: **margin**

- Ej.: `p {margin: 10%; }`, `div {margin: 30px; }`
- Valor **auto**: para equilibrar los márgenes
 - **Útil para centrar**
- **Entre dos cajas, los márgenes verticales se superponen.**
Se usa el valor mayor (!!)
- Aplicación a un solo lado: `margin-top`, `margin-bottom`, ...

Anchura, altura y desbordamiento

- **Anchura y altura:** `width` y `height`
 - Ancho del área de **contenido**.
 - Valor de medida (px, pt, etc.) o relativo.
 - Ej.: `p { width: 50%; }, div { width: 800px; }`

- **Desbordamiento del contenido:** `overflow`
 - Sucede cuando el contenido supera la anchura y la altura establecida.
 - **visible:** para que siempre se vea el contenido
 - `hidden:` si no cabe, se oculta el contenido sobrante.
 - `scroll:` añadir siempre barras de desplazamiento.
 - `auto:` añadir barras de desplazamiento si es necesario.

Tipos de cajas y visibilidad

- Dos **tipos** cajas (tipos de etiquetas): de **bloque** y en **línea**.
- Cambiar el tipo de etiqueta: **display**
 - `{none, block, inline}`
 - Valor `none`: hace desaparecer la caja.
- **NO confundir con la propiedad `visibility`**
 - `hidden`: oculta la caja, pero **queda el espacio (!!)**.
- Ejemplo: convertir vínculos (elemento en línea) en bloques
 - `#navigation a {display: block}`

Flujo de presentación

- Por defecto, la visualización de una página sigue el orden de definición de las etiquetas (**estático**)

- **Modificaciones:**
 - Sacar una caja del orden natural y posicionarla respecto al elemento padre (**absoluto**) o al navegador (**fijo**).
 - Desplazarla respecto a la posición que debería ocupar en el orden natural (**relativo**)

- Los desplazamientos ocasionan solapamientos de cajas. En caso de solaparse, se puede especificar el orden de **solapamiento**.

Posicionamiento absoluto

- Propiedad: **position: absolute**
 - Saca la caja fuera del flujo natural de presentación.
 - Si no se indica la posición, aparece en su posición natural, pero con **solapamientos** con el resto de cajas que se visualizan sin tenerla en cuenta.
 - Para establecer la posición respecto a los bordes del padre:
 - `top`, `right`, `bottom`, `left`
 - **Importante:** para que funcione correctamente el padre debe posicionarse.

- Ejemplo:

```
#bg {  
    position: absolute;  
    top: 250px;  
    left: 2%;  
}
```

Posicionamiento fijo

- Propiedad **position:fixed**
 - Parecido al absoluto.
 - El posicionamiento es respecto a la ventana del navegador.
 - No está bien soportado por todos los navegadores.

- Ejemplo:

```
#navigation {  
    position:fixed;  
    top: 10px;  
    left: 2%;  
}
```

Posicionamiento relativo

- Propiedad **position: relative**
 - Desplazamiento respecto a su posición natural.
 - El resto de elementos no se ve afectados.
 - También puede generar solapamientos.

- Ej.: sacar los encabezados a la izquierda

```
h2, h3 {  
    position: relative;  
    left: -25px  
}
```

Posicionamiento 3D

- Propiedad: `z-index`
 - Resuelve los problemas de **solapamiento** estableciendo un orden en las cajas.
 - Toma un valor entero. Mayor valor, más arriba.
 - Si no establecemos esta propiedad, las cajas se apilan en el orden de declaración.
- Ejemplo:

```
#bg { z-index: 1}  
#content {z-index: 2}  
#navigation {z-index: 3}
```

Elementos flotantes

- Propiedad: **float**
 - Comportamiento **similar al atributo *align* de imágenes y tablas.**
 - **Valores:** `{left, right}`
 - Se aplica para aquellas cajas cuyo ancho es menor que el de la caja padre que la contiene.
 - El resto de cajas flotan alrededor.

- Para evitar que floten: **clear**
 - **Similar al atributo *clear* de *br*.**
 - **Valores:** `{left, right, both, none}`

Alineación vertical

- Propiedad: `vertical-align`
 - Las cajas pueden alinearse respecto a la línea en la que se encuentran.
 - `middle`: centrado en la línea
 - `sub`: como subíndice
 - `super`: como superíndice
 - `top`: alineado con la parte más alta de la línea **de la línea.**
 - ...

Tablas con propiedades de estilo

- **Alineación de las celdas:**
 - Horizontal: `text-align`
 - Vertical: `vertical-align`

- **Espaciado:**
 - La propiedad `padding` sustituye al atributo `cellpadding`
 - La propiedad `border-spacing` equivale a `cellspacing`

- **Visualización:** `table-layout: fixed.`
 - Ancho de celdas determinado por la primera fila.

- Visualizar **celdas vacías:** `empty-cells {show, hide}`