

CSS: Hojas de Estilo en Cascada

B. Cristina Pelayo García-Bustelo

Creación de Sitios Web con XHTML, CSS y JavaScript
Agaete 2007

- 1 Anatomía de estilos
- 2 Herencia y Cascada
- 3 Propiedades de Aspecto
- 4 Modelo de caja
- 5 Modelo de Formato Visual

Definición

- W3C define CSS como “*mecanismo simple para añadir estilo a documentos Web*”
- Es un lenguaje de diseño estándar para la Web
- No malgasta ancho de banda
- Combinado con XHTML estructural permite separar presentación y estructura facilitando la accesibilidad y mantenimiento

Definición

- W3C define CSS como “*mecanismo simple para añadir estilo a documentos Web*”
- Es un lenguaje de diseño estándar para la Web
- No malgasta ancho de banda
- Combinado con XHTML estructural permite separar presentación y estructura facilitando la accesibilidad y mantenimiento

Definición

- W3C define CSS como “*mecanismo simple para añadir estilo a documentos Web*”
- Es un lenguaje de diseño estándar para la Web
- No malgasta ancho de banda
- Combinado con XHTML estructural permite separar presentación y estructura facilitando la accesibilidad y mantenimiento

Definición

- W3C define CSS como “*mecanismo simple para añadir estilo a documentos Web*”
- Es un lenguaje de diseño estándar para la Web
- No malgasta ancho de banda
- Combinado con XHTML estructural permite separar presentación y estructura facilitando la accesibilidad y mantenimiento

Las reglas generales

Las reglas generales de las declaraciones de estilos son:

- Un selector que va a determinar a qué elemento y en que condiciones se va a aplicar el estilo.
- La declaración de propiedades que serán aplicadas al elemento, entre paréntesis.
- Las propiedades se encuentran identificadas por su nombre seguidas de dos puntos (:) y el valor que se establece para esa propiedad.
- Si se aplican varias propiedades a un mismo selector, se separan por punto y coma (;)

Sintaxis

Regla

```
Selector{declaración; }
```

Declaración

```
Propiedad : Valor ;
```

Ejemplo

```
body {color: #333300;}
```


Declaraciones múltiples

Las reglas se pueden agrupar

```
Selector{  
  declaración1;  
  declaración2;  
  ...  
  declaraciónN;  
}
```

Ejemplo

```
body {  
  color: #333300;  
  background-color: #CCCCFF;  
}
```

Importante

- Las declaraciones deben terminar en punto y coma (;)
- CSS no distingue entre mayúsculas y minúsculas (salvo en los identificadores asociados a XHTML)

Tipos de Selectores

- Selector de tipos
- Selector universal
- Selector de clase
- Selector id
- Selectores de atributos
- Selectores descendentes
- Selectores secundarios
- Selectores combinados

Selector de tipos

Permite seleccionar todos los elementos de un determinado tipo y dar valor a las propiedades

Ejemplo: da color a todos los elemento h1

```
h1 {  
  color: #36C;  
}
```

Selector de varios elementos

Podemos aplicar el mismo estilo a varios elementos diferentes, separando los selectores con comas:

Ejemplo: color para h1, h2 y h3

```
h1 {  
  color: #36C;  
}  
h2 {  
  color: #36C;  
}  
h3 {  
  color: #36C;  
}
```

Ejemplo: mismo color a h1, h2 y h3

```
h1, h2, h3 {  
  color: #36C;  
}
```

Selector Universal

Para aplicar un estilo a todos los elementos se utiliza el selector universal, representado mediante un asterisco (*) seguido de la definición de una o varias propiedades.

Ejemplo: selecciona todos los elementos y les da color

```
* {  
background-color: yellow;  
}
```

Selector de Clase

En XHTML tenemos elementos con atributo **class**. Se puede utilizar como selector en CSS. Para ello hay que poner el nombre precedido de un punto (.) y del selector universal (puede ser omitido).

Ejemplo: selecciona todos los elementos **énfasis** y pone en color rojo

```
*.énfasis {  
color: red;  
}
```

Importante

Los nombres de las clases no deben coincidir ni con elementos XHTML ni con propiedades CSS para evitar confusiones

Aplicar varias clases a un mismo elemento

En XHTML se pueden aplicar varias clases a un mismo elemento para modificar sus propiedades CSS. Los nombres de las clases deben separarse por espacios en blanco.

Ejemplo

```
<div class="clase1 clase2"> Texto con clase 1 y 2</div>
```

Esta combinación permite que cada clase defina una característica y en cada elemento puede escoger cual utilizar.

Selector id

Permite seleccionar elementos en función del atributo **id**.

El selector **id** utiliza **#**

El selector universal puede ser omitido.

Ejemplo: selecciona todos los elementos **menu** y lo alinea

```
*#menu {  
text-align: left;  
}
```


Selectores de atributos

Se puede aplicar un estilo a un elemento que posea un atributo dado. Pueden aparecer de cuatro maneras:

- `[att]`: los elementos que tienen asignado el atributo “att”, cualquiera sea el valor del atributo.
- `[att = val]`: cuando el valor del atributo “att” del elemento es exactamente “val”.
- `[att ~ = val]`: cuando el valor del atributo “att” del elemento es una lista de palabras separadas por espacios, una de las cuales es exactamente “val”.
- `[att | = val]`: cuando el valor del atributo “att” del elemento es una lista de palabras separadas comenzando con “val”.

Selectores de atributos

Todos los elementos `<acronym>` que tengan un atributo “title” con cualquier valor, se mostrarán en color rojo y fondo gris, para atraer la atención. Cualquier elemento que tenga un atributo “title” tendrá fondo amarillo

Ejemplo

```
acronym[title] {  
  color: red;  
  background-color: gray;  
}  
[title] {  
  background-color: yellow;  
}
```

Selectores descendentes

Permite seleccionar elementos en función del contexto en el que aparece.

En CSS1 se denominaban contextuales en CSS2 descendentes o descendientes.

Ejemplo

Selecciona el elemento em dentro de una lista y lo pasa a mayúsculas.

```
ul em {  
  text-transform: uppercase;  
}
```

Selectores secundarios: de hijos

Permite seleccionar elementos que se encuentran por debajo del elemento principal en la jerarquía del documento.
Se utiliza el símbolo `>` para realizar la selección

Ejemplo

Selecciona los párrafos debajo del elemento `body` (no seleccionaría los elementos `p` dentro de los `div`).

```
body>p {  
font-weight: bold;  
}
```

Selectores secundarios: de hermanos

Permite seleccionar elementos que poseen el mismo padre en la jerarquía del documento.

Se utiliza el símbolo + para realizar la selección, marcando el orden en el que aparecen.

Ejemplo

Selecciona el elemento “h2” cuando aparece justo después de un elemento “h1” para reducir el margen.

```
h1 + h2 {  
margin-top: 0mm;  
}
```

Selectores combinados

Los selectores pueden combinarse para realizar una selección más “fina”.

Ejemplo 1

Selecciona las listas de clase “inventario” y modifica la viñeta.

```
ul.inventario {  
list-style: disc url(/images/buttom.gif) inside;  
}
```

Ejemplo 2

Selecciona todos los párrafos del documento.

```
body p{  
font-weight: bold;  
}
```

Definición Pseudo-classes y Pseudo-elementos

- Pseudo-clase: clasifican a los elementos basandose en ciertas características. Las pseudo-classes pueden ser dinámicas, en el sentido de que un elemento puede adquirir o perder una pseudo-clase a medida que el usuario interactúa con el documento.
- Pseudo-elementos: crean abstracciones acerca de la estructura del documento más allá de aquellas especificadas por el lenguaje del documento.

Pseudo-clases

- `:first-child` equivale a un elemento que es el primer hijo de algún otro elemento.
- `:link` se aplica a los vínculos que aún no han sido visitados.
- `:visited` se aplica una vez que el vínculo ha sido visitado por el usuario.
- `:hover` se aplica mientras el usuario señala un elemento pero no lo activa.
- `:active` se aplica mientras un elemento está siendo activado por el usuario
- `:focus` se aplica mientras un elemento tiene el foco (acepta eventos del teclado u otras formas de entrada de texto).
- `:lang(C)` se aplica a un elemento si está marcado con el lenguaje C.

Pseudo-elementos

- `:first-line` aplica estilos especiales a la primera línea de un párrafo.
- `:first-letter` puede ser usado para las “capitulares”, que son efectos tipográficos de uso frecuente.
- `:before` y `:after` pueden ser usados para insertar un contenido generado antes o después del contenido del elemento.

Unidades de longitud

- CSS admite varias categorías de unidades de longitud:
 - Absolutas (pt, mm, cm, in, pc): describen distancias físicas
 - Relativas (em, ex): distancia relativa a otras distancias
 - Píxel (px): es relativa, pero es diferente en pantalla e impresora, por ejemplo.

Unidades absolutas

- Especifican longitudes fijas
 - milímetro: mm
 - centímetro: cm (1 cm = 10 mm)
 - pulgada (inch): in (1 in = 25,4 mm)
 - punto (point): pt (72 pt = 1 in)
 - pica: pc (1 pc = 12 pt)
- Debemos evitar las medidas absolutas: no escalan
- Sólo deberíamos usarlas si conocemos las medidas del dispositivo de salida (p. ej. una página A-4)

Unidades relativas

- Especifican longitudes relativa al tamaño de la fuente
- La mas comun es “em”
 - Generalmente se refiere al tamaño de fuente del propio elemento
 - La única excepción es la propiedad font-size, que se refiere al tamaño de fuente del elemento padre

Ejemplo

```
h1 { font-size: 2em; }
```

<h1> será dos veces el tamaño del elemento <body>

Utilización de “em”

- Los diseñadores acostumbrados a la publicación en papel pueden estar tentados a especificar, por ejemplo, una fuente de 20 puntos
- “em” es siempre una mejor solución
- El uso de “em” no debe limitarse a los tamaños de fuente; deberíamos usarlo para todo:
 - Márgenes, espaciado (padding), sangrado...
 - A veces los porcentajes funcionan tan bien como em, o incluso mejor

Utilización de píxel

Ejemplo

```
h1 { border-width: 4px; }
```

- No sabemos las dimensiones de la pantalla, de la ventana del navegador...
- No representan lo mismo en la pantalla y en la impresora, por ejemplo

Porcentajes como medidas

- Muchas propiedades aceptan porcentajes como valores
- Muy parecidos a las unidades relativas
- A qué se aplica el porcentaje depende de cada propiedad concreta. Normalmente, es un porcentaje del valor de esa misma propiedad en el elemento padre

Ejemplo

```
h1 { font-size: 200 %; }
```

La fuente es el 200 % del tamaño de la fuente del padre (normalmente será el <body>)

- 1 Anatomía de estilos
- 2 Herencia y Cascada**
 - Herencia
 - Cascada
- 3 Propiedades de Aspecto
- 4 Modelo de caja
- 5 Modelo de Formato Visual

Herencia

- Herencia: las propiedades y sus valores se heredan en los elementos descendientes según la estructura del documento
- Algunas propiedades que no se heredan como los márgenes y el relleno.

Ejemplo

Las declaraciones de body son heredadas por los elementos que se encuentran dentro de body.

```
body{  
color: #000;  
font-family: Georgia, "Times New Roman", serif;  
}
```

Cascada

Las hojas de estilo pueden tener tres orígenes diferentes: el autor, el usuario y la aplicación del usuario.

- Autor (diseñador): El autor especifica las hojas de estilo para un documento fuente de acuerdo a las convenciones del lenguaje del documento.
- Usuario: El usuario puede especificar información de estilo para un documento particular. Por ejemplo, el usuario puede especificar un archivo que contenga una hoja de estilo o la aplicación del usuario puede proporcionar una interfaz que genere una hoja de estilo del usuario.
- Aplicaciones del usuario: Las aplicaciones del usuario con conformidad deben aplicar una hoja de estilo predeterminada previa a todas las hojas de estilo para un documento.

Selección según el medio

Las hojas de estilo pueden definirse para diferentes medios:

- all Todos los dispositivos
- braille Dispositivos táctiles braille
- embossed Impresoras braille
- handheld Dispositivos de mano (pantallas pequeñas, ancho de banda reducido, etc.)
- print Para documentos paginados y mostrados en vista de impresión
- projection Dispositivos de proyección de presentaciones
- screen Pantallas a color de equipos informáticos
- speech Para sintetizadores de voz. Similar a `.aural`
- tty Dispositivos de visualización con capacidades limitadas
- tv Televisores

La regla @media

Esta regla permite definir diferentes valores para las propiedades de estilos para diferentes medios dentro de una hoja de estilo.

Ejemplo

```
@media screen {  
p{font-family:verdana,sans-serif; font-size:14px}  
}  
@media print{  
p{font-family:times,serif; font-size:10px}  
}  
@media screen,print{  
p{font-weight:bold}  
}
```

La declaración !important

Esta declaración permite marcar como más “importante” el valor de una propiedad de otro para un mismo elemento.

Ejemplo

```
*{  
color: black !important;  
background-color: yellow;  
}  
div{color: blue;  
background-color: white;}
```

Selección según el autor

Para un medio dado, la prioridad se establece según el origen de los estilos y la declaración `!important`. En caso de igualdad, las prioridades son (de más a menos):

- Los estilos del usuario con `!important`
- Los estilos del autor(diseñador) con `!important`
- Los estilos del autor(diseñador)
- Los estilos del usuario
- Los estilos por defecto del navegador

Selección por especificidad

Se calcula una cifra de 4 números (abcd) según las siguientes normas:

- a=1 si un estilo se encuentra definido a través de un atributo “style”
- b=1 representa el número de “id” en el selector
- c=1 representa el número de clases, pseudo-clases y selectores de atributos que aparecen en el selector
- d=1 representa el número de elementos XHTML utilizados en el selector.
- Los estilos por defecto del navegador

Los números más altos obtenidos tendrá más prioridad que los más bajos.

Calculo de la especificidad

Ejemplo

Selector	Calculo Cifras	Especificidad
#id1	a=0 b=1 c=0 d=0	0100
ul ul li.red	a=0 b=0 c=1 d=3	0013
li.red	a=0 b=0 c=1 d=1	0011
li	a=0 b=0 c=0 d=1	0001

Selección por orden de aparición

Después de los criterios anteriores, si existe aún conflicto el orden de aparición de las reglas es lo que determina el estilo a aplicar. Cuando dos reglas tengan la misma especificidad, prima la última regla especificada.

- 1 Anatomía de estilos
- 2 Herencia y Cascada
- 3 Propiedades de Aspecto**
 - Colores y Fondo
 - Fuentes
 - Texto
 - Listas
- 4 Modelo de caja
- 5 Modelo de Formato Visual

Especificación de Colores

- Se puede especificar el color de fondo y el aspecto del fondo de un elemento (colores o imágenes).
- Para especificar un color se puede utilizar:
 - Nombre: CSS tiene 16 nombre definidos
 - Código RGB: Red-Green-Blue
 - Código hexadecimal

Colores predefinidos

aqua	black	blue	fuchsia
gray	green	lime	maroon
navy	olive	purple	red
silver	teal	yellow	white

Código RGB y hexadecimal

- Cada uno de los componentes (Rojo, Verde y Azul) se representa por un valor entre 0 y 100
- En CSS se utiliza

rgb

rgb (porcentaje, porcentaje, porcentaje)

rgb (0-255,0-255,0-255)

valor hexadecimal

#ff0000

#f00

Propiedad **Color**

```
color: <color> | inherit
```

- El valor de esta propiedad se especifica utilizando la sintaxis vista

Ejemplo

```
strong{color: red;}  
strong{color: rgb(255,0,0);}  
strong{color: #f00;}  
strong{color: #ff0000;}  
a:link{color: blue;}  
a:visited{color: purple;}
```

Propiedades de fondo

- Establecen aspectos relacionados con el fondo de un elemento
- Puede ser transparente, un color o una imagen
- Se puede establecer la posición de la imagen, si se repite o no y si se mueve o no al desplazar la página

Propiedades de fondo

background-color
background-image
background-repeat
background-attachment
background-position
background

Propiedad **background-color**

`background-color: <color> | transparent | inherit`

- Determina del color de fondo de un elemento
- “transparent” hace que los colores subyacentes se vean a través.
- Se debe asegurar que la combinación “color” y “background-color” se visualice correctamente.

Ejemplo

```
body{color: red;  
color: #333300;  
background-color: #CCCCFF;  
}
```


Propiedad **background-image**

`background-image: <url> | none | inherit`

- Determina la imagen de fondo de un elemento.
- Se debe especificar un color de fondo que será usado cuando la imagen no esté disponible.

Ejemplo

```
body{background-image: url(images/elmo.gif);}  
p { background-image: none }
```

Propiedad **background-repeat**

background-repeat: repeat | repeat-x | repeat-y | no-repeat | inherit

- Si se especifica una imagen de fondo, esta propiedad especifica si la imagen es repetida (mosaico), y de qué modo.
 - “repeat”: La imagen es repetida tanto horizontal como verticalmente.
 - “repeat-x”: La imagen es repetida sólo horizontalmente.
 - “repeat-y”: La imagen es repetida sólo verticalmente.
 - “no-repeat”: La imagen no se repite: sólo una copia de la imagen es visualizada.

Ejemplo

```
body{background-image: url(images/elmo.gif);  
background-repeat: repeat; }
```

Propiedad **background-position**

```
background-position: [ [<porcentaje> | <medida> ] {1,2}  
| [ [top | center | bottom]  
|| [left | center | right] ] ]
```

- Si se ha especificado una imagen de fondo, esta propiedad especifica su posición inicial.
- las tres formas de especificar la posición:
 - Porcentajes
 - Posición absoluta
 - Palabras reservadas (left, top, center, bottom...)

Propiedad **background-position** con porcentajes

Ejemplo

```
body{  
background-position: 14 % 84 %;  
background-image: url(images/elmo.gif);  
}
```

El punto ubicado a un '14 %' del costado izquierdo de la imagen y a un 84 % hacia abajo será situado en el punto ubicado a un 14 % del costado y a un 84 % hacia abajo en el área de relleno.

Propiedad **background-position** con posiciones absolutas

Ejemplo

```
body{  
background-image: url(images/elmo.gif);  
background-position: 2cm 3cm;  
}
```

La esquina superior izquierda de la imagen es ubicada 2cm a la derecha y 3cm abajo de la esquina superior izquierda del área de relleno.

Propiedad **background-position** con palabras reservadas

Ejemplo

```
body { background: url("elmo.gif") right top } /* 100 % 0 % */  
body { background: url("elmo.gif") top center } /* 50 % 0 % */  
body { background: url("elmo.gif") center } /* 50 % 50 % */  
body { background: url("elmo.gif") bottom } /* 50 % 100 %*/
```

Propiedad **background-attachment**

background-attachment: scroll | fixed

- Si se especifica una imagen de fondo, esta propiedad especifica si es fija con respecto al acceso visual ('fixed') o se desplaza junto con el documento ('scroll').
- Utilizado para las marcas de agua

Ejemplo

```
body{background-image: url(images/elmo.gif);  
background-repeat: repeat;  
background-attachment: fixed;  
}
```

Propiedad **background**

```
background: [<background-color> || <background-image> ||  
<background-repeat> || <background-attachment> ||  
<background-position>]
```

- Propiedad resumen para fijar la propiedades individuales del fondo (es decir, 'background-color', 'background-image', 'background-repeat', 'background-attachment' y 'background-position') en un mismo lugar en la hoja de estilo.
- La propiedad 'background' primero coloca a todas las propiedades individuales del fondo en sus valores iniciales, luego asigna los valores explícitamente dados en la declaración.

Ejemplos de la propiedad **background**

Ejemplo

```
body {background: red url(images/elmo.gif) repeat fixed; }
```

Equivale a:

```
body {  
background-color: red;  
background-image: url(images/elmo.gif);  
background-repeat: repeat;  
background-attachment: fixed;  
}
```

Introducción a las Fuentes

- Con las hojas de estilo, podemos especificar propiedades de las fuentes, como el tipo, tamaño, peso, etc.

Clasificación de las fuentes

- Las familias de fuentes se pueden clasificar en unas pocas categorías, atendiendo a diferentes cuestiones:
 - Si tiene “serifs” o es “sans-serif”
 - Si las letras tienen una anchura proporcional (variable) o fija (monoespaciada)
 - Si trata de imitar a la escritura a mano
 - Si está pensada para propósitos decorativos

Familias de las fuentes

- Serif
- Sans-serif
- Cursive
- Fantasy
- Monospace

Propiedad **font-family**

```
font-family: [[ <nombre-de-la-familia> | <familia-genérica> ],]*  
[<nombre-de-la-familia> | <familia-genérica>] | inherit
```

- Las fuentes deben estar instaladas en el ordenador del usuario
- Permite especificar una lista de fuentes separadas por comas
- Por si no tuviera ninguna de ellas, como último elemento de la lista deberíamos poner siempre la familia genérica a la que pertenecen

Propiedad **font-family**

Ejemplo

```
font-family: Georgia, Garamond, "Times New Roman", serif;
```

- Las familias concretas deben escribirse tal cual (respetando mayúsculas y minúsculas)
- Por si acaso, es mejor entrecomillar las que contengan espacios, dígitos o guiones
- Las familias genéricas son en minúsculas ¡No entrecomillarlas!
- No usar mas de 2 o 3 fuentes
- Un convenio es tener el texto en Serif y títulos en Sans-Serif
- Monoespaciadas para código, nombre de ficheros, etc.

Medidas de las fuentes

- Habitualmente las fuentes se miden en puntos, medida típica de las imprentas
- Con ella se mide el tamaño, el espaciado entre líneas, el grosor de los trazos, etcétera
 - Variantes:
 - Punto europeo continental (Didot = 0.376065 mm)
 - Punto angloamericano (pica = 0.351461 mm)
 - Otro punto angloamericano (1/72 in. = 0.352778 mm)
- CSS usa la última de estas variantes

Propiedad **font-size**

```
font-size: <longitud> | <porcentaje> | <tamaño-absoluto> |  
<tamaño-relativo> | inherit
```

- Longitud y porcentaje son los tipos generales de longitudes
- Tamaño absoluto y relativo son palabras reservadas específicas para esta propiedad.

Ejemplos de la propiedad **font-size**

Ejemplo

```
body {font-size: 12 pt; }  
pre {font-size: 0.4 em; }  
h1.capitulo {font-size: 16 px; }  
h1 {font-size: 1.2 em; }
```

Porcentajes en la propiedad **font-size**

- En esta propiedad utilizar porcentajes es equivalente a utilizar “em”

Ejemplo

```
h1 {font-size: 200 %; }
```

h1 es un 20 % mayor que el tamaño de fuente del elemento padre.

Es equivalente a:

```
h1 {font-size: 1.2 em; }
```

Valores absolutos de **font-size**

- Valores predeterminados:
 - xx-small
 - x-small
 - small
 - medium (predeterminada)
 - large
 - x-large

Valores relativos en **font-size**

- Permiten especificar el tamaño en función del contexto
- Dos palabras reservadas:
 - larger
 - smaller

Ejemplo

```
body {font-size: medium; }  
h1 {font-size: larger; }  
p {font-size: smaller; }
```

Propiedad **font-style**

`font-style: normal | italic | oblique | inherit`

- Italic y oblique son similares pero no iguales
- Las familias sans-serif las suelen considerar iguales

Propiedad **font-variant**

font-variant: normal | small-caps | inherit

- En una fuente small-caps, los signos para la letras minúsculas son iguales a las mayúsculas, pero en un tamaño menor y con proporciones ligeramente diferentes (en castellano se denominan versales o versalitas).
- Si no está disponible una versalita genuina, las aplicaciones del usuario deberán simularla, por ejemplo tomando una fuente normal y reemplazando las letras minúsculas por mayúsculas reducidas

Propiedad **font-weight**

```
font-weight: normal | bold | bolder | lighter | 100 | 200 | 300 | 400 |  
500 | 600 | 700 | 800 | 900 | inherit
```

- Especifica el peso de una fuente.
 - 100 to 900: cada número indica un peso que es al menos mas fuerte como su predecesor.
 - normal: Igual a '400'.
 - bold: Igual a '700'.
 - bolder: especifica el siguiente peso
 - lighter: especifica el anterior peso

Propiedad **font**

```
font:[ [ <font-style> || <font-variant> || <font-weight> ]?  
<font-size> [ / <line-height> ]? <font-family> ]
```

- Es una propiedad resumen de las anteriores
- Permite especificar casi todas las propiedades de una fuente a la vez
- También añade una nueva: line-height
- font-style, font-variant y font-weight son opcionales y pueden ir en cualquier orden
- font-size y font-family son obligatorios
- font-size puede ir seguido por una barra “/” y por el line-height

Ejemplos de la propiedad **font**

Ejemplo 1

```
h1 { font: 1.3em sans-serif; }
```

Equivale a:

```
h1 { font-size: 1.3em;  
font-family: sans-serif; }
```

Ejemplo 2

```
p { font: italic bold 12 pt/14pt Times, serif; }
```

Equivale a:

```
h1 { font-style: italic;  
font-weight: bold; font-size: 12 pt; font-family: Times, serif; }
```

Texto

- Existen propiedades para modificar la presentación visual de los caracteres, espacios, palabras y párrafos. Subrayar el texto:
 - Sangría: la propiedad 'text-indent'
 - Alineación: la propiedad 'text-align'
 - Decoración
 - Espaciado de letras y palabras: las propiedades 'letter-spacing' y 'word-spacing'
 - Cambio de minúsculas y mayúsculas: la propiedad 'text-transform'
 - Espacios en blanco: la propiedad 'white-space' Poner una línea encima

Propiedad **text-indent**

`text-indent : <medida> | <porcentaje> | inherit`

- Especifica la sangría de la primer línea de texto en un bloque
- Medida: El sangrado tiene una medida fija.
- Porcentaje: El sangrado es un porcentaje del ancho del bloque de contención.
- Puede ser negativo, pero pueden haber limitaciones específicas de la implementación

Propiedad **text-align**

`text-align` : left | right | center | justify | inherit

- Texto alineado a la izquierda, derecha, centro y justificado

Propiedad **text-decoration**

`text-decoration: none | [underline || overline || line-through || blink]`

- underline: Subrayar el texto
- overline: Pone una línea encima del texto
- line-through: Tacha el texto con una línea por medio
- blink: El texto parpadea (no en IE)

Propiedad **text-transform**

`text-transform: none | capitalize | uppercase | lowercase`

- Controla el efecto de los cambios entre mayúsculas y minúsculas en el texto de un elemento
- Pone en mayúsculas, minúsculas o capitaliza el texto

Propiedad **letter-spacing** y **word-spacing**

`letter-spacing: normal | <medida>`

`word-spacing : normal | <medida>`

- especifica el comportamiento de los espacios entre los caracteres o palabras del texto.
- normal: el espaciado es el normal de la fuente actual.
- medida: indica el espacio entre caracteres o palabras

Propiedad **white-space**

`white-space` : normal | pre | nowrap

- Declara cómo son tratados los espacios en blanco dentro del elemento.
- normal: cerrar las secuencias de espacios en blanco, y cortar las líneas
- pre: Las líneas son cortadas solamente en los avances de línea de la fuente
- nowrap: cierra los espacios en blanco como en normal, pero suprime los cortes de línea dentro del texto

Propiedades para **listas**

- list-style-type
- list-style-image
- list-style-position
- list-style
- Estas propiedades se aplican sólo a los elementos cuya propiedad display es “list-item”

Propiedad `list-style-type`

```
list-style-type : [ disc | circle | square | decimal | decimal-leading-zero |  
lower-roman | upper-roman | lower-greek | lower-latin | upper-latin | armenian |  
georgian | lower-alpha | upper-alpha | none | inherit ]
```

- Especifica la apariencia del marcador de los ítems de la lista si 'list-style-image' tiene el valor 'none'

Propiedad `list-style-image`

```
list-style-image : [ uri | none ]
```

- Imagen aplicable a los elementos de las listas

Propiedad `list-style-position`

`list-style-position` : listas [inside | outside]

- Posición dentro de la lista de los elementos marcadores de las listas
- `inside`: La caja del marcador está en la primer caja a nivel de línea en la caja de bloque principal, luego de la cuál fluye el contenido del elemento.
- `outside`: La caja del marcador está fuera de la caja de bloque principal.

Propiedad **list-style**

```
list-style : [ <list-style-type> || <list-style-position> ||  
<list-style-image> ]
```

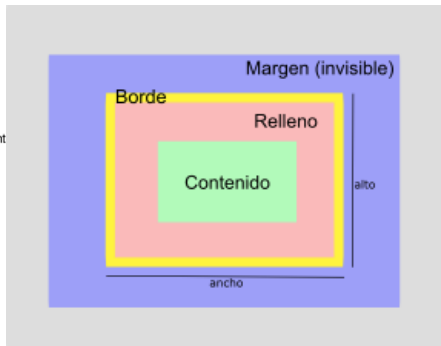
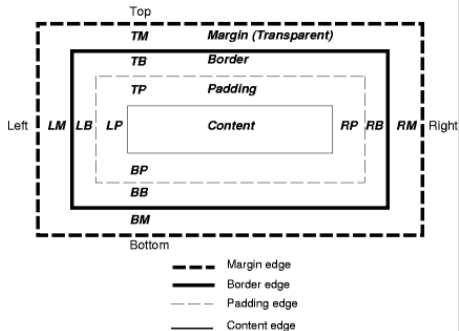
- Permite establecer el estilo de la lista, la imagen y/o la posición

- 1 Anatomía de estilos
- 2 Herencia y Cascada
- 3 Propiedades de Aspecto
- 4 Modelo de caja**
 - Bordes
 - Márgenes
 - Relleno
- 5 Modelo de Formato Visual

Modelo de caja

- CSS describe las cajas rectangulares que son generadas por los elementos en la estructura del documento
- Un elemento de nivel de bloque, como `<div>` o `<p>` se compone de una sola caja
- Uno de línea, como `` o ``, puede dividirse en varias cajas pequeñas (si está en varias líneas)
- El tamaño y posición de cada elemento son relativos a su caja circundante
 - margen: especifican el ancho del área del margen de una caja
 - relleno (padding): especifican el ancho del área de relleno de una caja
 - borde: el ancho, color y estilo del área del borde de una caja

Modelo de caja



Ejemplo de Calculo de la anchura de un elemento

- Anchura (width): 500 pixels
- Relleno (padding): 20 pixels
- Borde (border): 5 pixels
- Margen (margin): 15 pixels

Calculo de la anchura total

$$15+5+20+500+20+5+15 = 580 \text{ pixels}$$

Modelo de caja - Fondo

- El estilo del fondo de las distintas áreas de una caja es determinado como sigue:
 - Area del contenido: La propiedad 'background' del elemento generador.
 - Area del relleno (padding): La propiedad 'background' del elemento generador.
 - Area del borde (border): Las propiedades del borde del elemento generador.
 - Area del margen (margin): Los márgenes son siempre transparentes.

Borde

- Hay 20 propiedades de borde que permiten establecer el ancho, color y estilo de un borde
- Hay cinco propiedades “resumen” que permiten especificar todos los aspectos a la vez sobre uno o más bordes de un elemento:
 - border-left
 - border-right
 - border-top
 - border-bottom
 - border

Estilo del borde

```
'border-top-style', 'border-right-style', 'border-bottom-style',  
'border-left-style' : <border-style>
```

```
'border-style' : <border-style>{1,4}
```

- Valor inicial del estilo de borde es 'none', ningún borde será visible a menos que se establezca el estilo de borde.
- Todos los bordes son dibujados por encima del fondo de la caja.
- El color de los bordes de algunos estilos ('groove', 'ridge', 'inset' y 'outset') depende de la propiedad 'color' del elemento.

Estilos del borde

- none (ningún borde): fuerza al valor computado de 'border-width' a '0'.
- hidden: igual a 'none', excepto en términos de resolución de conflictos de bordes para los elementos de tabla.
- dotted: serie de puntos.
- dashed: pequeños segmentos de línea.
- solid: es un único segmento de línea.
- double: dos líneas sólidas. La suma de las dos líneas y el espacio entre ellas es igual al valor de 'border-width'.
- groove: como tallado en el lienzo.
- ridge: como sobresaliendo del lienzo.
- inset: la caja parece empotrada en el lienzo.
- outset: la caja parece sobresalir del lienzo.

Ejemplo de estilo del borde

Ejemplo

```
p {  
border-style: solid dotted  
/*borde sup-inf solido izq-dch puntos */ }
```

Anchura del borde

```
'border-top-width', 'border-right-width', 'border-bottom-width',  
'border-left-width' : [ thin | medium | thick | <longitud> ] | inherit
```

```
'border-width' : [ thin | medium | thick | <longitud> ]{1,4}
```

- thin: un borde fino.
- medium: un borde mediano.
- thick: un borde grueso.
- longitud: el grosor del borde tiene un valor explícito (no pueden ser negativas)

Ejemplo de anchura del borde

Ejemplo

```
h1 { border-width: thin }  
h1 { border-width: thin thick }  
h1 { border-width: thin thick medium }
```


Color del borde

```
'border-top-color', 'border-right-color', 'border-bottom-color',  
'border-left-color' : <color>
```

```
'border-color' : <color>{1,4} | transparent | inherit
```

- color: especifica un valor de color.
- transparent: el borde es transparente (puede tener grosor).
- Si no se especifica, toma el valor de `<color>` del elemento.

Ejemplo de color del borde

Ejemplo

```
p {  
  color: red;  
  background: white;  
  border: solid;  
  /* borde red */ }
```

Propiedades resumidas del borde

- Ancho, estilo y el color para el borde superior, derecho, inferior o izquierdo

'border-top', 'border-right', 'border-bottom', 'border-left':

$\langle \textit{border - top - width} \rangle \parallel \langle \textit{border - style} \rangle \parallel \langle \textit{color} \rangle$

- Ancho, el estilo y el color para los 4 bordes
- No puede definir diferentes valores para los cuatro bordes

'border':

$\langle \textit{border - width} \rangle \parallel \langle \textit{border - style} \rangle \parallel \langle \textit{color} \rangle$

Ejemplos de las propiedades resumidas para el borde

Ejemplo 1

```
p {  
border: solid red }  
Equivale a: p {border-top: solid red;  
border-right: solid red;  
border-bottom: solid red;  
border-left: solid red; }
```

Ejemplo 2: superposición

```
blockquote { border-color: red;  
border-left: double;  
color: black; /*El borde izq será negro*/  
}
```

Propiedades de Márgenes

```
margin-top, margin-right, margin-bottom, margin-left : <longitud>  
| <porcentaje> | auto | inherit
```

- Estas propiedades determinan los márgenes superior, derecho, inferior e izquierdo de una caja.
- El porcentaje se refiere al ancho del bloque que lo contiene
- Puede ser negativo, pero con cuidado
- Siempre es transparente

Propiedad Margin

```
margin: [ <longitud> | <porcentaje> | auto ]{1,4}|inherit
```

- El porcentaje se refiere al ancho del bloque que lo contiene
- Puede tener entre uno y cuatro valores:
 - Si sólo tiene uno, se aplica a los cuatro lados
 - Si hay dos, el primero es para los márgenes superior e inferior y el segundo para el derecho e izquierdo
 - Si hay tres, el primero es el superior, el segundo el derecho e izquierdo, y el tercero el inferior
 - Si hay cuatro: superior/derecho/inferior/izquierdo (siguiendo un reloj imaginario)

Ejemplos de Márgenes

Ejemplo

```
body { margin: 2em }  
body { margin: 1em 2em }  
body { margin: 1em 2em 3em }  
body { margin-top: 1em;  
margin-right: 2em;  
margin-bottom: 3em;  
margin-left: 2em;  
}
```

Propiedades de Padding (Relleno)

padding-top, padding-right, padding-bottom, padding-left:
<longitud> | <porcentaje> | inherit

- El porcentaje se refiere al ancho del bloque que lo contiene
- El relleno automáticamente coge el mismo fondo que el del elemento
- NUNCA puede ser negativo

Propiedad Padding (Relleno)

```
padding: [ <longitud> | <porcentaje> | auto ]{1,4} | inherit
```

- Describe cuánto espacio se inserta entre un elemento y su margen.
- Si hay borde, entre un elemento y su borde
- El porcentaje se refiere al ancho del bloque que lo contiene
- Si sólo se especifica un valor, éste se aplica a los cuatro lados
- Si falta algún valor, los que faltan se toman de su lado opuesto
 - Comenzando por arriba, en el sentido de las agujas del reloj
 - Si están los cuatro: arriba, derecha, abajo, izquierda

Ejemplos de Padding

- El color de la superficie o la imagen del área de relleno se especifica a través de la propiedad “background”

Ejemplo

```
h1 {  
background: white;  
padding: 1em 2em;  
}
```

- 1 Anatomía de estilos
- 2 Herencia y Cascada
- 3 Propiedades de Aspecto
- 4 Modelo de caja
- 5 Modelo de Formato Visual**
 - Dimensiones de los elementos
 - Elementos a nivel de línea y bloque
 - Esquema de Posicionamiento
 - Tipos de posicionamiento
 - Bloque de contención o contenedor

Introducción al Modelo de Formato Visual

- Determinar cómo las aplicaciones del usuario procesan la estructura del documento para los medios visuales.
- Cada elemento genera cero o más cajas de acuerdo al modelo de caja. La composición de estas cajas se gobierna por:
 - Dimensiones de la caja y tipo.
 - Esquema de posicionamiento (flujo normal, flotante y absoluto).
 - Relación entre los elementos de una estructura del documento.
 - Información externa (por ejemplo tamaño del acceso visual).
- Las cajas generadas actúan como bloques de contención para las cajas descendientes
- Cada caja tiene una posición dada con respecto a su bloque de contención, pero lo puede desbordar.

Propiedad **Width** - Anchura

'width' : <longitud> | <porcentaje> | auto | inherit

- El porcentaje se refiere al ancho del elemento padre
- Se aplica sólo a elementos de bloque e imágenes
- Normalmente no asignaremos explícitamente el ancho, sino que utilizamos márgenes, el relleno y los bordes

Propiedad **Height** - Altura

'height' : <longitud> | <porcentaje> | auto | inherit

- El porcentaje se refiere a la altura del elemento padre
- Se aplica sólo a elementos de bloque e imágenes
- Igual que con la anchura, no se asigna explícitamente la altura, sino que se utilizan márgenes, el relleno y los bordes

Propiedad **Overflow**

'overflow' : visible | hidden | scroll | auto | inherit

- Especifica si el contenido de un elemento a nivel de bloque es recortado cuando desborda la caja del elemento.
- Visible: el contenido no está recortado
- Hidden: el contenido es recortado, no se utilizan barras de desplazamiento.
- Scroll: el contenido es recortado y se muestran barras de desplazamiento.

Propiedades 'min-height' y 'max-height'

'min-height': <longitud> | <porcentaje>

'max-height': <longitud> | <porcentaje> | none

- Permite restringir las alturas de las cajas
- El porcentaje se refiere al ancho del elemento padre

Propiedades 'min-width' y 'max-width'

'min-width': <longitud> | <porcentaje>

'max-width': <longitud> | <porcentaje> | none

- Permite restringir los anchos de las cajas
- El porcentaje se refiere al ancho del elemento padre

Elementos a nivel de línea y bloque

- Elementos a nivel de bloque son los tratados visualmente como bloques (p.ej. párrafos).
- Los elementos a nivel de bloque generan una caja de bloque principal que sólo contiene cajas de bloque
- Elementos a nivel de línea son los que no forman nuevos bloques de contenido, el contenido es distribuido en líneas (ej., fragmentos de texto enfatizado dentro de un párrafo, imágenes en línea, etc.).
- Se pueden modificar con la propiedad 'display'.

Display

Establece como se “devuelven” los elementos

'display' : inline | block | list-item | run-in | inline-block | table | inline-table |
tablerow-group | table-header-group | table-footer-group | table-row |
tablecolumn-group | table-column | table-cell | table-caption | none

- Valor inicial es 'inline'.
- inline: el elemento genera una o mas cajas de línea.
- block: el elemento genera una caja de bloque principal.
- inline-block: genera una caja de bloque, que fluye como una caja en línea.
- list-item: genera una caja de bloque principal y una caja de línea list-item.
- run-in: crea cajas de bloque o de línea, según el contexto.
- table, inline-table, table-row-group, table-column, table-column-group, table-header-group, table-footer-group, table-row, table-cell y table-caption: el elemento se comporta como un elemento tabla
- none: no genere ninguna caja en la estructura del formato.

Esquema de Posicionamiento

- Flujo normal: el flujo normal incluye formato de bloque de cajas de bloque, formatos en línea de cajas en línea, posicionamiento relativo de cajas de bloque o en línea, y posicionamiento de cajas run-in.
- Flotantes: una caja se presenta primero de acuerdo al flujo normal, luego se saca del flujo normal y se mueve a la izquierda o derecha tanto como sea posible. El contenido puede fluir a lo largo del costado del flotante.
- Posicionamiento absoluto: una caja es quitada completamente del flujo normal (sin impacto sobre los hermanos siguientes) y se le asigna una posición con respecto al bloque de contención.
- Posicionamiento fijo: un elemento ocupa una posición fija dentro de la página.

Posicionamiento **flotante**

- Un flotante es una caja que es desplazada a la izquierda o a la derecha en la línea actual.
- El contenido puede fluir por su costado y debe tener un ancho explícito

Posicionamiento **flotante**: la Propiedad 'float'

'float' : left | right | none | inherit

- Especifica si una caja debe flotar a la izquierda, derecha o no debe flotar en absoluto
- Existen reglas específicas del comportamiento de los flotantes (ver especificación)
- Se debe especificar las dimensiones para que tenga efecto

Posicionamiento **flotante**: la Propiedad 'clear'

'clear' : none | left | right | both inherit

- Especifica si puede haber o no elementos flotantes al lado de un elemento, y en qué lado
- left y right: no puede haber elementos flotantes en su lado izquierdo y derecho, respectivamente
- both: no se permiten elementos flotantes, en ninguno de sus dos lados
- Se usa conjuntamente con float

Propiedad **Position**

'position' : static | relative | absolute | fixed

- Establece el esquema de posicionamiento junto con “float”
- static: caja normal, presentada de acuerdo al flujo normal. ('top', 'right', 'bottom', y 'left' no se aplican).
- relative: la posición de la caja es calculada de acuerdo al flujo normal, la caja es deslizada de modo relativo a su posición normal.
- absolute: la posición (y tamaño) de la caja se especifican con las propiedades 'top', 'right', 'bottom', y 'left'.
- fixed: la posición de la caja es calculada de acuerdo al modelo 'absolute', pero además, la caja se fija con respecto a alguna referencia.

Desplazamiento de las cajas

'top', 'right', 'bottom', 'left' : <medida> | <porcentaje> | auto | inherit

- Los elementos posicionados (diferentes de static) generan cajas posicionadas, situadas de acuerdo estas cuatro propiedades.
- top: especifica el desplazamiento del margen superior de una caja.
- right: especifica el desplazamiento del margen derecho de una caja.
- bottom: especifica el desplazamiento del margen inferior de una caja.
- left: especifica el desplazamiento del margen izquierdo de una caja.

Posicionamiento **relativo**

- La caja es situada de acuerdo al flujo normal y luego puede moverse con relación a su posición
- Mantienen su tamaño de flujo normal, incluyendo los saltos de línea y el espacio reservado para ellos.
- Otra posibilidad es marcar un elemento como “relative” sin cambiar realmente su posición, sólo para que sirva como contenedor de otros elementos con posicionamiento absoluto

Posicionamiento Absoluto

- Permite mantener un elemento siempre en la misma posición de la pantalla (p. ej. un logotipo)
- Generalmente en estos casos indicaremos su ancho (width), además de su posición (mediante left o right y top o bottom). También se puede indicar left y right a la vez y no especificar el ancho
- También permite dividir un documento en varias partes
 - Similar a los marcos de HTML, pero con mucha más accesibilidad y mucho más control sobre el aspecto
 - Si están dentro de otro elemento posicionado (ya sea relativo, absoluto o fijo), ése será su elemento contenedor

Posicionamiento Fijo

- Es un caso particular del posicionamiento absoluto
- La posición de la caja es calculada de acuerdo al modelo 'position:absolute', y además, la caja se fija con respecto a alguna referencia de la pantalla.

Relación entre Display, position y float

- Si “display” tiene el valor “none”, no se aplican “position” y “float”
- Si “position” tiene como valor “absolute” o “fixed”, “display” es definido como “block”(ver tabla de la especificación) y “float” como ‘none’. La posición de la caja será determinada por las propiedades ‘top’, ‘right’, ‘bottom’ y ‘left’ y el bloque de contención de la caja.
- Si “float” tiene un valor distinto a “none”, “display” es definida como “block” y la caja es flotante.
- Si es un elemento contenedor, display actúa como bloque
- De otro modo, las restantes propiedades ‘display’ se aplican como están especificadas.

Propiedad **z-index**

- Las cajas se sitúan en 3 dimensiones.
- Además de su posición horizontal y vertical, las cajas se ubican a lo largo de un eje Z y son procesadas una encima de otra.

'z-index' : auto | <nº entero>

- Indica el orden de un elemento posicionado, desde el frente hasta el fondo de la pantalla
- El número más bajo iría al fondo de la pantalla, y el más alto al frente

Bloque de contención o contenedor

- La posición y tamaño de las cajas de un elemento son computadas en relación a cierto rectángulo, llamado el bloque de contención del elemento.

Ejemplo

```
<html> ...  
<body id="body">  
  <div id="div1">  
    <p id="p1">Este es texto en el primer párrafo...</p>  
    <p id="p2">Este es texto <em id="em1"> en el <strong  
id="strong1">segundo</strong> párrafo.</em></p>  
  </div>  
</body>  
</html>
```

Bloque de contención o contenedor: **fixed**

- El bloque contenedor de los elementos posicionados “fixed” es siempre la pantalla
- El padre, o cualesquiera otros elementos posicionados, no tienen influencia sobre estos elementos

Bloque de contención o contenedor: **relative**

- Los elementos con posicionamiento relativo, al igual que los normales (static), no tienen bloque contenedor
- Las propiedades top, right, bottom y left mueven el elemento respecto a su posición normal
- Ellos sí pueden actuar como bloque contenedor de otros elementos con posicionamiento absoluto

Bloque de contención o contenedor: **absolute**

- El bloque contenedor de los elementos con posición absoluta suele ser el elemento raíz
- Si están dentro de otro elemento posicionado (ya sea relativo, absoluto o fijo), ése será su elemento contenedor

Referencias



Especificación CSS 2.0



Especificación CSS 2.1 (W3C Working Draft 06 November 2006)