

# Hojas de estilo en cascada (css)

## **Introducción**

Las hojas de estilo en cascada tienen el potencial de proporcionar a los diseñadores de páginas Web un increíble grado de control sobre el aspecto y funcionalidad de sus sitios.

Utilizando CSS podemos especificar los estilos utilizados en la página HTML de la totalidad de un sitio Web, de cada página o de cada etiqueta.

Utilizando CSS podemos reducir el tamaño de nuestros archivos HTML consiguiendo:

- reducción del tiempo de descarga
- reducción del espacio en disco del servidor
- disminución del tráfico de transmisión de los datos
- división del trabajo entre desarrolladores (PHP, ASP, etc.) y diseñadores.

Pero también tiene sus inconvenientes:

- es largo y complejo
- el navegador no muestra ningún mensaje de error si se produce cualquier problema en la hoja de estilo
- No todos los navegadores soportan CSS. Y si lo soportan, no la misma especificación o nivel.

## **Especificaciones css**

- CSS 1 (1996)
- CSS 2 (1998)
- CSS 3 (se está trabajando en él a fecha de 2005)

El CSS1 no habla casi nada acerca de tablas, no habla de nada de fuentes descargables y no divide entre la diferencia de representación en pantalla e impresora.

El CSS2 soporta fuentes descargables (aunque no en totalidad), soporta diferencias entre pantalla e impresoras.

En CSS2 se tiende a posicionar el 100% de los elementos mediante CSS, dejando las tablas para casos muy concretos. La tendencia es a ir eliminando las tablas y utilizando el posicionamiento CSS.

---

## Reglas básicas

---

La forma de trabajar con estilos varía de la forma tradicional.

Por ejemplo:

```
<h2><font color="red">Texto de encabezado h2 en rojo</font></h2>
```

en HTML tradicional. Se expresaría así con estilos:

```
<style>
h2 {color:red;}
</style>
<h2>Texto de encabezado h2 en rojo</h2>
```

A la expresión **h2 {color:red;}** se la denomina regla. Y cada regla consta de dos partes:

- Selector: la parte de la regla que selecciona a qué partes del documento se le debe aplicar el estilo. En nuestro ejemplo: h2.
- Declaración: que se compone de una lista de parejas "propiedad:valor" separadas por punto y coma. En el ejemplo color:red, siendo color la propiedad y red el valor.

Los selectores se pueden agrupar. En el siguiente caso:

```
h1 {color:purple;}
h2 {color:purple;}
h3 {color:purple;}
p {color:purple;}
```

Se puede indicar de la siguiente forma:

```
h1, h2, h3, p {color:purple;}
```

Las declaraciones también se pueden agrupar:

```
h1 {color:purple;}
h1 {font: Helvetica;}
h1 {background-color:Aqua;}
```

Queda de la siguiente forma:

```
h1 {
  color:purple;
  font: Helvetica;
  background-color:Aqua;
}
```

El atributo `class` nos permite asignar estilos al documento sin preocuparnos de los elementos concretos a los que se aplicará.

```
<p class="tipo1">Esto es un párrafo tipo1</p>
```

Donde previamente habremos declarado el estilo:

```
.tipo1 {
    font-weight: bold;
    color: red;
}
```

El punto inicial es necesario en la declaración del tipo. No debe utilizarse en la declaración del elemento.

El atributo `id` permite identificar, de manera única, a un elemento dentro de un documento html. En la declaración de tipo va precedido de `#`. Ejemplo:

```
<p id="tipo1">Esto es un párrafo tipo1</p>
```

Donde previamente habremos declarado el estilo:

```
#tipo1 {
    font-weight: bold;
    color: red;
}
```

Elementos anidados Si un elemento está dentro de otro "hereda" los atributos del primero:

```
<style>
    h1 { color: red;}
</style>
<body>
<h1>Todos los elementos de H1 <i> deben </i> ser de color rojo</h1>
</body>
```

En este caso "**deben**", que se encuentra entre el elemento `<i>` y `</i>` también aparece en rojo. Podemos, sin embargo definir el elemento `i` de manera distinta cuando se encuentre dentro de un elemento `h1`:

```
<style>
    h1 {color:red;}
    h1 i {color:blue;}
</style>
```

hará que, en el ejemplo anterior, "**deben**" aparezca en azul.

Los colores se pueden expresar igual que se ha visto en xhtml, o utilizar la función `rgb` con los parámetros rojo, verde y azul entre paréntesis:

```
color:rgb(255,255,0)
```

Unidades de medida: podemos encontrar:

absolutas:

**in** : pulgadas

**px**: píxel

**pt**: puntos ( 0.25 in = 18 pt )

**pc** :picas ( 1 pc = 12 pt )

relativas:

**em**: se corresponde al tamaño de la fuente para el párrafo en concreto. Ejemplo, si el tamaño en puntos de los elementos del párrafo es de 14, con 2 em se mostraría en 28 pt.

Comentarios en hojas externas Se pueden definir comentarios de forma similar a C++, es decir, se considerará un comentario cualquier sección del documento que comience con */\** y termine con *\*/*.

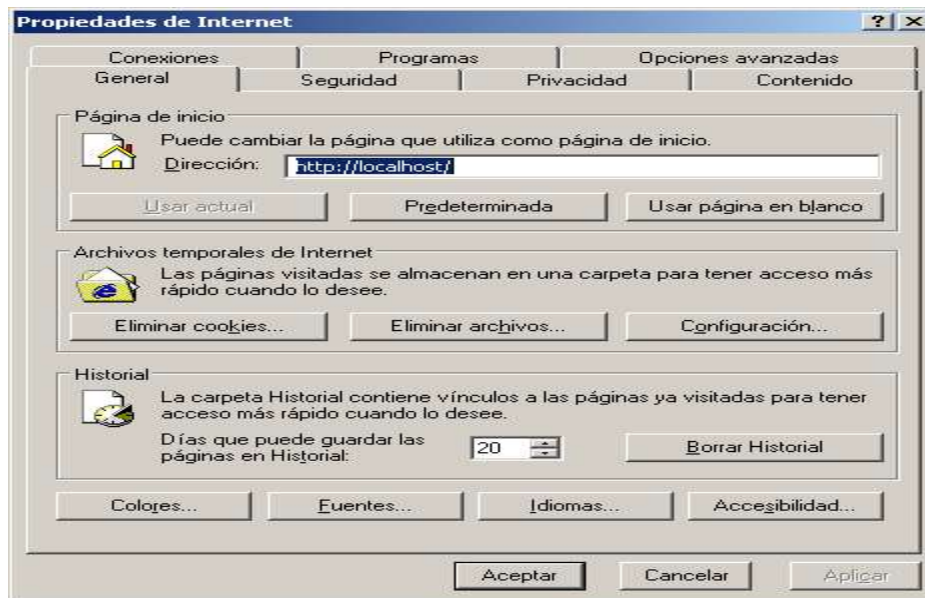
---

## **Propiedades en cascada**

Las CSS se llaman así por disponer de cuatro niveles diferentes en los que podemos aplicar estilos:

1. Un elemento puede disponer su propio estilo utilizando el atributo STYLE.
2. Una página puede contener un estilo global con el elemento `<STYLE>..</STYLE>` en la sección HEAD.
3. Los estilos pueden encontrarse en una hoja externa vinculada a la página.
4. El usuario puede especificar su propia hoja de estilo en el navegador.

En caso de conflicto se aplicará primero el nivel uno, después el dos, el tres y, por último, el cuatro.

**Hoja de estilo en el navegador:** (En Internet Explorer):

### Atributo de estilo

```
<h1 style="color:magenta;font-family:Helvetica;">Ejemplo en magenta y Helvetica</h1>
```

Las diferentes propiedades se separan con ;.

### Atributos a nivel de página

Se incluye un elemento style dentro de la sección head. Afectará a todo el documento.

```
<head>
  <style>
    h1 {
      color:magenta;
      font-family:Helvetica;
    }
  </style>
</body>
  <h1>Ejemplo en magenta y helvetica</h1>
</body>
```

## Hojas de estilo externas

Son archivos externos que siguen la misma sintaxis que las hojas de estilo incrustadas en la etiqueta STYLE.

Son archivos con extensión css.

Ejemplo, archivo primercss.css:

```
H1 {  
    font-size: 10px;  
    margin-left: 15px;  
}  
H2 {  
    font-size: 8px;  
    margin-left: 17px;  
}  
H3 {  
    font-size: 6px;  
    margin-left: 19px;  
}
```

Para vincular este archivo a nuestro documento HTML;

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

## Pseudoclases y pseudoelementos

Pseudoclases: son estilos especiales que ya proporciona HTML, como el hecho de que los enlaces se marquen y se subrayen es un ejemplo de estilo que ya existe incluso antes de ser especificado en nuestra hoja de estilo.

La forma de definir una pseudoclase

```
etiqueta:pseudoclase { regla de estilo }
```

Algunas pseudoclases:

- **:link** nos indica el estilo con el que debe aparecer el enlace antes de ser visitado.
- **:visited** nos indica el estilo con el que aparece un enlace visitado.
- **:active** nos indica el estilo con el que aparece el enlace en el momento de activarlo, o sea, al pulsar sobre él.
- **:hover** nos indica el estilo en el que aparece el estilo al pasar por encima el ratón.

Pseudoelementos: son elementos que no están reconocidos como parte del HTML estándar pero que el navegador va a ser capaz de identificar y, por lo tanto, proporcionarles un estilo determinado: La primera letra o la primera línea de un bloque de texto.

La forma de definir un pseudoelemento:

```
etiqueta:pseudoelemento { regla de estilo }
```

Algunos pseudoelementos:

- **:first-line** selecciona y aplica estilo a la primera línea de un bloque de texto. La cantidad de texto que compone la primera línea depende del tamaño y familia de la fuente, tamaño de la ventana, tamaño del bloque, existencia de objetos flotantes, etc.
- **:first-letter** selecciona la primera letra del texto de un bloque de texto para aplicarle determinados efectos tipográficos.

## **Propiedades de las hojas de estilo**

Vamos a dividir, para su estudio, las propiedades en 5 grandes grupos:

- Propiedades de texto
- Propiedades de las fuentes
- Propiedades de los colores y fondos
- Propiedades de márgenes y padding
- Propiedades de los bordes

Propiedades de texto: son propiedades que afectan a la presentación visual de caracteres, espacios, palabras y párrafos.

**Text-transform**: Especifica, por medio de palabras reservadas, si las letras del texto deben transformarse en mayúsculas, minúsculas, sólo la primera letra de cada palabra en mayúsculas, o si se de como está:

uppercase | lowercase | capitalize | none

**vertical-align**: Alineación vertical del texto en relación con la línea base del texto:

baseline | sub | super | top | text-top | middle | bottom | text-bottom | %

**text-align**: Fija la alineación del bloque al margen izquierdo, al margen derecho, centrado o a ambos.

**Text-indent**: fija la sangría o indentación de la primera línea del texto

xx unidad | %

**line-height**: indica la distancia entre dos líneas adyacentes.

Normal | xx unidad | %

**text-decoration**: fija una o más características “decorativas” del texto como subrayado, líneas superiores, caracteres tachados o parpadeantes

underline | overline | line-through | blink | none

**letter-spacing**: Especifica el espaciado entre letras.

Normal | xx unidad

**word-spacing**: especifica el espaciado entre palabras.

Propiedades de las fuentes: nos van a permitir controlar el tamaño, la familia, el grosor de la fuente. Algunas de las propiedades son:

**font-family:** indica la fuente a utilizar. Sus posibles valores son los nombres de las fuentes, pudiendo especificar un conjunto de familias. Se selecciona la primera que se encuentre en el sistema

**font-style:** indica si los caracteres irán en normal, en cursiva o en oblicua.

Normal | italic | oblique

**font-weight:** indica el grosor de la fuente dentro de una familia de fuentes. El valor 400 equivale a normal y 700 a bold

**font-size:** especifica el tamaño de la fuente. Puede ponerse como una longitud en unidades (absolutas o relativas), aunque también admite palabras reservadas que indican tamaños absolutos predefinidos ( xx-small, x-small, small, medium, large, x-large, xx-large)

Los valores larger y smaller especifican un tamaño en relación con los anteriores.

Propiedades de márgenes y padding: nos permite especificar los márgenes de cualquier elemento y con el padding controlamos la distancia entre el borde y el contenido.

**Margin-top**: fijamos el margen superiores: margin-top:1px

**Margin-bottom**: el margen inferior: margin-bottom: 1px

**margin-left**: el margen izquierdo

**margin-right**: margen derecho

**margin**: podemos especificar todos los valores de los márgenes de una sola vez

**padding-top**: distancia entre el borde superior y el contenido

**padding-bottom**: distancia entre el borde inferior y el contenido

**padding-left**: distancia entre el borde derecho y el contenido.

**Padding-right**: distancia entre el borde derecho y el contenido

**padding**: distancia entre el contenido y todos los bordes.

---

Propiedades de los bordes: especificamos el ancho, color y estilo del área del borde.

**Border-top-width, border-bottom-width, border-left-width, border-right-width** el ancho por encima, debajo, izquierda y derecha del elemento. Se puede especificar una longitud o las palabras reservadas: thin, medium y tick.

**Border-width:** especifica, en una sola propiedad, el ancho de todo el área de borde.

**Border-style:** especifica el estilo de visualización del borde. Los posibles estilos son:

- none,
- dotted (punteado),
- dashed(discontinuo),
- solid(liso),
- double(doble)
- groove(hundido),
- ridge(resaltado)
- inset(toda la caja hundida),
- outset(toda la caja resaltada).

**Border-color:** especifica el color del borde.

En **border-top**, **border-bottom**, **border-left** y **border-right** podemos especificar juntas las características de cada uno de los lados del borde: tamaño, estilo y color:

```
border-top: 1px, solid, #00ff00;
```

**border:** con esta propiedad podemos especificar de forma conjunta todas las propiedades de un borde: tamaño, estilo y color.