



CSS



CSS3

Las hojas de estilo en cascada es la traducción al castellano de CSS. Son un lenguaje que organiza los documentos estructurados en hojas de estilo. Sirven tanto para interpretar un documento en pantalla, un documento para impresora, un documento por voz o en dispositivos táctiles basados en Braille.

Fue desarrollada por el W3C (World Wide Web Consortium) y consigue separar los contenidos de los documentos escritos en HTML, XML, XHTML, SVG, o XUL. De esta forma se organizan en diferentes hojas de estilo, los colores, fondos, márgenes, bordes, tipos de letra.... Es muy útil para los desarrolladores de aplicaciones multiplataforma o en el desarrollo de aplicaciones web. Se puede modificar la apariencia de una página web de una forma rápida y sencilla, controlando el estilo y formato desde un mismo lugar aplicando a todo el documento.

CSS son las siglas de “Hojas de estilo en cascada” y que sirve para dar estilo a nuestra estructura creada en HTML. CSS3 es la última versión hasta la fecha

y presenta como principales características mayor control sobre el estilo de los elementos de nuestra página web y mayor número de efectos visuales. Las ventajas de usar CSS3 son:

- ✚ Más fácil de usar: El CSS permite guardar y almacenar los estilos por separado del contenido de una web, sea lo grande que sea. Imagínate un sitio web de más de 10.000 páginas, ¿cómo podrías cambiar los estilos en todas las páginas? Sin los archivos de CSS sería mucho más engorroso. Los CSS aportan un mejor control en la presentación de un sitio web que los elementos de HTML, por ejemplo.
- ✚ Más accesibles: Los CSS permiten modificar el formato de una página web según sean las necesidades del usuario. Por ejemplo, personas con deficiencias visuales.
- ✚ Más optimizados: El ancho de banda necesario para cargar páginas web con CSS es mucho menor que el resto. La hoja de estilo se almacena en cache y no tiene que descargar la página web una y otra vez. Los CSS, además, reducen más del 50% el ancho de banda y los costes de almacenamiento del sitio. Por lo tanto, estas páginas se cargan mucho más rápido.
- ✚ Multidispositivo: las hojas de estilo permiten configurar la página para diferentes dispositivos, como son móviles, tables, lectores de voz...

Evolución de CSS



Ilustración 1

Selectores de atributos en CSS3

En primer lugar encontramos 3 nuevos selectores de atributos:

Código :

```
elemento[atributo^="valor"]
```

Selecciona los elementos con ese atributo y que su valor comienza por la cadena de texto indicada en "valor".

Código :

```
elemento[atributo$="valor"]
```

Selecciona los elementos con ese atributo y que su valor termina por la cadena de texto indicada en "valor".

Código :

```
elemento[atributo*="valor"]
```

Selecciona los elementos con ese atributo y que su valor contiene la cadena de texto indicada en "valor".

Código :

```
/* Selecciona todos los enlaces que lleven a una página que contenga la palabra ejemplo* /
```

```
a[href*="ejemplo"]{...}
```

```
apunten a una dirección de correo*/
```

```
a[href^="mailto:"]{...}
```

```
apuntan a páginas .php*/ a[href$=".php"]{...}
```

Recordando un poco los selectores en CSS: son las etiquetas que identifican un estilo concreto. Por ejemplo

```
.micaja{
    background: #fff;
    type: Arial;
}
```

Describe una clase que en la página web se usa como

```
<div class="micaja">contenido </div>
```

En CSS3 aparecen tres nuevos selectores que aumentan considerablemente las posibilidades para controlar cuando y donde aplicar un determinado estilo:

`elemento[atributo^="valor"]`, selecciona todos los elementos que disponen de ese atributo y cuyo valor comienza exactamente por la cadena de texto indicada.

```
a[href^="https://"]{color:red}
```

Todos los enlaces a páginas que comienzan con https estarán en rojo

`elemento[atributo$="valor"]`, selecciona todos los elementos que disponen de ese atributo y cuyo valor termina exactamente por la cadena de texto indicada.

```
img[src$=".gif"] { border=2}
```

Todas aquellas imágenes tipo gif llevarán un borde de 2 px

`elemento[atributo*="valor"]`, selecciona todos los elementos que disponen de ese atributo y cuyo valor sea igual a la cadena de texto indicada.

```
p[title="importante"]{color:red; font-weight:bold;}
```

Todos los elemento p cuyo atributo title sea la palabra importante se colocan en color rojo y en negrita

`elemento[atributo~="valor"]`, selecciona todos los elementos que disponen de ese atributo y cuyo valor contiene la cadena de texto indicada, puede tener otras valores.

```
p[title="nota importante"]{color:red; font-weight:bold; font-size:18px}
```

Todos los elemento p cuyo atributo title contenga la palabra importante, puede contner otras, se colocan en color rojo, en negrita y con texto de 18px

Estos selectores amplían la forma de aplicar los estilos, ya no se aplican solo en función del tipo de elemento, del atributo id o class, podemos aplicar estilos en función del contenido.

Selector general de hermanos

El selector general de hermanos tiene un funcionamiento similar al selector adyacente de CSS2.1.

Mientras que en el adyacente la condición es que los elementos fuesen consecutivos en el código HTML, el selector general de hermanos tiene en consideración todos los elementos que sean consecutivos aunque no lo sean en el código HTML.

Código :

```
/*Selector adyacente en CSS2.1*/ h1 + h2{...}
/*Selector general de hermanos de CSS3*/
<h1>Título</h1>
<
<h1>Título</h1>
<p> párrafo de separación</p>
<h2>Subtítulo con selector general de hermanos</h2>
```

Existen elementos del lenguaje HTML que disponen de estados especiales asociados, eso se traduce en modificadores de estilos basados en estos estados. El mejor ejemplo para entender esto es el elemento `<a href...>` usado para los enlaces, este elemento puede estar en cuatro estados:

normal: Su estado normal, nunca se ha clickado en él.

visited: Cuando ya hemos visitado el link al que hace referencia

hover: Cuando tenemos el cursor situado encima del mismo

active: Cuando hacemos click sobre él

Así podemos usar las reglas de estilo del elemento `a` y sus cuatro estados y esto lo hacemos con las llamadas pseudo-clases o pseudo-selectores

CSS3 introduce nuevos modificadores o pseudo-clases que permite un control muy detallado sobre el aspecto de la página. Veamos los en detalle

`:root`

Selecciona el elemento padre de todo el documento, en las páginas web es el elemento HTML

`:first-of-type`

Primer elemento dentro de un contenedor. Así

`p:first-of-type` indica el primer párrafo dentro de un elemento padre (bloque `div`, `body`, `celda...`)

`:last-of-type`

Último elemento dentro de un contenedor. Así

`p:last-of-type` se refiere al último párrafo dentro de un elemento padre (bloque `div`, `body`, `celda...`)

`:only-of-type`

Selecciona al único elemento hijo de este tipo dentro de un contenedor

`p:only-of-type` si el elemento padre contiene un sólo elemento tipo párrafo esta regla se aplica a ese párrafo

`:only-child`

Este selector se refiere al único hijo dentro de un contenedor padre. Así

`p:only-child` selecciona un párrafo si es el único hijo del elemento padre.

`:first-child`, `:last-child`

Selecciona el primer hijo de un contenedor. En la segunda forma selecciona el último hijo

`p:first-child` se aplica al primer párrafo que sea el primer hijo dentro del elemento padre

`:nth-child(n)`, `:nth-last-child(n)`,

Selecciona el hijo de orden n (1º, 2º...). En la segunda forma empieza a contar por el final

`p:nth-child(2)` se aplica al párrafo que sea el segundo hijo dentro del elemento padre

`:nth-of-type(n)` , `nth-last-of-type(n)`

Selecciona el elemento de orden n (1°, 2°...). En la segunda forma empieza a contar por el final

`p:nth-child(2)` se aplica al segundo párrafo dentro del elemento padre, puede ser tercer hijo o cuarto...

`:empty`

Selecciona los elementos del tipo seleccionado que no tienen hijos

`div:empty` se aplicaría a los bloques DIV que no poseen hijos, es decir vacíos.

`:target`

Selecciona los destinos de los enlaces internos de una página, o sea, el destino dentro de la página al que se refieren los enlaces internos (`< a href="#cap1">Capítulo 1<a>`).

`:enabled`, `:disabled`, `:checked`

Selecciona elementos de formularios habilitados (`enabled`), deshabilitados (`disabled`) o marcados (`checked`)

`:not(selector)`

Selecciona los elementos que no son del tipo al que se aplica

`:not(p)` selecciona los elementos que no son párrafos

`::selection`

Se usa para referirse a lo seleccionado por el usuario en la página web

Bordes redondeados

Con CSS3 pueden crearse bordes redondeados, o usar una imagen para formar esos bordes o incluso crearlos con efecto de sombra.

A diferencia de CSS2 ya no es necesario andar buscando imágenes y componiéndolas para crear cajas con esquinas redondeadas. Para ello se usa la propiedad `border-radius`

Un ejemplo sencillo

```
div
{
border:2px solid #a1a1a1;
padding:40px;
background:#dddddd;
width:300px;
border-radius:25px;
}
```

Nos daría un cuadro como este

Un cuadro con esquinas redondeadas, fácil.

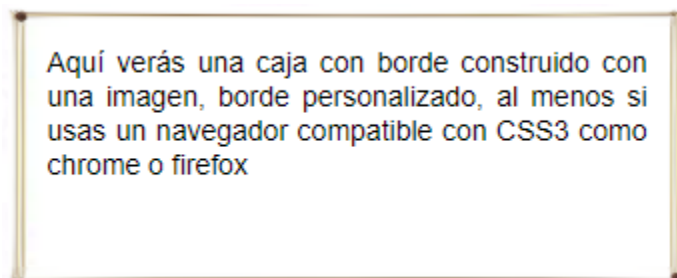
Bordes con imágenes

¿Aburrido de las opciones de bordes ofrecida por CSS2? Pues Vamos ahora hay otra posibilidad que nos soportada por Safari 3.1+, FireFox 3.5+ y Chrome 1.0+: se pueden usar imágenes en los bordes de elementos de nuestra web en lugar del borde normal.

Un ejemplo es éste código

```
#imgborde {
width: 300px;
height: 200px;
padding: 10px;
border-width: 10px;
border-style: solid; /* necesario para firefox */
-webkit-border-image: url(borde1.gif) 25% stretch stretch;
-moz-border-image: url(borde1.gif) 25% stretch stretch;
}
```


borde1.gif es la imagen que se usa para dibujar el borde la caja o elemento DIV como el siguiente



Si estás en Explorer verás un borde negro.

Como ves basta con darle la dirección de una imagen en el atributo **url**. Esta imagen contendrá cuatro lados, es decir es un marco y esos lados se usarán como partida

Los atributos que se usan son la **url** (dirección) de la imagen seguido de 4 valores para indicar la porción de cada lado del cuadro usado como imagen base para cada borde: superior, inferior, izquierdo o derecho. Si se usa un mismo porcentaje o valor para todos se puede poner una sola cifra.

Por último los lados horizontal y vertical puede simplemente usarse ajustando el tamaño de cada lado(**stretch**) para formar el cuadro completo, repitiendo cada lado como un tile (**repeat**) o repitiendo para ajustar los laterales y los bordes superior e inferior (**round**) y formar el cuadro completo.



Esta imagen tiene los patrones para formar cualquier caja con os bordes formados por imágenes. En tamaño decimos cuando de cada lado vamos a usar, en el modo indicamos si deformamos cada figura para formar el lado o la repetimos.

Cajas con sombras

Además de los bordes, podemos construir cuadros contenedores de imágenes o de texto con efecto sombra. Se usa el tributo `box-shadow` pero volvemos al problema de compatibilidades, por ello es necesario usar los prefijos para cada tipo de navegador.

Un ejemplo es éste código

```
#cajasombra{
background-color: #ddd;
width: 300px;
padding: 10px;
box-shadow: 5px 5px 3px #333;
-webkit-box-shadow: 5px 5px 3px #333;
-moz-box-shadow: 5px 5px 3px #333;
}
```

En total este nuevo efecto (`box-shadow`) necesita 4 atributos:

Desplazamiento horizontal de la sombra respecto a la caja

Desplazamiento vertical de la sombra respecto a la caja

Difuminado de la sombra, 0 indica ningún difuminado

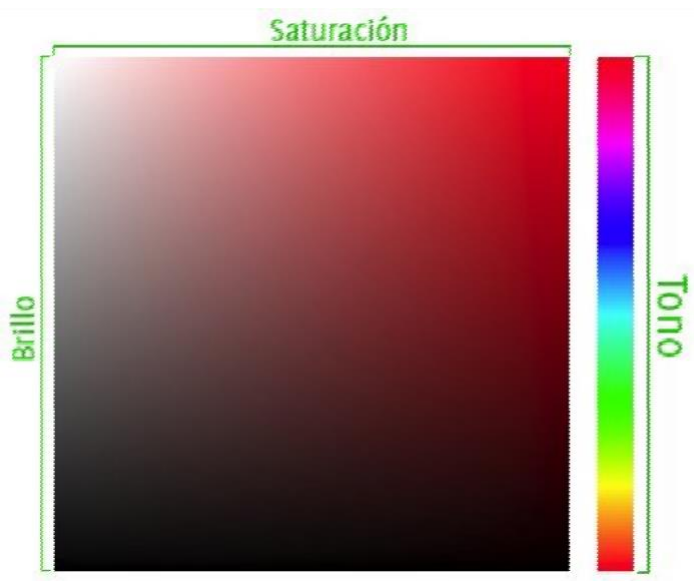
Por último el color de la sombra

este cuadro representa una caja con sombra
abajo a la derecha

Colores HSL y HSLA

CSS3 añade además un nuevo modelo de color conocido como HSL. Estas siglas provienen del inglés Hue, Saturation, Lighness o lo que es lo mismo, tono, saturación y brillo.

El modelo hsl es tal vez el más intuitivo de los modelos de color digitales



La sintaxis es la siguiente: Código :

```
/*atributo: hsl(tono,saturación,brillo);*/ background-color:
hsl(360,100%,20%);
```

Nótese que el valor tono puede tomar valores del 0 al 360 donde:

- 0, sería el rojo.
- 120, sería el verde.
- 240, sería el azul.
- 360, volvería a ser rojo.

Además, como en el caso del modelo RGB, al HSL se le puede añadir un canal alpha para definir la

transparencia del color. El resultado en código sería el siguiente, dando resultados análogos a RGBA





Código :

```
hsla(300,130%,65%,10%);
```

Tema Trausiciones.

Actualmente CSS3 permite la posibilidad de cambiar el estilo de un elemento mediante una transición animada. Este es tal vez el aspecto que me ha resultado más atractivo de todo CSS3 y es curioso que cuando redacté este mismo curso en Cristallab no escribí nada acerca de ellas. Sin embargo es una de las características con mayor potencial y más efectista de cuantas hayamos revisado hasta ahora.

Existen 4 propiedades base para crear transiciones:

-  `transition-property`: Define la propiedad que vamos a animar mediante la transición.
-  `transition-duration`: La duración en segundos de la animación.
-  `transition-timing-function`: La función matemática aplicada a la animación. Puede tomar 5 valores predefinidos (linear, ease, ease-in, ease-out o ease-in-out) o cubic-bezier(v, v, v, v) para definir tus propias funciones. Su valor por defecto es ease.
-  `transition-delay`: Indica el retardo en segundos. El valor por defecto es 0.

También se puede aplicar el método shorthand: Independientemente del orden:

`transition: [propiedad] [duración] [función] [retardo]`

Por ejemplo Código:

```
div{
0.2s;
0.2s;
width: 100px;
transition: width 1s ease-in 0.2s;
-o-transition: width 1s ease-in
-moz-transition: width 1s ease-in
```

```
-webkit-transition: width 1s ease-in  
0.2s;
```

```
div:hover{  
width: 500px;  
transition: width 1s ease-in 0.2s;  
-o-transition: width 1s ease-in  
0.2s;  
0.2s;  
-moz-transition: width 1s ease-in  
-webkit-transition: width 1s ease-in  
0.2s;  
}
```

En este ejemplo modificamos el ancho de una capa durante un segundo empezando 0.2 segundos después de hacer hover sobre el div.

El caso es que podemos aplicar transiciones a varias propiedades a la vez, simplemente separando las sentencias por comas:

Código:

```
transition: width 1s, height 1s;
```

En el caso de que fuesen todas las propiedades las que queramos cambiar mediante un efecto de transición, podemos indicar como variable propiedad el valor all que las engloba todas.

```
transition: all 1s ease-out;
```




Referencias Bibliográficas

<http://xlb.es/CursoCSS3.pdf>

<https://creatuweb.espaciolatino.com/tutorCSS3/bordesombra.htm><https://help.github.com/es>

<https://www.w3schools.com/css/>

