



Truquitos de CSS3

Estilo del navegador

Aparte de la hoja de estilos que aplica el diseñador de la página, existen otras dos hojas de estilo iniciales. Éstas aparecen de forma implícita al cargarse la página. Es el estilo que se aplica por defecto en caso de que el diseñador no haya aplicado ningún estilo.

En primer lugar está la hoja de estilos del navegador, o dispositivo en el que se muestra la página. Los navegadores aplican por defecto unos estilos básicos al cargar la página (tipo de letra, alineación, márgenes, etc.).

Estos estilos pueden que no sean los mismos en todos los navegadores, y a veces son la causa de que una página no se vea igual en todos los navegadores.

Estilo de usuario.

En segundo lugar está el estilo del usuario. Cada usuario puede crear una hoja de estilos para aplicarla a todas las páginas que visite con su navegador. Esta opción suele ser útil para personas discapacitadas, para adaptar los contenidos a su discapacidad (mayor contraste, letras más grandes, etc.).

Insertar una hoja de estilos de usuario suele hacerse de forma diferente según cual sea el navegador.

En Internet Explorer

Pulsa en el menú "Herramientas" y después en la opción "Opciones de Internet".

En la pestaña "General", en la parte de abajo, en la sección "Apariencia", pulsa sobre el botón "Accesibilidad".

En la nueva ventana que aparece, activa la opción "Formatear los documentos con mi hoja de estilos" y selecciona tu hoja de estilos de usuario pulsando sobre el botón "Examinar".

Pulsa Aceptar hasta volver al navegador.

En Mozilla Firefox

Guarda tu hoja de estilos en un archivo llamado "userContent.css"

Entra en el directorio de tu perfil de usuario de Firefox. En los sistemas operativos Windows se encuentra en "C: \ Documents and Settings \ [tu_usuario_de_windows] \ Datos de programa \ Mozilla \ Firefox \ Profiles \ [cadena_aleatoria_de_letras_y_numeros].default \ chrome.

Copia la hoja de estilos userContent.css en esa carpeta, remplazando la existente.

Reinicia el navegador para que se apliquen los cambios

En Opera

Pulsa sobre el menú "Herramientas" y después sobre la opción "Opciones"

Selecciona la pestaña Avanzado; en el menú de la izquierda pulsa en "contenidos"

Pulsa en el botón "opciones de estilo...".

Se abrirá una ventana. Debajo de donde pone "mi hoja de estilos" pulsa en seleccionar para seleccionar tu hoja de estilos de usuario.

Pulsa Aceptar hasta volver al navegador

En Safari

Pulsa sobre el menú "Editar" y después sobre la opción "Preferencias".

Selecciona la sección "Avanzado",

Pulsa sobre la lista desplegable llamada "Hoja de estilos" y selecciona la opción "Otra..."

En la ventana que aparece, selecciona tu hoja de estilos de usuario.

Prioridad en las hojas de estilo

Las hojas de estilo se aplican en el siguiente orden: Primero la hoja del navegador, segundo la hoja del usuario, y tercero la hoja del diseñador.

La última hoja aplicada es la que tiene mayor prioridad en caso de reglas contradictorias, por tanto la hoja del diseñador tiene más prioridad que la del usuario, y ésta más prioridad que la del navegador.

La palabra clave !important

La palabra !important (obsérvese que lleva el signo de admiración delante), colocada detrás de cualquier propiedad CSS, da prioridad a esa regla sobre las demás. Por ejemplo:

```
p { color: blue !important; }
```

```
p { color: red; }
```

En este ejemplo la primera regla anula a la segunda al llevar la declaración !important a pesar de que la segunda esté escrita después que la primera, lo que debería darle prioridad; por lo tanto los párrafos se verán en azul.

Esto no sólo funciona dentro de una misma hoja, sino que al aplicar diferentes hojas, la declaración !important tiene prioridad sobre las demás hojas.

En caso de que en un mismo estilo tengan la declaración !important la hoja del diseñador y la del usuario, tendrá prioridad la del usuario. Por tanto las hojas de estilo se aplican en el siguiente orden:

Prioridad de estilos

La declaración !important en la hoja del usuario tiene prioridad sobre todas las demás, en segundo lugar tiene prioridad la declaración !important del diseñador, en tercero la hoja del diseñador, en cuarto la hoja de estilos del usuario y en último la hoja del navegador.

Inicializar estilos

La hoja de estilos del navegador establece el estilo inicial por defecto a todos los elementos HTML: tamaños de letra, decoración del texto, márgenes, etc. Esta hoja de estilos siempre se aplica a todas las páginas web, por lo que si una página no incluye ninguna hoja de estilos propia, el aspecto con el que se muestra en el navegador es el de esta hoja de estilos.

El problema de las hojas de estilo de los navegadores es que los valores que aplican por defecto son diferentes en cada navegador, aunque suelen coincidir en algunos valores importantes.

Suelen coincidir por ejemplo en el tipo y color de letra (letra tipo serif, color negro), pero suelen variar en aspectos como valor de los márgenes, títulos de sección (etiquetas h1 ... h6), tabulación en las listas, etc. Esto hace que la misma página pueda verse diferente en distintos navegadores.

Para solucionar este problema debemos crear unos estilos iniciales. es decir, crear nosotros mismos los estilos por defecto, inicializar los estilos.

algunos sitios complejos o extensos utilizan su propia hoja para inicializar estilos, la cual se aplica la primera de todas. En otros basta con poner los estilos que se inicializan al principio de la hoja. Así por ejemplo poner al principio de la página :

```
* { margin: 0; padding: 0 ; }
```

nos asegura que todos los márgenes externos e internos se verán iguales en todos los navegadores.

Aparte de los márgenes también podemos definir otros elementos al principio: tipo, color y grosor de letra, etc. Normalmente se suele usar el selector universal (*) o el selector body para definir las características comunes a todo el sitio, e igualar las diferencias entre los distintos navegadores; y los selectores de etiqueta para dar un aspecto específico a las diferentes partes del contenido.

En un sitio complejo o extenso se suelen aplicar varias hojas de estilo. Debemos aplicar primero las de carácter más general, y en último lugar las más concretas. Así aplicaremos primero la hoja de inicialización, después la general del sitio, después la de la sección, y por último la de la página o páginas similares. Así nos aseguramos de que ningún estilo específico es sobrescrito por un estilo más general.

Uno de los problemas más habituales con los que se enfrentan los diseñadores al crear una página dividida en columnas es el de igualar la altura de todas ellas. Normalmente se indica el ancho de la columna, y el alto lo suele dar el contenido. Al acabar el contenido se acaba la columna, con lo que en algunas columnas queda un hueco debajo. Esto si las columnas llevan un color de fondo no queda bien, ya que se ve que una acaba antes que otra.

Con el diseño mediante tablas no ocurre este problema, ya que al ocupar cada columna una celda, éstas son siempre de la misma altura. Sin embargo el diseño de tablas es poco recomendable, y la gran mayoría de diseñadores no lo usan.

Hay varias soluciones para este problema, usando el código CSS. las cuales se verán a continuación:

En todas las soluciones propuestas partimos de un ejemplo de página con cabecera, pie de página y tres columnas creadas mediante "divs". El código HTML de la página es el siguiente:

```
<html>
<head>
<title> ... </title>
<body>
<div id="contenedor">
  <div id="cabecera"> ... </div>
  <div id="menu"> ... </div>
  <div id="cuerpo_principal"> ... </div>
  <div id="lateral_dcha"> ... </div>
  <div id="pie"> ... </div>
</div>
</body>

</html>
```


las columnas se definen mediante posicionamiento flotante en los "divs" menu, cuerpo_principal, y lateral_dcha. (propiedad float), El pie de página (div "pie") debe llevar la propiedad clear: both, para seguir el flujo normal de la página. El código CSS que define la estructura es el siguiente:

```
#contenedor { width: 100%; background-color: fuchsia;}
#cabecera { width: 100%; background-color: lime;}
#menu { width: 20%; float: left; background-color: silver; }
#cuerpo_principal { width: 60%; float: left; background-color: aqua; }
#lateral_dcha { width: 20%; float: left; background-color: yellow; }
#pie { clear:both; width: 100%; background-color: orange; }
```

Y la página tendrá un aspecto similar al del ejemplo;

Ejemplo página.

Ampliar márgenes inferiores.

La primera de las soluciones la publicó el diseñador Alex Robinson en su artículo Equal Height Columns - revisited y consiste en lo siguiente:

En primer lugar debemos aplicar la propiedad overflow: hidden; al elemento contenedor:

```
#contenedor { overflow: hidden; }
```

Después aplicamos un espacio de relleno inferior (padding-bottom) muy grande a todas las columnas y después un margen inferior negativo (margin-bottom) del mismo tamaño. El valor del espacio de relleno debe ser por lo menos tan grande como la columna más alta. Para evitar quedarse corto, se recomienda utilizar valores a partir de 10.000 píxeles:

```
#menu, #cuerpo_principal, #lateral_dcha { padding-bottom: 30000px;
margin-bottom: -30000px; }
```

Si tenemos posicionamiento flotante (propiedad float en la distribución de las columnas, todos los elementos que les sigan deben tenerlo también. Así al div "pie" le cambiaremos la propiedad clear: both por la propiedad float: left.

Las propiedades que hemos utilizado son las siguientes:

```
#contenedor { overflow: hidden; }

#menu,#cuerpo_principal,#lateral_dcha { padding-bottom: 30000px;
margin-bottom: -30000px; }

#pie { float: left }
```

Una vez aplicadas estas propiedades, la página de ejemplo anterior tendrá el siguiente aspecto:

Ampliar márgenes inferiores.

Los dos problemas principales que presenta esta solución son que se producen errores al imprimir la página, y que si utilizamos enlaces internos de tipo ancla (a otra parte de la misma página, las columnas se desplazan y desaparece parte de la página.

Sin embargo podemos hacer la página imprimible, creando un estilo para impresora, de forma que se anulen los colores de fondo, y las propiedades descritas anteriormente. En el ejemplo anterior añadiremos el siguiente código CSS:

```
@media print {
* { margin: 0; padding: 0; }
#contenedor { overflow:visible; }
#menu,#cuerpo_principal,#lateral_dcha { padding-bottom: 0px; margin-
bottom: 0px;
background-color: transparent; }
#pie { clear: both }
}
```

Con esto volvemos a convertir la página tal como estaba antes, y le hemos quitado el color a las columnas, con lo que la página podrá imprimirse, pero sin colores de fondo en las columnas.

El problema de los enlaces internos tipo ancla, no se le ha encontrado solución, por lo que si tenemos este tipo de enlaces en la página deberemos optar por otra solución.

Falsas Columnas

Esta otra solución la presentó el diseñador Dan Cederholm en su artículo Faux Columns. Si la solución anterior consistía en engañar al navegador, esta solución se basa en engañar al ojo del usuario.

La solución consiste en establecer una imagen de fondo repetida verticalmente en el elemento contenedor. La imagen debe imitar a las columnas que se formen, Como el contenedor es tan alto como la columna más alta, su imagen de fondo reemplaza a las otras columnas que se quedan más cortas. De esa manera da la sensación de que todas las columnas son de la misma altura.

El problema de esta solución es que la página tiene que ser de anchura fija, ya que la imagen de fondo es también fija, y siempre del mismo tamaño; no podemos por tanto usar porcentajes para definir el tamaño. Además debemos crear la imagen de fondo a medida de la página. Si queremos cambiar el fondo, deberemos cambiar también la imagen. En el ejemplo anterior deberemos variar el código CSS inicial que define la estructura de la siguiente manera:

```
#contenedor { width: 800px; background-image: url("fondo.gif"); margin: 0 auto; }
#cabecera { width: 800px; background-color: lime; }
#menu { width: 160px; float: left; background-color: silver; }
#cuerpo_principal { width: 480px; float: left; background-color: aqua; }
#lateral_dcha { width: 160px; float: left; background-color: yellow; }
#pie { clear: both; width: 800px; background-color: orange; }
```

Y la imagen de fondo utilizada, con sus medidas es la siguiente:

fondo de falsas columnas

La imagen, al repetirse verticalmente, crea unas columnas de fondo iguales a las de la página, con lo que al acabarse la columna, ésta continúa en la imagen de fondo.

El resultado lo podemos ver en el siguiente enlace:

Falsas columnas.

Uso de la propiedad display

La siguiente solución puede considerarse la única técnicamente correcta, ya que no se basa en ningún truco para engañar al ojo o al navegador, y fue propuesta por el diseñador Roger Johansson en su artículo *Equal height boxes with CSS*. y se basa en el uso de la propiedad `display`, de modo que se fuerza a los "divs" a comportarse como una tabla.

En primer lugar es necesario que las columnas estén dentro de un elemento contenedor, y a su vez éste dentro de otro elemento contenedor. La estructura de cajas en el código html será la siguiente:

```
<div id="contenedor">
  <div id="cabecera"> ... </div>

  <div id="contenidos1">
    <div id="contenidos2">
      <div id="menu"> ... </div>
      <div id="cuerpo_principal"> ... </div>
      <div id="lateral_dcha"> ... </div>
    </div>
  </div>

  <div id="pie"> ... </div>
</div>
```

A continuación, se utiliza la propiedad `display` para mostrar los "divs" añadidos como si fueran celdas de una tabla: El "div" más exterior llevará la propiedad `display: table`, el segundo llevará la propiedad `display: table-row` y los "divs" de las columnas llevarán la propiedad `display: table-cell`.

No deben aplicarse ninguna propiedad de posicionamiento a estos "divs", eliminaremos por tanto las propiedades `float` y `clear` y la propiedad `position` y otras de posicionamiento si las hubiera.

En el ejemplo anterior variaremos el código CSS inicial que define la estructura de la página de la siguiente manera:

```
#contenedor { width: 100%; background-color: fuchsia; }
#cabecera { width: 100%; background-color: lime; }
#menu { width: 20%; background-color: silver; }
#cuerpo_principal { width: 60%; background-color: aqua; }
#lateral_dcha { width: 20%; background-color: yellow; }
#pie { width: 100%; background-color: orange; }
```

```
#contenidos1 { display: table; }
#contenidos2 { display: table-row; }
#menu,#cuerpo_principal,#lateral_dcha { display: table-cell; }
```

La página la veremos de la siguiente manera:

Propiedad display.

Mediante la propiedad display, los elementos se comportan como una tabla, una fila de tabla o una celda de tabla, de forma que los elementos "div" que forman las columnas de la página en realidad se comportan como celdas de tabla, lo que permite que el navegador las muestre con la misma altura.

El único inconveniente es que Internet Explorer en versiones antiguas (version 7 y anteriores) no muestra bien este tipo de estructura, quedando la página desestructurada.

Nuevas tendencias en CSS3

El nuevo CSS3 añade nuevas propiedades, que permiten distribuir un texto en columnas, y además añade otro tipo de diseño para las páginas, el diseño de cajas. La distribución de un texto en columnas sirve para un texto dentro de una caja, pero no para varias cajas en sí mismas, mientras que en el diseño por cajas, se facilita la distribución de varias cajas en el espacio de la página, pero nada dice de la igualación de la altura entre ellas.

Podemos ver cómo funcionan estas nuevas propiedades en nuestro Manual de CSS3.

Propiedades de CSS 3

Uno de los diseños más demandados por los diseñadores de páginas es el poder poner en los elementos que configuran la página esquinas redondeadas. El estándar CSS 2.1 obliga a que la estructura de las cajas sea rectangular con las esquinas en ángulo de 90°, pero la nueva versión de CSS3 incluye la propiedad border-radius que permite establecer una curvatura para las esquinas.

En el valor de esta propiedad debe indicarse una medida obligatoriamente, y otra opcional. Si se indica sólo una medida indica que la esquina es circular, y la medida es el radio. Si se indican dos medidas la esquina es elíptica, y la primera medida es el radio horizontal de la elipse y la segunda el radio vertical.

La propiedad border-radius afecta a las cuatro esquinas del elemento seleccionado, pero también podemos aplicar el diseño redondeado a cada una de las esquinas mediante las propiedades border-top-right-radius (superior derecha), border-bottom-right-radius (inferior derecha), border-bottom-left-radius (inferior izquierda), border-top-left-radius (superior izquierda).

NOTA: A día de hoy los principales navegadores (incluido Internet Explorer 10) ya incluyen la propiedad border-radius sin necesidad de utilizar prefijos, por lo que lo que aquí se explica es para que se vean las esquinas redondeadas en navegadores antiguos. Esta era la solución antes de que apareciera la propiedad border-radius. la cual se explica en la página del Manual de CSS3 - Esquinas Redondeadas.

Imágenes de fondo

las soluciones más comunes se basan en utilizar imágenes de fondo. Si la caja es de tamaño fijo podemos usar una sola imagen para toda la caja. Si es de tamaño variable, debemos marcar en la caja varios "divs", incluyendo en cada uno una esquina.

Hay varios métodos para mostrar esquinas redondeadas con imágenes. a continuación vamos a mostrar uno de ellos:

Método para redondear las esquinas con imágenes.

código HTML

En primer lugar, dentro de la caja o div, donde queremos redondear las esquinas, crearemos 4 "divs", uno por cada esquina, Los dos de las esquinas superiores los colocaremos antes de los contenidos, y los dos de las esquinas inferiores, después de los contenidos: La estructura de la caja HTML será así:

```
<div class="caja">
  <div class="sup-izda"></div>
  <div class="sup-dcha"></div>

  <!-- .... aquí van los contenidos de la caja, parrafos,
  imágenes, listas, etc. -->

  <div class="inf-izda"></div>
  <div class="inf-dcha"></div>
</div>
```

En este código, dentro de la caja, cada esquina se corresponde con un elemento "div". Los dos primeros van delante del contenido, y corresponden a las esquinas superior izquierda y superior derecha; los dos segundos van detrás de los contenidos, y se corresponden con las esquinas inferior izquierda e inferior derecha.

La idea es simple, en cada esquina colocaremos un pequeño "div", el cual tendrá como imagen de fondo la esquina redondeada correspondiente.


Obtener las imágenes:

Para obtener las imágenes sólo necesitamos un programa de tratamiento de imágenes y el navegador Mozilla, Safari, o Google Chrome. Si no tienes ningún programa de tratamiento de imágenes puedes descargarte el programa gratuito Gimp 2.6.7, en <http://www.gimp.org/downloads>.

Una vez que tenemos el navegador y el programa de tratamiento de imágenes, vamos a crear las imágenes de las cuatro esquinas:

Escribiremos el código para ver la caja con esquinas redondeadas en cualquiera de los navegadores anteriores. En el código pondremos también los márgenes internos y bordes que queremos que tenga la caja.

```
.caja { width: 250px; border: 5px olive double; margin: 20px; padding: 20px; background-image: url("fondo.gif"); -moz-border-radius: 20px; -webkit-border-radius: 20px; border-radius: 20px; }
```



Referencias Bibliográficas

http://www.uco.es/~lr1maalm/css_avanzado.pdf

<https://www.lawebdelprogramador.com/pdf/10532-Guia-completa-de-CSS-3.html>

<http://es.tldp.org/Manuales-LuCAS/doc-curso-html/doc-curso-html.pdf>

