



MINISTERIO
DE EDUCACIÓN
Y CIENCIA

SECRETARÍA GENERAL
DE EDUCACIÓN
Y FORMACIÓN PROFESIONAL

DIRECCIÓN GENERAL
DE EDUCACIÓN,
FORMACIÓN PROFESIONAL
E INNOVACIÓN EDUCATIVA

CENTRO NACIONAL
DE INFORMACIÓN Y
COMUNICACIÓN EDUCATIVA

Edición HTML

Estilos CSS Técnicas de aplicación



SERVICIO DE
FORMACIÓN DEL
PROFESORADO

C/ TORRELAGUNA, 58
28027 - MADRID

Índice de contenido

Pies de imágenes.....	3
Escalar automáticamente imágenes.....	4
Crear imágenes de sustitución.....	5
Paso 1.....	5
Paso 2.....	5
Paso 3.....	6
Paso 4.....	6
Paso 5.....	6
Paso 6.....	8
Paso 7.....	8
Paso 8.....	8
Paso 9.....	9
Cajas con esquinas redondeadas.....	10
Centrando textos.....	11
Centrando bloques o imágenes.....	12
Centrar verticalmente.....	13
Deslizándose	14
"Loncheat" imágenes con Gimp.....	15
...y buscando alternativas.....	16
Encolumnar listas.....	18
Recuperar los indicadores de ítem.....	19
Encolumnar respetando el orden vertical.....	20
Sustitución de imágenes.....	22
Crear un periódico con texto encolumnado.....	23
Paso 1.....	24
Paso 2.....	24
Paso 3.....	24
Paso 4.....	25
Paso 5.....	25
Paso 6.....	26
Paso 7.....	27

PIES DE IMÁGENES

El lenguaje HTML no incluye ninguna etiqueta que permita introducir pies o títulos explicativos de las imágenes. Aunque en su momento se propuso en la especificación HTML3 finalmente no salió adelante y no se incluyó en la especificación HTML 4 vigente.

A pesar de la anterior limitación podemos conseguir introducir un título explicativo de la imagen y situarlo inmediatamente debajo de la misma utilizando CSS. El efecto sería similar al que se muestra en la zona de la derecha.

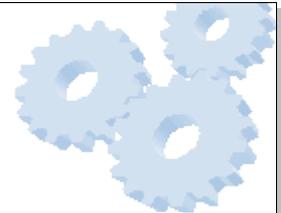
Para obtenerlo hemos tenido que crear un par de reglas CSS que se explican a continuación:

Lo primero ha sido crear una clase a la que hemos denominado imagen que flote a la derecha, tenga un margen izquierdo de 10 píxeles, un borde sólido de 1 píxel y un relleno de 10 píxeles y unas dimensiones para que el texto del pie que insertaremos posteriormente no desborde la anchura de la imagen.

Realmente lo único imprescindible, si pretendemos que la imagen deje fluir el texto sería indicar la propiedad `float` y la especificación de anchura por el motivo que se ha indicado en el párrafo anterior. El resto de los elementos tienen que ver más con cuestiones estéticas.



```
.imagen { float: right;
width:260px;
border: 1px solid rgb(153, 153, 153);
padding: 10px;
margin-left: 10px;
}
```



Ahora que ya tenemos definido el estilo creamos un contenedor genérico `div` en el que situaremos la imagen y el texto que deseamos que figure como pie de la misma. A este contenedor genérico le aplicaremos la clase cuyas propiedades acabamos de definir.

Si ahora insertamos dentro del contenedor un párrafo con el texto del pie, este asumirá las características que tengamos definidas por defecto, por lo que tendremos que crear un nuevo selector que afecte únicamente a los párrafos que se encuentren dentro del contenedor de clase imagen para el que definiremos las características que necesitamos: en nuestro ejemplo letra cursiva, tamaño 80% del definido por defecto y alineación centrada.

```
.imagen p { font-size: 80%;
font-style: italic;
text-align: center;
}
```



ESCALAR AUTOMÁTICAMENTE IMÁGENES

La técnica que acabamos de plantear puede ser un interesante recurso para crear un álbum de fotos porque nos permitiría crear una serie de miniaturas con título de todas las imágenes pero, tal como hemos planteado la técnica tendremos que realizar una tarea previa de redimensionado de todas las imágenes puesto que las que fueran mayores del tamaño especificado desbordarían los bordes del contenedor genérico.

Antes de continuar es importante señalar un principio básico que ya hemos apuntado en otros momentos, salvo en casos excepcionales es esencial realizar las modificaciones de tamaño para ajustar las miniaturas y ahorrar ancho de banda.

Hecha la salvedad anterior veamos una forma de hacer que las imágenes se ajusten automáticamente al espacio definido para el contenedor genérico. La idea es tan simple como establecer una clase que fije la anchura al 100% y aplicarla a la imagen. Al estar la imagen dentro del contenedor ésta será la referencia que se tome para calcular el tamaño. De hecho, la imagen del ejemplo tiene un tamaño real de 720 x 540 píxeles.



```
.escalada {width: 100%;}
```

Una cuestión importante es que, cuando insertamos una imagen en el documento, N|VU incluye el código con sus dimensiones y no hay que perder de vista que el código html o el estilo en línea tendrán preferencia sobre las propiedades establecidas en el estilo interno o externo. Para evitarlo y conseguir que el tamaño se ajuste automáticamente habrá que editar manualmente el código para eliminar las dimensiones.

95 Una de los componentes que puede incluir una página de centro es un álbum de fotos con imágenes de las instalaciones o los reportajes de actividades escolares.

Crea un álbum para mostrar imágenes con su correspondiente pie de foto. Para crear las reglas de estilo tendrás que considerar los siguientes aspectos:

- Anchura y altura de los contenedores genéricos.
- Fondo de los contenedores (ten en cuenta que si le adjudicas un color diferente al del fondo de la página puede funcionar visualmente como un passpartout)
- Existencia o no de borde.
- Tamaño de la fuente.
- Alineación del texto de pie

Por si quieres hacer la práctica sin buscar ahora mismo imágenes puedes utilizar las fotos de estas siete setas: **amanita muscaria**, **armillaria ostoyae**, **coprinus comatus**, **hypholoma del aliso**, **macrolepiota procera**, **pholiota penetrans** y **tricholomopsis rutilans**.

Como complemento puedes hacer que cada imagen apunte a un original en alta resolución (puedes obtenerlos en el **banco de imágenes** del CNICE)

96 Una propuesta complementaria para el álbum de imágenes: ¿qué tal quedaría cada contenedor si tuviera un fondo que imitara el carrete fotográfico?.

Gimp dispone de una utilidad que puede ayudarte para crear esta imagen. Puedes acceder a ella en el menú **SCRIPT-FU ➔ DECORACIÓN ➔ DIAPOSITIVA**

CREAR IMÁGENES DE SUSTITUCIÓN

Vamos a denominar imágenes de sustitución a aquellas que modifican su apariencia cuando el ratón pasa sobre ellas. Su utilidad principal será la creación de menús.

PASO 1

Crear una página cuyo prólogo recoja correctamente el tipo de documento. (Recuerda que N|VU introduce en prólogo correcto con la referreia completa cuando se crean documentos XHTML, tanto con DTD transicional como strict, pero que cuando se crean documentos HTML el prólogo es incorrecto porque falta la referencia absoluta a la DTD). Cuando la definición del tipo de documento no es correcta algunos navegadores pueden adoptar comportamientos erráticos.

Creemos una lista con los elementos que se mostrarán en el menú y creamos los enlaces (para nuestro ejemplo serán nulos):

- **Coprinus comatus**
- **Hypholoma del aliso**
- **Pholiota penetrans**
- **Amanita muscaria**
- **Macrolepiota procera**
- **Armillaria ostoyae**
- **Thricolomopsis rutilans**

Como es posible que finalmente deseemos situar nuestro menú en una zona de la página vamos a encerrarlo dentro de una división a la que le aplicaremos el identificador `id="menu"`

PASO 2

Crear el juego de imágenes que se utilizará para los efectos. Necesitaremos una imagen para el mostrar en la página y otra que se mostrará únicamente cuando el ratón pase sobre ella.

Aunque podrían utilizarse tamaños diferentes para ambas imágenes es preferible que sean del mismo tamaño para que todo sea más sencillo.

Para el ejemplo hemos seleccionado una colección de fotos de setas procedentes del banco de imágenes del CNICE.





PASO 3

Ajustaremos márgenes y rellenos para que la lista quede al borde de la división y eliminaremos los boliches del listado que hay dentro de nuestra división menú, para lo que recurriremos a crear un estilo interno utilizando el editor CSS.

Nueva regla de estilo

Crear un nuevo: estilo con nombre (introduzca abajo el nombre de la clase) estilo aplicado a todos los elementos del tipo (introduzca el tipo abajo) estilo aplicado a todos los elementos coincidentes con el siguiente selector

#menu ul

Crear regla de estilo

En la pestaña **CAJA** pondremos a 0 los valores de **MÁRGENES Y RELLENOS** izquierdos y en la pestaña **LISTAS** elegiremos **NONE** como **TIPO PARA ESTILO DE LA LISTA**, con lo que nuestra lista aparecerá ahora así.

Coprinus comatus
Hypholoma del aliso
Pholiota penetrans
Amanita muscaria
Macrolepiota procera
Armillaria ostoyae
Thricolompsis rutilans

PASO 4

Ahora haremos que nuestra lista se presente en horizontal, con cada ítem a continuación del anterior. Para ello crearemos en el editor CSS una nueva regla `#menu li` que afectará únicamente a los elementos de lista que se encuentren dentro de la división menú y le adjudicaremos en la pestaña **caja** el valor `left` a la propiedad `float`

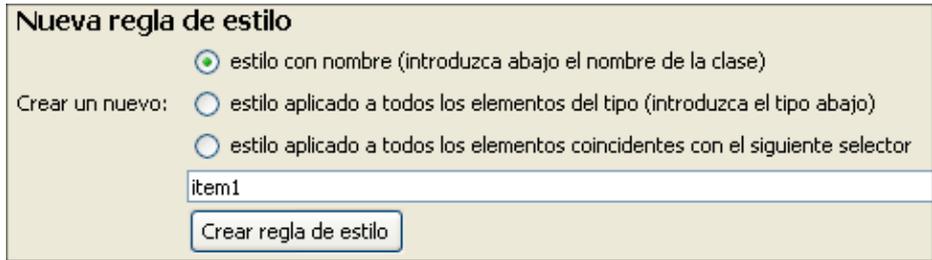
Coprinus comatusHypholoma del alisoPholiota penetransAmanita muscariaMacrolepiota proceraArmillaria ostoyaeThricolompsis rutilans

Como puede verse la apariencia es ahora bastante confusa al haberse colocado todos los ítems correlativamente, pero ya lo corregiremos más adelante.

PASO 5

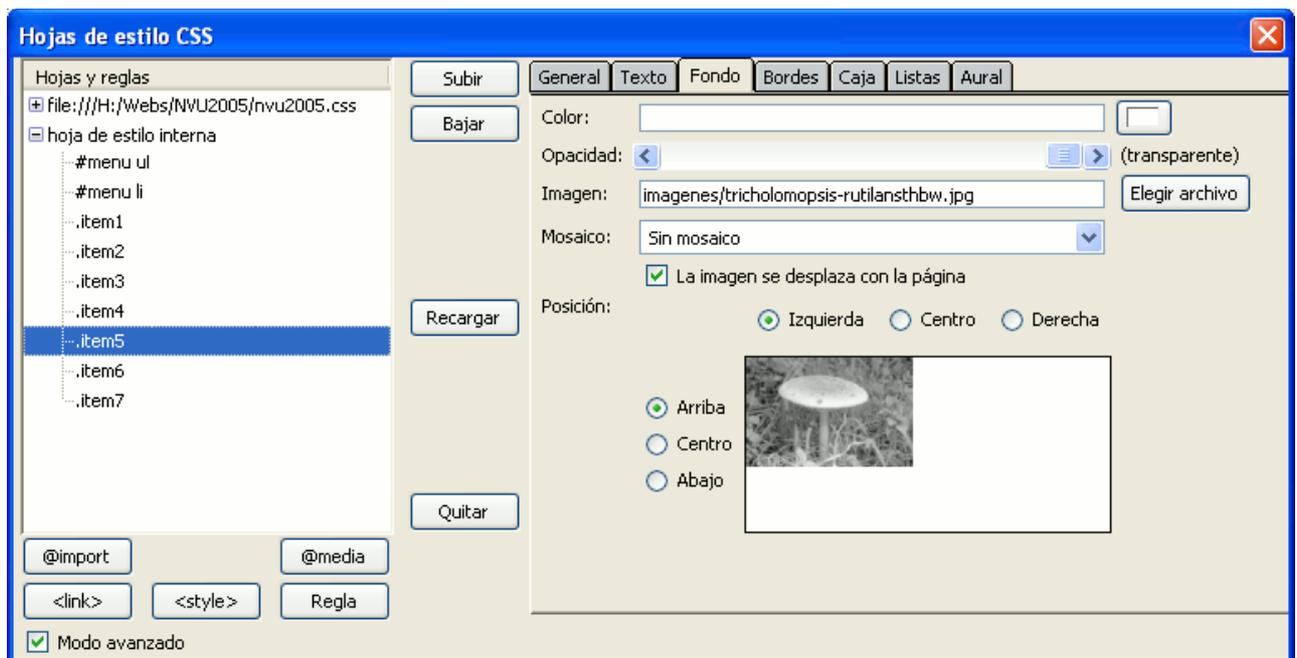
Colocaremos ahora una imagen como fondo de cada uno de los enlaces.

Para poder modificarlos posteriormente vamos a tener que crear una clase para cada uno de ellos. Así pues iremos creando una serie de nuevas reglas a las que iremos llamando item1, item2 y así sucesivamente. En este caso estamos creando clases en lugar de identificadores id, por lo que utilizaremos la opción estilo con nombre al crear cada una de las nuevas reglas. Es importante recordar que, aunque cuando vemos el código las clases se identifican porque se inician con un punto, no debemos ponerlo al adjudicarles el nombre, ya que será el programa el que se encargue de esta tarea. (De hecho si intentamos nombrar una clase empezando por un punto el editor no la creará).



Una vez que hayamos creado todas las clases las adjudicaremos a cada uno de los elementos de la lista. Para ello lo más cómodo será utilizar las referencias de la barra de estado, ir seleccionando cada elemento y pulsando con el botón derecho para seleccionar clases y adjudicarle la que le corresponde por su orden.

Cuando hayamos realizado esta tarea podemos completar el trabajo abriendo el selector y aplicando un fondo para cada uno de las clases tal como se muestra en la siguiente imagen.



Si vas a publicar la página utilizando el gestor de N|VU no habrá problemas puesto que él se encargará de convertir las rutas absolutas para las imágenes de fondo en relativas, pero la recomendación es que utilices la pestaña  Código fuente para modificarlo manualmente.

Lo que habremos conseguido hasta el momento es lo que se muestra a continuación y, la verdad es que no parece muy alentador. Pero no te preocupes porque ya hemos avanzado mucho y la situación variará bastante con las siguientes modificaciones.

[Coprinus comatus](#) [Hypholoma del aliso](#) [Pholiota penetrans](#) [Amanita muscaria](#) [Macrolepiota procera](#)
[Armillaria ostbyae](#) [Tricholomopsis rutilans](#)

PASO 6

Vamos ahora a darle un estilo a los enlaces de nuestro menú, para lo que crearemos una nueva regla `#menu` a que afectará a los enlaces que se hayan creado dentro de nuestro menú. (Hemos estado utilizando este identificador desde el principio pero aún no habíamos definido ninguna regla con sus propiedades)

Aplicaremos las siguientes propiedades y valores en la pestaña caja:

- Anchura= la de la imagen (en nuestro ejemplo 100px)
- Altura= 0, lo cual es muy posible que te sorprenda. El motivo es que no deseamos que se muestre el texto y por eso le adjudicamos este valor con lo que logramos que no se vea en un navegador visual, pero seguirá estando el texto para un navegador no visual que lea el contenido.
- Relleno superior= altura de la imagen. Aquí está el truco que compensa la altura 0, ya que hacemos que se reserve un espacio para el relleno.
- Overflow = `hidden`. Si no lo hiciéramos así, no nos habría servido de nada el truco de la altura 0, ya que el contenido que "se sale" del espacio se mostraría, mientras que ahora queda oculto.

Ahora hemos conseguido un cambio bastante importante del menú.



PASO 7

Como las imágenes han quedado juntas vamos a separarlas modificando la regla `#menu li` que ya habíamos creado añadiéndole en la pestaña caja un margen derecho de 1 píxel.



PASO 8

La siguiente tarea será establecer el comportamiento de cada elemento cuando el ratón pase por encima. Necesitaremos crear una seudoclase `:hover` para cada uno de los elementos de la lista y adjudicarle como fondo la imagen en color correspondiente en cada caso. Además, para garantizar que siempre se colocará por encima de la imagen en blanco y negro le daremos también, en la ventana caja, un valor 50 para la propiedad z-index (realmente sería suficiente darle un valor 2 ya que con ese estaría por encima del valor de profundidad por defecto).

Posiblemente sea más rápido en este caso acudir a la pestaña  y añadir manualmente las reglas, copiando y pegando la regla original, añadiéndole la seudoclase `:hover` al final y modificando el nombre de la imagen que, si hemos actuado como es habitual se llamará igual que la de blanco y negro pero añadiendo o quitando unos caracteres que serán iguales para todas las imágenes. (en el ejemplo las imágenes en blanco y negro finalizaban su nombre con las letras bw y las de color se llamaban igual pero sin esas letras)



PASO 9

Por último, solo nos resta corregir la flotación. Dado que los elementos de la lista tienen establecida la propiedad de flotación a la izquierda podemos encontrarnos con que el flujo de texto se produzca a la derecha de nuestra lista y, normalmente, nos interesará que no sea así y que la lista quede separada del resto. Para ello corregiremos la el contenedor general al que habíamos llamado `menu` dándole, en la pestaña caja las dimensiones de alto y ancho que deseemos adjudicarle en función de nuestras imágenes y especificando el valor `block` para la propiedad `display`.



En el ejemplo faltan los enlaces reales a las páginas de destino que suponemos que estarían creados desde el primer momento.

El código de nuestra hoja de estilos interna quedaría así:

```
<style type="text/css">
<!--
#menu ul { list-style-type: none;
margin-left: 0pt;
padding-left: 0pt;}

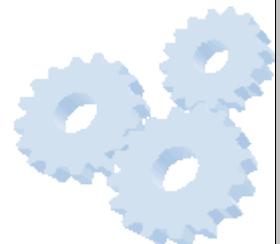
#menu li { float: left;}

.item1 { background-repeat: no-repeat;
background-image: url(imagenes/coprinus-comatusthbw.jpg) ;}
.item1:hover { background-repeat: no-repeat;
background-image: url(imagenes/coprinus-comatusth.jpg) ;
z-index:50;}

... resto de ítems de la lista
#menu a { display: block;
width: 100px;
padding-top: 66px;
overflow: hidden;
height: 0px;}

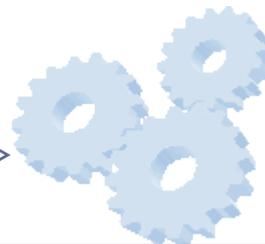
#menu li { float: left; margin-right:1px;}

#menu {display:block; width: 710px; height: 66px;}
-->
</style>
```



Y el código HTML en el cuerpo del documento sería:

```
<div id="menu">
<ul>
<li class="item1"><a href="#">Coprinus comatus</a></li>
<li class="item2"><a href="#">Hypholoma del aliso</a></li>
<li class="item3"><a href="#">Pholiota penetrans</a></li>
<li class="item4"><a href="#">Amanita muscaria</a></li>
<li class="item5"><a href="#">Macrolepiota procera</a></li>
<li class="item6"><a href="#">Armillaria ostoyae</a></li>
<li class="item7"><a href="#">Thricolompsis rutilans</a></li>
</ul>
</div>
```



CAJAS CON ESQUINAS REDONDEADAS

Cuatro esquinas

Con ese termino nos referimos a los muros que forman una casa.

Con cuatro "ladrillos" especiales podemos hacer que las uniones de los muros de la casa sean redondeadas.

El primer paso sería crear un contenedor general (en el ejemplo se ha denominado "bordesredondos") en el que situaremos los elementos que deseamos poner en el recuadro redondeado y asignarle las dimensiones, los márgenes y el color de fondo.

Dentro del contenedor general situaremos tres "filas" de elementos:

1. El borde superior con las esquinas redondeadas: utilizamos un contenedor genérico div que lleva como fondo la esquina izquierda y anidamos dentro del mismo otro contenedor genérico div que tiene como fondo la esquina superior derecha. Si en lugar de anidar los contenedores genéricos los hubiéramos puesto uno a continuación de otro, cada uno de ellos ocuparía el 100% de la anchura del contenedor general y se colocarían uno bajo el otro.
2. El contenido del recuadro.
3. El borde inferior y sus esquinas redondeadas: seguimos el mismo procedimiento que para el borde superior.

La imágenes que hemos utilizado para las equinas son las siguientes: . También se ha utilizado una imagen  para el fondo general de la caja aunque, dado que se trata de un color plano, podríamos haber optado por asignarle un color de fondo en lugar de utilizar una imagen.

El código CSS en la cabecera o en un archivo externo.

```
/* ancho y márgenes de la caja */
.bordesredondos { width: 20em; margin: auto; }
.bordesredondos { background: url(imagenes/fondo.gif) repeat; }

/* imágenes de fondo para las esquinas */
.arribaizq { background: url(imagenes/arriba_izq.gif) no-repeat top left; }
.arribader { background: url(imagenes/arriba_der.gif) no-repeat top right; }
.abajoizq { background: url(imagenes/abajo_izq.gif) no-repeat bottom left; }
.abajoder { background: url(imagenes/abajo_der.gif) no-repeat bottom
right; }

/* ancho y alto de las esquinas */
.arribaizq, .arribader, .abajoizq, .abajoder { width: 100%; height: 1em; }

/* desplazamiento de las esquinas inferiores para disminuir la distancia al
contenido */
.abajoizq, .abajoder { margin-top:-1em; }

/* detalles del contenido */
.contenidocaja { margin: 0 1em; }
```

En caso de que el código CSS vaya en un archivo externo habrá que ajustar las rutas de las imágenes tomando como punto de partida el archivo CSS que es el que contiene la llamada.

El código HTML en el cuerpo de la página.

```
<div class="bordesredondos">
<div class="arribaizq"><div class="arribader"></div></div>
<div class="contenidocaja">
Aquí va el texto que aparece en la caja
</div>
<div class="abajoizq"><div class="abajoder"></div></div>
</div>
```

CENTRANDO TEXTOS

Está claro que la utilización más inmediata del centrado es la de fragmentos de texto, creando reglas para que se aplique esta alineación a todos los elementos de un tipo, a una clase determinada o creando un estilo en línea para aplicarla a un único elemento en un momento puntual.

Ejemplos:

Este fragmento de título h5 está centrado porque lo hemos especificado en un estilo incrustado

```
<h5 style="text-align: center;">Este fragmento de título h5 está centrado porque lo hemos especificado en un estilo incrustado</h5>
```

A este fragmento se le aplicó clase "centro" definida como estilo en línea en la cabecera del documento

```
<style type="text/css"> .centro { text-align: center;} </style>
```

...

```
<p class="centro">A este fragmento se le aplico la clase "centro" definida como estilo en la cabecera del documento</p>
```

CENTRANDO BLOQUES O IMÁGENES

En ocasiones lo que necesitamos centrar no es el texto, sino el bloque que lo contiene. La forma de hacerlo es establecer los márgenes izquierdo y derecho como "auto", de forma que el visor calcule el espacio sobrante y lo divida a partes iguales entre ambas zonas. Lógicamente, como lo habitual será que el bloque tenga suficiente contenido como para ocupar la totalidad de la anchura disponible, habrá que aplicar esta técnica a bloques con una anchura fija.

Soneto a una nariz (Francisco de Quevedo)

Érase un hombre a una nariz pegado,
érase una nariz superlativa,
érase una nariz sayón y escriba,
érase un peje espada muy barbado
era un reloj de sol mal encarado,
érase una alquitara pensativa,
érase un elefante boca arriba,
era Ovidio Nasón más narizado.
Érase el espolón de una galera,
érase una pirámide de Egipto,
las doce tribus de narices era;
érase un naricismo infinito,
muchísimo nariz, nariz tan fiera
que en la cara de Anás fuera delito.

Para mostrar el poema hemos creado una clase para el estilo interno de la cabecera que luego le aplicamos al contenedor genérico en el que hemos encerrado el soneto.

```
.poema { padding: 0.5em 1em 1em;  
width: 25em;  
margin-left: auto;  
margin-right: auto;  
background-color: rgb(236, 225, 187);  
}
```

Aunque habrás visto el bloque del soneto centrado, tanto con Internet Explorer como con Mozilla, Firefox u Ópera, ha sido porque todas las páginas de este manual llevan aplicado un parche genérico que corrige la mayoría de los errores de interpretación de CSS2 que tiene Internet Explorer.

Si no queremos aplicar el parche completo tendríamos que aplicar la solución provisional que consiste en encerrar el bloque que deseamos centrar dentro de otro contenedor genérico al que le aplicamos una regla de estilo para que alinee el texto centrado: con ello se consigue que el bloque se sitúe en el centro. Puedes ver el mismo [ejemplo adaptado](#) y revisar el código para comprobar lo que estamos comentando.

Aprovechando la idea del centrado de bloques podemos aplicarla también para centrar imágenes, aunque en este caso hay que tener en cuenta una matización: las imágenes son elementos "en línea" y como tales no generan un salto anterior y posterior. Ya hemos visto en

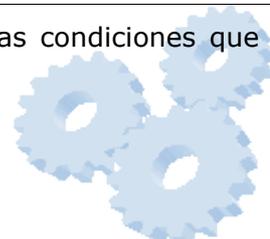
otros momentos que podemos considerarlas como una "letra" más dentro de la cadena de caracteres que componen un texto. Eso quiere decir que, para poder hacer que se centren tenemos que convertirlas en elementos de tipo "bloque". En este caso no necesitamos hacer ninguna maniobra especial para que se sitúen correctamente en Internet Explorer.

La siguiente imagen salta y se centra automáticamente aunque está incluida en la página inmediateamente después del punto con el que finaliza esta frase.



El código que hemos utilizado se ha limitado a crear una clase con las condiciones que habíamos reseñado previamente y aplicarla luego a la etiqueta `img`.

```
.imagenbloque { display: block;
margin-left: auto;
margin-right: auto;
}
```



Importante: si intentas aplicar esta técnica en un documento HTML creado por N|VU sin corregir el error que introduce en la generación de los prólogos pasarás un mal rato copiando y pegando códigos y viendo que Internet Explorer no te hace ni caso. Es uno de los ejemplos de funcionamiento errático de un navegador cuando el código es incorrecto. Puedes consultar el [error](#) y localizar las [plantillas](#) que lo solucionan.

CENTRAR VERTICALMENTE

No disponemos en CSS2 de una propiedad para centrar verticalmente pero, a pesar de ello, podemos combinar algunas propiedades existentes para lograrlo.

La idea en la que nos vamos a basar es que el contenido de las celdas de las tablas sí que admite el centrado vertical. Aprovechando esta particularidad crearemos un contenedor externo que se muestre como si fuera una tabla de celda y le aplicaremos el centrado vertical.

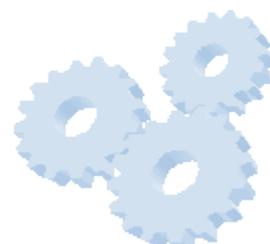
este es el texto que se debería centrar en vertical

El ejemplo que se muestra en la zona izquierda, a pesar de ser una aplicación de una sugerencia propuesta desde el W3C, sólo se reproduce correctamente cuando la página se visualiza con navegadores basados en el motor Gecko (Mozilla, Firefox, Galeón, Safari, etc) pero

no se respetan las propiedades si se visualiza con Internet Explorer u Opera.

Las propiedades que se han utilizado para construir la clase "centrarvertical" son las siguientes:

```
.centrarvertical { border: 1px solid rgb(153, 102, 51);
min-height: 100px;
width: 400px;
display: table-cell;
vertical-align: middle;
float: left;
margin-right: 1em;
color: rgb(255, 255, 204);
background-color: rgb(192, 192, 192);
font-size: x-large;
font-weight: bold;
}
```



DESLIZÁNDOSE ...



Bueno pues parece que vamos a empezar a hacer alguna cosa que creíamos que no se podía hacer con las páginas HTML.

Por lo que sabíamos hasta el momento, las imágenes forman un rectángulo que nos permite que el texto fluya por la izquierda o por la derecha del mismo.

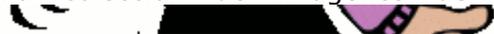
Esto parece bastante lógico cuando se trata de fotografías, ya que estamos acostumbrados a percibir las como un todo en el que la figura y el fondo componen un paralelogramo. Sin embargo la situación varía mucho cuando la imagen que vemos es un dibujo, denominado clipart en el lenguaje informático en el que nos encontramos unos contornos que lo separan claramente del fondo y le hacen adquirir una personalidad propia. Este es el caso de nuestra valiente patinadora, que se ha venido deslizando por la red desde su origen en la librería **Open Clipart** que es una galería de imágenes distribuidas bajo licencia Creative Commons y a la que puedes acceder para descargar trabajos de otros internautas o aportar el tuyo si tienes una buena mano.

El caso es que, correspondiendo a sus esfuerzos también hemos hecho lo posible para aprender a deslizarlos y para ello hemos contado con la inestimable ayuda de uno de los grandes expertos en aplicación de CSS: Eric Meyer, en cuya **web** encontrarás el artículo "ragged float" que explica el principio que se ha aplicado para crear este ejemplo.

¡Uf, menos mal! Aunque creo que han aparecido un par de referencias interesantes en los párrafos anteriores, ya estaba empezando a pensar que iba a ser incapaz de encontrar tema con el que ir rellenando el espacio para que pudieras comprobar el efecto de deslizamiento del texto siguiendo el perfil de una imagen. Como estoy escribiendo con una resolución de 1024x768 y puede que esta página se muestre dentro de una estructura de marcos, es seguro que ya habremos superado a nuestra amiga la patinadora, así que vamos a ver un par de detalles sobre la técnica empleada.

La idea en la que se apoya esta técnica es que la imagen flote y el texto fluya a su alrededor. Pero como lo que no queremos es que la imagen sea un rectángulo tenemos que hacer un par de trabajos para lograr el efecto deseado:

1. Empezamos por utilizar un programa de tratamiento de imagen para dividir la imagen en "lonchas" cuya anchura vaya ajustándose al contorno con lo que dispondremos de una colección de imágenes de diferentes anchos:



En el artículo de Eric Meyer se hace con todas las imágenes tengan la misma altura, sin embargo en nuestra patinadora hemos utilizado alturas diferentes haciendo fragmentos más altos cuando el borde derecho se encontraba a la misma altura como ocurre con la cabeza.

2. Colocamos las imágenes una a continuación de otra y hacemos, utilizando CSS que se coloquen alineadas una bajo otra y flotando a la izquierda.

A partir de aquí no hace falta más que empezar a escribir a continuación de la última imagen. Obviamente, si nuestra imagen estuviera insertada en otro punto de la página podríamos tener texto por encima, a su lado y por debajo según dicha posición.

Así pues, toda la técnica se basa en la siguiente regla CSS aplicada a los elementos `img`:

```
.patina { margin: 0 1em 0 0;  
float: left;  
clear: left;  
}
```

El único margen es el de la derecha, la flotación es a la izquierda y, para evitar que las imágenes aparezcan una a continuación de otra se ha añadido la propiedad `clear:left`, de manera que nada pueda flotar a la izquierda de cada imagen.

El artículo original de Eric Meyer incluía la propiedad `height` con un valor en píxeles ya que todas las imágenes tenían la misma altura. Como en nuestro caso no era posible probé primero a poner la altura con un valor "auto" y, finalmente opté por suprimir la propiedad, dado que las imágenes llevan ya la indicación de altura y ha funcionado correctamente en Internet Explorer 6, Mozilla 1.7.8, Firefox 1.04 y Opera 7.54 para Windows que han sido los navegadores sobre los que he probado la página.

Otro pequeño truco ha sido añadir la regla `p{position: relative; z-index:50;}` al descubrir que, en Internet Explorer las imágenes podían superponerse al texto en algunos momentos. Al incluir un índice de profundidad para los párrafos que son los elementos que bordean las imágenes se consigue evitar esta situación que no afectaba a los otros navegadores.

"LONCHEAR" IMÁGENES CON GIMP

Explicamos brevemente un método de trabajo para fragmentar una imagen en tiras utilizando Gimp:

1. Con la imagen original abierta utiliza el menú **IMAGEN** ➔ **CONFIGURAR LA REJILLA**. En la sección **ESPACIADO** desliga el ancho y el alto pulsando sobre el icono  y adjudícale los valores que consideres más adecuados a ambos parámetros. (15 píxeles para el alto y 5 para el ancho pueden ser una buena opción)
2. Tras aceptar acude al menú **VER** y marca las opciones **VER REJILLA** y **AJUSTAR A LA REJILLA**: de esta forma tendrás a la vista unos puntos que te permitirán ir seleccionando las zonas más acordes a las características de la imagen sin tener que preocuparte de "casar" una zona con la siguiente gracias a la opción de **AJUSTAR A LA REJILLA** que acabas de activar.
3. Selecciona la primera zona utilizando la herramienta de **SELECCIÓN RECTANGULAR**. Es **muy importante** que compruebes que están **desactivadas** las opciones **ALISADO** y **DIFUMINAR LOS BORDES**, ya que pretendemos que los cortes sean secos para que no se noten al volver a ensamblar las piezas.
4. Copia la primera selección y utiliza **EDITAR** ➔ **PEGAR COMO NUEVO** para crear la primera "loncha". Guárdala.
5. Vuelve a la imagen original y continúa repitiendo el procedimiento hasta que tengas todos los fragmentos.
6. No estaría de más que desactivaras la opción **AJUSTAR A LA REJILLA** porque si lo olvidas puede que en trabajos futuros pases un mal rato cuando intentes seleccionar una zona y el programa no te haga mucho caso hasta que no recuerdes que sigue activado. No he mencionado la de **VER REJILLA** porque esa es evidente y en cuanto te moleste su presencia te acordarás de quitarla.

...Y BUSCANDO ALTERNATIVAS

Al inicio de la página hemos visto una imagen situada en la esquina superior izquierda con el texto flotando a su alrededor y es posible que estés pensando que ya habías captado la idea y lo de repetirlo a la derecha parece poner en duda tu capacidad.

¿O acaso la cuestión no sería tan sencilla como modificar las propiedades float y clear poniéndoles el valor right para que todo quede igual pero al lado contrario?.

Bueno, vayan pues las disculpas si has podido molestarte porque está claro que no es esa la intención, así que vamos a aclarar el motivo de esta repetición, porque ¿no te ha parecido que lo de andar recortando las imágenes es un poco pesado? ¿será que hay otro método para conseguir el mismo efecto utilizando la imagen completa y que sean las reglas CSS las que se encarguen de hacer el reparto de los espacios?.

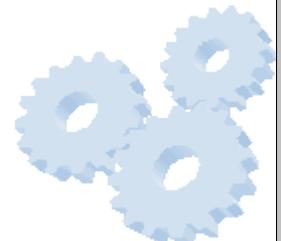


Efectivamente, de eso se trata. Hemos vuelto a utilizar una imagen de **Open Clipart** pero en esta ocasión no hemos necesitado partirla, sino que hemos recurrido a una vieja idea que ya utilizamos como recurso de posicionamiento cuando trabajamos exclusivamente con código HTML sin usar estilos. ¿Recuerdas el truco de la imagen transparente que podíamos usar para que ocupara un espacio y de esa forma desplazar otros elementos? Pues bien, la idea original en este caso es muy similar aunque algo más respetuosa con las cuestiones de accesibilidad ya que no tenemos que situar ningún elemento sin significado en la página.

Vamos pues con el desarrollo de la idea mostrando el código y los comentarios :

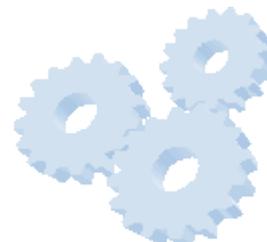
Nuestras reglas de estilo:

```
.contenedor { margin: 0; padding: 0;
background-repeat: no-repeat;
background-image: url(imagenes/arquitectoder.png);
background-position: right top;
background-color: rgb(255, 255, 255);
}
#a, #b, #c, #d, #e, #f, #g, #h, #i, #j, #k, #l, #m {
float: right;
clear: right;
margin: 0 0 0 1em;
padding:0;
height: 20px;
font-size:0px; /* necesario para que el tamaño de las zonas no se
redimensione si cambiamos el tamaño de fuente en el navegador */
}
#a {width:90px}
#b {width:100px}
#c {width:190px}
#d {width:190px}
#e {width:160px}
#f {width:140px}
#g {width:110px}
#h {width:100px}
#i {width:130px}
#j {width:70px}
```



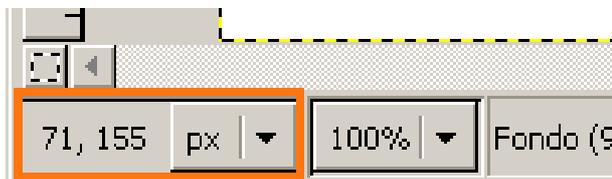
Y el código HTML dentro del cuerpo de la página:

```
<div class="contenedor">
<span id="a"></span>
<span id="b"></span>
<span id="c"></span>
<span id="d"></span>
<span id="e"></span>
<span id="f"></span>
<span id="g"></span>
<span id="h"></span>
<span id="i"></span>
<span id="j"></span>
<span id="k"></span>
<span id="l"></span>
<p> Aquí empieza el texto ..... hasta finalizar</p>
</div>
```



Se podría haber utilizado `div` en lugar de `span`, pero si la página se viera con un navegador que no soportara CSS se mostrarían tantos renglones vacíos como etiquetas `div` hubiéramos puesto. Al utilizar etiquetas `span` vacías no se mostraría nada. Es importante incluir en la regla para creación de los identificadores que el tamaño de la fuente sea 0px para evitar la redimensión.

Utilizando esta técnica no tendremos que modificar las imágenes, pero sí que tendremos que hacer algunas pruebas para poder ajustar exactamente las medidas de cada zona, para lo que necesitaremos recurrir a alguna utilidad de tratamiento de imagen que nos informe de las coordenadas.



En el caso de Gimp puedes seguir un procedimiento idéntico al anterior en cuanto a establecimiento y visualización de la rejilla. Pasando el puntero por la imagen podremos ir comprobando las coordenadas x,y en la zona izquierda de la barra de estado y así utilizaremos la información para establecer la anchura más adecuada..

Ten en cuenta que la caja de cada renglón empieza en su borde superior. Por este motivo verás que en algún momento el texto se superpone a la imagen ya que está tomando como referencia la zona en blanco superior que es la que realmente coincidía con el inicio del renglón. Es posible que esto nos obligue realizar un ajuste fino que puede pasar por modificar la altura de las zonas reservadas o la anchura de la zona previa. De todas maneras ten en cuenta que los ajustes no pueden ser perfectos cuando el visitante modifique el tamaño de la fuente, por lo que, dependiendo del tipo de imagen, tal vez tengas que hacer el ajuste menos fino para garantizar que el texto no se superponga a la imagen.

97

- Pulsa sobre la miniatura para guardar la imagen de esta esquina decorativa en tu



carpeta.

- Utiliza la técnica que prefieras de las dos explicadas para colocar la imagen en la zona superior izquierda de la pantalla y alinear el texto para que siga su contorno.
- Ten en cuenta que, para que el efecto quede completamente bien tendrás que ajustar correctamente los márgenes y rellenos.

ENCOLUMNAR LISTAS

En ocasiones podemos encontrarnos con listas que contienen un considerable número de ítems y que resulta poco apropiado presentar en una única columna.

Hemos utilizado como ejemplo la lista de pueblos que componen la comarca de la cuenca del Tajuña en la provincia de Madrid que son los quince que figuran a continuación.

- | | | |
|-------------------------|---------------------|-------------------------|
| • Ambite | • Arganda del Rey | • Campo Real |
| • Carabaña | • Morata de Tajuña | • Nuevo Baztán |
| • Olmeda de las Fuentes | • Orusco | • Perales de Tajuña |
| • Pozuelo del Rey | • Rivas Vaciamadrid | • San Martín de la Vega |
| • Tielmes | • Valdilecha | • Villar de Olmo |

Si nuestra intención es utilizar esta lista como punto de partida para realizar una descripción geográfica de los pueblos de la comarca parece evidente que es necesario aprovechar el ancho de la ventana. Aparte de que pueda resultar estéticamente más agradable se trata de favorecer al visitante, ya que le ahorraremos tener que recurrir a la barra de desplazamiento que sería casi inevitable con una lista de este tamaño.

Además, tal vez sería interesante eliminar los indicadores de cada ítem. Si estás viendo esta página con Internet Explorer o con Opera es posible que te haya sorprendido lo de eliminar los indicadores de ítem... porque no los estás viendo. Al parecer, cuando se utilizan anchuras para indicar el ancho de una lista estos navegadores prescinden de los indicadores, así que vamos a hacer dos cosas: primero veremos cómo hay que hacerlo para que desaparezcan los indicadores, esperando que los navegadores se ajusten en el futuro a los estándares y luego volvemos al mundo real e intentamos recuperar nuestros "bolos" para que se puedan ver en todos los navegadores.

Pongamos manos a la obra:

Como tenemos quince ítems vamos a hacer que se agrupen de cinco en cinco, esto es en tres columnas. Para ello le adjudicaremos a la lista una anchura de 14em para cada ítem, con un total de 42em para la lista. (Este cálculo es empírico: en función de la longitud de los ítems estimamos la anchura necesaria y la corregimos si nos hemos desviado mucho). Si fijamos los márgenes y rellenos de los elementos de la lista a 0 nos vendrá muy bien para poder ajustar la estimación de anchuras.

Aunque parezca un detalle nimio es importante que se especifique el ancho en ambos selectores para que el navegador pueda calcular cuántos ítems le cabrán en cada línea.

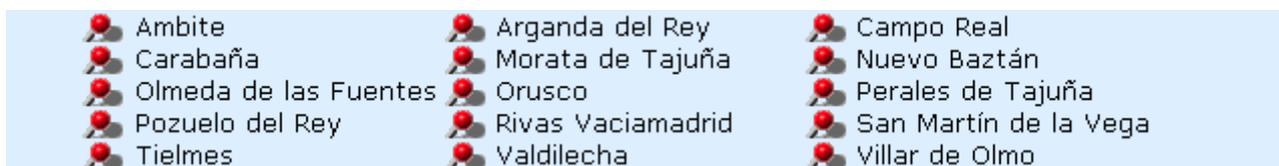
El tercer factor a tener en cuenta es que los ítems floten a la izquierda, de forma que vayan dejando hueco a su derecha para que se pueda colocar el ítem siguiente.

Tras aplicar estos criterios nuestro código CSS ha quedado así:

```
ul {width: 44em;
margin: 0 1.5em;
padding: 0;
}
ul li { float: left;
width: 14em;
margin: 0 ;
padding: 0;
}
```

Como se está utilizando un estilo en línea y en esta página va a aparecer también una lista ordenada no podemos definir sencillamente las propiedades del selector li ya que se aplicarían también a los ítems de la lista ordenada. Por ese motivo hemos definido la regla para los elementos li que sean hijos de un elemento ul.

RECUPERAR LOS INDICADORES DE ÍTEM



Ya hemos comentado antes que Internet Explorer y Opera no mostraban los indicadores de ítem cuando la lista tenía establecida una anchura. Esto incluye no sólo los indicadores por defecto, sino también las imágenes gráficas. Es posible que hayas pensado que lo que habíamos hecho era sustituir los indicadores por imágenes, pero no es así: ha habido que complicar un poco la situación buscando una imagen para ponerla como fondo de cada ítem y calcular los desplazamientos para que no se montara con el texto.

La explicación real de esto se basa en que, cada elemento de lista hace que el navegador genere dos cajas: una para el indicador y otra para el texto del ítem. Internet Explorer y Opera solo generan la caja correspondiente al texto cuando se encuentran una lista con anchura especificada. Debido a esto, para obtener un efecto similar al que se muestra habría que trabajar adjudicando una imagen de fondo y separándola del texto para que no este no se superpusiera. Además no habría que olvidarse de quitar los indicadores para que no aparecieran en los navegadores basados en el motor Gecko de Mozilla que interpretan correctamente la situación. Como ves todo bastante complicado. En el siguiente recuadro aparece el código y los comentarios.

```
li {float: left;
width: 12em; /* habrá que ajustar la anchura descontando el relleno que
antes no existía */
margin: 0 ;
list-style-type:none; /*eliminamos el indicador para que no se muestre en
navegadores basados en Mozilla */
background-image: url(imagenes/Rd_pin.gif);
background-position: 0;
background-repeat: no-repeat;
padding: 0 0 0 2em; /*añadimos un relleno en la zona izquierda para hacer
hueco para la imagen */
}
```

ENCOLUMNAR RESPETANDO EL ORDEN VERTICAL

En los ejemplos anteriores hemos utilizado listas desordenadas. Al menos así las hemos presentado, pero el contenido de las mismas estaba realmente ordenado alfabéticamente.

Al mostrarlas en columnas se ha seguido manteniendo el orden pero, probablemente, si los textos de cada ítem no hubieran sido tan largos, la tendencia natural de lectura habría sido agrupar visualmente las columnas y suponer que el orden de lectura es primero de arriba a abajo en una columna para luego saltar a la siguiente.

Para acentuar este fenómeno perceptivo vamos a mostrar ahora el mismo contenido ordenado de esta manera.

- | | | |
|---------------------|--------------------------|---------------------------|
| 1. Ambite | 6. Nuevo Baztán | 11. Rivas Vaciamadrid |
| 2. Arganda del Rey | 7. Olmeda de las Fuentes | 12. San Martín de la Vega |
| 3. Campo Real | 8. Orusco | 13. Tielmes |
| 4. Carabaña | 9. Perales de Tajuña | 14. Valdilecha |
| 5. Morata de Tajuña | 10. Pozuelo del Rey | 15. Villar de Olmo |

¿Cuál de las dos formas de ordenar te ha parecido más sencilla de leer? Está claro que cada persona tiene un esquema perceptivo propio y es posible que a tí te resulte más fácil leer con el orden alfabético en horizontal, pero lo más probable será que te haya resultado más cómoda la opción que acabamos de ver. Por si acaso piensas que influyen demasiado los números, vamos a mostrarla sin numeración pero respetando el orden alfabético para que te cerciores.

Ambite	Nuevo Baztán	Rivas Vaciamadrid
Arganda del Rey	Olmeda de las Fuentes	San Martín de la Vega
Campo Real	Orusco	Tielmes
Carabaña	Perales de Tajuña	Valdilecha
Morata de Tajuña	Pozuelo del Rey	Villar de Olmo

En este caso los ítems de la lista no pueden flotar, porque eso provocaría una ordenación horizontal como ya hemos visto.

Sin embargo necesitamos que se coloquen en columnas. Para ello crearemos tantas clases como columnas necesitemos y especificaremos en ellas el margen izquierdo de forma que al aplicárselas a cada ítem hagamos que estos se sitúen escalonadamente.

También vamos a adjudicarle una altura a la línea. Enseguida veremos para qué nos sirve.

En estos momentos nuestro código presentaría el siguiente estado:

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

/*ahora especificamos ol li para que la regla afecte sólo a los items de
listas ordenadas */
ol li { margin: 0; padding: 0; line-height: 1.3em; }

/*creamos las diferentes columnas. No son exactamente iguales debido a la
anchura heterogénea del contenido de los ítems */
.col1 {margin-left: 1em;}
.col2 {margin-left: 15em;}
.col3 {margin-left: 31em;}
```

Y con eso conseguiríamos... este pequeño desastre

1. Ambite
2. Arganda del Rey
3. Campo Real
4. Carabaña
5. Morata de Tajuña
6. Nuevo Baztán
7. Olmeda de las Fuentes
8. Orusco
9. Perales de Tajuña
10. Pozuelo del Rey
11. Rivas Vaciamadrid
12. San Martín de la Vega
13. Tielmes
14. Valdilecha
15. Villar de Olmo

Es verdad que el texto parece encolumnado, pero lo que habíamos dicho antes de que queríamos que las columnas se situaran escalonadamente ha resultado profético.

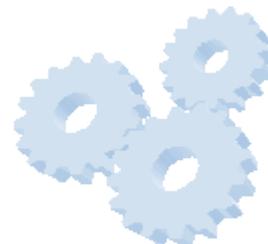
Ahora bien, si lo pensamos un poco ha ocurrido lo que era de esperar: el ítem 6 se ha colocado donde podía esperar, esto es, debajo del ítem 5 pero respetando el margen que le hemos indicado. El 7, 8, 9 y 10 se han ido colocando cada uno debajo del anterior con su margen correctamente calculado. Del 11 al 16 se ha producido el mismo proceso acorde a lo que habíamos indicado hasta ahora.

Está claro que nos falta introducir una pequeña corrección y aquí es donde aparece la utilidad de esa altura de línea que habíamos indicado como propiedad de los selectores li. Como cada ítem ha ido ocupando 1.3 em quiere decir que cada columna ha consumido $1.3 \times 5 = 6.5$ em verticales. Si creamos una clase que tenga un margen superior negativo de esa cantidad y la aplicamos a los ítems en los que se produce el salto de columna conseguiremos que suban hasta situarse a la altura del primer ítem. Los que vayan a continuación se irán alineando correctamente bajo él hasta que llegue el siguiente salto de columna. Esto se traduciría en la siguiente clase que añadimos a nuestros estilos:

```
.saltoarriba {margin-top: -6.5em;}
```

Con ello completaríamos las reglas necesarias y lo que haríamos dentro del código HTML sería aplicar un par de clases al mismo elemento cuando llegáramos al primero de cada columna, con lo cual conseguimos el efecto que perseguíamos.

```
<ol>
  <li class="col1">Ambite</li>
  <li class="col1">Arganda del Rey</li>
  <li class="col1">Campo Real</li>
  <li class="col1">Carabaña</li>
  <li class="col1">Morata de Tajuña</li>
  <li class="col2 saltoarriba">Nuevo Baztán</li>
  <li class="col2">Olmeda de las Fuentes</li>
  <li class="col2">Orusco</li>
  <li class="col2">Perales de Tajuña</li>
  <li class="col2">Pozuelo del Rey</li>
  <li class="col3 saltoarriba">Rivas Vaciamadrid</li>
  <li class="col3">San Martín de la Vega</li>
  <li class="col3">Tielmes</li>
  <li class="col3">Valdilecha</li>
  <li class="col3">Villar de Olmo</li>
</ol>
```



98 Una posible aplicación de esta técnica sería la creación de una página índice que llevara a las páginas personales de cada uno de los profesores del centro o de los alumnos de una clase.

Prueba a crear una lista ordenada alfabéticamente con los alumnos de tu clase (puede ser numerada o bien sin indicadores según prefieras).

La gran ventaja de esta forma de presentar las listas es que se aprovecha muy bien el espacio y se favorece el acceso a todos los elementos que queremos mostrar, al hacer que no sea necesario utilizar la barra de desplazamiento. Posiblemente te sobre media pantalla y a pesar de ello se verán todos los nombres de los alumnos o profesores sin dificultad.

SUSTITUCIÓN DE IMÁGENES

En algún momento a lo largo del curso hemos indicado la posibilidad de sustituir algún texto por imágenes. En una ocasión se sugirió como forma de preservar el diseño que habíamos creado para un título y en otra para ocultar una dirección de correo a los rastreadores pero hacerla visible a los visitantes.

Si consideras este último motivo es posible que se te ocurra pensar que los lectores de páginas para personas con deficiencia visual hacen algo muy similar a lo que hacen los rastreadores de direcciones: ir leyendo el texto que aparece en el código fuente. Así que, si en el código fuente no hay ningún texto...estaremos haciendo que ese fragmento de página no exista para el lector y, por tanto, para el visitante con problemas de visión que está visitando la página.

Este es el gran problema de la sustitución de imágenes: la accesibilidad. Hay multitud de técnicas de sustitución de imágenes publicadas en la web para obtener un diseño visual rico (suelen recibir el nombre IR por el inglés Image Replacement). Ninguna de ellas es perfecta y todas adolecen de alguna dificultad. La que vamos a ver ahora, que utilizaremos para crear la cabecera de un periódico, es accesible pero puede provocar problemas con la indexación en algunos buscadores.

Empecemos por comprobar el efecto y luego vemos la idea y la forma de traducirla al código CSS necesario.

ANDO.ENRED.ANDO

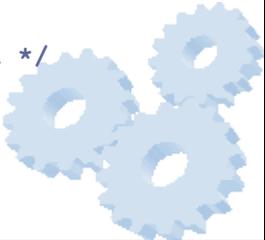
PERIÓDICO ESCOLAR DEL CEIP "COLENET"

El resultado se apoya en las siguientes ideas:

- Crearemos un elemento en bloque (`p`, `div`, `h1`, `h2`, ...) y utilizaremos una imagen sin repetir que mostrará el texto como fondo del elemento.
- Para que la imagen de fondo con el texto se muestre completa tendremos que darle al elemento unas dimensiones iguales a las de dicha imagen
- Para que los lectores de páginas tengan acceso al contenido textual del elemento crearemos una zona en línea `` en la que aparecerá el contenido que hay que leer. Con esto conseguimos también que los usuarios puedan ver el texto incluso si desactivan los estilos.
- Ocultamos el texto desplazándolo fuera de la zona de visibilidad con una instrucción `top` o `left` que lo saquen de la pantalla. Esta es la pega de esta técnica, ya que algunos buscadores han penalizado a páginas que empleaban esta técnica para incluir palabras que el visitante no veía pero que el robot del buscador sí leía, haciendo que la página subiera de posición de forma fraudulenta.

Vistas las ideas vamos al código CSS necesario.

```
<style type="text/css">
#cabecera {
background-image: url(imagenes/andoenredando.png);
background-repeat: no-repeat;
width: 685px;
height: 62px;
margin-left: auto;
margin-right: auto;
}
<!-- Recuerda que los márgenes automáticos para el centrado no funcionan en
Internet Explorer sin un parche -->
#subcabecera {
background-image: url(imagenes/andoenredando2.png);
background-repeat: no-repeat;
width: 375px;
height: 20px;
margin-left: auto;
margin-right: auto;
}
/* Ahora el desplazamiento para sacar el texto de la pantalla */
span.oculta {
position: relative;
left: -5000px;
}
</style>
```



Y ahora el código HTML

```
<h1 title="ando.enred.ando" id="cabecera">
<span class="oculta">ando.enred.ando</span>
</h1>
<h2 id="subcabecera" title="Periódico escolar del CEIP
'Colenet'">
<span class="oculta">Periódico escolar del CEIP "Colenet"</span>
</h2>
```



Para aumentar aún más la accesibilidad se han añadido atributos `title`, de forma que un usuario que no tuviera desactivada la utilización de CSS pero sí la carga de imágenes, contara con una referencia de lo que hay en el hueco que deja el elemento que no está viendo

CREAR UN PERIÓDICO CON TEXTO ENCOLUMNADO

Una opción cuando se crea un periódico es hacerlo de forma que el texto se presente en columnas, de forma similar a la apariencia de un periódico en papel. Hasta que no se habilite un soporte para columnas en futuras especificaciones no disponemos de la posibilidad directa de hacerlo con etiquetas HTML o reglas CSS por lo que el procedimiento será bastante artesanal y tendremos que equilibrar manualmente las columnas.

Aprovecharemos el ejemplo que se muestra, con una primera página de un **periódico escolar**, para ver cómo se puede crear una estructura con una cabecera, tres columnas y un pie.

PASO 1

La primera opción que tomaremos será si nuestra página ocupará el ancho de la pantalla o bien le adjudicaremos una anchura fija. En el ejemplo hemos optado por fijar el ancho tomando como referencia la **cabecera** que ya habíamos creado en una práctica anterior.

Dado que es previsible que queden amplios huecos a los lados empezaremos por crear una regla para que el fondo de la página tenga un color gris claro que sirva para delimitar la página.

```
body { background-color: rgb(238, 238, 238); }
```



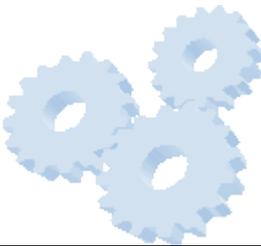
PASO 2

Crearemos un contenedor general que será el que incluya todos los elementos de la página y le adjudicaremos un tamaño.

Lo separamos ligeramente del borde superior e inferior y le adjudicamos un margen izquierdo y derecho automáticos para que quede centrado en la página. **Importante:** recuerda que el margen automático no sirve para centrar en Internet Explorer si no se aplica un parche. La alternativa es crear un primer contenedor con el texto centrado y, dentro de él, el contenedor real. En el ejemplo, como en toda la documentación utilizamos el parche de Dean Edwards.

Marcamos la posición como relativa sin indicar ningún desplazamiento con el objetivo de que pueda servir como referencia a los fragmentos situados dentro del contenedor

```
#contenedor { margin: 10px auto;
padding: 0px;
background-color: rgb(255, 255, 255);
position: relative;
width: 750px;
height: 100%;
min-height: 100%;
text-align: justify;
}
```



PASO 3

Creamos un identificador para el encabezamiento. Su anchura será el 100% del contenedor padre, esto es los 750px y, como altura utilizamos 100 píxeles, como resultado de la que va a formarse al sumar los fragmentos de las cabeceras que ya hemos creado previamente más un espacio de 8 píxeles para ampliar ligeramente la distancia entre cabecera y texto.

Como ves hay una corrección para Internet Explorer. El motivo es que vamos a juntar la cabecera y la subcabecera desplazando ésta hacia arriba, pero si no indicamos que queremos ocultar el espacio sobrante Internet Explorer calculará la altura total del contenedor teniendo en cuenta la altura de los dos elementos más sus márgenes, sin tener en cuenta el desplazamiento hacia arriba de la subcabecera.

Añadimos también un borde inferior para separar visualmente el encabezamiento del resto del documento.

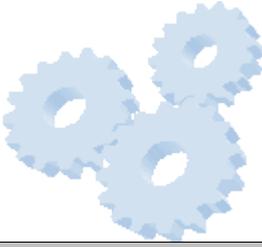
```
#encabezamiento { border-bottom: thin solid gray;
width: 750px;
height: 100px;
overflow: hidden; /* Corrección necesaria para Internet Explorer */
}
```



PASO 4

Rellenamos el encabezamiento con la cabecera y subcabecera que habíamos creado previamente. El código es prácticamente idéntico al que habíamos creado, salvo en que adjudicamos posición relativa a la subcabecera y la desplazamos 20 píxeles hacia arriba para reducir la distancia respecto a la cabecera principal. (Este desplazamiento es el que ha hecho necesario introducir la corrección que hemos mencionado en el paso anterior)

```
#cabecera { margin: 0pt auto;
background-image: url(imagenes/andoenredando.png);
background-repeat: no-repeat;
background-position: 0px 10px;
width: 685px;
height: 72px;
}
#subcabecera { background-image: url(imagenes/andoenredando2.png);
background-repeat: no-repeat;
width: 375px;
height: 20px;
position: relative;
top: -20px;
margin-left: auto;
margin-right: auto;
}
span.oculta { position: relative;
left: -5000px;
}
```



PASO 5

Vamos a crear tres columnas iguales, por lo que su anchura individual será de 250 píxeles que comprenderán el contenido, los rellenos, los márgenes y los bordes si los hubiera.

Aunque los identificadores han recibido los nombres `izq`, `cent` y `der` habrían sido innecesarios en este caso al ser las tres iguales. Si hubieran compartido algunas características pero su anchura hubiese sido diferente habríamos tenido que especificarla de forma individual para cada uno.

Observa que, al haber dejado un relleno de 10 píxeles el ancho del contenido queda en 230, de forma que $10+230+10$ sumen los 250 píxeles totales de los que disponemos para esta columna. Si hubiéramos especificado márgenes o bordes habría que haberlos tenido en cuenta para detraer su anchura del valor de `width` que se refiere, únicamente, al contenido de la caja

```
#izq, #cent, #der { padding: 10px;
float: left;
width: 230px;
}
```



PASO 6

Hemos procedido a rellenar las columnas de izquierda a derecha.

Para incluir la foto del segundo título (a "doble columna") se ha creado un nuevo contenedor que se ha situado inmediatamente detrás del que hace de columna izquierda.

```
#fotoprincipal { padding: 10px 10px 0px 0px;
width: 480px;
float: left;
}
```

Podríamos haberlo situado en cualquier otro punto de la página y haberlo posicionado de forma absoluta, pero teniendo en cuenta los factores de accesibilidad es preferible hacerlo en el flujo lógico de la página. Para que puedas comprobar el efecto tienes a continuación una miniatura de la página con los estilos aplicados.

A continuación puedes ver en otra miniatura **sin los estilos** para comprobar que el orden de lectura mantiene la secuencia lógica cuando se ve la página sin los estilos.



este caso sí que se trataría de una mala utilización del etiquetado, puesto que el objetivo es conseguir una modificación de la apariencia visual por lo que, simplemente se ha optado por crear una clase.

Para el ejemplo se ha utilizado la extensión Lorem Ipsum para generar un bloque de texto que rellena el espacio.

A los titulares se les ha aplicado una cabecera **h2** lo cual parece coherente con su sentido dentro de la estructura del documento. Como el título **h1** lo habíamos aplicado al nombre del periódico y es posible que la **h2** quedara un poco pequeña se ha creado una clase para especificar la apariencia visual.

En el caso de las entradas podría haberse recurrido a un título de menor nivel (un **h4** por ejemplo) pero en

PASO 7

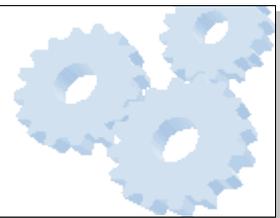
Por último se ha creado un contenedor para el pie. Su colocación en el flujo es después de la última columna, pero como no sabemos si esta será más corta que alguna de las precedentes incluimos una regla para que se impida la flotación de otros elementos en ambos lados y de esta forma garantizamos que siempre se colocará debajo de la columna más larga.

Hemos puesto una línea divisoria y un color de fondo para separarlo visualmente del resto del texto.

No hemos necesitado indicar la anchura ya que al tratarse de un elemento en bloque que no deja flotar a otros elementos utilizará la totalidad de la anchura del contenedor padre.

Sí es importante poner a cero los márgenes y rellenos para evitar que algún navegador les adjudique valores por defecto diferentes lo cual descompondría la distribución que vamos a hacer a continuación.

```
#pie { border-top: thin solid gray;
clear: both;
font-weight: bold;
margin:0;
padding:0;
}
```



Para rellenar el pie se ha optado por crear tres clases que permitan incluir texto en tres zonas del mismo. Al haber especificado las anchuras no necesitamos indicar flotación en la clase fecha porque ocupa exactamente el espacio que queda disponible dentro del contenedor

```
.numero { float: left;
width: 190px;
}
.centropie { float: left;
text-align: center;
width: 350px;
}
.fecha { width: 200px;
text-align: right;
}
```

