

## Modelo de cajas

El modelo de cajas o *"box model"* es seguramente la característica más importante del lenguaje de hojas de estilos CSS, ya que condiciona el diseño de todas las páginas web. El modelo de cajas es el comportamiento de CSS que hace que todos los elementos de las páginas se representen mediante cajas rectangulares.

Las cajas de una página se crean automáticamente. Cada vez que se inserta una etiqueta HTML, se crea una nueva caja rectangular que encierra los contenidos de ese elemento. La siguiente imagen muestra las tres cajas rectangulares que crean las tres etiquetas HTML que incluye la página:

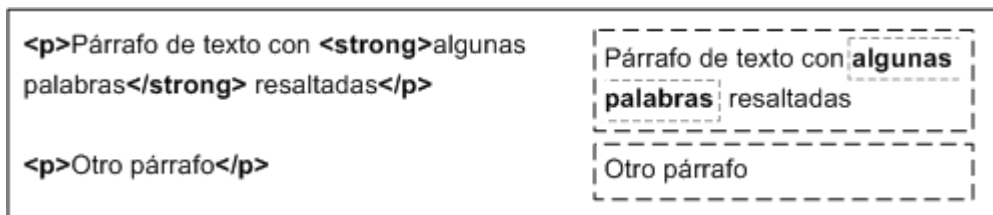


Figura 4.1. Las cajas se crean automáticamente al definir cada elemento HTML

Las cajas de las páginas no son visibles a simple vista porque inicialmente no muestran ningún color de fondo ni ningún borde. La siguiente imagen muestra las cajas que forman la página web de <http://www.alistapart.com/> después de forzar a que todas las cajas muestren un borde:



Figura 4.2. Cajas que forman la página alistapart.com

Los navegadores crean y colocan las cajas de forma automática, pero CSS permite modificar todas sus características. Cada una de las cajas está formada por seis partes, tal y como muestra la siguiente imagen:

### THE CSS BOX MODEL HIERARCHY

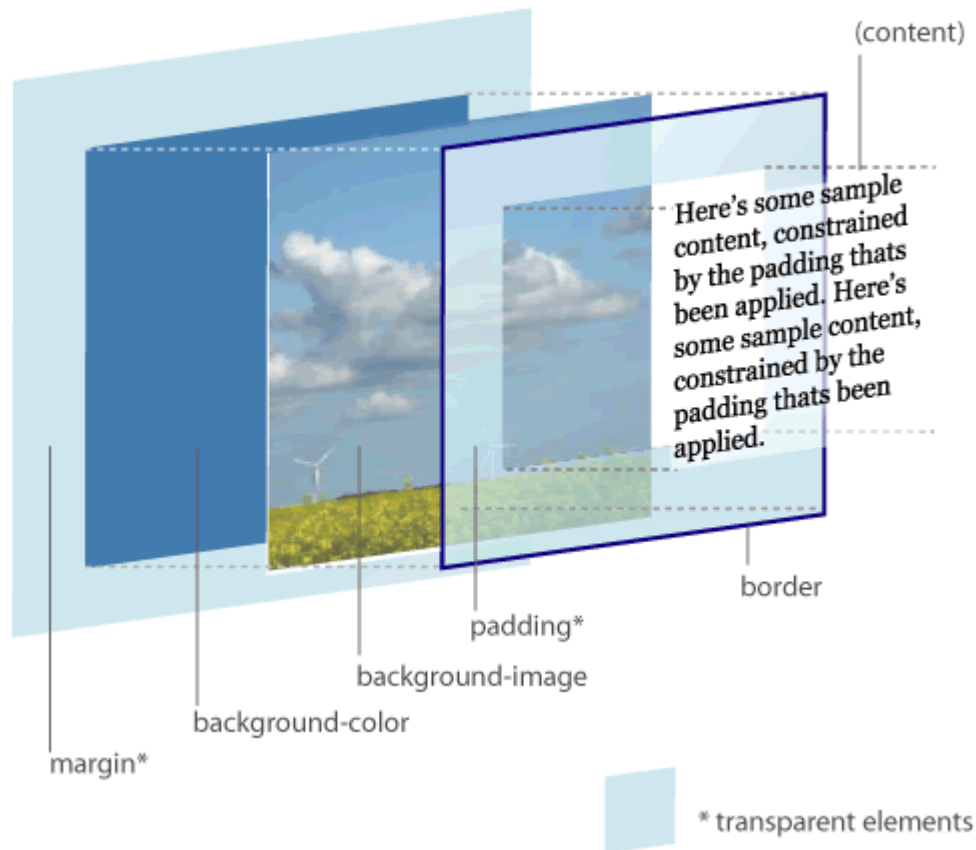


Figura 4.3. Representación tridimensional del box model de CSS

(Esquema utilizado de <http://www.hicksdesign.co.uk/boxmodel/>)

Las partes que componen cada caja y su orden de visualización desde el punto de vista del usuario son las siguientes:

- Contenido (*content*): se trata del contenido HTML del elemento (las palabras de un párrafo, una imagen, el texto de una lista de elementos, etc.)
- Relleno (*padding*): espacio libre opcional existente entre el contenido y el borde.
- Borde (*border*): línea que encierra completamente el contenido y su relleno.
- Imagen de fondo (*background image*): imagen que se muestra por detrás del contenido y el espacio de relleno.
- Color de fondo (*background color*): color que se muestra por detrás del contenido y el espacio de relleno.
- Margen (*margin*): separación opcional existente entre la caja y el resto de cajas adyacentes.

El relleno y el margen son transparentes, por lo que en el espacio ocupado por el relleno se muestra el color o imagen de fondo (si están definidos) y en el espacio ocupado por

el margen se muestra el color o imagen de fondo de su elemento padre (si están definidos). Si ningún elemento padre tiene definido un color o imagen de fondo, se muestra el color o imagen de fondo de la propia página (si están definidos).

Si una caja define tanto un color como una imagen de fondo, la imagen tiene más prioridad y es la que se visualiza. No obstante, si la imagen de fondo no cubre totalmente la caja del elemento o si la imagen tiene zonas transparentes, también se visualiza el color de fondo. Combinando imágenes transparentes y colores de fondo se pueden lograr efectos gráficos muy interesantes.

## 4.1. Anchura y altura

### 4.1.1. Anchura

La propiedad CSS que controla la anchura de los elementos se denomina `width`.

<b>width</b>	Anchura
<b>Valores</b>	<medida>   <porcentaje>   auto   inherit
<b>Se aplica a</b>	Todos los elementos, salvo los elementos en línea que no sean imágenes, las filas de tabla y los grupos de filas de tabla
<b>Valor inicial</b>	auto
<b>Descripción</b>	Establece la anchura de un elemento

La propiedad `width` no admite valores negativos y los valores en porcentaje se calculan a partir de la anchura de su elemento padre. El valor `inherit` indica que la anchura del elemento se hereda de su elemento padre. El valor `auto`, que es el que se utiliza si no se establece de forma explícita un valor a esta propiedad, indica que el navegador debe calcular automáticamente la anchura del elemento, teniendo en cuenta sus contenidos y el sitio disponible en la página.

El siguiente ejemplo establece el valor de la anchura del elemento `<div>` lateral:

```
#lateral { width: 200px; }  
  
<div id="lateral">  
  ...  
</div>
```

CSS define otras dos propiedades relacionadas con la anchura de los elementos: `min-width` y `max-width`, que se verán más adelante.

### 4.1.2. Altura

La propiedad CSS que controla la altura de los elementos se denomina `height`.

<b>height</b>	Altura
<b>Valores</b>	<medida>   <porcentaje>   auto   inherit
<b>Se aplica a</b>	Todos los elementos, salvo los elementos en línea que no sean imágenes, las columnas de tabla y los grupos de columnas de tabla

<b>Valor inicial</b>	auto
<b>Descripción</b>	Establece la altura de un elemento

Al igual que sucede con `width`, la propiedad `height` no admite valores negativos. Si se indica un porcentaje, se toma como referencia la altura del elemento padre. Si el elemento padre no tiene una altura definida explícitamente, se asigna el valor `auto` a la altura.

El valor `inherit` indica que la altura del elemento se hereda de su elemento padre. El valor `auto`, que es el que se utiliza si no se establece de forma explícita un valor a esta propiedad, indica que el navegador debe calcular automáticamente la altura del elemento, teniendo en cuenta sus contenidos y el sitio disponible en la página.

El siguiente ejemplo establece el valor de la altura del elemento `<div>` de cabecera:

```
#cabecera { height: 60px; }

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

CSS define otras dos propiedades relacionadas con la altura de los elementos: `min-height` y `max-height`, que se verán más adelante.

## 4.2. Margen y relleno

### 4.2.1. Margen

CSS define cuatro propiedades para controlar cada uno de los márgenes horizontales y verticales de un elemento.

<b>margin-top</b>	Margen superior
<b>margin-right</b>	Margen derecho
<b>margin-bottom</b>	Margen inferior
<b>margin-left</b>	Margen izquierdo
<b>Valores</b>	<medida>   <porcentaje>   auto   inherit
<b>Se aplica a</b>	Todos los elementos, salvo margin-top y margin-bottom que sólo se aplican a los elementos de bloque y a las imágenes
<b>Valor inicial</b>	0
<b>Descripción</b>	Establece cada uno de los márgenes horizontales y verticales de un elemento

Cada una de las propiedades establece la separación entre el borde lateral de la caja y el resto de cajas adyacentes:

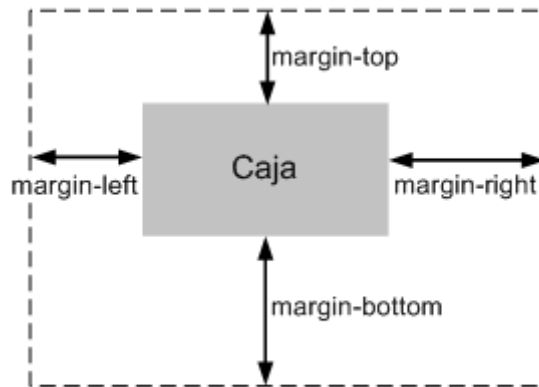


Figura 4.4. Las cuatro propiedades relacionadas con los márgenes

Las unidades más utilizadas para indicar los márgenes de un elemento son los píxeles (cuando se requiere una precisión total), los `em` (para hacer diseños que mantengan las proporciones) y los porcentajes (para hacer diseños líquidos o fluidos).

El siguiente ejemplo añade un margen izquierdo al segundo párrafo:

```
.destacado {  
  margin-left: 2em;  
}
```

```
<p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nam et  
elit.  
Vivamus placerat lorem. Maecenas sapien. Integer ut massa. Cras diam  
ipsum,  
laoreet non, tincidunt a, viverra sed, tortor.</p>
```

```
<p class="destacado">Vestibulum lectus diam, luctus vel, venenatis  
ultrices,  
cursus vel, tellus. Etiam placerat erat non sem. Nulla molestie odio  
non  
nisl tincidunt faucibus.</p>
```

```
<p>Aliquam euismod sapien eu libero. Ut tempor orci at nulla. Nam in  
eros  
egestas massa vehicula nonummy. Morbi posuere, nibh ultricies  
consectetur tincidunt,  
risus turpis laoreet elit, ut tincidunt risus sem et nunc.</p>
```

A continuación se muestra el aspecto del ejemplo anterior en cualquier navegador:

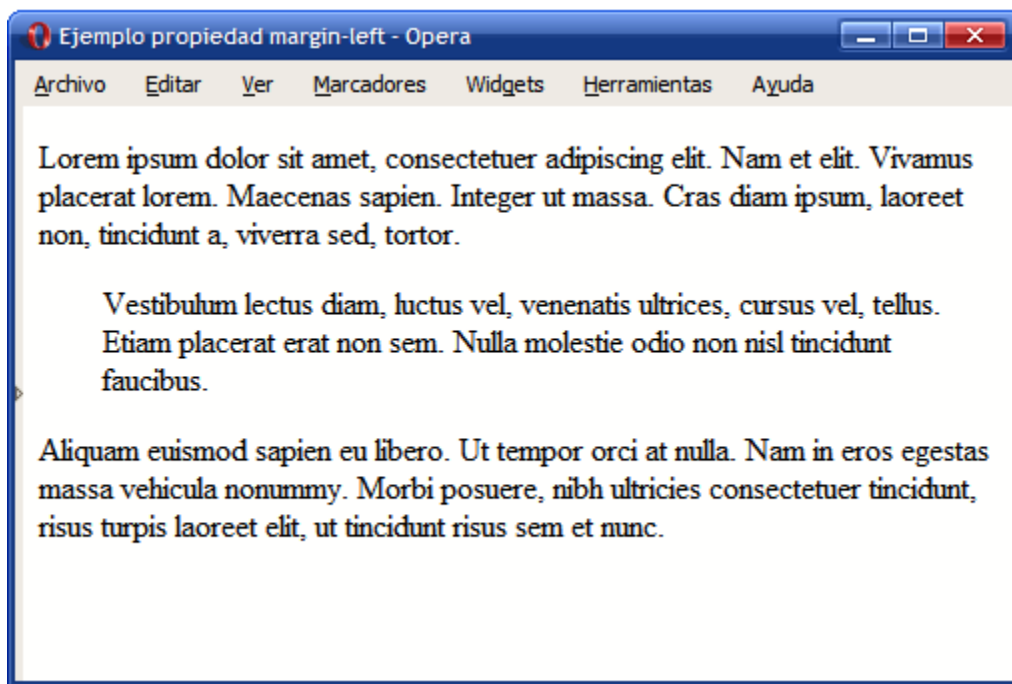


Figura 4.5. Ejemplo de propiedad margin-left

Algunos diseñadores web utilizan la etiqueta `<blockquote>` para encerrar los contenidos de un párrafo que se quiere mostrar tabulado respecto al resto de contenidos, como en el ejemplo anterior. Se trata de un error grave porque utiliza código XHTML erróneo para modificar el aspecto de los contenidos. Como ya se sabe, CSS es el único responsable del aspecto de los contenidos y dispone de propiedades como `margin-left` que permite conseguir los mismos resultados de forma correcta.

Los márgenes verticales (`margin-top` y `margin-bottom`) sólo se pueden aplicar a los elementos de bloque y las imágenes, mientras que los márgenes laterales (`margin-left` y `margin-right`) se pueden aplicar a cualquier elemento, tal y como muestra la siguiente imagen:

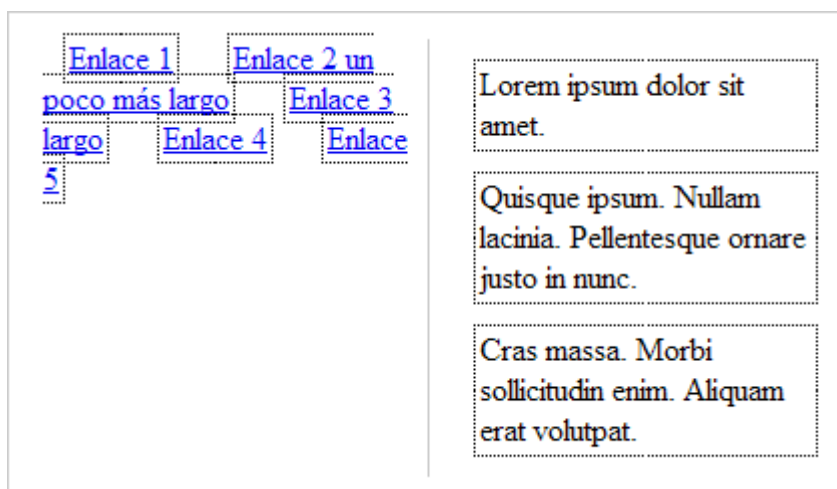


Figura 4.6. Los márgenes verticales sólo se aplican a los elementos de bloque e imágenes

La imagen anterior muestra el resultado de aplicar los mismos márgenes a varios enlaces (elementos en línea) y varios párrafos (elementos de bloque). En los elementos en línea los márgenes verticales no tienen ningún efecto, por lo que los enlaces no muestran ninguna separación vertical, al contrario de lo que sucede con los párrafos. Sin embargo, los márgenes laterales funcionan sobre cualquier tipo de elemento, por lo que los enlaces se muestran separados entre sí y los párrafos aumentan su separación con los bordes laterales de su elemento contenedor.

El siguiente ejemplo utiliza el mismo valor en los cuatro márgenes de cada imagen para facilitar su identificación y mejorar el diseño general de la página:

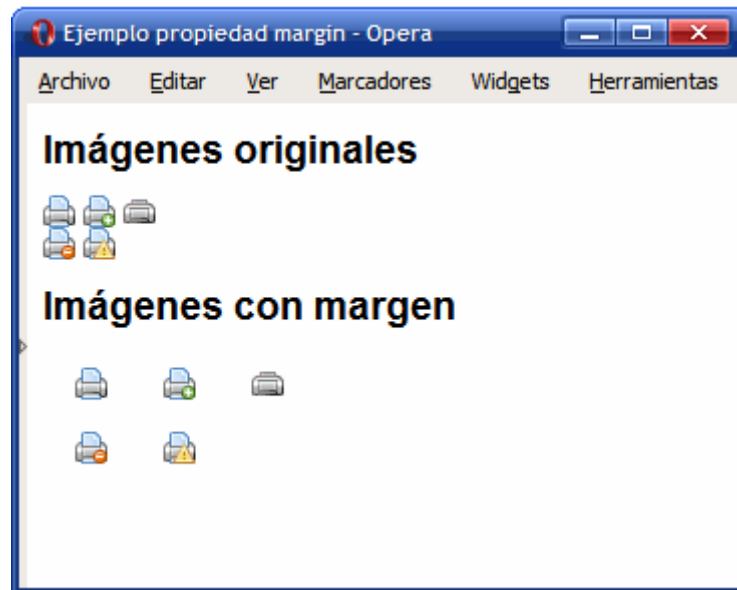


Figura 4.7. Ejemplo de propiedad margin

El código CSS del ejemplo anterior se muestra a continuación:

```
div img {
    margin-top: .5em;
    margin-bottom: .5em;
    margin-left: 1em;
    margin-right: .5em;
}
```

Además de las cuatro propiedades que controlan cada uno de los márgenes del elemento, CSS define una propiedad que permite establecer los cuatro márgenes de forma directa empleando una única propiedad. Este tipo de propiedades resumidas se denominan propiedades de tipo "*shorthand*" y CSS define varias propiedades de este tipo, como se verá más adelante.

La propiedad que permite definir de forma simultánea los cuatro márgenes se denomina `margin`.

<b>margin</b>	Margen
<b>Valores</b>	( <medida>   <porcentaje>   auto ) {1, 4}   inherit
<b>Se aplica a</b>	Todos los elementos salvo algunos casos especiales de elementos

	mostrados como tablas
<b>Valor inicial</b>	-
<b>Descripción</b>	Establece de forma directa todos los márgenes de un elemento

La notación `{1, 4}` de la definición anterior significa que la propiedad `margin` admite entre uno y cuatro valores, con el siguiente significado:

- Si solo se indica un valor, todos los márgenes tienen ese valor.
- Si se indican dos valores, el primero se asigna al margen superior e inferior y el segundo se asigna a los márgenes izquierdo y derecho.
- Si se indican tres valores, el primero se asigna al margen superior, el tercero se asigna al margen inferior y el segundo valor se asigna los márgenes izquierdo y derecho.
- Si se indican los cuatro valores, el orden de asignación es: margen superior, margen derecho, margen inferior y margen izquierdo.

El ejemplo anterior de márgenes se puede reescribir utilizando la propiedad `margin`:

Código CSS original:

```
div img {
  margin-top: .5em;
  margin-bottom: .5em;
  margin-left: 1em;
  margin-right: .5em;
}
```

Alternativa directa:

```
div img {
  margin: .5em .5em .5em 1em;
}
```

Otra alternativa:

```
div img {
  margin: .5em;
  margin-left: 1em;
}
```

El comportamiento de los márgenes verticales es más complejo de lo que se puede imaginar. Cuando se juntan dos o más márgenes verticales, se fusionan de forma automática y la altura del nuevo margen será igual a la altura del margen más alto de los que se han fusionado.



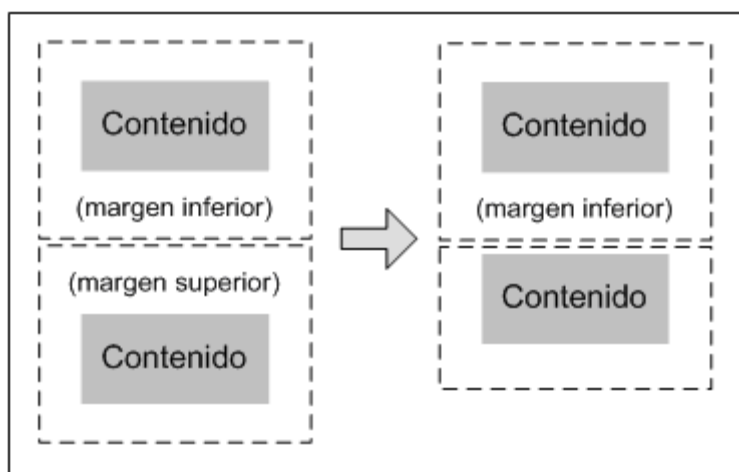


Figura 4.8. Fusión automática de los márgenes verticales

De la misma forma, si un elemento está contenido dentro de otro elemento, sus márgenes verticales se fusionan y resultan en un nuevo margen de la misma altura que el mayor margen de los que se han fusionado:

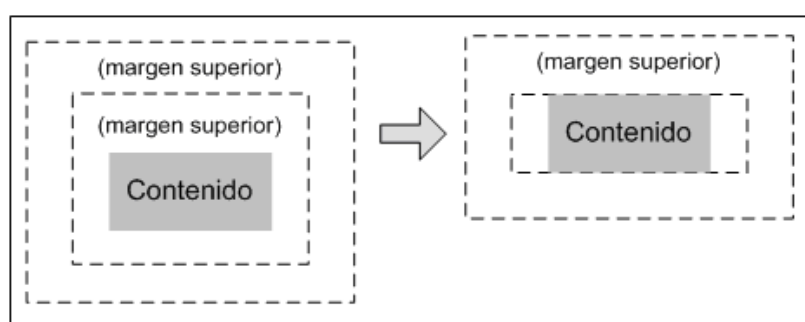


Figura 4.9. Fusión de los márgenes de los elementos interiores

Aunque en principio puede parecer un comportamiento extraño, la razón por la que se propuso este mecanismo de fusión automática de márgenes verticales es el de dar uniformidad a las páginas web habituales. En una página con varios párrafos, si no se diera este comportamiento y se estableciera un determinado margen a todos los párrafos, el primer párrafo no mostraría un aspecto homogéneo respecto de los demás.

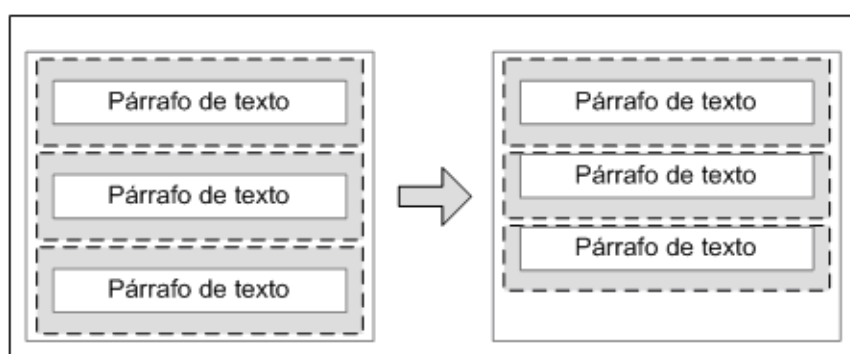


Figura 4.10. Motivo por el que se fusionan automáticamente los márgenes verticales

En el caso de un elemento que se encuentra en el interior de otro y sus márgenes se fusionan de forma automática, se puede evitar este comportamiento añadiendo un pequeño relleno (`padding: 1px`) o un borde (`border: 1px solid transparent`) al elemento contenedor.

#### 4.2.2. Relleno

CSS define cuatro propiedades para controlar cada uno de los espacios de relleno horizontales y verticales de un elemento.

<b>padding-top</b>	Relleno superior
<b>padding-right</b>	Relleno derecho
<b>padding-bottom</b>	Relleno inferior
<b>padding-left</b>	Relleno izquierdo
<b>Valores</b>	<medida>   <porcentaje>   inherit
<b>Se aplica a</b>	Todos los elementos excepto algunos elementos de tablas como grupos de cabeceras y grupos de pies de tabla
<b>Valor inicial</b>	0
<b>Descripción</b>	Establece cada uno de los rellenos horizontales y verticales de un elemento

Cada una de las propiedades establece la separación entre el lateral de los contenidos y el borde lateral de la caja:

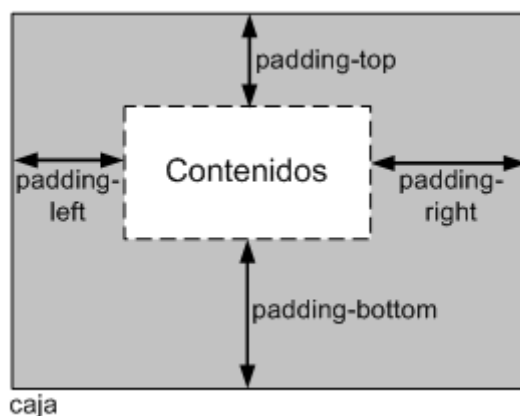


Figura 4.11. Las cuatro propiedades relacionadas con los rellenos

La siguiente imagen muestra la diferencia entre el margen y el relleno de los elementos:

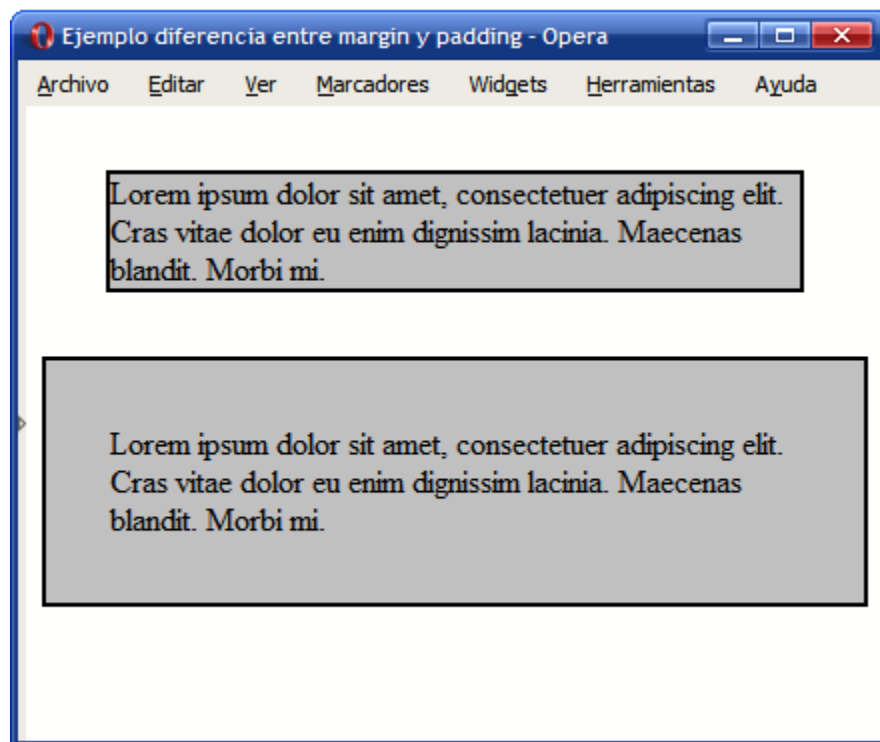


Figura 4.12. Diferencia entre relleno (padding) y margen (margin)

El código HTML y CSS del ejemplo se muestra a continuación:

```
.margen {
  margin-top: 2em; margin-right: 2em; margin-bottom: 2em; margin-left:
2em;
}
.relleno {
  padding-top: 2em; padding-right: 2em; padding-bottom: 2em; padding-
left: 2em;
}

<p class="margen">Lorem ipsum dolor sit amet, consectetur adipiscing elit.
Cras vitae dolor eu enim dignissim lacinia. Maecenas blandit. Morbi
mi.</p>

<p class="relleno">Lorem ipsum dolor sit amet, consectetur adipiscing elit.
Cras vitae dolor eu enim dignissim lacinia. Maecenas blandit. Morbi
mi.</p>
```

Como sucede con la propiedad `margin`, CSS también define una propiedad de tipo "shorthand" para establecer los cuatro rellenos de un elemento de forma directa. La propiedad que permite definir de forma simultanea los cuatro márgenes se denomina `padding`.

<b>padding</b>	Relleno
<b>Valores</b>	( <medida>   <porcentaje> ) {1, 4}   inherit
<b>Se aplica a</b>	Todos los elementos excepto algunos elementos de tablas como grupos de cabeceras y grupos de pies de tabla

<b>Valor inicial</b>	-
<b>Descripción</b>	Establece de forma directa todos los rellenos de los elementos

La notación {1, 4} de la definición anterior significa que la propiedad `padding` admite entre uno y cuatro valores, con el mismo significado que el de la propiedad `margin`. Ejemplo:

```
body {padding: 2em} /* Todos los rellenos valen 2em */
body {padding: 1em 2em} /* Superior e inferior = 1em, Izquierdo y derecho = 2em */
body {padding: 1em 2em 3em} /* Superior = 1em, derecho = 2em, inferior = 3em, izquierdo = 2em */
body {padding: 1em 2em 3em 4em} /* Superior = 1em, derecho = 2em, inferior = 3em, izquierdo = 4em */
```

## 4.3. Bordes

CSS permite definir el aspecto de cada uno de los cuatro bordes horizontales y verticales de los elementos. Para cada borde se puede establecer su anchura, su color y su estilo.

### 4.3.1. Anchura

La anchura de los bordes se controla con las cuatro propiedades siguientes:

<b>border-top-width</b>	Anchura del borde superior
<b>border-right-width</b>	Anchura del borde derecho
<b>border-bottom-width</b>	Anchura del borde inferior
<b>border-left-width</b>	Anchura del borde izquierdo
<b>Valores</b>	( <medida>   thin   medium   thick )   inherit
<b>Se aplica a</b>	Todos los elementos
<b>Valor inicial</b>	Medium
<b>Descripción</b>	Establece la anchura de cada uno de los cuatro bordes de los elementos

La anchura de los bordes se puede indicar mediante una medida (absoluta o relativa y en cualquier unidad de medida de las definidas) o mediante las palabras clave `thin` (borde delgado), `medium` (borde normal) y `thick` (borde ancho).

La medida más habitual para indicar la anchura de los bordes es el píxel, ya que permite un control preciso del grosor. Las palabras clave apenas se utilizan, ya que impiden mostrar bordes iguales en diferentes navegadores. El motivo es que el estándar CSS no indica explícitamente el grosor al que equivale cada palabra clave. Así por ejemplo, el grosor `medium` equivale a 4px en Internet Explorer y 3px en el resto de navegadores.

El siguiente ejemplo muestra un elemento con cuatro anchuras diferentes de borde:

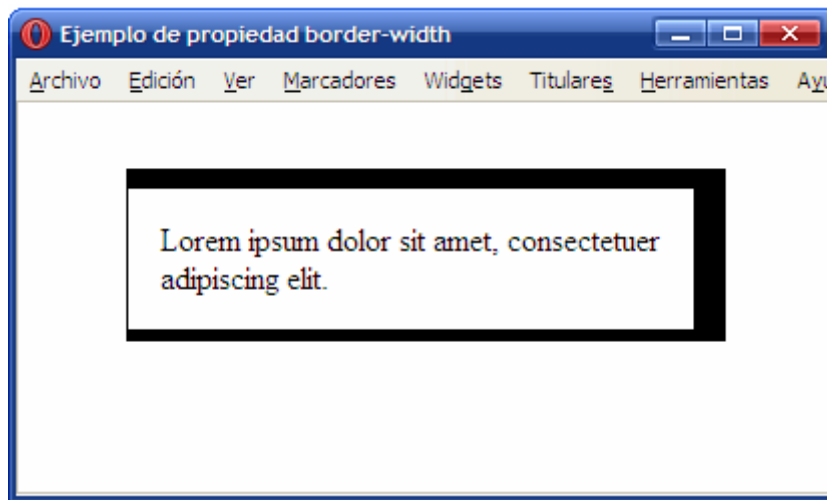


Figura 4.13. Ejemplo de propiedad border-width

Las reglas CSS utilizadas se muestran a continuación:

```
div {
  border-top-width: 10px;
  border-right-width: 1em;
  border-bottom-width: thick;
  border-left-width: thin;
}
```

Si se quiere establecer la misma anchura a todos los bordes, CSS permite la utilización de un atajo mediante una propiedad de tipo "*shorthand*", que permiten indicar varias propiedades de forma resumida:

<b>border-width</b>	Anchura del borde
<b>Valores</b>	( <medida>   thin   medium   thick ) {1, 4}   inherit
<b>Se aplica a</b>	Todos los elementos
<b>Valor inicial</b>	Medium
<b>Descripción</b>	Establece la anchura de todos los bordes del elemento

La propiedad `border-width` permite indicar entre uno y cuatro valores. El significado de cada caso es el habitual de las propiedades "*shorthand*":

```
p { border-width: thin } /* thin thin thin thin */
p { border-width: thin thick } /* thin thick thin thick */
p { border-width: thin thick medium } /* thin thick medium thick */
/*
p { border-width: thin thick medium thin } /* thin thick medium thin */
*/
```

Si se indica un solo valor, se aplica a los cuatro bordes. Si se indican dos valores, el primero se aplica al borde superior e inferior y el segundo valor se aplica al borde izquierdo y derecho.

Si se indican tres valores, el primero se aplica al borde superior, el segundo se aplica al borde izquierdo y derecho y el tercer valor se aplica al borde inferior. Si se indican los cuatro valores, el orden de aplicación es superior, derecho, inferior e izquierdo.

### 4.3.2. Color

El color de los bordes se controla con las cuatro propiedades siguientes:

<b>border-top-color</b>	Color del borde superior
<b>border-right-color</b>	Color del borde derecho
<b>border-bottom-color</b>	Color del borde inferior
<b>border-left-color</b>	Color del borde izquierdo
<b>Valores</b>	<color>   transparent   inherit
<b>Se aplica a</b>	Todos los elementos
<b>Valor inicial</b>	-
<b>Descripción</b>	Establece el color de cada uno de los cuatro bordes de los elementos

El ejemplo anterior se puede modificar para mostrar cada uno de los bordes de un color diferente:

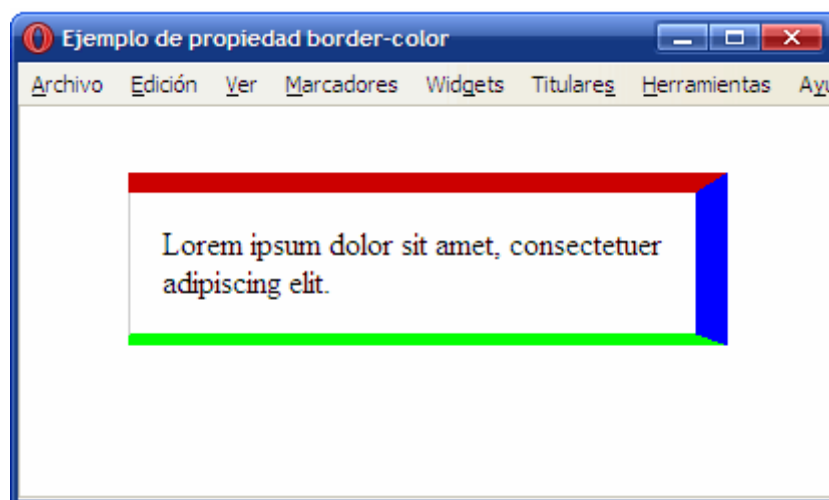


Figura 4.14. Ejemplo de propiedad border-color

Las reglas CSS necesarias para mostrar los colores anteriores son las siguientes:

```
div {
  border-top-color: #CC0000;
  border-right-color: blue;
  border-bottom-color: #00FF00;
  border-left-color: #CCC;
}
```

Si se quiere establecer el mismo color para todos los bordes, CSS permite la utilización de un atajo mediante una propiedad de tipo *"shorthand"*, que permiten indicar varias propiedades de forma resumida:

<b>border-color</b>	Color del borde
<b>Valores</b>	( <color>   transparent ) {1, 4}   inherit
<b>Se aplica a</b>	Todos los elementos
<b>Valor inicial</b>	-
<b>Descripción</b>	Establece el color de todos los bordes del elemento

En este caso, al igual que sucede con la propiedad `border-width`, es posible indicar de uno a cuatro valores y las reglas de aplicación son idénticas a la propiedad `border-width`.

### 4.3.3. Estilo

Por último, CSS permite establecer el estilo de cada uno de los bordes mediante las siguientes propiedades:

<b>border-top-style</b>	Estilo del borde superior
<b>border-right-style</b>	Estilo del borde derecho
<b>border-bottom-style</b>	Estilo del borde inferior
<b>border-left-style</b>	Estilo del borde izquierdo
<b>Valores</b>	none   hidden   dotted   dashed   solid   double   groove   ridge   inset   outset   inherit
<b>Se aplica a</b>	Todos los elementos
<b>Valor inicial</b>	none
<b>Descripción</b>	Establece el estilo de cada uno de los cuatro bordes de los elementos

El estilo de los bordes sólo se puede indicar mediante alguna de las palabras reservadas definidas por CSS. Como el valor por defecto de esta propiedad es `none`, los elementos no muestran ningún borde visible a menos que se establezca explícitamente un estilo de borde.

Siguiendo el ejemplo anterior, se puede modificar el estilo de cada uno de los bordes:

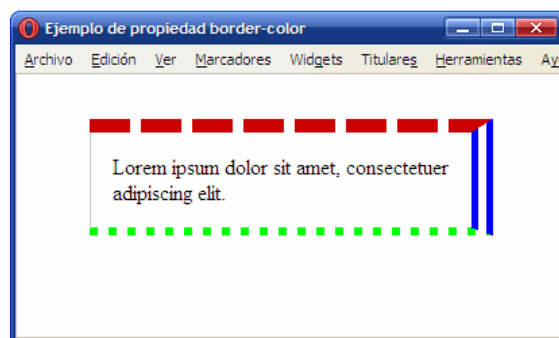


Figura 4.15. Ejemplo de propiedad border-style

Las reglas CSS necesarias para mostrar los estilos anteriores son las siguientes:

```
div {
  border-top-style: dashed;
  border-right-style: double;
  border-bottom-style: dotted;
  border-left-style: solid;
}
```

El aspecto con el que los navegadores muestran los diferentes tipos de borde se muestra a continuación:

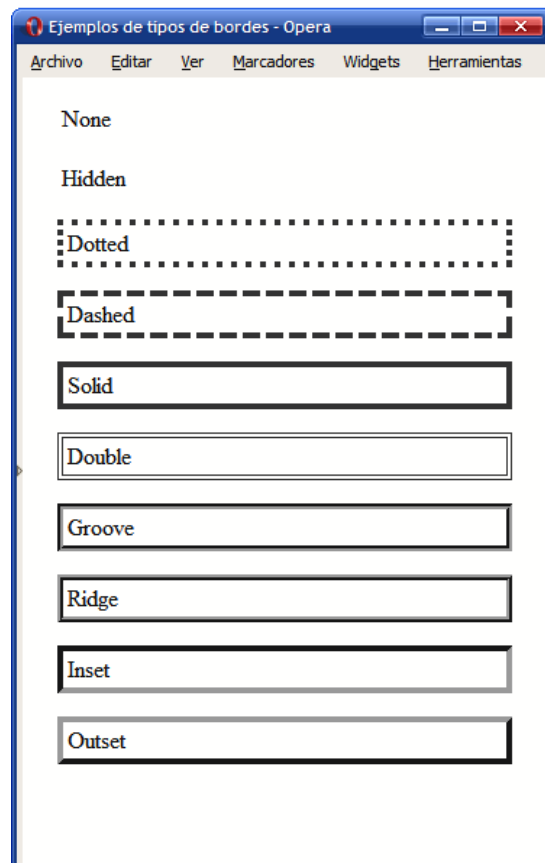


Figura 4.16. Tipos de bordes definidos por CSS

Los bordes más utilizados en los diseños habituales son `solid` y `dashed`, seguidos de `double` y `dotted`. Los estilos `none` y `hidden` son idénticos visualmente, pero se diferencian en la forma que los navegadores resuelven los conflictos entre los bordes de las celdas adyacentes en las tablas.

Si se quiere establecer el mismo estilo para todos los bordes, CSS define una propiedad de tipo "shorthand":

border-style	Estilo del borde
Valores	(none   hidden   dotted   dashed   solid   double   groove   ridge   inset   outset ) {1, 4}   inherit
Se aplica a	Todos los elementos
Valor inicial	-



<b>Descripción</b>	Establece el estilo de todos los bordes del elemento
--------------------	--

Como es habitual, la propiedad permite indicar de uno a cuatro valores diferentes y las reglas de aplicación son las habituales de las propiedades *"shorthand"*.

#### 4.3.4. Propiedades shorthand

Como sucede con los márgenes y los rellenos, CSS define una serie de propiedades de tipo *"shorthand"* que permiten establecer todos los atributos de los bordes de forma directa. CSS ha definido una propiedad *"shorthand"* para cada uno de los cuatro bordes y una propiedad *"shorthand"* global.

Antes de presentar las propiedades, es conveniente definir los tres siguientes tipos de valores:

```
<medida_borde> = <medida> | thin | medium | thick
<color_borde> = <color> | transparent
<estilo_borde> = none | hidden | dotted | dashed | solid | double |
groove | ridge | inset | outset
```

<b>border-top</b>	Estilo completo del borde superior
<b>border-right</b>	Estilo completo del borde derecho
<b>border-bottom</b>	Estilo completo del borde inferior
<b>border-left</b>	Estilo completo del borde izquierdo
<b>Valores</b>	( <medida_borde>    <color_borde>    <estilo_borde> )   inherit
<b>Se aplica a</b>	Todos los elementos
<b>Valor inicial</b>	-
<b>Descripción</b>	Establece el estilo completo de cada uno de los cuatro bordes de los elementos

Las propiedades *"shorthand"* permiten establecer alguno o todos los atributos de cada borde. El siguiente ejemplo establece el color y el tipo del borde inferior, pero no su anchura:

```
h1 {
  border-bottom: solid red;
}
```

En el ejemplo anterior, la anchura del borde será la correspondiente al valor por defecto (medium). Este otro ejemplo muestra la forma habitual utilizada para establecer el estilo de cada borde:

```
div {
  border-top: 1px solid #369;
  border-bottom: 3px double #369;
}
```

Por ultimo, CSS define una propiedad de tipo *"shorthand"* global para establecer el valor de todos los atributos de todos los bordes de forma directa:

<b>border</b>	Estilo completo de todos los bordes
<b>Valores</b>	( <medida_borde>    <color_borde>    <estilo_borde> )   inherit
<b>Se aplica a</b>	Todos los elementos
<b>Valor inicial</b>	-
<b>Descripción</b>	Establece el estilo completo de todos los bordes de los elementos

Las siguientes reglas CSS son equivalentes:

```
div {
  border-top: 1px solid red;
  border-right: 1px solid red;
  border-bottom: 1px solid red;
  border-left: 1px solid red;
}
```

```
div { border: 1px solid red; }
```

Como el valor por defecto de la propiedad `border-style` es `none`, si una propiedad *shorthand* no establece explícitamente el estilo de un borde, el elemento no muestra ese borde:

```
/* Sólo se establece el color, por lo que el estilo es
   "none" y el borde no se muestra */
div { border: red; }

/* Se establece el grosor y el color del borde, pero no
   su estilo, por lo que es "none" y el borde no se muestra */
div { border-bottom: 5px blue; }
```

Cuando los cuatro bordes no son idénticos pero sí muy parecidos, se puede utilizar la propiedad `border` para establecer de forma directa los atributos comunes de todos los bordes y posteriormente especificar para cada uno de los cuatro bordes sus propiedades particulares:

```
h1 {
  border: solid #000;
  border-top-width: 6px;
  border-left-width: 8px;
}
```

## 4.4. Margen, relleno, bordes y modelo de cajas

La anchura y altura de un elemento no solamente se calculan teniendo en cuenta sus propiedades `width` y `height`. El margen, el relleno y los bordes establecidos a un elemento determinan la anchura y altura final del elemento. En el siguiente ejemplo se muestran los estilos CSS de un elemento:

```
div {
  width: 300px;
  padding-left: 50px;
  padding-right: 50px;
  margin-left: 30px;
```

```
margin-right: 30px;
border: 10px solid black;
}
```

La anchura total con la que se muestra el elemento no son los 300 píxel indicados en la propiedad `width`, sino que se tienen en cuenta todos sus márgenes, rellenos y bordes:

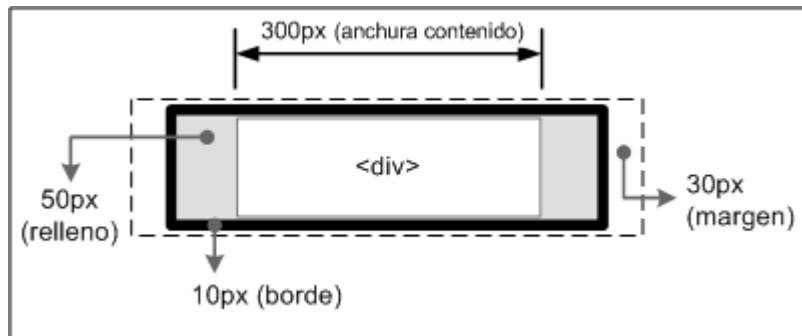


Figura 4.17. La anchura total de un elemento tiene en cuenta los márgenes, rellenos y bordes

De esta forma, la anchura del elemento en pantalla sería igual a la suma de la anchura original, los márgenes, los bordes y los rellenos:

$$30\text{px} + 10\text{px} + 50\text{px} + 300\text{px} + 50\text{px} + 10\text{px} + 30\text{px} = 480 \text{ píxel}$$

Así, la anchura/altura establecida con CSS siempre hace referencia a la anchura/altura del contenido. La anchura/altura total del elemento debe tener en cuenta además los valores del resto de partes que componen la caja del *box model*.

Por otra parte, la guerra de navegadores que se produjo en los años 90 provocó que cada fabricante (Microsoft y Netscape) añadiera sus propias extensiones y mejoras en sus productos. Posteriormente, aparecieron los estándares publicados por el W3C y los fabricantes se encontraron con el problema de la incompatibilidad entre sus implementaciones anteriores de HTML y CSS y las implementaciones que requerían los estándares.

La solución que adoptaron fue la de incluir en el navegador dos modos diferentes de funcionamiento: modo compatible con las páginas antiguas (denominado "*modo quirks*") y que se podría traducir como "*modo raro*") y modo compatible con los nuevos estándares (denominado "*modo estándar*"). El modo *quirks* es equivalente a la forma en la que se visualizaban las páginas en los navegadores Internet Explorer 4 y Netscape Navigator 4.

La diferencia más notable entre los dos modos es el tratamiento del "*box model*", lo que puede afectar gravemente al diseño de las páginas HTML. Los navegadores seleccionan automáticamente el modo en el que muestran las páginas en función del `DOCTYPE` definido por el documento. En general, los siguientes tipos de `DOCTYPE` activan el modo *quirks* en los navegadores:

- No utilizar ningún `DOCTYPE`

- DOCTYPE anterior a HTML 4.0 (`<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 3.2 Final//EN">`) \* DOCTYPE de HTML 4.01 sin URL (`<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">`)

En el caso concreto de Internet Explorer, también activan el modo quirks los modos XHTML 1.0 que incluyen la declaración de XML (por ejemplo `<?xml version="1.0" encoding="UTF-8"?>`) al principio de la página web:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
```

Se pueden consultar todos los casos concretos que activan el modo *quirks* para cada navegador en la página <http://hsivonen.iki.fi/doctype/>

La versión 5.5 y anteriores de Internet Explorer y las versiones 6 y 7 en modo *quirks* siguen su propio modelo de cálculo de anchuras y alturas que es muy diferente al método definido por el estándar.

La siguiente imagen muestra el elemento del ejemplo anterior en la versión 6 de Internet Explorer en modo estándar:

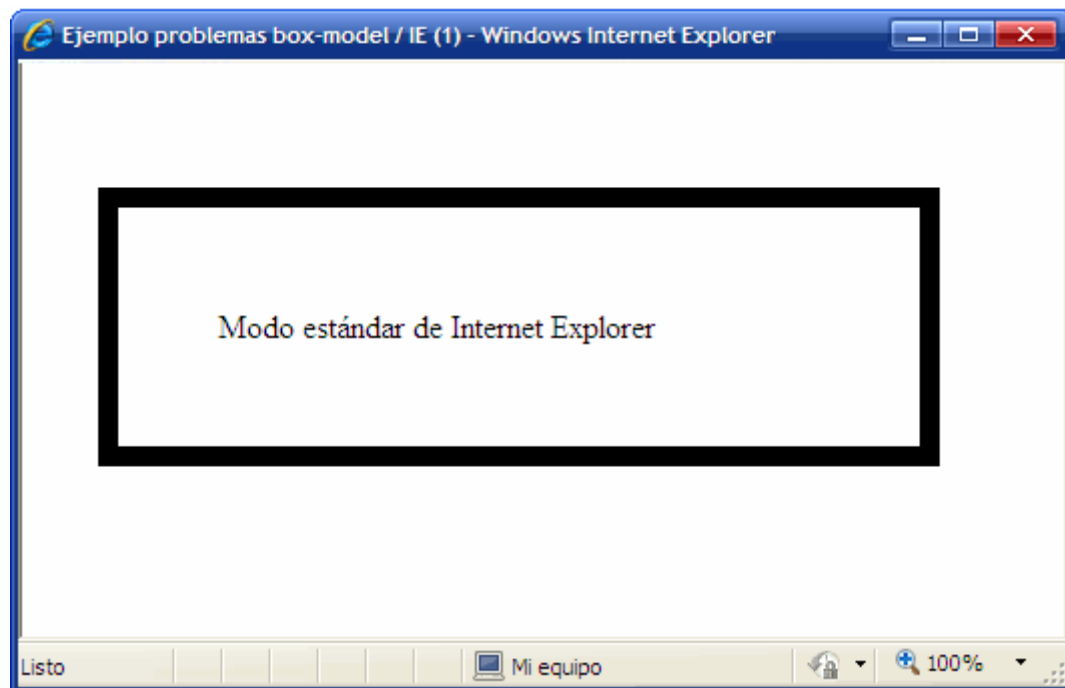


Figura 4.18. Internet Explorer 6 en modo estándar

La anchura del elemento es la que se obtiene de sumar la anchura de su contenido (300), sus bordes ( $2 \times 10$ ) y sus rellenos ( $2 \times 50$ ). Por lo tanto, la anchura del elemento son 420 píxel, a los que se suman los 30 píxel de margen lateral a cada lado.

Sin embargo, el mismo ejemplo en el modo *quirks* de la versión 6 de Internet Explorer muestra el siguiente aspecto:

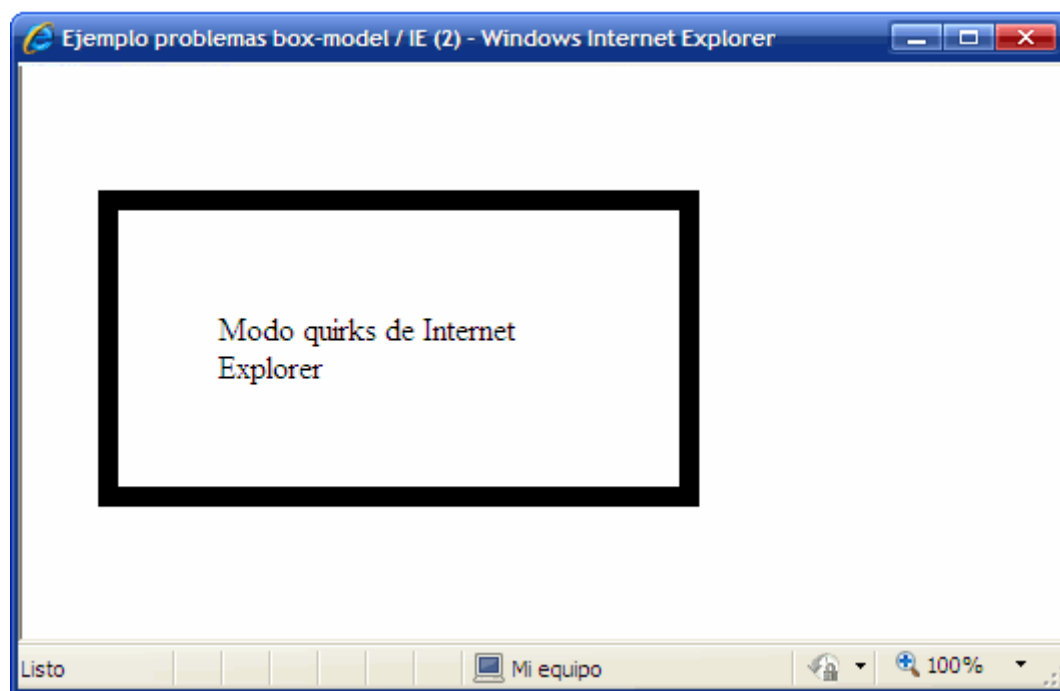


Figura 4.19. Internet Explorer 6 en modo quirks

Las versiones anteriores de Internet Explorer y las versiones 6 y 7 en modo *quirks* consideran que la anchura establecida por CSS no sólo es la anchura del contenido, sino que también incluye los bordes y el relleno.

Por lo tanto, en este caso la anchura total del elemento (sin contar los márgenes laterales) es de 300 píxel, el mismo valor que se indica en la propiedad `width`. El espacio ocupado por los bordes del elemento ( $2 \times 10$ ) y sus rellenos ( $2 \times 50$ ) se resta de la anchura de su contenido.

Para evitar este problema y crear diseños con el mismo aspecto en cualquier navegador, es necesario evitar el modo *quirks* de Internet Explorer. Por tanto, todas las páginas deben incluir la declaración apropiada de `DOCTYPE`.

#### 4.4.1. Los modos de compatibilidad de Internet Explorer 8

El navegador Internet Explorer 8 introduce el concepto de "*compatibilidad de la página*" para asegurar que todas las páginas HTML se vean correctamente en cualquier versión de ese navegador. En realidad, esta nueva característica es una mejora del *modo quirks* explicado anteriormente.

Internet Explorer 8, a diferencia de sus versiones anteriores, soporta completamente el estándar CSS 2.1. Sin embargo, muchos sitios web se diseñaron para Internet Explorer 6 y 7, por lo que incluyen trucos, *hacks* y filtros que arreglan los errores y carencias de esas versiones del navegador.

Para evitar que las páginas diseñadas para navegadores anteriores se vean mal en esta nueva versión, Internet Explorer 8 incluye la opción de "compatibilidad de la página",

que permite indicar la versión de Internet Explorer para la que la página ha sido diseñada.

De esta forma, si la página no se visualiza correctamente en Internet Explorer 8, se puede indicar al navegador que la muestre como si fuera Internet Explorer 6 o 7. En realidad, Internet Explorer 8 incluye seis modos de funcionamiento:

- **Modo IE5:** la página se muestra según el modo *quirks* de Internet Explorer 7, que es casi idéntico a como se veían las páginas en el navegador Internet Explorer 5.
- **Modo IE7:** la página se muestra en el modo estándar de Internet Explorer 7, sin importar si la página contiene o no la directiva `<!DOCTYPE>`.
- **Modo IE8:** los contenidos se muestran en el modo estándar de Internet Explorer 8, que es el más parecido al del resto de navegadores que soportan los estándares (Firefox, Opera, Safari y Google Chrome).
- **Emular el modo IE7:** el navegador decide cómo mostrar los contenidos a partir de la directiva `<!DOCTYPE>` de la página. Si esa directiva es una de las que activan el modo estándar, la página se muestra en el modo estándar de Internet Explorer 7. En otro caso, se muestra en el modo *quirks* de Internet Explorer 5. Este modo es el más útil para la mayoría de sitios web.
- **Emular el modo IE8:** el navegador decide cómo mostrar los contenidos a partir de la directiva `<!DOCTYPE>` de la página. Si esa directiva es una de las que activan el modo estándar, la página se muestra en el modo estándar de Internet Explorer 8. En otro caso, se muestra en el modo *quirks* de Internet Explorer 5.
- **Modo límite ("edge mode"):** indica a Internet Explorer que los contenidos se deben mostrar en el modo de compatibilidad más avanzado disponible. Actualmente, este modo es equivalente al modo IE8. Si las futuras versiones Internet Explorer 9 y 10 incluyeran mejor compatibilidad, las páginas se visualizarían en ese modo avanzado de compatibilidad.

El modo de compatibilidad de la página se indica mediante una nueva etiqueta `<meta>` con la propiedad `X-UA-Compatible` y cuyo valor es el que utiliza Internet Explorer 8 para determinar el modo que se utiliza:

```
<!-- Modo IE5 -->
<head>
  <meta http-equiv="X-UA-Compatible" content="IE=5" />
  ...
</head>

<!-- Modo IE7 -->
<head>
  <meta http-equiv="X-UA-Compatible" content="IE=7" />
  ...
</head>

<!-- Modo IE8 -->
<head>
  <meta http-equiv="X-UA-Compatible" content="IE=8" />
  ...
</head>

<!-- Emular el modo IE7 -->
<head>
  <meta http-equiv="X-UA-Compatible" content="IE=EmulateIE7" />
  ...
```

```

</head>

<!-- Emular el modo IE8 -->
<head>
  <meta http-equiv="X-UA-Compatible" content="IE=EmulateIE8" />
  ...
</head>

<!-- Modo límite -->
<head>
  <meta http-equiv="X-UA-Compatible" content="IE=edge" />
  ...
</head>

```

No obstante, esta opción de compatibilidad de la página debe entenderse como una solución temporal que evita que los sitios web se vean mal en Internet Explorer 8. La única solución correcta a largo plazo consiste en actualizar las páginas para que sus diseños sigan los estándares web.

## 4.5. Fondos

El último elemento que forma el *box model* es el fondo de la caja del elemento. El fondo puede ser un color simple o una imagen. El fondo solamente se visualiza en el área ocupada por el contenido y su relleno, ya que el color de los bordes se controla directamente desde los bordes y las zonas de los márgenes siempre son transparentes.

Para establecer un color o imagen de fondo en la página entera, se debe establecer un fondo al elemento `<body>`. Si se establece un fondo a la página, como el valor inicial del fondo de los elementos es transparente, todos los elementos de la página se visualizan con el mismo fondo a menos que algún elemento especifique su propio fondo.

CSS define cinco propiedades para establecer el fondo de cada elemento (`background-color`, `background-image`, `background-repeat`, `background-attachment`, `background-position`) y otra propiedad de tipo "*shorthand*" (`background`).

<b>background-color</b>	Color de fondo
<b>Valores</b>	<code>&lt;color&gt;</code>   <code>transparent</code>   <code>inherit</code>
<b>Se aplica a</b>	Todos los elementos
<b>Valor inicial</b>	<code>transparent</code>
<b>Descripción</b>	Establece un color de fondo para los elementos

El siguiente ejemplo muestra una página web con un color gris claro de fondo:

```

body {
  background-color: #F5F5F5;
}

```

En ocasiones, es necesario crear un fondo más complejo que un simple color. CSS permite mostrar una imagen como fondo de cualquier elemento:

<b>background-image</b>	Imagen de fondo
-------------------------	-----------------

<b>Valores</b>	<url>   none   inherit
<b>Se aplica a</b>	Todos los elementos
<b>Valor inicial</b>	none
<b>Descripción</b>	Establece una imagen como fondo para los elementos

CSS permite establecer de forma simultánea un color y una imagen de fondo. En este caso, la imagen se muestra delante del color, por lo que solamente si la imagen contiene zonas transparentes es posible ver el color de fondo.

El siguiente ejemplo muestra una imagen como fondo de toda la página:

```
body { background-image: url("imagenes/fondo.png") }
```

Las imágenes de fondo se indican a través de su URL, que puede ser absoluta o relativa. Suele ser recomendable crear una carpeta de imágenes que se encuentre en el mismo directorio que los archivos CSS y que almacene todas las imágenes utilizadas en el diseño de las páginas.

Así, las imágenes correspondientes al diseño de la página se mantienen separadas del resto de imágenes del sitio y el código CSS es más sencillo (por utilizar URL relativas) y más fácil de mantener (por no tener que actualizar URL absolutas en caso de que se cambie la estructura del sitio web).

Por otra parte, suele ser habitual indicar un color de fondo siempre que se muestra una imagen de fondo. En caso de que la imagen no se pueda mostrar o contenga errores, el navegador mostrará el color indicado (que debería ser, en lo posible, similar a la imagen) y la página no parecerá que contiene errores.

Si la imagen que se quiere mostrar es demasiado grande para el fondo del elemento, solamente se muestra la parte de imagen comprendida en el tamaño del elemento. Si la imagen es más pequeña que el elemento, CSS la repite horizontal y verticalmente hasta llenar el fondo del elemento.

Este comportamiento es útil para establecer un fondo complejo a una página web entera. El siguiente ejemplo utiliza una imagen muy pequeña para establecer un fondo complejo a toda una página:

Imagen original



Figura 4.20. Imagen original utilizada para el fondo de la página

Reglas CSS

```
body {
    background-image:url(imagenes/fondo.gif);
}
```

Resultado



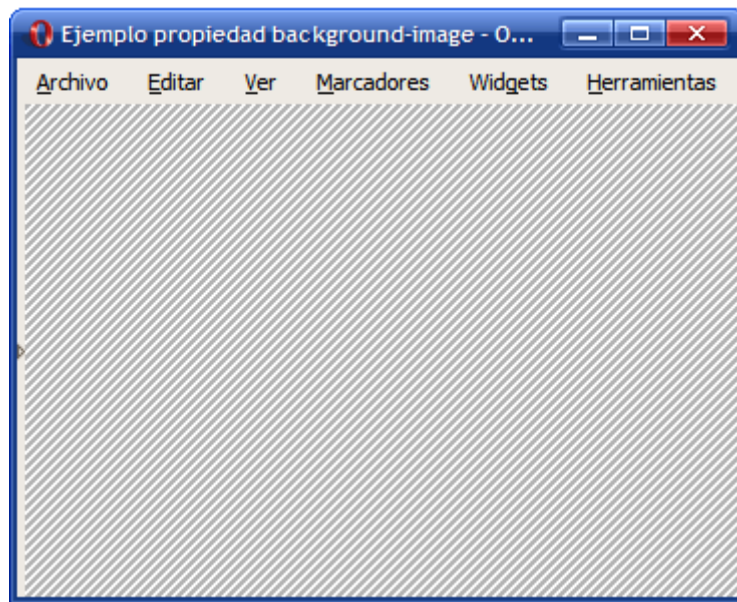


Figura 4.21. Página con una imagen de fondo

Con una imagen muy pequeña (y que por tanto, se puede descargar en muy poco tiempo) se consigue cubrir completamente el fondo de la página, con lo que se consigue un gran ahorro de ancho de banda.

En ocasiones, no es conveniente que la imagen de fondo se repita horizontal y verticalmente. Para ello, CSS introduce la propiedad `background-repeat` que permite controlar la forma de repetición de las imágenes de fondo.

<b>background-repeat</b>	Repetición de la imagen de fondo
<b>Valores</b>	<code>repeat</code>   <code>repeat-x</code>   <code>repeat-y</code>   <code>no-repeat</code>   <code>inherit</code>
<b>Se aplica a</b>	Todos los elementos
<b>Valor inicial</b>	<code>repeat</code>
<b>Descripción</b>	Controla la forma en la que se repiten las imágenes de fondo

El valor `repeat` indica que la imagen se debe repetir en todas direcciones y por tanto, es el comportamiento por defecto. El valor `no-repeat` muestra una sola vez la imagen y no se repite en ninguna dirección. El valor `repeat-x` repite la imagen sólo horizontalmente y el valor `repeat-y` repite la imagen solamente de forma vertical.

El sitio web <http://www.kottke.org/> utiliza el valor `repeat-x` para mostrar una imagen de fondo en la cabecera de la página:



Figura 4.22. Uso de repeat-x en la página de Kottke.org

Las reglas CSS definidas para la cabecera son:

```
#hdr {
  background: url("/images/ds.gif") repeat-x;
  width: 100%;
  text-align: center;
}
```

Por otra parte, el sitio web <http://veerle.duoh.com/> utiliza el valor `repeat-y` para mostrar el fondo de una columna de contenidos:

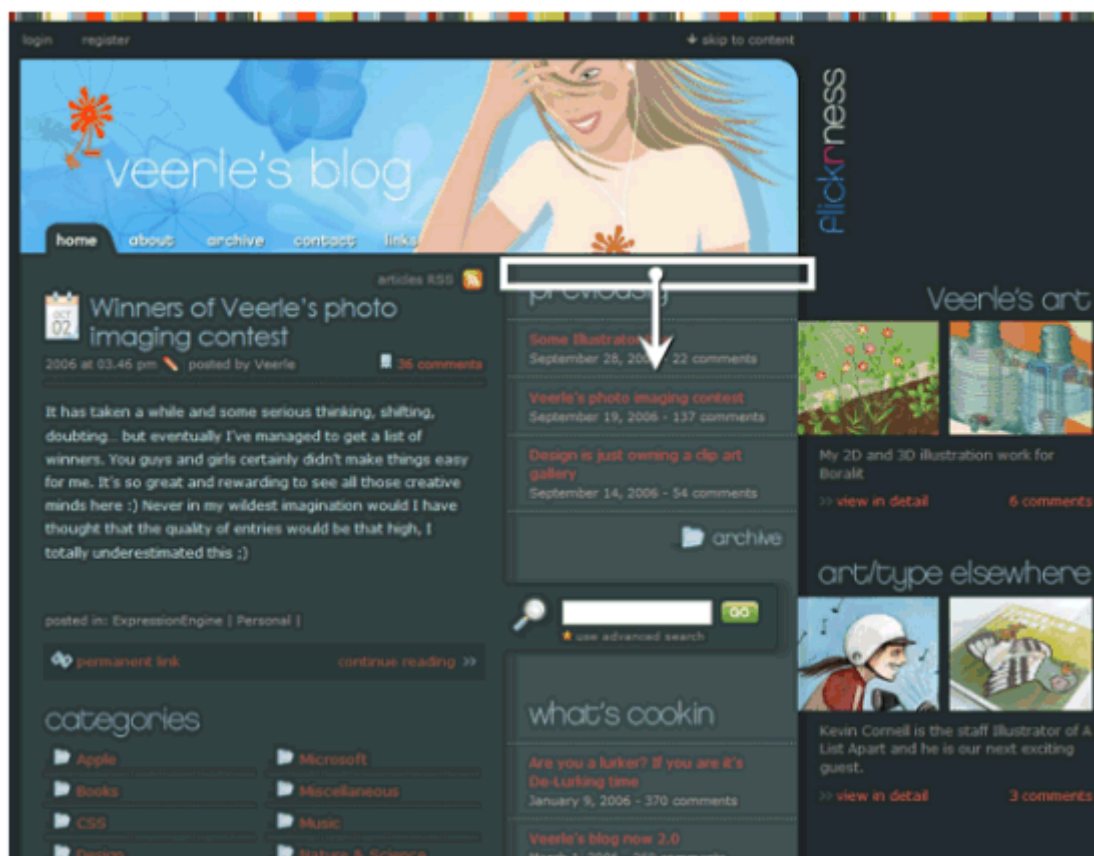


Figura 4.23. Uso de repeat-y en la página de Veerle.duoh.com

Las reglas CSS definidas para esa columna de contenidos son:

```
.wide #content-secondary {
  width: 272px;
  margin: 13px 0 0 0;
  position: relative;
  margin-left: -8px;
  background: url("../graphics/wide/bg-content-secondary.gif") repeat-
y;
}
```

Además de seleccionar el tipo de repetición de las imágenes de fondo, CSS permite controlar la posición de la imagen dentro del fondo del elemento mediante la propiedad `background-position`.

background-position	Posición de la imagen de fondo
<b>Valores</b>	( ( <porcentaje>   <medida>   left   center   right ) ( <porcentaje>   <medida>   top   center   bottom )? )   ( ( left   center   right )    ( top   center   bottom ) )   inherit
<b>Se aplica a</b>	Todos los elementos
<b>Valor inicial</b>	0% 0%
<b>Descripción</b>	Controla la posición en la que se muestra la imagen en el fondo del elemento

La propiedad `background-position` permite indicar la distancia que se desplaza la imagen de fondo respecto de su posición original situada en la esquina superior izquierda.

Si se indican dos porcentajes o dos medidas, el primero indica el desplazamiento horizontal y el segundo el desplazamiento vertical respecto del origen (situado en la esquina superior izquierda). Si solamente se indica un porcentaje o una medida, se considera que es el desplazamiento horizontal y al desplazamiento vertical se le asigna automáticamente el valor de 50%.

Cuando se utilizan porcentajes, su interpretación no es intuitiva. Si el valor de la propiedad `background-position` se indica mediante dos porcentajes `x% y%`, el navegador coloca el punto `(x%, y%)` de la imagen de fondo en el punto `(x%, y%)` del elemento.

Las palabras clave permitidas son equivalentes a algunos porcentajes significativos: `top = 0%`, `left = 0%`, `center = 50%`, `bottom = 100%`, `right = 100%`.

CSS permite mezclar porcentajes y palabras clave, como por ejemplo `50% 2cm`, `center 2cm`, `center 10%`.

Si se utilizan solamente palabras clave, el orden es indiferente y por tanto, es equivalente indicar `top left` y `left top`.

El siguiente ejemplo muestra una misma imagen de fondo posicionada de tres formas diferentes:

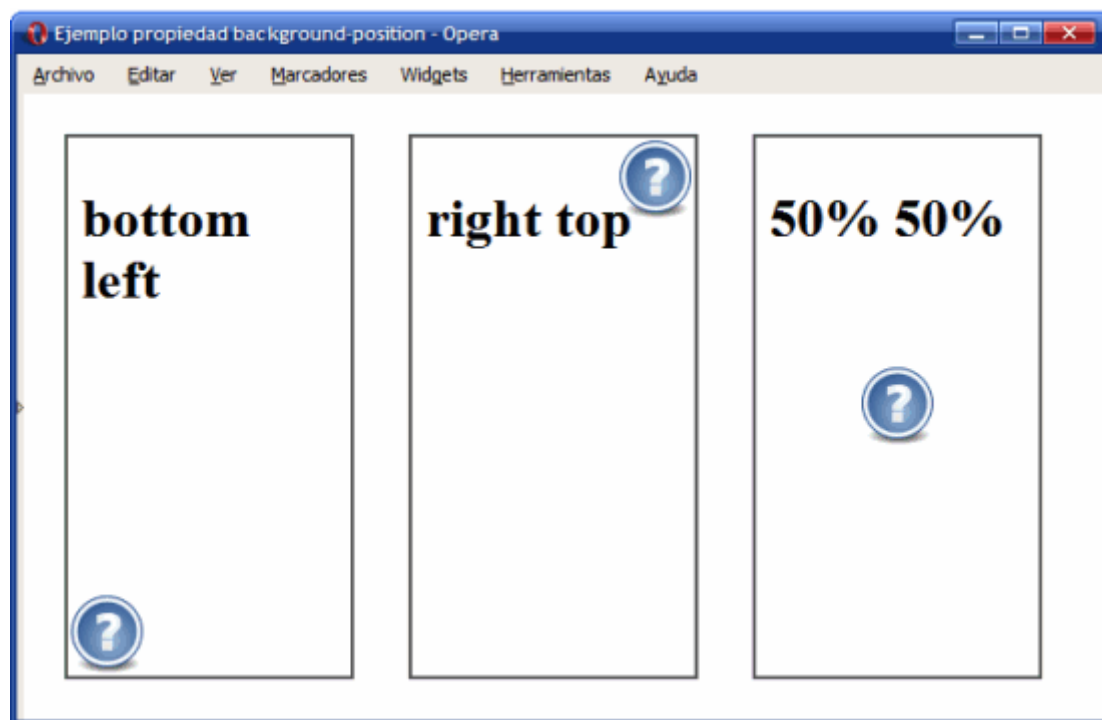


Figura 4.24. Ejemplo de propiedad `background-position`

Las reglas CSS del ejemplo anterior se muestran a continuación:

```
#caja1 {
  background-image: url("images/help.png");
  background-repeat: no-repeat;
  background-position: bottom left;
}
#caja2 {
  background-image: url("images/help.png");
  background-repeat: no-repeat;
  background-position: right top;
}
#caja3 {
  background-image: url("images/help.png");
  background-repeat: no-repeat;
  background-position: 50% 50%;
}

<div id="caja1"><h1>bottom left</h1></div>
<div id="caja2"><h1>right top</h1></div>
<div id="caja3"><h1>50% 50%</h1></div>
```

Opcionalmente, se puede indicar que el fondo permanezca fijo cuando la ventana del navegador se desplaza mediante las barras de *scroll*. Se trata de un comportamiento que en general no es deseable y que algunos navegadores no soportan correctamente. La propiedad que controla este comportamiento es `background-attachment`.

<b>background-attachment</b>	Comportamiento de la imagen de fondo
<b>Valores</b>	<code>scroll</code>   <code>fixed</code>   <code>inherit</code>
<b>Se aplica a</b>	Todos los elementos
<b>Valor inicial</b>	<code>scroll</code>
<b>Descripción</b>	Controla la forma en la que se visualiza la imagen de fondo: permanece fija cuando se hace scroll en la ventana del navegador o se desplaza junto con la ventana

Para hacer que una imagen de fondo se muestre fija al desplazar la ventana del navegador, se debe añadir la propiedad `background-attachment: fixed`.

Por último, CSS define una propiedad de tipo *"shorthand"* para indicar todas las propiedades de los colores e imágenes de fondo de forma directa. La propiedad se denomina `background` y es la que generalmente se utiliza para establecer las propiedades del fondo de los elementos.

<b>background</b>	Fondo de un elemento
<b>Valores</b>	( <background-color>    <background-image>    <background-repeat>    <background-attachment>    <background-position> )   <code>inherit</code>
<b>Se aplica a</b>	Todos los elementos
<b>Valor inicial</b>	-
<b>Descripción</b>	Establece todas las propiedades del fondo de un elemento

El orden en el que se indican las propiedades es indiferente, aunque en general se sigue el formato indicado de color, url de imagen, repetición y posición.

El siguiente ejemplo muestra la ventaja de utilizar la propiedad `background`:

```
/* Color e imagen de fondo de la página mediante una propiedad shorthand */
body { background: #222d2d url(./graphics/colorstrip.gif) repeat-x 0 0; }

/* La propiedad shorthand anterior es equivalente a las siguientes propiedades */
body {
    background-color: #222d2d;
    background-image: url("./graphics/colorstrip.gif");
    background-repeat: repeat-x;
    background-position: 0 0;
}
```

La propiedad `background` permite asignar todos o sólo algunos de todos los valores que se pueden definir para los fondos de los elementos:

```
background: url("./graphics/wide/bg-content-secondary.gif") repeat-y;

background: url("./graphics/wide/footer-content-secondary.gif") no-repeat bottom left;

background: transparent url("./graphics/navigation.gif") no-repeat 0 -27px;

background: none;

background: #293838 url("./graphics/icons/icon-permalink-big.gif") no-repeat center left;
```