

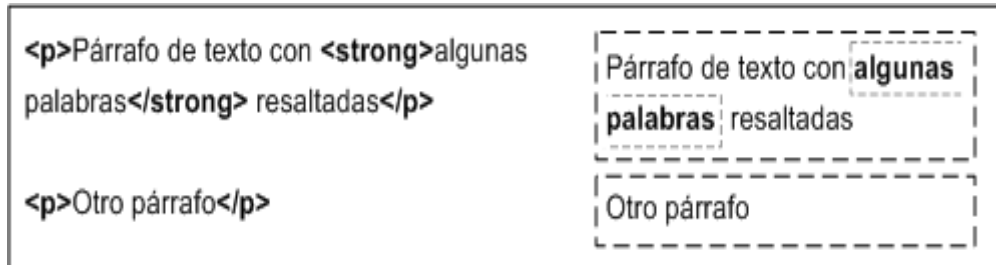
TEMA 7. MODELO DE CAJAS.

1	MODELO DE CAJAS.....	2
2	ANCHO Y ALTO.....	3
2.1	ANCHO.....	3
2.2	ALTO.....	3
3	MARGEN.....	3
4	RELLENO.....	7
5	BORDES.....	8
5.1	ANCHO DEL BORDE.....	8
5.2	COLOR DEL BORDE.....	9
5.3	ESTILO DEL BORDE.....	10
5.4	PROPIEDADES SHORTHAND.....	11
6	OUTLINE.....	12
7	MARGEN, RELLENO, BORDES Y MODELO DE CAJAS.....	13
8	ELEMENTOS EN LÍNEA Y ELEMENTOS EN BLOQUE.....	14

1 MODELO DE CAJAS

El modelo de cajas o "*box model*" es una de las características más importante de CSS. El modelo de cajas nos permite crear elementos que se representen mediante cajas rectangulares.

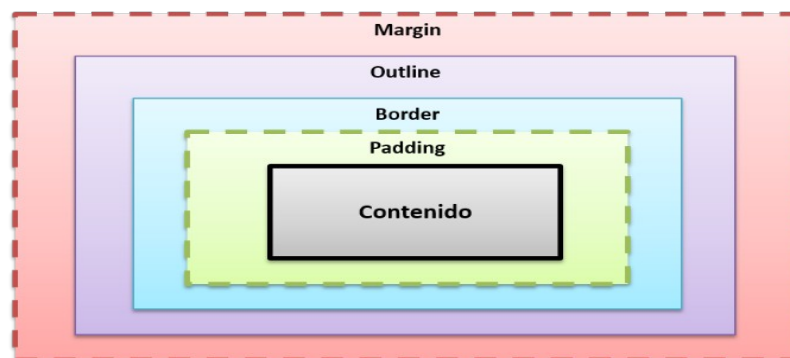
Cada vez que se inserta una etiqueta HTML, se crea una nueva caja rectangular que encierra los contenidos de ese elemento.



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.

Los navegadores crean y colocan las cajas de forma automática, pero CSS permite modificar todas sus características.



Box model en CSS

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.
- ✓ Outline: línea que se sitúa encima del borde. **Lo vemos aquí, pero no forma parte del modelo de cajas.**
- ✓ Margen (margin): separación entre una caja y el resto de las 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).

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.

2 ANCHO Y ALTO

2.1 ANCHO

La propiedad CSS que controla el ancho de una caja es width.

PROPIEDAD	WIDTH
Valores	unidad de medida porcentaje auto
Descripción	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 class="lateral">  
...  
</div>
```

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

2.2 ALTO

La propiedad CSS que controla la altura de una caja es height.

PROPIEDAD	HEIGHT
Valores	unidad de medida porcentaje auto
Descripción	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 class="cabecera">  
...  
</div>
```

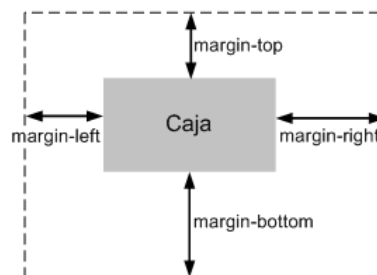
CSS define otras dos propiedades relacionadas con la altura de los elementos: `min-height` y `max-height`, que se verán más adelante.

3 MARGEN

La propiedad para controlar el margen es *margin* que es un *shorthand* que define cuatro propiedades para controlar cada uno de los márgenes horizontales y verticales de un elemento.

PROPIEDADES	MARGIN-TOP, MARGIN-RIGHT, MARGIN-BOTTOM, MARGIN-LEFT
Valores	unidad de medida porcentaje auto
Descripción	Establece cada uno de los márgenes horizontales y verticales de un elemento

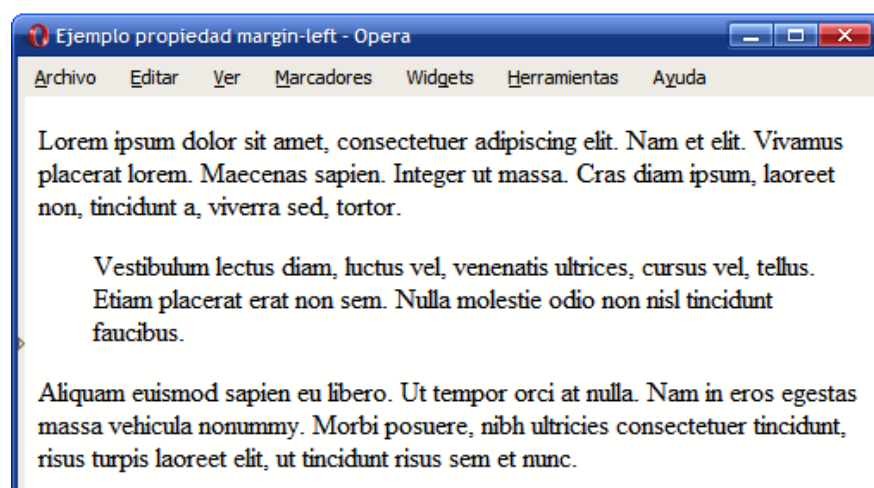
Cada una de las propiedades establece la separación entre el borde lateral de la caja y el resto de las cajas adyacentes:



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

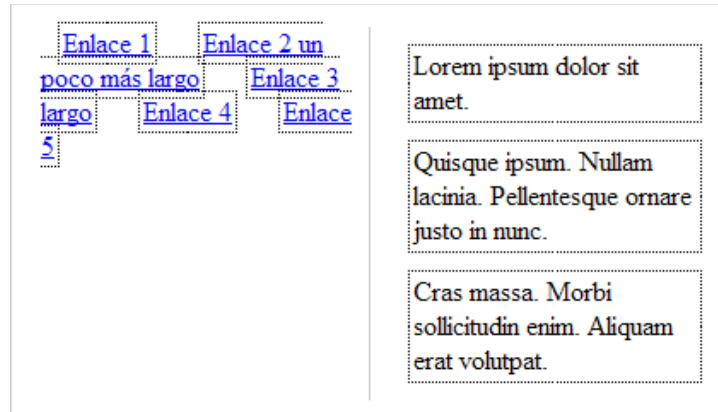
```
.destacado {  
  margin-left: 2em;  
}  
<p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nam et elit.  
Vivamus placerat lorem. Maecenas sapien. Integer ut massa. Cras diam ipsum,  
laoreet non, tincidunt a, viverra sed, tortor.</p>  
<p class="destacado">Vestibulum lectus diam, luctus 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:



Ejemplo de propiedad margin-left

Recuerda que 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, tanto a los elementos en línea como a los elementos de bloque.



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.

Además de las cuatro propiedades que controlan cada uno de los márgenes del elemento, CSS define una propiedad especial que permite establecer los cuatro márgenes de forma simultánea. Estas propiedades especiales se denominan "*propiedades 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.

PROPIEDAD	MARGIN
Valores	(unidad de medida porcentaje auto) {1, 4}
Descripción	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.

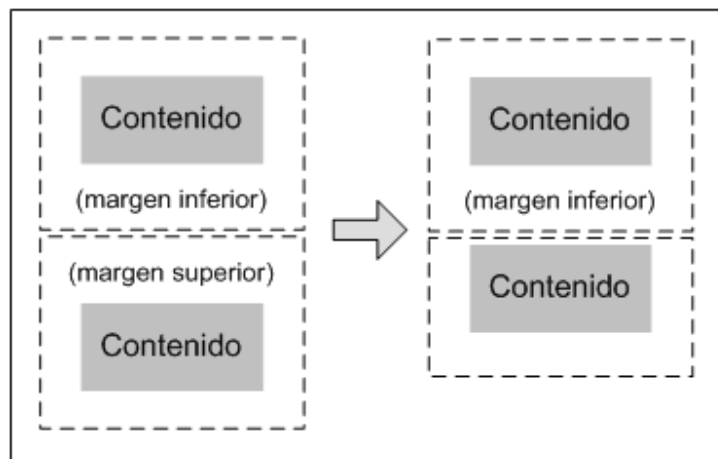
El ejemplo anterior de márgenes se puede reescribir utilizando la propiedad margin:

Código CSS original:

```
div img {
```

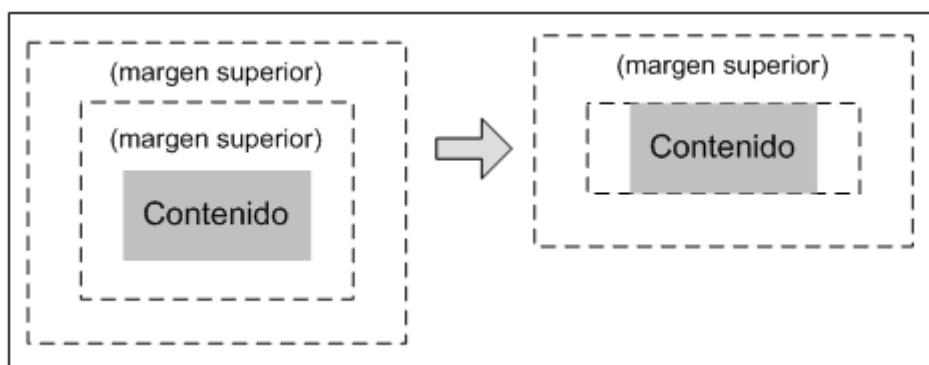
```
margin-top: .5em;  
margin-bottom: .5em;  
margin-left: 1em;  
margin-right: .5em;  
}  
Alternativa directa:  
div img {  
  margin: .5em .5em .5em 1em;  
}  
Otra alternativa:  
div img {  
  margin: .5em;  
  margin-left: 1em;  
}
```

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.



Fusión automática de los márgenes verticales

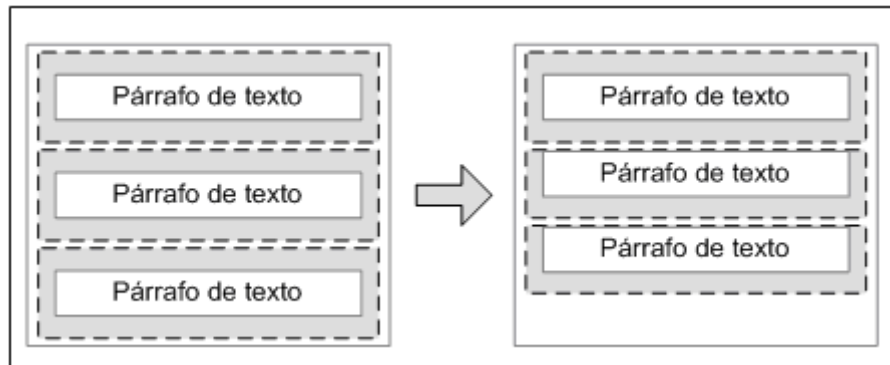
De la misma forma, si un elemento está contenido dentro de otro elemento, sus márgenes verticales se fusionan y resultan en un nuevo margen de la misma altura que el mayor margen de los que se han fusionado:



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.



Motivo por el que se fusionan automáticamente los márgenes verticales

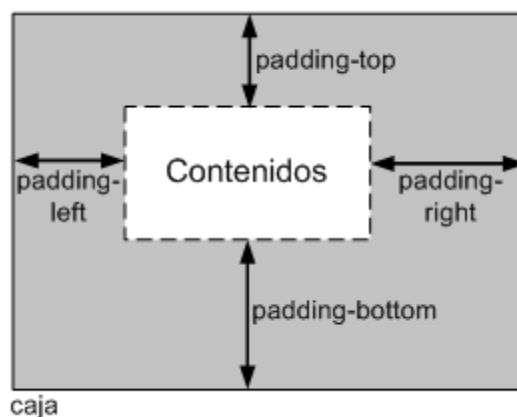
En el caso de un elemento que se encuentra en el interior de otro y sus márgenes se fusionan de forma automática, se puede evitar este comportamiento añadiendo un pequeño relleno (`padding: 1px`) o un borde (`border: 1px solid transparent`) al elemento contenedor.

4 RELLENO

La propiedad para controlar el relleno es *padding* que es un *shorthand* que define cuatro propiedades que nos permiten controlar la distancia entre el contenido y el borde.

PROPIEDADES	PADDING-TOP, PADDING-RIGHT, PADDING-BOTTOM, PADDING-LEFT
Valores	unidad de medida porcentaje
Descripción	Establece cada uno de los rellenos horizontales y verticales de un elemento

Cada una de estas propiedades establece la separación entre el contenido y los bordes laterales de la caja del elemento:



Las cuatro propiedades relacionadas con los rellenos

Como sucede con los márgenes, CSS también define una propiedad de tipo "*shorthand*" llamada *padding* para establecer los cuatro rellenos de un elemento de forma simultánea.

PROPIEDAD	PADDING
Valores	(unidad de medida porcentaje) {1, 4}
Descripción	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 */
```

5 BORDES

CSS permite modificar el aspecto de cada uno de los cuatro bordes de la caja de un elemento. Para cada borde se puede establecer su anchura o grosor, su color y su estilo, por lo que en total CSS define 20 propiedades relacionadas con los bordes.

5.1 ANCHO DEL BORDE

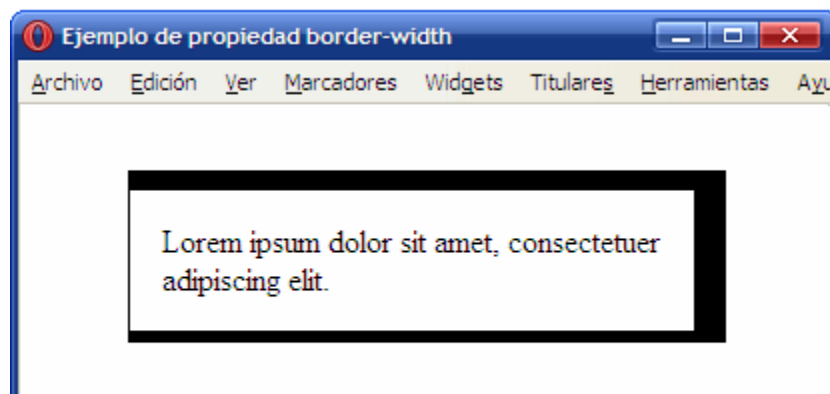
El ancho de los bordes se controla con las cuatro propiedades siguientes:

PROPIEDAD S	BORDER-TOP-WIDTH, BORDER-RIGHT-WIDTH, BORDER-BOTTOM-WIDTH, BORDER-LEFT-WIDTH
Valores	(unidad de medida thin medium thick)
Descripción	Establece la anchura de cada uno de los cuatro bordes de los elementos

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

La unidad de medida más habitual para establecer el grosor de los bordes es el píxel, ya que es la que permite un control más preciso sobre el grosor. Las palabras clave apenas se utilizan, ya que el estándar CSS no indica explícitamente el grosor al que equivale cada palabra clave, por lo que pueden producirse diferencias visuales entre navegadores. Así, por ejemplo, el grosor medium equivale a 4px en algunas versiones de Internet Explorer y a 3px en el resto de navegadores.

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



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 de forma simultánea la anchura de todos los bordes de una caja, es necesario utilizar una propiedad "shorthand" llamada border-width:

PROPIEDAD	BORDER-WIDTH
D	
Valores	(unidad de medida thin medium thick) {1, 4}
Descripción	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.

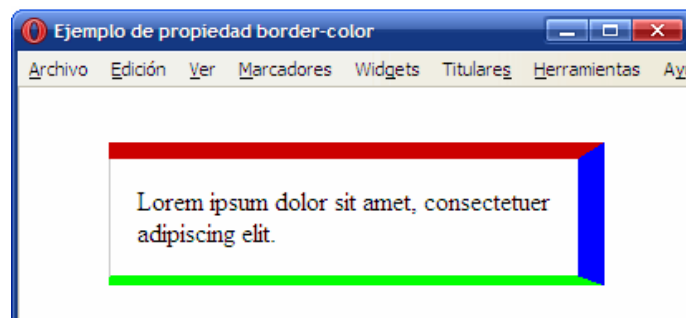
Si se indican tres valores, el primero se aplica al borde superior, el segundo se aplica al borde izquierdo y derecho y el tercer valor se aplica al borde inferior. Si se indican los cuatro valores, el orden de aplicación es superior, derecho, inferior e izquierdo.

5.2 COLOR DEL BORDE

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

PROPIEDAD	BORDER-TOP-COLOR, BORDER-RIGHT-COLOR, BORDER-BOTTOM-COLOR,BORDER-LEFT-COLOR
S	
Valores	color transparent
Descripción	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:



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;  
}
```

CSS incluye una propiedad "*shorthand*" llamada `border-color` para establecer de forma simultánea el color de todos los bordes de una caja:

PROPIEDAD	BORDER-COLOR
Valores	(color transparent) {1, 4}
Descripción	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 las de la propiedad `border-width`.

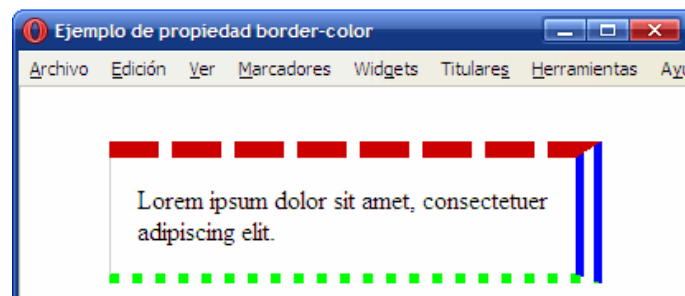
5.3 ESTILO DEL BORDE

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

PROPIEDADES	BORDER-TOP-STYLE, BORDER-RIGHT-STYLE, BORDER-BOTTOM-STYLE,BORDER-LEFT-STYLE
Valores	none hidden dotted dashed solid double groove ridge inset outset
Descripción	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:

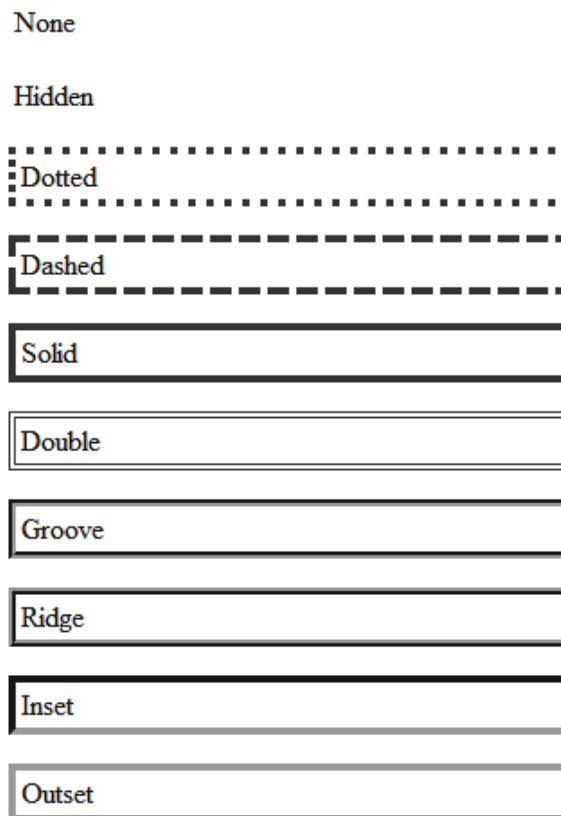


Ejemplo de propiedad `border-style`

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:



Tipos de bordes definidos por CSS

Para establecer de forma simultánea los estilos de todos los bordes de una caja, es necesario utilizar la propiedad *"shorthand"* llamada border-style:

PROPIEDAD	BORDER-STYLE
D	
Valores	(none hidden dotted dashed solid double groove ridge inset outset) {1, 4}
Descripción	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"*.

5.4 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 simultánea. CSS incluye una propiedad *"shorthand"* para cada uno de los cuatro bordes y una propiedad *"shorthand"* global.

PROPIEDAD	BORDER-TOP, BORDER-RIGHT, BORDER-BOTTOM, BORDER-LEFT
S	
Valores	(unidad de medida_borde color_borde estilo_borde)
Descripción	Establece el estilo completo de cada uno de los cuatro bordes de los elementos

El significado de cada uno de los valores especiales es el siguiente:

- ✓ <medida_borde>: una medida CSS o bien thin, medium, thick.

- ✓ `<color_borde>`: un color de CSS o la palabra clave `transparent`
- ✓ `<estilo_borde>`: `none`, `hidden`, `dotted`, `dashed`, `solid`, `double`, `groove`, `ridge`, `inset`, `outset`.

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

PROPIEDAD	BORDER
D	
Valores	(unidad de medida_borde color_borde estilo_borde)
Descripción	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;  
}
```

6 OUTLINE

Aunque outline es una propiedad css, que no pertenece al modelo de cajas, vamos a ver su funcionamiento aquí.

Es la línea que hay entre el borde y el margen.

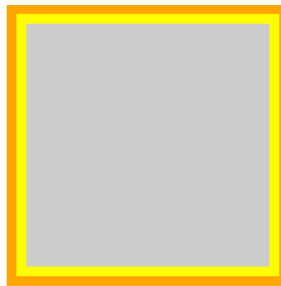
Se comporta igual que el borde con dos diferencias:

- ✓ El tamaño no se tiene en cuenta añade al tamaño de la caja como hace borde (por defecto), es decir, si tenemos una caja que mide 100px y le ponemos un outline, la caja va a seguir midiendo 100px.
- ✓ Tenemos una propiedad nueva que outline.offset, que nos permite establecer una separación entre el borde y el outline.

Por lo demás se comporta igual que el borde tiene las mismas propiedades para el estilo, el grosor y el color.

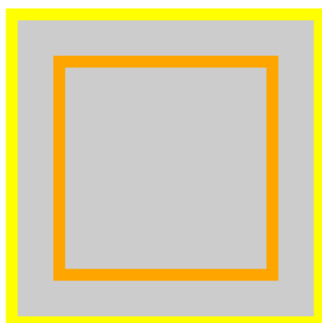
Se coloca justo por fuera del borde, en el caso de que el borde no exista se coloca, después del padding y si el padding no existe se coloca justo después del contenido.

```
h1 {  
  outline: 5px solid orange;  
}
```

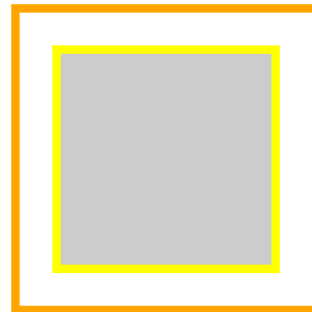


Tenemos una línea alrededor del borde esa es la propiedad outline.

En el caso de que queramos darle una separación entre el borde y outline, podemos utilizar la propiedad outline-offset, que nos va a permitir indicar la separación que queremos entre ambos. La separación la podemos dar con un valor positivo o negativo. Si lo hacemos con un valor negativo nuestro offset aparecerá en este caso dentro del borde y si le damos un valor positivo aparecerá por fuera del borde



```
h1 {  
  outline: 5px solid orange;  
  outline-offset : -30px;  
}
```



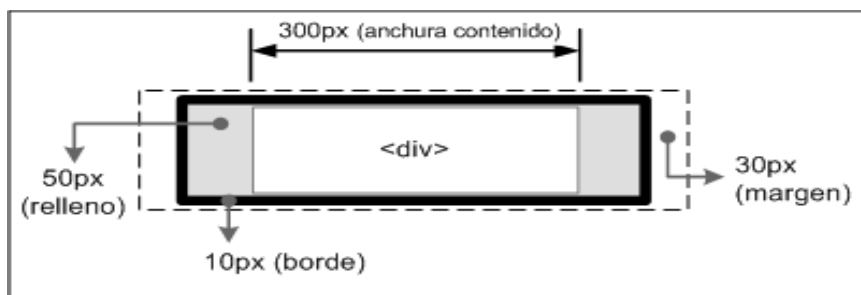
```
h1 {  
  outline: 5px solid orange;  
  outline-offset : 30px;  
}
```

7 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 también se añaden todos sus márgenes, rellenos y bordes:



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:

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

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*.

Para evitar los posibles problemas que nos pueda traer incrementar el tamaño definido para la caja con los elementos del modelo de cajas, podemos hacer que la caja tenga siempre el tamaño definido con **height** y **width** y que los elementos del modelo de caja no incrementen el tamaño de la caja. Para ellos tenemos la propiedad **box-sizing**.

```
.caja {  
  box-sizing: border-box;  
}
```

O bien

```
.caja {  
  box-sizing: content-box;  
}
```

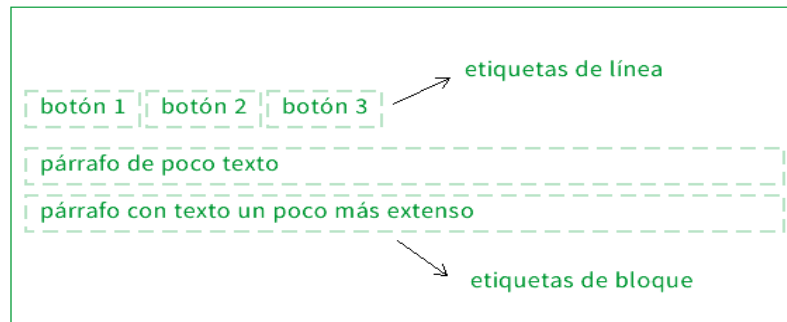
El valor *border-box*, hace que los elementos del modelo de cajas no incrementen el tamaño de la caja, es decir, la caja va a tener las medidas que hayamos dados en las propiedades **height** y **width**.

El valor *content-box*, es el valor por defecto, y en este caso los elementos del modelo de cajas si incrementan el tamaño de la caja.

8 ELEMENTOS EN LÍNEA Y ELEMENTOS EN BLOQUE

Al crear una caja, esta puede tener dos comportamientos en función de la etiqueta HTML utilizada, puede tener un comportamiento en bloque o en línea. **Por lo que decimos que existen dos tipos de elementos (o etiquetas) elementos en bloque y elementos en línea.**

Un elemento en línea ocupa el espacio mínimo necesario en horizontal, y permite que otro elemento se coloque a su lado. En cambio, un elemento en bloque ocupa todo el ancho disponible y no permite que otro elemento se coloque a su lado.



Elementos en línea y elementos en bloque.

ELEMENTOS EN LÍNEA:

big, small, br, acronym, cite, code, dfn, em, kbd, strong, a, br, img, script, span, sub, sup, button, input, label, select, textarea.

ELEMENTOS DE BLOQUE

address, blockquote, center, dir, div, dl, fieldset, form, h1, h2, h3, h4, h5, h6, hr, ol, p, pre, table, ul, dd, dt, li, tbody, td, tfoot, th, thead, tr, button.

Una característica a tener en cuenta es que un elemento en bloque tiene ancho y alto, mientras que un elemento en línea no.

Otra característica es que un elemento en línea sólo tiene margen horizontal, no tiene margen vertical, mientras que un elemento de bloque tiene margen horizontal y vertical.

Podemos convertir un elemento en línea en un elemento en bloque, para ello vamos a utilizar la propiedad display.

```
a{  
    display:block;  
}
```

También podríamos utilizar el valor inline-block que no permite mantener el elemento en línea, pero poder darle ancho, alto y todos los márgenes.

```
a{  
    display : inline-block;  
}
```

De la misma forma podemos convertir un elemento de bloque, en un elemento en línea.

```
div{  
    display : inline;  
}
```

Al convertir un elemento de bloque en elemento en línea, tomará las características de un elemento en línea, no tendrá ni ancho, ni alto, ni podremos darle márgenes verticales.

