

Introducción a CSS3



Cascading **Style Sheets**



CSS3

- El lenguaje HTML está limitado a la hora de aplicarle forma a un documento.
- Sus estructuras tienen poca flexibilidad a la hora de dar forma al contenido mostrado.
- HTML se creó originariamente para uso científico y posteriormente fue adoptado para el desarrollo web.
- Para “organizar” se utilizan elementos HTML en un uso diferente de su objetivo (tablas por ejemplo).
- Todos estos problemas dieron lugar al origen de CSS.



CSS3

- Principales características
 - Permite definir el estilo de cada elemento HTML de manera exacta.
 - Permite escalar tamaños en función del tamaño de la pantalla.
 - Aísla el contenido de la presentación.
 - Permite crear plantillas de estilos que pueden importarse en otros HTML.

CSS3

- Cómo se define un estilo en css:

Declaración

```
Selector {  
    propiedad: valor;  
    ...}
```

Ejemplo:

```
H1 {  
    color: #CC9900;  
}
```

CSS3

- Un selector es un identificador para saber sobre qué elemento HTML debemos aplicar el estilo.
- Existen diferentes tipos de selectores.
- Se pueden crear jerarquías de estilos.
- Podemos definir estilos por defecto para los elementos HTML estándar.
- Los estilos pueden sobrescribir a otro, el orden de sobreescritura es el mismo en el que se cargan los ficheros css o se lee el fichero.
- Si queremos que un estilo prevalezca sobre el resto, podemos indicarlo con **!important**

- Normalmente se le aplica un estilo por defecto a los elementos HTML para conformar una plantilla de estilo.
- La personalización definitiva se suele realizar haciendo uso de la propiedad "class".
- No es recomendado usar el selector por id. Sólo se recomienda cuando el componente, además de estilo realiza alguna funcionalidad javascript.
- Cuando se incluyan plantillas de estilo en un archivo HTML hay que asegurar que no incorporan estilos CSS que sobrescriban los anteriores.
- En caso de conflicto, se debe asegurar que lo estilos quedan situados por encima de los otros (aunque no siempre es posible)

Tipos de selectores:

- De elemento HTML

h1, table, div, span...

- De identificador

Todos los elementos HTML cuya propiedad "id" tenga un determinado valor, tendrán ese estilo.

- De clase

- **Todos los elementos HTML cuya propiedad "class" tenga un determinado valor tendrán ese estilo.**

- Los estilos CSS deben darse de alta en un archivo acorde para ello, no obstante:
 - Pueden declararse dentro de un HTML mediante la etiqueta `<style>`: **Estilos embebidos**
 - Pueden aplicarse directamente sobre un elemento concreto en la propiedad "style": **Estilos en línea**
- Lo correcto es llevar los estilos a un o unos archivos **css**, pero se permite añadir pequeños retoques directamente sobre el HTML: **Archivos externos**

CSS3

Estilos en línea

Una de las técnicas más simples para incorporar estilos CSS a un documento HTML es la de asignar los estilos dentro de las etiquetas por medio del atributo ***style***.

```
<!DOCTYPE html>
```

```
<html>
```

```
<body>
```

Esto es un encabezado

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod

```
<h1 style="color:blue; margin-left:30px;">
```

Esto es un encabezado

```
</h1>
```

```
<p>This is a paragraph.</p>
```

```
</body>
```

```
</html>
```

CSS3

Estilos embebidos

Una mejor alternativa es insertar los estilos en la cabecera del documento y luego usar referencias para afectar los elementos HTML correspondientes:

```
<!DOCTYPE html>
<html>
<head>
  <style>
    body {
      background-color: linen;
    }
    h1 {
      color: maroon;
      margin-left: 40px;
    }
  </style>
</head>
<body>
<h1>Encabezado</h1> <p> Lorem ipsum dolor </p> </body>  </html>
```

Encabezado

Lorem ipsum dolor sit amet, consectetur adipisicing elit.

Ejemplo: 1 Div Semantica Estilos

CSS3

Archivos externos

Declarar los estilos en la cabecera del documento ahorra espacio y vuelve al código más consistente y actualizable,, mediante el elemento **<link>** para insertar este archivo dentro de cada documento que los necesite:

<link rel="stylesheet" href="css/misestilos.css">

Este método nos permite cambiar los estilos por completo simplemente incluyendo un archivo diferente.

También nos permite modificar o adaptar nuestros documentos a cada circunstancia o dispositivo.

CSS3

```
<!DOCTYPE html>
<html lang="es">
<head>
<title>Este texto es el título del documento</title>
  <link rel="stylesheet" type="text/css" href="mystyle.css">
</head>
<body>
  <p>Mi texto</p>
</body>
</html>
```

CSS3 Referencias

Tipo referencia	Ejemplo
palabra clave del elemento	<p>Se afecta cada elemento de la misma clase en el documento. Por ejemplo, la siguiente regla cambiará los estilos de todos los elementos <code><p></code>:</p> <p>p { font-size: 20px }</p>
referencia por el atributo id	<p>id es un nombre que identifica al elemento. Esto significa que el valor de este atributo no puede ser duplicado. Para referenciar un elemento en particular usando el atributo id se usa el símbolo # al frente del valor :</p> <p>#texto1 { font-size: 20px }</p>
referencia por el atributo class	<p>La mayoría del tiempo, en lugar de utilizar el atributo id para propósitos de estilos es mejor utilizar class. Este atributo es más flexible y puede ser asignado a cada elemento HTML en el documento que comparte un diseño similar:</p> <p>.texto1 { font-size: 20px }</p>
Selector Universal	<p>* {margin: 0px; padding: 0px;}</p>

Referencia por *id*

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<style>
```

```
#para1 {
```

```
    text-align: center;
```

```
    color: blue;
```

```
}
```

```
</style>
```

```
</head>
```

```
<body>
```

Hello World!

Este párrafo no tiene estilo

```
<p id="para1">Hello World!</p>
```

```
<p>Este párrafo no tiene estilo</p>
```

```
</body>
```

```
</html>
```

Referencia por *class*

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<style>
```

```
.center {
```

```
    text-align: center;
```

```
    color: red;
```

```
}
```

```
</style>
```

```
</head>
```

```
<body>
```

Encabezado rojo y alineado al centro

Párrafo rojo y alineado al centro

```
<h1 class="center"> Encabezado rojo y alineado al  
centro</h1>
```

```
<p class="center">Párrafo rojo y alineado al centro</p>
```

```
</body>
```

```
</html>
```

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<style>
```

```
p.center {
```

```
    text-align: center;
```

```
    color: red;
```

```
}
```

```
</style>
```

```
</head>
```

```
<body>
```

Este encabezado no ha sido afectado

El párrafo es rojo alineado al centro

```
<h1 class="center">Este encabezado no ha sido  
afectado</h1>
```

```
<p class="center">El párrafo es rojo  
alineado al centro</p>
```

```
</body>
```

```
</html>
```


Los elementos HTML también pueden referirse a más de una clase.

```
<!DOCTYPE html>
<html>
<head>
<style>

p.center {
    text-align: center;
    color: blue;
}

p.large {
    font-size: 300%;
}
</style>
</head>
```

```
<body>

<h1 class="center">no hay estilo</h1>
<p class="center">párrafo azul alineado al
    centro</p>
<p class="center large"> Párrafo azul,
    alineado al centro y tamaño de letra
    grande</p>
</body>
</html>
```

no hay estilo

párrafo azul alineado al centro

Párrafo azul, alineado al centro y tamaño de letra grande

Los elementos HTML también se pueden agrupar

```
h1, h2, p {  
  text-align: center;  
  color: red;  
}
```

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<style>
```

```
h1, h2, p {  
  text-align: center;  
  color: green;  
}
```

```
</style>
```

```
</head>
```

```
<body>
```

```
<h1>Hello World!</h1>
```

```
<h2>Encabezado 2!</h2>
```

```
<p>Párrafo</p>
```

```
</body>
```

```
</html>
```

Hello World!

Encabezado 2!

Párrafo

Ejercicios

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<style>
```

```
</style>
```

```
</head>
```

```
<body>
```

```
<h1>This is a Heading</h1>
```

```
<p>This is a paragraph.</p>
```

```
<p>This is another paragraph.</p>
```

```
</body>
```

```
</html>
```

Cambiar los elementos <p> a rojo

Ejercicios

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<style>
```

Cambiar todos los elementos con el identificador *par1* a color azul

```
</style>
```

```
</head>
```

```
<body>
```

```
<h1>This is a Heading</h1>
```

```
<p id="par1">This is a paragraph.</p>
```

```
<p>This is another paragraph.</p>
```

```
</body>
```

```
</html>
```

Ejercicios

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<style>
```

Cambiar todos los elementos identificados con la clase *colortext* a verde

```
</style>
```

```
</head>
```

```
<body>
```

```
<h1>This is a Heading</h1>
```

```
<p>This is a paragraph.</p>
```

```
<p class="colortext">This is another  
paragraph.</p>
```

```
<p class="colortext">This is also a  
paragraph.</p>
```

```
</body>
```

```
</html>
```

Ejercicios

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<style>
```

Cambiar los elementos *p* y *h1* a rojo. Agrupa los elementos para minimizar código

```
</style>
```

```
</head>
```

```
<body>
```

```
<h1>This is a heading</h1>
```

```
<h2>This is a smaller heading</h2>
```

```
<p>This is a paragraph.</p>
```

```
<p>This is another paragraph.</p>
```

```
</body>
```

```
</html>
```

Más ejemplos

```
body {  
    background-color: lightblue;  
}  
  
h1 {  
    color: navy;  
    margin-left: 20px;  
}
```

```
<h1 style="background-color:DodgerBlue;">Hello World</h1>  
<p style="background-color:Tomato;">Lorem ipsum...</p>
```

```
<h1 style="color:Tomato;">Hello World</h1>  
<p style="color:DodgerBlue;">Lorem ipsum...</p>  
<p style="color:MediumSeaGreen;">Ut wisi enim...</p>
```

```
<h1 style="border:2px solid Tomato;">Hello World</h1>  
<h1 style="border:2px solid DodgerBlue;">Hello World</h1>  
<h1 style="border:2px solid Violet;">Hello World</h1>
```

Más ejemplos

```
body {  
    background-image: url("paper.gif");  
}
```

```
body {  
    background-image: url("gradient_bg.png");  
    background-repeat: repeat-x;  
}
```

```
body {  
    background-image: url("img_tree.png");  
    background-repeat: no-repeat;  
}
```

```
body {  
    background-image: url("img_tree.png");  
    background-repeat: no-repeat;  
    background-position: right top;  
}
```




```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<style>
```

```
body {
```

```
    background-image: url("paper.gif");
```

```
    background-repeat: repeat-y;
```

```
}
```

```
</style>
```

```
</head>
```

```
<body>
```

```
<h1>Propiedad background-repeat </h1>
```

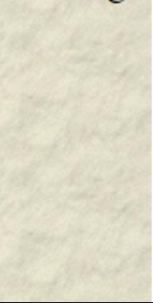
```
<p>la imagen se repite solo  
verticalmente.</p>
```

```
</body>
```

```
</html>
```

Propiedad background-repeat

la imagen se repite solo verticalmente.





```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<style>
```

```
body {
```

```
    background-image: url("paper.gif");
```

```
    background-repeat: no-repeat;
```

```
    background-position: right top;
```

```
}
```

```
</style>
```

```
</head>
```

```
<body>
```

```
<h1>Propiedad background-repeat </h1>
```


```
<p>la imagen se repite solo  
verticalmente.</p>
```

```
</body>
```

```
</html>
```

Propiedad background-repeat

la imagen background





CSS3

Modelo de caja

CSS tiene un conjunto predeterminado de propiedades destinado a sobrescribir los estilos provistos por navegadores y obtener la organización deseada. Estas propiedades no son específicas, tienen que ser combinadas para formar reglas que luego serán usadas para agrupar cajas y obtener la correcta disposición en pantalla. La combinación de estas reglas es normalmente llamada modelo o sistema de disposición. Todas estas reglas aplicadas juntas constituyen lo que se llama un **modelo de caja**.

CSS3

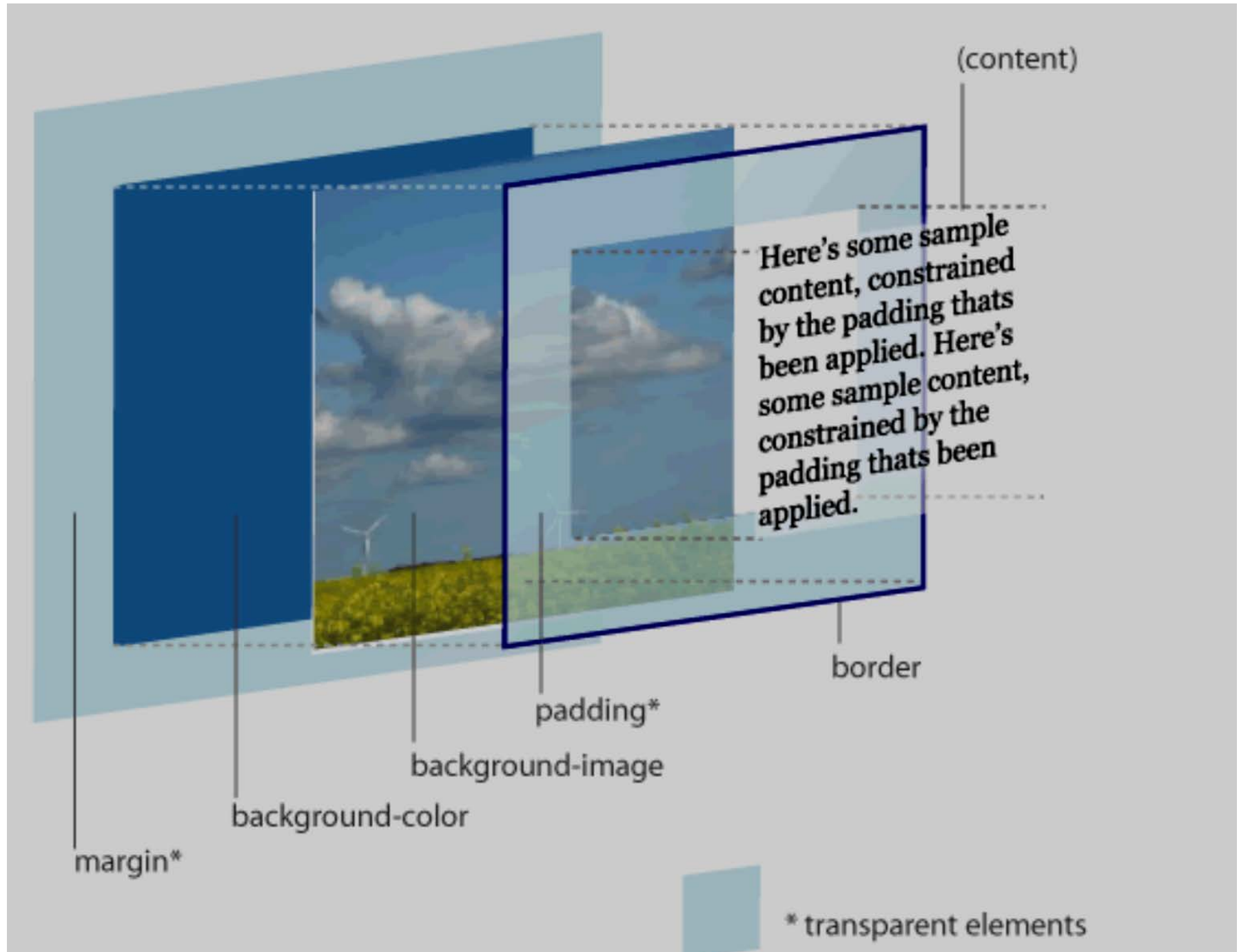
Modelo de caja


Con elementos `<div>` y CSS se pueden crear cajas en la pantalla, posicionar estas cajas a un lado o a otro y darles un tamaño, color o borde específico entre otras características.

CSS provee propiedades específicas que permiten organizar las cajas acorde a necesidades específicas. Estas propiedades son lo suficientemente poderosas como para crear un modelo de caja que se transformó en lo que se conoce como Modelo de Caja Tradicional.

Ejemplo [2PantillaBasica.html](#)
[estiloCaja.css](#)

Jerarquía del modelo de caja





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.



width propiedad que controla la anchura de la caja de los elementos se denomina .

* Se aplica a todos los elementos, salvo los elementos en línea que no sean imágenes, las filas de tabla y los grupos de filas de tabla

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



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

* Se aplica a todos los elementos, salvo los elementos en línea que no sean imágenes, las filas de tabla y los grupos de filas de tabla

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.

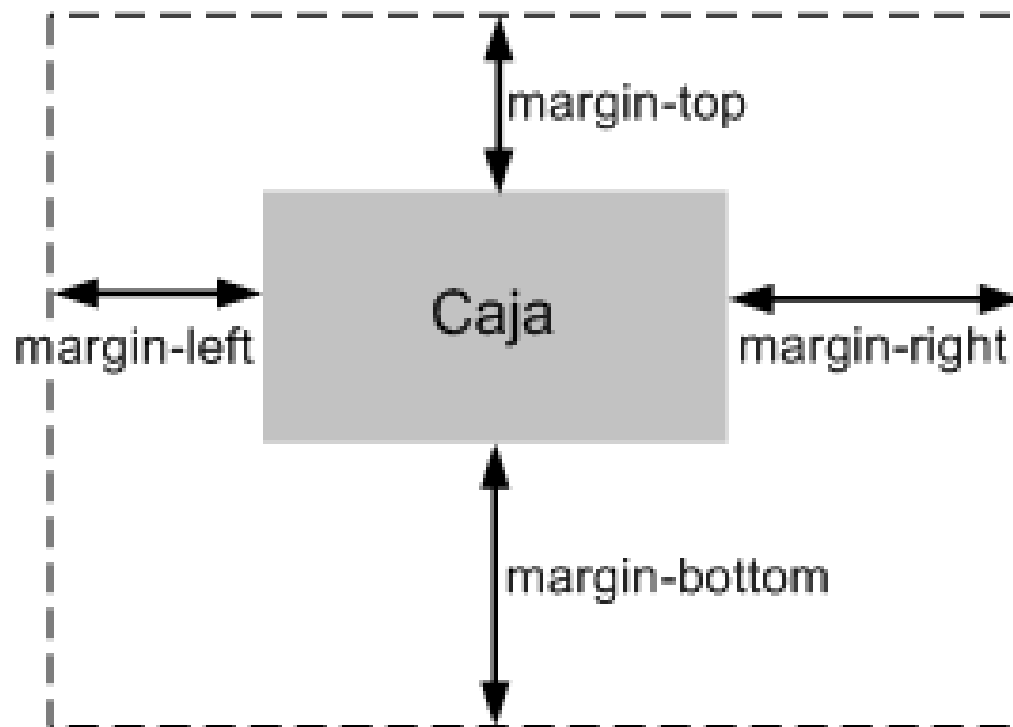
El valor inherit indica que la altura del elemento se hereda de su elemento padre. El valor auto, que es el que se utiliza si no se establece de forma explícita un valor a esta propiedad, indica que el navegador debe calcular automáticamente la altura del elemento, teniendo en cuenta sus contenidos y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

Margen y relleno

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

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



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

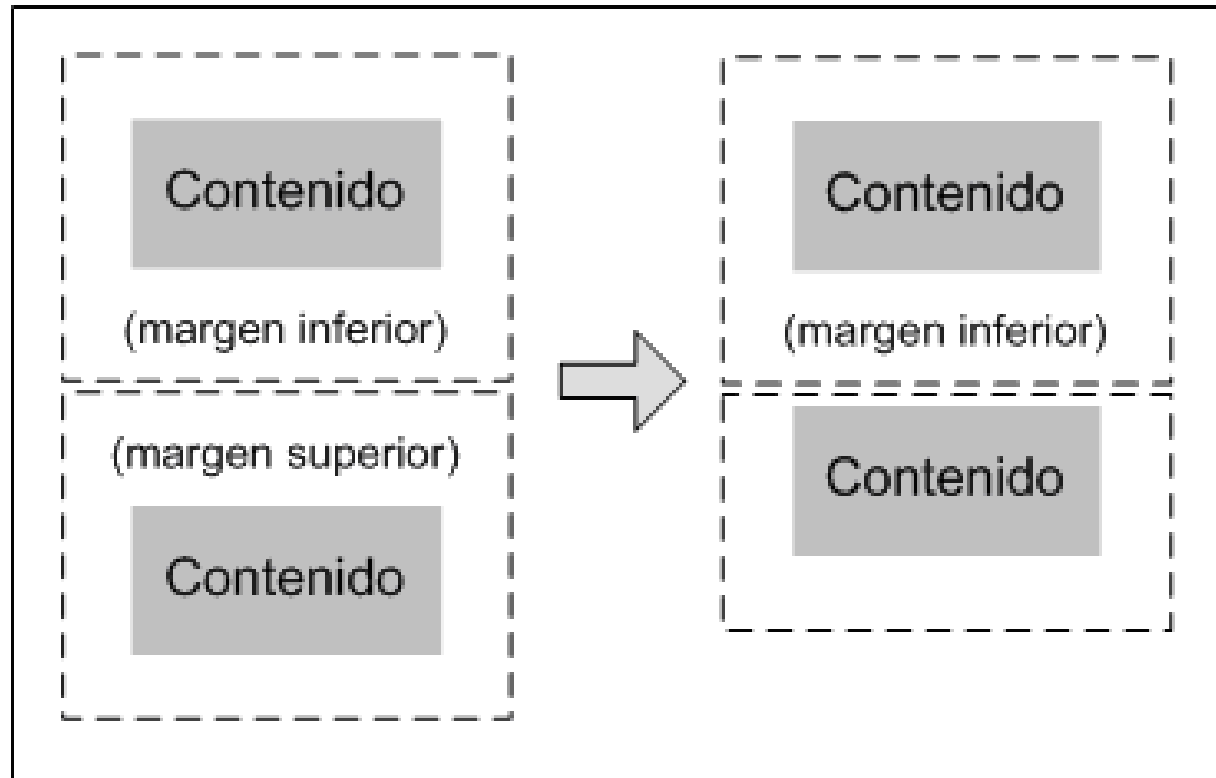
La propiedad que permite definir de forma simultanea los cuatro márgenes se denomina **margin**.

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

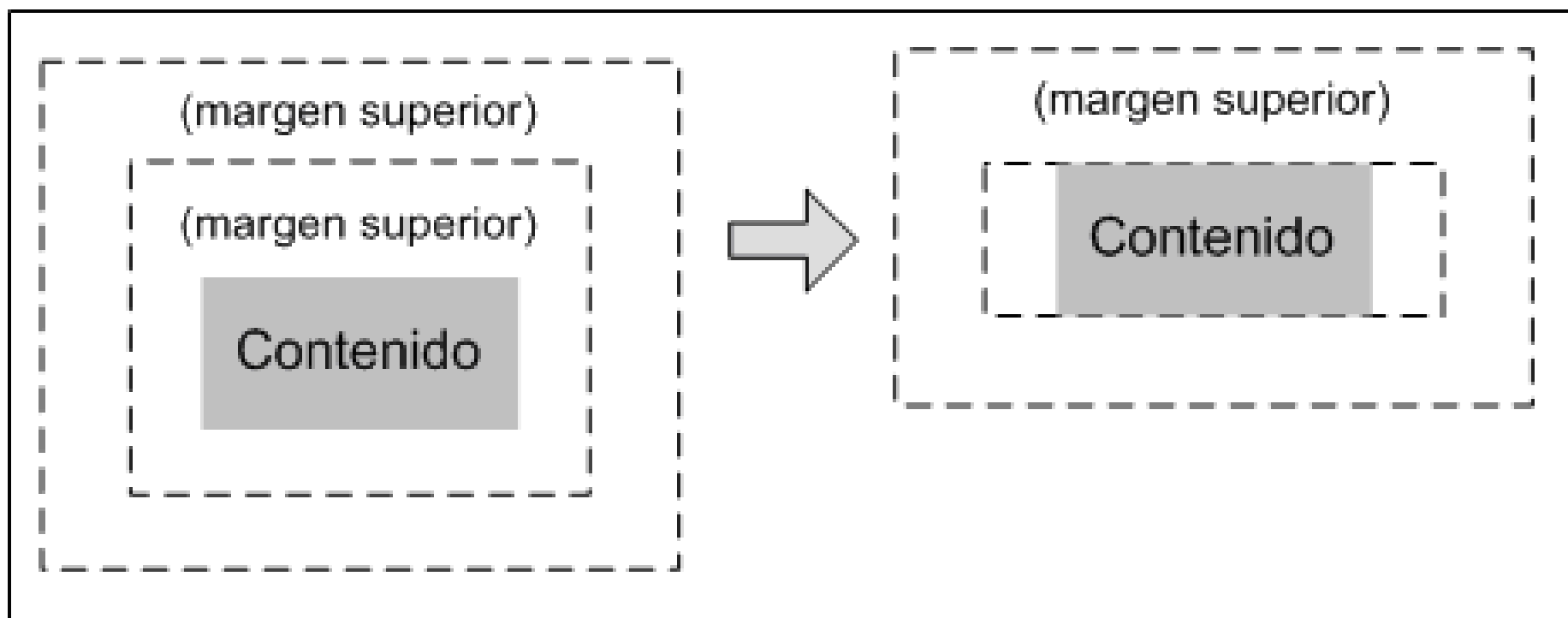
```
div img {  
margin: .5em;  
margin-left: 1em;  
}
```

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

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.

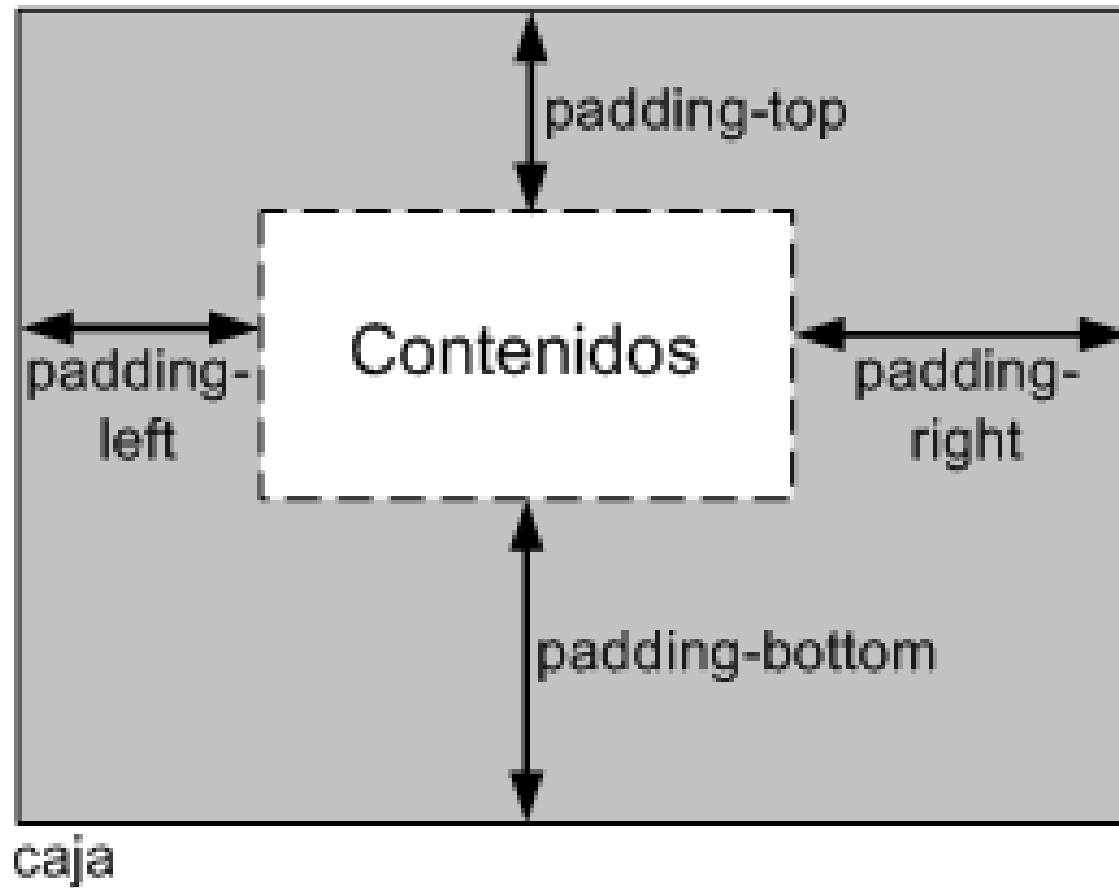


Si un elemento está contenido dentro de otro elemento, sus márgenes verticales se fusionan y resultan en un nuevo margen de la misma altura que el mayor margen de los que se han fusionado:



Relleno

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



La propiedad el padding admite entre uno y cuatro valores, con el mismo significado que de la propiedad margin.

```
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 */
```

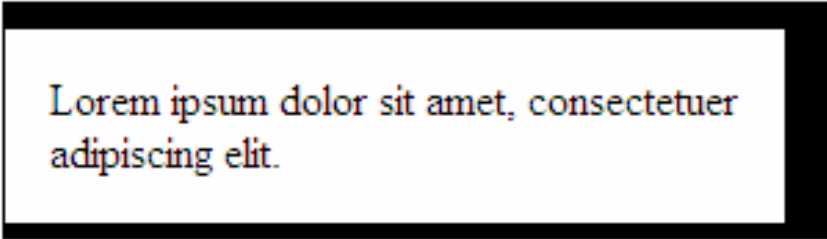
```
body {padding: 1em 2em 3em 4em} /* Superior = 1em,  
derecho = 2em, inferior = 3em, izquierdo = 4em */
```

Bordes


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.

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



Lorem ipsum dolor sit amet, consectetur
adipiscing elit.



Si se quiere establecer de forma simultánea la anchura de todos los bordes de una caja, es necesario utilizar la propiedad `border-width`: La propiedad `border-width` permite indicar entre uno y cuatro valores.

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

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

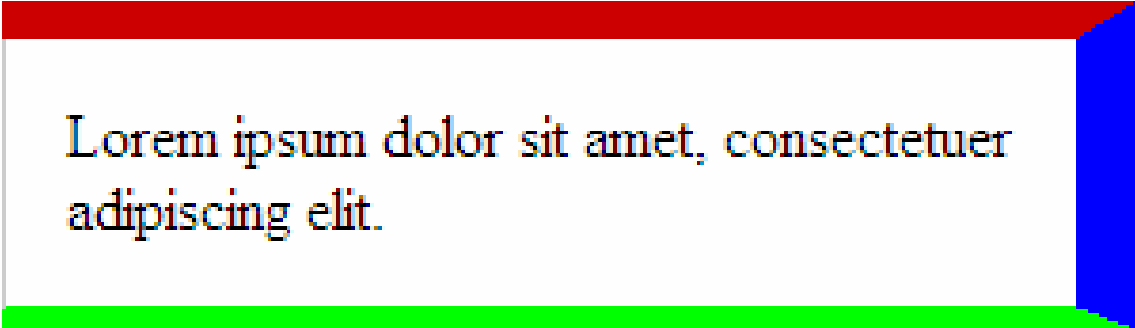
```
p { border-width: thin thick medium } /* thin thick medium  
thick */
```

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


Color

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

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



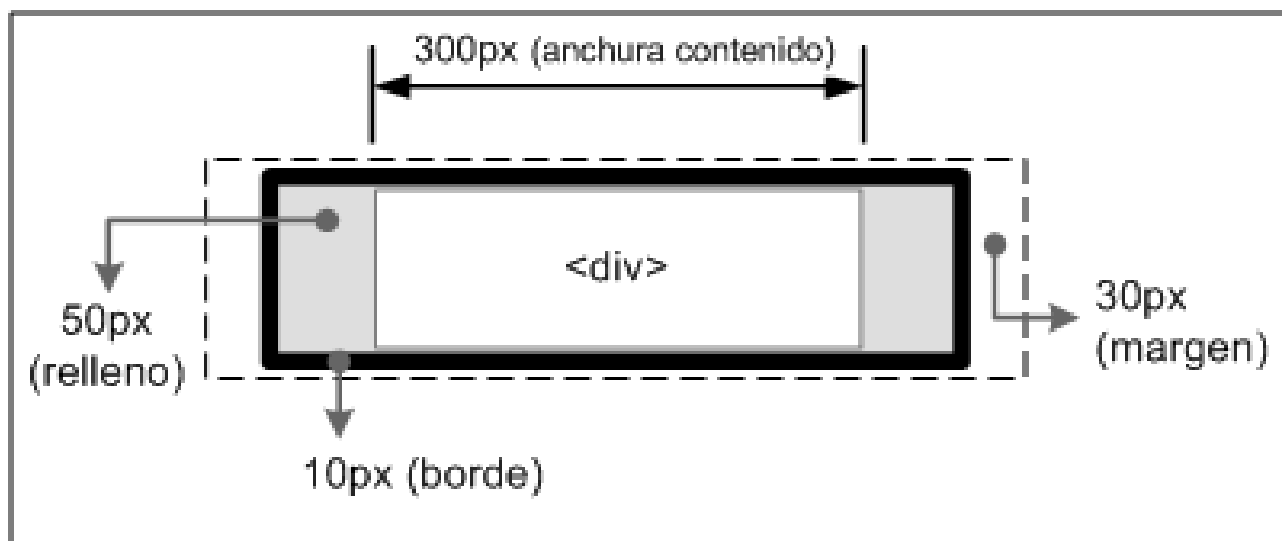
Lorem ipsum dolor sit amet, consectetur
adipiscing elit.

CSS incluye la propiedad `border-color` para establecer de forma simultánea el color de todos los bordes de una caja:

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.



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

Este es el título principal del sitio web

[principal](#) [fotos](#) [videos](#) [contacto](#)

Título del mensaje uno

Subtítulo del mensaje uno

publicado 10-12-2011

Este es el texto de mi primer mensaje



Esta es la imagen del primer mensaje

comentarios (0)

Título del mensaje dos

Subtítulo del mensaje dos

publicado 15-12-2011

Este es el texto de mi segundo mensaje



Esta es la imagen del primer mensaje

comentarios (0)

Mensaje número uno
Mensaje número dos

cabecera

menu

columna

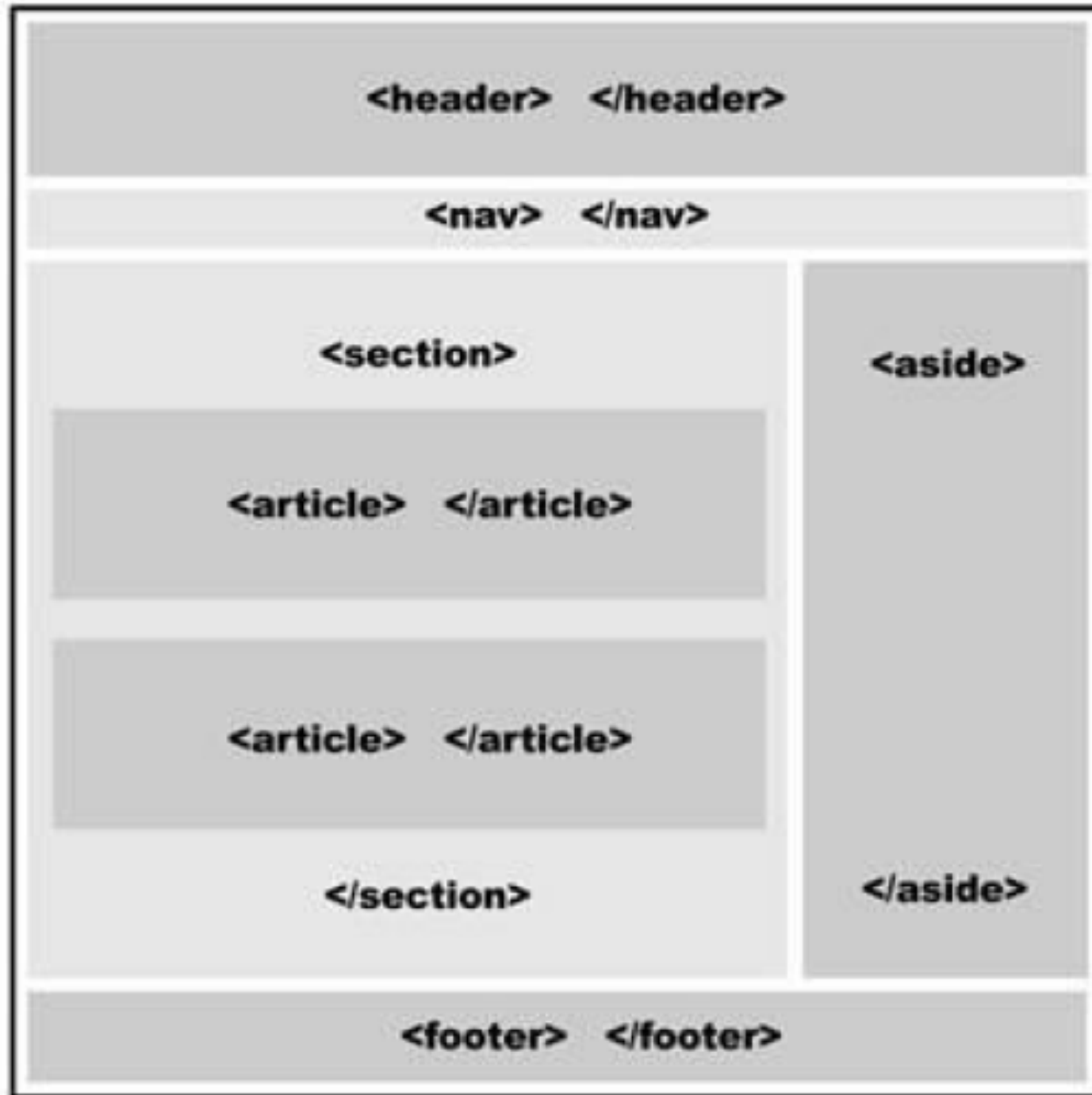
div
agrupar


article

seccion

article

Estructuras semánticas HTML5





display: block

display: block

**display: block
float: left**

**display: block
float: left**

**display: block
clear: both**

Modelo de caja

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

todo elemento tendrá un margen interno y externo de 0 pixeles.

```
h1 {  
font: bold 20px verdana, sans-serif;  
}  
h2 {  
font: bold 14px verdana, sans-serif;  
}
```

cambiando el grosor, tamaño y tipo de letra del texto dentro de los elementos **<h1>** y **<h2>**

Las propiedades que pueden ser declaradas usando **font** son:
font-style, font-variant, font-weight, font-size/line-height, font-family

En ese orden, son opcionales

Modelo de caja

```
header, section, footer, aside, nav, article, figure, figcaption, hgroup  
{  
  display: block;  
}
```

Declarar los nuevos elementos HTML5 como elementos block para asegurarnos de que serán tratados como regularmente se hace con elementos `<div>` y de este modo construir nuestro modelo de caja.

```
body {  
  text-align: center;  
}
```

Para centrar la página en la pantalla se necesita centrar el contenido dentro del cuerpo

```
#agrupar {  
  width: 960px;  
  margin: 15px auto;  
  text-align: left;  
}
```

La página web está centrada y tiene un tamaño fijo de 960 píxeles. Y el texto en ésta sección estará alineado a la izquierda.

auto significa 15px en los cuatro puntos izq, der, superior y inferior

Modelo de caja

```
#cabecera {  
background: #FFFBB9;  
border: 1px solid  
#999999;  
padding: 20px;  
}
```

Fondo amarillo, borde sólido de 1 pixel y un margen interior de 20 pixeles usando la propiedad ***padding***.

```
#menu {  
background: #CCCCCC;  
padding: 5px 15px;  
}  
#menu li {  
display: inline-block;  
list-style: none;  
padding: 5px;  
font: bold 14px verdana,  
sans-serif;  
}
```

Los elementos `` dentro de este elemento **`<nav>`** en particular usando el selector **`#menu li`**, el estilo ***display: inline-block*** los convierte en lo que se llama ***cajas inline***

Modelo de caja

```
#seccion {  
float: left;  
width: 660px;  
margin: 20px;  
}
```

```
#columna {  
float: left;  
width: 220px;  
margin: 20px 0px;  
padding: 20px;  
background: #CCCCCC;  
}
```

Fondo amarillo, borde sólido de 1 pixel y un margen interior de 20 pixeles usando la propiedad ***padding***.

Los elementos `` dentro de este elemento **`<nav>`** en particular usando el selector **`#menu li`**, el estilo ***display: inline-block*** los convierte en lo que se llama ***cajas inline***

Modelo de caja

```
#pie {  
clear: both;  
text-align: center;  
padding: 20px;  
border-top: 2px solid #999999;  
}
```

Borde de 2 píxeles en la parte superior de **<footer>**, un margen interno (***padding***) de 20 píxeles, y centra el texto dentro del elemento.

A sí mismo, restaura el normal flujo del documento con la propiedad ***clear***.

Esta propiedad simplemente restaura las condiciones normales del área ocupada por el elemento, no permitiéndole posicionarse adyacente a una caja flotante.

El valor usualmente utilizado es ***both***, el cual significa que ambos lados del elemento serán restaurados y el elemento seguirá el flujo normal.

Sin esta propiedad, el navegador presenta el documento en pantalla como si los elementos flotantes no existieran y las cajas se superponen.

```
article {  
background: #FFFBCC;  
border: 1px solid #999999;  
padding: 20px;  
margin-bottom: 15px;  
}
```

```
article footer {  
text-align: right;  
}
```

```
time {  
color: #999999;  
}
```

```
figcaption {  
font: italic 14px verdana, sans-serif;  
}
```

Otros estilos