



APRENDERAPROGRAMAR.COM

CONCEPTOS DE MARGEN
Y RELLENO CSS.
DIFERENCIAS ENTRE
MARGIN Y PADDING CSS
CON EL BOX MODEL.
EJEMPLOS (CU01028D)

Sección: Cursos

Categoría: Tutorial básico del programador web: CSS desde cero

Fecha revisión: 2029

Resumen: Entrega nº28 del Tutorial básico: "CSS desde cero".

Autor: César Krall

MARGEN Y RELLENO

Para avanzar en la comprensión del funcionamiento de CSS debemos conocer, saber utilizar y diferenciar dos propiedades importantes: margin y padding. Al igual que con los bordes, tendremos distintas propiedades derivadas de las principales y también tendremos shorthand notations o notaciones abreviadas para poder expresarlas.

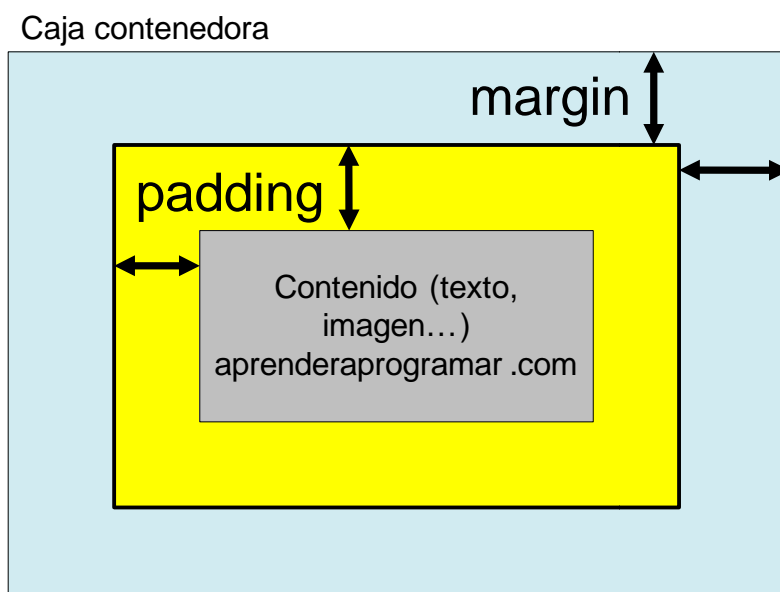


MARGIN Y PADDING

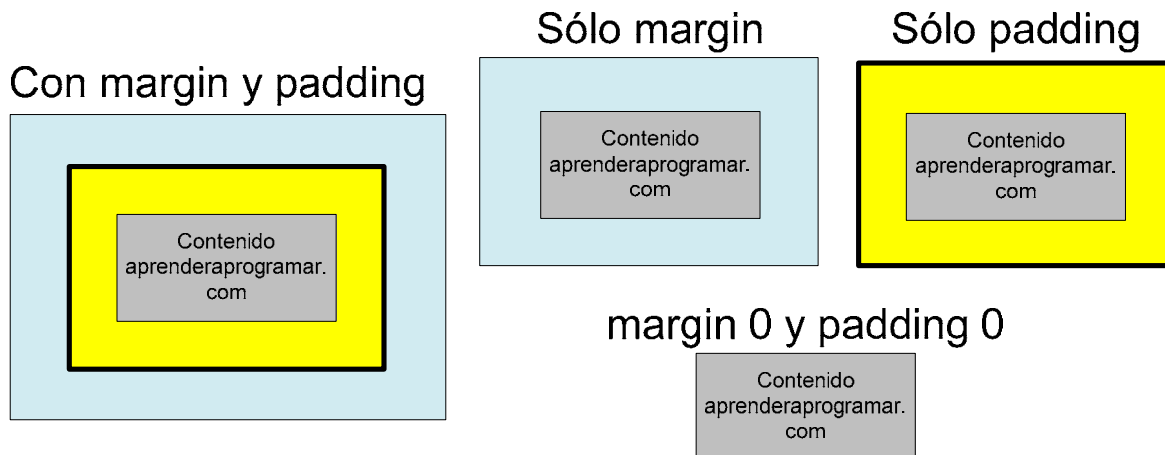
Empezaremos realizando una definición de conceptos:

- a) Contenido de un elemento HTML es aquello que se pretende mostrar al visualizar ese elemento: puede ser un texto, una imagen o quizás otros elementos como un objeto reproductor de sonido ó un mapa. El contenido en CSS tiene forma de caja rectangular.
- b) Borde: es el perímetro que engloba el contenido y el posible relleno (en inglés padding) entre el borde y el contenido.
- c) Relleno (padding): es el espacio transparente que se puede dejar, opcionalmente, entre el contenido y el borde del elemento. El relleno no tiene color (pero permite ver el color de fondo).
- d) Margen: es el espacio transparente que se puede dejar, opcionalmente, entre el borde del elemento y el borde de otras cajas adyacentes. El margen no tiene color (pero permite ver el color de fondo).

Viendo esquemas gráficos y código se comprenderán con más facilidad estos conceptos:

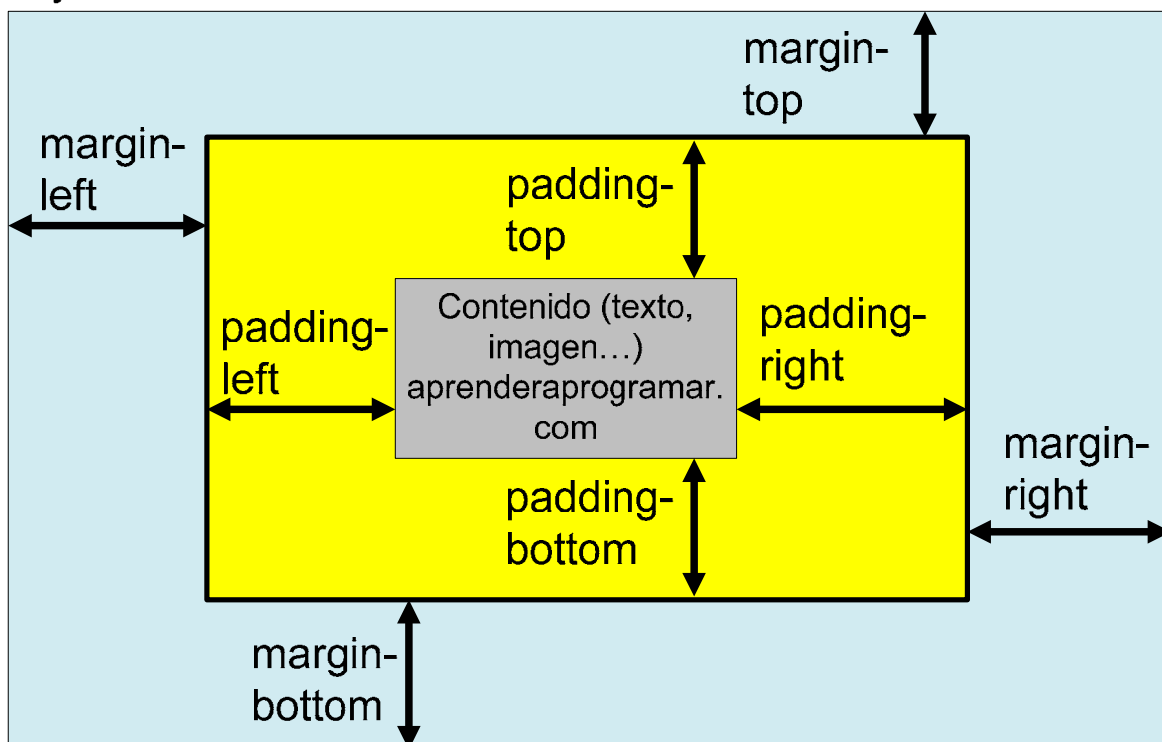


Un elemento HTML puede tener margin y padding, sólo tener margin, sólo tener padding, o no tener ni margin ni padding. En caso de no tener margin ni padding el elemento aparecerá "ajustado" exactamente a su caja contenedora.



Al igual que ocurría con las propiedades de borde, las propiedades de margen y relleno se pueden subdividir para aplicarlas a cada uno de los lados de la caja CSS por separado.

Caja contenedora



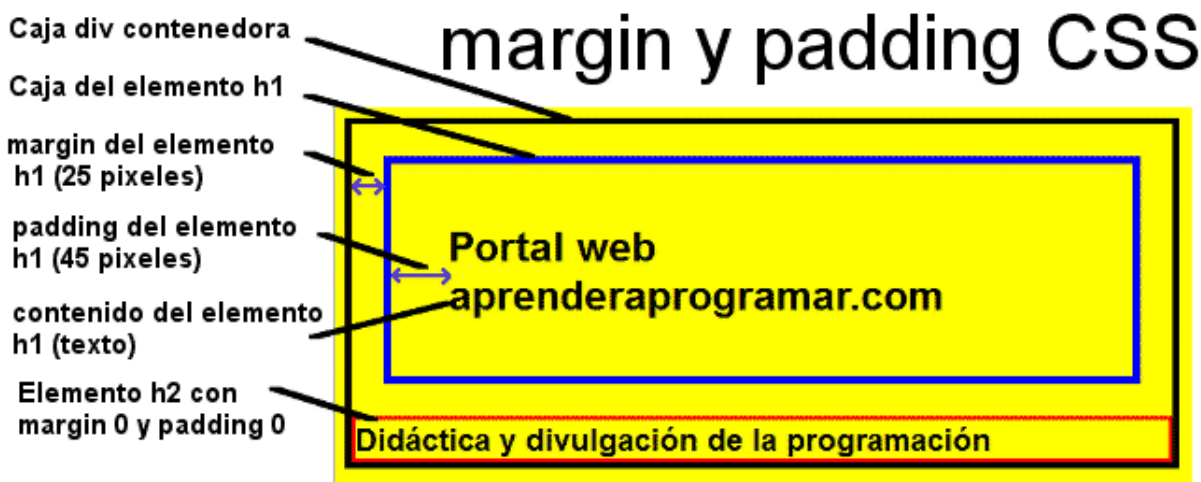
EJEMPLO EN CÓDIGO

Para ver la aplicación práctica de estos conceptos vamos a utilizar [el código HTML de base](#) que venimos empleando a lo largo del curso.

Define los siguientes estilos css y visualiza el resultado en tu navegador:

```

/* Curso CSS estilos aprenderaprogramar.com */
*{font-family: arial;}
body {background-color: yellow; width: 60%;}
h1{margin: 25px; padding: 45px; border: solid 6px blue; }
h2{border-style: solid; border-color:red; margin: 0; padding:0;}
div {border: solid thick black;}
div div {border: solid medium purple;}
div div div {border: dashed medium grey;}
img {border: solid #FF00FF 2px;}
    
```



Vamos a analizar línea a línea el código CSS y los resultados obtenidos. La primera línea es un comentario. En la segunda línea establecemos que las fuentes de texto a emplear en todos los elementos del documento sean tipo arial. En la tercera línea establecemos que el elemento body tenga color de fondo amarillo y ocupe el 60% del ancho de pantalla disponible.

En la cuarta línea definimos que el margen de los elementos h1 será de 25 píxeles. Al no especificar top, right, bottom o left, este margen se aplicará en las cuatro direcciones (mismo comportamiento que el que hemos estudiado previamente con los bordes). Igualmente definimos que el relleno para los elementos h1 será de 45 px en todas las direcciones y que el borde se muestre de color azul.

En la quinta línea definimos que los elementos h2 muestren un borde de color rojo y que no llevarán margen ni relleno.

En las siguientes líneas establecemos bordes para que se muestren las cajas contenedoras de los elementos HTML de distintos colores. Estas líneas no nos interesa comentarlas ahora.

En los resultados de visualización comprobamos lo siguiente: si los elementos h2 no tienen margen, se mostrarán pegados a las cajas adyacentes. Así vemos que ocurre con la caja del div contenedor, pero no

con la caja del h1. ¿Es esto correcto? Sí, porque el h1 tiene establecido un margen y por tanto la caja del elemento h2 no puede pegarse a la caja del elemento h1 porque hay un margen que lo impide.

Prueba a hacer el siguiente cambio en el código: `h2{border-style: solid; border-color:red; margin: 25px; padding:0;}`

Al visualizar el resultado comprobarás que los márgenes verticales (top y bottom) contiguos no son aditivos, es decir, si la caja h1 tiene de margen 25px y la caja de h2 tiene de margen 25px, el espacio vertical que quedará entre ambas cajas es 25px y no 50px. ¿Por qué? Porque la referencia para establecer un margen vertical es el borde de las cajas y no el límite del margen de una caja adyacente. Si la caja h1 tuviera de margen vertical 25px y la caja h2 tuviera de margen vertical 10px el espacio entre ambas cajas sería de 25px, el mayor entre los dos valores.

El comportamiento de los márgenes horizontales es diferente: en este caso, sí es aditivo. Supongamos que tenemos dos imágenes una al lado de otra y establemos `img {margin:25px;}`. En este caso el espacio que habrá entre ambas imágenes sí es de 50 px, es decir, los márgenes horizontales sí se suman, mientras que los verticales no se suman. Este comportamiento peculiar deberemos tenerlo en cuenta cuando trabajemos con CSS. Y como CSS no es matemáticas, habrá que realizar pruebas y comprobaciones para verificar que se cumple aquello que esperamos.

La idea clave con que debemos quedarnos es que los márgenes verticales tienen un comportamiento distinto al de los márgenes horizontales.

Continuamos con el análisis de los resultados. Si la caja h2 no tiene padding, la línea de borde quedará ceñida al contenido (en este caso al texto). Así vemos que ocurre por la izquierda, por arriba y por abajo. Sin embargo por la derecha el borde no está ajustado al texto. ¿Por qué? Porque los elementos h1 y h2 son elementos de tipo block, y por tanto tienden a ocupar todo el ancho de pantalla. Más adelante veremos que podemos cambiar el comportamiento para que se comporten como block ó como inline según nuestro criterio.

Finalmente queremos llamar la atención sobre el hecho de que se muestre un pequeño espacio entre el borde del div y el borde del elemento body, de ahí que se vea una pequeña franja amarilla por fuera del contenedor div. Esta situación se debe a los márgenes o rellenos que por defecto aplica el navegador. Podemos eliminar estos márgenes o rellenos por defecto incluyendo en el selector universal este código: `*{margin:0; padding:0;}` Prueba a aplicarlo y compruébalo. A prácticas de este tipo, consistentes en escribir código para anular el comportamiento por defecto que introducen los navegadores, se le suele denominar "reseteo". Es una práctica que siguen muchos diseñadores y programadores, pero no todos. Algunos programadores están en contra de utilizar prácticas de reseteo.

El código de reseteo puede ser bastante más amplio que el que hemos indicado y hablaremos de ello más adelante. Nosotros iremos introduciendo algunas prácticas de reseteo cuando lo consideremos necesario para el desarrollo del curso.

EJERCICIO

Para cada una de las siguientes afirmaciones indica si la afirmación es verdadera o falsa y justifica brevemente tu respuesta:

- a) CSS muestra los contenidos con tres formas básicas: rectangular, circular y elipsoidal.
- b) Al crear un documento HTML, pueden aparecer márgenes y rellenos que no hayan sido explícitamente introducidos por nosotros como programadores, sino que hayan sido introducidos por el navegador que estemos empleando.
- c) El reseteo CSS consiste en la recarga de la página para borrar la caché del navegador, de modo que los estilos se recarguen completamente.

Para comprobar si tus respuestas son correctas puedes consultar en los foros aprenderaprogramar.com.

Próxima entrega: CU01029D

Acceso al curso completo en aprenderaprogramar.com -- > Cursos, o en la dirección siguiente:
http://aprenderaprogramar.com/index.php?option=com_content&view=category&id=75&Itemid=203