



APRENDERAPROGRAMAR.COM

PADDING Y MARGIN CSS.
TOP, RIGHT, BOTTOM Y
LEFT. MARGIN NEGATIVO
Y CENTRAR CON MARGIN
AUTO. EJEMPLOS
(CU01029D)

Sección: Cursos

Categoría: Tutorial básico del programador web: CSS desde cero

Fecha revisión: 2029

Resumen: Entrega nº29 del Tutorial básico: "CSS desde cero".

Autor: César Krall

MARGIN Y PADDING

Ya debemos tener claro el concepto de margen y de relleno en CSS. Ambos son elementos muy usados en los diseños y que presentan ciertas similitudes y ciertas diferencias. En algunos casos el efecto visual al que llevan es muy parecido. Sin embargo, conviene tener clara la diferencia entre ambos conceptos y saber utilizarlos adecuadamente.



PROPIEDAD CSS PADDING

La propiedad CSS padding nos permite fijar el relleno de un elemento. La sintaxis a emplear es del tipo:

```
selectorElemento { padding valor1Uds valor2Uds valor3Uds valor4Uds; }
```

PROPIEDAD CSS padding	
Función de la propiedad	Permite definir un relleno entre el borde de caja y el contenido.
Valor por defecto	0 (no existe relleno).
Aplicable a	Elementos tipo block y elementos insertados en una posición que son reemplazados por un objeto (entre ellos img, input, textarea, select, object).
¿Se hereda directamente?	No
Valores posibles para esta propiedad	Un valor absoluto o relativo (establece relleno en los cuatro sentidos)
	Dos valores (el primero de ellos aplica a top y bottom y el segundo a right y left)
	Tres valores (el primero para top, el segundo right / left y el tercero bottom).
	Cuatro valores (se aplican a top, right, bottom y left en este orden)
	inherit (heredado del elemento padre)
Ejemplos aprenderaprogramar.com	<pre>#menu1 {padding: 55px 0.5em 177px 2px;} .imgVentana { padding: 15px;} h1 {padding: 10% 5%;} .container1 {padding: 0 auto;}</pre>

La propiedad padding se puede aplicar en todos los lados del elemento, o bien a cualquiera de los cuatro lados que conforman una caja CSS. Existen cuatro propiedades cuyo funcionamiento es análogo al de padding, pero que permiten dar un tratamiento diferenciado a cada uno de los lados de la caja CSS: **padding-top**, **padding-right**, **padding-bottom** y **padding-left**. La sintaxis a utilizar y la forma de funcionamiento es la misma que hemos explicado.

El valor de padding se puede establecer en %, realizándose el cálculo **respecto al ancho del elemento contenedor** (un valor 100% sería un relleno igual al ancho del elemento contenedor).

La propiedad padding no admite valores negativos.

Recordar que tanto margin como padding generan un espacio transparente. El color o imagen que se vean en dicho espacio serán el color de fondo (background-color) o la imagen de fondo (background-image) que estén definidos para el elemento afectado.

Tanto con margin como con padding pueden haber determinadas situaciones en que los navegadores, sobre todo los más antiguos, no respondan como sería de esperar, aunque la mayor parte de los navegadores actuales responden adecuadamente.

PROPIEDAD CSS MARGIN

La propiedad CSS margin nos permite fijar el margen de un elemento. La sintaxis a emplear es del tipo:

```
selectorElemento { margin valor1Uds valor2Uds valor3Uds valor4Uds; }
```

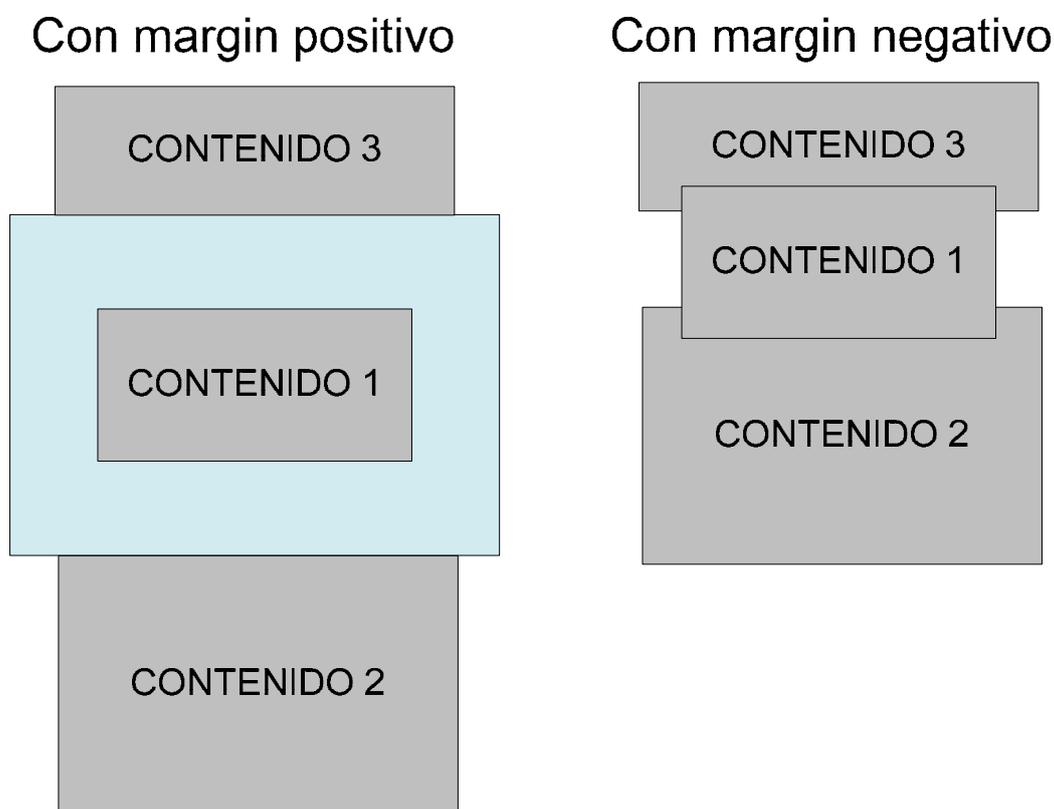
PROPIEDAD CSS margin	
Función de la propiedad	Permite definir el margen entre el borde de la caja del elemento y el borde de las cajas adyacentes.
Valor por defecto	0 (no existe margen).
Aplicable a	Elementos tipo block y elementos insertados en una posición que son reemplazados por un objeto (entre ellos img, input, textarea, select, object).
¿Se hereda directamente?	No
Valores posibles para esta propiedad	auto (el navegador aplicará el margen por defecto)
	Un valor absoluto o relativo (establece márgenes en los cuatro lados de la caja)
	Dos valores (el primero de ellos aplica a top y bottom y el segundo a right y left)
	Tres valores (el primero para top, el segundo right / left y el tercero bottom).
	Cuatro valores (se aplican a top, right, bottom y left en este orden)
	inherit (heredado del elemento padre)
Ejemplos aprenderaprogramar.com	<pre>#menu1 {margin: 55px 0.5em 177px 2px;} .imgVentana { margin: 15px;} h1{margin: 10% 5%;} .container1 {margin: 0 auto;}</pre>

La propiedad `margin` se puede aplicar en todos los lados del elemento, o bien a cualquiera de los cuatro lados que conforman una caja CSS. Existen cuatro propiedades cuyo funcionamiento es análogo al de `margin`, pero que permiten dar un tratamiento diferenciado a cada uno de los lados de la caja CSS: **`margin-top`**, **`margin-right`**, **`margin-bottom`** y **`margin-left`**. La sintaxis a utilizar y la forma de funcionamiento es la misma que hemos explicado anteriormente.

El valor de `margin` se puede establecer en %, realizándose el cálculo **respecto al ancho del elemento contenedor** (un valor 100% sería un margen igual al ancho del elemento contenedor).

A diferencia de otras propiedades, se admite que un valor de `margin` pueda ser negativo (aunque esto no es usual y no debería usarse excepto cuando razonadamente es necesario y aconsejable. Un margen negativo puede dar lugar, en algunos casos, a visualizaciones incorrectas o extrañas). Para entender el efecto de valores de `margin` negativos podemos pensar en los márgenes como aviones que tienen un espacio básico definido por la anchura de sus alas y fuselaje, que definen un espacio rectangular que sería el `margin 0` (no hay `margin` y el avión "empuja" a otros aviones colindantes). Ese espacio se puede aumentar hacia fuera estableciendo un `margin` positivo que sería como "establecer un perímetro de seguridad para el avión", lo que impide que otros aviones entren en el espacio reservado a nuestro avión. Si el `margin` se establece en un valor negativo, el avión no sólo no tiene un espacio extra hacia fuera, sino que tampoco tiene su espacio básico y se coloca invadiendo el espacio delimitado por otro avión contiguo. `Margin` positivo sería "empujar hacia fuera" y `margin` negativo sería "solaparse" con elementos adyacentes.

En la siguiente imagen vemos cómo afecta establecer un `margin` positivo o negativo a un elemento "Contenido 1" que tiene otros dos elementos vecinos.



La invasión del espacio colindante puede ser en todas direcciones o establecerse específicamente para alguna de las cuatro direcciones posibles (top, right, bottom o left) usando margin-top, margin-right, margin-bottom o margin-left. Un aspecto a tener en cuenta es que cuando existe superposición de un elemento con otros habrá que comprobar o establecer qué elemento debe quedar encima y qué elemento debe quedar debajo. En la imagen anterior hemos dibujado el elemento "Contenido 1" situándose encima de los otros elementos, pero en la práctica esto dependerá de diversos factores y no siempre será así. Existen propiedades CSS con las que podremos controlar el orden de apilamiento de elementos cuando existe superposición. Lo veremos más adelante.

Otro aspecto que merece ser comentado es el resultado que se obtendrá al establecer un valor de margen "auto". Elegir auto equivale a indicar que el margen debe ser establecido automáticamente por el navegador. En general, si un elemento es de tipo block y no tiene establecido una anchura específica, cuando establecemos el margen con valor "auto" el margen que aplicará el navegador será cero (sin margen). En cambio, si el elemento es de tipo block pero tiene un valor de ancho (width) definido, al establecer el margen auto el navegador establecerá un margen izquierdo y derecho iguales, de modo que el contenido estará equidistante respecto al borde del contenedor. El efecto que se aprecia, en este caso, es que el elemento aparece centrado.

Ejemplo:

a) Código que da lugar a un margen cero: `h1{ border: solid; margin:auto; }`, el elemento h1 se ve a todo lo ancho y sin márgenes.

b) Código que da lugar a el centrado horizontal del elemento: `h1{border: solid 6px blue; width:65%; margin:auto;}`. El elemento h1 se ve sin márgenes superior (top) ni inferior (bottom), pero se ve centrado (con márgenes right y left de igual valor).

Prueba a visualizar la página de pruebas en tu navegador con este código. En el primer caso h1 es un elemento block sin ancho definido, por lo que tiende a ocupar todo el ancho disponible. En este caso margin: auto; da lugar a que se aplique margen cero. En el segundo caso h1 es un elemento block con ancho el 65% del elemento contenedor, por tanto tiene un ancho definido. En este caso margin: auto; da lugar a que se apliquen márgenes iguales por ambos lados del elemento y a que visualmente el elemento aparezca centrado.

EJERCICIO

Crea un documento HTML con 2 elementos div de anchura 250 píxeles, con un margen de 20px en todas direcciones y uno junto al otro (en horizontal). En cada div introduce un texto (p.ej. div 1, div 2) y aplícale los siguientes estilos de borde y relleno a ambos elementos. Color de fondo #FFB6C1. La parte superior con borde de puntos redondeados, grosor 15 píxeles, color #DC143C y relleno de 30 píxeles. La parte derecha con borde de trazos o segmentos rectangulares, grosor 10 píxeles, color verde y relleno de 45 píxeles. La parte inferior con borde de línea doble, grosor 10 píxeles, color #FF00FF y relleno 0 píxeles. La parte izquierda con borde con efecto ridge, grosor 40 píxeles, color #2F4F4F y relleno 60 píxeles.

Responde a las siguientes preguntas:

- a) ¿Cuál es el ancho total ocupado por cada div (incluyendo sus bordes y rellenos)?
- b) ¿Cuál es el alto total ocupado por cada div (incluyendo sus bordes y rellenos)?
- c) ¿Cuál es el ancho total desde el límite izquierdo del borde del div más a la izquierda hasta el límite derecho del borde del div más a la derecha (teniendo en cuenta márgenes, bordes y rellenos)?

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

Próxima entrega: CU01030D

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