



APRENDERAPROGRAMAR.COM

PROPIEDAD POSITION CSS:
STATIC, RELATIVE,
ABSOLUTE, FIXED. TOP,
RIGHT, BOTTOM, LEFT.
EJEMPLOS EJERCICIOS DE
POSICIONAMIENTO
RESUELTOS (CU01032D)

Sección: Cursos

Categoría: Tutorial básico del programador web: CSS desde cero

Fecha revisión: 2029

Resumen: Entrega nº32 del Tutorial básico: "CSS desde cero".

Autor: César Krall

POSICIONAMIENTO CSS

Hasta ahora conocemos que los elementos HTML se dividen en dos grandes tipos: elementos block que tienden a ocupar el espacio disponible a todo lo ancho y en caso de existir varios se sitúan unos debajo de otros y elementos inline que ocupan el espacio necesario dentro de una línea y en caso de existir varios se sitúan uno junto a otro en la misma línea (siempre que haya espacio).



Los principales elementos HTML tipo block son:

- a) **Divisores:** div
- b) **Párrafos:** p
- c) **Formularios:** form
- d) **Títulos:** h1, h2, h3, h4, h5, h6.
- e) **Listas:** ul, ol
- f) **Elementos de listas:** li
- g) **Tablas:** table
- h) **Otros:** center, pre, tbody, td, th, tr...

Los principales elementos HTML tipo inline son:

- a) **Links:** a
- b) **Divisores:** span
- c) **Imágenes:** img (aunque este elemento tiene características especiales)
- d) **Controles de formulario:** input, label
- e) **Otros:** strong, u, select...

Queda fuera de la clasificación el elemento body, que engloba todo el espacio disponible en el dispositivo de visualización.

Cuando tenemos distintos elementos tipo block cada uno se coloca en una nueva línea distinta a la del elemento anterior, es decir, se colocan uno debajo de otro:

Bloque1 tiene este contenido de texto

Bloque2 tiene este contenido de texto

Bloque3 tiene este contenido de texto ...

Cuando tenemos distintos elementos en línea se colocan uno junto a otro:

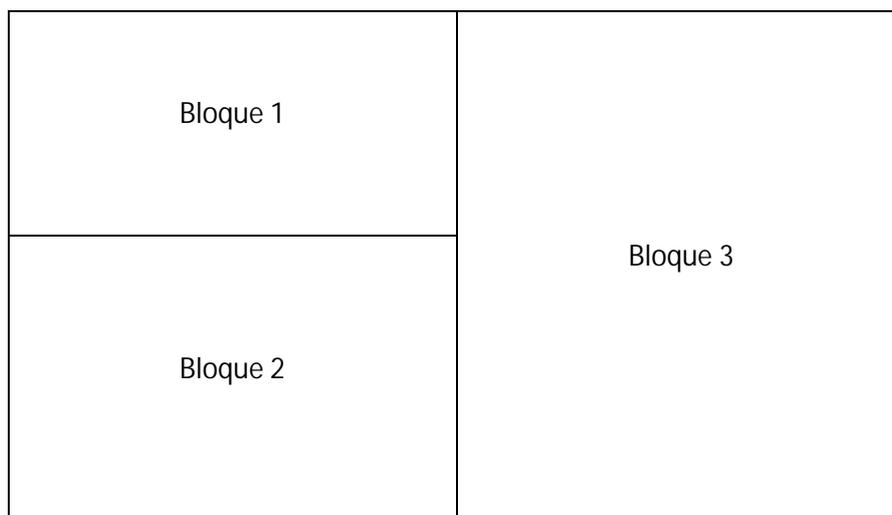
ElementoEnLínea1 ElementoEnLínea2 ElementoEnLínea3

Además los elementos en línea que están dentro de un bloque se insertan en la misma línea que el contenido del bloque. Por ejemplo:

Este texto es un párrafo [y esto es un link inline](#) y aquí continúa el párrafo.

Recordar que las propiedades width y height sólo son aplicables a elementos tipo block y elementos insertados en una posición que son reemplazados por un objeto (entre ellos img, input, textarea, select, object).

Ahora bien, cuando creamos páginas web necesitaremos que elementos de dimensiones determinadas se sitúen en posiciones determinadas. Por ejemplo podemos querer tener tres elementos con dimensiones width y height definidas (por tanto bloques) situados en determinadas posiciones como vemos en este esquema:



Con el código simple que hemos visto hasta ahora esto no sería posible, ya que unos elementos se situarían debajo de otros, sin cumplir nuestro objetivo de situarlos en posiciones específicas.

CSS tiene distintas propiedades y posibilidades que nos van a permitir gestionar el posicionamiento de elementos para crear diseños atractivos y conforme a nuestras necesidades. Comenzaremos el estudio de estas posibilidades con la propiedad CSS position.

PROPIEDAD POSITION

Esta propiedad nos permite establecer la posición de un elemento.

La sintaxis a emplear es del tipo:

```
selectorElemento {position: especificaciónDePosición; }
```

PROPIEDAD CSS position	
Función de la propiedad	Permite definir el tipo de posición de un elemento y su comportamiento.
Valor por defecto	static
Aplicable a	Elementos tipo block y algunos elementos inline especiales como img.
Valores posibles para esta propiedad	static (comportamiento normal o por defecto)
	relative (permite que un elemento se desplace respecto a lo que hubiera sido su posición normal; el resto de elementos continúan en su posición ignorando al que se desplaza, lo que puede crear superposiciones; el espacio libre que deja el elemento queda libre).
	absolute (permite que un elemento se desplace respecto al origen de coordenadas del primer elemento contenedor posicionado ó respecto a la esquina superior izquierda de la ventana de visualización; el resto de elementos actúan como si el desplazado no existiera, por lo que su espacio será ocupado por otros elementos; puede crear superposiciones)
	fixed (permite fijar un elemento en una posición respecto al origen de coordenadas del primer elemento contenedor posicionado ó respecto a la esquina superior de la ventana de visualización; el elemento se mantendrá en la ventana de visualización o viewport, siempre en una misma posición aunque el usuario se desplace por la web haciendo scroll)
inherit (se heredan las características del elemento padre).	
Ejemplos aprenderaprogramar.com	#content1 {position: relative; } .elementoFijado {position: fixed; }

La posición absoluta de un elemento que no está dentro de ningún otro se determina respecto a la parte superior izquierda del elemento html (parte superior izquierda de la ventana donde se visualiza la página web; normalmente coincide con la parte superior izquierda del elemento body).

Si un elemento está dentro de varios contenedores (p.ej. dentro de varios div) el origen de coordenadas se establece respecto a la esquina superior izquierda del contenedor que tenga un valor de position establecido y distinto de static o, en caso de no existir ningún elemento contenedor posicionado, respecto a la esquina superior izquierda del elemento html (parte superior izquierda de la ventana donde se ve la página web). Esto se comprenderá con los ejemplos de código que veremos.

Los ejemplos de código sobre el uso de esta propiedad los veremos una vez hayamos estudiado algunas propiedades que se usan habitualmente junto a ésta (top, right, bottom y left).

Esta no es la única propiedad para establecer posiciones de elementos. Hay otras propiedades importantes como float, que veremos más adelante.

PROPIEDADES TOP, RIGHT, BOTTOM Y LEFT

Estas propiedades nos permiten definir el desplazamiento de la posición de un elemento respecto a un origen de coordenadas y el origen de coordenadas que se toma.

PROPIEDADES CSS top, right, bottom y left	
Función de la propiedad	Permiten definir el desplazamiento de un elemento y el origen de coordenadas que se tomará para el mismo.
Valor por defecto	Auto
Aplicable a	Elementos que están posicionados (no aplica si la posición es static).
Valores posibles para estas propiedades	auto (no hay desplazamiento especificado y como origen de coordenadas se toma la esquina superior izquierda del elemento padre)
	Una unidad de medida relativa (se admiten porcentajes).
	Una unidad de medida absoluta.
	inherit (se heredan las características del elemento padre).
Ejemplos aprenderaprogramar.com	<pre>#menu1 { position: relative; top: 40px; left: 2em; } #moduloEGM { position: absolute; bottom: 5%; left: 30px; }</pre>

Para elementos con posición relative estas propiedades definen cuánto se desplaza el elemento respecto a lo que sería su posición normal.

Para elementos con posición absolute o posición fixed estas propiedades definen cuánto se desplaza el elemento respecto a las coordenadas de origen.

Se admiten valores de desplazamientos negativos.

En el caso de usar porcentajes, éste se calcula:

- a) Respecto al valor de altura (height) del elemento si se aplican con top ó bottom.
- b) Respecto al valor de anchura (width) del elemento si se aplican con right ó left.

No tiene sentido utilizar top y bottom al mismo tiempo, porque sería decir que el elemento sube y baja. Hay que utilizar sólo una de estas dos propiedades. Lo mismo podemos decir para right y left.

EJEMPLOS DE POSICIONAMIENTO RELATIVO

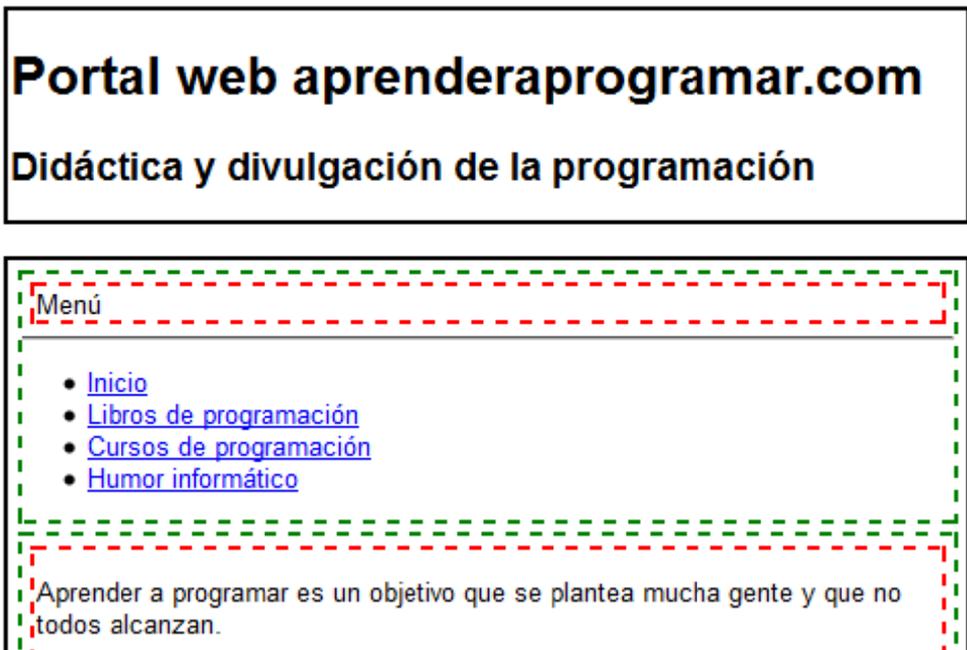
A continuación veremos algunos ejemplos de uso de las propiedades anteriormente especificadas. Para ello vamos a utilizar [el código HTML de base](#) que venimos empleando a lo largo del curso.

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

```

/* Curso CSS estilos aprenderaprogramar.com*/
*{font-family: arial; }
body {width: 600px;}
div {border-style: solid;}
div div {border-style: dashed; border-color: green; margin: 5px;}
div div div {border-style: dashed; border-color: red; margin: 5px;}
    
```

Con esto lo único que hemos hecho es poner bordes y márgenes para visualizar las cajas que conforman la página web. El resultado debe ser similar a este:



Vemos un div en color verde correspondiente al menú (dentro del cual hay otro div en color rojo con el texto "Menú", una línea generada por el <hr/> y una lista ul con los items de menú).

Accede al código HTML y establece como id de dicho div "menu1". El código HTML de ese fragmento de web quedará así:

```

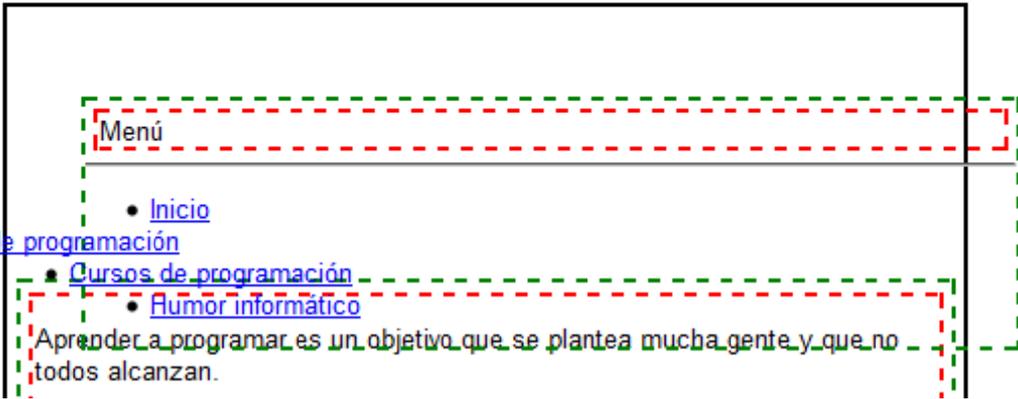
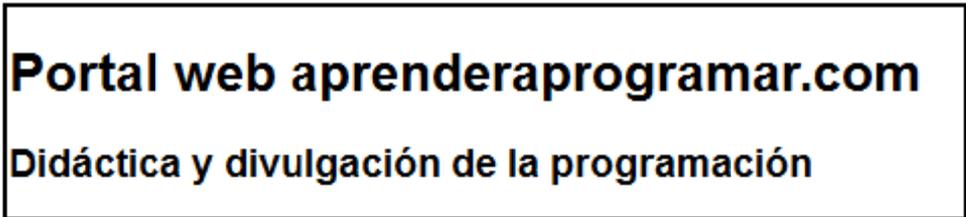
<!-- menu -->
<div id="menu1">
<div>Menú</div>
<hr/>
<ul>
<li><a href="#">Inicio</a></li>
<li><a href="libros.html" >Libros de programación</a> </li>
<li><a href="cursos.html" >Cursos de programación</a> </li>
<li> <a href="humor.html" >Humor informático</a> </li>
</ul>
</div>
<!-- fin menu -->
    
```

Ahora vamos a modificar la definición de estilos CSS para hacer lo siguiente: modificaremos la posición del elemento con id "menu1" para que se desplace 50 pixeles hacia abajo y 40 pixeles hacia la derecha respecto a lo que debía ser su posición original. Además vamos a hacer que el segundo elemento de la

lista de items de menú se desplace 150 pixeles hacia la izquierda respecto a lo que debiera ser su posición normal y el tercer elemento de la lista 50 pixeles a la izquierda respecto de lo que debiera ser su posición normal. Para ello usamos este código y visualizamos el resultado:

```

/* Curso CSS estilos aprenderaprogramar.com*/
*{font-family: arial; }
body {width: 600px;}
div {border-style: solid;}
div div {border-style: dashed; border-color: green; margin: 5px;}
div div div {border-style: dashed; border-color: red; margin: 5px;}
#menu1 { position:relative; top:50px; left: 40px; }
li:nth-child(2) {position: relative; right: 150px;}
li:nth-child(3) {position: relative; right: 50px;}
    
```



Comprobamos que la caja del div se ha desplazado (con todo su contenido). Igualmente los elementos de lista indicados se han desplazado. Comprobamos algunas cosas curiosas: al desplazar la caja del div, ésta sale por fuera de las dimensiones del elemento body y **se ha superpuesto** con la caja que existía debajo. Al desplazar el segundo elemento de lista 150 px hacia la izquierda, el elemento ha quedado cortado y no se visualiza parte de él (si el desplazamiento hubiera sido todavía mayor, no se vería ninguna parte de este item de la lista aparentando haber “desaparecido”. El tercer elemento de la lista se ve desplazado pero no se ha salido de la pantalla.

Para desplazar hacia abajo usamos top y para desplazar hacia la derecha usamos left. Puede parecer confuso porque left significa izquierda, pero usar left no significa “desplazar hacia la izquierda”, sino “respecto a lo que sería la posición normal de su lateral izquierdo, desplazar lo indicado”, correspondiendo un valor positivo a “desplazar hacia la derecha” y un valor negativo a “desplazar hacia la izquierda”.

EJEMPLOS DE POSICIONAMIENTO ABSOLUTO

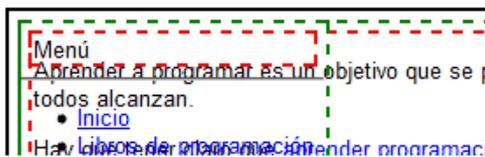
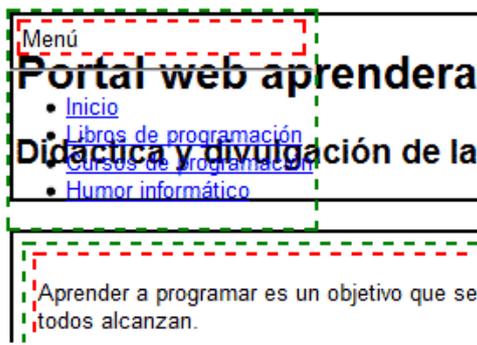
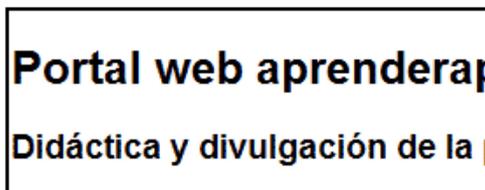
Seguimos trabajando sobre el HTML y hoja de estilos anterior. Define los siguientes estilos css y visualiza el resultado en tu navegador:

```
/* Curso CSS estilos aprenderaprogramar.com*/
*{font-family: arial; }
body {width: 600px;}
div {border-style: solid;}
div div {border-style: dashed; border-color: green; margin: 5px;}
div div div {border-style: dashed; border-color: red; margin: 5px;}
#menu1 { position:absolute; }
```

```
/* Curso CSS estilos aprenderaprogramar.com*/
*{font-family: arial; }
body {width: 600px;}
div {border-style: solid;}
div div {border-style: dashed; border-color: green; margin: 5px;}
div div div {border-style: dashed; border-color: red; margin: 5px;}
#menu1 { position:absolute; left:0; top:0; }
```

Position: absolute;
top: auto;
left: auto;

Position: absolute;
top: 0;
left: 0;



El código #menu1 { position:absolute; } equivale a #menu1 { position:absolute; left: auto; top: auto; } lo que supone que se toma como origen de coordenadas el del elemento padre. De ahí que en la imagen de la izquierda se vea la caja en su posición natural, pero al estar posicionada de forma absoluta es ignorada por el resto de elementos y su espacio es ocupado como si no estuviera ahí.

El código #menu1 { position:absolute; left:0; top:0; } significa que el origen de coordenadas será el del primer contenedor posicionado o, si no existe, la esquina superior izquierda de la ventana de visualización. En este caso se toma la esquina superior izquierda de la ventana como origen de coordenadas.

Cuando se establece position en absolute o fixed el ancho del bloque se ajusta al contenido (excepto si está establecido específicamente mediante la propiedad width).

Prueba a cambiar los valores de left y top y comprueba sus resultados. Por ejemplo si estableces valores left: -50px; top: -50px; verás cómo la caja “se sale de la pantalla” por la esquina superior izquierda. Si estableces right:0; top:0; verás que la caja se sitúa en la parte superior derecha de la pantalla.

Supón ahora que quieres usar como origen de coordenadas absolutas la esquina superior izquierda del div contenedor del que estamos usando como ejemplo. Ese div contenedor tiene, aparte del menú, varias cosas más como texto con imágenes y un formulario.

Para hacer que el div del menú tome para origen de coordenadas el div contenedor en lugar de la esquina superior izquierda de la pantalla hemos de posicionar el contenedor. **Posicionarlo significa darle un valor de position distinto del valor por defecto (static).** Esto podemos hacerlo de varias maneras. Por ejemplo podríamos poner un id al div contenedor y aplicarle una regla nombreEscogido {position: relative; }. No es necesario establecer un desplazamiento ya que nosotros ahora lo único que queremos es que dicho elemento se identifique como un elemento posicionado, y para ello establecer su propiedad position como relative es suficiente.

También podemos hacerlo como vemos en este código:

```
/* Curso CSS estilos aprenderaprogramar.com*/
*{font-family: arial; }
body {width: 600px; border-style:dotted;}
div {border-style: solid; position:relative;}
div div {border-style: dashed; border-color: green; margin: 5px;}
div div div {border-style: dashed; border-color: red; margin: 5px;}
#menu1 { position:absolute; left:150px; top:30px; }
```

Hemos establecido para todos los div su propiedad position como relative. Al ir a realizar el posicionamiento del elemento con id menu1 el navegador busca el primer div contenedor posicionado y realiza el desplazamiento respecto a su esquina superior izquierda. Comprueba la diferencia al mantener todos los div con position:relative; y visualiza cómo cambia el resultado respecto a eliminar ese posicionamiento para todos los div: el origen de coordenadas que se toma es distinto.

EJEMPLOS DE POSICIONAMIENTO FIXED

Seguimos trabajando sobre el HTML y hoja de estilos anterior. Escribe y comprueba los resultados con este código:

```
/* Curso CSS estilos aprenderaprogramar.com*/
*{font-family: arial; }
body {width: 600px; height:2400px; border-style:dotted;}
div {border-style: solid;}
div div {border-style: dashed; border-color: green; margin: 5px;}
div div div {border-style: dashed; border-color: red; margin: 5px;}

#menu1 { position:fixed; left:50; top:0;}
```

Hemos establecido un valor de height para body aleatoriamente grande para forzar que aparezcan barras de scroll vertical en la página web.

Al hacer scroll, comprobarás que la caja del elemento con id menu1 se mantiene fija en una posición de la página aunque nos desplacemos con el scroll.

El comportamiento del origen de coordenadas con fixed es igual al que hemos explicado con absolute.

EJERCICIO 1

Define un documento HTML con un div padre (divPadre), dentro del cual existan otras 3 cajas contenedoras div (div1, div2 y div3), cada una de ellas con unas dimensiones de 300x300px, 40 píxeles de margin en todas direcciones, 30 píxeles de padding en todas direcciones y un background color diferente. Usando posicionamiento relativo genera un desplazamiento de los div de la siguiente manera:

- a) El div 1 deberá desplazarse 200 píxeles a la derecha y 100 píxeles hacia abajo respecto a lo que sería su posición normal.
- b) El div 2 deberá desplazarse 50 píxeles a la izquierda y 50 píxeles hacia arriba respecto a lo que sería su posición normal.
- c) El div 3 deberá desplazarse 450 píxeles a la derecha y 300 píxeles hacia arriba respecto a lo que sería su posición normal.

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

EJERCICIO 2

Define un documento HTML con 3 cajas contenedoras div (div1, div2 y div3), la primera con unas dimensiones de 600x600px y un background color amarillo. La segunda con dimensiones 400x400px y un background color verde. La tercera con dimensiones 100x100px y background color azul. Usando posicionamiento absoluto establece para el div2 y el div3 el mismo origen que para el div1, de modo que el efecto generado sea ver un cuadrado amarillo dentro del cual hay un cuadrado verde dentro del cual hay un cuadrado azul. Para comprobar si tus respuestas son correctas puedes consultar en los foros aprenderaprogramar.com.

EJERCICIO 3

Define un documento HTML con varios div que contengan suficiente texto como para que la página se muestre con scroll (barras de desplazamiento). El primero de los div debe contener el texto <<Esta página web utiliza cookies. Si continúa navegando acepta el uso de cookies.>>, un valor height (altura) de 100 píxeles y color de fondo amarillo. Usando posicionamiento fixed, fija este div en la parte superior de la página de modo que se continúe visualizando aún cuando hagamos scroll. Para comprobar si tus respuestas son correctas puedes consultar en los foros aprenderaprogramar.com.

Próxima entrega: CU01033D

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