



APRENDERAPROGRAMAR.COM

PROPIEDAD DISPLAY CSS.
INLINE, BLOCK, NONE,
LIST-ITEM, INLINE-BLOCK.
EJERCICIOS DE EJEMPLO
RESUELTOS EN CÓDIGO
CSS (CU01033D)

Sección: Cursos

Categoría: Tutorial básico del programador web: CSS desde cero

Fecha revisión: 2029

Resumen: Entrega nº33 del Tutorial básico: "CSS desde cero".

Autor: César Krall

DISPLAY CSS

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). La propiedad display nos permite alterar el tipo de caja con que se muestra un elemento.



Sabemos que por ejemplo los elementos tipo título como h1 son elementos block, y por tanto de forma “natural” tienden a ocupar todo el ancho de la página. Por el contrario elementos como los links a o las imágenes son elementos inline. Veamos cómo la propiedad display permite alterar estas cualidades.

PROPIEDAD DISPLAY

Esta propiedad nos permite establecer el tipo de caja que el navegador empleará para visualizar un elemento, siendo los tipos más comunes inline y block, aunque existen bastantes otros.

La sintaxis a emplear es del tipo:

```
selectorElemento {display: especificaciónDeVisualización; }
```

PROPIEDAD CSS display	
Función de la propiedad	Permite definir el tipo de posición de caja para visualizar un elemento.
Valor por defecto	Depende del elemento (inline para elementos inline y block para elementos tipo block)
Aplicable a	Todos los elementos.
Valores posibles para esta propiedad	inline (el elemento se muestra en una caja inline)
	block (el elemento se muestra en una caja block).
	none (el elemento no se muestra; el efecto es como si no existiera, por lo que su espacio será ocupado por otros elementos)
	list-item (el elemento se comporta como si fuera un elemento li)
	inline-block (el elemento genera una caja block pero que se comporta como si fuera inline admitiendo otros elementos en la misma línea; el comportamiento se asemeja al de los elementos img)
	Otros que llevan a que el elemento simule el comportamiento de otro (inline-table, table, table-caption, table-cell, table-column, table-column-group, table-footer-group, table-header-group, table-row, table-row-group)

PROPIEDAD CSS display	
	Otros avanzados (flex, inline-flex, grid, inline-grid, run-in)
	inherit (se heredan las características del elemento padre).
Ejemplos aprenderaprogramar.com	#content1 {display: inline;} .elementoMonter {display: block;}

La propiedad display admite numerosos valores, pero los más usados son inline, block e inline-block. Son estos tres valores los que debemos aprender ahora. El resto los iremos conociendo progresivamente a medida que nos puedan resultar necesarios o cuando profundicemos en el conocimiento de CSS. Tener en cuenta también que estas otras propiedades a que nos referimos pueden no ser reconocidas por muchos navegadores.

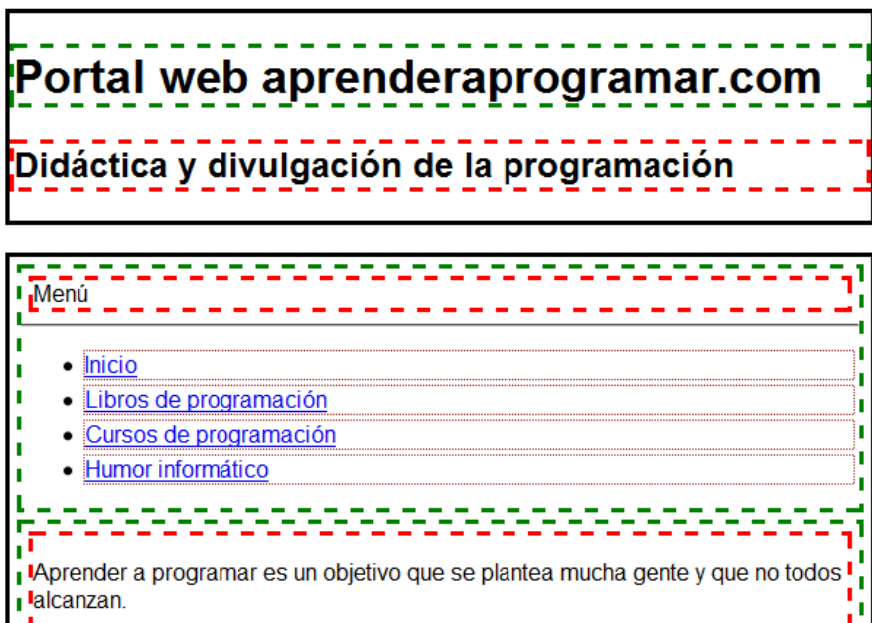
A continuación veremos algunos ejemplos de uso de las propiedades anteriormente especificadas. Para ello vamos a utilizar [el código HTML de base](#) que venimos empleando a lo largo del curso.

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;}
h1 {border-style: dashed; border-color: green;}
h2 {border-style: dashed; border-color: red; }
li {border-style: dotted; border-width:thin; border-color:brown; margin: 3px;}
    
```

Con esto lo único que hemos hecho es poner bordes y márgenes para visualizar las cajas que conforman la página web. El resultado será 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".

EJERCICIO RESUELTO

Sin ejecutar código, indica cuáles deberían ser los resultados obtenidos al añadir las líneas que se indican en la tabla al archivo CSS. Completa la tabla primero sin ejecutar el código. Luego, compara tu solución con la expuesta en la solución.

Fragmento de código añadido	Resultado
<code>li {display: inline;}</code>	
<code>img {display: block;}</code>	
<code>#menu1 {display:none;}</code>	
<code>h2 {margin-left:30px; display:list-item;}</code>	
<code>li {display: inline-block;}</code>	
<code>ul {display: table; }</code> <code>li {display: table-cell; padding:10px;}</code>	

SOLUCIÓN

Prueba a visualizar los resultados en tu navegador al añadir estos fragmentos de código y comprueba que obtienes los mismos resultados que se indican a continuación:

Fragmento de código añadido	Resultado
<code>li {display: inline;}</code>	Los elementos de item de menú que son li y que ocupaban cada uno una línea, se ponen en la misma línea. Sólo cuando no caben en una línea continúan en la siguiente.
<code>img {display: block;}</code>	Las imágenes que estaban en una sola línea unas junto a otras se sitúan cada una en una línea comportándose como elementos block
<code>#menu1 {display:none;}</code>	El menú (con sus cajas interiores, items de menú, etc.) desaparece completamente y su espacio es ocupado por el texto "Aprender a programar es..."
<code>h2 {margin-left:30px; display:list-item;}</code>	El título h2 aparece con una viñeta como si fuera un elemento li de una lista
<code>li {display: inline-block;}</code>	A diferencia de establecer sólo inline, si un elemento salta de línea lo hace completamente por ser un bloque. Su texto no puede cortarse entre líneas como ocurriría si fuera inline.
<code>ul {display: table; } li {display: table-cell; padding:10px;}</code>	Hace que la lista se comporte como una tabla donde cada item es una celda.

Hay una curiosidad. En la imagen donde vimos las cajas podemos observar que el elemento h2 tiene un espacio libre por arriba y por debajo, que corresponde a un margin-top y margin-bottom automático que aplica el navegador por defecto para este tipo de elementos. Si escribimos `h2 {display: inline;}` comprobamos no sólo que la caja deja de ocupar todo el ancho disponible sino que los márgenes desaparecen ¿Por qué? Porque en los elementos inline no son aplicables (o se ignoran si están establecidas) ciertas propiedades como width, height, margin y float. De este modo, al cambiar el formato de caja no sólo estamos afectando a que los elementos ocupen una línea a todo lo ancho o no, sino también otras propiedades que pueden quedar desactivadas cuando se cambia el tipo de caja usando la propiedad display. Tener en cuenta también que los elementos inline sólo pueden contener otros elementos inline o texto en su interior.

Otra curiosidad. Posiblemente no obtengas nada coherente, pero comprueba qué ocurre si como css utilizas esta única declaración `*{font-family: arial; display: inline-block;}`

Si te surgen dudas puedes consultar en los foros aprenderaprogramar.com.

Próxima entrega: CU01034D

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