



APRENDERAPROGRAMAR.COM

PROPIEDADES CSS WIDTH
Y HEIGHT. VALOR POR
DEFECTO AUTO
(AUTOMÁTICO) EJEMPLOS
PRÁCTICOS Y EJERCICIOS
RESUELTOS (CU01025D)

Sección: Cursos

Categoría: Tutorial básico del programador web: CSS desde cero

Fecha revisión: 2029

Resumen: Entrega nº25 del Tutorial básico: "CSS desde cero".

Autor: César Krall

WIDTH Y HEIGHT

Dos propiedades básicas en CSS son width (ancho) y height (alto). Las usaremos con frecuencia para determinar el tamaño de muchos elementos en nuestro diseño CSS. Estas propiedades serán designadas con alguna de las unidades de medida que hemos estudiado previamente, usando la que resulte más adecuada en función del elemento al que se vayan a aplicar.



PROPIEDADES CSS width y height	
Valor por defecto	auto (en este caso, es determinado por el navegador)
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	Una unidad de medida absoluta como in, cm, mm, pt, ó pc
	Una unidad de medida relativa como %, em ó ex
	Una unidad de medida híbrida como px (pixel)
	inherit (indica que se hereda del elemento padre)
Ejemplos aprenderaprogramar.com	<pre>html {height: 100%;} body {background-color: yellow; width: 900px;} .logo-module {width: 16em;}</pre>

Recordemos que la sintaxis a emplear es la siguiente:

```

selectorDeElemento {width: valorDePropiedadUnidades;
                        height: valorDePropiedadUnidades;}
```

Un aspecto importante es que las propiedades width y height no son aplicables a cualquier elemento. Por ejemplo elementos de tipo inline como ... no admiten estas propiedades. Si intentamos aplicar width o height a un elemento que no lo admite, el navegador simplemente lo ignorará.

Las propiedades width y height no siempre se especifican. Se puede especificar una de ellas, ambas o ninguna. Para aquella propiedad que no se especifica, el elemento tomará como anchura y altura el valor que por defecto o por herencia le sean de aplicación.

No debe existir espacio entre el valor establecido y las unidades cuando escribamos el código. Por ejemplo si establecemos #menu1 {width: 80%;} no debe existir espacio entre 80 y %. Si establecemos #menu1 {width: 600px;} no debe existir espacio entre 600 y px. En el caso de unidades que se expresen con decimales, la separación decimal debe indicarse con un punto en lugar de con una coma. Por ejemplo #menu1 {font-size: 12.75pt;} es correcto pero #menu1 {font-size: 12,75pt;} es incorrecto y no será interpretado por el navegador.

En el caso de decimales inferiores a uno, se admite omitir el 0 previo al punto decimal o el 0 de terminación. Por ejemplo sería tan válido escribir #margen1 {width: 0.90%;} como #margen1 {width: .9%;} como #margen1 {width: .90%;}

En el caso de valores cero (nulo), no es necesario indicar unidades de medida. Por ejemplo #soundCode1 {width: 0px;} es igual de válido que #soundCode1 {width: 0;}

EJEMPLO PRÁCTICO

Define los siguientes estilos para el documento HTML que estamos usando como [base de ejemplo](#) para el curso y visualiza el resultado:

```
/* Curso CSS estilos aprenderaprogramar.com */
html{background-color: pink;}
body {background-color: yellow; width: 800px;}
h1{background-color: blue; width: 50%;}
h2{background-color: red; width: 8em}
```



Con los conocimientos que hemos ido adquiriendo a lo largo del curso ya hemos de ser capaces de interpretar qué es lo que visualizamos. Vamos a hacerlo. El fondo se muestra rosa porque hemos definido que el background-color del elemento html sea rosado. En amarillo vemos el elemento body. El ancho del elemento body es de 800 px. El elemento body no tiene el mismo ancho que el elemento html por dos motivos: por un lado, la pantalla donde se ha visualizado la web para capturar la imagen que hemos mostrado tiene una resolución de 1024 x 768 pixeles. Como nosotros hemos definido que el

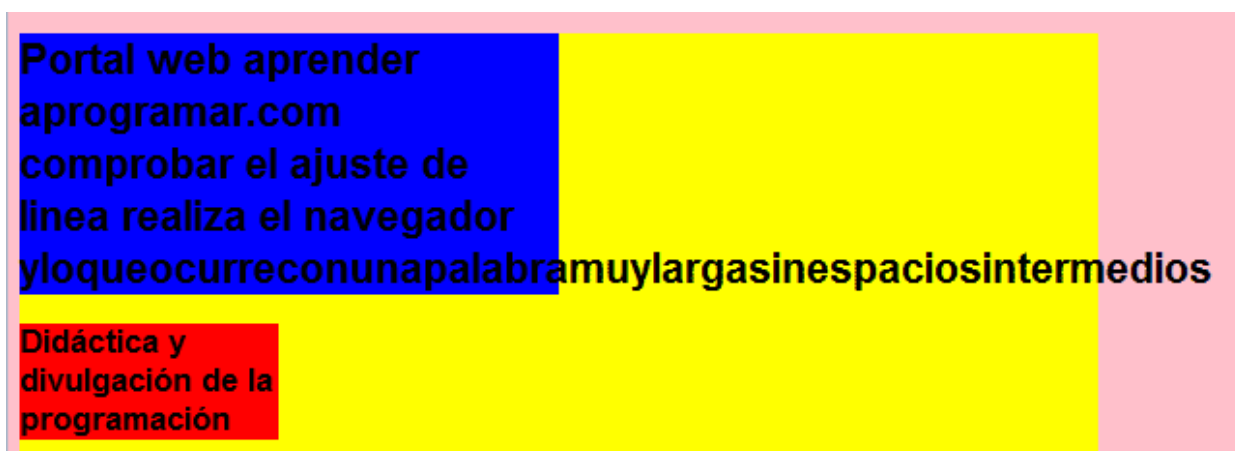
elemento body tendrá 800 px, el resto que falta hasta los 1024 px disponibles se ven con el color de fondo del elemento html. Otro motivo por el que el elemento body no tiene el mismo ancho que el elemento html es que el navegador introduce unos márgenes por defecto (ya veremos más adelante cómo se pueden anular). Prueba a cambiar el width de body y ponle distintos valores como 200 px y 2400 px. Para valores de ancho que exceden el tamaño máximo en pixeles de anchura de la pantalla el navegador incorporará automáticamente unas barras de desplazamiento o scrolls.

El elemento h1 con el texto "Portal web aprenderaprogramar.com" tiene un ancho del 50%. ¿Pero el 50% de qué? Pues el 50% de su elemento contenedor (elemento dentro del cual está o elemento padre). Como su elemento contenedor es body y éste tiene 800 px de ancho, el ancho de h1 es la mitad de este valor, es decir, 400 px.

Con el elemento h1 comprobamos una cosa: que el texto se muestra en dos líneas y que el tamaño del texto de la segunda línea excede en anchura al tamaño de su elemento contenedor, el h1 con anchura de 400 px. Por tanto ya sabemos que el texto ocupa algo más de 400 px y por ello se sale del espacio azul y ocupa parte del espacio amarillo. ¿No sería más lógico que el texto se cortara cuando llegara al límite de su contenedor? ¿O que siguiera en la siguiente línea en vez de salirse hacia la derecha? Quizás, pero el comportamiento que observamos está determinado por el navegador que estamos usando. Se está aplicando la siguiente regla: si el texto excede el ancho de su elemento contenedor, se crean nuevas líneas aumentando la altura (height) automáticamente siempre que existan espacios dentro del texto que permitan introducir el salto de línea. Si no existe espacio para introducir ese salto de línea, el texto se expande hacia la derecha sobrepasando el tamaño de su contenedor.

El elemento h2 tiene fijado un width de 8 em. El em es una medida relativa al tamaño de letra que se esté usando. ¿A qué tamaño de letra en este caso? Pues al tamaño de letra del h2, que como no lo hemos especificado, lo determina por defecto el navegador. El tamaño por defecto puede variar entre navegadores, pero de forma aproximada podemos decir que serán similares a estos: 16px (1em o 100%) como tamaño normal de fuente, h1 de tamaño 2.25em (36px) y h2 de 1.5em (24px). En este caso hemos comprobado que el navegador que usamos aplica 24px de tamaño al h2, entonces 8em se interpreta como 8 veces el tamaño de fuente del elemento, resultando $8 \times 24 = 192$ pixeles.

Para verificar que el navegador maneja la propiedad height automáticamente (a falta de especificación por nuestra parte su valor es **auto**) y lo que ocurre cuando el texto no se puede ajustar al espacio disponible fijate en la siguiente imagen que refleja lo que ocurre al escribir un texto más largo:



Como vemos el texto se sale del elemento contenedor h1, superponiéndose sobre el elemento padre, body, y al tener más ancho aún se expande hasta ocupar espacio del elemento padre del padre o "abuelo", el elemento html. Si el texto se alargara aún más el navegador introduciría un scroll para permitir que se pueda leer.

Para entender mejor la diferencia entre que width y height estén especificados con un valor o que no lo estén y mantengan el valor por defecto "auto" fíjate en la siguiente imagen. Hemos escrito un texto muy largo en el h1 y en el lado izquierdo vemos lo que ocurre si definimos h1{background-color: blue; width: 50%;} mientras que en el lado derecho vemos el resultado para el código h1{background-color: blue; width: 50%; height: 150px;}



En el lado izquierdo vemos cómo el navegador adapta automáticamente el valor height del h1 para que el texto no se salga de él. Al no estar especificada la propiedad height tiene valor "auto". En el lado derecho, al tener restringida la altura del h1 a 150 pixeles, cuando el texto no cabe en su contenedor continúa expandiéndose hacia abajo.

Lo que nos interesa de todo esto es ser capaces de interpretar lo que ocurre cuando se obtiene una visualización errónea o distorsionada debido al código CSS. Hay que tener en cuenta, sin embargo, que los problemas de visualización en páginas web pueden tener otros orígenes distintos del CSS como el código de programación, configuraciones de servidor o navegador, etc.

EJERCICIO RESUELTO

Crea una hoja de estilos css de forma tal que el elemento body tenga el 100% del ancho de la pantalla, los elementos h1 un 80% y los elementos h2 un 60%. Escribe un CSS para obtener la misma visualización pero expresando los valores en pixeles.

SOLUCIÓN

A continuación proponemos una solución. Si tu respuesta no coincide con lo que indicamos o tienes dudas, puedes consultar en los foros aprenderaprogramar.com.

La definición de estilos en % sería la siguiente. Hemos añadido colores de fondo para que se visualice mejor:

```
/* Curso CSS estilos aprenderaprogramar.com */
html {background-color: pink; }
body {background-color: yellow; width: 100%;}
h1 {background-color: blue; width: 80%;}
h2 {background-color: red; width: 60%;}
```

Comprueba el resultado al aplicar estos estilos al código HTML base que estamos usando para el curso. Si quisiéramos realizar una definición equivalente en pixeles, tendríamos que consultar la resolución en pixeles de la pantalla con que estamos trabajando (en Windows se puede ver yendo al botón Inicio, luego "Panel de Control" y luego "Ajustar resolución de pantalla")

Supongamos que estamos en un monitor de 1280 x 800 pixeles. El 100 % son los 1280 pixeles, el 80 % son $1280 \times (80/100) = 1024$ pixeles y el 60 % son $1280 \times (60/100) = 768$ pixeles. En este caso tendríamos que escribir:

```
/* Curso CSS estilos aprenderaprogramar.com */
html {background-color: pink; }
body {background-color: yellow; width: 1280px;}
h1 {background-color: blue; width: 1024px;}
h2 {background-color: red; width: 768px;}
```

Supongamos que estamos en un monitor de 1366 x 768 pixeles. En este caso el 100 % son los 1366 pixeles, el 80 % 1092.8 pixeles y el 60 % son 819.60 pixeles. El código a escribir sería:

```
/* Curso CSS estilos aprenderaprogramar.com */
html {background-color: pink; }
body {background-color: yellow; width: 1366px;}
h1 {background-color: blue; width: 1092.8px;}
h2 {background-color: red; width: 819.6px;}
```

Para otras resoluciones de pantalla tendríamos que hacer los mismos cálculos.

Fijate que usando tanto por ciento la representación o aspecto será igual en todos los dispositivos o pantallas, mientras que usando pixeles el aspecto depende de la pantalla en que estemos viendo la web. De ahí que con frecuencia se aluda al pixel como una medida absoluta o híbrida.

Próxima entrega: CU01026D

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