

CSS: Hojas de estilo en Cascada

¿Qué es CSS? (Definición del World Wide Web Consortium)

Hojas de Estilo en Cascada (Cascading Style Sheets), es un mecanismo simple que describe cómo se va a mostrar un documento en la pantalla, o cómo se va a imprimir, o incluso cómo va a ser pronunciada la información presente en ese documento a través de un dispositivo de lectura. Esta forma de descripción de estilos ofrece a los desarrolladores el control total sobre estilo y formato de sus documentos.

CSS se utiliza para dar estilo a documentos HTML, separando el contenido de la presentación. Los estilos definen la forma de mostrar los elementos HTML. CSS permite a los desarrolladores Web controlar el estilo y el formato de múltiples páginas Web al mismo tiempo.

¿Cómo se define el formato?

La estructura básica es:

```
selector { propiedad: valor;
           propiedad: valor;
           .....
           propiedad: valor;}
```

Ejemplo:

```
p { color: red;
    text-decoration: underline;
    text-align: center;}
```

p es el selector, color¹ la propiedad y red, el valor.

Cómo aplicar CSS

Para vincular una hoja de estilos con un documento HTML o XHTML hay tres opciones. Si hay más de una opción definida, prevalece la primera, y luego la segunda.

1) Estilo incorporado directamente con el elemento:

En el mismo documento HTML escribimos, por ejemplo:

```
<h1 style="color:red;">Título importante</h1>
```

Salvo en casos muy puntuales, esta forma es muy poco recomendable.

2) Estilo incorporado en el <head>.

Se recomienda cuando el formato va a ser usado por un único archivo html.

En el mismo documento HTML escribimos, por ejemplo:

```
<!DOCTYPE html>
<html lang="es-ar">
<head>
<meta charset="UTF-8">
  <title>Título de la ventana</title>
<style type="text/css">
p {
  color: red;
  text-align: center;}
</style>
</head>
<body>
.....
```

¹ En CSS, se usa la ortografía estadounidense *color*, en lugar de la británica *colour*.

3) Estilo en un archivo aparte.

Se recomienda cuando el formato va a ser usado por varios archivos html.

En el archivo HTML, dentro del <head>, escribimos:

```
<!DOCTYPE html>
<html lang="es-ar">
<head>
<meta charset="UTF-8">
  <title>Título de la ventana</title>
  <!-- La siguiente línea vincula este archivo con la hoja de estilo: -->
  <link rel="stylesheet" href="hoja_de_estilo.css" type="text/css" />
</head>
<body>...</body></html>
```

En un archivo aparte, que, en este ejemplo, se llama “hoja_de_estilo.css” (podría tener cualquier nombre) escribimos las reglas CSS, por ejemplo:

```
p { color: red;
    text-align: center;}
h1 {color: blue;
     text-decoration: underline;}
```

Este archivo no lleva ningún tipo de encabezado, sólo las reglas CSS una tras otra.

Selectores

Etiquetas: son el nombre de una etiqueta HTML. Por ejemplo: p, h1, li, etc.

Clases: definen el estilo para los elementos de una clase. Comienzan con un punto.

Ejemplo: .verde {color: green;} dará formato (color verde) a todos los elementos que hayan sido definidos como de clase “verde” en el código HTML. Por ejemplo:

```
<h1 class="verde">      <p class="verde">      <div class="verde"> etc.
```

Etiqueta y clase: Siguiendo el ejemplo anterior, pero especificándolo más:

```
p.verde {text-align: justify;}
```

dará alineación justificada a los párrafos de la clase “verde” (<p class=“verde”>), pero no a otros elementos de clase “verde” (como <h1 class=“verde”>, por ejemplo).

Pseudoclases: Son un caso muy particular, y se aplican fundamentalmente a la etiqueta “a” (enlaces).

a:link define las propiedades para el enlace no visitado.

a:visited define las propiedades para el enlace visitado.

a:hover define las propiedades para el enlace cuando el puntero del ratón pasa por sobre él.

Identificadores: definen el estilo para el elemento nombrado con un determinado identificador en el código HTML. Comienzan con el símbolo “#”. Ejemplo:

```
#encabezado {background-color: blue;}
```

Dará fondo azul al elemento identificado como “encabezado”. El elemento podría ser

```
<div id="encabezado"> o bien <p id="encabezado"> o <h1 id="encabezado">
```

Selectores agrupados:

Separados por comas: la coma podría traducirse como “y”:

img, table {border: 0;} Las imágenes y las tablas no tendrán borde.

a, p, h1, .verde {color: green;} dará color verde a los enlaces, párrafos, títulos principales y elementos de la clase “verde”.

Separados por espacios en blanco: el espacio podría traducirse como “contenido en”, leyéndolo de derecha a izquierda:

p strong {color: red;} Los que estén dentro de párrafos serán rojos; los otros no.

ul li {text-decoration: underline} Los elementos de las listas no numeradas estarán subrayados, pero los otros no.

Unidades de medida

Hay distintas unidades para expresar tamaños y distancias:

Relativas: dependen de otro elemento.

- ✓ **Píxeles:** Las imágenes en los monitores están formadas por pequeños puntitos llamados píxeles. Para indicar una distancia de 3 píxeles escribiremos 3px
- ✓ **em²:** Es el tamaño de la letra “m”. Para indicar una distancia equivalente a 3 letras “m”, escribiremos 3em
- ✓ **Porcentaje:** Refiere a un porcentaje del elemento contenedor. Supongamos que tenemos una imagen dentro de un <div>, y queremos que dicha imagen ocupe la mitad del ancho del <div>, entonces le daremos a la imagen un tamaño del 50%.

Absolutas: están definidas en sí mismas.

- ✓ **Centímetros o milímetros:** Si queremos una distancia de 3 centímetros (o milímetros) escribimos 3cm (o 3mm)
- ✓ **Pulgadas³:** Si queremos una distancia de 3 pulgadas escribimos 3in
- ✓ **Puntos:** Cuando se utiliza un procesador de texto, (como Word u OpenOffice), es la medida que se usa en esos programas para el tamaño de la letra (“Arial 10”, por ejemplo). Un punto equivale a 1/72 pulgada. Si queremos una letra de 9 puntos escribimos 9pt

Aunque, en principio, parezca más sencillo utilizar medidas absolutas, es muy recomendable utilizar las medidas relativas, porque se adaptan mejor a las necesidades de los lectores de nuestra página Web.

Herencia

Además de las propiedades que se definan para el propio elemento, éstos *heredan* propiedades de otros elementos. Si un valor heredado contradice a uno definido para el propio elemento, este último tiene prioridad. Podríamos definir dos clases de herencia:

Por inclusión:

Un elemento que está dentro de otro hereda las propiedades de este último. Ejemplo:

Código HTML:

```
<p> Esto es <strong>muy importante</strong>.</p>
```

CSS:

```
p {color:red;}
strong {text-decoration: underline;}
```

Se definió que el será subrayado, y que el párrafo será rojo. Pero no se especificó de qué color será el . Por lo tanto, tomará el color del elemento que lo contiene (<p>), es decir, también será rojo. Resultado: **Esto es muy importante.**

Por generalidad:

Un elemento que pertenece a una clase o tiene asignado un identificador, hereda del caso más general. Ejemplo:

Código HTML:

```
<p class="especial">Esto es un párrafo especial.</p>
<p id="nombre">Don Quijote de la Mancha. </p>
```

CSS:

```
p {color:red;}
.especial {text-decoration: underline;}
#nombre {text-align: center;}
```

El párrafo de clase “especial” estará subrayado, y el identificado con id=“nombre” estará centrado. Pero en ningún caso se especificó cuál será el color. Pero como hemos definido en el primer renglón que todos los párrafos serán rojos, tanto el de class=“especial” como el de id=“nombre” serán de este color.

2 ¡No tiene nada que ver con la etiqueta (texto enfatizado) de HTML!

3 Una pulgada son 2,54cm. “Pulgada” en inglés se dice “inch”, de ahí que escribamos “in” como abreviatura.

Propiedades y valores

Si bien podemos escribir las propiedades directamente en el editor de textos (como *gedit* o *bloc de notas*), nosotros utilizaremos el programa “Asistente CSS”, que está en la página <http://escuelapadrejose.com.ar/tercero>

Este programa nos permitirá elegir las propiedades que deseemos de una lista, y nos mostrará en la parte inferior el código que deberemos copiar y pegar en el editor. Cuando ya hayas hecho muchas hojas de estilo CSS, probablemente te acuerdes de memoria casi todas las propiedades, y ya no necesites el “Asistente CSS”.

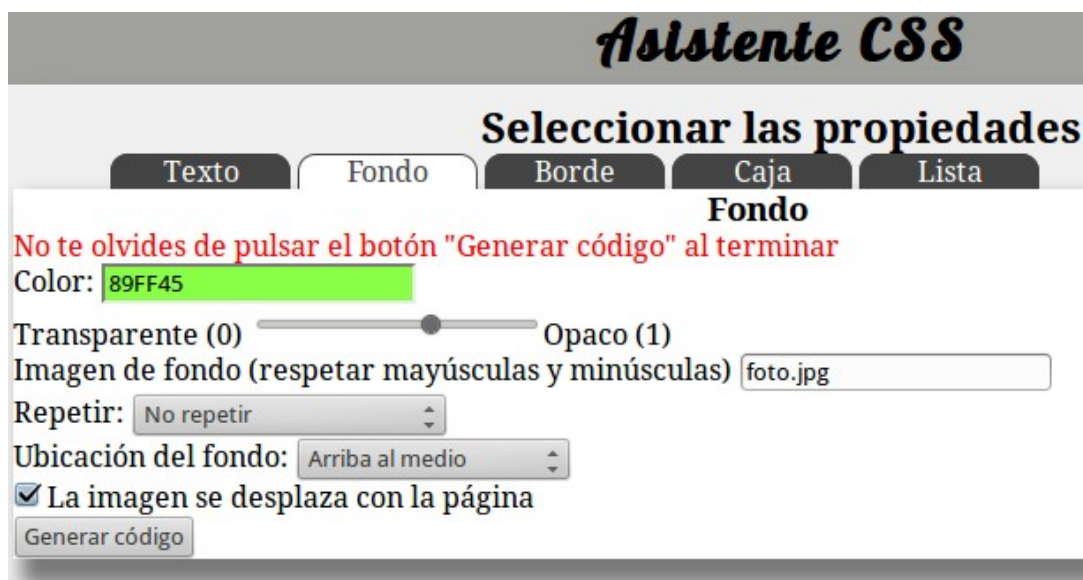
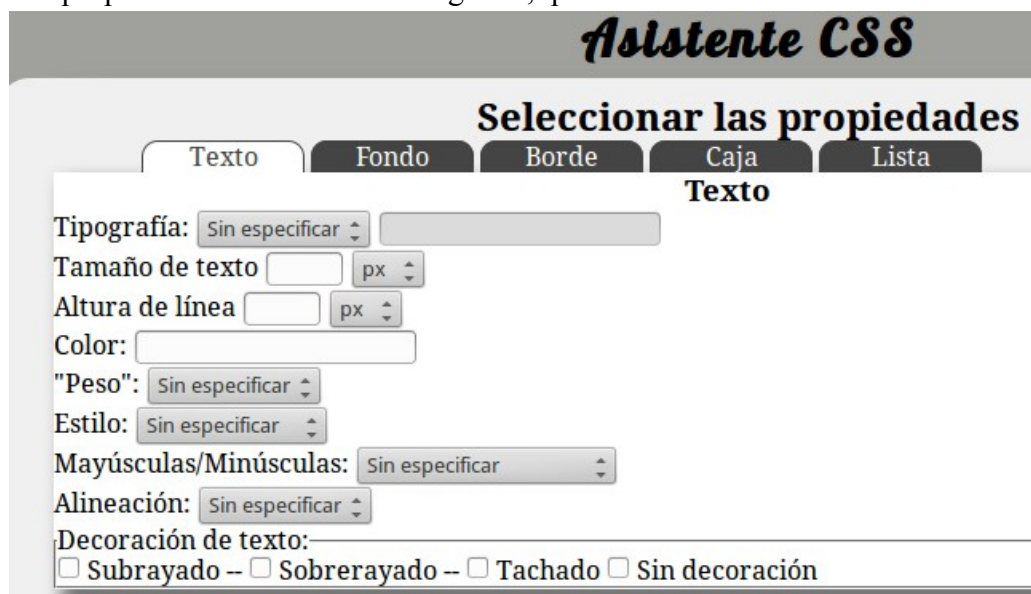
Este programa divide a las propiedades en diferentes categorías, que veremos a continuación.

Texto

Reune las propiedades de texto. No requieren demasiadas explicaciones.

Fondo

Para darle formato al fondo de un elemento, existen las siguientes propiedades:



- Color: Color de fondo. (Puede hacerse semi-transparente, aunque no funciona en todos los navegadores).
- Imagen de fondo: Podemos incluir el nombre de un archivo de imagen.
- Repetir: Para definir si la imagen de fondo se repetirá o no.
- Ubicación de fondo: Punto de comienzo de un fondo.
- Podemos definir también si la imagen seguirá el desplazamiento de la página o no.

Importante: las imágenes utilizadas como fondo no pueden ser redimensionadas.

Borde

Seleccionar las propiedades

Texto Fondo **Borde** Caja Lista

Bordes

No te olvides de pulsar el botón "Generar bordes" al terminar

☒ Los 4 bordes son iguales

Los 4 bordes:

Tipo de borde: No especificar

Grosor del borde: Píxeles (px)


Color:

Transparente (0) Opaco (1)

☒ Las 4 esquinas son iguales

No te olvides de pulsar el botón "Generar esquinas" al terminar

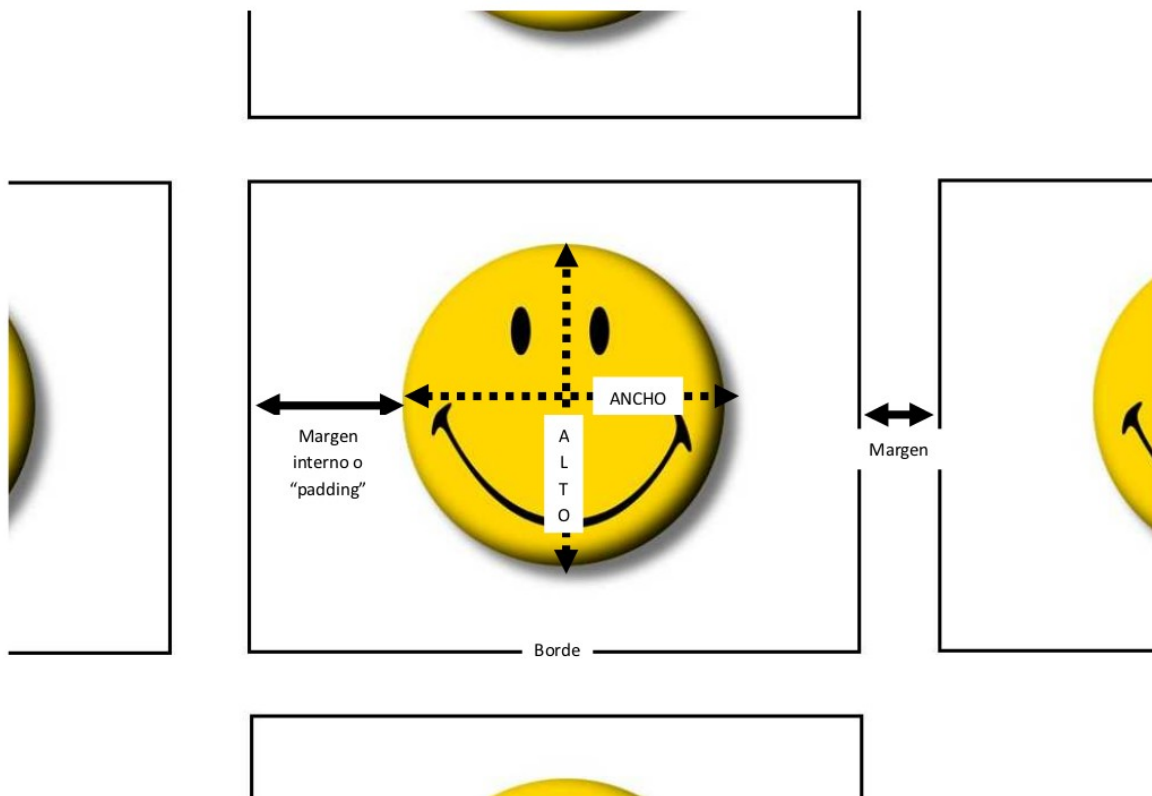
Tipo de esquinas: Ángulo recto Radios:



Define las propiedades de los bordes y las esquinas, para los cuatro a la vez, o bien para cada uno en particular.

No requiere mayores explicaciones.

Modelo de caja



Si en el ejemplo anterior, definimos:

Ancho: 100px – Alto: 100px – Margen interno: 20px – Margen: 15 px – Borde: 2px

El ancho total que ocupa la caja será:

Margen izq+ Borde izq + Margen interno izq + Ancho + Margen interno dcho + Borde dcho + margen dcho
15px + 2 px + 20 px + 100 px + 20px + 2 px + 15px = 174px

Si hay dos elementos uno al lado del otro, los márgenes se suman. Si hay dos elementos uno

debajo del otro, el espacio que hay entre ambos es igual al mayor de los márgenes, el menor desaparece.

Nuevamente en el ejemplo anterior, suponiendo que el mismo elemento se incluyó cinco veces (como muestra la figura), cada una de ellas con márgenes de 15px. Entre las figuras que están juntas habría 30px de separación (los márgenes se suman). Entre las figuras que están una debajo de la otra habría 15px de separación (los márgenes se superponen).

Si defino los márgenes izquierdo y derecho como “auto”, el elemento aparecerá centrado horizontalmente. Si defino como “auto”, los márgenes superior e inferior, el elemento aparecerá centrado verticalmente.

Modelo de caja: Ancho, altura y márgenes.

En la siguiente imagen, se define el **ancho**: 50% del elemento que lo contenga. Pero, al mismo tiempo, se le define una anchura máxima (11cm) y una anchura mínima (6cm). Esto significa que la anchura será del 50% de su elemento padre, siempre y cuando no supere los 11cm ni sea menor de 6cm.

Con la **altura**, se sigue un razonamiento similar.

Las columnas “**Márgenes**” y “**Márgenes internos**” responden a lo que ya se explicó.

Ancho 50 % - Mínimo 6 cm - Máximo 11 cm

Altura 85 % - Mínima 500 px - Máxima 1000 px

Márgenes:

Superior: [] px ☐ Automático

Derecho: [] px ☐ Automático

Inferior: [] px ☐ Automático

Izquierdo: [] px ☐ Automático

Márgenes internos (padding):

Superior: [] px ☐ Automático

Derecho: [] px ☐ Automático

Inferior: [] px ☐ Automático

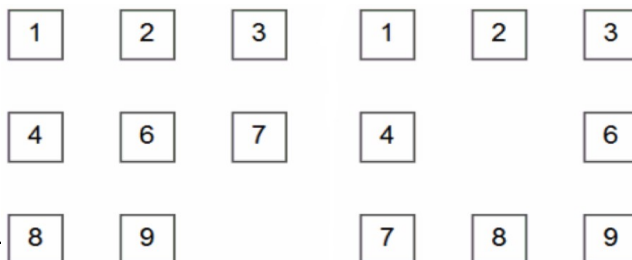
Izquierdo: [] px ☐ Automático

“Mostrar como” y “visibilidad”

La opción “**Mostrar como:**” permite mostrar un elemento como “en línea” o “de bloque”⁴, independientemente de las características del elemento. Por ejemplo, un enlace (etiqueta <a>) es un elemento en línea, pero, mediante esta opción, puede mostrarse como elemento de bloque. Si elegimos la opción “No mostrar”, es como si el elemento no existiera.

La opción “**Visibilidad**” permite ocultar elementos. En este ejemplo se ve la diferencia con “Mostrar como: no mostrar”, respecto al cuadrado “5”.

mostrar como: no mostrar visibilidad: oculto



⁴Ver apunte HTML.

“Flotar” y “Despejar”

La propiedad **flotar** posiciona el elemento lo más a la derecha o lo más a la izquierda posible, y hace que el resto de los elementos “fluyan” a su alrededor.



`<div id="menu">` sin “flotar”



`<div id="menu">` “flotando” a la izquierda

La propiedad **despejar**: **izquierda**; hace aparecer el elemento en la primera línea en la que no haya ningún otro elemento “flotando” a la izquierda. El mismo razonamiento vale para **derecha** y **ambos**. En la siguiente imagen se le aplicó la propiedad **flotar**: **izquierda**; al menú y **despejar**: **izquierda**; al párrafo:

Las mejores noticias de diseño web

Menú principal

- [Noticias en diarios](#)
- [Noticias en blogs](#)
- [Libros recomendados](#)

El diseño es el código

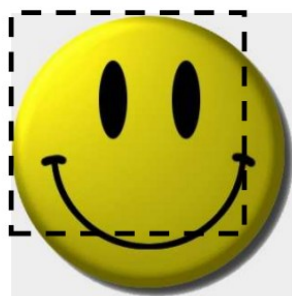
Por Miguel Ripoll

El congreso **El futuro del diseño web**, celebrado en Londres, ha marcado las tendencias sobre hacia dónde va la industria del diseño web. Código limpio, accesibilidad, usabilidad y los estándares del Consorcio **W3C** son el camino por el que la industria debe ir, a pesar de los esfuerzos de las grandes empresas por vender su visión cerrada y propietaria de la Red.

Los dos principales patrocinadores de la reunión de Londres, Adobe y Microsoft, que promocionan Flash y Silverlight, respectivamente, recibieron durante sus

Desbordamiento (overflow)

Explicaremos la propiedad **desbordamiento** con un ejemplo. Supongamos que tenemos una imagen de 300px de ancho, contenida en un `<div>` de 200px de ancho. Esta propiedad indica cómo se comportará la página respecto a esos 100px sobrantes. Si elegimos **visible**; la imagen se ve completa. Si elegimos **oculto**; la parte de la imagen que está fuera del `<div>` no se ve. Si elegimos **barras de desplazamiento**, el `<div>` tendrá barra de desplazamiento para ver la imagen. Si elegimos **automático**; se mostrará la barra de desplazamiento sólo en caso de que el elemento no quepa en el contenedor.



visible



hidden



scroll

Posición

La opción posicionamiento tiene las opciones: **estático**; **fijo**; **absoluto**; y **relativo**; que explicaremos en detalle.

estático es la posición “natural” de los elementos, la que tienen cuando no se define ninguna posición.

absoluto define la posición del elemento respecto a un punto fijo. Cabe preguntarse respecto de qué se posicionan los elementos. Respuesta: respecto del ángulo superior izquierdo del primer elemento que lo contenga cuya posición no sea “estático”. Los otros elementos actuarán como si el elemento posicionado de forma absoluta no existiera.

relativo ubica al elemento en su posición estático y luego le aplica el desplazamiento. Así, si la posición es relativa y el desplazamiento “top” es 50px, el elemento se posicionará 50px debajo de su posición “estática”. Los otros elementos actúan como si el elemento estuviera en posición “estático”.

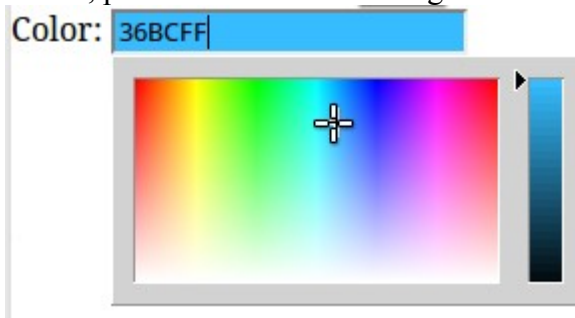
fijo posiciona al elemento respecto del ángulo superior izquierdo de la página. Cuando la página se desplaza, el elemento permanece fijo.

Para indicar el desplazamiento del elemento (tanto para el posicionamiento fijo, absoluto o relativo), se utilizan las propiedades **top**; **left**; **right**; y **bottom**;

Posicionar elementos puede causar que los mismos se solapen (se “tapan” entre ellos). Para indicar cuál irá delante y cuál detrás, se puede usar la propiedad **z-index: N**; donde “N” es un número entero (a mayor “N”, más adelante se ubica el elemento).

Colores

Si bien se pueden usar algunos colores por su nombre en inglés, para la mayoría de los casos se establece un código hexadecimal, podemos tomar esos códigos del selector de colores.



Bibliografía

- Eguíluz Pérez, Javier: “Introducción a CSS”. España, junio de 2008. <http://www.librosweb.es/css>
- World Wide Web Consortium: “Guía breve de CSS”. <http://www.w3c.es/divulgacion/guiasbreves/HojasEstilo>
- World Wide Web Consortium: “Guía de referencia CSS”. <http://www.w3c.es/Divulgacion/GuiasReferencia/CSS21/>
- W3C Schools: <http://www.w3schools.com/css/>

El conocimiento es libre: este documento está bajo una licencia Creative Commons 2.5 Argentina Atribución-No comercial-Compartir Igual. <http://creativecommons.org/licenses/by-nc-sa/2.5/ar/>
Este documento fue redactado y diseñado utilizando **Software Libre**.