

Tema 21. Hojas de estilos.

Indice

1. Los selectores CSS3.....	2
1.1. Selectores jerárquicos.....	2
1.2. Pseudoclases de selección.....	5
1.3. Selectores de atributos.....	9
2. Bordes redondeados.....	12
3. Bordes decorados.....	18
4. Sombras.....	22
4.1. Sombras sobre el texto.....	22
4.2 Sombras sobre un elemento caja.....	24
5. Tipos de letra personalizados.....	26
6. Las letras huecas.....	31
7. Manejar títulos largos.....	34
8. Los guiones automáticos.....	36
9. Redimensionar un elemento.....	37
10. Ajustar la altura de los caracteres.....	39
11. Fondos múltiples.....	42
12. Degradado de color.....	43
12.1. Los degradados lineales.....	43
12.2. Los degradados circulares.....	47
13. Opacidad y transparencia.....	50
14. Columnas múltiples.....	53
15. Transformaciones.....	55
16. Transiciones.....	60
17. Los filtros gráficos.....	65
18. Las máscaras.....	67
19. La fusión de imágenes.....	69

1. Los selectores CSS3

Los selectores básicos de CSS 1 son bastante limitados. Las especificaciones CSS2 y CSS3 agregan multitud de selectores.

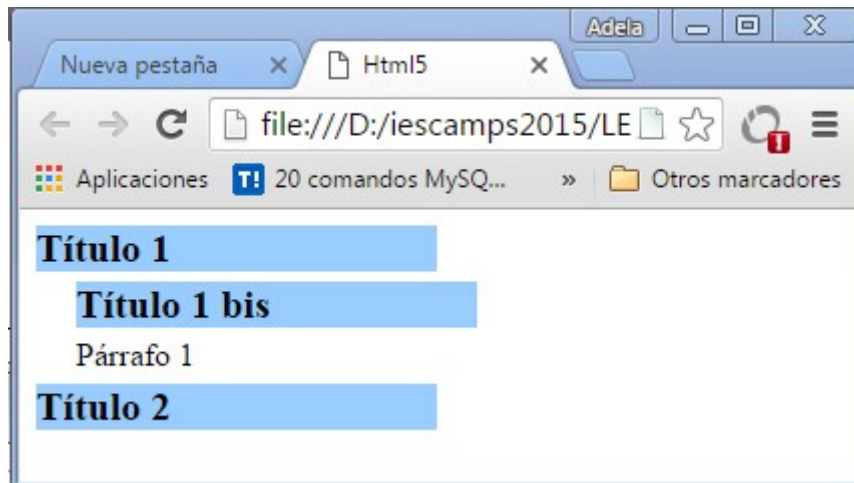
1.1. Selectores jerárquicos

A B	Descendentes	Selecciona cualquier elemento B que es descendiente del elemento A.
A > B	Hijos	Selecciona cualquier elemento B que es hijo directo del elemento A.
A + B	Hermano adyacente	Selecciona cualquier elemento B precedido inmediatamente del elemento A.

Ejemplo 1

```
<!DOCTYPE html>
<html lang="es">
<head>
<title>Html5</title>
<meta charset="UTF-8">
<style type="text/css">
#division h1 { width: 200px;
                background-color: #9cf;}
h1 { margin-top: 5px;
      margin-bottom:5px;
      font-size: 20px;}
p { margin-top: 5px;
    margin-bottom:5px;}
</style>
</head>
<body>
<div id="division">
<h1>Título 1</h1>
<div id="contenido" style="padding-left: 20px;">
<h1> Título 1 bis</h1>
```

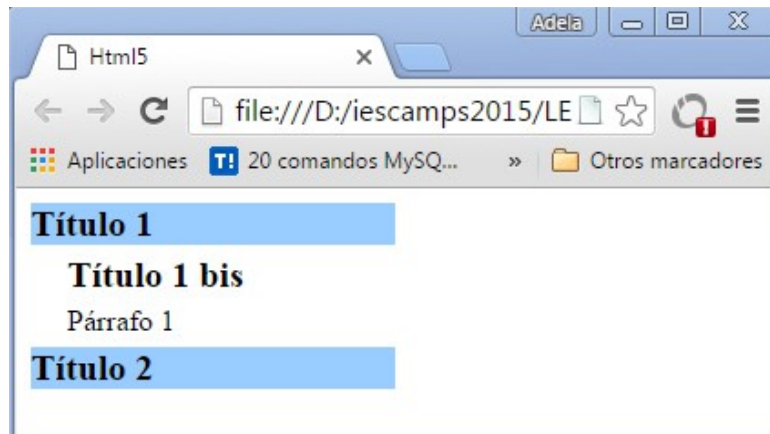
```
<p>Párrafo 1</p>
</div>
<h1>Título 2</h1>
</div>
</body>
</html>
```



Ejemplo 2

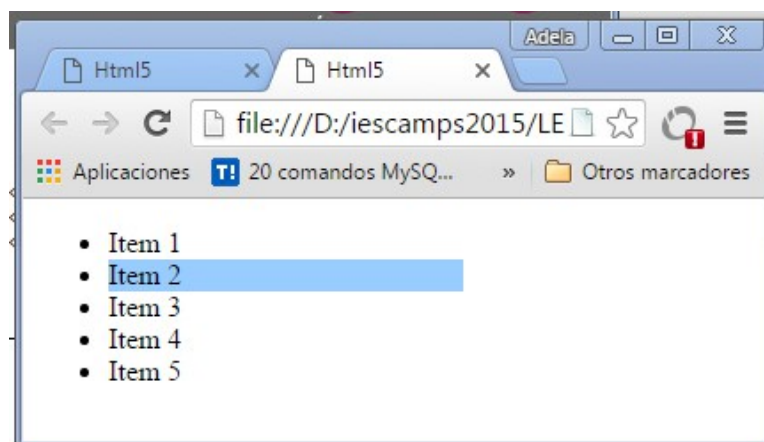
```
<!DOCTYPE html>
<html lang="es">
<head>
<title>Html5</title>
<meta charset="UTF-8">
<style type="text/css">
#division > h1 { width: 200px;
                 background-color: #9cf;}
h1 { margin-top: 5px;
     margin-bottom: 5px;
     font-size: 20px;}
p { margin-top: 5px;
   margin-bottom: 5px;}
</style>
</head>
<body>
<div id="division">
<h1>Título 1</h1>
<div id="contenido" style="padding-left: 20px;">
<h1> Título 1 bis</h1>
<p>Párrafo 1</p>
</div>
<h1>Título 2</h1>
```

```
</div>
</body>
</html>
```



Ejemplo 3

```
<!DOCTYPE html>
<html lang="es">
<head>
<title>Html5</title>
<meta charset="UTF-8">
<style type="text/css">
li.uno + li { width: 200px;
              background-color: #9cf;}
</style>
</head>
<body>
<ul>
<li class="uno">Item 1</li>
<li>Item 2</li>
<li>Item 3</li>
<li>Item 4</li>
<li>Item 5</li>
</ul>
</body>
</html>
```



1.2. Pseudoclases de selección

:root	Representa el elemento raíz o el elemento en el nivel más alto del documento. Por ejemplo, en Html5, la etiqueta <html>.
:empty	Corresponde a los elementos vacíos y que no tienen hijos.
:only-child	Devuelve el hijo único. Este elemento no tiene elementos hermanos.
:first-child	Selecciona el primer elemento hijo.
:last-child	Selecciona el último elemento hijo.
:nth-child(n)	Designa el enésimo elemento hijo, donde n es un número o bien las palabras claves even (pares) y odd (impares).
:nth-last-child(n)	Selecciona el enésimo hijo a partir del último elemento.
:only-of-type	Devuelve el único elemento de este tipo.
:first-of-type	Representa el primer elemento de este tipo.
:last-of-type	Representa el último elemento de este tipo.
:nth-of-type(n)	Designa al enésimo elemento de este tipo, donde n es un número o bien las palabras claves even (pares) y odd

	(impares).
<code>:nth-last-of-type (n)</code>	Selecciona el enésimo elemento de este tipo partiendo del último elemento.

Estos selectores están implementados en Firefox 3.6+, Safari 5+, Chrome 7+, Opera 10.6+ e Internet Explorer 9+.

*****La numeración de n comienza en 1 y no en 0 como en JavaScript.**

Ejemplos

Los ejemplos se basan en el documento Html5 siguiente. Comprende una etiqueta padre `` y cinco etiquetas hijas ``.

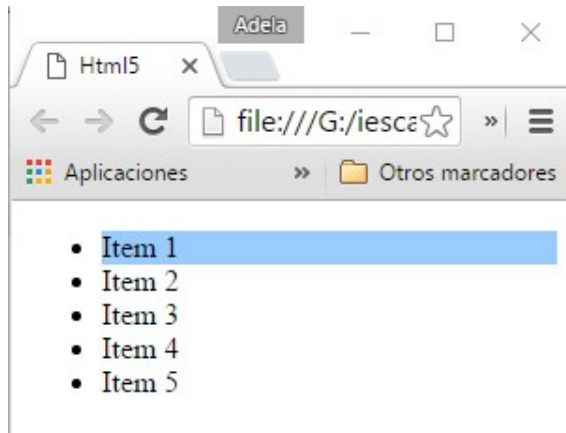
```
<!DOCTYPE html>
<html lang="es">
<head>
<title>Html5</title>
<meta charset="UTF-8">
</head>
<body>
<ul>
<li>Item 1</li>
<li>Item 2</li>
<li>Item 3</li>
<li>Item 4</li>
<li>Item 5</li>
</ul>
</body>
</html>
```

El primer elemento de la lista

```
ul :first-child { background-color: #9cf;}
```

o

```
li:first-of-type { background-color: #9cf;}
```

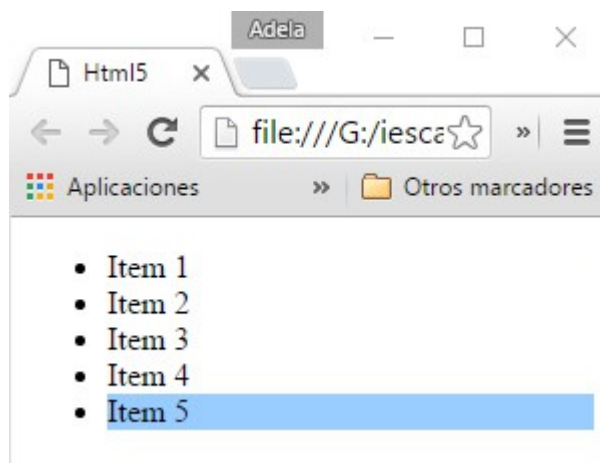


El último elemento de la lista

```
ul :last-child { background-color: #9cf;}
```

o

```
li:last-of-type { background-color: #9cf;}
```

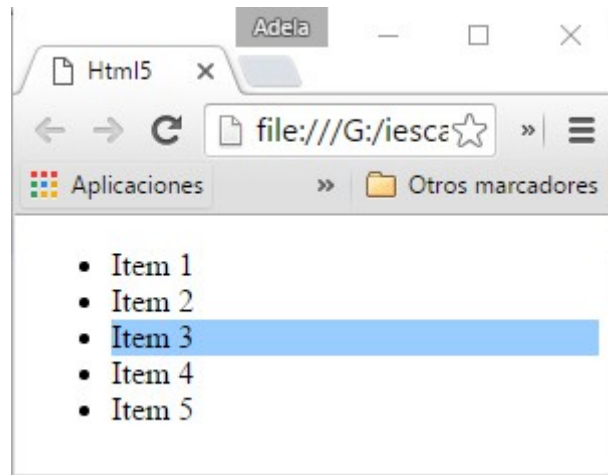


El tercer elemento de la lista

```
ul :nth-child(3) { background-color: #9cf;}
```

o

```
li:nth-of-type(3) { background-color: #9cf;}
```

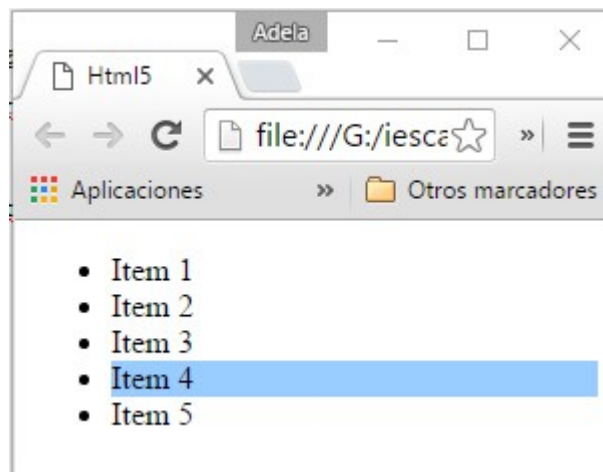


El cuarto elemento de la lista (o el segundo a partir del último elemento)

```
ul :nth-last-child(2) { background-color: #9cf;}
```

o

```
li:nth-last-of-type(2) { background-color: #9cf;}
```

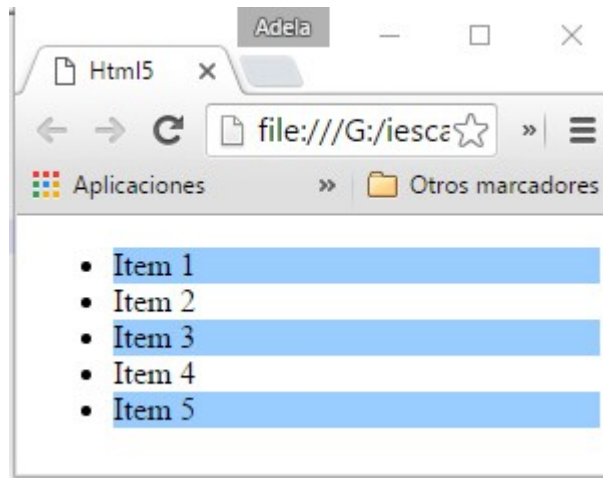


Elementos impares

```
ul :nth-child(odd) { background-color: #9cf;}
```

o

```
li:nth-of-type(odd) { background-color: #9cf;}
```

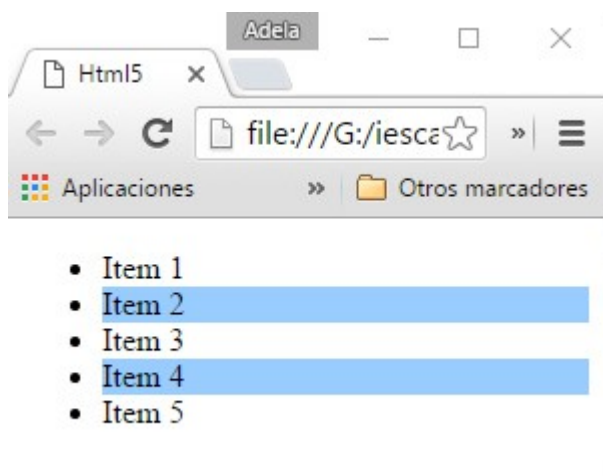



Elementos pares

```
ul :nth-child(even) { background-color: #9cf;}
```

o

```
li:nth-of-type(even) { background-color: #9cf;}
```



1.3. Selectores de atributos

[atr]	Designa un elemento que contiene el atributo atr informado.
[atr="valor"]	Designa un elemento que contiene el atributo atr con el valor indicado.
[atr~="valor"]	Corresponde con todo elemento en donde el atributo atr contenga

	una lista de valores separados por espacios y donde alguno de ellos sea este valor.
[atr "valor"]	Corresponde con todo elemento en donde el atributo atr contenga una lista de valores separados por guiones comenzando por el valor.
[atr^="valor"]	Representa un elemento en donde el atributo atr comience con el prefijo indicado.
[atr\$="valor"]	Representa un elemento en donde el atributo atr termine con el sufijo indicado.
[atr*="valor"]	Representa un elemento en donde el atributo atr contenga una instancia u ocurrencia del valor indicado.

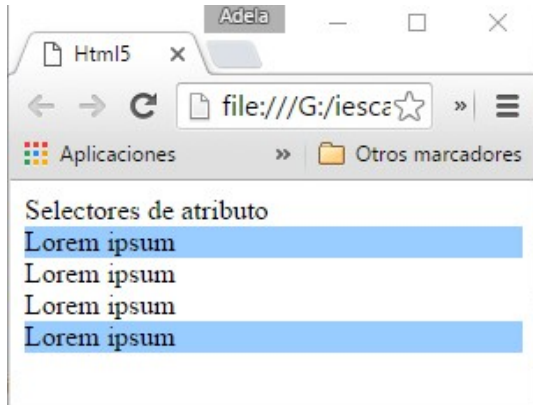
Estos selectores de atributos están implementados en Firefox 3.6+, Safari 5+, Chrome 7+, Opera 10.6+ e Internet Explorer 9+.

Ejemplos

```
<!DOCTYPE html>
<html lang="es">
<head>
<title>Html5</title>
<meta charset="UTF-8">
</head>
<body>
<div class="titulo">Selectores de atributo</div>
<div name="uno">Lorem ipsum</div>
<div title="dos">Lorem ipsum </div>
<div title="tres">Lorem ipsum </div>
<div name="ultimo">Lorem ipsum </div>
</body>
</html>
```

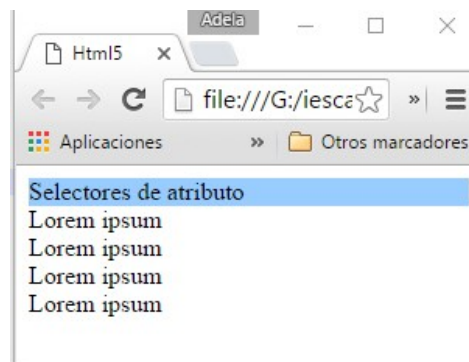
Elemento(s) con el atributo name

```
div[name] { background-color: #9cf;}
```



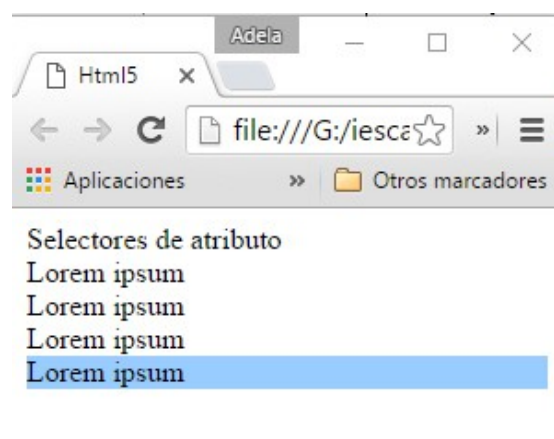
Elemento con el atributo class="titulo"

```
div[class="titulo"] { background-color: #9cf; }
```



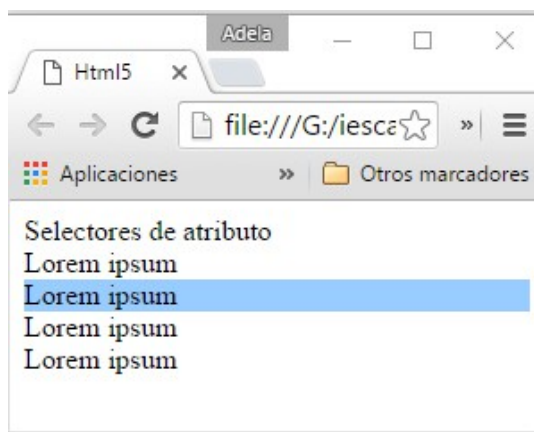
Elemento con un atributo nombrado que comience por "ult"

```
div[name^="ultimo"] { background-color: #9cf; }
```



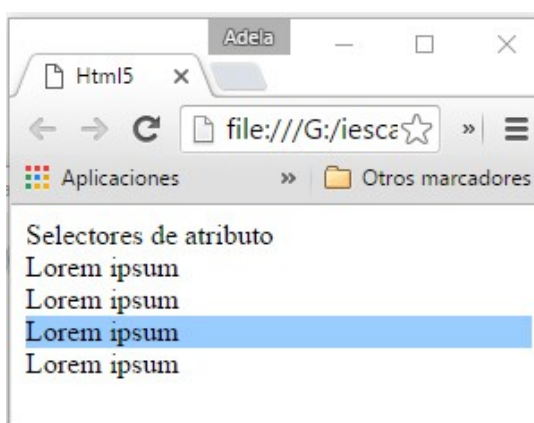
Elemento en que el atributo title termine por "os"

```
div[title$="os"] { background-color: #9cf;}
```



Elemento en que el atributo title contenga "re"

```
div[title*="re"] {background-color: #9cf;}
```



Destaquemos que es posible combinar estos selectores.

```
div[name="uno"][class="titulo"]
```

2. Bordes redondeados

Después de casi 20 años de formas rectangulares con ángulos rectos y puntiagudos, por fin llegan los bordes redondeados.

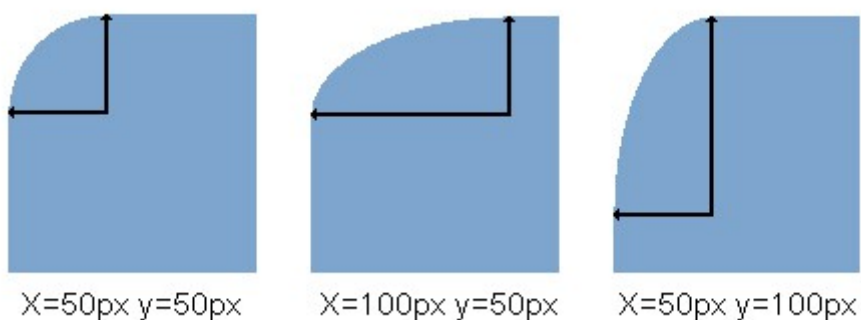
Hasta el día de hoy, los bordes redondeados se conseguían a partir de una serie de trucos y astucias que solían consistir en múltiples divisiones <div> con imágenes de fondo. Con CSS3, los bordes redondeados se diseñan directamente a partir de algunas pocas líneas de código de propiedades de estilo.

Los bordes redondeados en los cuatro extremos se introducen mediante la propiedad:

border-radius	x y; donde x e y son un valor o un porcentaje.
---------------	---

Los valores de x e y determinan los radios horizontal y vertical de un cuarto de elipse, lo cual provocará la curvatura de la esquina.

Los siguientes ejemplos son más descriptivos.



Es posible indicar un único valor. En ese caso, el valor de x será igual al valor de y.

Ejemplo

```
<!DOCTYPE html>
<html lang="es">
<head>
<title>Html5</title>
<meta charset="UTF-8">
<style type="text/css">
#round { width: 200px; height: 80px;
    background-color: rgb(185,205,225);
    border: 1px solid gray;
    border-radius: 1em;}
</style>
</head>
<body>
<div id="round"></div>
</body>
```

</html>



Característica de las hojas de estilo CSS3. Los navegadores a menudo no han escuchado la recomendación oficial de W3C para implementar las propiedades CSS3. Durante un tiempo, se encuentran estas propiedades precedidas por un prefijo: -moz- para Firefox, -ms- para Internet Explorer, -webkit- para Safari y Google Chrome, -o- para Opera. Con el tiempo y la sucesión de versiones, las propiedades CSS3 se reconocen sin prefijo.

```
-moz-border-radius : 1em ;
```

```
-webkit-border-radius : 1em ;
```

La propiedad `border-radius` está soportada por Internet Explorer 9+ sin prefijo.

- ◆ Firefox 3.6+ con el prefijo -moz-, desde Firefox 4.0+ sin prefijo.
- ◆ Chrome 4.0+ con el prefijo -webkit-, desde Chrome 5.0+ sin prefijo.
- ◆ Safari 3.1+ con el prefijo -webkit-, desde Safari 5.0+ sin prefijo.
- ◆ Opera 10.5 sin prefijo.
- ◆ iOS Safari 3.2+ con el prefijo -webkit-, desde la versión 4.0+ sin prefijo.
- ◆ Android Browser 2.1+ con el prefijo -webkit-, desde la 2.2+ sin prefijo.

Una versión perfectamente compatible de nuestro ejemplo sería:

```
<!DOCTYPE html>  
<html lang="es">
```

```
<head>
<title>Html5</title>
<meta charset="UTF-8">
<style type="text/css">
#round { width: 200px; height: 80px;
    background-color: rgb(185,205,225);
    border: 1px solid gray;
    border-radius: 1em;
    -moz-border-radius: 1em;
    -webkit-border-radius: 1em;}
</style>
</head>
<body>
<div id="round"></div>
</body>
</html>
```

Como ocurre a menudo en CSS, es posible especificar de forma independiente un radio para cada esquina.

border-top-right-radius	x y;
border-bottom-right-radius	donde x e y son un valor o un porcentaje
border-bottom-left-radius	border-top-left-radius: 10px 5px;
border-top-left-radius	

También es posible definir el borde redondeado para cada esquina a partir de una escritura abreviada que se lee, como de costumbre, según el sentido de las agujas del reloj comenzando por la parte superior (top, right, bottom, left).

border-radius: lista de valores / lista de valores

La primera lista de valores representa los radios horizontales de las esquinas, la segunda lista representa los radios verticales.

border-radius: 5px 10px 5px 10px / 10px 5px 10px 5px;

La notación anterior con los prefijos -moz y -webkit sería:

```
-moz-border-radius-topright
-moz-border-radius-bottomright
-moz-border-radius-bottomleft
```

```
-moz-border-radius-topleft
```

```
-webkit-border-top-right-radius  
-webkit-border-bottom-right-radius  
-webkit-border-bottom-left-radius  
-webkit-border-top-left-radius
```

Cabe subrayar las diferencias en lo que respecta a la posición de la palabra `radius` y a las sintaxis `top-right` y `topright`.

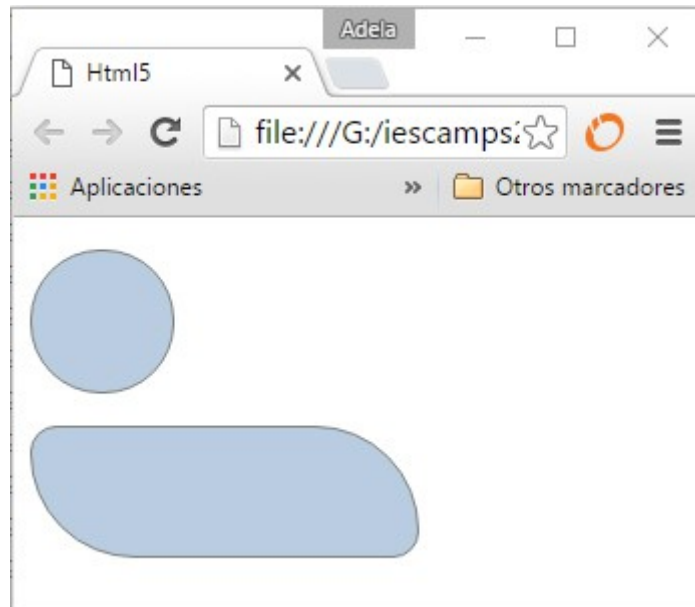
Todo esto permite realizar figuras bastante poco habituales en el diseño de páginas.

Ejemplo 1

```
<!DOCTYPE html>  
<html lang="es">  
<head>  
<title>Html5</title>  
<meta charset="UTF-8">  
<style type="text/css">  
#round1 { width: 70px; height: 70px;  
    background-color: rgb(185,205,225);  
    border: 1px solid gray;  
    border-radius: 35px;  
    -moz-border-radius: 35px;  
    -webkit-border-radius: 35px;}  
#round2 { width: 12em; height: 4em;  
    background-color: rgb(185,205,225);  
    border: 1px solid gray;  
    border-radius: 1em 4em 1em 4em;  
    -moz-border-radius: 1em 4em 1em 4em;  
    -webkit-border-radius: 1em 4em 1em 4em;}  
</style>  
</head>  
<body>  
<p></p>  
<div id="round1"></div>  
<p></p>  
<div id="round2"></div>
```



```
</body>  
</html>
```



Ejemplo 2

Los bordes redondeados no sólo se aplican a las divisiones <div>. He aquí un ejemplo de aplicación en una tabla.

```
<!DOCTYPE html>  
<html lang="es">  
<head>  
<title>Html5</title>  
<meta charset="UTF-8">  
<style type="text/css">  
table { width: 200px;  
    background-color: rgb(185,205,225);  
    text-align: center;  
    margin: 30px;  
    border: 1px solid gray;  
    border-radius: 10px;  
    -moz-border-radius: 10px;  
    -webkit-border-radius: 10px;}  
</style>  
</head>  
<body>  
<table>  
<tr><td>1</td><td>2</td><td>3</td></tr>  
<tr><td>4</td><td>5</td><td>6</td></tr>
```

```
<tr><td>7</td><td>8</td><td>9</td></tr>
</table>
</body>
</html>
```

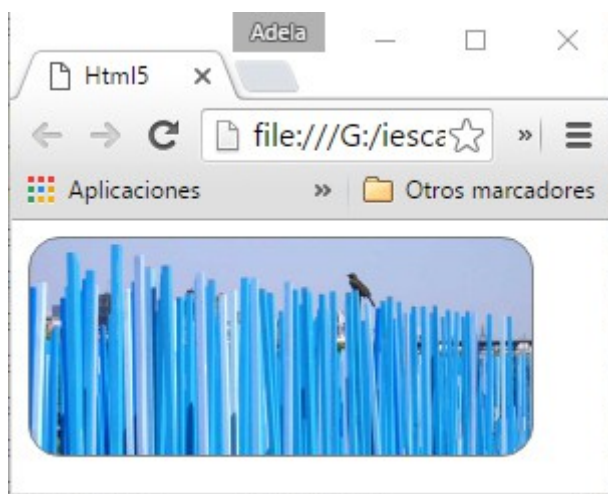


Los bordes redondeados no son posibles si se establece la propiedad collapse.

Ejemplo 3

También es posible aplicar bordes redondos a las imágenes sin tener que utilizar un programa de diseño gráfico.

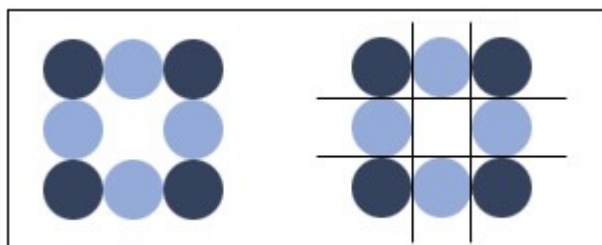
```
<!DOCTYPE html>
<html lang="es">
<head>
<title>Html5</title>
<meta charset="UTF-8">
<style type="text/css">
#round { width: 250px; height: 108px;
border-radius: 1em;
-moz-border-radius: 1em;
-webkit-border-radius: 1em;
border: 1px solid gray;
background-image: url(blue.png);}
</style>
</head>
<body>
<div id="round"></div>
</body>
</html>
```



3. Bordes decorados

Una de las novedades en CSS3 es que permiten utilizar una imagen y aplicarla al borde de un elemento caja.

La propiedad `border-image` toma una imagen rectangular y la divide en 9 partes (ver la captura de pantalla a continuación). Las 8 partes correspondientes al contorno se utilizan para las esquinas y para los lados. El centro se oculta para permitir la visualización del contenido.



La escritura de esta propiedad es bastante compleja.

border-image	<pre>url(archivo_imagen) a b c d valor valor;</pre> <p>donde</p> <p>url(archivo_imagen) es la ruta relativa de la imagen que va a servirnos para confeccionar el borde.</p> <p>a, b, c y d representan la dimensión de las partes de la rendija a los cuatro lados del borde.</p> <p>valor puede valer <code>round</code>, <code>repeat</code> o <code>stretch</code>.</p>
---------------------	---

Por ejemplo:

```
border-image: url(border.png) 27 27 27 27 round round;
```

Comentarios

- ◆ Es posible señalar solamente un parámetro. En este caso, se aplica a los cuatro lados. Con dos parámetros, se determina la dimensión de los lados superior/inferior y derecho/izquierdo. Tres parámetros definen respectivamente los lados superior, derecho/izquierdo e inferior. Por último, con cuatro parámetros se definen, en orden, los bordes superior, derecho, inferior e izquierdo.
- ◆ La palabra clave `round` reproduce las imágenes y las redimensiona para que se ajusten exactamente a la anchura y altura del elemento. La palabra clave `repeat` (no soportada en algunos navegadores) efectúa el mismo trabajo que `round` aunque sin ajustarlas. Y por último, `stretch` (valor por defecto) estira la imagen y la ajusta a las dimensiones de la caja. Un único valor determina los cuatro lados. Dos valores se aplican respectivamente a los lados superior/inferior y derecho/izquierdo.

La propiedad `border-image` está soportada por:

- ◆ Firefox 3.6+ con el prefijo `-moz-`, desde Firefox 15.0+ sin prefijo.
- ◆ Chrome 4.0+ con el prefijo `-webkit-`, desde Chrome 16.0+ sin prefijo.
- ◆ Safari 3.1+ con el prefijo `-webkit-`, desde Safari 6.0+ sin prefijo.
- ◆ Opera 11.0 + con el prefijo `-o-`.
- ◆ Internet Explorer incluso en su versión 10 no reconoce esta propiedad.
- ◆ iOS Safari 3.2+ con el prefijo `-webkit-`.
- ◆ Android Browser 2.1+ con el prefijo `-webkit-`.

Ejemplo 1

Basémonos en la imagen anterior (`border.png`), disponible en el área de descargas.

```
<!DOCTYPE html>
<html lang="es">
<head>
<title>Html5</title>
<meta charset="UTF-8">
<style type="text/css">
```

```
#box { width: 220px; height: 80px;
      border-width: 20px;
      border-image:url(border.png) 30 30 round;
      -moz-border-image:url(border.png) 30 30 round;
      -webkit-border-image:url(border.png) 30 30 round;
      padding: 10px;
      text-align: center;}
</style>
</head>
<body>
<div id="box"><h1>Html5 y CSS3</h1></div>
</body>
</html>
```



Observe que es necesario fijar la anchura del borde (border-width), lo cual parece lógico.

Ejemplo 2

Con el valor de estiramiento `stretch`:

```
<!DOCTYPE html>
<html lang="es">
<head>
<title>Html5</title>
<meta charset="UTF-8">
<style type="text/css">
#box { width: 220px; height: 80px;
      border-width: 20px;
      border-image:url(border.png) 30 30 round stretch;
      -moz-border-image:url(border.png) 30 30 round stretch;
      -webkit-border-image:url(border.png) 30 30 round stretch;
```

```
padding: 10px;
text-align: center;}
</style>
</head>
<body>
<div id="box"><h1>Html5 y CSS3</h1></div>
</body>
</html>
```



Ejemplo 3

Tomemos la imagen de partida (border_image.png):



```
<!DOCTYPE html>
<html lang="es">
<head>
<title>Html5</title>
<meta charset="UTF-8">
<style type="text/css">
#box { width: 220px; height: 80px;
border-width: 20px;
border-image:url(border_image.png) 30 30 round;
```

```
-moz-border-image:url(border_image.png) 30 30 round;
-webkit-border-image:url(border_image.png) 30 30 round;
padding: 10px;
text-align: center;}
</style>
</head>
<body>
<div id="box"><h1>Html5 y CSS3</h1></div>
</body>
</html>
```



4. Sombras

Con las propiedades de estilo CSS3 es posible, a día de hoy, agregar un efecto de sombra sobre el texto o los elementos de tipo caja.

4.1. Sombras sobre el texto

text-shadow:	<p>x y z color donde</p> <ul style="list-style-type: none"> - x es el margen de la sombra hacia la derecha. - y es el margen de la sombra hacia abajo. - z es la intensidad de degradado o de desenfocado (es opcional, con valor por defecto 0). - color es el color de la sombra.
--------------	---

Los parámetros `x` e `y` admiten valores negativos. El margen de la sombra se desplaza en este caso hacia la izquierda y hacia arriba.

La propiedad `text-shadow` está implementada en:

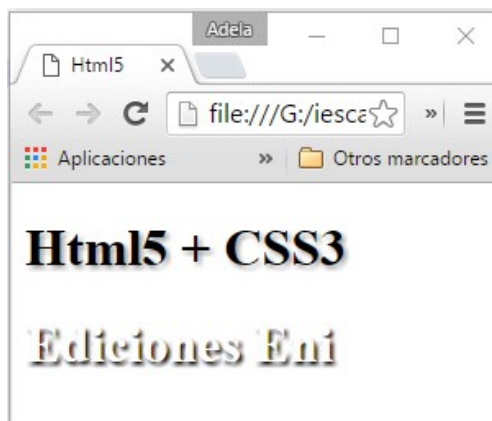
- Internet Explorer 10+.
- ◆ Firefox 3.6+.
- ◆ Chrome 4.0+.
- ◆ Safari 3.1+.
- ◆ Opera 10.0+.
- ◆ iOS Safari 3.2+.
- ◆ Android Browser 2.1+.

Para Internet Explorer 9, es preciso utilizar el filtro propietario de Microsoft Shadow. Para conseguir un efecto de sombra compatible hay que agregar, por ejemplo, el código:

```
.sombreado {  
filter:progid:DXImageTransform.Microsoft.Shadow(color='#999999',  
Direction=135,  
Strength=4);}
```

Ejemplo

```
<!DOCTYPE html>  
<html lang="es">  
<head>  
<title>Html5</title>  
<meta charset="UTF-8">  
<style type="text/css">  
.sombra { text-shadow: 2px 2px 4px #999;  
filter:progid:DXImageTransform.Microsoft.Shadow(color='#aaaaaa',  
Direction=135, Strength=6);}  
.relieve { color : white;  
text-shadow: 2px 2px 3px black;  
filter:progid:DXImageTransform.Microsoft.Shadow(color='#aaaaaa',  
Direction=135, Strength=12);}  
</style>  
</head>  
<body>  
<h1 class="sombra">Html5 + CSS3</h1>  
<h1 class="relieve">Ediciones Eni</h1>  
</body>  
</html>
```

Para los amantes de la precisión, text-shadow ya estaba incluido en la especificación CSS2 (de ahí la ausencia de prefijos). Se retiró de la versión CSS2.1 y se ha vuelto a incluir con la especificación CSS3.

4.2 Sombras sobre un elemento caja

box-shadow	<p>x y z color donde</p> <ul style="list-style-type: none"> - x es el margen de la sombra hacia la derecha. - y es el margen de la sombra hacia abajo. - z es la intensidad de degradado o de desenfocado (es opcional, con valor por defecto 0). - color es el color de la sombra.
------------	---

Los parámetros **x** e **y** admiten valores negativos. El margen de la sombra se desplaza en este caso hacia la izquierda y hacia arriba.

Observe que la propiedad se aplica sobre la caja del elemento y no sobre su borde. El sombreado no afecta, por tanto, al tamaño de la caja del elemento.

La propiedad `box-shadow` está implementada por:

- ◆ Internet Explore 10 sin prefijo.
- ◆ Firefox 3.6+ con el prefijo -moz-, desde Firefox 4.0+ sin prefijo.
- ◆ Chrome 4.0+ con el prefijo -webkit-, desde Chrome 10.0+ sin prefijo.
- ◆ Safari 3.1+ con el prefijo -webkit-, desde Safari 5.1+ sin prefijo.
- ◆ Opera 10.5 sin prefijo.
- ◆ iOS Safari 3.2+ con el prefijo -webkit-, desde la versión 5.0+ sin prefijo.

- ◆ Android Browser 2.1+ con el prefijo -webkit-, desde la versión 4.0+ sin prefijo.

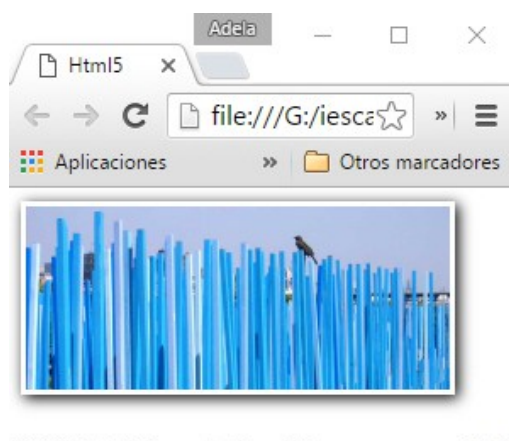
Para Internet Explorer 9, es preciso utilizar el filtro propietario de Microsoft Shadow. Para conseguir un efecto de sombra compatible hay que agregar, por ejemplo, el código:

```
.sombreado
{ filter:progid:DXImageTransform.Microsoft.Shadow(color='#aaaaaa',
Direction=135, Strength=12);}
```

Ejemplo

```
<!DOCTYPE html>
<html lang="es">
<head>
<title>Html5</title>
<meta charset="UTF-8">
<style type="text/css">
.sombra { border: 3px solid white;
          box-shadow: 2px 2px 12px black;
          -moz-box-shadow: 2px 2px 12px black;
          -webkit-box-shadow: 2px 2px 12px black;
          filter:progid:DXImageTransform.Microsoft.Shadow(color='#aaaaaa',
          Direction=135, Strength=12);}
</style>
</head>
<body>

</body>
</html>
```



El efecto de sombreado se habría podido aplicar directamente sobre la imagen.

5. Tipos de letra personalizados

Hasta el presente las páginas Web no eran gran cosa en lo que respecta a tipografía. En efecto, como los tipos de letra dependían de las tipografías instaladas en el ordenador del internauta, había pocos tipos de letra disponibles al mismo tiempo en los puestos de usuario Windows o Mac.

Las hojas de estilo CSS3 permiten importar y utilizar una tipografía personalizada incluso si no está instalada en el puesto del usuario. Esta propiedad `@font-face` constituye una innovación significativa para los diseñadores Web que podrán incluir, de este modo, tipografías más creativas.

La adopción de un tipo de letra particular se realiza en dos fases.

Primero hay que descargar el tipo de letra para, a continuación, incorporarlo en la página.

```
@font-face { font-family: "nombre_tipografia";  
             url("nombre_tipografia.extension_tipografia"); }
```

Comentarios

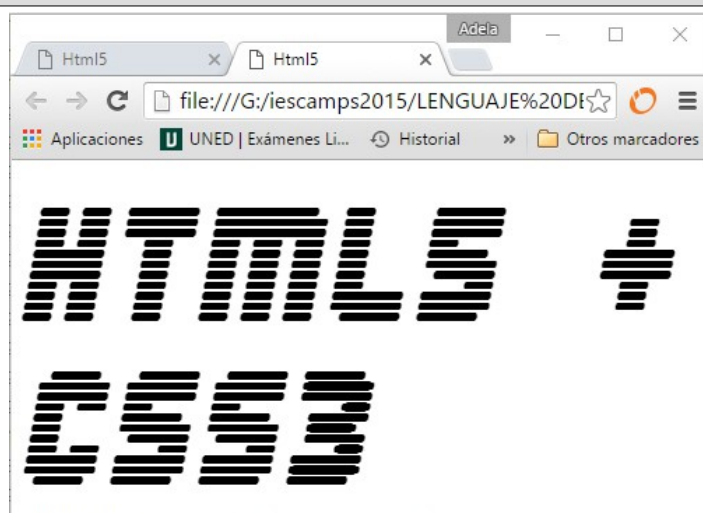
- ◆ `font-family: "nombre_tipografia"` indica con qué nombre se va a referenciar al tipo de letra en la página.
- ◆ `url("nombre_tipografia.extension_tipografia")` indica la url, mediante direccionamiento relativo o absoluto, del tipo de letra personalizado. Usando direccionamiento relativo, el tipo de letra estará situado en la carpeta del sitio Web igual que lo hace una imagen o un archivo de hoja de estilos externo. Se mantendrá la preferencia por tipografías adaptadas a la Web (**webfonts**) donde el tamaño reducido no penalice el tiempo de descarga.
- ◆ **extension_tipografia** determina la extensión del tipo de letra, por ejemplo la extensión ttf (True Type Fonts).

Una vez hemos descargado el tipo de letra mediante `@font-face`, es posible aplicarla a algún elemento, como con cualquier otro tipo de letra, mediante la propiedad `font-family`.

Ejemplo

```
<!DOCTYPE html>  
<html lang="es">  
<head>
```

```
<title>Html5</title>
<meta charset="UTF-8">
<style type="text/css">
@font-face { font-family: "mitipografia";
             src: url('fantasia.ttf');}
.fontface { font: 100px mitipografia, fantasy;}
</style>
</head>
<body>
<div class="fontface">Html5 + CSS3</div>
</body>
</html>
```



Los archivos tienen que estar en línea o en un servidor local.

¡Pero no todo es tan simple!

En primer lugar, existen diferentes formatos de fuentes adaptadas a la Web (Web fonts):

Formato .eot	El formato eot (Embedded OpenType) es un formato compacto de fuente OpenType. Este formato es propietario de Microsoft.
Formato .ttf	El formato ttf (True Type Font) es el formato de fuente de caracteres más extendido.
Formato .otf	El formato otf (OpenType Font) es una evolución del formato ttf en el que se han enriquecido las posibilidades tipográficas.
Formato .woff	El formato woff (Web Open Font Format) es un formato de fuente de caracteres comprimido para uso en la Web. Este nuevo formato, desarrollado en 2009, ha sido adoptado rápidamente por diferentes

navegadores.

La propiedad @font-face y los diversos formatos se reconocen por:

- ◆ Internet Explorer 9+, pero solo el formato eot.
- ◆ Firefox 3.6+ para los formatos ttf, otf y woff.
- ◆ Chrome 4.0+ para los formatos ttf y otf. Chrome 5.0+ para el formato woff.
- ◆ Safari 3.1+ para los formatos ttf y otf. Safari 5.1+ para el formato woff.
- ◆ Opera 10.0+ para los formatos ttf y otf. Opera 11.1+ para el formato woff.
- ◆ iOS Safari 4.2+ para los formatos ttf y otf. iOS Safari 5.0+ para el formato woff.
- ◆ Android Browser 2.2+ solo para los formatos ttf y otf.

Una propiedad de estilo compatible podría ser:

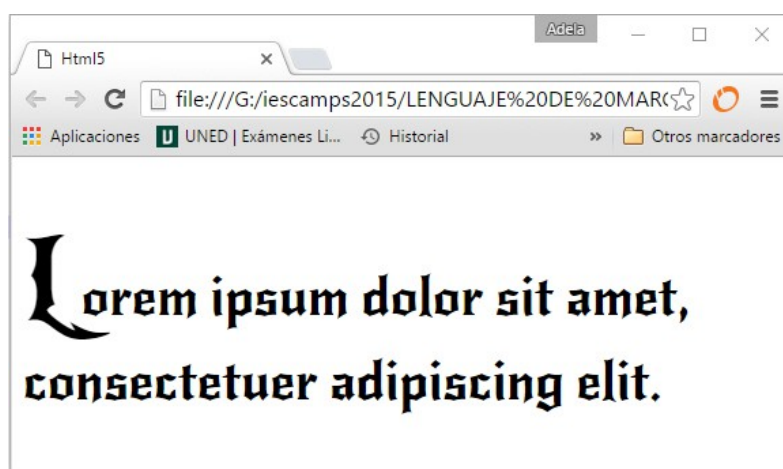
```
@font-face{ font-family: 'mitipografia';  
            src: url('rocker.eot') format('embedded-opentype'),  
            url('rocker.woff') format('woff'),  
            url('rocker.ttf') format('truetype');}
```

Esto sin contar con que Internet Explorer 9 tiene ciertas dificultades con el atributo src cuando se hace referencia a un tipo de letra eot y después a un tipo de letra ttf, o woff. Será preciso utilizar la siguiente astucia para conseguir una hoja de estilos compatible con todos los navegadores de nuestro estudio.

```
<!DOCTYPE html>  
<html lang="es">  
<head>  
<title>Html5</title>  
<meta charset="UTF-8">  
<style type="text/css">  
@font-face{  
    font-family: 'rocker';  
    src: url('rocker.eot');  
    src: url('rocker.eot?#iefix') format('embedded-opentype'),  
        url('rocker.woff') format('woff'),  
        url('rocker.ttf') format('truetype');}  
.police { font: 40px 'rocker', fantasy;}
```

```
p:first-letter { font-size:200%;}  
</style>  
</head>  
<body>  
<p class="police">  
Lorem ipsum dolor sit amet, consectetur adipiscing elit.  
</p>  
</body>  
</html>
```

El problema se ha resuelto en Internet Explorer 10.



Para obtener una fuente de caracteres en los diferentes formatos existen, afortunadamente, sitios dedicados a esta función. Por ejemplo : www.font2web.com o www.fontsquirrel.com/fontface/generator.

Todo esto ilustra que cuando W3C no emite ninguna norma (en este caso para el formato de fuente), los navegadores siempre complican las cosas.

Existe otra forma de utilizar tipos de letra particulares en una página Web sin tener que recurrir a `@font-face` y que es perfectamente compatible con los distintos navegadores. Consiste en utilizar **Google Font Api**. Las tipografías proporcionadas por este servicio están perfectamente adaptadas a la visualización en un navegador y tienen un peso reducido.

Basta con agregar una hoja de estilo a su página Html para obtener la nueva tipografía.

```
<link  
href='http://fonts.googleapis.com/css?  
family=Reenie+Beanie&subset=latin'
```

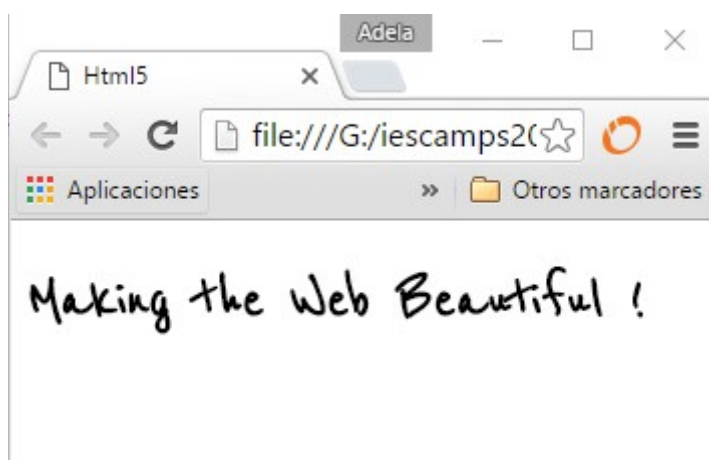
```
rel='stylesheet' type='text/css'>
```

A continuación, basta con aplicar la tipografía en una clase CSS mediante la propiedad `font-family`.

No es preciso, entonces, pasar por la propiedad `@font-face`, ni tener que ocuparse del formato del tipo de letra. Todo esto lo gestiona Google para asegurar la mejor visualización en cada navegador.

Ejemplo

```
<!DOCTYPE html>
<html lang="es">
<head>
<title>Html5</title>
<meta charset="UTF-8">
<link
href='http://fonts.googleapis.com/css?family=Reenie+Beanie&subset=
latin' rel='stylesheet' type='text/css'>
<style>
h1 { font-family: 'Reenie Beanie', arial, serif; }
</style>
<body>
<h1>Making the Web Beautiful !</h1>
</body>
</html>
```



Para encontrar tipos de letra adaptados a la Web, es posible realizar una búsqueda en Google con la palabra clave `webfont` y obtendremos la dirección de numerosos sitios

especializados.

Citemos, no obstante, los siguientes:

- ◆ www.fontsquirrel.com/
- ◆ kernest.com/
- ◆ www.webfontlist.com/
- ◆ www.smashingmagazine.com/2007/11/08/40-excellent-freefonts-for-professional-design/

No olvide que puede existir un copyright sobre la tipografía y que, algunas de ellas, pueden requerir un pago.

Queda la cuestión de cómo convertir tipos de letra ttf en formato eot y otros formatos de tipografía.

6. Las letras huecas

Durante mucho tiempo la tipografía ha sido el pariente pobre de las páginas Web. Con la propiedad CSS3 `text-stroke`, ya puede mostrar letras huecas.

Esta propiedad está disponible con el prefijo `-webkit-` en Google Chrome 4.0, Safari 3.1+, iOS Safari 4.0+ y Android Browser 2.1+.

<code>text-stroke-width: valor;</code>	Grosor del borde de las letras (en pixels).
<code>text-stroke-color: color;</code>	Color del borde de las letras.
<code>text-stroke: valor color;</code>	Contracción de las dos propiedades anteriores.
<code>text-fill-color: color;</code>	Color de rellenos de las letras (Se permite el valor <code>transparent</code>).

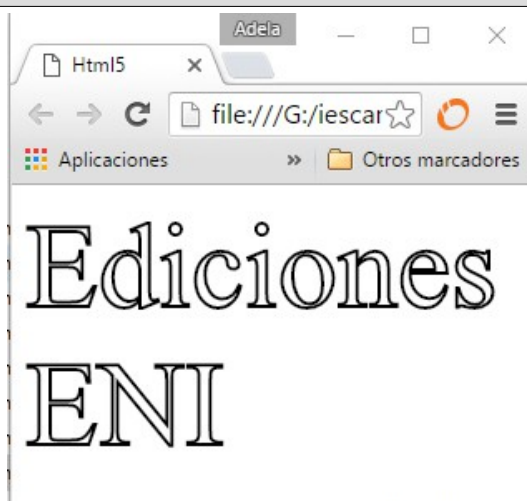
```
h1 { -webkit-text-stroke: 1px black;  
      -webkit-text-fill-color: transparent;}
```

Es posible visualizar los diferentes parámetros en la siguiente url:

www.westciv.com/tools/textStroke/

Ejemplo


```
<!DOCTYPE html>
<html lang="es">
<head>
<title>Html5</title>
<meta charset="UTF-8">
<style type="text/css">
.stroke { font-size: 72px;
        -webkit-text-stroke: 2px #000;
        -webkit-text-fill-color: transparent;}
</style>
</head>
<body>
<div class="stroke">Ediciones ENI</div>
</body>
</html>
```



Algunos, sin embargo, pueden lamentar la falta de compatibilidad de este efecto. Por otra parte, es posible utilizar un truco para reproducir este efecto de letras huecas en Firefox, Opera, Internet Explorer 10 y, por supuesto, Chrome así como Safari. Consiste en utilizar la propiedad `text-shadow` con un tamaño de 1 px sin propagación (blur) para las cuatro posiciones, arriba a la derecha, arriba a la izquierda, abajo a izquierda y abajo a derecha.

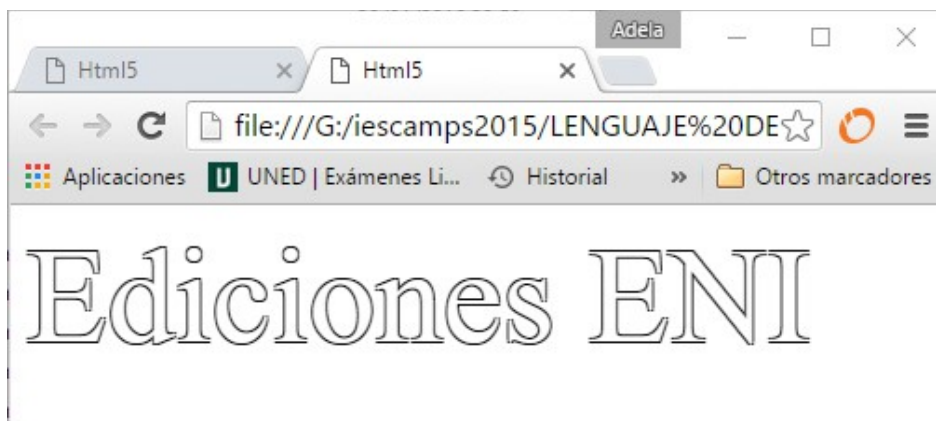
La propiedad `color` sustituye a `text-fill-color`.

```
h1 { color: white;
      text-shadow:
        -1px -1px 0 #000,
        1px -1px 0 #000,
        -1px 1px 0 #000,
```

```
1px 1px 0 #000;}
```

Ejemplo

```
<!DOCTYPE html>
<html lang="es">
<head>
<title>Html5</title>
<meta charset="UTF-8">
<style type="text/css">
.compatible { font-size: 72px;
               color: white;
               text-shadow:
                 -1px -1px 0 #000,
                 1px -1px 0 #000,
                 -1px 1px 0 #000,
                 1px 1px 0 #000;}
</style>
</head>
<body>
<div class="compatible">Ediciones ENI</div>
</body>
</html>
```



7. Manejar títulos largos

La propiedad CSS `text-overflow` es una propiedad muy útil para ciertos contenidos recortados voluntariamente, como un resumen de artículo o comentarios de visitantes.

<code>text-overflow</code>	<code>clip;</code> <code>ellipsis;</code> <code>string;</code>
----------------------------	--

Comentarios

- ◆ El valor `clip` corta (a veces bruscamente) el texto excedente.
- ◆ El valor `ellipsis` termina el texto cortado con puntos suspensivos.
- ◆ El valor `string` termina el texto con una cadena de caracteres (string), por ejemplo ", etc.". Sólo funciona en Firefox.

La propiedad `text-overflow` no corta nada, solo sirve para indicar qué hacer (por ejemplo insertar unos puntos suspensivos) en un texto que ya se ha cortado por `overflow: hidden`.

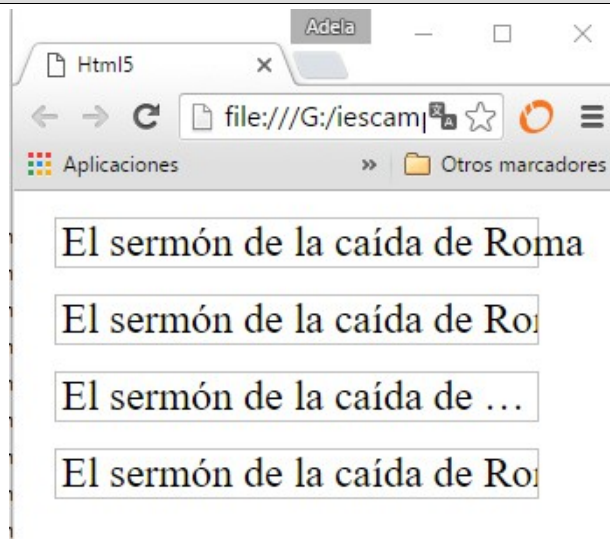
Internet Explorer 6.0+, Firefox 7.0+, Chrome 4.0+, Safari 3.1+, Opera 11+, iOS Safari 3.2+ y Android Browser 2.1+ reconocen esta propiedad sin prefijo alguno.

Ejemplo

```
<!doctype html>
<html>
<head>
<title>Html5</title>
<meta charset="utf-8">
```

```
<style type="text/css">
div { margin: 15px;
      font-size: 150%;
      width: 272px;
      padding-left: 3px;
      border: 1px solid silver;}
.nada { white-space: nowrap;}
.clip { white-space: nowrap;
        overflow: hidden;
        text-overflow: clip;}
.ellipsis { white-space: nowrap;
            overflow: hidden;
            text-overflow: ellipsis;}
.string { white-space: nowrap;
          overflow: hidden;
          text-overflow: ", etc.";}
</style>
</head>
<body>
<div class="nada">El sermón de la caída de Roma</div>
<div class="clip"> El sermón de la caída de Roma </div>
<div class="ellipsis"> El sermón de la caída de Roma </div>
<div class="string"> El sermón de la caída de Roma </div>
</body>
</html>
```

8. Los guiones



automáticos

La propiedad CSS `hyphens` indica al navegador el método que tiene que utilizar para separar sílabas con el fin de mejorar la presentación del texto.

<code>hyphens</code>	<code>none;</code>
----------------------	--------------------

```
manual;  
  
auto;
```

Comentarios

- ◆ **none**: las palabras no se cortan, incluso si un carácter en el texto sugiere posibles puntos de corte.
- ◆ **manual**: las palabras se cortan solo a nivel de los caracteres que sugieren la posibilidad de un corte.
- ◆ **auto**: el navegador es libre de realizar los cortes en los sitios que considere apropiados, siguiendo la regla que haya escogido. Esta regla puede variar de un navegador a otro.

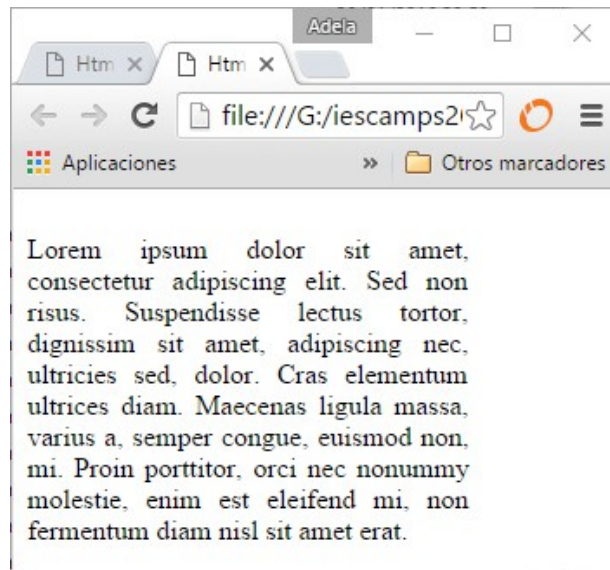
De momento, solo reconocen esta propiedad Internet Explorer 10+ con el prefijo -ms-, Firefox 6.0+ con el prefijo -moz-, Safari 5.1+ con el prefijo -webkit- e iOS Safari 4.2+ con el prefijo -webkit-. Aunque parezca raro, ni Google Chrome, ni Android Browser ni Opera la reconocen.

Ejemplo

```
<!doctype html>  
<html>  
<head>  
<title>Html5</title>  
<meta charset="utf-8">  
<style type="text/css">  
.bloc { width: 255px;  
    text-align: justify;  
    -webkit-hyphens: auto;  
    -moz-hyphens: auto;  
    -ms-hyphens: auto;  
    hyphens: auto;}  
</style>  
</head>  
<body>  
<div class="bloc">  
<br>  
Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed non  
risus. Suspendisse lectus tortor, dignissim sit amet, adipiscing  
nec, ultricies sed, dolor. Cras elementum ultrices diam. Maecenas
```

*ligula massa, varius a, semper congue, euismod non, mi. Proin
porttitor, orci nec nonummy molestie, enim est eleifend mi, non
fermentum diam nisl sit amet erat.*

```
</div>  
</body>  
</html>
```



9. Redimensionar un elemento

La propiedad CSS3 `resize` permite redimensionar un elemento.

<code>resize</code>	<code>none;</code> <code>both;</code> <code>horizontal;</code> <code>vertical;</code>
---------------------	--

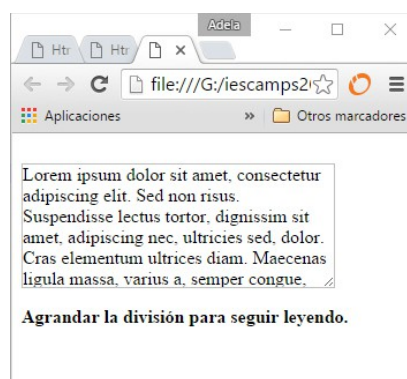
Comentarios

- ◆ `none`: el usuario no puede redimensionar el elemento.
- ◆ `both`: el usuario puede modificar la altura y la anchura del elemento.
- ◆ `horizontal`: el usuario puede modificar la anchura del elemento.
- ◆ `vertical`: el usuario puede modificar la altura del elemento.

Firefox 4.0+, Chrome 4.0+, Safari 4.0+, y Opera 12.1+, reconocen esta propiedad. Internet Explorer no la reconoce ni en su versión 10. Tampoco iOS Safari ni Android Browser la reconocen.

Ejemplo

```
<!doctype html>
<html>
<head>
<title>Html5</title>
<meta charset="utf-8">
<style type="text/css">
.bloque { resize:vertical;
        overflow:hidden;
        width:270px; height:60px;
        border:1px solid silver;}
</style>
</head>
<body>
<br>
<div class="bloque">
Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed non
    risus. Suspendisse lectus tortor, dignissim sit amet, adipiscing
    nec, ultricies sed, dolor. Cras elementum ultrices diam. Maecenas
    ligula massa, varius a, semper congue, euismod non, mi. Proin
    porttitor, orci nec nonummy molestie, enim est eleifend mi, non
    fermentum diam nisl sit amet erat.
</div>
<p><b>Agrandar la división para seguir leyendo.<b></P>
</body>
</html>
```



10. Ajustar la altura de los caracteres

Tipos de letra distintos, a pesar de tener el mismo tamaño de caracteres, pueden parecer más grandes o más pequeños que los tipos de letra vecinos.

Además, con la propiedad `font-family`, no tenemos el control del tipo de letra que realmente se muestra. En efecto, cuando un tipo de letra no existe en el ordenador del usuario, los navegadores utilizan otro tipo de letra en su lugar, y que no tiene por qué tener las mismas características dimensionales que la tipografía original, incluso si el tamaño de los caracteres es idéntico.

La propiedad `font-size-adjust` permite obtener caracteres siempre del mismo tamaño, tanto para las mayúsculas como para las minúsculas, fijando un coeficiente de aspecto. La propiedad `font-size-adjust` debe definirse según el valor del coeficiente de aspecto del tipo de letra de preferencia.

```
font-size-adjust: 0.465;
```

Este coeficiente de aspecto puede definirse por tanteo, aunque es más sencillo consultar algún sitio Web, como por ejemplo www.brunildo.org/test/aspect-table.html, dedicado al efecto.

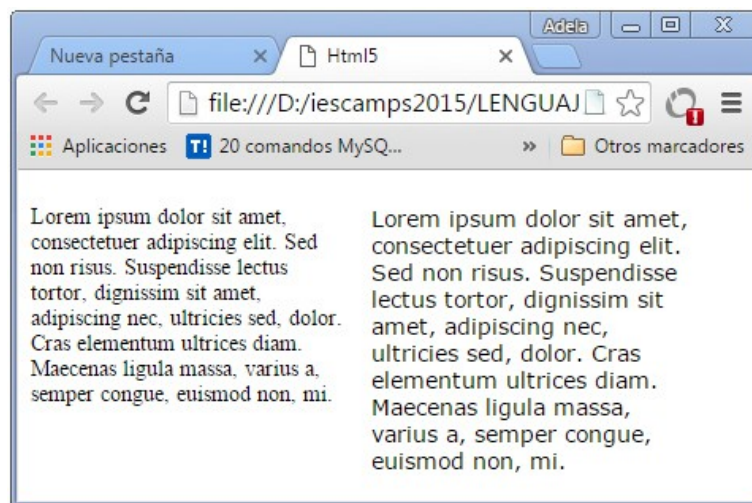
De momento, solamente Firefox 3.6+ ha implementado la propiedad `font-size-adjust` que, sin embargo, es muy útil.

Ejemplo sin font-size-adjust

```
<!DOCTYPE html>
<html lang="es">
<head>
<title>Html5</title>
<meta charset="UTF-8">
<style type="text/css">
div { width: 200px;
    float: left;
    margin-right: 15px;}
#calibri p { font: 14px Calibri sans-serif; }
#verdana p { font: 14px Verdana, sans-serif;}
</style>
</head>
<body>
<div id="calibri">
<p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed
non risus. Suspendisse lectus tortor, dignissim sit amet,
adipiscing nec, ultricies sed, dolor. Cras elementum ultrices
```



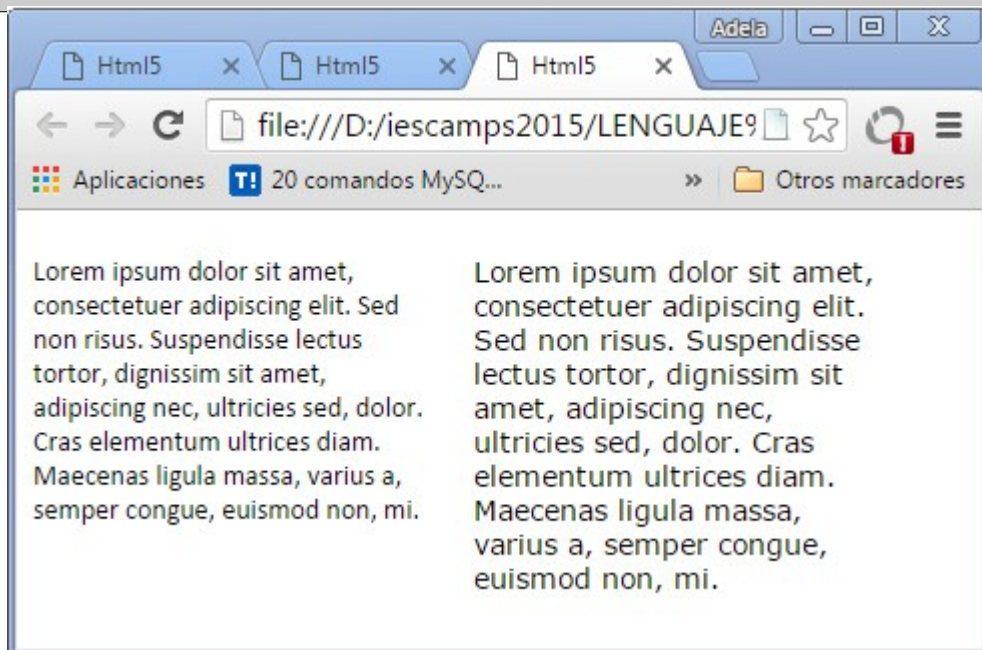
```
diam. Maecenas ligula massa, varius a, semper congue, euismod non,
mi. </p>
</div>
<div id="verdana">
<p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed
non risus. Suspendisse lectus tortor, dignissim sit amet,
adipiscing nec, ultricies sed, dolor. Cras elementum ultrices
diam. Maecenas ligula massa, varius a, semper congue, euismod non, mi. </p>
</div>
</body>
</html>
```



Ejemplo con font-size-adjust

```
<!DOCTYPE html>
<html lang="es">
<head>
<title>Html5</title>
<meta charset="UTF-8">
<style type="text/css">
div { width: 200px;
float: left;
margin-right: 20px;}
#calibri p { font: 14px Calibri, sans-serif;}
#verdana p { font: 14px Verdana, sans-serif;
font-size-adjust: 0.465;}
```

```
</style>
</head>
<body>
<div id="calibri">
<p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed
non risus. Suspendisse lectus tortor, dignissim sit amet,
adipiscing nec, ultricies sed, dolor. Cras elementum ultrices
diam. Maecenas ligula massa, varius a, semper congue, euismod non,
mi. </p>
</div>
<div id="verdana">
<p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed
non risus. Suspendisse lectus tortor, dignissim sit amet,
adipiscing nec, ultricies sed, dolor. Cras elementum ultrices
diam. Maecenas ligula massa, varius a, semper congue, euismod non,
mi. </p>
</div>
</body>
</html>
```



Para obtener una información más detallada acerca de la propiedad `font-size-adjust`, consulte la página <http://webdesignernotebook.com/css/the-little-known-font-size-adjust-css3-property/>.

11. Fondos múltiples

Las hojas de estilo CSS3 hacen posible visualizar varias imágenes en un mismo plano de fondo, y permiten acumular valores en las propiedades `background-image`, `background-position` y `background-repeat` (Parte CSS3 - Fondos).

Este efecto no era posible sino superponiendo divisiones **<div>** definidas con posición absoluta, un poco a la manera de los calcos en las aplicaciones gráficas.

Todos los navegadores de nuestro estudio implementan los fondos múltiples: Internet Explorer 9, Firefox 3.6+, Opera 10.1+, Safari 4+ y Chrome 4+, iOS Safari 3.2+ y Android Browser 2.1+.

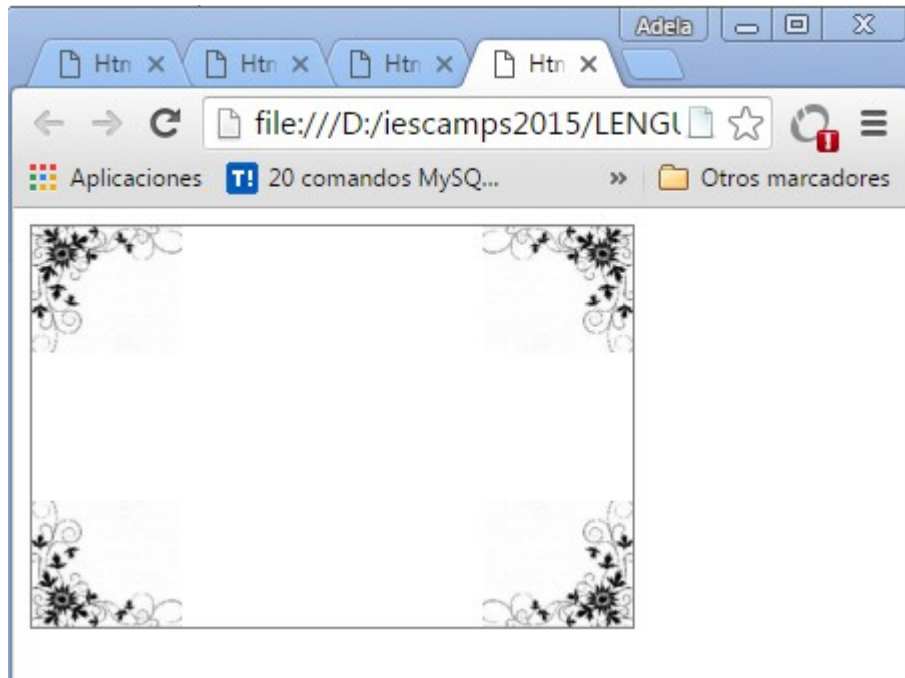
Ejemplo

Tomemos las imágenes `flores1.png`, `flores2.png`, `flores3.png` y `flores4.png`:



```
<!DOCTYPE html>
<html lang="es">
<head>
<title>Html5</title>
<meta charset="UTF-8">
<style type="text/css">
#box { width: 300px; height: 200px;
      border: 1px solid gray;
      background: url(flores1.png) top left no-repeat,
                  url(flores2.png) top right no-repeat,
                  url(flores3.png) bottom right no-repeat,
                  url(flores4.png) bottom left no-repeat;}
</style>
</head>
<body>
```

```
<div id="box"></div>
</body>
</html>
```



12. Degradado de color

12.1. Los degradados lineales

Para Webkit (versiones antiguas)

Los degradados lineales, para los navegadores que utilicen el prefijo **-webkit-**, se definen según la propiedad:

<code>background: -webkit-gradient</code>	<code>linear;</code>
	<code>origen;</code>
	<code>fin;</code>
	<code>from(color) ;</code>
	<code>to(color)) ;</code>

Comentarios

- `linear` indica un degradado lineal.
- `origen` es el punto inicial del degradado. Está determinado por dos valores separados por un espacio. La sintaxis soporta números, porcentajes o las palabras clave `top`, `right`, `bottom` y `left`. Por ejemplo, `top left`.
- `fin` es el punto final del degradado. Está determinado por dos valores separados por un espacio. La sintaxis soporta números, porcentajes o las palabras clave `top`, `right`, `bottom` y `left`. Por ejemplo, `bottom right`.
- `from(color)`, el color de inicio del degradado.
- `to(color)`, el color de fin del degradado.

Ejemplo

```
background: webkit-gradient(linear, left top, left bottom,  
from(#555), to(#383838));
```

Desde 2011, CSS Working Group ha creado un proyecto (draft) reclamando más simplicidad en la sintaxis de la propiedad `gradient`.

Webkit ha introducido una nueva sintaxis para alinearse con otros navegadores.

Para el resto de navegadores

<code>background: linear-gradient</code>	Punto de inicio; color de inicio; color de fin;
--	---

Comentario

El valor punto de inicio se determina con `top` (por defecto), `left`, `right` o `bottom`, por ejemplo `top`.

Esta propiedad se utiliza con los prefijos `-webkit-`, `-moz-` o `-ms-`.

Ejemplo

background: -webkit-linear-gradient(top, #555, #383838) ;

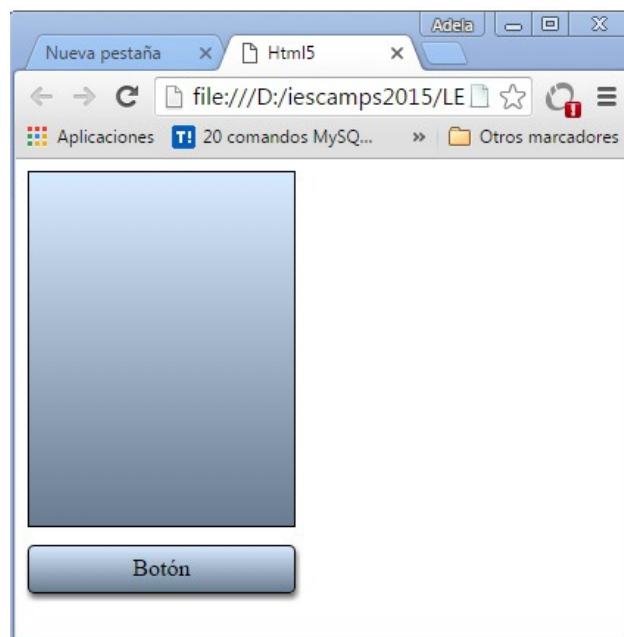
Esta propiedad se reconoce en:

- Internet Explorer 10 con el prefijo -ms-.
- Firefox 3.6+ con el prefijo -moz-, Firefox 16+ sin prefijo.
- Chrome 10+ con el prefijo -webkit-.
- Safari 4+ con el prefijo -webkit-.
- Opera 11.1+ con el prefijo -o-, Opera 12.1+ sin prefijo.
- iOS Safari 3.2+ con el prefijo -webkit-.
- Android Browser 2.1+ con el prefijo -webkit-.

Ejemplo final

```
<!doctype html>
<html>
<head>
<title>Html5</title>
<meta charset="utf-8">
<style type="text/css">
.gradient1 {
width: 180px; height: 240px;
border: 1px solid black;
background:-webkit-gradient(linear,left top,left bottom,
from(#D7EBFF), to(#697D91));
background:-webkit-linear-gradient(top, #D7EBFF, #697D91);
background:-moz-linear-gradient(top, #D7EBFF, #697D91);
background:-ms-linear-gradient(top, #D7EBFF, #697D91);
background:-o-linear-gradient(top, #D7EBFF, #697D91);
background:linear-gradient(top, #D7EBFF, #697D91)}
.gradient2 {
display: block;
height: 25px;
width: 180px;
padding-top: 6px;
margin-bottom: 12px;
border: 1px solid black;
margin-top: 12px;
text-align: center;
border-radius: 4px;
-webkit-border-radius: 4px;
```

```
-moz-border-radius: 4px;
box-shadow: 1px 3px 5px rgba(0,0,0,0.6);
-moz-box-shadow: 1px 3px 5px rgba(0,0,0,0.6);
-webkit-box-shadow: 1px 3px 5px rgba(0,0,0,0.6);
background:-webkit-gradient(linear,left top,left bottom, from(#D7EBFF), to(#697D91));
background:-webkit-linear-gradient(top, #D7EBFF, #697D91);
background:-moz-linear-gradient(top, #D7EBFF, #697D91);
background:-ms-linear-gradient(top, #D7EBFF, #697D91);
background:-o-linear-gradient(top, #D7EBFF, #697D91);
background:linear-gradient(top, #D7EBFF, #697D91)}
</style>
</head>
<body>
<div class="gradient1"></div>
<div class="gradient2">Botón</div>
</body>
</html>
```



Internet Explorer 8 y 9 no implementan la propiedad de estilo gradient. Se basa en un filtro propietario de Microsoft.

```
filter:
progid:DXImageTransform.Microsoft.Gradient(StartColorStr=' #00FF00'
, EndColorStr=' #000000', GradientType=1);
```

Con **StartColorStr** para el color de inicio, **EndColorStr** para el color de fin y

gradientType con valor 0 para degradados verticales y 1 para degradados horizontales.

Los degradados también se pueden aplicar a las propiedades CSS que soportan una imagen como background-image, border-image, list-style-image o content.

La web ofrece generadores de degradados para visualizarlos y obtener el código fuente.

Por ejemplo:

<http://www.colozilla.com/gradient-editor>. ¡Una referencia en la materia!

<http://gradients.glrzad.com/>

12.2. Los degradados circulares

Para los degradados circulares:

background: radial-gradient	posición de inicio; forma; color de inicio; color de fin;
-----------------------------	--

Comentarios

- `Radial-gradient` para los degradados circulares.
- `Posición de inicio` es el punto central del degradado circular. Palabra clave a elegir entre `top`, `right`, `bottom`, `left`, `center` o un par de valores (separados por un espacio) elaborado a partir de `top`, `right`, `bottom` o `left`. Por ejemplo, `center`. Este punto de inicio también se puede expresar en %.
- `forma`: la forma del degradado radial, `circle` (circular) o `ellipse` (elíptica).
- `color de inicio`, `color de fin`: el color de inicio del degradado y el de fin.

Esta propiedad se reconoce en:

- Internet Explorer 10 con el prefijo `-ms-`.
- Firefox 3.6+ con el prefijo `-moz-`, Firefox 16+ sin prefijo.
- Chrome 10+ con el prefijo `-webkit-`.
- Safari 4+ con el prefijo `-webkit-`.

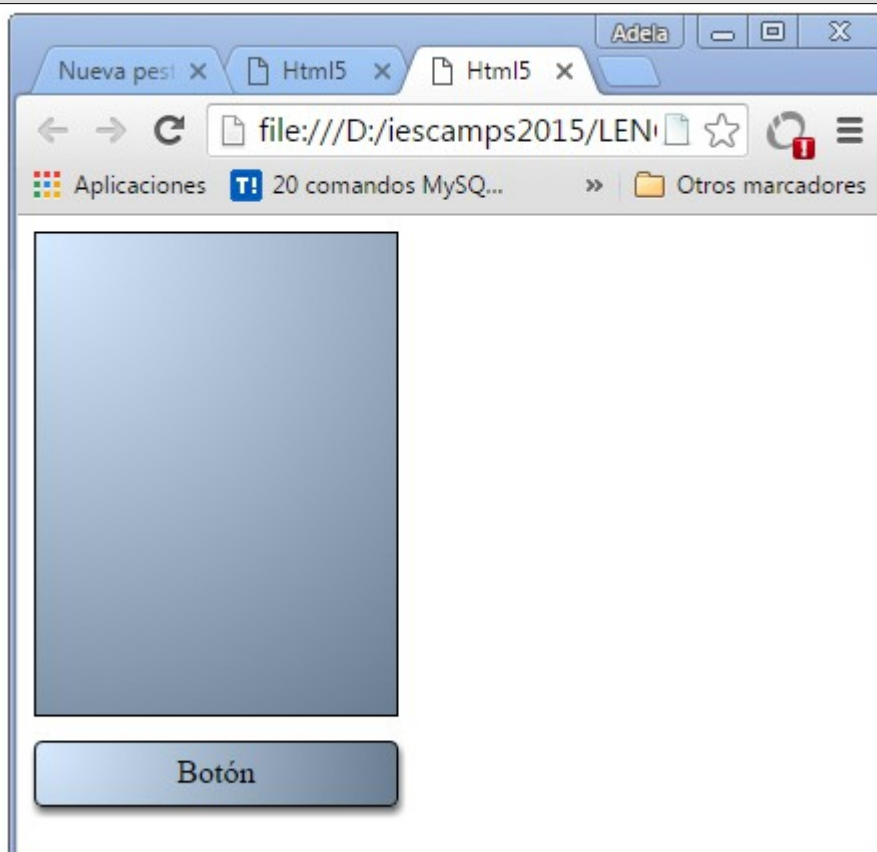
- Opera 11.1+ con el prefijo -o-, Opera 12.1+ sin prefijo.
- iOS Safari 3.2+ con el prefijo -webkit-.
- Android Browser 2.1+ con el prefijo -webkit-.

```
background-image: -webkit-radial-gradient(50% 50%, ellipse, #53B8EB, #B8FB97);  
background-image: -moz-radial-gradient(50% 50%, ellipse, #53B8EB, #B8FB97);  
background-image: -ms-radial-gradient(50% 50%, ellipse, #53B8EB, #B8FB97);  
background-image: -o-radial-gradient(50% 50%, ellipse, #53B8EB, #B8FB97);  
background-image: radial-gradient(50% 50%, ellipse, #53B8EB, #B8FB97);
```

Ejemplo

```
<!doctype html>  
<html>  
<head>  
<title>Html5</title>  
<meta charset="utf-8">  
<style type="text/css">  
.gradient1 {  
width: 180px; height: 240px;  
border: 1px solid black;  
background: -moz-radial-gradient(top left, circle, #D7EBFF,  
#697D91);  
background: -webkit-radial-gradient(top left, circle, #D7EBFF,  
#697D91);  
background: -ms-radial-gradient(top left, circle, #D7EBFF,  
#697D91);  
background: -o-radial-gradient(top left, circle, #D7EBFF,  
#697D91);  
background: radial-gradient(top left, circle, #D7EBFF, #697D91);}  
.gradient2 {  
display: block;  
height: 25px;  
width: 180px;  
padding-top: 6px;  
margin-bottom: 12px;  
border: 1px solid black;  
margin-top: 12px;  
text-align: center;  
border-radius: 4px;  
-webkit-border-radius: 4px;  
-moz-border-radius: 4px;  
box-shadow: 1px 3px 5px rgba(0,0,0,0.6);
```

```
-moz-box-shadow: 1px 3px 5px rgba(0,0,0,0.6);  
-webkit-box-shadow: 1px 3px 5px rgba(0,0,0,0.6);  
background: -moz-radial-gradient(top left, circle, #D7EBFF,  
#697D91);  
background: -webkit-radial-gradient(top left, circle, #D7EBFF,  
#697D91);  
background: -ms-radial-gradient(top left, circle, #D7EBFF,  
#697D91);  
background: -o-radial-gradient(top left, circle, #D7EBFF,  
#697D91);  
background: radial-gradient(top left, circle, #D7EBFF,  
#697D91);}  
</style>  
</head>  
<body>  
<div class="gradient1"></div>  
<div class="gradient2">Botón</div>  
</body>  
</html>
```



Mozilla Firefox añade además otras opciones que permiten crear degradados más complejos.

13. Opacidad y transparencia

Con las hojas de estilo CSS3 es posible variar la opacidad o la transparencia de un elemento.

Esta propiedad ha sido adoptada desde hace tiempo y es compatible con Firefox 2+, Safari 3.1+, Chrome 4+, Opera 9+, iOS Safari 3.2+ y Android Browser 2.1+.

No hay que preocuparse por los prefijos `-moz-` y `-webkit-`.

opacity:	Valor comprendido entre 1 y 0. Con el valor 1, la opacidad es completa y la transparencia nula. Con 0, la opacidad es nula y el elemento completamente transparente.
----------	--

Comentarios

- El resultado de la propiedad `opacity` está muy próximo al de la notación de color RGBA con transparencia. No obstante hay que destacar que la propiedad `opacity` se aplica al elemento íntegro, y a todos sus descendientes. RGBA solamente se aplica al elemento seleccionado.
- El elemento dotado de la propiedad `opacity` es translúcido, y deja entrever el elemento que se encuentra situado por debajo de él (ver los ejemplos a continuación).

En este punto de nuestro estudio de las hojas de estilo CSS3, no nos sorprenderá que Internet Explorer no implemente la propiedad `opacity`. Es preciso pasar por el filtro alpha, propietario de Microsoft.

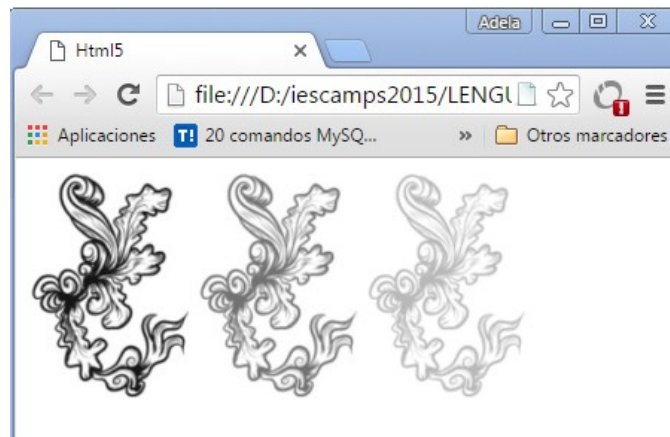
`filter:alpha(opacity=x)` donde x es un valor comprendido entre 100 y 0.

Ejemplo 1

```
<!DOCTYPE html>
<html lang="es">
<head>
<title>Html5</title>
<meta charset="UTF-8">
<body>



</body>
</html>
```

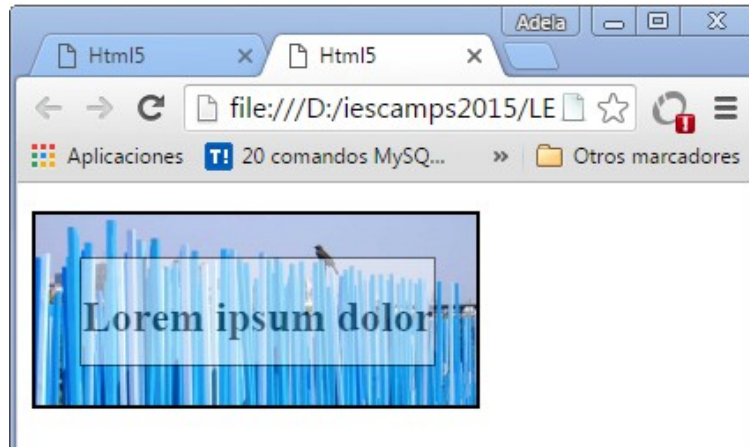


Ejemplo 2

Ilustremos la translucidez.

```
<!DOCTYPE html>
<html lang="es">
<head>
<title>Html5</title>
<meta charset="UTF-8">
<style type="text/css">
div.background { width: 250px; height: 108px;
background:url(blue.png);
border: 2px solid black;}
div.box { width: 200px; height: 60px;
margin: 24px 25px;
border: 1px solid black;
background-color: #ffffff;
filter: alpha(opacity=50);
opacity: 0.5;
text-align: center;}
</style>
</head>
<body>
<p></p>
<div class="background">
<div class="box">
```

```
<h2>Lorem ipsum dolor</h2>
</div>
</div>
</body>
</html>
```

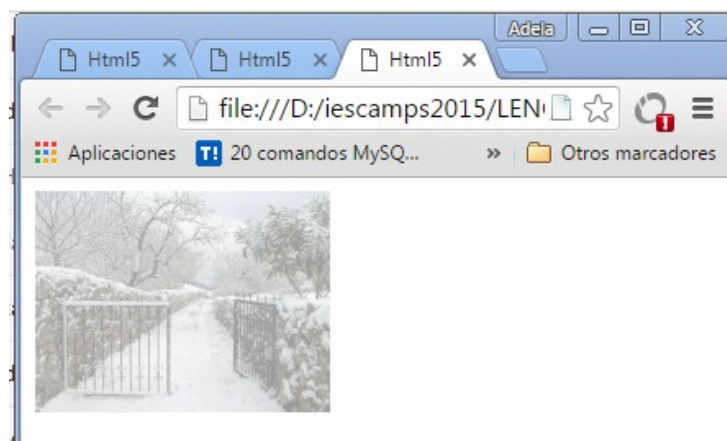


Ejemplo 3

Es tentador modificar la opacidad de la imagen cuando se pase el cursor del ratón sobre ella.

```
<!DOCTYPE html>
<html lang="es">
<head>
<title>Html5</title>
<meta charset="UTF-8">
<body>

</body>
</html>
```



En un primer instante la imagen se muestra con una opacidad de 0.4. Al pasar por encima el cursor del ratón (onmouseover) ésta se muestra con una opacidad máxima. Cuando se retira el cursor de la misma (onmouseout) la imagen vuelve a su estado inicial con una opacidad de 0.4.

14. Columnas múltiples

Presentar un contenido en varias columnas, como con Adobe InDesign, es otra novedad de las hojas de estilo CSS3.

Column-count:	Valor entero que determina el número de columnas en las que se va a visualizar el contenido del elemento.
column-width:	Valor que describe el ancho óptimo de cada columna (opcional).
column-gap:	Valor que determina el espacio o el padding entre dos columnas.
column-rule:	Define un borde entre las columnas.

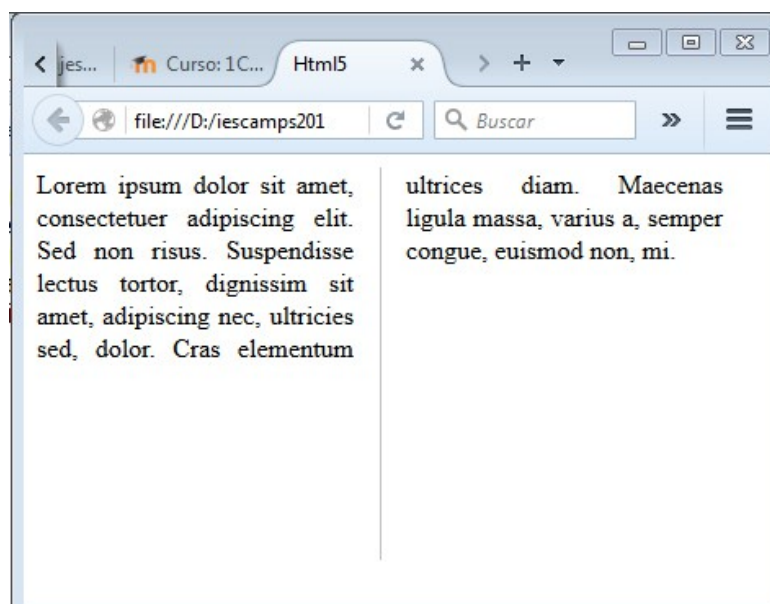
La propiedad column se reconoce por:

- Internet Explorer 10+.
- Firefox 3.0+ con el prefijo -moz-.
- Chrome 4.0+ con el prefijo -webkit-.
- Safari 3.1+ con el prefijo -webkit-.
- Opera 11.1 sin prefijo.
- iOS Safari 3.2+ con el prefijo -webkit-.
- Android Browser 2.1+ con el prefijo -webkit-.

-moz-column-count:	-webkit-column-count:
-moz-column-width:	-webkit-column-width:
-moz-column-gap:	-webkit-column-gap:
-moz-column-rule:	-webkit-column-rule:

Ejemplo

```
<!DOCTYPE html>
<html lang="es">
<head>
<title>Html5</title>
<meta charset="UTF-8">
<style type="text/css">
#box { width: 420px; height: 240px;
    text-align: justify;
    column-count: 2;
    column-gap: 2em;
    column-rule: 1px solid silver;
    -moz-column-count: 2;
    -moz-column-gap: 2em;
    -moz-column-rule: 1px solid silver;
    -webkit-column-count: 2;
    -webkit-column-gap: 2em;
    -webkit-column-rule: 1px solid silver;}
</style>
</head>
<body>
<div id="box">
Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed non
    risus. Suspendisse lectus tortor, dignissim sit in amet,
    adipiscing nec, ...
</div>
</body>
</html>
```



15.Transformaciones

Queda todavía una propiedad CSS3 con unos resultados, cuando menos, espectaculares. En efecto, sus posibilidades rompen en cierto modo con la rigidez del diseño de páginas Web. Esta propiedad permite aplicar distintas transformaciones sobre un elemento como por ejemplo una rotación, un cambio de escala, un zoom o una deformación oblicua. Unida a las transiciones (véase el punto siguiente), juntas forman un binomio que promete modificar en profundidad la visualización de los sitios Web del futuro.

transform:	<code>rotate(ángulo), rotateX(ángulo), rotateY(ángulo)</code> <code>scale(número), scaleX(número), scaleY(número)</code> <code>skew(ángulo), skewX(ángulo), skewY(ángulo)</code> <code>translate(longitud), translateX(longitud),</code> <code>translateY(longitud)</code>
------------	--

La propiedad transform está implementada por:

- Internet Explorer 9 con el prefijo -ms-, sin prefijo desde la versión 10+.
- Firefox 3.6+ con el prefijo -moz-, sin prefijo desde Firefox 16.0+.
- Chrome 4.0+ con el prefijo -webkit-.
- Safari 3.1+ con el prefijo -webkit-.
- Opera 10.5 con el prefijo -o-, sin prefijo desde Opera 12.1+.
- iOS Safari 3.2+ con el prefijo -webkit-.
- Android Browser 2.1+ con el prefijo -webkit-.

Veamos los distintos parámetros de esta propiedad transform.

Rotación

La rotación de un elemento se obtiene mediante el parámetro `rotate(ángulo)` donde ángulo puede definirse en grados (deg), radianes (rad) o gradientes (grad). Los valores positivos efectúan la rotación del elemento en el sentido de las agujas del reloj y los valores negativos lo hacen en sentido inverso. Cuando se define un único valor, éste aplica a los ejes X e Y. Con dos valores, se aplican respectivamente al eje de las X y al de las Y.

Ejemplo

```
<!DOCTYPE html>
<html lang="es">
<head>
<title>Html5</title>
<meta charset="UTF-8">
<style type="text/css">
body { padding-left: 15px;}
#box { width: 200px; height: 80px;
      font-size: 70px;
      border: 1px solid black;
      padding-left: 10px;
      -moz-transform: rotate(-20deg);
      -webkit-transform: rotate(-20deg);
      -o-transform: rotate(-20deg);
      transform: rotate(-20deg);}
</style>
</head>
<body>
<h1 id="box">CSS 3</h1>
</body>
</html>
```

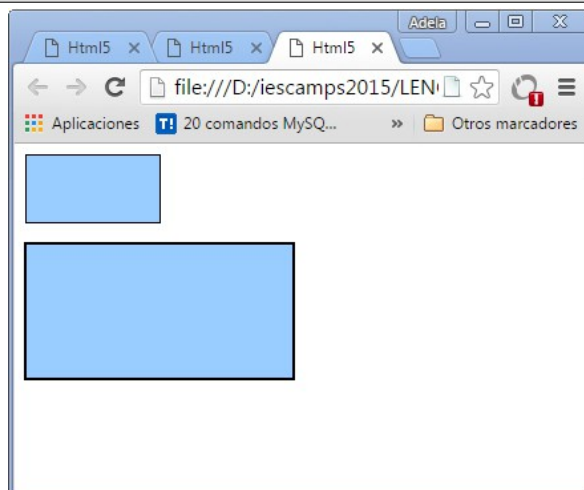


Cambio de escala

El cambio de escala de un elemento se obtiene mediante el parámetro scale(número) donde el número puede ser un valor entero o decimal, positivo o negativo. Los valores positivos realizan un zoom acercándose y los negativos un zoom alejándose. Cuando se define un único valor, éste aplica a los ejes X e Y. Con dos valores, se aplican respectivamente al eje de las X y al de las Y.

Ejemplo

```
<!DOCTYPE html>
<html lang="es">
<head>
<title>Html5</title>
<meta charset="UTF-8">
<style type="text/css">
div { border: 1px solid black;
      background-color: #9cf;
      width: 100px; height: 50px;}
#box { margin: 40px 50px;
      -moz-transform: scale(2);
      -webkit-transform: scale(2);
      -o-transform: scale(2);
      transform: scale(2);}
</style>
</head>
<body>
<div></div>
<div id="box" style=""></div>
</body>
</html>
```

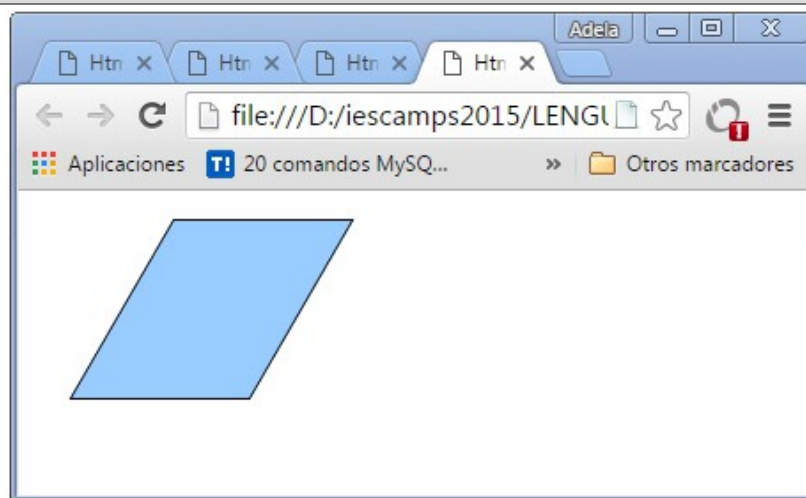


Deformación oblicua

La deformación oblicua de un elemento se obtiene mediante el parámetro skew(ángulo) donde el ángulo puede estar definido en grados (deg), radianes (rad) o gradientes (grad). Los valores positivos realizan la rotación del elemento en el sentido de las agujas del reloj y los valores negativos lo hacen en sentido contrario. Cuando se define un único valor, éste aplica a los ejes X e Y. Con dos valores, se aplican respectivamente al eje de las X y al de las Y.

Ejemplo

```
<!DOCTYPE html>
<html lang="es">
<head>
<title>Html5</title>
<meta charset="UTF-8">
<style type="text/css">
#box { width: 100px; height: 100px;
      border: 1px solid black;
      background-color: #9cf;
      margin-left: 50px;
      -moz-transform: skew(-30deg);
      -webkit-transform: skew(-30deg);
      -o-transform: skew(-30deg);
      transform: skew(-30deg);}
</style>
</head>
<body>
<p></p>
<div id="box" style=""></div>
</body>
</html>
```



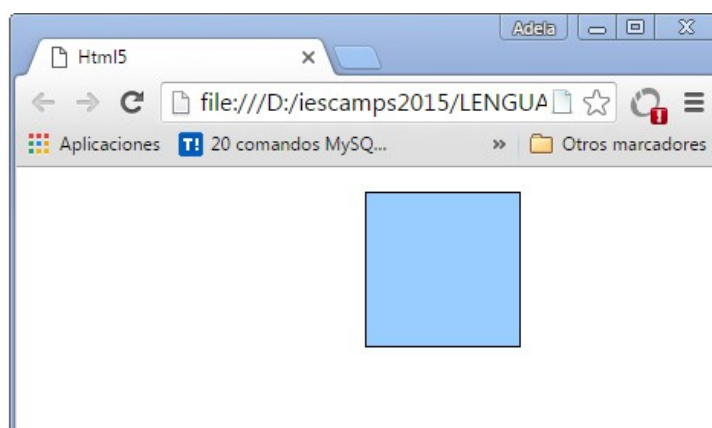
Desplazamiento

El desplazamiento de un elemento se obtiene mediante el parámetro `translate(longitud)` donde la longitud puede estar definida en px, %, em, in, mm o cm. Sobre el eje de las X los valores positivos efectúan un desplazamiento hacia la derecha y los valores negativos hacia la izquierda. Para el eje de las Y, los valores positivos efectúan un desplazamiento

hacia arriba y los valores negativos hacia abajo. Cuando se define un único valor, éste aplica a los ejes X e Y. Con dos valores, se aplican respectivamente al eje de las X y al de las Y.

Ejemplo

```
<!DOCTYPE html>
<html lang="es">
<head>
<title>Html5</title>
<meta charset="UTF-8">
<style type="text/css">
#box { width: 100px; height: 100px;
      border: 1px solid black;
      background-color: #9cf;
      -moz-transform: translate(220px, 0);
      -webkit-transform: translate(220px, 0);
      -o-transform: translate(220px, 0);
      transform: translate(220px, 0);
}
</style>
</head>
<body>
<p></p>
<div id="box" style=""></div>
</body>
</html>
```



Para probar todas las transformaciones posibles, no deje de consultar el sitio Web <http://www.westciv.com/tools/transforms/>

16. Transiciones

Las propiedades CSS3 de transición aportan movimiento en el diseño de la página, de forma similar a lo que se realiza mediante JavaScript.

Las animaciones se realizan principalmente en base a cuatro propiedades:

<code>transition-property</code>	Precisa las propiedades CSS que se desea animar, por ejemplo el color (color) y la anchura (width). Estas propiedades se precisan en una lista, separadas por comas. La palabra clave all (valor por defecto) designa todas las propiedades animadas del elemento.
<code>transition-duration</code>	Precisa la duración de la transición. Se expresa en s (segundos) o ms (milisegundos).
<code>transition-timing-function</code>	Precisa la función de transición que se quiere utilizar: ease: rápida al principio y más lenta al final. linear: rapidez constante en toda la duración de la animación. ease-in: lento al principio y acelerado hacia el final. ease-out: rápido al principio y desacelerado hacia el final. ease-in-out: inicio y final más lentos.
<code>transition-delay</code>	Precisa el retardo (o el avance) del inicio de la transición. Se expresa en s (segundos) o ms (milisegundos).

La forma abreviada `transition` evita tener que definir cada propiedad. Por ejemplo:

```
transition: width 2s ease, height 3s linear;
```

La propiedad `transition` está implementada en:

- Internet Explorer 10+ sin prefijo.
- Firefox 4.0+ con el prefijo -moz-, sin prefijo desde Firefox 16.0+.
- Chrome 4.0+ con el prefijo -webkit-.
- Safari 3.1+ con el prefijo -webkit-.
- Opera 10.5 con el prefijo -o-, sin prefijo desde Opera 12.1+.

- iOS Safari 3.2+ con el prefijo -webkit-.
- Android Browser 2.1+ con el prefijo -webkit-.

Ejemplo 1

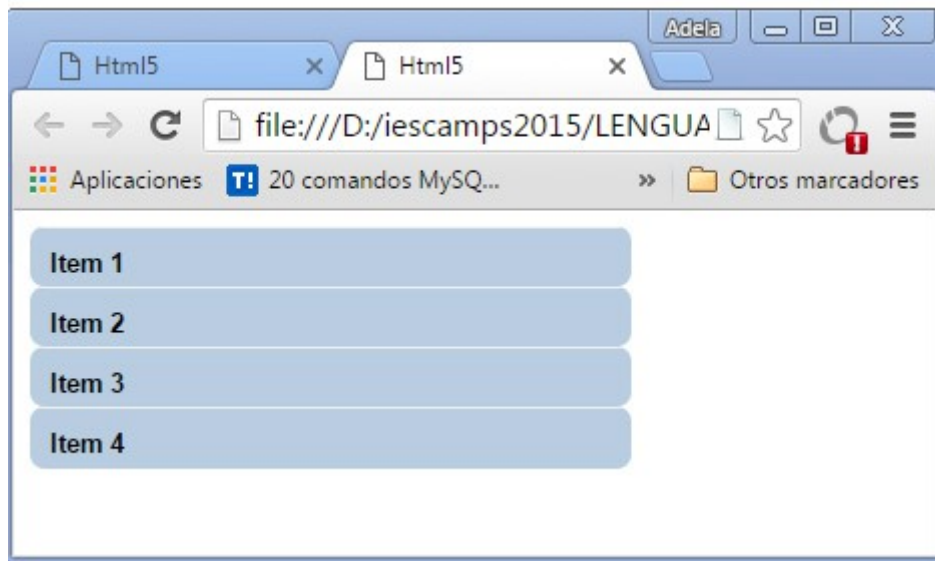
La presentación del contenido en forma de acordeón ha tenido un verdadero éxito en la red. La propiedad CSS3 transform permite realizar este efecto mediante hojas de estilo, únicamente, sin tener que recurrir a código JavaScript.

```
<!DOCTYPE html>
<html lang="es">
<head>
<title>Html5</title>
<meta charset="UTF-8">
<style type="text/css">
.acordeonvertical>ul { margin: 0;
                    padding: 0;
                    list-style: none;
                    width: 300px;}
.acordeonvertical>ul>li {
display: block;
overflow: hidden;
margin: 0;
padding: 0;
list-style: none;
height: 30px;
width: 300px;
background-color: rgb(215,230,245);
-webkit-border-radius: 7px;
-moz-border-radius: 7px;
border-radius: 7px;
transition: height 0.3s ease-in-out;
-moz-transition: height 0.3s ease-in-out;
-webkit-transition: height 0.3s ease-in-out;
-o-transition: height 0.3s ease-in-out;}
.acordeonvertical>ul>li>h3 {
display: block;
margin: 0;
padding: 10px;
height: 19px;
border-top: #f0f0f0 1px solid;
font-family: Arial, Helvetica, sans-serif;
font-size: 80%;
color: #000;
background: rgb(185,205,225);}
.acordeonvertical>ul>li>div { margin:0;
                    overflow: auto;
                    padding: 10px;
```

```

        height: 150px;}
.acordeonvertical>ul>li:hover { height: 150px;}
.acordeonvertical:hover>ul>li:hover>h3 {
color: #fff;
background: rgb(125,165,205);
font-size: 80%;}
.acordeonvertical>ul>li>h3:hover { cursor:pointer;}
</style>
</head>
<body>
<div class="acordeonvertical">
<ul>
<li>
<h3>Item 1</h3>
<div>
Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed non
risus. Suspendisse lectus tortor, dignissim sit amet, adipiscing
nec, ultricies sed, dolor. Cras elementum ultrices diam.
</div>
</li>
<li>
<h3>Item 2</h3>
<div>
Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed non
risus. Suspendisse lectus tortor, dignissim sit amet, adipiscing
nec, ultricies sed, dolor. Cras elementum ultrices diam.
</div>
</li>
<li>
<h3>Item 3</h3>
<div>
Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed non
risus. Suspendisse lectus tortor, dignissim sit amet, adipiscing
nec, ultricies sed, dolor. Cras elementum ultrices diam.
</div>
</li>
<li>
<h3>Item 4</h3>
<div>
Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed non
risus. Suspendisse lectus tortor, dignissim sit amet, adipiscing
nec, ultricies sed, dolor. Cras elementum ultrices diam.
</div>
</li>
</ul>
</div>
</body>
</html>

```



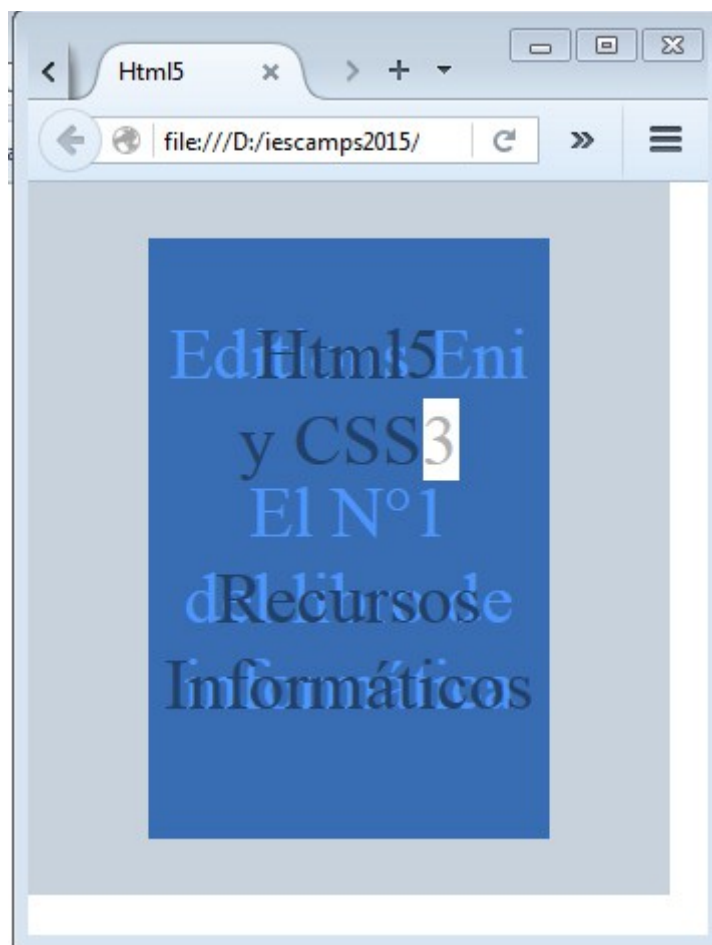
Ejemplo 2

Una tarjeta que se convierte en 3D.

```
<!doctype html>
<html>
<head>
<title>Html5</title>
<meta charset="utf-8">
<script>
function flip (event){
var element = event.currentTarget;
element.className = (element.className == 'tarjeta') ?
'tarjeta flipped' : 'tarjeta';
}
</script>
<style>
body { margin: 0px;}
#box { height: 356px; width: 320px;
background-color: rgba(145,165,185, 0.5);
-webkit-perspective: 600;}
.tarjeta { position: absolute;
height: 300px;
width: 200px;
left: 60px;
top: 28px;
-webkit-transform-style: preserve-3d;
-webkit-transition-property: -webkit-transform;
-webkit-transition-duration: 1.5s;}
```



```
.tarjeta.flipped { -webkit-transform: rotateY(180deg);}
.verso { position: absolute;
        height: 300px;
        width: 200px;
        -webkit-border-radius: 10px;
        -webkit-box-shadow: 0px 2px 6px rgba(0, 0, 0, 0.5);
        -webkit-backface-visibility: hidden;}
.verso > p { margin-top: 36px;
            margin-bottom: 0;
            text-align: center;
            font-size: 36px;}
.recto { color: rgb(78,150,249);
        background-color: rgb(34,65,108);}
.back { color: rgb(34,65,108);
        background-color: rgba(78,150,249,0.5);
        -webkit-transform: rotateY(180deg);}
</style>
</head>
<body>
<div id="box">
<div id="tarjeta" class="tarjeta" onclick="flip(event)">
<div id="recto" class="recto verso">
<p>Editions Eni</p>
<p>El N°1<br>del libro de <br>informática</p>
</div>
<div id="back" class="back verso">
<p>Html5<br>y CSS3</p>
<p>Recursos<br>Informáticos</p>
</div>
</div>
</div>
</body>
</html>
```



17. Los filtros gráficos

Con la propiedad `filter`, es posible aplicar sobre imágenes o sobre otro elemento Html efectos de filtros hasta ahora reservados a programa como Photoshop, Corel Paint Shop o Gimp, por citar algunos.

No se entusiasme muy rápido porque esta especificación de W3C es muy reciente (Editor's Draft del 16 de noviembre de 2012) y de momento solo la implementan Google Chrome 18+ y Safari 6+ con el prefijo `-webkit-`.

Su sintaxis es:

```
-webkit-filter: filtre(valor);
```

Los filtros disponibles son :

<code>grayscale(valor)</code>	Nivel de gris, donde valor es un número entre 0 y 1 o un porcentaje.
<code>sepia(valor)</code>	Efecto sepia, donde valor es un número entre 0 y 1 o un porcentaje.
<code>saturate(valor)</code>	Saturación, donde valor es un número entre 0 y 10 o un porcentaje.
<code>hue-rotate(ángulo)</code>	El ángulo se expresa en grados.
<code>invert(valor)</code>	Efecto negativo, donde valor es un número entre 0 y 1 o un porcentaje.
<code>opacity(valor)</code>	Opacidad, donde valor es un número entre 0 y 1 o un porcentaje.
<code>brightness(valor)</code>	Luminosidad, donde valor es un número entre 0 y 1 o un porcentaje.
<code>contrast(valor)</code>	Contraste, donde valor es un número entre 0 y 1 o un porcentaje.
<code>blur(valor)</code>	Difuminar, donde el valor se expresa en pixels.
<code>drop-shadow(sombra)</code>	Sombra, por ejemplo <code>drop-shadow(6px 6px 20px black)</code> .

Para visualizar estos efectos, vaya a las webs <http://jsbin.com/uziyih> o <http://html5-demos.appspot.com/static/css/filters/index.html>.

Se anuncian multitud de filtros como halftone, motion-blur, posterize, bump, affine-transform, bloom, gloom, mosaic, displace, edge-detect, pinch, twirl.

Ejemplo

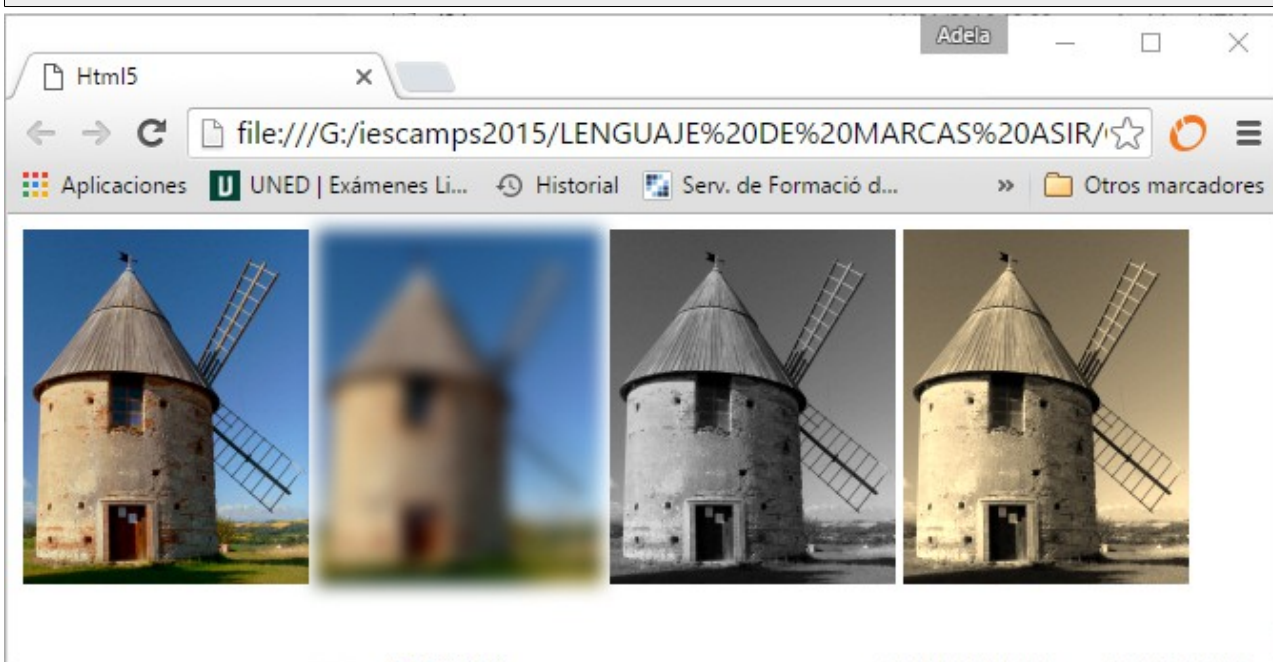
Apliquemos algunos filtros (blur, grayscale, sepia) a una imagen.

```
<!DOCTYPE html>
<html lang="es">
<head>
<title>Html5</title>
```

```
<meta charset="UTF-8">
</head>
<body>




</body>
</html>
```



Una gran cantidad de efectos con un código mínimo.

18. Las máscaras

La propiedad CSS mask permite ocultar una parte de elementos visuales aplicando una máscara o un degradado. Este efecto, que habitualmente realizan los programas gráficos, se implementa con una propiedad de estilo.

Chrome 4+, Safari 4+, iOS 3.2+, Android 2.1+ con el prefijo -webkit- implementan las máscaras.

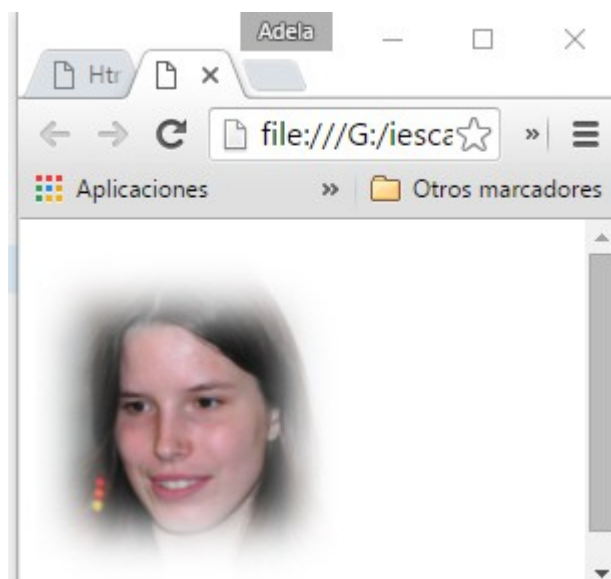
Este efecto, solo implementado por el motor Webkit, va a ser una especificación de W3C (Editor's Draft 11 de diciembre de 2012).

Apuntemos también que una máscara es una imagen png compuesta de píxeles y de transparencia.

Ejemplo 1

Ilustramos este efecto de máscara con la aplicación de una máscara (mask.png, ver a continuación).

```
<!DOCTYPE html>
<html lang="es">
<head>
<title>Html5</title>
<meta charset="UTF-8">
<style type="text/css">
.mascara{ -webkit-mask: url(mask.png) no-repeat center center;}
</style>
</head>
<body>
<br>
</body>
</html>
```



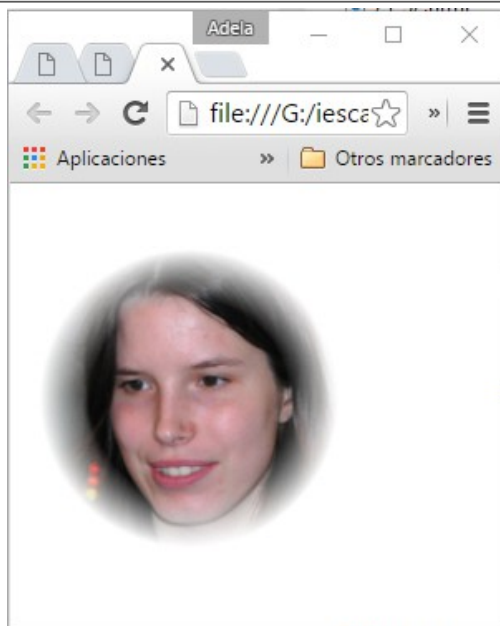
Ejemplo 2

Aplicamos un máscara para un degradado circular.

```
<!DOCTYPE html>
<html lang="es">
<head>
<title>Html5</title>
```

```
<meta charset="UTF-8">
</head>
<body>

</body>
</html>
```



19. La fusión de imágenes

Siempre en el ámbito de los gráficos, la propiedad CSS `cross-fade` permite realizar fusiones de imágenes.

Solo Google Chrome 18+ con el prefijo `-webkit-` implementa esta propiedad, todavía experimental .

<code>background-image:</code>	<code>-webkit-cross-fade(imagen1, imagen2, porcentaje);</code>
--------------------------------	--

Comentario

El valor porcentaje es el porcentaje de opacidad aplicado a la segunda imagen de la fusión.

Ejemplo

Tenemos dos imágenes (paisaje.png y retrato.png). Apliquemos sobre ellas un efecto de fusión.

El resultado es :

```
<!doctype html>
<html>
<head>
<title>Html5</title>
<meta charset="utf-8">
<style type="text/css">
.image { width: 350px; height: 253px;}
.ejemplo .image {
background-image: -webkit-cross-fade(url(paisaje.png),
url(retrato.png), 25%);
}
</style>
</head>
<body>
<br>
<div class="ejemplo">
<div class="image"></div>
</div>
</body>
</html>
```

