

Nuevas propiedades CSS3

Para facilitar el aprendizaje de las nuevas propiedades CSS3 vamos a ir aplicándolas sobre la misma plantilla o documento de ejemplo. Por este motivo, comenzaremos por crear un documento HTML sencillo con algunos estilos básicos.

Plantilla.html:

```
<!DOCTYPE html>
<html lang="es">
<head>
<title>Curso HTML5 y CSS3</title>
<link rel="stylesheet" href="estiloscss3.css">
<!--[if lt IE 9]>
<script src="http://html5shiv.googlecode.com/svn/trunk/html5.js">
</script>
<![endif]-->
</head>
<body>
<header id="principal">
<h1>Curso HTML5 y CSS3</h1>
</header>
</body>
</html>
```

Estiloscss3.css:

```
body { text-align: center; }
#principal {
display: block;
width: 500px;
margin: 50px auto;
padding: 15px;
text-align: center;
border: 1px solid #999999;
background: #DDDDDD;
}
#titulo { font: bold 36px verdana, sans-serif; }
```

No hay nada nuevo en estas reglas, sólo los estilos necesarios para dar forma a la plantilla y crear una caja de 500 píxeles de ancho posicionada en el centro de la ventana, con un fondo gris, un borde y un texto grande en su interior que dice “Curso HTML5 y CSS3”.

Vamos a destacar el uso de la propiedad `display: block` para asegurarnos de que el navegador interpreta el elemento `<header>` como un elemento en bloque. Para ver más usos de esta propiedad se recomienda consultar la siguiente URL:

http://librosweb.es/css_avanzado/capitulo_4/propiedad_display.html.

Si visualizamos la página se verá así:



Curso HTML5 y CSS3

Bordes con esquinas redondeadas (border-radius)

Durante muchos años los diseñadores han sufrido intentando lograr el efecto de esquinas redondeadas en las cajas de sus páginas web, eran esa clase de cosas que nos hacía pensar: “debería ser fácil hacerlo”.

Gracias a la propiedad `border-radius` de CSS3, ahora podemos hacerlo de forma muy sencilla (si queremos asegurarnos la compatibilidad con el mayor número de navegadores posible, debemos usar los prefijos `-moz-` y `-webkit-` para que funcione en navegadores basados en motores Gecko y WebKit, como Firefox, Safari y Google Chrome).

Si todas las esquinas tienen la misma curvatura podemos utilizar un solo valor. Las siguientes propiedades las añadiremos a la regla `#principal` de nuestro ejemplo:

```
-moz-border-radius: 20px;  
-webkit-border-radius: 20px;  
border-radius: 20px;
```

Curso HTML5 y CSS3

Sin embargo, como ocurre con las propiedades `margin` y `padding`, podemos también declarar un valor diferente para cada esquina. Los valores se aplicarán en el siguiente orden: esquina superior izquierda, esquina superior derecha, esquina inferior derecha y esquina inferior izquierda:

```
-moz-border-radius: 20px 10px 30px 50px;  
-webkit-border-radius: 20px 10px 30px 50px;  
border-radius: 20px 10px 30px 50px;
```

Curso HTML5 y CSS3

Al igual que con `margin` o `padding`, `border-radius` también puede trabajar sólo con dos valores: el primer valor será asignado a la primera y tercera esquina (superior izquierda, inferior derecha), y el segundo valor a la segunda y cuarta esquina (superior derecha, inferior izquierda):

```
-moz-border-radius: 20px 50px;  
-webkit-border-radius: 20px 50px;  
border-radius: 20px 50px;
```

Curso HTML5 y CSS3

También podemos dar forma a las esquinas declarando un segundo grupo de valores separados por una barra. Los valores a la izquierda de la barra representarán el radio horizontal, mientras que los valores a la derecha representan el radio vertical. La combinación de estos valores genera una elipse:

```
-moz-border-radius: 50px / 20px;  
-webkit-border-radius: 50px / 20px;  
border-radius: 50px / 20px;
```

Curso HTML5 y CSS3

Para probar de una forma sencilla cómo quedaría esta propiedad puedes utilizar la herramienta disponible en la url: <http://www.cssmatic.com/border-radius>

Sombras en las cajas (box-shadow)

Otro muy buen efecto que había sido complicado de lograr hasta este momento, son las sombras. Tradicionalmente, los diseñadores combinaban imágenes, elementos y algunas propiedades CSS para generar sombras. Gracias a CSS3 y a la nueva propiedad box-shadow podemos aplicar sombras a nuestras cajas con una simple línea de código.

La propiedad box-shadow necesita al menos tres valores: Los dos primeros valores, expresados en píxeles, establecen el desplazamiento de la sombra y el tercero indicará el color de la misma.

Este desplazamiento puede ser positivo o negativo. Los valores indican, respectivamente, la distancia horizontal y vertical desde la sombra al elemento. Valores negativos posicionarán la sombra a la izquierda y arriba del elemento, mientras que valores positivos crearán la sombra a la derecha y debajo del elemento. Valores de 0 píxeles o nulos posicionarán la sombra, exactamente, detrás del elemento, permitiendo la posibilidad de crear un efecto difuminado a su alrededor.

Veamos cómo quedaría nuestra caja, después de aplicarle los siguientes estilos:

```
-moz-box-shadow: 5px 5px rgb(150,150,150);  
-webkit-box-shadow: 5px 5px rgb(150,150,150);  
box-shadow: 5px 5px rgb(150,150,150);
```

Curso HTML5 y CSS3

Como se puede observar, al igual que ocurría con la propiedad border-radius, debemos incluir los prefijos oportunos para que funcione en todos los navegadores.

La sombra que hemos obtenido es sólida, sin gradientes o transparencias (no es, realmente, como una sombra suele aparecer). Existen algunos parámetros que nos van a permitir mejorar la apariencia de la sombra.

Otro valor que se puede agregar a la propiedad ya estudiada antes de indicar el color es la distancia de difuminación. Este será un valor expresado en píxeles que indicará como se va a difuminar la sombra:

```
-moz-box-shadow: 5px 5px 10px rgb(150,150,150);  
-webkit-box-shadow: 5px 5px 10px rgb(150,150,150);  
box-shadow: 5px 5px 10px rgb(150,150,150);
```

Curso HTML5 y CSS3

Si añadimos un cuarto valor en píxeles antes del color, veremos cómo aumenta el tamaño de la sombra en todas las direcciones. Este efecto cambia un poco la naturaleza de la sombra expandiendo el área que cubre:

```
-moz-box-shadow: 5px 5px 10px 10px rgb(150,150,150);  
-webkit-box-shadow: 5px 5px 10px 10px rgb(150,150,150);  
box-shadow: 5px 5px 10px 10px rgb(150,150,150);
```

Curso HTML5 y CSS3

El último valor posible para box-shadow no es un número, sino una palabra clave: `inset`. Esta palabra clave la indicaremos después del color, y convierte la sombra externa en una sombra interna, lo cual provee un efecto de profundidad al elemento afectado.

```
-moz-box-shadow: 5px 5px 10px rgb(150,150,150) inset;  
-webkit-box-shadow: 5px 5px 10px rgb(150,150,150) inset;  
box-shadow: 5px 5px 10px rgb(150,150,150) inset;
```

Curso HTML5 y CSS3

Es importante tener en cuenta que las sombras no expanden el elemento o incrementan su tamaño, por lo que, siempre tendremos que controlar cuidadosamente que el espacio disponible es suficiente para que la sombra sea expuesta y correctamente dibujada en la pantalla sin superponerse con otros elementos cercanos.

Para probar de una forma sencilla cómo quedaría esta propiedad puedes utilizar la herramienta disponible en la url: <http://www.cssmatic.com/box-shadow>

Sombras en los textos (text-shadow)

La propiedad box-shadow fue diseñada especialmente para ser aplicada en cajas. Si intentamos aplicar este efecto a un elemento `<h1>`, por ejemplo, la caja invisible ocupada por este elemento en la pantalla tendrá una sombra, pero no el contenido

del elemento. Para crear sombras para figuras irregulares como textos, existe una propiedad especial llamada `text-shadow`.

Los valores para `text-shadow` son similares a los usados para `box-shadow`. Podemos declarar la distancia horizontal y vertical de la sombra con respecto al objeto, el radio de difuminación y el color de la sombra.

Veamos cómo queda nuestro título al aplicarle una sombra:

```
#titulo
{
font: bold 36px verdana, sans-serif;
text-shadow: 5px 5px 5px rgb(150,150,150);
}
```

Curso HTML5 y CSS3

Para probar de una forma sencilla cómo quedaría esta propiedad puedes utilizar la herramienta disponible en la url: <http://css3gen.com/text-shadow/>

Fondos con gradiente

Los gradientes son uno de los efectos más utilizados en la web. Anteriormente, era necesario el uso de imágenes para lograr este efecto, pero ahora es realmente fácil de hacer usando CSS.

Los gradientes son configurados como fondos, por lo que podemos usar las propiedades `background` o `background-image` para declararlos.

Tenemos dos tipos de gradientes: gradiente lineal y gradiente radial.

Gradiente lineal

Los gradientes lineales los indicamos con la función `linear-gradient`, que recibe los siguientes parámetros:

- Punto de comienzo : indica el punto donde comenzará el gradiente. Puede ser especificado en píxeles, porcentaje o usando las palabras clave `top`, `bottom`, `left` y `right`. También puede ser reemplazado por un ángulo para declarar una dirección específica del gradiente.
- Color inicial: indica con que color comenzará el gradiente.
- Color final: indica con que color terminará el gradiente.
-

Al igual que las propiedades anteriores cada navegador tendrá sus prefijos para trabajar con esta función.

A continuación, veremos cómo queda nuestra plantilla de ejemplo aplicando un gradiente a la regla `#principal`:

```
background: -webkit-linear-gradient(top, #FFFFFF, #006699);
background: -moz-linear-gradient(top, #FFFFFF, #006699);
background: linear-gradient(to bottom, #FFFFFF, #006699);
```

Curso HTML5 y CSS3

```
background: -webkit-linear-gradient(30deg, #FFFFFF, #006699);  
background: -moz-linear-gradient(30deg, #FFFFFF, #006699);  
background: linear-gradient(30deg, #FFFFFF, #006699);
```

Curso HTML5 y CSS3

También podemos declarar los puntos de terminación para cada color de la siguiente forma:

```
background: -webkit-linear-gradient(30deg, #FFFFFF 50%, #006699 90%);  
background: -moz-linear-gradient(30deg, #FFFFFF 50%, #006699 90%);  
background: linear-gradient(30deg, #FFFFFF 50%, #006699 90%);
```

Curso HTML5 y CSS3

Gradiente radial

La sintaxis estándar para los gradientes radiales sólo difiere en unos pocos aspectos con respecto a la anterior. Debemos usar la función radial-gradient() y un nuevo atributo para la forma:

```
background: -webkit-radial-gradient(circle, #FFFFFF 0%, #006699 200%);  
background: -moz-radial-gradient(circle, #FFFFFF 0%, #006699 200%);  
background: radial-gradient(circle, #FFFFFF 0%, #006699 200%);
```

Curso HTML5 y CSS3

Existen dos posibles valores para la forma: `circle` y `ellipse` (el valor por defecto será `ellipse`). Además, como puedes observar en el ejemplo, para cada color se debe indicar la posición donde las transiciones comienzan (en el ejemplo, el gradiente comenzaría en la posición 0% y terminaría en la posición 200%).

Transparencias

Tenemos dos formas de definir las transparencias de un elemento: expresando el color del elemento con transparencia o indicando un valor que indicará la opacidad del elemento.

RGBA

Hasta ahora, hemos declarado los colores como sólidos utilizando valores hexadecimales o la función `rgb()`. CSS3 ha agregado una nueva función llamada `rgba()` que simplifica la asignación de colores y transparencias.

La función `rgba()` tiene cuatro atributos. Los tres primeros son similares a los usados en `rgb()` y, simplemente, declaran los valores para los colores rojo, verde y azul en números decimales del 0 al 255. El último, en cambio, corresponde a la nueva capacidad de opacidad. Este valor se debe encontrar dentro de un rango que va de 0 a 1, con 0 como totalmente transparente y 1 como totalmente opaco. Por ejemplo, podemos mejorar la sombra del texto de nuestro título de ejemplo utilizando esta función de la siguiente forma:

```
#titulo {  
    font: bold 36px verdana, sans-serif;  
    text-shadow: 5px 5px 5px rgba(0,0,0,0.5);  
}
```

Curso HTML5 y CSS3

Reemplazamos la función `rgb()` por `rgba()` en la sombra del título y agregamos un valor de opacidad/transparencia de 0.5.

Ahora, la sombra de nuestro título se mezclará con el fondo creando un efecto mucho más natural.

HSLA

La función `hsla()` es, simplemente, otra función para generar colores que, en lugar de utilizar distintos valores para los colores rojo, verde y azul, utiliza distintos valores de tono, saturación y luminosidad.

La sintaxis es la siguiente: `hsla(tono, saturación, luminosidad, opacidad)`.

Si utilizamos esta función para cambiar el color de la fuente de nuestro título, obtendremos el siguiente resultado:

```
color: hsla(120, 100%, 50%, 0.5);
```

Curso HTML5 y CSS3

Propiedad `opacity`

El valor de la propiedad `opacity` se establece mediante un número comprendido entre 0.0 y 1.0 (al igual que antes, 0.0 será transparente y 1.0 totalmente opaco). En el siguiente ejemplo, se establece la propiedad `opacity` con un valor de 0.5 para conseguir una transparencia del 50% sobre dos de los elementos `<div>`:

```
#segundo, #tercero { opacity: 0.5; }
#primero { background-color: blue; }
#segundo { background-color: red; }
#tercero { background-color: green; }
```

Outline

La propiedad `outline` es una vieja propiedad CSS que ha sido expandida en CSS3 para incluir un valor de desplazamiento.

Esta propiedad se usaba para crear un segundo borde, y ahora ese borde puede ser mostrado alejado del borde real del elemento.

Por ejemplo, podemos añadir un segundo borde de 2 píxeles y un desplazamiento de 15 píxeles a la caja de nuestro título, si ponemos las siguientes propiedades en la regla `#principal`:

```
outline: 2px dashed #000099;
outline-offset: 15px;
```



Border-image

Los posibles efectos logrados por las propiedades `border` y `outline` están limitados a líneas simples y sólo algunas opciones de configuración. La nueva propiedad `border-image` fue incorporada para superar estas limitaciones y dejar en manos del diseñador la calidad y variedad de bordes disponibles, ofreciendo la alternativa de utilizar imágenes propias.

Al igual que otras propiedades CSS, la propiedad `border-image` es una propiedad compuesta por varios valores, y estos valores simples podemos indicarlos mediante otras propiedades más simples que veremos a continuación.

Imagen de borde: `border-image-source`

La propiedad `border-image-source` establece la imagen a utilizar como imagen de borde. El elemento tiene que tener definido un `border2` y la imagen se amplía o reduce para mostrarse completa en función del ancho que le hayamos puesto a la propiedad `border`. Si sólo se utiliza esta propiedad, la imagen se muestra en las cuatro esquinas del elemento.

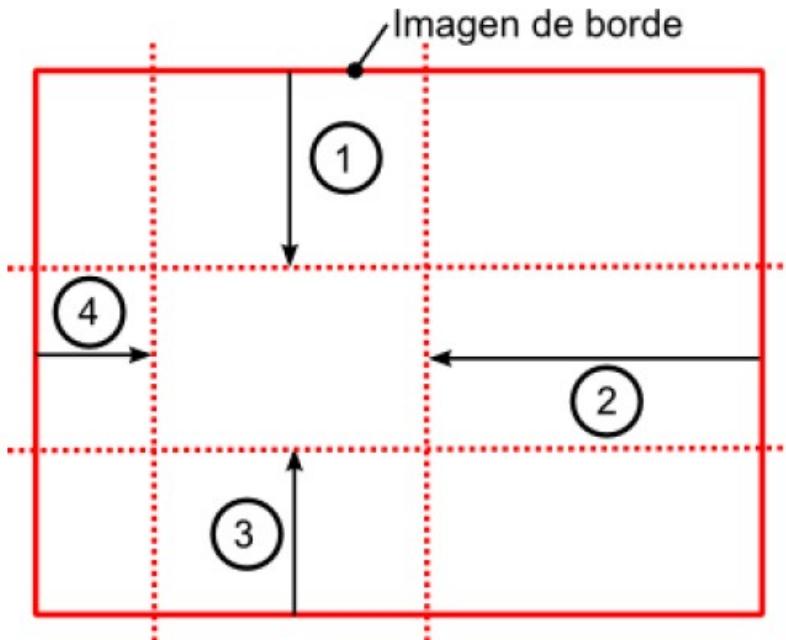
```
#principal {
    display: block;
    width: 500px;
    margin: 50px auto;
    padding: 15px;
    text-align: center;
    border: 50px;
    border-style:solid;
    border-image-source: url("fo_bola_100.png");
}
```

Troceado de la imagen de borde: border-image-slice

La propiedad `border-image-slice` permite trocear la imagen de borde, de manera que cada trozo se coloque en un lado del borde.

Los posibles valores de la propiedad `border-image-slice` son:

- De uno a cuatro valores en píxeles. Estos valores indican a qué distancia de los bordes se recorta la imagen, como muestra la imagen siguiente:

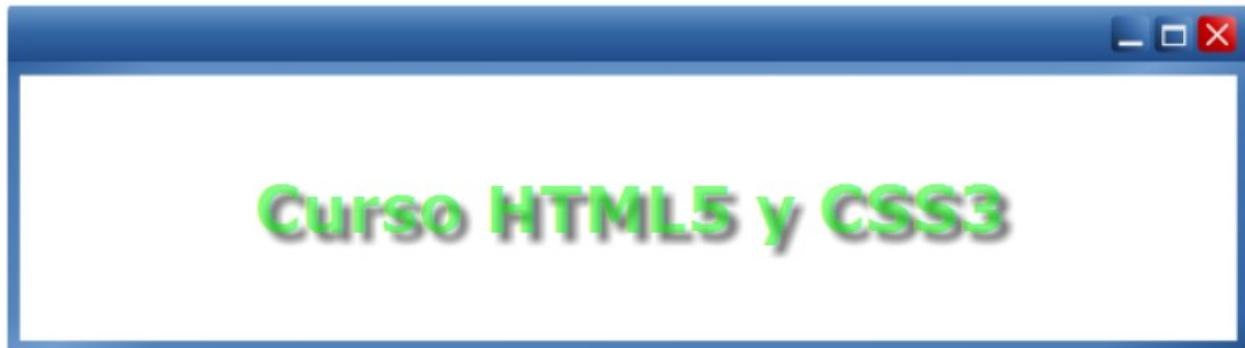


Cada uno de los ocho trozos (descartando el trozo central) se utiliza en cada una de las ocho zonas correspondientes del borde. Al igual que en otras propiedades como `margin` o `padding`, podemos indicar sólo dos valores, o incluso, un único valor:

```
border: 27px;  
border-style:solid;  
border-image-source: url("fo_pastilla_60.png");  
border-image-slice: 27;
```



```
border-width: 50px 80px 20px 80px;  
border-style:solid;  
border-image-source: url("fo_ventana.png");  
border-image-slice: 50 80 20 80;
```



fill: este valor hace que la parte central de la imagen se utilice para rellenar el elemento.

```
border: 27px;  
border-style:solid;  
border-image-source: url("fo_pastilla_60.png");  
border-image-slice: 27 fill;
```



Repetición de la imagen de borde: border-image-repeat

La propiedad border-image-repeat indica de qué forma se realizará el relleno de los bordes del elemento. Los posibles valores son:

- stretch: la imagen se estira o encoge para ocupar todo el espacio necesario

```
border: 40px;  
border-style:solid;  
border-image-source: url("fo_hormigas_210.png");  
border-image-slice: 70;  
border-image-repeat: stretch;
```
- repeat: la imagen se repite para ocupar todo el espacio necesario
- round: la imagen se repite el máximo número entero de veces posible hasta llenar el espacio, la imagen será reescalada para que ninguna de las repeticiones quede cortada.