



LENGUAJE DE MARCAS

UT-3. Novedades CSS3

2021 - 2022



Nuevas propiedades de CSS3

Son muchas las propiedades añadidas en CSS3, y muy irregular la implantación en los diferentes navegadores, hasta el punto de que cada uno de ellos en muchos casos ha desarrollado por su cuenta algunas de las implementaciones añadiendo un prefijo distintivo que es:

- o para ópera

- moz para firefox,

- webkit para chrome, safari y otros que usen el Webkit y

- ms para Internet Explorer.



Nuevas propiedades de CSS3

ejemplo:

```
<head>
<style>
#grad1{
  height: 200px;
  background: -webkit-linear-gradient(red, blue); /* For Safari 5.1 to 6.0 */
  background: -o-linear-gradient(red, blue); /* For Opera 11.1 to 12.0 */
  background: -moz-linear-gradient(red, blue); /* For Firefox 3.6 to 15 */
  background: linear-gradient(red, blue); /* Standard syntax */
}
</style>
</head>
```



Bordes redondeados (border-radius)

La propiedad border-radius permite crear esquinas redondeadas. Especificamos en píxeles u otra medida el radio de redondeo de la o las esquinas.

Podemos indicar un único valor que se asignará a los cuatro vértices:

```
#recuadro1{  
  border-radius: 20px;  
  background-color:#ddd;  
  width:200px;  
  padding:10px;  
}
```



Bordes redondeados (border-radius)

La propiedad border-radius permite crear esquinas redondeadas. Especificamos en píxeles u otra medida el radio de redondeo de la o las esquinas.

Podemos indicar un único valor que se asignará a los cuatro vértices:

```
#recuadro1{  
  border-radius: 20px;  
  background-color:#ddd;  
  width:200px;  
  padding:10px;  
}
```

Recuadro 1

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aenean commodo ligula eget dolor. Aenean massa. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Donec quam felis, ultricies nec,



Bordes redondeados (border-radius)

También podemos indicar el redondeo de cada vértice en forma independiente (el orden de los valores son la esquina superior izquierda, la esquina superior derecha, la esquina inferior derecha y por último la esquina inferior izquierda):

```
#recuadro2{  
  border-radius: 20px 40px 60px 80px;  
  background-color:#aa0;  
  width:200px;  
  padding:10px;  
  margin-top:10px;  
}
```

Recuadro 2

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aenean commodo ligula eget dolor. Aenean massa. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Donec quam felis, ultricies nec,



Bordes redondeados de alguno de los vértices (border-top-left-radius etc.)

Si tenemos que redondear solo alguno de los cuatro vértices podemos utilizar alguna de las siguientes propiedades:

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

Con esta definición de la propiedades border-top-left-radius y border-bottom-right-radius deben aparecer redondeados los vértices superior izquierdo e inferior derecho:

```
#recuadro1{  
  border-top-left-radius: 20px;  
  border-bottom-right-radius: 20px;  
  background-color:#ddd;  
  width:200px;  
  padding:10px;  
}
```

Recuadro 1

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aenean commodo ligula eget dolor. Aenean massa. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Donec quam felis, ultricies nec,



Sombras (box-shadow)

La propiedad box-shadow permite definir una sombra a un objeto de la página. Debemos definir tres valores y un color, por ejemplo:

```
#recuadro1{  
  box-shadow: 30px 10px 20px #aaa;  
  border-radius: 20px;  
  background-color:#ddd;  
  width:200px;  
  padding:10px;  
}
```

Recuadro 1

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aenean commodo ligula eget dolor. Aenean massa. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Donec quam felis, ultricies nec,



Sombras (box-shadow)

La propiedad tiene 3 valores y un color, los valores son los siguientes:

- El desplazamiento horizontal de la sombra, positivo significa que la sombra se encuentra a la derecha del objeto, un desplazamiento negativo pondrá la sombra a la izquierda.
- El desplazamiento vertical, uno negativo la sombra será en la parte superior del objeto, uno positivo la sombra estará por debajo.
- El tercer parámetro es el radio de desenfoque, si se pone a 0 la sombra será fuerte y con color liso, más grande el número, más borrosa será.
- El último valor es el color a aplicar a la sombra.



Sombras múltiples (box-shadow)

Con la propiedad box-shadow podemos aplicar múltiples sombras a un objeto. Para esto debemos aplicar la siguiente sintaxis:

```
box-shadow: [desplazamiento en x] [desplazamiento en y] [desenfoque] [color]
            ,
            [desplazamiento en x] [desplazamiento en y] [desenfoque] [color]
etc.
```

Por ejemplo si queremos que aparezca una sombra de color rojo en la parte superior izquierda y una sombra verde en la parte inferior derecha podemos aplicar lo siguiente:



Sombras múltiples (box-shadow)

```
#recuadro1{
  box-shadow: -30px -30px 20px #f00,
              30px 30px 20px #0f0;
  border-radius: 20px;
  background-color:#ddd;
  width:200px;
  padding:10px;
}
```

Recuadro 1

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aenean commodo ligula eget dolor. Aenean massa. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Donec quam felis, ultricies nec,

Sombras interiores (box-shadow)

Otra posibilidad de la propiedad box-shadow es la de implementar la sombra interior al objeto, para esto debemos anteceder a los valores la palabra inset. Por ejemplo si queremos un recuadro con sombra interior de color rojo y sombra exterior de color amarilla podemos aplicar los siguientes valores:

```
#recuadro1{
  box-shadow: inset 0px 0px 40px #f00,
              0px 0px 40px #ff0;
  border-radius: 20px;
  background-color:#ddd;
  width:200px;
  padding:10px;
}
```

Recuadro 1

Lorem ipsum dolor sit amet,
consectetur adipiscing elit.
Aenean commodo ligula eget
dolor. Aenean massa. Cum
sociis natoque penatibus et
magnis dis parturient montes,
nascetur ridiculus mus. Donec
quam felis, ultricies nec,



Sombras con transparencias (box-shadow)

Otra posibilidad que podemos aplicar a la propiedad box-shadow es definir un valor de transparencia de la sombra. Para esto debemos utilizar la función rgba, ejemplo:

```
box-shadow: 30px 30px 30px rgba(255,0,0,0.3);
```

Mediante la función rgba indicamos el color de la sombra con los primeros tres parámetros (en este ejemplo 255 de rojo, 0 de verde y 0 de azul), luego el cuarto parámetro es el índice de transparencia (donde 0 indica que es totalmente transparente la sombra y 1 es totalmente opaca, podemos utilizar cualquier valor entre 1 y 0)



Sombras con transparencias (box-shadow)

Para confeccionar un recuadro con sombra de color rojo con un índice de transparencia del 50% luego debemos especificarlo de la siguiente manera:

```
#recuadro1{  
  box-shadow: 30px 30px 30px rgba(255,0,0,0.5);  
  border-radius: 20px;  
  background-color:#ddd;  
  width:200px;  
  padding:10px;  
}
```

Recuadro 1

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aenean commodo ligula eget dolor. Aenean massa. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Donec quam felis, ultricies nec,

Recuadro 2



Sombras de texto (text-shadow)

La propiedad text-shadow nos permite definir una sombra a un texto, la sintaxis más común es:

```
Elemento {  
    text-shadow: desplazamientoX desplazamientoY radio-de-desenfocado color;  
}
```

La propiedad tiene 4 valores que son los siguientes:

- El desplazamiento horizontal de la sombra, un desplazamiento negativo pondrá la sombra a la izquierda.
- El desplazamiento vertical, un valor negativo dispone la sombra en la parte superior del texto, uno positivo la sombra estará por debajo del texto.
- El tercer parámetro es el radio de desenfoque, si se pone a 0 la sombra será fuerte y con color liso, más grande el número, más borrosa será.
- El último parámetro es el color de la sombra.



Sombras de texto (text-shadow)

Por ejemplo si queremos que un texto tenga una sombra en la parte inferior a derecha con un pequeño desenfoque de color gris luego debemos implementar el siguiente código:

```
#titulo1 {  
  text-shadow: 5px 5px 5px #aaa;  
}
```

Titulo sombreado

Si queremos que la sombra se disponga en la parte superior izquierda de cada letra luego debemos definir los siguientes valores:

```
#titulo2 {  
  text-shadow: -5px -5px 5px #aaa;
```

Titulo sombreado



Sombras de texto (text-shadow)

Otra sintaxis de text-shadow es aplicar varias sombras al texto, por ejemplo:

```
#titulo3 {  
  text-shadow: 3px 3px 5px #f00,  
              6px 6px 5px #0f0,  
              9px 9px 5px #00f;  
}
```

Titulo sombreado



Opacidad (opacity)

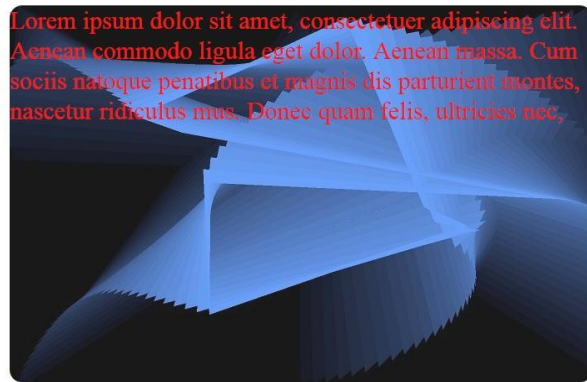
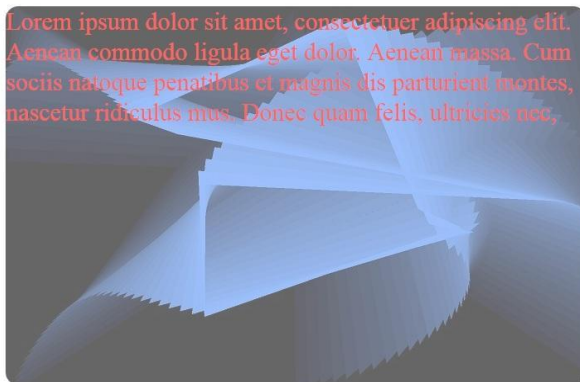
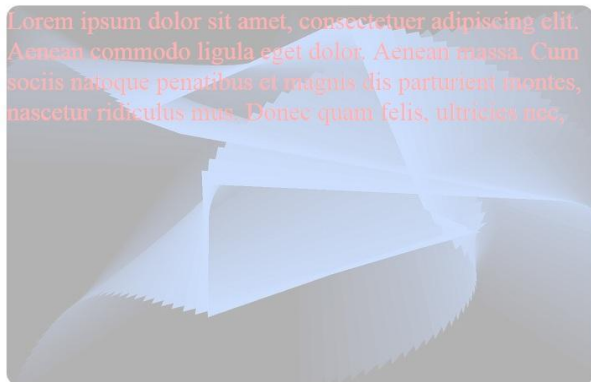
La opacidad es una característica de los objetos de no dejar pasar la luz (mientras un objeto es más opaco significa que no deja pasar la luz) Un elemento HTML dispone de la propiedad opacity para definir cual es su opacidad. La sintaxis es la siguiente:

```
Elemento {  
    opacity: valor;  
}
```

El valor es un número comprendido entre 0 y 1. El 0 significa que es totalmente transparente (luego no se verá nada en pantalla, pero el espacio ocupado por el elemento HTML queda reservado), el 1 significa que es totalmente opaco (no deja pasar la luz)

Opacidad (opacity)

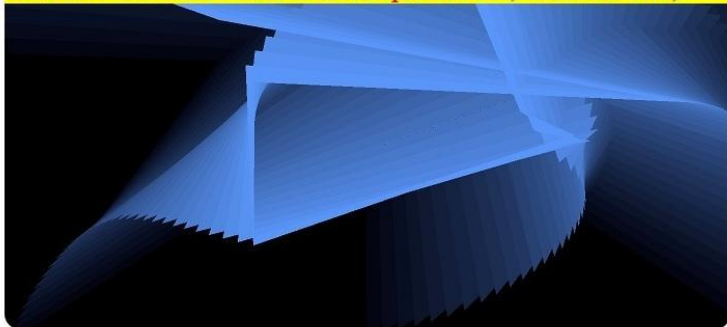
En el ejemplo podemos ver tres recuadro con una imagen de fondo y un texto en su interior con diferentes niveles de opacidad (tengamos en cuenta que cuando le asignamos una opacidad a un elemento HTML luego todos los elementos contenidos en dicho elementos heredan dicha opacidad):



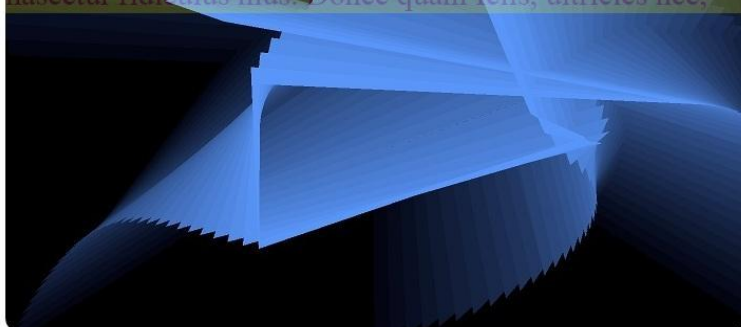
Opacidad (opacity)

Luego tenemos dos cuadros para ver el efecto si no disponemos opacidad en el recuadro y el texto tiene un color de fondo y con opacidad en el texto

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aenean commodo ligula eget dolor. Aenean massa. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Donec quam felis, ultricies nec,



Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aenean commodo ligula eget dolor. Aenean massa. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Donec quam felis, ultricies nec,





Opacidad (opacity)

En CSS3 aparece una variante de la asignación del color mediante la función rgba.

Elemento {

color: rgba(rojo,verde,azul,opacidad);

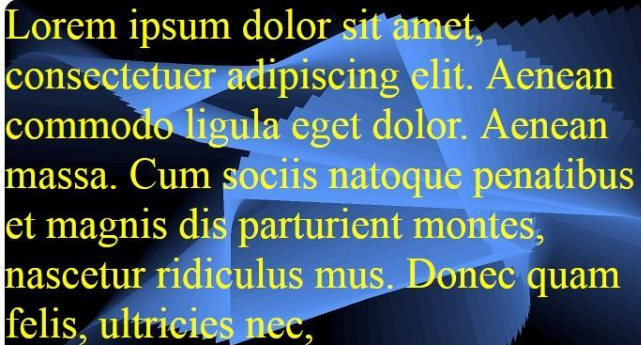
}

Para definir un color debemos indicar cuatro valores, los tres primeros son valores enteros entre 0 y 255, que indican la cantidad de rojo, verde y azul. El cuarto valor es un número entre 0 y 1 que indica la opacidad que se aplica al color. Si indicamos un 1 se grafica totalmente opaco (es el valor por defecto) un valor menor hará más transparente el gráfico.

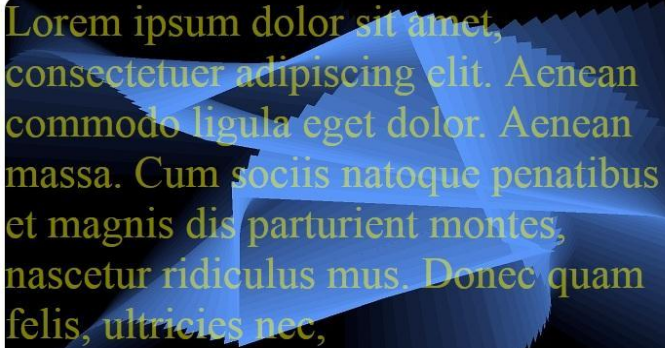


Opacidad (opacity)

Veamos un ejemplo: primero dispondremos un texto dentro de un recuadro que tiene una imagen y no utilizaremos la función `rgba` para definir la opacidad, sino utilizaremos la función `rgb` para definir solo el color. Luego el mismo texto definiendo opacidad



Lorem ipsum dolor sit amet,
consectetur adipiscing elit. Aenean
commodo ligula eget dolor. Aenean
massa. Cum sociis natoque penatibus
et magnis dis parturient montes,
nascetur ridiculus mus. Donec quam
felis, ultricies nec,



Lorem ipsum dolor sit amet,
consectetur adipiscing elit. Aenean
commodo ligula eget dolor. Aenean
massa. Cum sociis natoque penatibus
et magnis dis parturient montes,
nascetur ridiculus mus. Donec quam
felis, ultricies nec,



Múltiples columnas (column-count)

Otra novedad que nos trae CSS3 es permitir disponer un bloque de texto en múltiples columnas con la única indicación de la cantidad de columnas que queremos que lo divida:

```
Elemento {  
    column-count: cantidad de columnas;  
}
```

Por ejemplo para generar un cuadro con tres columnas debemos implementar el siguiente código:

```
#recuadro1{  
    -moz-column-count:3;  
    -webkit-column-count:3;  
    column-count:3;  
    border-radius: 20px;  
    background-color:#ddd;  
    width:600px;  
    padding:10px;  
}
```

Recuadro 1

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aenean commodo ligula eget dolor. Aenean massa. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Donec quam felis, ultricies nec,

Lorem ipsum dolor sit amet,

consectetur adipiscing elit. Aenean commodo ligula eget dolor. Aenean massa. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Donec quam felis, ultricies nec,

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aenean commodo ligula eget dolor. Aenean massa. Cum sociis natoque penatibus et

magnis dis parturient montes, nascetur ridiculus mus. Donec quam felis, ultricies nec,

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aenean commodo ligula eget dolor. Aenean massa. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Donec quam felis, ultricies nec,



Múltiples columnas (column-count)

Es posible que tengamos que anteceder los prefijos -moz, -webkit para que funcionen con el Firefox y Chrome (depende de versión). El Opera ya implementa la propiedad definitiva column-count.

Otra posibilidad es definir la anchura la columna y que dinámicamente se generen las columnas que “quepan”:

```
column-width:200px;
```




Múltiples columnas y línea separadora (column-rule)

Podemos configurar una línea separadora entre las columnas cuando utilizamos múltiples columnas. La sintaxis es la siguiente:

```
Elemento {  
  column-rule: grosor estilo color;  
}
```

Las tres propiedades se pueden asignar de forma independiente:

```
Elemento {  
  column-rule-width: grosor;  
  column-rule-style: estilo;  
  column-rule-color: color;  
}
```

`column-rule-style: none|hidden|dotted|dashed|solid|double|groove|ridge|inset|outset|initial|inherit;`

Múltiples columnas y línea separadora (column-rule)

En los ejemplos podemos ver distintas codificaciones:

Recuadro 1

Lorem ipsum dolor sit amet,
consectetur adipiscing elit.
Aenean commodo ligula eget
dolor. Aenean massa. Cum

consectetur adipiscing elit.
Aenean commodo ligula eget
dolor. Aenean massa. Cum
sociis natoque penatibus et
magnis dis parturient
montes, nascetur ridiculus

magnis dis parturient
montes, nascetur ridiculus
mus. Donec quam felis,
ultricies nec,
Lorem ipsum dolor sit amet,

Recuadro 3

Lorem ipsum dolor sit amet,
consectetur adipiscing elit.
Aenean commodo ligula eget
dolor. Aenean massa. Cum
sociis natoque penatibus et
magnis dis parturient
montes, nascetur ridiculus
mus. Donec quam felis,
ultricies nec,

Lorem ipsum dolor sit amet,

consectetur adipiscing elit.
Aenean commodo ligula eget
dolor. Aenean massa. Cum
sociis natoque penatibus et
magnis dis parturient
montes, nascetur ridiculus
mus. Donec quam felis,
ultricies nec,

Lorem ipsum dolor sit amet,
consectetur adipiscing elit.
Aenean commodo ligula eget
dolor. Aenean massa. Cum
sociis natoque penatibus et

magnis dis parturient
montes, nascetur ridiculus
mus. Donec quam felis,
ultricies nec,
Lorem ipsum dolor sit amet,
consectetur adipiscing elit.
Aenean commodo ligula eget
dolor. Aenean massa. Cum
sociis natoque penatibus et
magnis dis parturient
montes, nascetur ridiculus
mus. Donec quam felis,
ultricies nec,

Recuadro 2

Lorem ipsum dolor sit amet,
consectetur adipiscing elit.
Aenean commodo ligula eget
dolor. Aenean massa. Cum
sociis natoque penatibus et

magnis dis parturient
montes, nascetur ridiculus
mus. Donec quam felis,
ultricies nec,

Lorem ipsum dolor sit amet,

consectetur adipiscing elit.
Aenean commodo ligula eget
dolor. Aenean massa. Cum
sociis natoque penatibus et
magnis dis parturient
montes, nascetur ridiculus
mus. Donec quam felis,

magnis dis parturient
montes, nascetur ridiculus
mus. Donec quam felis,
ultricies nec,
Lorem ipsum dolor sit amet,
consectetur adipiscing elit.

Recuadro 4

Lorem ipsum dolor sit amet,
consectetur adipiscing elit.
Aenean commodo ligula eget
dolor. Aenean massa. Cum
sociis natoque penatibus et
magnis dis parturient
montes, nascetur ridiculus
mus. Donec quam felis,
ultricies nec,

Lorem ipsum dolor sit amet,

consectetur adipiscing elit.
Aenean commodo ligula eget
dolor. Aenean massa. Cum
sociis natoque penatibus et
magnis dis parturient
montes, nascetur ridiculus
mus. Donec quam felis,
ultricies nec,

Lorem ipsum dolor sit amet,
consectetur adipiscing elit.
Aenean commodo ligula eget
dolor. Aenean massa. Cum
sociis natoque penatibus et

magnis dis parturient
montes, nascetur ridiculus
mus. Donec quam felis,
ultricies nec,
Lorem ipsum dolor sit amet,
consectetur adipiscing elit.
Aenean commodo ligula eget
dolor. Aenean massa. Cum
sociis natoque penatibus et
magnis dis parturient
montes, nascetur ridiculus
mus. Donec quam felis,
ultricies nec,



Imágenes de fondo (background-image)

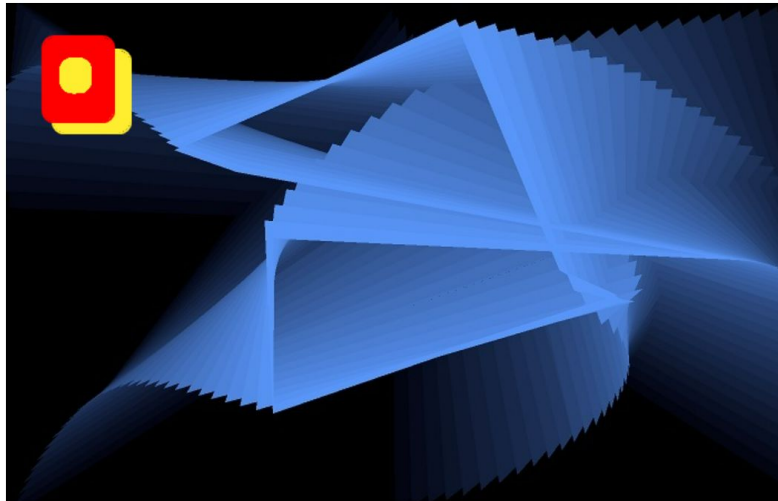
La propiedad background-image permite insertar un conjunto de imágenes dentro de un elemento. Para ello debemos especificar sus nombres:

```
Elemento {  
    background-image: url("imagen1", url("imagen2", ...);  
}
```

La primera imagen que se dibuja es la que indicamos al final de la lista.

Imágenes de fondo (background-image)

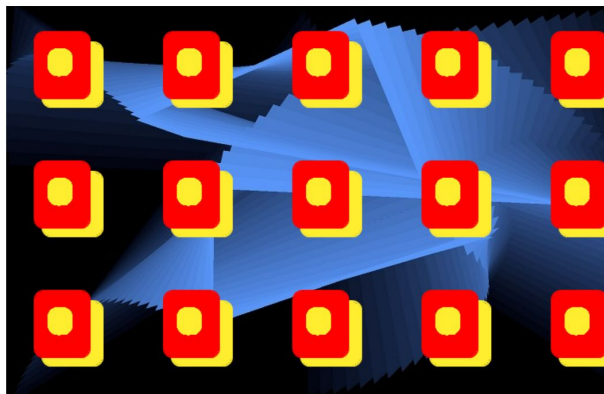
Por ejemplo vamos a mostrar una imagen de fondo con formato jpg y sobre esta una de tipo png, la primera:



Imágenes de fondo (background-image)

El recuadro coincide con el tamaño de la imagen "foto1.jpg" (700*450) Luego la imagen "logo1.png" es de 150*150 píxeles.

Como vemos primero se dibuja la imagen "foto1.jpg" (que es la última) y luego sobre esta la imagen "logo1.png". Otro cosa importante es inicializar la propiedad background-repeat con el valor no-repeat. En caso de no inicializar dicha propiedad tendremos a la imagen logo1.png repetida dentro del recuadro:





Imágenes de fondo en diferentes posiciones (background-position)

Podemos mediante la propiedad background-position indicar la posición de la imagen con respecto al contenedor.

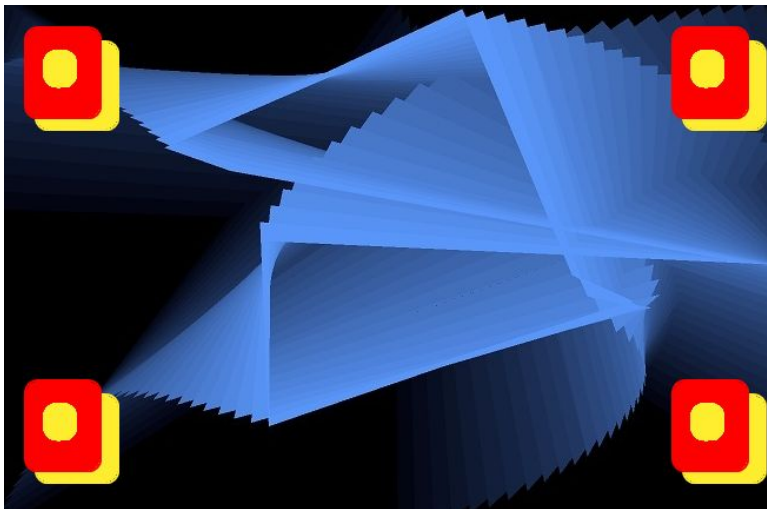
La sintaxis es la siguiente:

```
Elemento {  
    background-image: url("imagen1", url("imagen2", ...);  
    background-position: posx posy , posx posy;  
}
```

El par de valores que indicamos coincide con la posición de las imágenes que especificamos en la propiedad background-image.

Imágenes de fondo en diferentes posiciones (background-position)

En el siguiente ejemplo tenemos una página que muestre una imagen de 700*400 y dentro de esta otra imagen que se ubique en los cuatro vértices de la imagen más grande.





Imágenes de fondo en diferentes posiciones (background-position)

Hay que recordar que la primer imagen que se muestra es la que especificamos última en la propiedad background-image.

Luego en la propiedad background-position indicamos en el mismo orden la ubicación de cada image.

La primera la ubicamos en el vértice superior izquierdo ya que especificamos los valores 0% 0%.

La segunda imagen la especificamos en el vértice superior derecho con los valores 100% 0%.

Así indicamos las otras dos imágenes.

La quinta imagen no es necesario indicar el background-position ya que el ancho y alto coincide con los valores asignados a las propiedades width y height.



Imágenes de fondo (background-size)

Esta propiedad nos permite escalar una imagen dispuesta en el fondo. La sintaxis es:

```
Elemento {  
    background-size: ancho alto;  
}
```

Estas longitudes se las puede especificar en píxeles, porcentajes etc.

Imágenes de fondo (background-size)

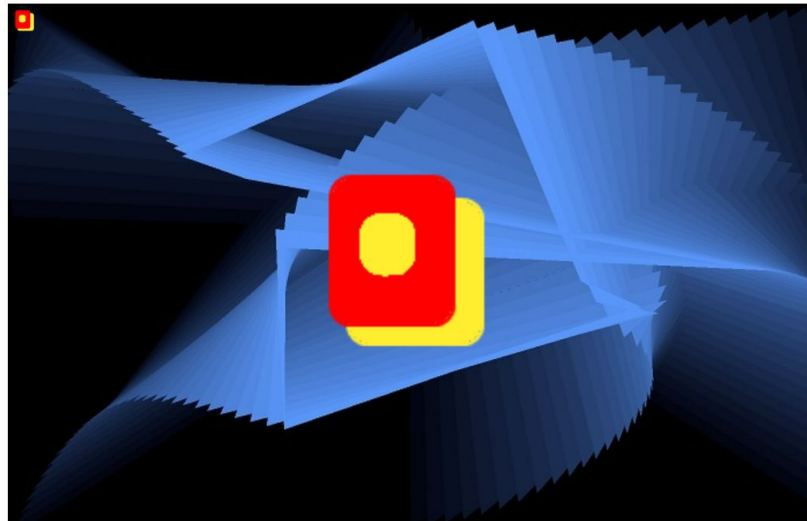
Dibujamos tres imágenes "foto1.jpg" aparece en el fondo y luego dos veces mostramos "logo1.png" y le damos como tamaño al primero 30*30 píxeles y al segundo 250*250.

```
#recuadro1{
  background-image:    url("logo1.png"),
                      url("logo1.png"),
                      url("foto1.jpg");

  background-position: 0% 0%,
                      50% 50%;

  background-size:     30px 30px,
                      250px 250px,
                      700px 450px;

  background-repeat: no-repeat;
  width:700px;
  height:450px;
}
```





Imágenes de fondo (background-size)

Si queremos que una imagen tome el tamaño por defecto que tiene la imagen sin reescalar podemos utilizar la palabra clave "auto", por ejemplo para mostrar la primer imagen con el tamaño original podemos escribir:

```
#recuadro1{
  background-image:    url("logo1.png"),
                      url("logo1.png"),
                      url("foto1.jpg");

  background-position: 0% 0%,
                      50% 50%;

  background-size:      30px 30px,
                      250px 250px,
                      auto auto;

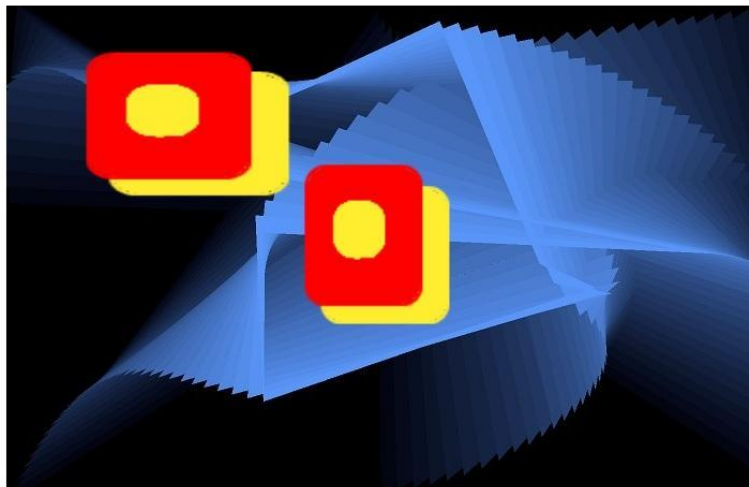
  background-repeat: no-repeat;
  width:700px;
  height:450px;
}
```

Imágenes de fondo (background-size)

Si utilizamos como unidad porcentajes, la dimensión se basa en el elemento contenedor.

Así un ancho y un alto de 50%, por ejemplo, hará que el fondo de la imagen llenar el recipiente 1/4:

```
#recuadro1{  
  background-image:    url("logo1.png"),  
                      url("logo1.png"),  
                      url("foto1.jpg");  
  
  background-position: 0% 0%,  
                    50% 50%;  
  
  background-size:     50% 50%,  
                    250px 250px,  
                    auto auto;  
  
  background-repeat: no-repeat;  
  width:700px;  
  height:450px;  
}
```





Imágenes de fondo (background-origin)

La propiedad background-origin establece el punto donde se comienza a dibujar el fondo. Pudiendo ser en el borde, la almohadilla o el contenido (border-box, padding-box o content-box)

```
Elemento {  
    background-origin: border-box/padding-box/content-box;  
}
```



Imágenes de fondo (background-origin)

Vemos el ejemplo con los tres valores posibles:



Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aenean commodo ligula eget dolor. Aenean massa. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Donec quam felis, ultricies nec,



Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aenean commodo ligula eget dolor. Aenean massa. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Donec quam felis, ultricies nec,



Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aenean commodo ligula eget dolor. Aenean massa. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Donec quam felis, ultricies nec,



Transformaciones y transiciones

La propiedad **transform** permite realizar diferentes transformaciones en 2D o 3D sobre cualquier elemento (textos, cajas...) de nuestra web. Implementada en los navegadores teniendo que usar los prefijos propios de cada navegador (moz, webkit y o).

Las transformaciones disponibles son rotación cambio de escala, deformación oblicua y desplazamiento.



Transformaciones y transiciones

La **rotación** se puede expresar en grados (deg), radianes (rad), aplicarse a los ejes X e Y por separado (rotateX y rotateY), a ambos por igual (rotate), y especificar un número de grados positivo (en el sentido de las agujas del reloj) o negativo (sentido contrario). Un ejemplo:

```
transform: rotate(-20deg);  
-moz-transform: rotate(-20deg);  
-webkit-transform: rotate(-20deg);  
-o-transform: rotate(-20deg);
```




Transformaciones y transiciones

Los **cambios de escala** se especifican mediante un número entero o decimal que puede ser mayor que cero (ampliación). Menor (reducción) o negativo (reflexión). También podemos aplicar un factor de escala diferente al eje X y al Y (scaleX y scaleY).

```
-moz-transform: scale(2.5);  
-webkit-transform: scale(2.5);  
-o-transform: scale(2.5);  
transform: scale(2.5);
```



Transformaciones y transiciones

La **deformación oblicua** usa el parámetro `shew` y nos permite especificar un ángulo en grados, radianes o gradientes (grad), positivos, negativos (para el sentido, igual que en la rotación) y especificar una deformación común o separada para cada eje:

```
-moz-transform: skew(-30deg);  
-webkit-transform: skew(-30deg);  
-o-transform: skew(-30deg);  
transform: skew(-30deg);
```

http://www.w3schools.com/cssref/css3_pr_transform.asp



Transformaciones y transiciones

La propiedad **translate** nos permite definir un desplazamiento indicando este en una medida concreta o en tanto por ciento, con valores positivos (abajo y hacia la derecha) o negativos (arriba y hacia la izquierda). En este caso el valor de la propiedad puede llevar uno o dos valores aplicando en el primer caso a ambos ejes y en el segundo distinguiendo entre ambos:

```
-moz-transform: translate(100px, -50px);  
-webkit-transform: translate(100px, -50px);  
-o-transform: translate(100px, -50px);  
transform: translate(100px, -50px);
```

Otras transformaciones en diferentes estados de normalización son: matrix, matrix3d, translate3d, translateZ, scale3d, scaleZ, rotate3d, rotateZ y perspective.



Transformaciones y transiciones

La propiedad **transition**, por último, nos va a permitir realizar pequeños efectos de animación sin necesidad de recurrir a javascript. Usamos para ello cuatro propiedades: transitionproperty, transition duration, transitiontimingfunction y transitiondelay (o transition, sin más).

http://www.w3schools.com/cssref/css3_pr_transition.asp

Sobre el ejemplo explicaremos los diferentes parámetros:



Transformaciones y transiciones

```
div.trancision {  
    margin: 10px auto;  
    padding: 10px;  
text-align: center; width: 200px; height: 100px;  
    background-color: pink;  
    border: 5px solid blue;  
  
    -webkit-transition: border 5s ease-out ;  
    -moz-transition: border 5s ease-out;  
    -o-transition: all 5s ease-out;  
    transition: border 5s ease-out;  
}  
  
div.trancision:hover{  
    border-right: 200px solid #9cf;  
}
```



Transformaciones y transiciones

En la primera regla especificamos que la transición afectará al borde (**trancision property**), que durará 5 segundos (**trancision timingfunction**) y que será rápida al principio y más lenta al final (**easeout**). En la segunda regla, que se activará al colocar el ratón encima, hacemos un cambio en el borde del elemento (tanto de tamaño como de color) y la transición realizará la transformación entre el estado original y el final con los datos especificados. Al retirar el ratón la animación se invertirá.

La propiedad puede ser el ancho (width), el alto (height), el color, una lista de propiedades diferentes separadas por coma o all (todas las propiedades que cambian entre uno y otro estado). Los tiempos y retrasos se especifican en segundos (s) o milisegundos (ms). Por último, el timingfunction puede ser constante (linear), de lento a rápido (easein), de rápido a lento (ease o ease out) o lento al principio y al final y más rápido en el centro (easeinout).



Animaciones

Las animaciones CSS3 permiten animar la transición entre un estilo CSS y otro.

Las animaciones constan de dos componentes: un estilo que describe la animación CSS y un conjunto de fotogramas que indican su estado inicial y final, así como posibles puntos intermedios en la misma.

Las animaciones CSS tienen tres ventajas principales sobre las técnicas tradicionales de animación basada en scripts:

- Son muy fáciles de usar para animaciones sencillas, puedes hacerlo incluso sin tener conocimientos de Javascript.
- La animación se muestra correctamente, incluso en equipos poco potentes
- Al ser el navegador quien controle la secuencia de la animación, permitimos que optimice el rendimiento y eficiencia de la misma, por ejemplo, reduciendo la frecuencia de actualización de la animación ejecutándola en pestañas que no estén visibles.



Animaciones - Subpropiedades

Para crear una secuencia de animación CSS, hay que dar estilo al elemento que se quiere animar con la propiedad `animation` y sus sub-propiedades. Con ellas podemos no solo configurar el ritmo y la duración de la animación sino otros detalles sobre la secuencia de la animación.

Las subpropiedades de `animation` son:

- **animation-delay**: Tiempo de retardo entre el momento en que el elemento se carga y el comienzo de la secuencia de la animación.
- **animation-direction**: Indica si la animación debe retroceder hasta el fotograma de inicio al finalizar la secuencia o si debe comenzar desde el principio al llegar al final.
- **animation-duration**: Indica la cantidad de tiempo que la animación consume en completar su ciclo (duración).
- **animation-iteration-count**: El número de veces que se repite. Podemos indicar `infinite` para repetir la animación indefinidamente.



Animaciones - Subpropiedades

- **animation-name:** Especifica el nombre de la regla @keyframes que describe los fotogramas de la animación.
- **animation-play-state:** Permite pausar y reanudar la secuencia de la animación
- **animation-timing-function:** Indica el ritmo de la animación, es decir, como se muestran los fotogramas de la animación, estableciendo curvas de aceleración.
- **animation-fill-mode:** Especifica qué valores tendrán las propiedades después de finalizar la animación (los de antes de ejecutarla, los del último fotograma de la animación o ambos).

http://www.w3schools.com/cssref/css3_pr_animation.asp



Animaciones - Definiendo secuencia

Una vez configurado el tiempo de la animación, necesitamos definir su apariencia. Esto lo haremos estableciendo dos fotogramas más usando la regla **@keyframes**. Cada fotograma describe cómo se muestra cada elemento animado en un momento dado durante la secuencia de la animación.

Desde que se define el tiempo y el ritmo de la animación, el fotograma usa percentage para indicar en qué momento de la secuencia de la animación tiene lugar. 0% es el principio, 100% es el estado final de la animación. Debemos especificar estos dos momentos para que el navegador sepa dónde debe comenzar y finalizar; debido a su importancias, estos dos momentos tienen alias especiales: from y to.

Además puedes, opcionalmente, incluir fotogramas que describan pasos intermedios entre el punto inicial y final de la animación (se define por ejemplo los estilos al 50% o al 75%).



Sitios web orientados a diseño

<http://www.csszengarden.com> => dedicado al diseño mediante CSS.

<http://www.openwebdesign.org> => que ofrece plantillas CSS.

<http://css.maxdesign.com.au> => que ofrece tutoriales en línea

<http://browsershots.org> => visualiza la página web en cientos de navegadores incluyendo diferentes sistemas operativos.

<http://aprendamoscss.blogspot.com.es/2012/07/menu-desplegable-con-css.html> => menú desplegable.

<http://www.noupe.com/essentials/freebies-tools-templates/100-great-css-menu-tutorials.html> => 100 tipos de menús.