

# Transformaciones, transiciones y animaciones de CSS3



---

Gracias por compartir:

---

En este tutorial vamos a ver las transformaciones de CSS3, así como las transiciones y animaciones de CSS3. Las transformaciones de CSS3 permiten modificar la visualización de los elementos de una página web tanto en 2D como en 3D. Con las transiciones de CSS3 podemos crear exactamente esto, un efecto de transición, de los elementos en la pantalla. Con las **animaciones de CSS3** podemos crear muchos efectos con movimiento sin necesidad de [JavaScript](#), uso de frameworks o de la librería jQuery, de aquí su potencia. La combinación de HTML5 con CSS3 y en algunos casos JavaScript (por ejemplo, para el elemento de dibujo canvas), podremos crear sitios de la potencia de Flash, de aquí el declive de esta tecnología en web no optimizada para SEO y su paso a la creación de aplicaciones y juegos para móviles.

Si quieres practicar algunas de las propiedades que aquí veremos, te recomiendo que te descargues gratis esta [plantilla de HTML5 con animaciones de CSS3](#).

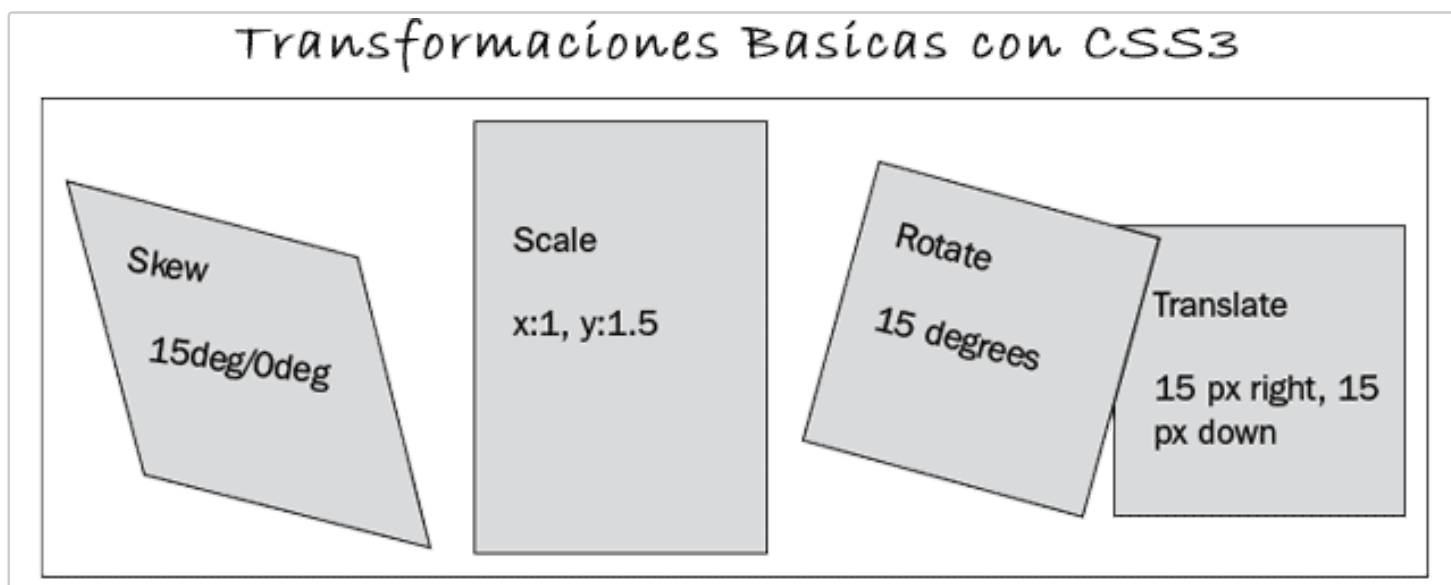
---

# TRANSFORMACIONES, TRANSICIONES Y ANIMACIONES DE CSS3

## LAS TRANSFORMACIONES DE CSS3

Las transformaciones de CSS3 pueden ser en 2D y en 3D, siendo estas últimas considerablemente más complejas al tener que tener en cuenta los tres ejes de coordenadas. En este tutorial veremos las transformaciones en 2D. Si te interesan las transformaciones de CSS3 en 3D, [te recomiendo este artículo](#) (en inglés). En cualquier caso, en el apartado animaciones de CSS3 de este artículo veremos un ejemplo en 3D.

En este ejemplo podemos ver en qué consisten las transformaciones de CSS3 de rotar, escalar y distorsionar. También hay otras muy usadas como la de trasladar.



De las transformaciones de CSS3 en 2D, las más usadas son:

- **Rotate.** Rotate te permite rotar un elemento dándole un ángulo de giro en grados.

- **Scale.** Scale te permite escalar un elemento, toma valores positivos y negativos y se le pueden poner decimales.
- **Translate.** Translate nos permite trasladar un elemento a la vez en el eje de las X y de las Y, dándole las coordenadas iniciales y finales.

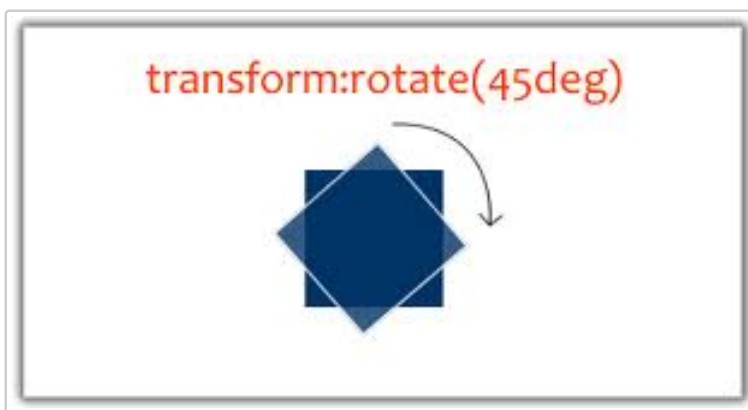
Si no queremos que se aplique ninguna transformación, la propiedad será de none

```
1 .ejemplo {  
2   transform:none;  
3 }
```

---

## LAS ROTACIONES DE CSS3

Como hemos visto, la propiedad de transformación de CSS3 tiene muchas aplicaciones, una de ellas la de rotar un elemento. Se puede aplicar tanto a elemento inline como a elementos de bloque.



```
1 .ejemplo {  
2   -webkit-transform:  
   rotate (45deg);  
3   -moz-transform:  
   rotate (45deg);  
4   -o-transform:  
   rotate (45deg);
```

```
5  -ms-transform:
6  rotate (45deg);
7  transform: rotate
   (45deg);
   }
```

Para no estar perdiendo el tiempo creando los prefijos propietarios de navegador, os recomiendo la [herramienta gratuita Prefixr](#)

Fíjate que los grados se marcan en positivo si el elemento se rota en el sentido de las agujas del reloj y en negativo si es al revés.

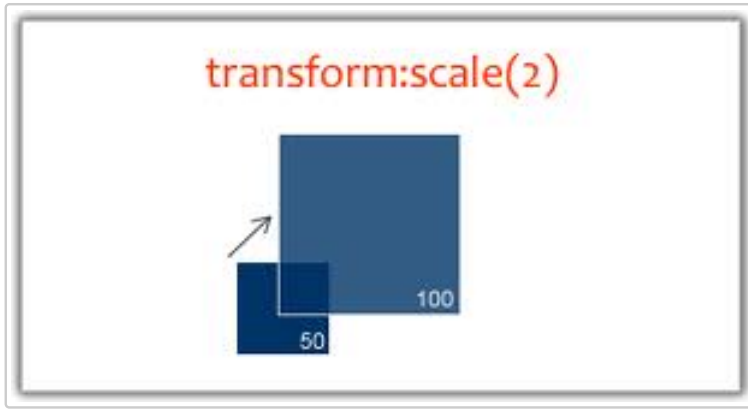
Por defecto, el punto de referencia que toma como eje para hacer la rotación es el centro, pero también se puede especificar otro punto. Aquí un ejemplo en que se traslada el eje de rotación a la parte superior izquierda del elemento:

```
01  -webkit-transform-
02  origin: left top;
03  -moz-transform-
04  origin: left top;
05  -o-transform-
06  origin: left top;
07  -ms-transform-
08  origin: left top;
09  transform-
10  origin: lefttop;
11  -webkit-transform:
12  rotate (-20deg);
13  -moz-transform: rotate
14  (-20deg);
15  -o-transform: rotate
16  (-20deg);
17  -ms-transform: rotate
18  (-20deg);
19  transform: rotate
20  (-20deg);
```

La gracia de estas propiedades es que con las *animaciones de CSS3* vamos a poder crear efectos visuales muy atractivos.

## ESCALAR UN ELEMENTO CON CSS3

Otro punto interesante va a ser escalar un elemento con CSS3.



Si el valor es positivo, el elemento se hace más grande, y si es negativo se hace más pequeño. En el siguiente ejemplo, se escalará el elemento multiplicando por dos:

```
1 -webkit-transform:
  scale(2);
2 -moz-transform:
  scale(2);
3 -o-transform:
  scale(2);
4 -ms-transform:
  scale(2);
5 transform: scale(2);
```

También es posible que queramos dar dos valores diferentes según las X y las Y, con lo que creamos un efecto de escala no proporcional que nos da un efecto visual de distorsión.

```
1 -webkit-transform:
  scale (.5, 2);
2 -moz-transform: scale
  (.5, 2);
3 -o-transform: scale
  (.5, 2);
```

```
4 -ms-transform: scale  
  (.5, 2);  
5 transform: scale  
  (.5, 2);
```

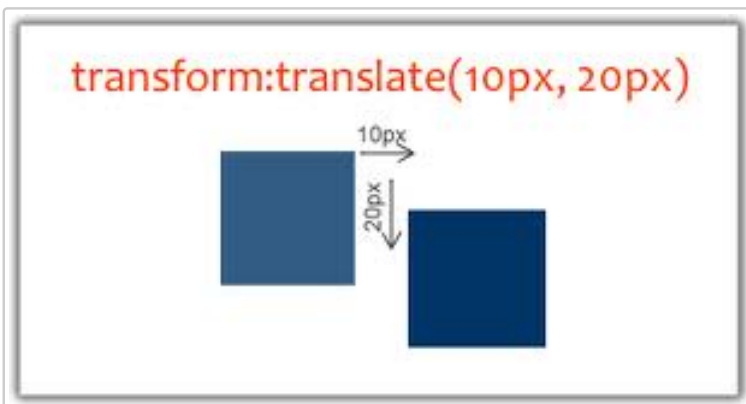
Si lo deseamos, podemos aplicar el cambio de escala sólo en una dirección. Utilizamos `scaleX` para sólo escalar en el eje de las X y `ScaleY` para sólo escalar en el eje de las Y:

```
1 -webkit-transform:  
  scaleY(3);  
2 -moz-transform:  
  scaleY(3);  
3 -o-transform:  
  scaleY(3);  
4 -ms-transform:  
  scaleY(3);  
5 transform: scaleY(3);
```

---

## TRASLADAR ELEMENTOS CON CSS3

Este nombre puede llevar a confusión, ya que esta propiedad de CSS3 no hace propiamente una transición, sino que hace que un elemento pase de estar en una posición a estar en otra. Con las transiciones y las animaciones de CSS3 vamos a poder darle un efecto visual de movimiento.



```
1 -webkit-transform:
  translate(10px, 20px);
2 -moz-transform:
  translate(10px, 20px);
3 -o-transform:
  translate(10px, 20px);
4 -ms-transform:
  translate(10px, 20px);
5 transform:
  translate(10px, 20px);
```

---

## COMBINAR TRANSFORMACIONES

como bien te puedes imaginar, la potencia de las transformaciones, transiciones y animaciones de CSS3 está en combinarlas. Veamos un ejemplo que combina varias transformaciones de CSS3:

```
01 -webkit-transform:
    scale(2, 1);
02 -moz-transform:
    scale(2, 1);
03 -o-transform:
    scale(2, 1);
04 -ms-transform:
    scale(2, 1);
05 transform:
    scale(2, 1);
06
07 -webkit-transform:
    translate(10px, 20px);
08 -moz-transform:
    translate(10px, 20px);
09 -o-transform:
    translate(10px, 20px);
10 -ms-transform:
    translate(10px, 20px);
11 transform:
    translate(10px, 20px);
```

---

# LAS TRANSICIONES DE CSS3

Para activar una transición es necesario que se detecte un evento. Por ejemplo con :hover, :active o :focus. Si combinamos las transiciones con las transformaciones de CSS3, daremos un efecto de movimiento.

Veamos un ejemplo:

```
01 #nav a{
02     background-
03     color:red;
04 }
05 #nav a:hover,
06 #nav a:focus{
07     background-
08     color:blue;
09     -webkit-transition-
10     property:background-
11     color;
12     -moz-transition-
13     property:background-
14     color;
15     -o-transition-
16     property:background-
17     color;
18     -ms-transition-
19     property:background-
20     color;
21     transition-
22     property:background-
23     color;
24     -webkit-transition-
25     duration:2s;
26     -moz-transition-
27     duration:2s;
```



```
17 -o-transition-  
18 duration:2s;  
19 -ms-transition-  
20 duration:2s;  
21 transition-  
22 duration:2s;  
23 }
```

En este ejemplo estamos indicando que la propiedad de transición a variar es el color del background y que se tome 2 segundos en hacer progresivamente el cambio de color. Las transiciones de CSS3 pueden tomar estos valores:

- transition-property
  - transition-duration
  - transition-delay
  - transition-timing-function
- 

## LISTA DE VALORES QUE PUEDE TOMAR TRANSITION-PROPERTY

A continuación te dejo una lista con todas las propiedades de las transiciones de CSS3.

- background-color
- background-image
- background-position
- border-bottom-color
- border-bottom-width

- border-color
- border-left-color
- border-left-width
- border-right-color
- border-right-width
- border-spacing
- border-top-color
- border-top-width
- border-width
- bottom
- color
- crop
- font-size
- font-weight
- grid-\*
- height
- left
- letter-spacing
- line-height
- margin-bottom
- margin-left
- margin-right

- margin-top
- max-height
- max-width
- min-height
- min-width
- opacity
- outline-color
- outline-offset
- outline-width
- padding-bottom
- padding-left
- padding-right
- padding-top
- right
- text-indent
- text-shadow
- top
- vertical-align
- visibility
- width
- word-spacing
- z-index

- zoom

## DURACIÓN DE LA TRANSICIÓN DE CSS3

Con `transition-duration` especificamos el tiempo que va a tardar la transición. Este tiempo se puede expresar tanto en segundos como en milisegundos. Así, 2.4s y 2400ms tendrán una misma duración.

Imaginemos que tenemos un elemento con el id `#desplazamiento` que queremos mover en la pantalla con las transiciones de CSS3. El modo en que lo haríamos sería el siguiente:

```
01 #desplazamiento a {
02   position: absolute;
03   left: 0;
04
05   -webkit-transition-
06     property: left;
07   -moz-transition-
08     property: left;
09   -o-transition-
10     property: left;
11   -ms-transition-
12     property: left;
13   transition-
14     property: left;
15
16   -webkit-transition-
17     duration: 2s;
18   -moz-transition-
19     duration: 2s;
20   -o-transition-
21     duration: 2s;
22   -ms-transition-
23     duration: 2s;
24   transition-
25     duration: 2s;
26 }
27
28 #desplazamiento
29 a:hover {
```

```
19 | left:15px;  
20 | }
```

Aquí lo que sucede es que el enlace que contenga el ID desplazamiento se desplazará 15px cuando el usuario pase el ratón por encima. Fíjate que en donde especificamos el valor para el evento, en este caso :hover, sólo decimos el cambio que se hará y el resto de propiedades se especifican en la etiqueta normal. Date cuenta que hemos de especificar dos cosas, el transition-property o la propiedad de transformación y la duración de la misma con transition-duration.

## RETRASO EN LA TRANSICIÓN DE CSS3

Con la propiedad transition-delay podemos indicar un tiempo de espera antes de empezar a realizar la transición. Esta propiedad puede tomar tanto valores positivos como negativos. En el caso que le demos un valor negativo, lo que sucede es que la transición tiene lugar en el momento de accionar el evento, lo que empieza en el punto que habría empezado en caso de haberlo hecho en el tiempo indicado. Así, si indico un retraso de -3s y tengo una animación de 10s, en el momento de empezar no lo hará en el segundo 0, sino en el segundo 3.

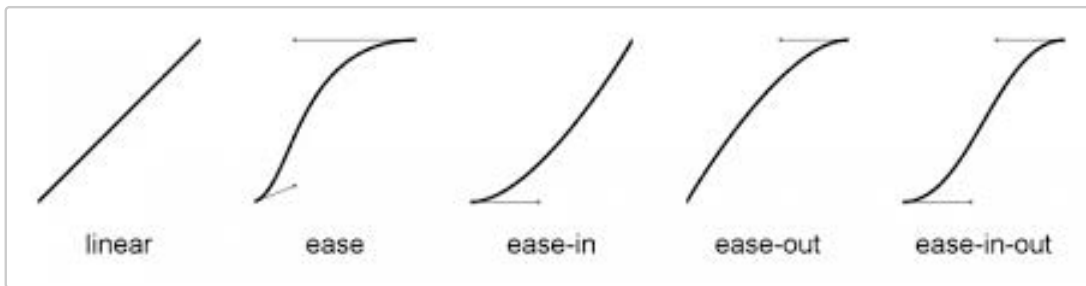
En el caso de IE y Opera no aceptan valores comprendidos entre los -10ms y los 10ms.

## VELOCIDAD DE LA TRANSICIÓN DE CSS3

La propiedad transition-timing-function especifica la velocidad en que se desarrollará la transición, utilizando las llamadas curvas de Bezier. Así, podemos programar la transición para que empiece a una velocidad, luego frene y después se acelere o darle el movimiento que creamos conveniente.

Este punto es quizás el más complejo y existen varias herramientas online para generar las [curvas de Bezier](#). Podemos programarlas nosotros, o darle uno de estos valores: linear, ease, ease-in, ease-

out y ease-in-out. Las curvas de Bezier de estos valores son los que siguen:



No voy a entrar en las matemáticas detrás de una curva de Bezier, recomiendo que vayas probando o bien utilices una herramienta online que haga el trabajo por ti.

Para más información de las transiciones de CSS3 te recomiendo este [excelente artículo de Smashing Magazine](#). Aquí sólo una cosa más, al igual que con muchas otras propiedades de CSS, podemos poner varios valores en una sola línea. Veamos un ejemplo:

```
1 .elemento {  
2   height: 100px;  
3   transition:  
4   height 2s linear;  
5 }  
6 .elemento:hover {  
7   height: 200px;  
8 }
```

## LAS ANIMACIONES DE CSS3

**Las animaciones de CSS3 son la verdadera potencia del CSS3 en acción.** Veamos ahora cómo creamos animaciones de CSS3.

# LA CREACIÓN DE ANIMACIONES EN CSS3

Una animación tiene lugar en el tiempo, con lo que vamos a tener que tomar diferentes puntos dentro de un fragmento de tiempo para especificar que sucede en cada uno de ellos. Es lo que se llaman **keyframes**. Para los que habéis trabajado en Flash u otros programas de animación ya conocéis el concepto. En cada keyframe podemos introducir un cambio y la suma de estos cambios es lo que va a dar lugar a la animación final.

Para indicar la duración de una animación utilizaremos la propiedad `animation-duration`.

La velocidad de la animación estará gestionado por la propiedad `animation-timing-function` en la que podremos especificar la curva de Bezier de la animación, tal y como sucedía en las transiciones de CSS3.

También en las animaciones de CSS3 podremos especificar un tiempo de espera antes de iniciar la animación con la propiedad `animation-delay`.

Con la propiedad `animation-direction` podremos indicarle que la propiedad se haga en sentido inverso al habitual, es decir, que empiece por el final.

También podemos especificar la de veces que se repetirá la animación con la propiedad `animation-iteration-count`.

Resumiendo, las propiedades de las animaciones son:

- `animation-name`
- `animation-duration`
- `animation-timing-function`
- `animation-delay`

- `animation-iteration-count`
- `animation-direction`

Para ahorrar tiempo, te recomiendo [animate.CSS](#), con muchas animaciones de CSS3.

Para muchos ejemplos de animaciones de CSS3 en 2D te recomiendo este fantástico [artículo de Smashing Magazine](#)

---

## HERRAMIENTAS ONLINE

[CSS3 Maker](#)

[CSS3 Generator](#)

[WestCIV Tools](#)