



APRENDERAPROGRAMAR.COM

ANIMACIONES CSS.  
TRANSITION-PROPERTY,  
TRANSITION-DURATION,  
TRANSITION-TIMING-  
FUNCTION Y TRANSITION-  
DELAY. EJEMPLOS.  
(CU01063D)

Sección: Cursos

Categoría: Tutorial básico del programador web: CSS desde cero

Fecha revisión: 2029

**Resumen:** Entrega nº63 del Tutorial básico: "CSS desde cero".

Autor: César Krall

## MÁS EFECTOS CSS

Combinando la propiedad transform con hover podemos crear efectos dinámicos sobre una web. Sin embargo, la animación que se puede conseguir con este efecto es bastante limitada. CSS ha introducido nuevas técnicas que permiten elaborar animaciones más sofisticadas que pueden colaborar a no tener que recurrir a tecnologías como Javascript ó Flash.



## CONCEPTO DE TRANSICIÓN CSS

Una transición CSS es un efecto de cambio progresivo desde un estado inicial hasta un estado final de un elemento HTML. En el estado inicial el elemento tiene unas propiedades y en el estado final tiene una o varias propiedades que han cambiado. El navegador se encarga de generar los pasos intermedios que suponen cambiar de un estado a otro, de la forma que especifiquemos a través del código.

Una transición CSS empieza cuando se produce una situación de “disparo” del cambio. El disparo del cambio puede ser generado de distintas maneras, pero la forma más básica y la que vamos a ver en los ejemplos que pondremos a continuación es que el cambio empieza cuando el usuario pone el ratón encima del elemento HTML afectado por la transición, es decir, que usamos la pseudoclase :hover para definir la transición que empieza cuando el usuario pone el puntero del mouse encima del elemento.

Una transición empieza y termina y no se repite indefinidamente (a no ser que el usuario vuelva a poner el ratón encima del elemento otra vez), aunque veremos más adelante que CSS ofrece más posibilidades para generar animaciones.

Mediante transiciones CSS podemos generar efectos dinámicos que anteriormente sólo se podían generar recurriendo a otros lenguajes o tecnologías. Constituyen un recurso interesante, pero a la vez peligroso, en el sentido de que se puede caer en la tentación de crear páginas web con gran cantidad de efectos y movimientos pero que resulten demasiado sobrecargadas y poco útiles para los usuarios.

## PROPIEDAD TRANSITION-PROPERTY

Esta propiedad puede no ser reconocida por los navegadores antiguos o requerir del uso de prefijos específicos para algunos navegadores actuales. Consulta en Mozilla Developer Network para conocer si debes aplicar prefijos.

CSS permite cambiar las propiedades de un elemento HTML mediante una transición con efecto de animación progresiva. El primer parámetro a definir es qué propiedad es la que se va a cambiar durante la animación. Por ejemplo, podemos cambiar la altura ó la anchura de un elemento. También se pueden cambiar varias propiedades simultáneamente, por ejemplo la altura, la anchura y el color o la imagen de fondo, tamaño de texto, etc. Para definir qué propiedad o propiedades van a ser incluidas en la animación se usa la propiedad transition-property.

PROPIEDAD CSS transition-property	
<b>Función de la propiedad</b>	Permite definir qué elementos van a ser afectados por una transición.
<b>Valor por defecto</b>	all
<b>Aplicable a</b>	Todos los elementos.
<b>Valores posibles para esta propiedad</b>	all (indica que se verán afectadas por la transición todas las propiedades que se pueda)
	none (indica que no se aplicará la transición a ninguna propiedad)
	nombreDeLaPropiedad (indicando un nombre de propiedad se especifica la propiedad que se va a ver afectada; si son varias, separarlas con comas.)
	inherit (se heredan las características del elemento padre).
<b>Ejemplos aprenderaprogramar.com</b>	.myContainer { transition-property: color, transform; } .myContainerSP { transition-property: font-size; }

## PROPIEDAD TRANSITION-DURATION

Esta propiedad puede no ser reconocida por los navegadores antiguos o requerir del uso de prefijos específicos para algunos navegadores actuales. Consulta en Mozilla Developer Network para conocer si debes aplicar prefijos. transition-duration sirve para especificar cuál tiene que ser la duración de la transición en segundos (s) o en milisegundos (ms). Su valor por defecto es 0 (esto equivale a que no habrá transición). En caso de que haya varias propiedades afectadas, los tiempos se especifican separados por comas y se asocian a cada propiedad en el mismo orden en que los define la propiedad transition-property. Ejemplo: transition-property: color, transform; transition-duration: 10s, 4s;

En este caso la transición para la propiedad color durará 10 segundos, será más lenta. La transición para la propiedad transform durará 4 segundos, será más rápida.

Otro ejemplo con valores expresados en milisegundos: transition-duration: 120ms, 750ms;

## PROPIEDAD TRANSITION-DELAY

Esta propiedad puede no ser reconocida por los navegadores antiguos o requerir del uso de prefijos específicos para algunos navegadores actuales. Consulta en Mozilla Developer Network para conocer si debes aplicar prefijos.

Esta propiedad sirve para especificar cuál tiene que ser el retardo con el que empiece la transición en segundos (s) o en milisegundos (ms), a partir del momento en que se produzca la situación "de disparo". Su valor por defecto es 0 (esto equivale a que la transición comenzará inmediatamente cuando se produzca la situación de disparo). En caso de que haya varias propiedades afectadas, los tiempos se especifican separados por comas y se asocian a cada propiedad en el mismo orden en que los define la propiedad transition-property. Ejemplo:

transition-property: color, transform; transition-delay: 2s, 0s;

En este caso la transición para la propiedad empezará con un retardo de 2 segundos, no se hará patente inmediatamente. El retardo para la propiedad transform será de cero segundos, lo que significa que se empezarán a apreciar los efectos de la transición inmediatamente una vez se produzca la situación de disparo.

Otro ejemplo con valores expresados en milisegundos: transition-delay: 120ms, 750ms;

### PROPIEDAD TRANSITION-TIMING-FUNCTION

Esta propiedad puede no ser reconocida por los navegadores antiguos o requerir del uso de prefijos específicos para algunos navegadores actuales. Consulta en Mozilla Developer Network para conocer si debes aplicar prefijos.

PROPIEDAD CSS transition-timing-function	
<b>Función de la propiedad</b>	Permite definir la velocidad a la que transcurre la transición respetando la duración establecida. A efectos prácticos, permite que el usuario vea la animación como progresiva a velocidad constante o con cierta aceleración.
<b>Valor por defecto</b>	ease
<b>Aplicable a</b>	Todos los elementos.
<b>Valores posibles para esta propiedad</b>	<b>ease</b> : comienzo rápido, luego velocidad constante y final lento.
	<b>ease-in</b> : al revés que ease.
	<b>ease-out</b> : similar a ease con final a velocidad constante.
	<b>ease-in-out</b> : comienzo lento y progresión a velocidad constante.
	<b>linear</b> : progresión a velocidad constante.
	<b>step-start</b> : salto de la situación inicial a la final al comienzo.
	<b>step-end</b> : salto de la situación inicial a la final al final.
	<b>steps</b> (numeroDePasos, end) [El parámetro numeroDePasos establece el número de pasos intermedios entre la situación inicial y la final, lo que equivale a que la transición se vea como "pequeños saltos".]
	<b>cubic-bezier</b> (valor0a1, valor0a1, valor0a1, valor0a1) [Esta forma de definición permite especificar la curva temporal de forma matemática para definir formas de cambio personalizadas]
<b>inherit</b> (se heredan las características del elemento padre).	
<b>Ejemplos</b> <b>aprenderaprogramar.com</b>	<pre>.myContainer { transition-timing-function:ease;} .myContainerSP { transition-timing-function:steps(6, end);}</pre>

## EJEMPLO DE APLICACIÓN

Escribe el siguiente código y comprueba los resultados (ten en cuenta que para algunos navegadores puede ser necesario que indiques prefijos). El efecto a apreciar es cómo un div con un texto e imagen va rotando y cambiando de color hasta volver a su posición inicial pero con el color cambiado.

```
<html>
<head>
<title>Portal web - aprenderaprogramar.com</title> <meta charset="utf-8">
<style type="text/css">
*{margin:0; padding:0; font-family: sans-serif;}

.bloquePub{ border: 5px solid; width: 260px;
margin:60px 0 0 90px; background-color:yellow;
font-size: 20px; text-align:center;
padding-top: 20px; word-wrap:break-word;
transition-property: color, transform;
transition-duration: 6s, 4s;
transition-timing-function: ease;
transition-delay: 2s, 0s;
}

.bloquePub:hover {color: red; transform:rotate(360deg);}

h2{margin: 15px 0 0 140px;}
img {padding:10px; }
}
</style>
</head>
<body>
<h2>CSS transition</h2>
<div id="mainCont" style="border-style:none; border-width:0; background-color:white;">
<div class="bloquePub" style="border-radius: 40px;">aprenderaprogramar.com</a></div>
</div>
</body>
</html>
```

El mismo efecto se puede conseguir usando la propiedad shorthand transition como veremos a continuación.

## PROPIEDAD SHORTHAND TRANSITION

Esta propiedad puede no ser reconocida por los navegadores antiguos o requerir del uso de prefijos específicos para algunos navegadores actuales. Consulta en Mozilla Developer Network para conocer si debes aplicar prefijos.

Las funciones relacionadas con transiciones transition-property, transition-duration, transition-delay y transition-timing-function se pueden agrupar en una sola propiedad shorthand denominada transition. La sintaxis a emplear es de este tipo:

```
selectorDeElemento { valorTransitionProperty  valorTransitionDuration  
                        valorTransitionDelay   valorTransitionTimingFunction}
```

Para aplicar esta propiedad en el ejemplo visto anteriormente reemplazaríamos las cuatro propiedades relacionadas con transition por esta línea:

transition: color 6s ease 2s, transform 4s ease 0s;

El efecto lo vemos descrito en esta secuencia de imágenes:



### LO ELEGANTE Y LO EXCESIVO

Con las propiedades que estamos viendo que permiten efectos dinámicos usando CSS muchos diseñadores y programadores son capaces de generar efectos elegantes y atractivos para páginas web. Sin embargo, muchos otros llenan sus páginas web de complejos efectos visuales que son muy espectaculares, pero luego resultan desagradables para el usuario debido a que en la página todo se está moviendo y dando vueltas, lo que puede llegar incluso a generar mareos. Por tanto, concluimos con una recomendación. Si usas estos efectos, hazlo con prudencia y moderación.

## EJERCICIO

Estudia el siguiente código CSS y responde a las cuestiones planteadas:

```
#transEj1 div { transition:all 2s ease-in-out; perspective: 800px; perspective-origin: 50% 100px; }  
#transEj1:hover #rotateX { transform:rotateX(180deg);}  
#transEj1:hover #rotateY { transform:rotateY(180deg);}  
#transEj1:hover #rotateZ { transform:rotateZ(180deg);}
```

- a) Crea un documento HTML donde se vean los estilos que tenemos en el código aplicados a distintos elementos.
- b) Explica el significado de cada una de las partes del código (por #transEj1 div indica los estilos que se aplicarán a div dentro de elementos con id="transEj1". transition: all indica que ..., 2s indica que ... , ease-in-out indica que..., etc.)

Para comprobar si tu código y respuestas son correctas puedes consultar en los foros [aprenderaprogramar.com](http://aprenderaprogramar.com).

**Próxima entrega:** CU01064D

**Acceso al curso completo** en [aprenderaprogramar.com](http://aprenderaprogramar.com) -- > Cursos, o en la dirección siguiente:  
[http://aprenderaprogramar.com/index.php?option=com\\_content&view=category&id=75&Itemid=203](http://aprenderaprogramar.com/index.php?option=com_content&view=category&id=75&Itemid=203)