

CSS Gráfico

Transiciones, Transformaciones y Animaciones

Transiciones

En CSS existen funcionalidades interactivas con pseudo-clases como `:hover`. Sin embargo, dichas acciones ocurren de golpe, pasando de un estado a otro.

Las **transiciones** nos ofrecen una gran flexibilidad y se basan en un principio muy básico, conseguir un efecto suavizado entre un estado inicial y un estado final.

¿Cómo utilizar las transiciones CSS?

Para crear un efecto de transición, debe especificar principalmente dos cosas:

- La propiedad que va a cambiar con el evento.
- La duración del efecto.

Las propiedades relacionadas:

Propiedades	Valor
transition-property	all none <u>propiedad</u>
transition-duration	0 <u>tiempo</u>
transition-timing-function	ease linear ease-in ease-out ease-in-out cubic-bezier(<u>A</u> , <u>B</u> , <u>C</u> , <u>D</u>)
transition-delay	0 <u>tiempo</u>

Transition-property

Especifica la **propiedad a la que afectará la transición**. Podemos especificarla concreta (por ejemplo **width o color**) o simplemente especificar **all** para que se aplique a todos los elementos. Por otro lado, **none** hace que no se aplique ninguna transición.

Nota: Debes saber que no todos los elementos permiten animación debido a su complejidad.

Transition-Duration

Con la propiedad transition-duration especificaremos la **duración de la transición**

Transition-Timing-Function

Permite indicar la **variación de velocidad en una transición**, haciendo que sea más rápida al principio y más lenta al final, o viceversa, entre otras posibilidades.

Los valores que puede tomar la propiedad son los siguientes:

Valor	Inicio	Transcurso	Final	Equivalente en cubic-bezier
ease	Lento	Rápido	Lento	(0.25, 0.1, 0.25, 1)
linear	Normal	Normal	Normal	(0, 0, 1, 1)
ease-in	Lento	Normal	Normal	(0.42, 0, 1, 1)
ease-out	Normal	Normal	Lento	(0, 0, 0.58, 1)
ease-in-out	Lento	Normal	Lento	(0.42, 0, 0.58, 1)
cubic-bezier(<u>A</u> , <u>B</u> , <u>C</u> , <u>D</u>)	-	-	-	Transición personalizada

Transition-delay

Ofrece la posibilidad de **retrasar el inicio de la transición** los segundos o milisegundos especificados.

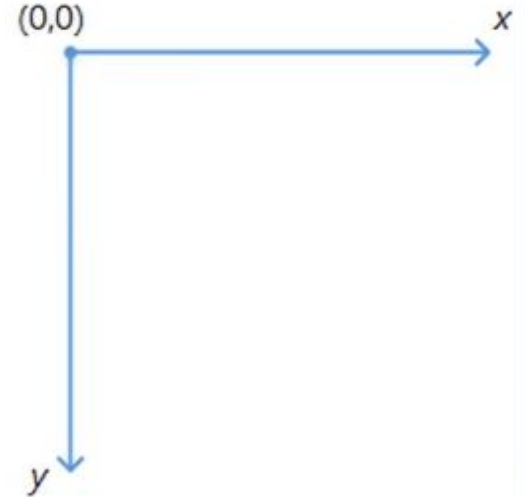
Atajo:

Podemos resumir todas estas operaciones en una propiedad de atajo denominada transition. El orden de sus parámetros es el siguiente:

transition: <property> <duration> <timing-function> <delay>

Transformaciones

Son capaces de realizar todo tipo de efectos visuales

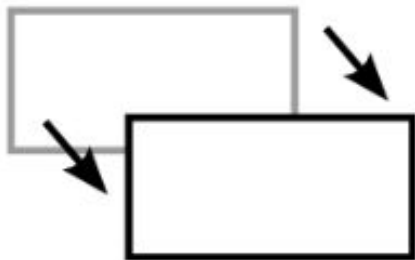


Propiedades	Formato	Significado
transform:	<u>función1</u> , <u>función2</u> , ...	Aplica una o varias funciones de transformación sobre un elemento.

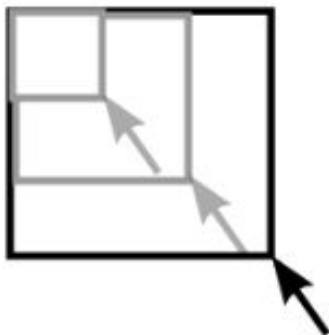
Funciones de transformación

Existen múltiples propiedades CSS que ofrecen diferentes funcionalidades de transformación en dos dimensiones, que veremos a continuación:

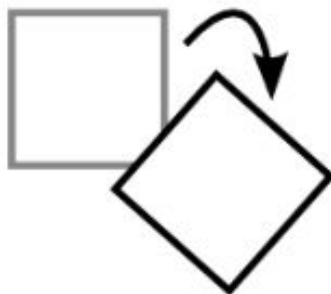
translate()



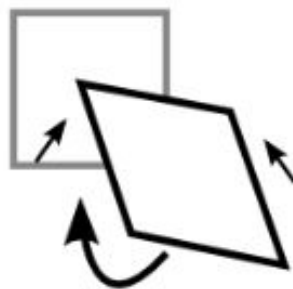
scale()



rotate()



skew()



Animaciones

Una animación permite a un elemento de cambiar gradualmente de un estilo a otro.

Para ello necesitamos Keyframes:

Un **keyframe** es una herramienta que permite a los desarrolladores controlar los pasos intermedios en una secuencia de animación.

Estructura:

```
@keyframes <nombre_de_la_animacion> {  
    [ from | <percentage> ] [ to |  
    <percentage> ]  
}
```

```
@keyframes mymove {  
    from {top: 0px;}  
    to {top: 200px;}  
}
```

```
@keyframes example {  
    0%   {background-color: red;}  
    25%  {background-color: yellow;}  
    50%  {background-color: blue;}  
    100% {background-color: green;}  
}
```

0-100%

from (same as 0%)

to (same as 100%)

Propiedades de animación

Propiedades	Valor
animation-name	none <u>nombre</u>
animation-duration	0 <u>tiempo</u>
animation-timing-function	ease linear ease-in ease-out ease-in-out cubic-bezier(<u>A</u> , <u>B</u> , <u>C</u> , <u>D</u>)
animation-delay	0 <u>tiempo</u>
animation-iteration-count	1 infinite <u>número</u>
animation-direction	normal reverse alternate alternate-reverse
animation-fill-mode	none forwards backwards both
animation-play-state	running paused

Animation-name

Permite **especificar el nombre** del keyframe a utilizar.

Animation-duration, animation-timing-function y animation-delay

Especifican la **duración, velocidad de cambio** y el **retraso** de una animación.

Animation-iteration-count

Indica el número de veces que se repite la animación, pudiendo establecer un **número concreto de repeticiones** o indicando **infinite** para que se repita continuamente.

Animation-direction

Indicar el orden en el que se reproducen los keyframes, pudiendo escoger un valor de los siguientes:

Valor	Significado
normal	Los fotogramas se reproducen desde el principio al final.
reverse	Los fotogramas se reproducen desde el final al principio.
alternate	En iteraciones par, de forma normal. Impares, a la inversa.
alternate-reverse	En iteraciones impares, de forma normal. Pares, normal.

Animation-fill-mode

Indicar que debe mostrar la animación cuando ha finalizado y ya no se está reproduciendo; si mostrar el estado inicial (backwards), el estado final (forwards) o una combinación de ambas (both).

Animation-play-state

Permite establecer la animación a estado de reproducción (running) o pausarla (paused)

Atajo:

```
animation:    <name>    <duration>    <timing-function>  
<delay>    <iteration-count>    <direction>    <fill-mode>  
<play-state>
```

Ejercicio:

STARWARS

Google-Fonts

Hind Madurai

Indian Type Foundry (5 styles)



The sky was
cloudless and of a
deep dark blue.
