# Transiciones y animaciones

## Transiciones:

Proporcionan una forma de animar los cambios de las propiedades CSS, en lugar de que los cambios surtan efecto de manera instantánea, es decir, las transiciones ocurren en un intervalo de tiempo que se puede especificar.

Propiedades de las transiciones

Las transiciones CSS se controlan mediante la propiedad abreviada **transition .**

En este caso, si se añaden dos tiempos, el primero es la velocidad a la que ocurre la transición y el segundo es el retardo, que es el tiempo que hay que esperar para que empiece la transición.

Se pueden controlar los componentes de forma individual, pero no es lo recomendado. En este caso, se usan las siguientes propiedades:

**Transition-property**: Especifica el nombre de las propiedades CSS a las que debería aplicarse las transiciones.

**Transition-duration**: Especifica la duración en la que sucederán las transiciones.

**Transition -timing-function:** Especifica la curva cúbica Bézier que se usa para definir como se comportan los valores intermedios para las propiedades. Es la velocidad. Existen valores predeterminados, que son:

* Linear: Sigue un ritmo lineal.
* Ease-in: Empieza lento y termina rápido.
* Ease-out: Empieza rápido y termina lento.

**Transition-delay:** Define el tiempo de espera entre el momento en que se cambia una propiedad y el inicio de la transición.

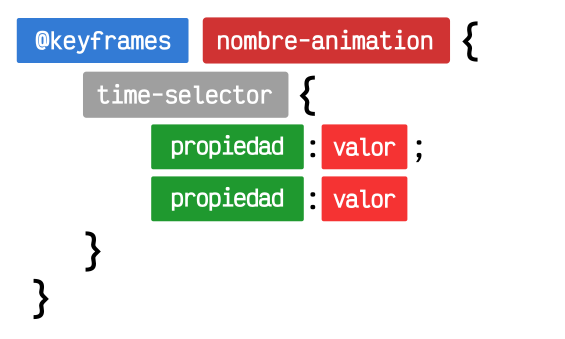
## Animaciones:

Para crear animaciones hay que seguir dos pasos:

1. Utilizar la propiedad **animation** para indicar el elemento HTML que vamos a animar. Tiene distintas propiedades:

* **Animation-name:** Permite especificar el nombre de la animación, para poder utilizarlo con el @keyframes. Distingue entre mayúsculas y minúsculas.
* **Animation-duration:** Si no se indica que son segundos, se interpreta como el número de veces que se repite la animación.
* **Animation-function**
* **Animation-delay:** Retardo en iniciar la animación. Si no se indica que son segundos, se interpreta como el número de veces que se repite la animación.

1. Definir, mediante **@keyframes** la animación en cuestión y su comportamiento. La estructura de los @keyframes es la siguiente:



Cada uno de estos time-selector será un momento clave de cada uno de los fotogramas clave de nuestra animación, y ya veremos que pueden definirse muchos en una misma animación.

Ejemplo:

@keyframes change-color {

0% {

background: red; /\* Primer fotograma \*/

}

50% {

background: yellow; /\* Segundo fotograma \*/

width: 400px;

}

100% {

background: green; /\* Último fotograma \*/

}

}

.animated {

background: grey;

color: #FFF;

width: 150px;

height: 150px;

animation: change-color 2s ease 0s infinite;

}

## Transformaciones.

Las transformaciones se pueden efectuar en CSS mediante la propiedad **transfrom** que permite recibir una función de transformación determinada, la cual será aplicada en el elemento HTML en cuestión, seleccionado mediante CSS. Dicho elemento HTML, se verá transformado visualmente.

Por ejemplo:

Queremos transformar el elemento HTML con la clase element y trasladarlo 25px a la derecha y 150px hacia abajo:

.element {

transform: translate(25px, 150px);

}

Otro ejemplo:

.element {

width: 50px;

height: 50px;

background: grey;

transform: translate(400px, 100px) rotate(50deg);

}

Se traslada 400px en el eje X y 100px en el eje Y.

Escale(): Para escalar. Hace un elemento mas pequeño o mas grande.

Si un elemento tiene dos transform, se ponen seguidos, sin separar por comas, como en el ejemplo anterior, porque si están separados, solo se va a aplicar el último.