Animaciones, construcción

Las **animaciones** se activan ni bien entramos a la página , pero igualmente son fácilmente controlables a través de **JS**. Veamos qué **propiedades** nos permite construir una animación.



animation-name

Propiedad obligatoria, es la única forma de vincular el elemento con la animación. Hace referencia al nombre de la animación, que vamos a generar en la regla, veamos un ejemplo para entenderlo.

```
@keyframes nombreAnimacion {
from {propiedad: valor; propiedad: valor; propiedad: valor;} /*cmo empieza la animación*/
to { propiedad: valor; propiedad: valor; propiedad:valor;} /*como termina la animación*/
}
selector { animation-name: nombreAnimacion;}
```

animation-duration

Es una **propiedad obligatoria**, porque el valor predeterminado es **0** es decir que si no pongo ningún duration la animación nunca a suceder. Hace referencia a cuánto dura la **animación** y sus **valores** posibles pueden ser en **ms o s.**

Cuanto más tiempo le ponemos a la animación más lenta va a ser

```
#panorama {
    position: relative;
    width: 3190px;
    height: 350px;
    animation-name: sky;
    animation-duration: 25s;
    animation-iteration-count: infinite;
    animation-timing-function: linear;
}
```

animation-delay & animation-timing-function

Animation delay, hace referencia a **cuánto tarda la animación en comenzar** y sus valores posibles pueden ser en **ms o s.**

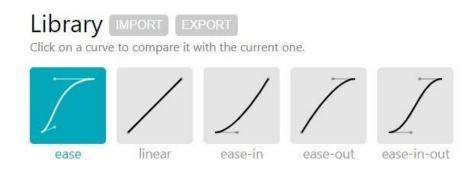
Por otro lado, animation-timing-function es cómo sucede la transición, los valores posibles son:

- ease, valor predeterminado, inicia despacio, sucede rápido y termina despacio
- **linear**, mantiene la transición con estilo de velocidad igual en toda la duración de la misma
- ease-in, inicia despacio, luego sucede normal y termina normal

animation-delay & animation-timing-function

- ease-out,inicia normal, sucede normal y termina despacio
- ease-in-out,inicia despacio, sucede normal y termina despacio

Con el **valor bezier**, podemos indicar de manera personalizada en una curva cuál **va a ser la velocidad**, como es medio complicado hacerlo , para eso tenemos aplicaciones como http://cubic-bezier.com/



animation-play-state

Hace referencia al estado de la **animación,running o paused**. La idea es que si uno pausa la animación la pueda activa con **JS**.

```
<head>
  width: 200px;
  height: 200px;
  border-radius: 20px;
  padding: 20px;
  background: blue;
  animation-name: animacion:
  animation-duration: 3s;
   animation-timing-function: linear;
  animation-delay: 1s;
@keyframes animacion {
from { background: red; width: 300px; transform: rotate(34deg); }
to { background: blue; height: 30px; }
 /head>
body
(div> transiciones </div>
          </body>
```

animation-play-state

Podemos generar una alternativa en los estados de la animación, veamos un ejemplo

```
head>
style
div {
  width: 200px;
  height: 200px;
  border-radius: 20px;
  padding: 20px;
  background: blue;
  animation-name: animacion;
  animation-duration: 6s;
  animation-timing-function: linear;
  animation-delay: 1s;
@keyframes animacion {
0% { background: red; width: 300px; transform: rotate(34deg); }
20% { background: blue; height: 30px; }
50% {background: green;}
100% { background: lightblue; }
 /head
body
<div> transiciones </div>
          </body>
```

animation-play-state

También podemos trabajar con el shorthand,

```
width: 200px;
  height: 200px;
  border-radius: 20px;
  padding: 20px;
  background: blue;
  animation: animacion 3s linear;
@keyframes animacion {
0% { background: red; width: 300px; transform: rotate(34deg); }
20% { background: blue; height: 30px; }
50% {background: green;}
100% { background: lightblue; }
```

animation-direction

Indica la dirección de la animación,

- normal, va del from al to, es decir la dirección normal.
- reverse, va del to al from, es decir al revés.
- Codo a Codo
 alternate, va del to al from , y luego vuelve del from al to.
- alternate-reverse, va del from al to, y luego vuelve del to al from.

animation-fill-mode

Nos permite saber como termina y en qué modo lo hace la animación,

- Backwards,
 No empieza del elemento sin animación sino que lo hace del from, luego termina en el elemento sin animación.
- Forwards,
 Empieza normalmente, pero no vuelve al final al elemento sin animación sino que termina en el to (depende igual la propiedad animation-direction si termina en el from o en el to).
- Normal,
 Empieza el elemento común sin animación, luego va del from al to, y vuelve al elemento sin animación.

animation-fill-mode

Un ejemplo completo de animación incluyendo animation-fill-mode sería,

```
div {
  width: 200px;
  height: 200px;
  border-radius: 20px;
  padding: 20px;
  background: blue;
  animation: animacion 3s linear forwards infinite alternate;
@keyframes animacion {
0% { background: red; width: 300px; transform: rotate(34deg); }
20% { background: blue; height: 30px; }
50% {background: green;}
100% { background: lightblue; }
 /head>
body
<div> transiciones </div>
```