

# Animaciones

## Módulo 05

# Introducción

# Animation-name

# Implementación

## 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

# Implementación

## 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

# Implementación

## animation-delay

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



```
_animacion.scss x  personalizados.html
css > _animacion.scss > #cubo
14
15  #cubo {
16    width: 100%;
17    height: 100%;
18    position: absolute;
19    transform-style: preserve-3d;
20    transform: translateZ(-150px);
21    animation-name: girar;
22    animation-duration: 4s;
23    animation-delay: 2s;
24
25
26  }
27
```



# Animation-timing-function

# Implementación

## animation-timing-function

Es el **ritmo o cómo sucede la animación**, 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

Para conocer más detalles sobre estos valores les recomendamos visitar la página de mozilla,

**<https://developer.mozilla.org/es/docs/Web/CSS/animation-timing-function>**



# Implementación

## animation-timing-function

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/>**



### Library

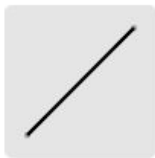
IMPORT

EXPORT

Click on a curve to compare it with the current one.



ease



linear



ease-in



ease-out



ease-in-out

# Implementación

## animation-timing-function

En el ejemplo debajo podemos ver el uso **de tal propiedad**,



```
_animacion.scss x personalizados.html
css > _animacion.scss > #cubo
14
15  #cubo {
16    width: 100%;
17    height: 100%;
18    position: absolute;
19    transform-style: preserve-3d;
20    transform: translateZ(-150px);
21    animation-name: girar;
22    animation-duration: 4s;
23    animation-delay: 2s;
24    animation-timing-function: linear;
25
26  }
```

# Implementación

## Llamar la animación

Si en el caso anterior quisiera crear la animación a ser llamada como vimos en la explicación anterior una opción podría ser de la siguiente forma,



```
@keyframes girar {  
  0% {  
    transform: translateZ(-150px) rotateY(0deg);  
  }  
  100% {  
    transform: translateZ(-150px) rotateY(360deg);  
  }  
}
```

# Revisión

- Repasar los conceptos vistos de **animation**.
- Trabajar con el **bonus track** debajo de todo para practicar los **elementos individualmente**.
- Ver todos los videos y materiales necesarios antes de continuar



# ¡Muchas gracias!

¡Sigamos trabajando!