



# Transiciones y animaciones



## Transiciones y animaciones

Las transiciones y animaciones nos permiten crear movimiento y dinamismo en nuestra aplicación.

Ésto lo podemos lograr cambiando ciertas propiedades CSS de manera gradual durante un tiempo determinado.

Podemos hacer que nuestros elementos cambien de color, tamaño, ubicación y muchas otras cosas más.

**No todas las propiedades CSS son animables**

# Transition

# Transition

La propiedad **transition** acepta cuatro parámetros y puede ser declarada de las siguientes maneras:

**Notas:** transition nos ayuda a definir el comportamiento de la transición, pero es agnóstico a los cambios. Las transiciones necesitan un disparador (por ejemplo, la pseudo-clase :hover)



```
.contenedor {  
  transition: [property] [duration] [ease] [delay]  
}
```



```
.contenedor {  
  transition-property: color;  
  transition-duration: 300ms;  
  transition-timing-function: ease-in;  
  transition-delay: 1s;  
}
```

# Animation



## Animation - Keyframes

Antes de entender la propiedad **animation** tenemos que hablar sobre **@keyframes**.

Los **@keyframes** nos permiten definir las partes de una animación utilizando **porcentajes** o las palabras “**from**” y “**to**” (desde y hasta)

```
@keyframes [name] {  
  from {  
    [styles]  
  }  
  to {  
    [styles]  
  }  
}
```

```
@keyframes [name] {  
  0% {  
    [styles]  
  }  
  100% {  
    [styles]  
  }  
}
```

## Animation - Intro

Si los **@keyframes** son “lo que hace la animación”, entonces la propiedad **animation** lo que define es “cómo lo hace”.

La propiedad **animation** nos va a permitir declarar la **duración**, **delay**, **cantidad de iteraciones** y mucho más.

```
.elemento {  
  animation-name: estirar;  
  animation-duration: 2s;  
  animation-timing-function: ease-in-out;  
  animation-delay: 100ms;  
  animation-direction: normal;  
  animation-fill-mode: normal;  
  animation-play-state: running;  
}
```

```
.elemento {  
  animation: [name] [duration] [timing-function] [delay]  
             [iteration-count] [direction] [fill-mode]  
             [play-state]  
}
```



# Ejemplo 🙌

## Animation - Ejemplo



```
p {  
  animation-duration: 3s;  
  animation-name: slidein;  
}  
  
@keyframes slidein {  
  from {  
    margin-left: 100%;  
    width: 300%;  
  }  
  
  to {  
    margin-left: 0%;  
    width: 100%;  
  }  
}
```

El estilo del elemento `<p>` especifica, a través de la propiedad **animation-duration**, qué la animación debe durar 3 segundos desde el inicio al fin y que el nombre de los **@keyframes** que definen los fotogramas de la secuencia de la animación es **"slidein"**.

## Animation - Ejemplo

Los **fotogramas** se definen usando la regla **@keyframes**. En nuestro ejemplo, tenemos solo dos fotogramas. El primero de ellos sucede en el 0% (hemos usado su alias `from`). Aquí, configuramos el margen izquierdo del elemento, poniéndolo al 100% (es decir, en el borde derecho del elemento contenedor), y su ancho al 300% (o tres veces el ancho del elemento contenedor). Esto hace que en el primer fotograma de la animación tengamos el encabezado fuera del borde derecho de la ventana del navegador.

El segundo (y último) fotograma sucede en el 100% (hemos usado su alias `to`). Hemos puesto el margen derecho al 0% y el ancho del elemento al 100%. Esto produce que el encabezado, al finalizar la animación, esté en el borde derecho del área de contenido.

# LINKS

## LINKS

- **Transitions & animations**

- [https://developer.mozilla.org/en-US/docs/Web/CSS/CSS\\_animated\\_properties](https://developer.mozilla.org/en-US/docs/Web/CSS/CSS_animated_properties)
- <https://developer.mozilla.org/es/docs/Web/CSS/animation>
- [https://developer.mozilla.org/es/docs/Web/CSS/CSS\\_Transitions/Using\\_CSS\\_transitions](https://developer.mozilla.org/es/docs/Web/CSS/CSS_Transitions/Using_CSS_transitions)
- <https://css-tricks.com/almanac/properties/t/transition/>
- <https://www.joshwcomeau.com/animation/css-transitions/>



[gustavgueez](#)



[gustavgueez](#)



[gustavgueez](#)

**GUSTAVO RODRIGUEZ**

FULL STACK DEVELOPER  
SOLCRE