



Css, nos brinda dos posibilidades a la hora de hacer animaciones web: transiciones y animaciones.

## Transiciones CSS

Las transiciones permiten animar los cambios de las propiedades CSS, cuando un elemento cambia de estado y que ese cambio no sea brusco, sino que suceda en un intervalo de tiempo en forma fluida. Por ejemplo, si tenemos un cambio en las propiedades de un vínculo en su estado normal y su estado hover:

```
a{
    color: #fff;
    background-color:#66ccff;
}
a:hover{
    background-color:#ff8d35;
}
```

Esto produciría un cambio brusco en el color de fondo del vínculo, que pasaría de tener un color a otro al pasar el mouse.

Para que este cambio sea fluido, podemos agregarle una transición:

```
a{
    color: #fff;
    background-color:#66ccff;
    transition: all 2s;
}
a:hover{
    background-color:#ff8d35;
}
```



```
}
```

Ahora el cambio, queda animado y esa animación dura 2 segundos.

Para que una transición se dispare necesitamos que haya un cambio de estado en el elemento html, que puede ser:

- :hover
- :focus
- :visited
- :linked

Las propiedades que tienen las transiciones son:

- **transition-property:** sirve para indicar sobre qué propiedad se aplicará la transición. Se pueden especificar varias propiedades separándolas por comas. El valor por defecto es all que indica que la transición se aplicará sobre todas las propiedades que se modifiquen entre un estado y otro.
- **transition-duration:** indica el tiempo de la transición, se expresa en segundos o decimas de segundos separados por puntos. Por ejemplo: 0.5s  
El valor por defecto es 0.
- **transition-delay:** podemos establecer un tiempo de espera (delay) para que comience la transición, una vez que el elemento haya cambiado de estado. Ese tiempo, lo agregaríamos con esta propiedad.
- **transition-timing-function:** indica cómo será la progresión de la transición entre un estado y otro.

El valor por defecto es ease, que lo que hace es suavizar ambos extremos de la transición.

- linear: este tipo de progresión es una progresión mecánica,
  - ease o ease-in-out: la transición comienza y acaba lenta.
  - ease-in: la transición comienza lenta y luego va más rápido. Representa una aceleración
  - ease-out: la transición comienza rápida y termina lenta. Representa una desaceleración.
  - cubic-bezier(n,n,n,n): es un tipo de progresión personalizada.
- **transition:** propiedad abreviada que permite unificar todas las propiedades anteriores en una sola.  
Un ejemplo sería transition: all 2s;



## Animaciones CSS

A diferencia de las transiciones, las animaciones, no dependen de los estados de los elementos HTML para comenzar.

Pueden tener varias posiciones o cuadros claves en el movimiento, denominados keyframes y por ende nos brindan mayor precisión a la hora de hacer una animación más compleja.

Lo primero que tenemos que hacer es crear la animación y establecer los cuadros claves. La animación la declaramos con @keyframes y seguido va el nombre que le asignemos a la animación.

Cada porcentaje representa un cuadro clave. Puedo establecer ilimitados cuadros claves en una animación.

```
@keyframes nombre-animacion {  
    0% {  
        transform: translateX(0px);  
    }  
  
    50% {  
        transform: translateX(400px);  
    }  
  
    100% {  
        transform: translateX(0px);  
    }  
}
```



Una vez creada la animación, podemos invocarla desde cualquier selector css y nuestro elemento html quedará animado:

```
.circulo{  
    animation-name: nombre-animacion;  
    animation-timing-function: ease;  
    animation-duration: 1s;  
    animation-iteration-count: infinite;  
  
}
```

En este ejemplo, los elementos de clase circulo quedaran animados moviéndose a derecha y volviendo a la posición inicial. La animación durará 1 segundo y se reproducirá en forma infinita.

Propiedades de las animaciones:

Las propiedades de las animaciones son bastante parecidas a las de las transiciones. Para poder aplicar animaciones sobre nuestros elementos contamos con las siguientes propiedades:

- **animation-name:** este es el nombre que le hayamos puesto a la animación, cuando establecimos los keyframes. La forma de invocarla es a través de su nombre.
- **animation-duration:** el tiempo que tardará la animación. Se expresa en segundos.
- **animation-timing-function:** indica la curva de progresión de la animación (igual que las transiciones).
- **animation-iteration-count:** indica el número de veces que se repetirá la animación. Su valor por defecto es 1. Podemos incrementarlo o bien si queremos que se reproduzca indefinidamente, establecer el valor "infinite"
- **animation-delay:** indica el retardo con el que se iniciará la animación. Por defecto es 0.
- **animation:** forma abreviada, permite unificar las propiedades anteriores en una sola.  
Por ejemplo:  
animation: nombre-animacion 3s ease 0.5s infinite;



## Recursos:

- Animate Css

<https://daneden.github.io/animate.css/>

Es una galería de animaciones css, que podemos utilizar para animar cualquier elemento.

Simplemente, tenemos que vincular el css a nuestra página y agregarle al elemento, la clase con la animación que querramos, más la clase animated, que indica que queremos que ese elemento se anime.

- Wow js

<https://github.com/matthieua/WOW>

Esta librería js detecta el scroll del navegador y puede combinarse con facilidad con animate.css, disparando las animaciones en el momento que el usuario baja con el scroll por esa sección.

## Referencias:

- W3 Schools Css3 - Animations

[https://www.w3schools.com/css/css3\\_animations.asp](https://www.w3schools.com/css/css3_animations.asp)