**Transiciones**

* **Transition-duration:** Indica cuanto tiempo dura una transición. (La transición no sirve en el hover)
* **Transition-property:** Indica que propiedades se van a animar
* **Transition-timing-function:** determina la velocidad de la transición en diferentes momentos, lo que permite controlar la aceleración y la desaceleración del cambio de propiedades
* **linear:** El cambio de la propiedad es uniforme, lo que significa que la velocidad de la transición es constante.
* **ease:** Comienza lento, se acelera en el medio, y luego vuelve a desacelerarse al final. Es útil para la mayoría de las animaciones suaves.
* **ease-in:** Comienza lento y luego acelera hasta el final. Es bueno para situaciones donde necesitas enfatizar el proceso de aceleración.
* **ease-out:** Comienza rápido y desacelera hacia el final. Es efectivo cuando se quiere una transición que se asiente suavemente.
* **ease-in-out:** Comienza y termina lentamente, pero es más rápido en el medio. Ideal para transiciones que requieren un efecto visible de "aceleración y desaceleración".
* **Steps:** Define cuantos pasos tiene que hacer para terminar la animación
* **Cubic-bezier:** Recibe 4 parametros. Permite definir una curva Bézier cúbica específica. Con cuatro puntos de control, puedes ajustar minuciosamente el comportamiento de la transición. Los primeros dos valores controlan la curva de aceleración, mientras que los últimos dos controlan la desaceleración.
* **Transition-delay:** Es lo que demora en empezar la animación.

**Estados:** se refieren a diferentes formas o condiciones en las que se puede encontrar un elemento HTML en un momento dado, como resultado de la interacción del usuario o de cambios en el entorno de la página. Estos estados se pueden estilizar específicamente usando selectores de pseudo-clases en CSS

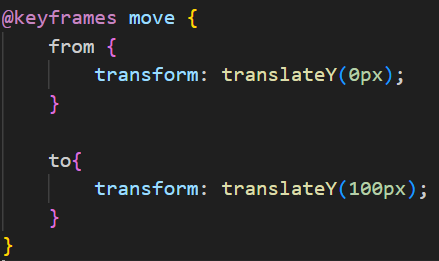
* **Hover:** Se aplica cuando el usuario pasa el cursor sobre un elemento.
* **Active:** Se aplica cuando un elemento está siendo activado por el usuario, típicamente entre el momento que se hace clic y se suelta el clic.
* **Focus:** Ocurre cuando un elemento ha recibido el foco, usualmente a través de la navegación por teclado (por ejemplo, usando la tecla TAB). Este estado es crucial para la accesibilidad, permitiendo a los usuarios de teclado saber en qué elemento se encuentra el foco
* **Checked:** Específico para input de tipo checkbox o radio, este estado indica que el elemento de formulario está seleccionado.
* **Disabled:** Aplica a elementos de formulario que están deshabilitados, lo que significa que no pueden interactuar con el usuario y no se enviarán como parte del formulario.
* **Visited:** Este estado es específico para enlaces (<a>), y se aplica a los enlaces que ya han sido visitados por el usuario. Esto permite diferenciar visualmente entre enlaces vistos y no vistos.

**Animaciones**

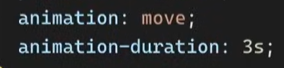
Con los keyframes van a permitir establecer como se va a animar algo.

**Pasos para hacer una animación:**

* Definir los keyframes de un punto inicial a un punto final.

****

* Especificar el tipo de animacion y la duración



* **Animation-timing-function:** Como va a cambiar la animación en el tiempo
* **Animation-iteration-count:** Las veces que se repite la animación
* **Animation-timeline:** Esta propiedad se utiliza para enlazar una animación con una línea de tiempo externa, en lugar de que esta sea controlada por el tiempo transcurrido.
* **Animation-direction:** Dirección a la cual la animación se va a mover
* **Animation-fill-mode:** Esta propiedad controla si los estilos aplicados al elemento objetivo deben persistir después de que la animación haya terminado o si deben revertirse a los estilos definidos en el estado inicial.
* **animation-timeline: view():** Realiza la animacion cuando es visible el elemento