**TRANSFORM:** módulo de CSS que define cómo los elementos diseñados con CSS se pueden transformar en un espacio bidimensional o tridimensional. Permite manipular visualmente un elemento al inclinarlo, rotarlo o escalarlo.

**Sintaxis**: se le pasa como valor la función que queremos que ejecute, así el elemento tiene una escala de 2, aplicado en el lado X e Y. Ej.: transform: function; (transform: scale(2);).

**Funciones**:

• Scale: especifica con uno o dos valores. Representan la cantidad de escala que se aplicará en cada dirección. Scale(x) scale(x, y).

• Skew: especifica con uno o dos valores. Representan la cantidad de sesgo que se aplicará en cada dirección. Si solo especifica un valor, se usa para el eje x y no habrá sesgos en el eje y.

• Rotate: define una transformación que rota un elemento alrededor de un punto fijo en el plano 2D, sin deformarlo. La medida que se utiliza es deg (representa un ángulo en grados).

• Translate: compuesta por un vector bidimensional cuyas coordenadas definen cuánto se moverá el elemento en cada dirección.

• Transform-origin: permite modificar el origen de las transformaciones de un elemento. Podemos pasarle como valor en px, top, bottom, right, left.

**KEYFRAMES**: para crear una animación que se podrá reutilizar. Deberemos indicar cómo serán los diferentes valores, pero esta vez durante toda la animación en los puntos que queramos. Todas las animaciones van desde 0% a 100%, es decir, podemos determinar tanto el inicio, el final, como cualquier otro punto intermedio entre estos dos valores.

1. Crear una animación en sí: para ello vamos a usar la palabra reservada @keyframes y asignarle un nombre. Ej.: @keyframes fantasma {/\* Aquí definiremos los puntos de la animación \*/}
2. Dentro de las llaves: se modifican los valores de determinadas propiedades a lo largo de la animación: de 0 a 100% en los puntos que queramos. Ej.: @keyframes fantasma { 0% { opacity: 0% } 25% { transform: translateY(-20px); opacity: 100% } 100% { opacity: 0% } }
3. Asignando una animación: al selector del elemento se le pueden dar algunas propiedades para la animación:

* *Animation-name*: para definir cuál es la animación que queremos utilizar.
* *Animation-duration*: establece el tiempo que debe durar. El valor puede ser en *seg* (s) o *miliseg* (ms).
* *Animation-iteration-count*: define la cantidad de veces que se repetirá. Puede ser un *nº* o la palabra *infinite* para que se reproduzca infinitamente.

1. Forma abreviada: versión más corta para determinar las diferentes propiedades en una misma línea. Se utiliza la propiedad animation. Ej.: .fantasma {animation: fantasma 3s infinite;}

**TRANSITION**: modificar el valor de una propiedad de manera gradual.

1. Valor inicial: se debe determinar cuál es el elemento que queremos manipular y a qué propiedad queremos agregarle efecto. Ej.: button {background-color: #eaeaea; color: #000}.
2. Interacción: determinar ante qué interacción este elemento va a cambiar el valor de su propiedad.

Ej.: button:hover {background-color: #1a73e8; color: #fff}.

1. Valores de transición: en el selector del elemento agregamos el valor de la propiedad transition. Los valores que podemos utilizar son:

* *Transition-property*: define los nombres de las propiedades CSS en las que el efecto de la transición debe aplicarse. Podemos poner varias separadas por comas.
* *Transition-duration*: establece el tiempo que debe durar la transición. Por defecto, el valor es de 0s o 0ms (no se producirá ninguna animación). Ej.: button {background-color: #eaeaea; color: #000; transition-property: background-color, color; transition-duration: 0.5s;}.

1. Propiedad abreviada: versión más corta para determinar cuáles propiedades y qué duración tendrá el efecto. Se utiliza la propiedad transition. Ej.: button {background-color: #eaeaea; color: #000; transition: background-color 0.5s, color 0.5s;}
2. Muchas propiedades: si queremos crear una transición para todas las propiedades que se modifiquen usamos la palabra reservada *all*. Ej.: button {background-color: #1a73e8; color: #fff; transition: all 0.5s}.