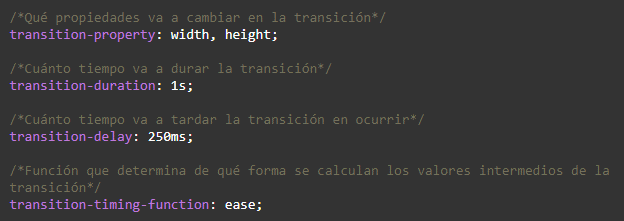


Transiciones en css

Con la propiedad transición podemos agregar una animación simple a un elemento HTML

Estas son las propiedades que se utilizan para realizar la transición:



La transition-timing-functionpropiedad especifica la curva de velocidad del efecto de transición.

También se puede crear en una sola línea agrupando todos sus valores   


Para que una transición ocurra debe existir una interacción del usuario completada en un sitio web.  
  
tengo una duda y espero resolverla con el tiempo, si aplico opacity a un elemento, este elemento desaparece como con display none o sigue ocupando su espacio en pantalla

Le encuentro una utilidad bastante grande a esta propiedad ya que con ella los hover se verán más llamativos para nuestros usuarios



Valores de transition-timing-funciton

facilidad: valor predeterminado. Especifica un efecto de transición con un inicio lento, luego rápido y luego un final lento (equivalente a cubic-bezier (0.25,0.1,0.25,1)).

lineal: especifica un efecto de transición con la misma velocidad de principio a fin (equivalente a cubic-bezier (0,0,1,1)).

easy-in: especifica un efecto de transición con un inicio lento (equivalente a cubic-bezier (0.42,0,1,1))

easy-out: especifica un efecto de transición con un final lento (equivalente a cubic-bezier (0,0,0.58,1)).

easy-in-out Especifica un efecto de transición con un inicio y un final lentos (equivalente a cubic-bezier (0.42,0,0.58,1)).

paso-inicio Equivalente a pasos (1, inicio).

step-end Equivalente a pasos (1, end).

steps (int, start | end) Especifica una función paso a paso, con dos parámetros. El primer parámetro especifica el número de intervalos en la función. Debe ser un número entero positivo (mayor que 0). El segundo parámetro, que es opcional, es el valor "inicio" o "final", y especifica el punto en el que se produce el cambio de valores dentro del intervalo. Si se omite el segundo parámetro, se le asigna el valor "fin".

cubic-bezier (n, n, n, n) Defina sus propios valores en la función cubic-bezier. Los valores posibles son valores numéricos de 0 a 1

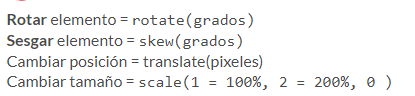
initial Establece esta propiedad en su valor predeterminado. Leer acerca de initial

heredar Hereda esta propiedad de su elemento padre. Leer sobre heredar



Trasformaciones

Sintaxis de Rotate

Rotate: funciona de acuerdo con las manecillas del reloj en valores positivos y en valores negativos van en dirección contraria.

RotateX: solo funciona en el eje X

RotateY: solo funciona en el Eje y

RotateZ: actúa como si el elemento estuviera en 2D y realiza su rotación de la misma manera que rotate.

Rotate3d: rotara en tres dimisiones x,z,y y el ultimo valor serán los grados que van rotar

Trasformación de traslación y perspectiva: Translate nos sirve para poder mover un elemento arriba, abajo, derecha e izquierda.

Utilizamos la propiedad perspective para poder darle profundidad a un elemento y obtener un efecto 3D.

El valor traslate acepta dos valores, el primero es x el segundo y.



También existe el eje z, este establece la profundidad del campo dentro de la animación de nuestro elemento, para poder trabajar con este eje debemos la propiedad perspective.

Es buena practica aplicar las transformaciones y transiciones al elemento padre que contiene nuestros elementos con la animacion



Transformación

Scale:

Esta propiedad no sirve para hacer un escalado de nuestros elementos sin aumentar su tamaño con las propiedades width y heigth los valores que acepta este elemento valores numéricos positivos, desde 0 a 1, donde 0 reducirá su tamaño hasta desaparecer el mismo elemento y uno donde será su valor real.  
sí valor ingresado es mayor a 1 escalará el tamaño real del elemento, por ejemplo, si ingresamos 2 el tamaño del elemento será el doble al original

Skew

Esta propiedad ayuda agregar una animación de segado al elemento, igual que la anteriores su valor valido es en grados además los ejes editables serán el eje “X” y “Y”.  
el orden de los valores ingresado será el siguiente:



podemos especificar que eje vamos a editar agregando “X” o “Y” antes de los paréntesis de acuerdo con lo que deseemos de esta manera especificaremos, en caso contrario de que no especifiquemos y solo agreguemos un valor, el eje transformado será el X.

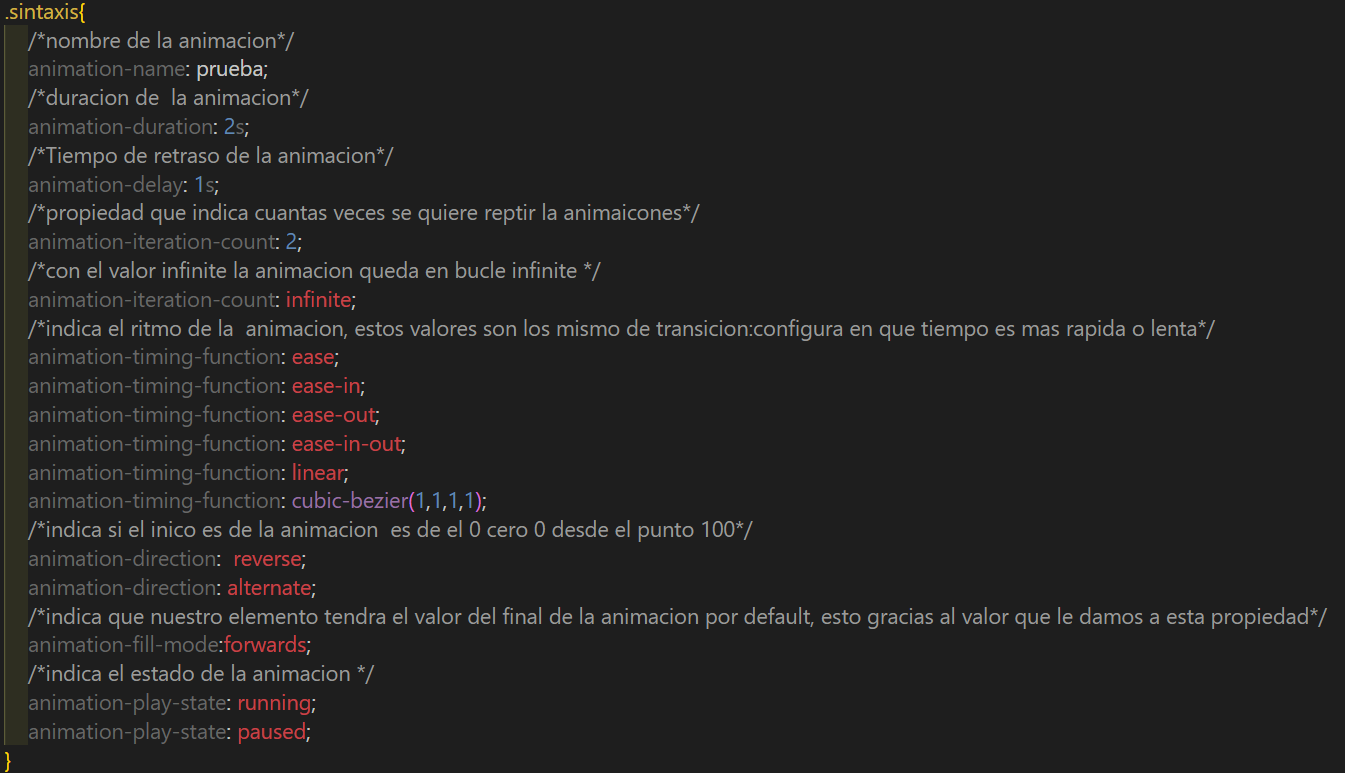
Ya he tenido mis experiencias con este elemento dentro de mis desarrollos y me parece una buena habilidad para ideas o animaciones en css  
Punto de transformación es: es el lugar dentro de viewport donde se inicia la transformación del elemento, por default este se encuentra en el centro del eje “X” y “Y”.



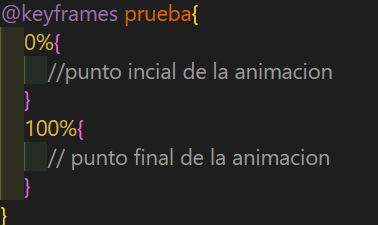
Estas por defecto vienen en el centro.



Animaciones con keyFrames



Estas propiedades estarán dentro del elemento al cual le vamos agregar las animaciones.

esta será la animación como tal, a esta se le pueden agregar los elementos que deseemos que pasen en el tiempo que dure la animación. Además de ello se le pueden agregar puntos intermedios.

En Css3 permite animar algún elemento Estático que tengamos dentro de nuestra página, inicia por un punto cero hasta un punto 100 donde se indica su final



Aceleración y velocidades de una animación además de anidación de animaciones.

A nuestra animación le podemos agregar velocidad o aceleración al momento de su ejecución, eso agregar velocidades para poder agregar mas efecto

ease: por defecto

ease-in: se suaviza al entrar o al inicio

ease-out: se suaviza al salir o al finalizar

ease-in-out: se suaviza al entrar y salir, básicamente se acelera solo la mitad de la animación

steps(frames por segundo): como los fps en los juegos en linea (?), mientras más bajo será menos fluido  
  
estos son algunos efecto que trae css por defecto, pero podemos agregar uno más personalizado con el valor cubic zier , esta página nos ayudara para saber cómo va quedando

https://cubic-bezier.com/

Con estas propiedades le podemos ajustar la velocidad y aceleración que se reproducen la animación.

Además podemos anidar o utilizar mas de una animación dentro de un mismo elemento, teniendo muy presente que para agregar una nueva animación debe ir separada con un coma “,” esto debe ir en cada propiedad “animation” que se agrega en el elemento que vamos a anima. Por ejemplo:



en este caso se agregan dos animaciones, los 5 segundos es la duración de nuestra primera animación, 2s segundos es para la segunda animación. Así como en el ejemplo irán todas las propiedades



Optimización de animaciones con Will change y developer tools, además de las propiedades animables en css y animaciones por vía js

https://developer.mozilla.org/en-US/docs/Web/CSS/will-changeCon los developer tools de Chrome podemos verificar varias cosas en las animaciones con CSS3; podemos incluso probar cambios y optimizar el rendering.

Para usarlo abro el inspector y en los 3 puntos verticales de la esquina superior derecha de la ventana del mismo puedo ir a la pestaña ‘More Tools’ > ‘Animations’ o ‘More Tools’ > ‘Render’. Esto agregará una pestaña nueva para cada herramienta en la misma ventana del ‘console’.

Otra herramienta que puede ser útil desde CSS3 es will-change.  
will-change prepara (avisa/notifica) al navegador para que al renderizar una animación no genere impresiones extra del objeto o imagen que ‘va a cambiar’.

[developer.mozilla.org](https://developer.mozilla.org/es/docs/Web/CSS/will-change) nos advierte que will-change es para ser usado sólo como un último recurso en situaciones de problemas existentes de performance y no para ‘anticipar’ problemas.

Dicho esto, los recursos para los que will-change está optimizado son opacity y transform aunque mozilla agrega [otros](https://developer.mozilla.org/es/docs/Web/CSS/will-change).

Este es el link en el cual estarán todas las propiedades que se le pueden agregar una animación o transformación

[propiedades posibles de animar](https://developer.mozilla.org/en-US/docs/Web/CSS/CSS_animated_properties)

sintaxis animación JS

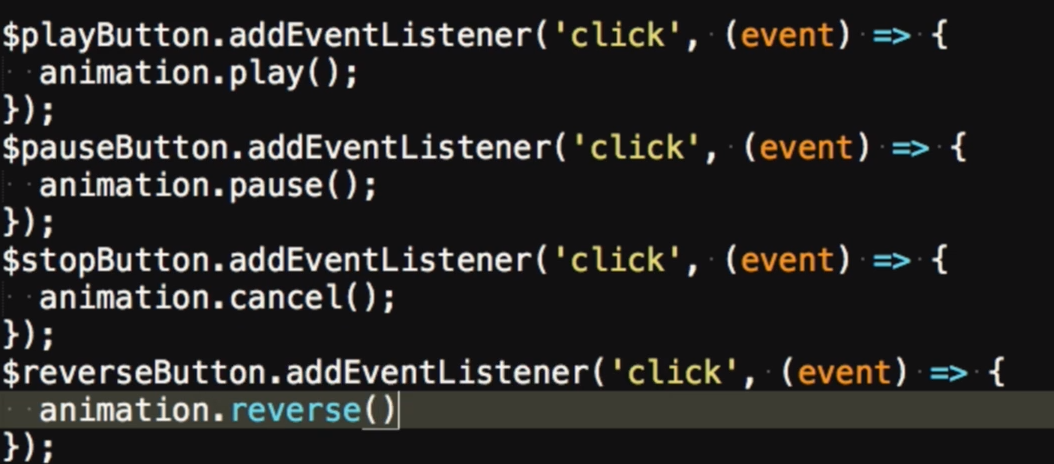


El primer parámetro lleva los keyframe, el segundo las opciones.

para utilizar las animaciones en js utilizamos el método anímateesste método recibe dos parámetros, el primero es un array de objetos, que es el que llevara las propiedades para animar, el segundo parámetro es la objeto con las propiedades que tendrá nuestra animación como duración delay y demás

se puede hacer una sobra interna con box-shadow con el valor reservado inset. Ejemplo





Con los métodos de play, pause, cancel y reverse podemos agregar interacción a nuestra animación

