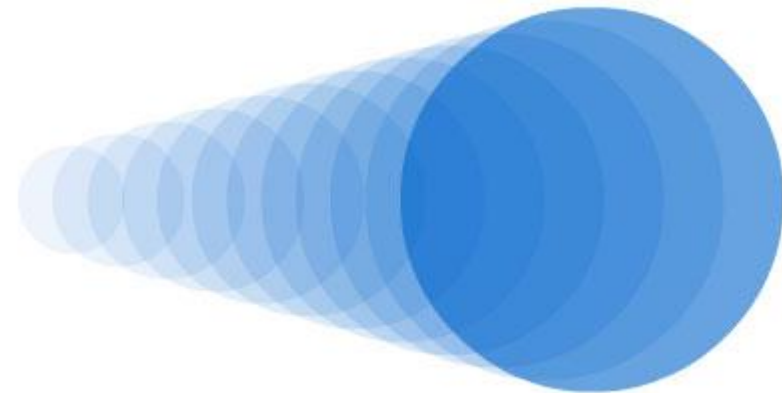
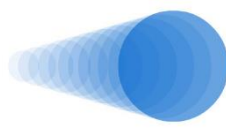


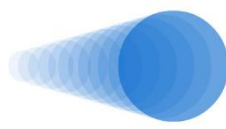
# transition transform



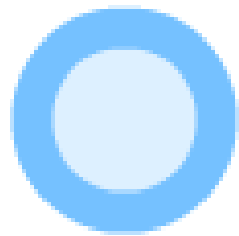


## transition

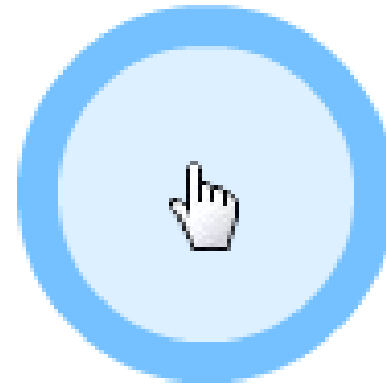
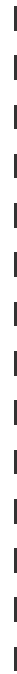
- Permite definir la transición entre dos estados de un elemento.
- Hay diferentes estados que pueden ser definidos utilizando pseudo-clases como `:hover` o `:active` o aplicado dinámicamente usando *JavaScript*.
- Es una propiedad abreviada de `transition-property`, `transition-duration`, `transition-timing-function`, y `transition-delay`.



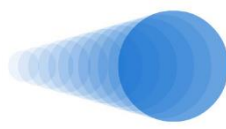
transition



*normal*



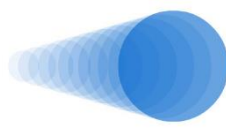
*hover*



**transition-property: all**

Sólo las propiedades que se enumeran aquí son animadas durante las transiciones.

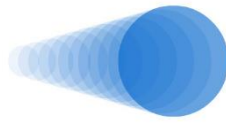
los cambios en el resto de las propiedades suceden de manera instantánea como siempre.



**transition-duration: 0s**

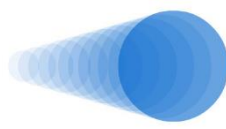
Especifica la duración en la que sucederán las transiciones.

Puedes especificar una única duración que se aplique a todas las propiedades durante la transición o valores múltiples que permitan a cada propiedad de transición un período de tiempo diferente.



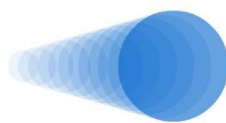
## **transition-timing-function:ease**

Especifica la curva *cúbic-bézier* que se usa para definir cómo se computan los valores intermedios para las propiedades.

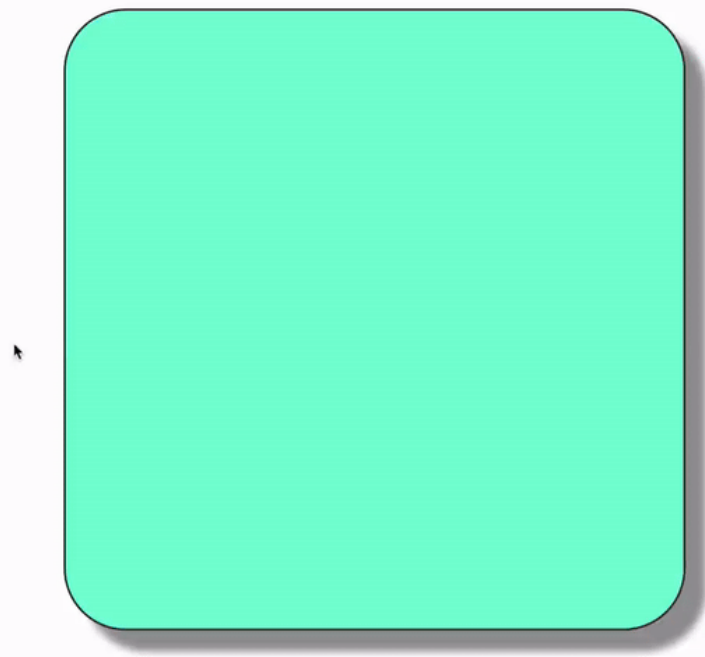


**transition-delay: 0s**

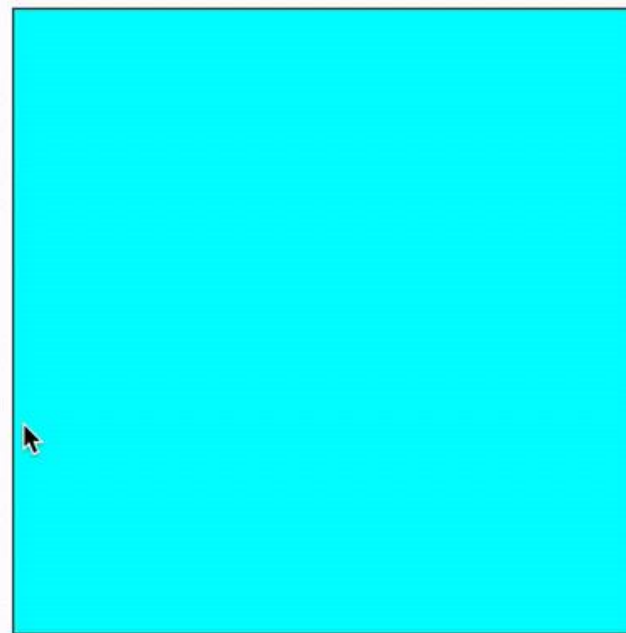
Define el tiempo de espera entre el momento en que se cambia una propiedad y el inicio de la transición.



Sin transition

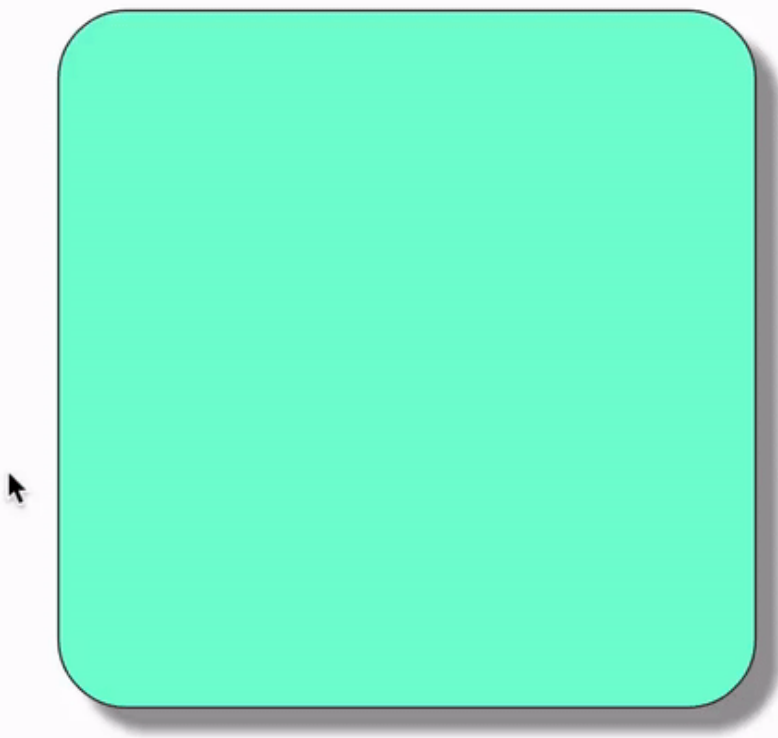
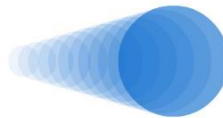


con transition

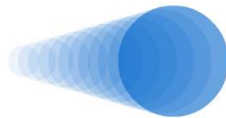






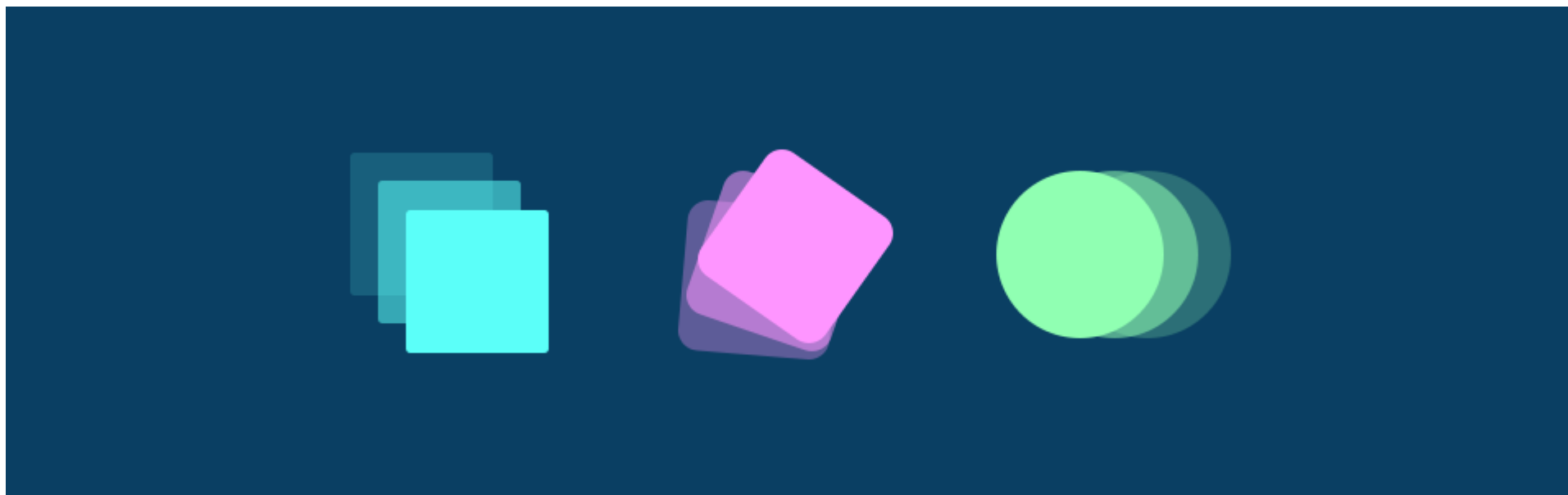
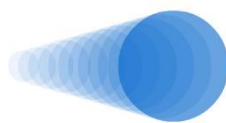


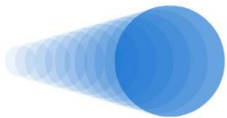
```
.box {  
  width: 300px;  
  height: 300px;  
  background-color: aquamarine;  
  margin: auto;  
  border: 1px solid #333;  
  box-shadow: 10px 10px 5px 0 #999;  
  border-radius: 10%;  
  transition: 1s ease all;  
}  
  
.box:hover {  
  box-shadow: none;  
  border-radius: 0;  
  background-color: cyan;  
}
```



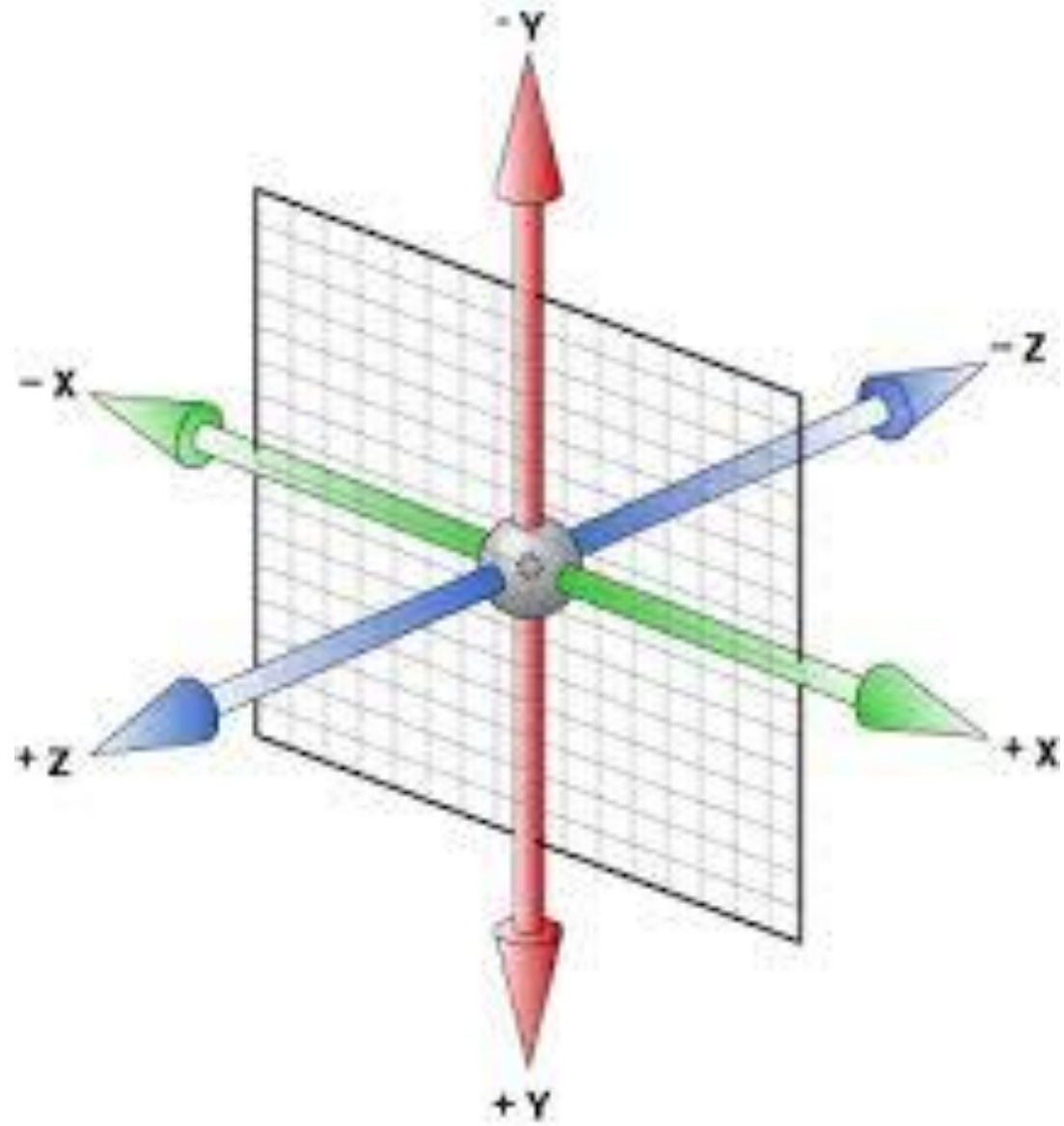
**transition-timing-function**

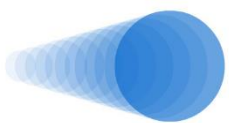
<https://cubic-bezier.com>





- transform
- permite modificar el espacio de coordenadas del elemento.
- Estos pueden ser **trasladados, rotados, escalados o sesgados** de acuerdo a los valores establecidos.



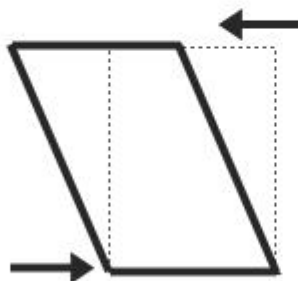


transform

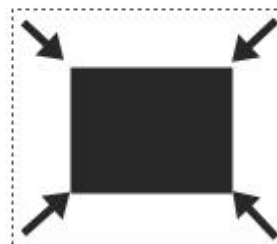
**Rotate**



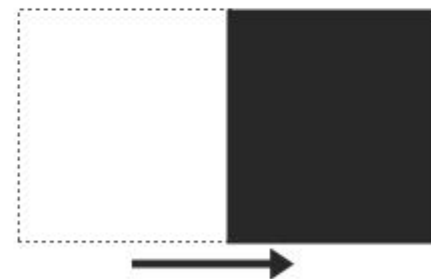
**Skew**

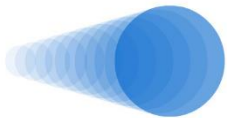


**Scale**



**Translate**



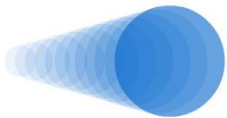


```
/* Function values */  
transform: translate(12px, 50%);  
transform: translateX(2em);  
transform: translateY(3in);  
  
transform: scale(2, 0.5);  
transform: scaleX(2);  
transform: scaleY(0.5);  
  
transform: rotate(0.5turn);  
  
transform: skew(30deg, 20deg);  
transform: skewX(30deg);  
transform: skewY(1.07rad);
```

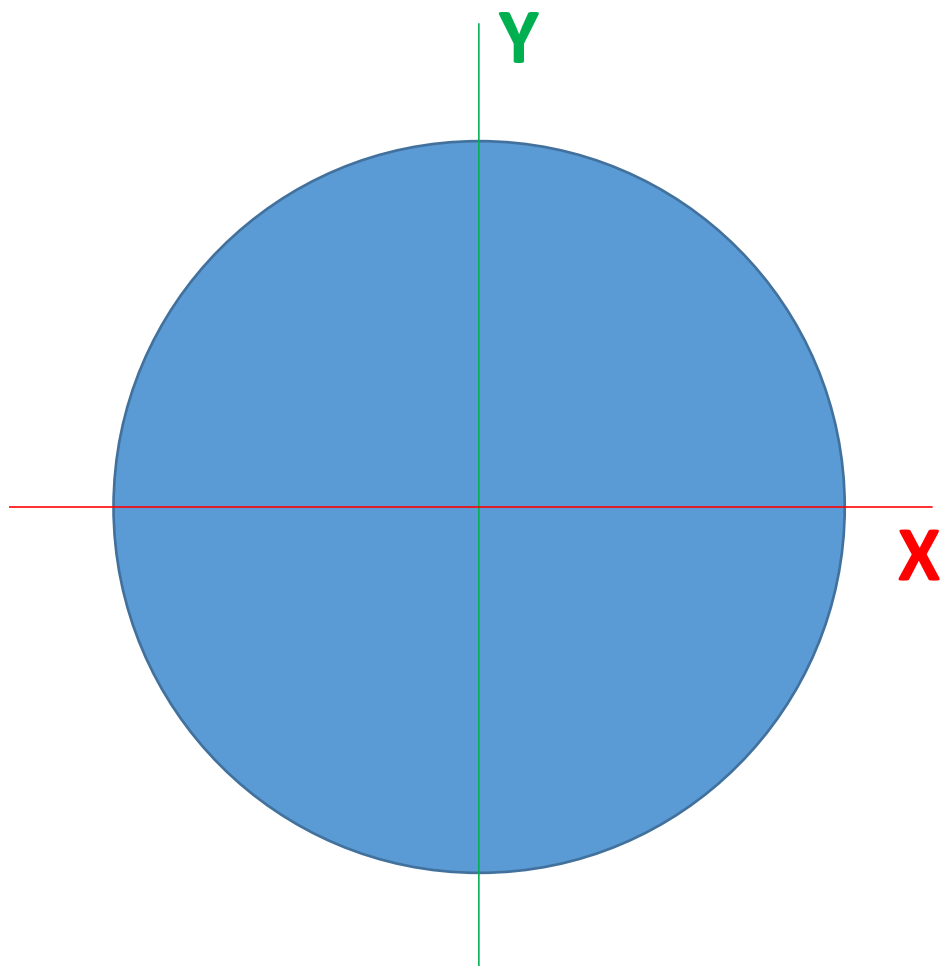




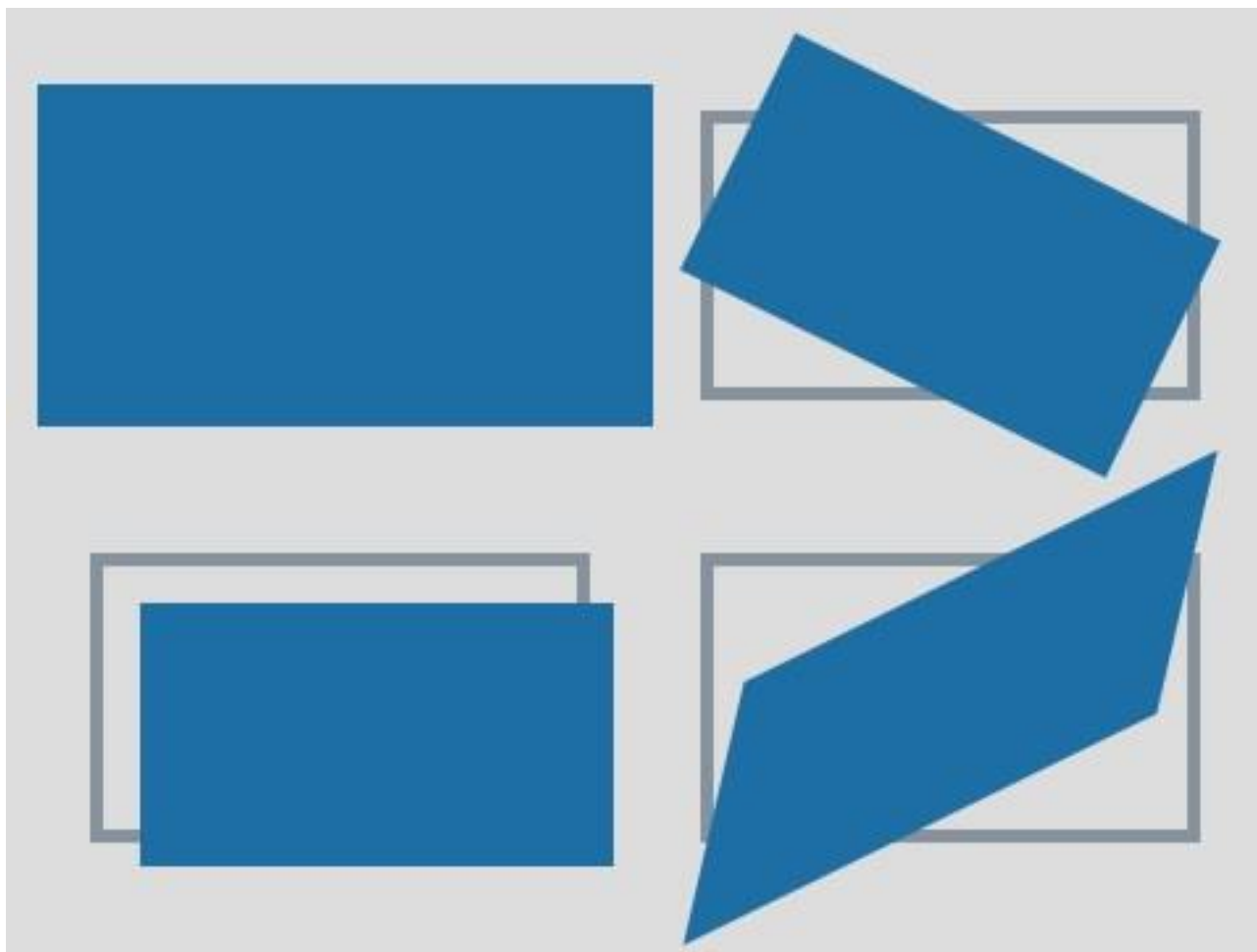
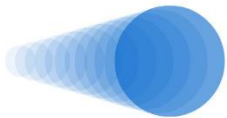
```
/* Function values */  
transform: translate3d(12px, 50%, 3em);  
transform: translateZ(2px);  
transform: scale3d(2.5, 1.2, 0.3);  
transform: scaleZ(0.3);  
transform: rotate3d(1, 2.0, 3.0, 10deg);  
transform: rotateX(10deg);  
transform: rotateY(10deg);  
transform: rotateZ(10deg);  
transform: perspective(17px);  
  
/* Multiple function values */  
transform: translateX(10px) rotate(10deg) translateY(5px);
```



Transform-origin



left	0%
center	50%
right	100%
top	0%
bottom	100%



<https://codepen.io/vineethtrv/pen/XKKEgM>