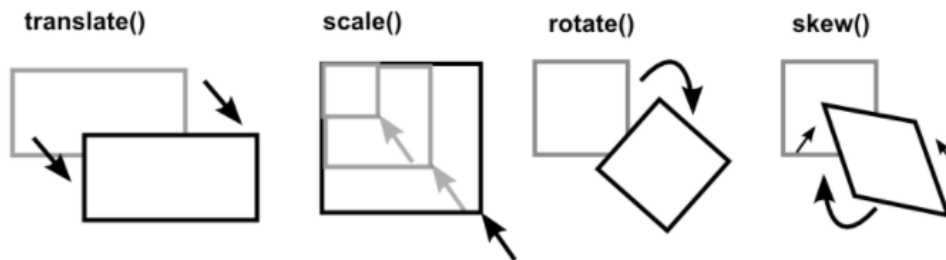


Funciones 2D

Existen múltiples propiedades CSS que ofrecen diferentes funcionalidades de transformación en dos dimensiones, que veremos a continuación:



Traslaciones

Las **funciones de translación** son aquellas que realizan una transformación en la que **mueven** un elemento de un lugar a otro. Si especificamos un valor positivo en el eje X (*horizontal*), lo moveremos hacia la derecha, y si especificamos un valor negativo, lo moveremos hacia la izquierda. Lo mismo con el eje Y (*vertical*):

Funciones	Significado
<code>translateX(x)</code>	Traslada el elemento una distancia de <code>SIZE</code> <i>x</i> horizontalmente.
<code>translateY(y)</code>	Traslada el elemento una distancia de <code>SIZE</code> <i>y</i> verticalmente.
<code>translate(x, y)</code>	Propiedad de atajo de las dos anteriores.

Por ejemplo, `transform: translate(20px, -30px)` traslada el elemento 20 píxeles a la derecha y 30 píxeles hacia arriba, que es equivalente a utilizar `transform: translateX(20px) translateY(-30px)`.

Escalado

Las **funciones de escalado** realizan una transformación en la que aumentan o reducen el tamaño de un elemento, basándose en el parámetro indicado, que no es más que un factor de escala:

Funciones	Significado
<code>scaleX(fx)</code>	Reescala el elemento a un nuevo tamaño con un factor <code>NUMBER</code> <i>fx</i> horizontal.
<code>scaleY(fy)</code>	Reescala el elemento a un nuevo tamaño con un factor <code>NUMBER</code> <i>fy</i> vertical.
<code>scale(fx, fy)</code>	Propiedad de atajo de las dos anteriores.

En este ejemplo, `transform: scale(2, 2)` realiza una transformación de escalado del elemento, ampliándolo al doble de su tamaño original. Si utilizamos `scale()` con dos parámetros iguales, estamos manteniendo la proporción del elemento, pero si utilizamos diferentes valores, acabaría deformándose.

Rotaciones

Las **funciones de rotación** simplemente giran el elemento el número de grados indicado:

Funciones	Significado
<code>rotate(xdeg)</code>	Establece una rotación 2D en <code>DIRECTION</code> <code>xdeg</code> grados sólo para el eje horizontal X.
<code>rotateY(ydeg)</code>	Establece una rotación 2D en <code>DIRECTION</code> <code>ydeg</code> grados sólo para el eje vertical Y.
<code>rotate(deg)</code>	Establece una rotación 2D en <code>DIRECTION</code> <code>deg</code> grados sobre si mismo.

Con `transform: rotate(5deg)` realizamos una rotación de 5 grados del elemento sobre si mismo. Utilizando `rotateX()` y `rotateY()` podemos hacer lo mismo respecto al eje X o el eje Y respectivamente.

Deformaciones

Por último, las **funciones de deformación** establecen un ángulo para torcer, tumbar o inclinar un elemento en 2D:

Funciones	Significado
<code>skewX(xdeg)</code>	Establece un ángulo de <code>DIRECTION</code> <code>xdeg</code> para una deformación 2D respecto al eje X
<code>skewY(ydeg)</code>	Establece un ángulo de <code>DIRECTION</code> <code>ydeg</code> para una deformación 2D respecto al eje Y

Aunque la función `skew()` existe, no debería ser utilizada, ya que está marcada como obsoleta y serán retiradas de los navegadores en el futuro. En su lugar deberían utilizarse `skewX()` o `skewY()`.

Transformaciones múltiples

Recuerda que si estableces varias propiedades `transform` en el mismo elemento con diferentes funciones de transformación, la segunda propiedad de transformación sobrescribirá a la anterior, como lo haría cualquier propiedad de CSS:

```
div {  
  transform: rotate(5deg);  
  transform: scale(2,2); /* Sobrescribe la anterior */  
}
```

CSS

Para evitar este comportamiento, una forma sencilla se basa en emplear múltiples transformaciones separándolas mediante un espacio. En el siguiente ejemplo, aplicamos una función de rotación, una función de escalado y una función de traslación de forma simultánea:

```
div {  
  transform: rotate(5deg) scale(2,2) translate(20px, 40px);  
}
```

CSS