

# {css}



## Clase 33

# Clase 33

## <índice>

### Transformaciones en CSS

¿Qué son? y uso

---

Translaciones

---

Escalado

---

Rotaciones

---

Deformaciones

---

Transformaciones en 3D y perspectiva

---

**¿Qué son? y uso**

## ¿Qué son?

Las **transformaciones** son una de las características de CSS más interesantes y potentes que se introducen en el lenguaje para convertir las hojas de estilo en un sistema capaz de realizar efectos visuales 2D y 3D

# ¿Qué son?

## La propiedad **transform**

Propiedad	Descripción	Valor
transform	Aplica una (o varias) transformaciones CSS al elemento.	<a href="#">función 1</a> , <a href="#">función 2</a> , <a href="#">función 3...</a>

# ¿Qué son?

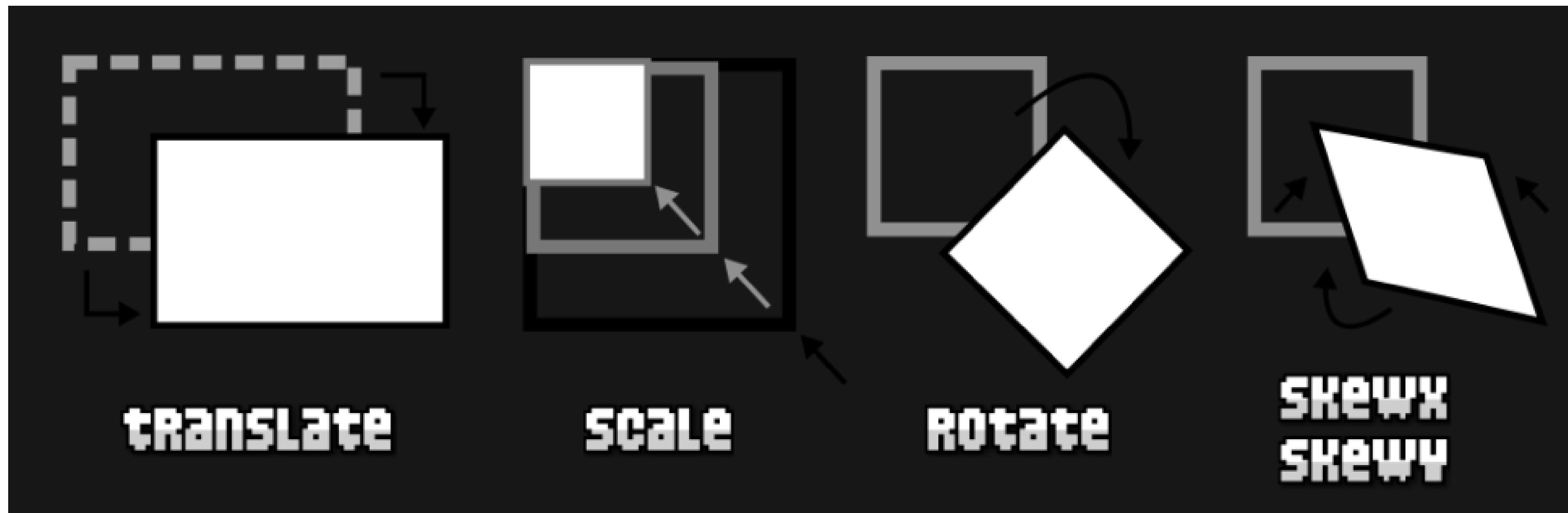
¿Os acordáis de los naipes?

```
.element {  
    width: 100px;  
    height: 100px;  
    background: indigo;  
    transform: rotate(45deg);  
}
```

```
<div class="element"></div>
```

# ¿Qué son?

¿Qué tipo de funciones podemos aplicar?



# ¿Qué son?

¿Qué tipo de funciones podemos aplicar?

Tipo de transformación	Descripción
<u>Translación 2D</u>	Desplaza un elemento en el eje X ( <a href="#">izquierda</a> , <a href="#">derecha</a> ) y/o en el eje Y ( <a href="#">arriba</a> , <a href="#">abajo</a> )
<u>Escalado 2D</u>	Escala el elemento una determinada cantidad más grande o más pequeña. También se puede voltear.
<u>Rotación 2D</u>	Gira el elemento sobre su eje X o sobre su eje Y. También se puede girar sobre sí mismo.
<u>Deformación 2D</u>	Inclina el elemento sobre su eje X o sobre su eje Y.

¿Qué son?

Transformaciones múltiples

# ¿Qué son?

## Transformaciones múltiples

```
.element {  
  transform: rotate(5deg); /* No hace efecto */  
  transform: scale(2);    /* Sobreesccribe la anterior */  
}
```

# ¿Qué son?

## Transformaciones múltiples

```
.element {  
  transform: rotate(5deg), /* No hace efecto */  
  transform: scale(2);    /* Sobreescribe la anterior */  
}
```

# ¿Qué son?

## Transformaciones múltiples

```
.element {  
  transform: rotate(5deg) scale(2) translate(25px, 150px);  
}
```

# ¿Qué son?

## Orden de las transformaciones

## ¿Qué son?

Otro detalle que conviene tener en cuenta a la hora de aplicar múltiples transformaciones es que el orden de transformación importa. No es lo mismo realizar una rotación y luego una translación, que la misma translación primero y luego la misma rotación. Veámoslo en un ejemplo.

# ¿Qué son?

```
.element {  
    width: 50px;  
    height: 50px;  
    background: grey;  
    transform: translate(150px, 100px) rotate(25deg);  
}
```

# ¿Qué son?

## Punto de origen

Propiedades	Formato	Significado
transform-origin	POSX POSY	Cambia el punto de origen del elemento en una transformación.

# ¿Qué son?

¿Cómo afecta al posicionamiento de elementos?

# Translación

# Translación

Las funciones de translación son aquellas que realizan una transformación en la que mueven un elemento de un lugar a otro. Para ello, las utilizaremos en el interior de la propiedad CSS transform y elegiremos una de las siguientes funciones de translación.

# Translación

Funciones	Significado
<code>translateX(x)</code>	Traslada el elemento una distancia de <b>SIZE</b> <code>x</code> horizontalmente.
<code>translateY(y)</code>	Traslada el elemento una distancia de <b>SIZE</b> <code>y</code> verticalmente.
<code>translate(x, y)</code>	Propiedad de atajo de las dos anteriores.
<code>translate(x)</code>	Equivalente a <code>translate(x, 0)</code>
<code>translateZ(z)</code>	<a href="#">Ver en Transformaciones 3D</a>
<code>translate3d(x, y, z)</code>	<a href="#">Ver en Transformaciones 3D</a>

# Translación

Respecto al ámbito 2D, tenemos 3 funciones de translación: `translateX()` y `translateY()` para cada eje, y la propiedad de atajo `translate()`, que es una mezcla de las dos anteriores en una sola función de translación.

# Translación

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. Ocurre exactamente lo mismo con el eje Y (vertical), que con valores negativos lo movemos hacia arriba y con valores positivos lo movemos hacia abajo

# Translación

```
.element {  
  transform: translateX(20px) translateY(-30px);  
}  
  
/* La transformación anterior es equivalente a esta (atajo) */  
.element {  
  transform: translate(20px, -30px);  
}
```

# Translación

## Propiedad translate

En nuevas versiones de los navegadores, ya se soporta la propiedad individual `translate`, y no hace falta utilizarla dentro de la propiedad `transform`.

Propiedad	Valor	Significado
<code>translate</code>	<code>none</code>	No aplica desplazamiento. Valor por defecto.
<code>translate</code>	<code>SIZE</code>	Desplaza un elemento el tamaño especificado en el eje X.
<code>translate</code>	<code>SIZE</code> <code>SIZE</code>	Desplaza un elemento una cierta cantidad en el eje X y eje Y.
<code>translate</code>	<code>SIZE</code> <code>SIZE</code> <code>SIZE</code>	Desplaza un elemento el eje X, eje Y y eje Z.

# Translación

```
.element {  
    translate: 50px;          /* Equivalente a translateX(50px) */  
    translate: 50px 150px;    /* Equivalente a translate(50px, 150px) */  
    translate: 0 150px;      /* Equivalente a translateY(150px) */  
    translate: 50px 150px 100px; /* Equivalente a translate(50px, 150px, 100px) */  
    translate: 0px 0px 30px;   /* Equivalente a translateZ(30px) */  
}
```

# Translación

Demos

# **Escalado**

# Escalado

Las funciones de **escalado** son aquellas que realizan una transformación en la que aumentan o reducen el tamaño de un elemento. Para ello, las utilizaremos en el interior de la propiedad CSS transform y elegiremos una de las siguientes funciones de escalado.

# Escalado

Funciones	Significado
scaleX(fx)	Reescala el elemento un número de <b>NUMBER</b> <code>fx</code> veces (sólo en horizontal).
scaleYfy()	Reescala el elemento un número de <b>NUMBER</b> <code>fy</code> veces (sólo en vertical).
scale(fx, fy)	Propiedad de <b>atajo</b> de las dos anteriores (escalado simétrico).
scale(fx)	Equivalente al anterior: <code>scale(fx, fx)</code> .
scaleZ(fz)	<a href="#">Ver en Transformaciones 3D</a>
scale3D(fx, fy, fz)	<a href="#">Ver en Transformaciones 3D</a>

# Escalado

Por ejemplo, la propiedad transform: scale(2, 0.5) realiza una transformación de escalado del elemento, ampliéndolo al doble de su tamaño original en el eje X (horizontal) y a la mitad en el eje Y (vertical).

# Escalado

## La propiedad **scale**

En nuevas versiones de los navegadores, ya se soporta la propiedad individual **scale**, y no hace falta utilizarla dentro de la propiedad **transform**.

Propiedad	Valor	Significado
scale	<b>none</b>	No aplica escalado. Valor por defecto.
scale	<b>NUMBER</b>	Aplica el factor de escala simétrico al eje X/Y. Igual a <code>scale: x y 1;</code>
scale	<b>NUMBER</b>	Aplica los factores de escala al eje X y eje Y. Igual a <code>scale: x y 1;</code>
scale	<b>NUMBER</b>	Aplica un factor de escala a cada eje. Igual a <code>scale: x y z;</code>

# Escalado

Función espejo con CSS

# Escalado

DEMO

# Rotación

# Rotación

Las funciones de rotación son aquellas que realizan una transformación en la que giran un elemento respecto a un eje específico. Para ello, las utilizaremos en el interior de la propiedad CSS transform y elegiremos una de las siguientes funciones de rotación.

# Rotación

## Rotación

Las funciones de rotación simplemente giran el elemento una cierta cantidad respecto al eje involucrado. Disponemos de las siguientes funciones de rotación

# Rotación

Funciones	Significado
rotateX(x)	Establece una rotación 2D en <b>ANGLE</b> <b>x</b> sólo para el eje horizontal X.
rotateY(y)	Establece una rotación 2D en <b>ANGLE</b> <b>y</b> sólo para el eje vertical Y.
rotateZ(z)	Establece una rotación 2D en <b>ANGLE</b> <b>z</b> sobre si mismo.
rotate(z)	Alias a la anterior.
rotate3d(x, y, z, a)	<a href="#">Ver en Transformaciones 3D</a>

Demo

# Rotación

## La propiedad **rotate**

En nuevas versiones de los navegadores, ya se soporta la propiedad individual `rotate`, y no hace falta utilizarla dentro de la propiedad `transform`.

Propiedad	Valor	Significado
<code>rotate</code>	<code>none</code>	No aplica rotación. Valor por defecto.
<code>rotate</code>	<code>ANGLE</code>	Rota el elemento sobre si mismo. Equivalente a <code>rotateZ()</code> .
<code>rotate</code>	<code>AXIS</code> <code>ANGLE</code>	Rota el elemento sobre el eje ( <code>x</code> , <code>y</code> o <code>z</code> ) indicado.
<code>rotate</code>	<code>NUMBER</code> <code>NUMBER</code> <code>NUMBER</code> <code>ANGLE</code>	Indica un vector de rotaciones con el ángulo indicado.

# Rotación

```
.element {  
    rotate: 45deg;          /* Equivale a transform: rotateZ(45deg); */  
  
    rotate: x 45deg;        /* Equivale a transform: rotateX(45deg); */  
    rotate: y 120deg;        /* Equivale a transform: rotateY(120deg); */  
  
    rotate: 0 0 1 45deg;    /* Equivale a transform: rotateZ(45deg); */  
    rotate: 1 0 0 15deg;    /* Equivale a transform: rotateX(15deg); */  
    rotate: 0 1 1 5deg;     /* Equivale a transform: rotateY(5deg) rotateZ(5deg); */  
}
```

# Rotación

DEMO

# Deformaciones

# Deformaciones

Por último, las funciones de deformación establecen un ángulo para torcer, tumbar o inclinar un elemento en 2D. A diferencia de las anteriores, no tiene función correspondiente para 3D.

# Deformaciones

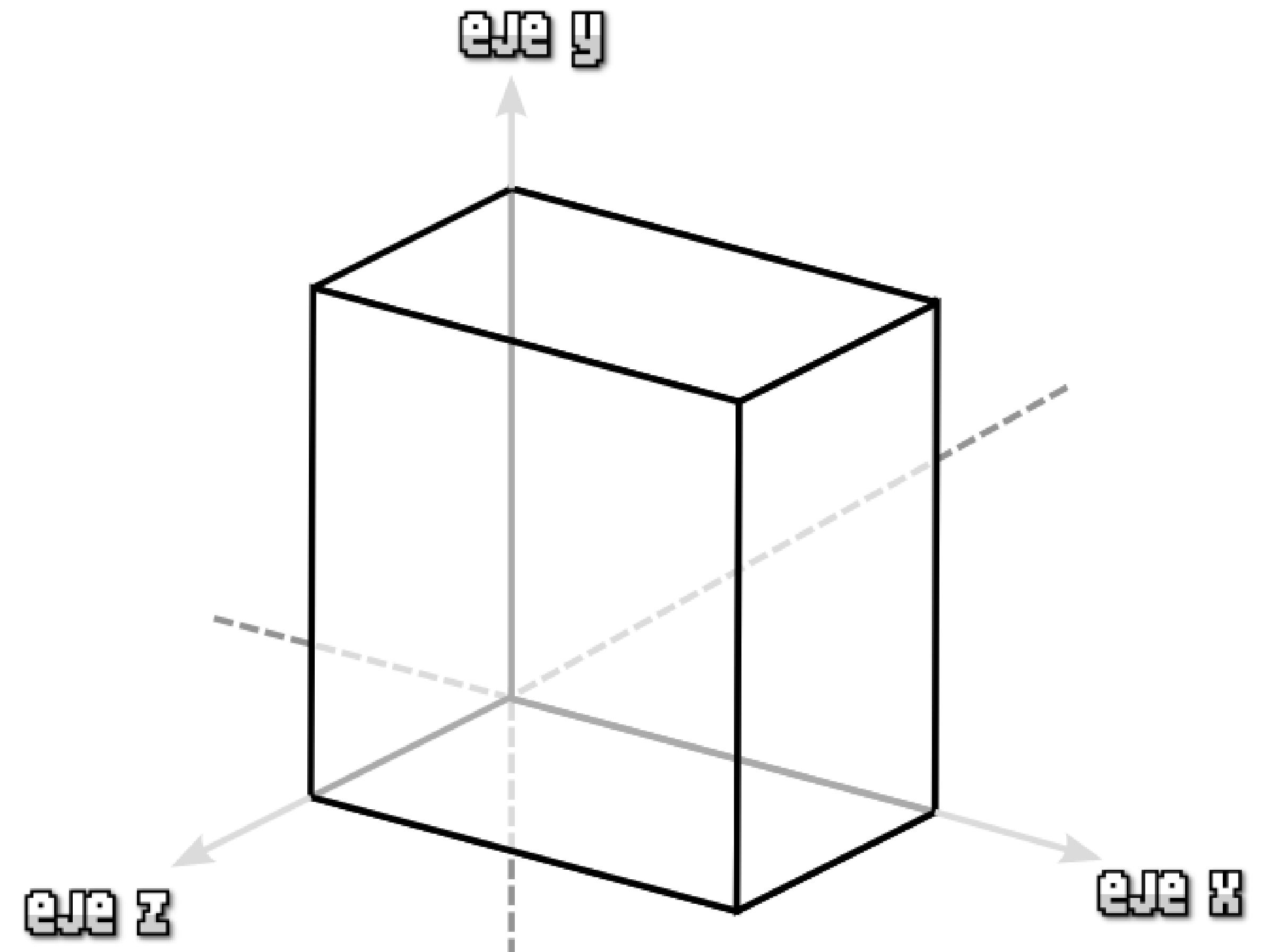
Funciones	Significado
<code>skewX(xdeg)</code>	Establece un ángulo de <b>ANGLE</b> <code>xdeg</code> para una deformación 2D respecto al eje X.
<code>skewY(ydeg)</code>	Establece un ángulo de <b>ANGLE</b> <code>ydeg</code> para una deformación 2D respecto al eje Y.

# Deformaciones

Demo

# **Transformaciones en 3D y perspectiva**

# 3D



**3D**

Demo de perspectiva 1

Demo de perspectiva 2

Demo TRANSFORMACIONES 3D

**<Despedida>**

Email

**bienvenidosaez@gmail.com**

Instagram

**@bienvenidosaez**

Youtube

**youtube.com/bienvenidosaez**