

Transformaciones

Módulo 04

Transformaciones 2D

Implementación

Transformaciones 2D

Las transformaciones, me permiten rotar, escalar y trasladar los elementos de mi **HTML**, para luego hacer un buen complemento con animaciones y transiciones de **CSS3**.

Veamos un ejemplo de sintaxis para comprender mejor las transformaciones,

```
<style>
div {
  transform: tipoTransformacion(parametro);
}

</style>
```



Implementación

ROTATE

Rotamos el objeto que puede ser de línea o de bloque tanto a favor como en el sentido contrario a las agujas del reloj, es decir que los valores posibles son, 0 a 360deg o en el sentido contrario de las agujas del reloj 0 a -360deg,

```
<style>
div {
  width: 200px;
  height: 200px;
  border-radius: 20px;
  padding: 20px;
  background: linear-gradient(blue,red);
  transform: rotate(34deg);
}
</style>
```

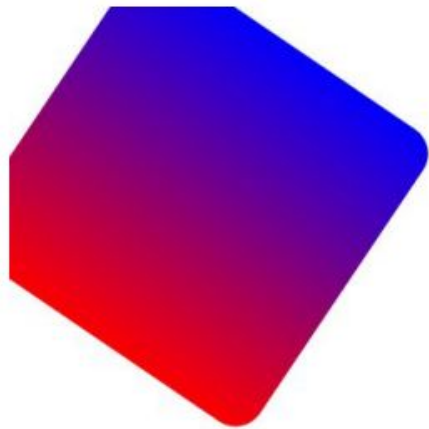
```
<div>
  mi contenedor
</div>
```



Implementación

ROTATE

En base al [código anterior el resultado será.](#)



Implementación

Transform-origin

Se pueden utilizar, **medidas de longitud** , **px**, **em**, **%** (en referencia al alto o al ancho del elemento que estoy transformando), **así como palabras (center, top,bottom,etc)**, es decir aquellos valores que por ejemplo utilizamos en la propiedad **background-position**.

```
<style>
div {
  width: 200px;
  height: 200px;
  border-radius: 20px;
  padding: 20px;
  background: linear-gradient(blue,red);
  transform: rotate(34deg);
  transform-origin: 20px top;
}
</style>
```



Transformaciones 3D

Implementación

Rotación 3D

La rotación 3D va desde **0 a 180deg** **0 a -180deg**, y se puede realizar **tanto con x, y así como z**,

```
<style>
div {
  width: 200px;
  height: 200px;
  border-radius: 20px;
  padding: 20px;
  background: linear-gradient(blue,red);
  transform: rotateY(-34deg);

}

</style>
```



Implementación

Rotación 3D

También podemos **trabajar de la siguiente manera,**

```
<style>
div {
  width: 200px;
  height: 200px;
  border-radius: 20px;
  padding: 20px;
  background: linear-gradient(blue,red);
  transform: rotateX(100deg);

}

</style>
```



Implementación

Rotación 3D

El [código anterior podemos](#) se verá de la siguiente manera,



Implementación

¿Cómo empezar a trabajar de forma más eficiente?

Podemos favorecer la cantidad de líneas dentro de nuestro css utilizando **un shorthand o declaración de una sólo línea**,

```
div {  
  width: 200px;  
  height: 200px;  
  border-radius: 20px;  
  padding: 20px;  
  background: linear-gradient(blue,red);  
  transform: rotateY(160deg) rotateX(-34deg);  
}
```



Implementación

¿Cómo empezar a trabajar con valores negativos?

También podemos trabajar con valores negativos como en el ejemplo anterior. **Por supuesto también podemos sumar una rotación en 2D veamos un ejemplo,**

```
<style>
div {
  width: 200px;
  height: 200px;
  border-radius: 20px;
  padding: 20px;
  background: linear-gradient(blue,red);
  transform: rotate(34deg) rotateY(-30deg);
}
</style>
```



Implementación

Resultados,

El [código anterior](#) se verá de la siguiente manera en el navegador,



Revisión

- Repasar los conceptos vistos de **transform**.
- Trabajar con el **bonus track** debajo de todo para practicar los **elementos individualmente**.
- Ver todos los videos y materiales necesarios antes de continuar



¡Muchas gracias!

¡Sigamos trabajando!