Transformaciones Avanzado

Módulo 04

Perspective

Implementación

Perspective

La **propiedad perspective se aplica al padre del elemento** que **contiene la rotación en 3D.**

Por esa razón debemos seguir la siguiente estructura en nuestro **archivo.html**

```
<padre>

<hijo> </hijo> <!--elemento es el que va a tener la transformación en 3D-->

</padre> <!--elemento que va a tener la perspectiva-->
```

Implementación

Perspective

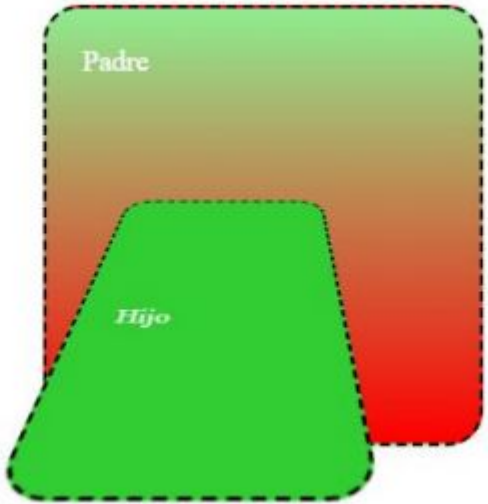
Ahora trabajamos en nuestros **estilos.css,**

```
<style>
div {
   width: 200px;
   height: 200px;
   border-radius: 20px;
   padding: 20px;
   background: linear-gradient(lightgreen,red);
   color: white;
   border: 2px dashed black;    }
#padre {   margin-left: 100px; perspective: 200px;
   perspective-origin: center bottom;}
 #hijo { background: limegreen; width: 100px;  transform: rotateX(34deg); line-height: 230px;}

                              </style>


<div id="padre">Padre  <div id="hijo">  Hijo </div> </div>
```

Implementación

Perspective, resultados

El resultado del código anterior será el siguiente,



Scale

Implementación

SCALE

Me permite **escalar un elemento tanto haciéndolo más pequeño o más grande.** Veamos un ejemplo,

```
<style>
div {
   width: 200px;
   height: 200px;
   border-radius: 20px;
   padding: 20px;
   background: linear-gradient(lightgreen,red);
   color: white;
   border: 2px dashed black;
   transform: scale(2);    }

                              </style>


<div>contenido</div>
```

Implementación

SCALE

La alternativas para **trabajar varían, veamos todas las posibles,**

```
transform: scale(width,height)
/*escalamos tanto el ancho como el alto*/
transform: scaleX(n)
/*ancho del elemento*/
transform: scaleY(n)
/*alto del elemento*/
```

Implementación

SCALE

Ahora veamos un ejemplo en nuestros estilos.css,

```
<style>
div {
   width: 200px;
   height: 200px;
   border-radius: 20px;
   padding: 20px;
   background: linear-gradient(lightgreen,red);
   color: white;
   border: 2px dashed black;
   transform: scale(0.5, 2);   }


                              </style>



<div>contenido</div>
```

Translate

Implementación

TRANSLATE

La diferencia entre trasladar un elemento en una animación en una transición **con translate a hacerlo con position , es que es mucho más fluido el movimiento.**

Te permite **trasladar un elemento a través de las transformaciones, las posibles variantes son,**

```
transform: translate(x,y);
transform: translateX();
transform: translateY();
```

Implementación

TRANSLATE

Un ejemplo de lo explicado anteriormente en nuestros **estilos.css,**

```
<!DOCTYPE html>
<head>
<style>
div {
   width: 200px;
   height: 200px;
   border-radius: 20px;
   padding: 20px;
   background: linear-gradient(lightgreen,red);
   color: white;
   border: 2px dashed black;
   transform: translate(100px, 200px);    }


                                    </style>
```

Skew

Implementación

SKEW

Esto sería lo que por ejemplo en **PHOTOSHOP O ILLUSTRATOR** llamamos **sesgar,** las posibles variantes son,

```
transform: skew(x,y)
transform: skewX()
transform: skewY()
```

Implementación

SKEW

Un ejemplo en nuestros **estilos.css** es,

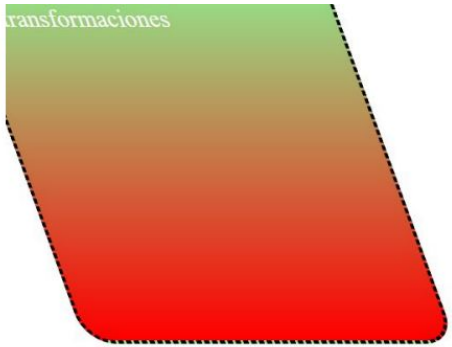
```
<style>
div {
   width: 200px;
   height: 200px;
   border-radius: 20px;
   padding: 20px;
   background: linear-gradient(lightgreen,red);
   color: white;
   border: 2px dashed black;
   transform: skewX(20deg)    }

                              </style>

<div> transformaciones </div>
```

Implementación

SKEW

No tiene sentido trabajar ni con 0 ni con 180deg porque ese sería el valor predeterminado, el elemento no tiene cambios es decir no está sesgado, **la idea es siempre trabajar con valores entre el rango antes referenciado.**

En todos los casos anteriores, es decir en el **skew y en el translate podemos hacerlo también sobre el eje Z, por ejemplo skewZ() o translateZ(),**



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!