



Carmelo José Jaén Díaz



# Transformaciones en CSS: rotar, torcer, escalar o deplazar

- 
- C.F.G.S. DAW
  - 6 horas semanales
  - Mes aprox. de impartición: Ene - Feb
  - iPasen - [cjaedia071@g.educaand.es](mailto:cjaedia071@g.educaand.es)

# ———— Índice ————



Objetivo

---

Glosario

---

Interacción persona - ordenador

---

Objetivos

---

Características. Usable.

---

Características. Visual.

---

Características. Educativo y actualizado.

---

# **OBJETIVO**

---



- **Analizar y seleccionar los colores y las tipografías adecuados para la visualización en la pantalla.**
- **Utilizar marcos y tablas para presentar la información de manera ordenada.**
- **Identificar nuevos elementos y etiquetas en HTML5.**
- **Reconocer las posibilidades de modificar etiquetas HTML.**
- **Valorar la utilidad de las hojas de estilo para conseguir un diseño uniforme en todo el sitio web.**

# GLOSARIO

---



**Formularios.** Documentos interactivos utilizados para recoger información en un sitio web. Esta información es enviada al servidor, donde es procesada. Cada formulario contiene uno o varios tipos de controles que permiten recolectar la información de varias formas diferentes.

**Fuentes seguras.** Fuentes tipográficas que los usuarios tenían instaladas por defecto en su dispositivo. En la actualidad, gracias a que la mayoría de los navegadores soportan la directiva @font-face, es posible utilizar casi cualquier tipografía a través de Google Fonts.

**Guías de estilo.** Documentos con directrices que permiten la normalización de estilos. En estas guías se recogen los criterios y normas que debe seguir un proyecto; de esta forma se ofrece una apariencia más uniforme y atractiva para el usuario.

**HTML.** Lenguaje de marcado de hipertexto utilizado en las páginas web. Este tipo de lenguaje presenta una forma estructurada y agradable, con hipervínculos que conducen a otros documentos y con inserciones multimedia (sonido, imágenes, vídeos...).

# GLOSARIO

---



**HTML5.** Última versión del lenguaje para la programación de páginas web HTML. Los sitios implementados con este lenguaje solo pueden visualizarse correctamente en los navegadores más actuales.

**Legibilidad.** Cualidad deseable en una familia tipográfica. Se trata de la facilidad de la lectura de una letra. Esta cualidad puede venir determinada por varios parámetros como el interletrado, el interpalabrado o el interlineado.

**Marcos.** Son las ventanas independientes incorporadas dentro de la página general. Gracias a ellos, cada página quedará dividida en varias subpáginas, permitiendo realizar un diseño más organizado y limpio a la vista. Con HTML5 ha quedado obsoleto.

**Tipografía.** Se trata del tipo de letra que se escoge para un determinado diseño. Según la RAE, significa "modo o estilo en que está impreso un texto" o "clase de tipos de imprenta".

# INTRODUCCIÓN

---



Las transformaciones CSS nos permiten rotar, torcer, escalar o desplazar los elementos de nuestra página web. Este tipo de propiedades de CSS3 son muy interesantes para convertir el lenguaje de hojas de estilo en un sistema capaz de realizar todo tipo de efectos visuales.



# TRANSFORMACIONES EN CSS

---



Las dos propiedades que nos sirven para definir las transformaciones son `transform` y `transform-origin`.

# TRANSFORMACIONES EN CSS

## Propiedad *transform-origin*



Con `transform-origin` modificamos la posición de origen que se utilizará para la transformación. Por defecto es el centro del elemento.

VALORES ACEPTADOS POR `transform-origin`:

- Palabras clave como:
  - `top`, `left`, `right`, `bottom`, `center`. Combinarlas como: `top left`, `bottom right`, etc.

EJEMPLO

```
transform-origin: top left; /* Esquina superior izquierda */
transform-origin: bottom right; /* Esquina inferior derecha */
transform-origin: center; /* Centro del elemento (valor por defecto) */
```



# TRANSFORMACIONES EN CSS

## Propiedad *transform-origin*

---



Con `transform-origin` modificamos la posición de origen que se utilizará para la transformación. Por defecto es el centro del elemento.

VALORES ACEPTADOS POR `transform-origin`:

- Porcentajes:
  - `0%` representa el inicio. `100%` representa el final. Ejemplo: `50% 50%` (equivalente al centro del elemento).

EJEMPLO

```
transform-origin: 25% 75%; /* 25% desde la izquierda, 75% desde arriba */
```

# TRANSFORMACIONES EN CSS

## Propiedad *transform-origin*

---



Con `transform-origin` modificamos la posición de origen que se utilizará para la transformación. Por defecto es el centro del elemento.

VALORES ACEPTADOS POR `transform-origin`:

- Coordenadas absolutas (píxeles, em, etc.):
  - Ejemplo: `20px 30px`.

EJEMPLO

```
transform-origin: 20px 30px; /* 20px desde la izquierda, 30px desde  
arriba */
```

# TRANSFORMACIONES EN CSS

## Propiedad *transform-origin*. Ejemplo

---



Código HTML:

```
<h2>Rota 45 grados desde el centro</h2>  
<div class="box-center"></div>  
<h2>Rota 45 grados desde la esquina superior izquierda</h2>  
<div class="box-top-left"></div>
```

# TRANSFORMACIONES EN CSS

## Propiedad *transform-origin*. Ejemplo

---



### Código CSS:

```
.box-center {  
  width: 100px;  
  height: 100px;  
  background-color: coral;  
  margin: 50px auto;  
  transition: transform 1s ease;  
}  
.box-center:hover {  
  transform: rotate(45deg); /* Rota 45  
grados desde el centro */  
}
```

```
.box-top-left {  
  width: 100px;  
  height: 100px;  
  background-color: skyblue;  
  margin: 50px auto;  
  transition: transform 1s ease;  
  transform-origin: top left; /* Cambia  
el origen de transformación a la esquina  
superior izquierda */  
}  
.box-top-left:hover {  
  transform: rotate(45deg); /* Rota 45  
grados desde la esquina superior  
izquierda */  
}
```

# TRANSFORMACIONES EN CSS

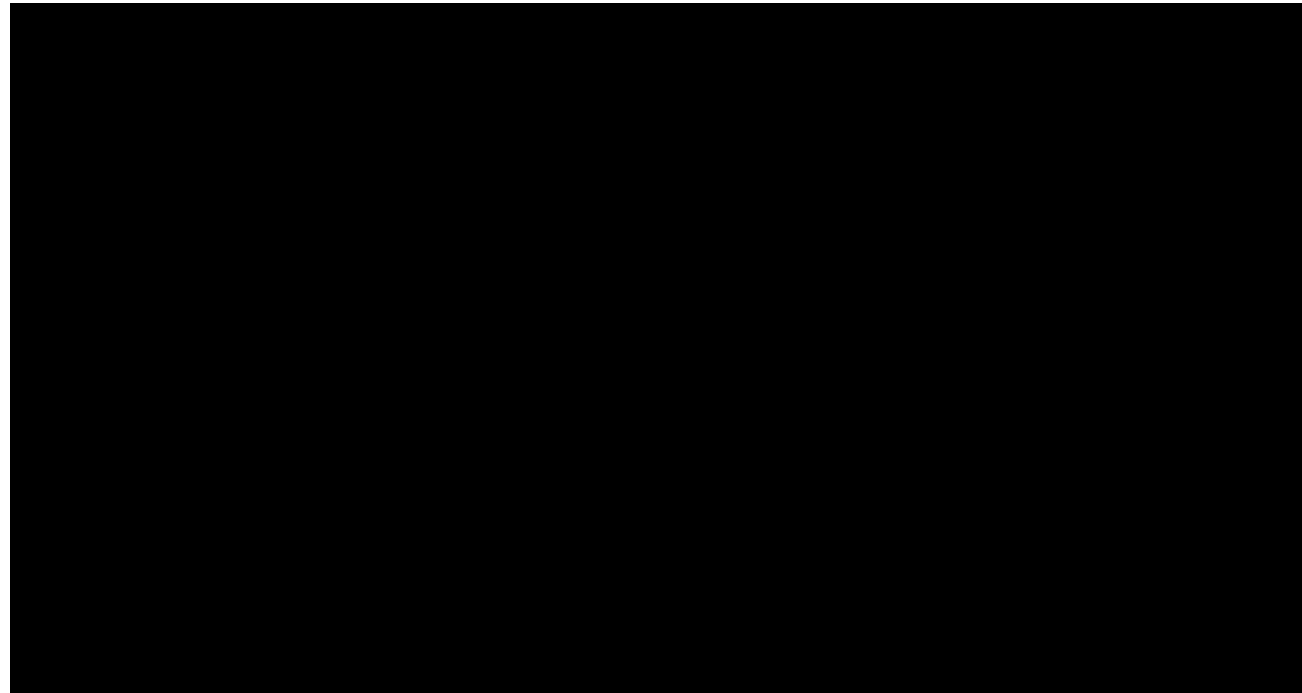
## Propiedad *transform-origin*. Ejemplo

---



La salida del código anterior puede verse en la imagen adjunta o en el siguiente link:

<https://codepen.io/Carmelo-Jos-Ja-n-D-az/pen/oqvWJJB>



# TRANSFORMACIONES EN CSS

## Propiedad *transform*

---



La propiedad `transform` se utiliza para aplicar una o más transformaciones a un elemento. Su sintaxis básica es:

```
transform: tipo_de_transformación(valor);
```

Puedes aplicar varias transformaciones a la vez separándolas por un espacio.

```
transform: tipo_de_transformación(valor)  
otro_tipo_de_transformación(valor);
```

# TIPOS DE TRANSFORMACIONES PARA LA PROPIEDAD *transform*



Las transformaciones que podemos aplicar son las siguientes:

- **SCALE**

Modifica el tamaño de los elementos. La función `scale()` se establece con uno o dos valores, que representan la cantidad de escala que se aplica en cada dirección: `scale(x)` o `scale(x,y)`. Se define mediante un valor numérico de manera que cuando un valor de coordenadas está fuera del rango  $[-1, 1]$ , el elemento crece a lo largo de esa dimensión. Cuando está dentro del rango el elemento se encoge.

```
transform: scale(0.5); /* Escala el elemento a la mitad */
```

# TIPOS DE TRANSFORMACIONES PARA LA PROPIEDAD *transform*



Las transformaciones que podemos aplicar son las siguientes:

- **TRANSLATE**

Cambia la posición del elemento hacia la izquierda, derecha, arriba o abajo. La función `translate()` se establece con uno o dos valores: `translate(x)` o `translate(x,y)`. Los valores `x` e `y` son los vectores de translación en las coordenadas `x` e `y`. Sus valores pueden estar definidos en píxeles, porcentajes,...

```
transform: translate(10px); /* Traslada el elemento 10px hacia la derecha */
```



# TIPOS DE TRANSFORMACIONES PARA LA PROPIEDAD *transform*



Las transformaciones que podemos aplicar son las siguientes:

- ROTATE

Gira o rota los elementos en grados: rotate(v).

```
transform: rotate(45deg); /* Rota el elemento 45 grados */
```

# TIPOS DE TRANSFORMACIONES PARA LA PROPIEDAD *transform*



Las transformaciones que podemos aplicar son las siguientes:

- **SKEW**

Distorsiona los elementos según el ángulo en grados. La función *skew()* se establece con uno o dos valores: *skew(x)* o *skew(x,y)*.

```
transform: skew(45deg); /* Distorsiona el elemento 45 grados en el eje x */
```

# TIPOS DE TRANSFORMACIONES PARA LA PROPIEDAD *transform*



Las transformaciones que podemos aplicar son las siguientes:

- **MATRIX**

Mueve o transforma los elementos con precisión de píxel. La función *matrix()* se establece con seis valores numéricos: *matrix(a,b,c,d,x,y)*. Los dos últimos valores representan la translación y los primeros la transformación lineal.

```
transform: matrix(0.5, 0.1, 0.5, 1, 10, -2);
```

# TIPOS DE TRANSFORMACIONES PARA LA PROPIEDAD *transform*



La propiedad `transform` se usa junto con la propiedad transition vista en la sección anterior para que la transformación pueda tener una transición espaciada en el tiempo:

```
.caja1{
  -webkit-transition: 1s linear;
  transition: 1s linear;
}
.caja1:hover{
  -webkit-transform: scale(.5);
  transform: scale(.5);
}
```

Recuerda utilizar la extensión que te facilita la tarea de crear los prefijos para navegadores

# EJERCICIO PROPUESTO I

---



Crea un contenedor de 180 píxeles de ancho y de alto, incluye un texto en su interior y aplícale un color de fondo. Realiza las siguientes transformaciones cuando el usuario pase por encima del contenedor el puntero del ratón:

- Modifica la escala del contenedor.
- Modifica la rotación del contenedor.
- Modifica la posición del elemento mediante “translate”.
- Distorsiona el contenedor mediante “skew”.

# EJERCICIO PROPUESTO I

## Solución

---



Código HTML:

```
<div class="container">
  <div class="caja caja1">Scale</div>
  <div class="caja caja2">Rotate</div>
  <div class="caja caja3">Translate</div>
  <div class="caja caja4">Skew</div>
  <div class="caja caja5">SkewY</div>
  <div class="caja caja6">SkewX</div>
  <div class="caja caja7">TranslateX</div>
  <div class="caja caja8">TranslateY</div>
  <div class="caja caja9">Perspective rotate</div>
  <div class="caja caja10">Perspective rotate</div>
  <div class="caja caja11">Matrix</div>
</div>
```

# EJERCICIO PROPUESTO I

## Solución

---



Código CSS:

```
.container{
  position: relative;
}
.caja{
  width: 180px;
  height: 180px;
  color: #FFF;
  text-align: center;
  line-height: 180px;
  margin: 15px;
  float: left;
  font-size: 18px;
  font-family: Arial;
}
```

```
.caja1{
  background-color: #C0392B;
  -webkit-transition: 1s linear;
  transition: 1s linear;
}
.caja1:hover{
  -webkit-transform: scale(.5);
  transform: scale(.5);
}
.caja2{
  background-color: #9B59B6;
  -webkit-transition: 1s linear;
  transition: 1s linear;
}
.caja2:hover{
  -webkit-transform: rotate(360deg);
  transform: rotate(360deg);}
```

# EJERCICIO PROPUESTO I

## Solución

---



```
.caja3{
  -webkit-transition: 1s linear;
  transition: 1s linear;
  background-color: #2980B9;
}
.caja3:hover{
  -webkit-transform: translate(10%);
  transform: translate(10%);
}
.caja4{
  background-color: #3498DB;
  -webkit-transition: 1s linear;
  transition: 1s linear;
}
.caja4:hover{
  -webkit-transform: skew(10deg);
  transform: skew(10deg);}
```

```
.caja5{
  background-color: #17A589;
  -webkit-transition: 1s linear;
  transition: 1s linear;
}
.caja5:hover{
  -webkit-transform: skewY(10deg);
  transform: skewY(10deg);
}
.caja6{
  background-color: #F1C40F;
  -webkit-transition: 1s linear;
  transition: 1s linear;
}
.caja6:hover{
  -webkit-transform: skewX(-5deg);
  transform: skewX(-5deg);}
```



# EJERCICIO PROPUESTO I

## Solución

---



```
.caja7{
  -webkit-transition: 1s linear;
  transition: 1s linear;
  background-color: #E67E22;
}
.caja7:hover{
  -webkit-transform: translateX(20px);
  transform: translateX(20px);
}
.caja8{
  -webkit-transition: 1s linear;
  transition: 1s linear;
  background-color: #95A5A6;
}
.caja8:hover{
  -webkit-transform: translateY(20px);
  transform: translateY(20px);}
```

```
.caja9{
  -webkit-transition: 1s linear;
  transition: 1s linear;
  background-color: #2C3E50;
}
.caja9:hover{
  -webkit-transform: perspective(150px)
  rotateX(45deg);
  transform: perspective(150px)
  rotateX(45deg);
}
```

# EJERCICIO PROPUESTO I

## Solución

---



```
.caja10{
  -webkit-transition: 1s linear;
  transition: 1s linear;
  background-color: grey;
}

.caja10:hover{
  -webkit-transform: perspective(150px)
  rotateY(45deg);
  transform: perspective(150px)
  rotateY(45deg);
}
```

```
.caja11{
  -webkit-transition: 1s linear;
  transition: 1s linear;
  background-color: black;
}

.caja11:hover{
  -webkit-transform: matrix(0.5, 0.1,
0.5,1,10, -2);
  transform: matrix(0.5, 0.1, 0.5, 1,
10, -2);
}
```

# EJERCICIO PROPUESTO I

## Solución



La salida del código anterior puede verse en la imagen adjunta o en el siguiente link:

<https://codepen.io/Carmelo-Jos-Ja-n-D-az/pen/PwYmVOX>



# EJERCICIO PROPUESTO II

---



Centra vertical y horizontalmente una imagen o un contenedor dentro de un div mediante el uso de la propiedad `transform` y posición absoluta en CSS.

1. `div { position: relative; }`: Este estilo establece la posición relativa para el div contenedor. Esto es necesario para que los elementos posicionados absolutamente dentro de él se alineen correctamente con respecto a él.
2. `div img { position: absolute; left: 50%; top: 50%; transform: translate(-50%, -50%); }`: Estos estilos se aplican a la imagen (o cualquier otro elemento dentro del div). Se establece una posición absoluta para la imagen, lo que la saca del flujo normal del documento y la coloca en relación con su primer ancestro posicionado. Luego, las propiedades *left* y *top* se establecen en 50%, lo que significa que el borde superior izquierdo de la imagen estará posicionado en el 50% del ancho y el 50% de la altura del div contenedor. Finalmente, la propiedad *transform: translate(-50%, -50%)* se utiliza para mover la imagen hacia arriba y hacia la izquierda en un 50% de su propio tamaño, lo que logra el centrado vertical y horizontal.

# EJERCICIO PROPUESTO II

## Solución



Código HTML:

```
<div>  
    
</div>
```

# EJERCICIO PROPUESTO II

## Solución

---



Código CSS:

```
div {  
    /* Establecer tamaño suficiente en el contenedor para que contenga a la imagen o  
    redimensionar imagen*/  
    height: 300px;  
    background-color: #ffffffc5;  
    position: relative;  
}  
div img {  
    position: absolute;  
    left: 50%;  
    top: 50%;  
    transform: translate(-50%, -50%);  
    -webkit-transform: translate(-50%, -50%);  
}
```

# EJERCICIO PROPUESTO I

## Solución



La salida del código anterior puede verse en la imagen adjunta o en el siguiente link:

<https://codepen.io/Carmelo-Jos-Ja-n-D-az/pen/VYZbqxd>



# EJERCICIO PROPUESTO II

---



Este método es útil para centrar imágenes o contenedores dentro de un div de manera rápida y sencilla. Sin embargo, no es adecuado para centrar texto, ya que no tiene en cuenta el tamaño de la letra y puede no producir resultados precisos en ese caso.