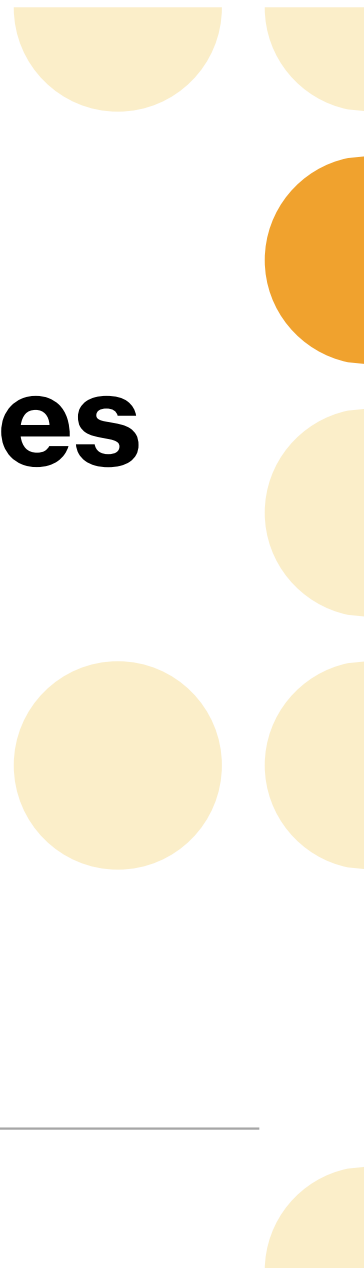


# **Transformaciones y Animaciones CSS**

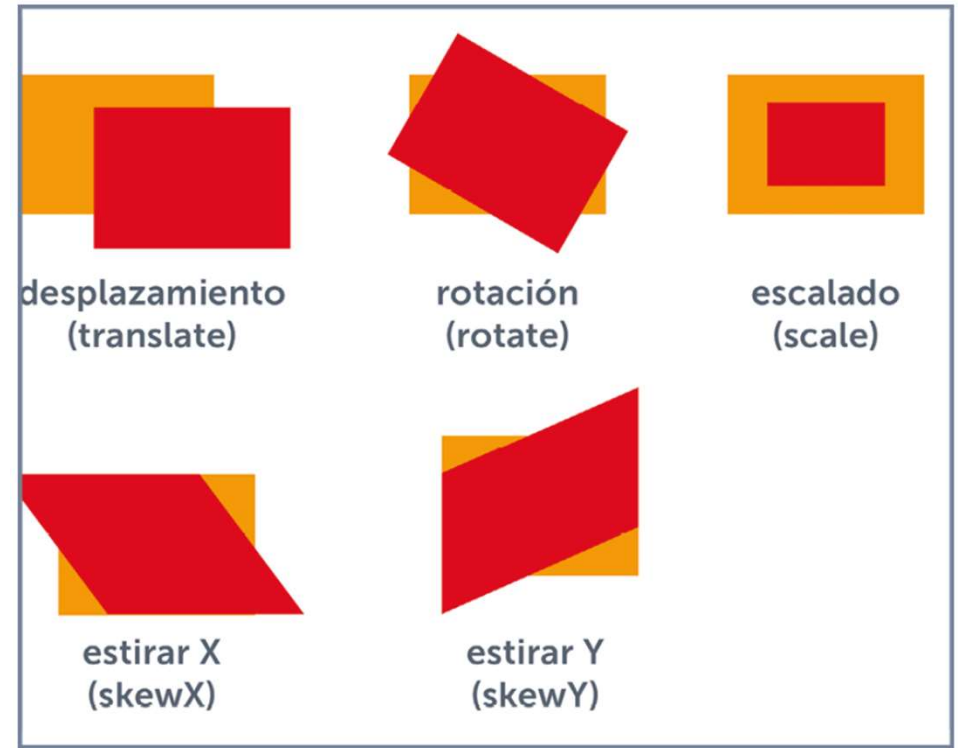
---



# Transformaciones

Las transformaciones en CSS son las operaciones realizadas mediante la propiedad **transform**, la cual permite las operaciones típicas vectoriales sobre un elemento HTML.

En realidad, hay dos tipos de transformaciones, las transformaciones 2D, que sólo utilizan los ejes horizontal y vertical para transformar, lo que implica que sólo realiza transformaciones en el plano; y las transformaciones 3D en las que interviene el eje Z causando la ilusión de que el elemento tenga profundidad.



Transformaciones 2D

# 2D

La sintaxis de **transform** es la siguiente:

```
selector { transform: valor1 valor2 ...; }
```

Donde **selector** es el elemento al que se aplica la transformación y **valor** es una función

Las funciones a utilizar pueden ser:

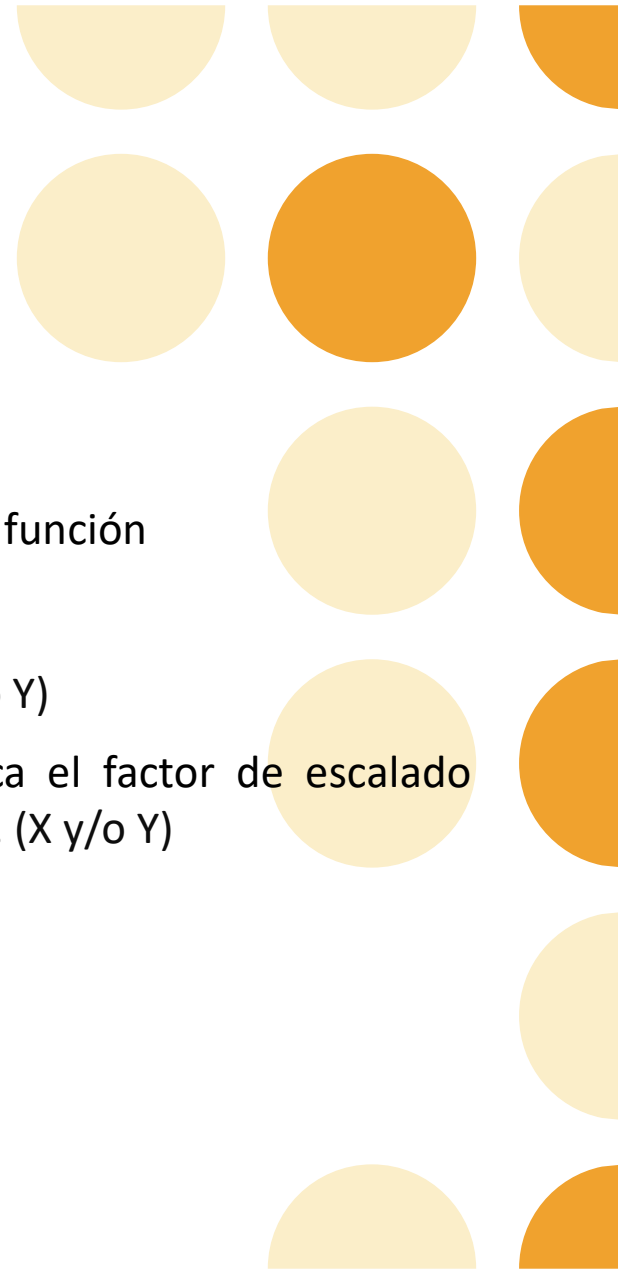
**translate(x,y)** para mover un elemento a lo largo de un eje específico (X y/o Y)

**Scale(x,y)** Realiza un escalado sobre el elemento. El primer valor indica el factor de escalado horizontal (1 es el 100% y 5 escala al 50%) y el segundo el escalado vertical. (X y/o Y)

**scaleX(x)** Escala el elemento sólo en horizontal.

**scaleY(y)** Escala el elemento sólo en vertical.

---



**rotate(ángulo)**. Rota el elemento el número de grados indicados

**skewX(angle)**. Estira el elemento en el eje horizontal

**skewY(angle)**. Estira el elemento en el eje vertical

**skew(ánguloX, ánguloY)**. Estira el elemento usando un ángulo en horizontal y otro en vertical.

Amplia conocimientos:

CSS 2D Transforms [https://www.w3schools.com/css/css3\\_2dtransforms.asp](https://www.w3schools.com/css/css3_2dtransforms.asp)

---

# 3D

Permiten modificar la apariencia y posición de elementos en un espacio tridimensional mediante la propiedad **transform**

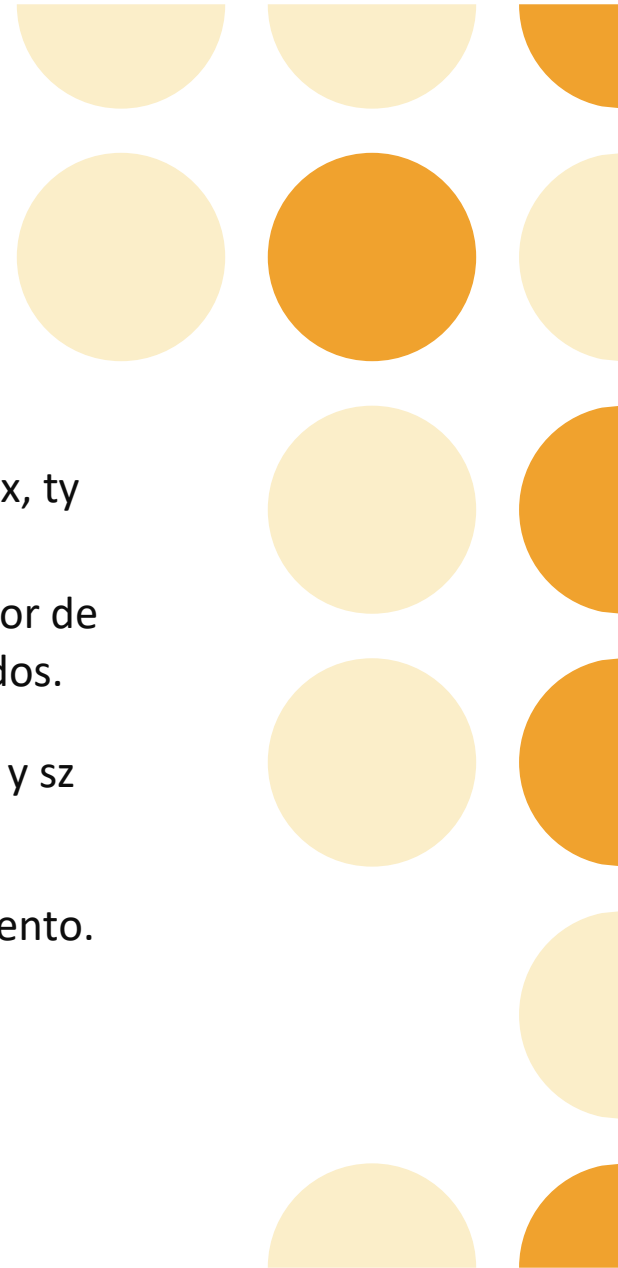
**translate3d(tx, ty, tz)**: Mueve un elemento en los ejes X, Y y Z. Los valores tx, ty y tz representan los desplazamientos en cada eje.

**rotateX(angle), rotateY(angle), rotateZ(angle)**: Rota un elemento alrededor de los ejes X, Y o Z. El parámetro angle especifica el ángulo de rotación en grados.

**scale3d(sx, sy, sz)**: Escala un elemento en los ejes X, Y y Z. Los valores sx, sy y sz representan los factores de escala en cada eje.

**perspective(depth)**: Especifica la perspectiva desde la cual se ve un elemento. Cuanto mayor sea el valor de depth, más pronunciado será el efecto de perspectiva

---



```

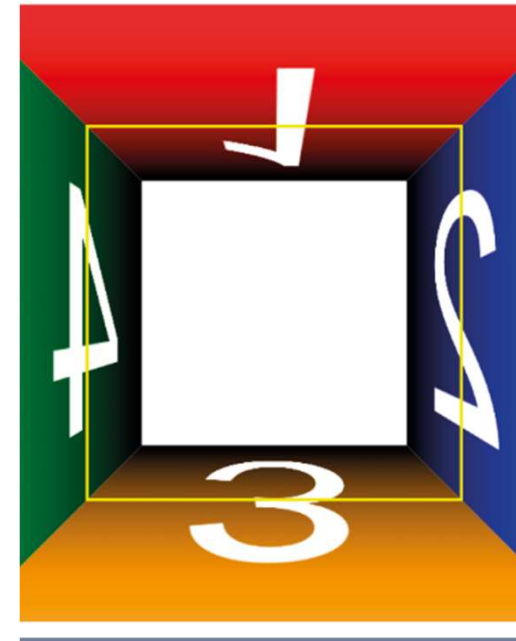
<!doctype html>
<html lang="es">
<head>
  <meta charset="UTF-8"> <title>Ejemplos 3D</title>
  <style>
    #contenedor{
      position:fixed;
      left:400px;
      top:400px;
      width:400px;
      height:400px;
      border:4px solid yellow;
      transform-style: preserve-3d;
      perspective-origin: center center;
      perspective:500px;
    }
    #contenedor div{
      position:absolute;
      width:100%;
      height:100%;
      left:0;
      top:0;
      font-size:350px;
      text-align: center;
      color:white;
    }
    #capa1{
      background:linear-gradient(black, red);
      transform:rotateX(90deg) translateZ(200px);
    }

```

```

    #capa2{
      background:linear-gradient(-90deg,black, blue);
      transform:rotateY(90deg) translateZ(200px);
    }
    #capa3{
      background:linear-gradient(0deg,black, orange);
      transform:rotateX(-90deg) translateZ(200px);
    }
    #capa4{
      background:linear-gradient(90deg,black, green);
      transform:rotateY(-90deg) translateZ(200px);
    }
  </style>
</head>
<body>
  <div id="contenedor">
    <div id="capa1">1</div>
    <div id="capa2">2</div>
    <div id="capa3">3</div>
    <div id="capa4">4</div>
  </div>
</body>
</html>

```



Amplia conocimientos:

CSS 3D Transforms

[https://www.w3schools.com/css/css3\\_3dtransforms.asp](https://www.w3schools.com/css/css3_3dtransforms.asp)

# Transiciones

The background is a blue gradient with a network of white dots and lines. A dark blue rectangle on the left contains the title. On the right, there are several white circles of different sizes and one solid orange circle.



Permite pasar de un estado del elemento a otro haciendo un paso gradual en el tiempo que se configure. Por ejemplo, podemos hacer que una capa pase a doblar su tamaño poco a poco.

Este tipo de efectos se solían hacer con JavaScript, ahora es posible desde CSS aunque desde luego no con todas las posibilidades de JavaScript

Permiten suavizar los cambios de estilo de un elemento cuando cambia su estado, como por ejemplo al pasar el cursor sobre él (:**hover**).

Se definen con la propiedad **transition** y pueden controlar la duración, el retardo, la función de tiempo y otras propiedades. Las transiciones pueden aplicarse a propiedades como **color**, **background-color**, **opacity**, **width**, **height**, entre otras.

Su sintaxis básica es:

**transition: propiedad duración función-de-tiempo retraso;**

---



**transition:** **propiedad duración función-de-tiempo retraso;**

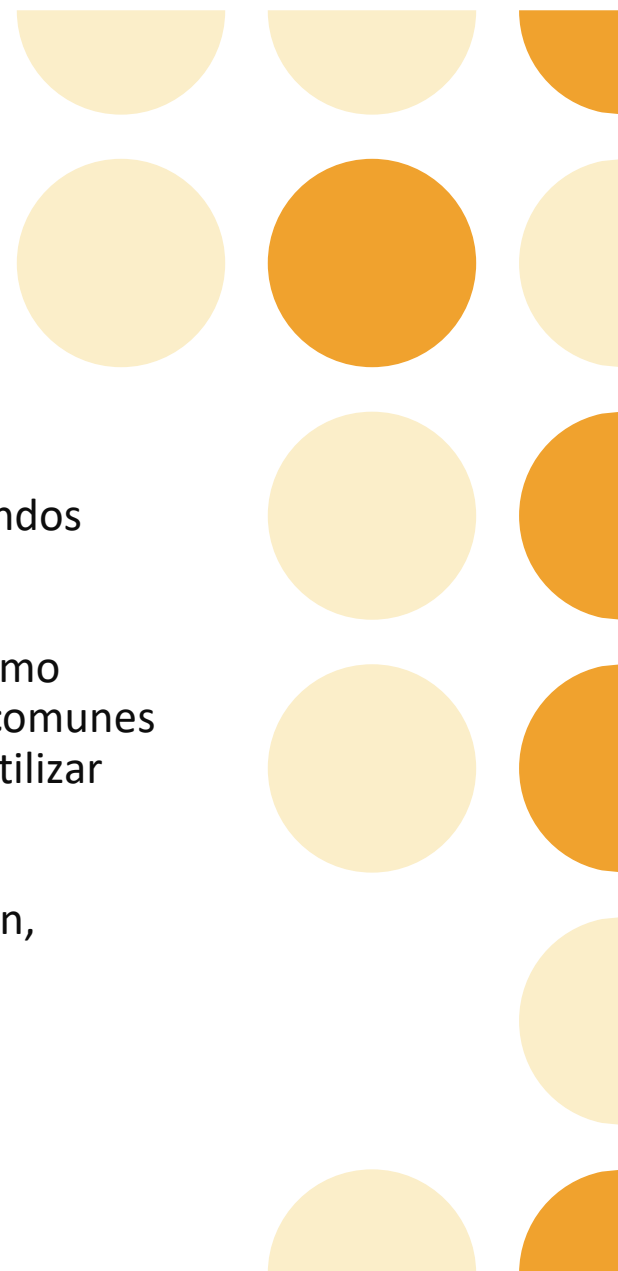
**propiedad:** Especifica la propiedad CSS que se desea animar, como **color**, **background-color**, **opacity**, **width**, **height**, etc.

**duración:** Especifica la duración de la transición en segundos (s) o milisegundos (ms).

**función-de-tiempo** (opcional): Define la función de tiempo que controla cómo cambia la velocidad de la transición a lo largo del tiempo. Algunos valores comunes son **ease**, **linear**, **ease-in**, **ease-out**, **ease-in-out**, pero también se pueden utilizar funciones de tiempo personalizadas.

**retraso** (opcional): Especifica un retraso antes de que comience la transición, también en segundos (s) o milisegundos (ms)

---



Las funciones de tiempo pueden ser:

**linear**. Es la que se emplea por defecto, la transición tiene siempre la misma velocidad.

**ease**. La transición frena la velocidad al final.

**ease-out**. La velocidad de la transición va poco a poco disminuyendo.

**ease-in**. La velocidad de la transición empieza lenta y luego va acelerando.

**ease-in-out**. La velocidad empieza frenada, acelera y vuelve a frenar al final.

Amplia conocimientos:

CSS Transitions

[https://www.w3schools.com/css/css3\\_transitions.asp](https://www.w3schools.com/css/css3_transitions.asp)

---

# Animaciones

---



Las animaciones son parecidas a las transiciones con la diferencia de que tienen la posibilidad de que se ejecuten directamente, sin hacer ningún tipo de evento.

Son más potentes (y más complejas de crear) que las transiciones, y tienen una gran conexión con el lenguaje JavaScript.

Las animaciones requieren primero de un bloque **@keyframes** que es el encargado de establecer las propiedades de la animación y de dar nombre a la misma.

Dentro de ese bloque se establecen dos elementos:

**from** Con las propiedades iniciales del elemento al que se aplicas la animación

**to** Con las propiedades finales

Su sintaxis básica es:

```
@keyframes nombreDeLaAnimacion {  
  from { /* Estilos del elemento en el inicio de la animación */ }  
  to { /* Estilos del elemento al final de la animación */ } }
```

---

También se pueden especificar los puntos clave (keyframes) intermedios usando porcentajes:

```
@keyframes nombreDeLaAnimacion {  
  0% { /* Estilos del elemento al principio de la animación */ }  
  50% { /* Estilos del elemento a la mitad de la animación */ }  
  100% { /* Estilos del elemento al final de la animación */ }  
}
```

Después de definir la animación con **@keyframes**, se puede aplicar a un elemento utilizando la propiedad **animation** con el nombre de la animación y otras opciones de configuración, como la duración, el retardo, la repetición y la dirección de la animación.

La sintaxis básica de animation es:

```
selector { animation: nombre-de-la-animacion duracion tiempo-funcion retraso repeticion direccion llenado; }
```

---

**nombre-de-la-animacion:** Especifica el nombre de la animación definida con @keyframes

**duracion:** Especifica la duración de la animación en segundos (s) o milisegundos (ms).

**tiempo-funcion:** Define la función de tiempo que controla cómo cambia la velocidad de la animación a lo largo del tiempo. Puede ser ease, linear, ease-in, ease-out, ease-in-out o una función personalizada.

**retraso** (opcional): Especifica un retraso antes de que comience la animación, también en segundos (s) o milisegundos (ms)

**repeticion** (opcional): Especifica el número de veces que se repetirá la animación (1, infinite para repetición infinita, o un número entero)

---

**direccion** (opcional): Especifica la dirección de la animación:

**normal:** La animación se ejecuta de principio a fin y luego se repite si está configurada para repetirse

**reverse:** se ejecuta al revés, es decir, desde el estado final al estado inicial. Luego se repite en la dirección opuesta si está configurada para repetirse

**alternate:** se ejecuta de principio a fin y luego al revés, alternando entre ambas direcciones en cada ciclo. Si está configurada para repetirse, este patrón se mantiene en cada ciclo

**alternate-reverse:** Similar a **alternate**, pero comienza con la ejecución al revés, es decir, desde el estado final al estado inicial, y luego va hacia adelante en cada ciclo

**llenado** (opcional): Especifica cómo se aplicarán los estilos antes y después de la animación

**none:** No se aplicarán estilos antes o después de la animación

**forwards:** Los estilos finales de la animación se aplicarán después de que la animación haya terminado. Es decir, el elemento mantendrá los estilos de la última keyframe de la animación

**backwards:** Los estilos iniciales de la animación se aplicarán antes de que comience la animación. Esto significa que el elemento se verá como si estuviera en el primer keyframe de la animación antes de que comience la reproducción

**both:** Combina forwards y backwards, por lo que los estilos iniciales se aplicarán antes de que comience la animación y los estilos finales se aplicarán después de que la animación haya terminado

Amplia conocimientos: CSS Animations [https://www.w3schools.com/css/css3\\_animations.asp](https://www.w3schools.com/css/css3_animations.asp)

---



The background is a dark, gradient surface. On the left, several bright, parallel neon lines in shades of cyan and magenta extend from the bottom towards the top right, creating a sense of depth. On the right side, there are several white-outlined circles of varying sizes, some of which are partially cut off by the edge of the frame. A solid orange circle is also visible. A thin white horizontal line is positioned near the bottom right.

**Web Responsive**

“**Media queries**” (consultas de medios) es una técnica en CSS que permite aplicar estilos diferentes según las características del dispositivo o el medio en el que se está visualizando el contenido.

Con las “media queries”, se puede adaptar el diseño de tu página web para que se vea bien en diferentes dispositivos, como pantallas de ordenador, tablets o smartphones.

Las “media queries” se basan en condiciones que evalúan características como el ancho de la pantalla (**width**), la orientación (**orientation**), la resolución del dispositivo (**resolution**), la densidad de píxeles (**pixel density**), entre otras.

Dependiendo de estas condiciones, se pueden establecer reglas de estilo específicas que se apliquen solo cuando se cumpla la condición especificada.

Por ejemplo, se puede utilizar media queries para aplicar un diseño de página diferente en dispositivos móviles, ajustar el tamaño de los elementos según el ancho de la pantalla o incluso ocultar ciertos elementos en impresiones de páginas

---



La sintaxis básica es:

```
@media tipo_de_medio y (condición) {  
    /* Reglas de estilo que se aplicarán si la condición es verdadera */  
}
```

**@media**: Esta es la directiva @ que indica que se está utilizando una regla de media query.

**tipo\_de\_medio**: Especifica el tipo de medio al que se aplicarán los estilos. Puede ser **all** (todos los dispositivos), **screen** (pantallas), **print** (impresoras), **speech** (sintetizadores de voz) u otros.

**(condición)**: Especifica la condición que debe cumplirse para que se apliquen los estilos dentro de la media query. Puede ser una combinación de características como **min-width**, **max-width**, **orientation**, **min-resolution**, etc.

Amplia conocimientos. Responsive Web Design Media Queries  
[https://www.w3schools.com/css/css\\_rwd\\_mediaqueries.asp](https://www.w3schools.com/css/css_rwd_mediaqueries.asp)

---

