


# UD02. CSS. Transformaciones

Diseño de interfaces WEB  
2º Curso

- 
- Transformaciones
  - Transformaciones dinámicas
  - Transiciones

# Transformaciones

A partir de CCS3 es posible modificar parte de las características físicas de un elemento. Entre esas características se encuentran la posición, la rotación, la escala o es sesgo.

Estas transformaciones se producen de manera previa al renderizado, es decir, en pantalla aparece el elemento ya

Matemáticamente estas operaciones se basan en el producto de vectores y matrices.

# Transformaciones

## Desplazamiento

Es la transformación más sencilla. Consiste en sumar o restar píxeles a la posición en la que el elemento se encuentra actualmente.

🔊 Se pueden utilizar coordenadas relativas como em, ex, o % pero no es aconsejable. En caso de querer usarse es más recomendable el uso de coordenadas relativas al tamaño de la vista (viewport) vw, vh, vmin, vmax que estudiaremos en diseño responsive

```
#miDiv { transform: translate(200px,100px); }
```

Existen funciones para tratar cada uno de los desplazamientos por separado

```
#miDiv {  
    transform: translateX(200px);  
    transform: translateY(100px);  
}
```

# Transformaciones

## Rotación

El eje de giro está en el centro del elemento (por ejemplo el div). Ojo!puede no coincidir con el eje del dibujo en sí.

El valor se da en grados sexagesimales (0-360).

Signo positivo → sentido de las agujas del reloj

Signo positivo → sentido antihorario

```
#miDiv { transform: rotate(45deg); }
```

# Transformaciones

## Escala

Esta propiedad escala el elemento en los dos ejes, tomando como eje de referencia el centro del elemento

El valor se indica en tanto por 1, es decir un valor mayor que uno implica un aumento de escala mientras que un valor menor de 1 indica una reducción.

```
#miDiv { transform: scale(2,1.5); }
```

# Transformaciones

## Sesgo

Esta propiedad deforma el objeto rotándolo en los ejes X e Y. Ambos ejes son ortogonales entre sí y pasan por el centro del elemento

Aplicando un skew en X la sensación visual es que estamos tumbando el elemento, mientras que en Y parece que lo acercamos o alejamos del punto de vista

```
#miDiv { transform: skew(20deg,15deg); }
```

Existen funciones para tratar cada uno de los giros por separado

```
#miDiv {  
    transform: skewX(20deg);  
    transform: skewY(10deg);  
}
```

# Transformaciones

## Simultáneas

Es posible indicar que las diferentes operaciones se realicen de manera simultánea

```
#miDiv { transform: translate(200px,100px) rotate(45deg)  
skew(20deg,15deg); }
```

OJO. En este caso el orden de los factores si altera el resultado. No es lo mismo desplazar y girar, por ejemplo, que girar y desplazar



# Transformaciones

## Dinámicas

Es posible combinar estas operaciones con pseudoclases de manera que consigamos pequeñas “animaciones”

```
#miDiv:hover { transform: rotate(10deg); }
```

# Transiciones

Son modificaciones que podemos dar a ciertas propiedades (incluyendo las transformaciones) indicando el tiempo en el que queremos que se produzcan.

Podemos también indicar el tipo de movimiento (opcional) así como el retardo (el tiempo que hay que esperar hasta que se inicie) (opcional)

```
#miDiv { transition: width 2s linear 1s; }
```

Es posible indicar varias transiciones para que sucedan de manera simultánea.

```
#miDiv { transition: width 2s, height 3s; }
```

# Transiciones

## Propiedades

Las propiedades que podemos modificar mediante transiciones son:

all  
background-color  
border  
border-radius  
color  
top  
bottom  
left  
right  
box-shadow  
width  
height

line-height  
margin  
opacity  
word-spacing  
letter-spacing  
fill  
padding  
stroke  
text-shadow  
vertical-align  
visibility  
z-index

# Transiciones

## Tipo de animación

Indica las velocidades que tendrá la transición

- *linear*: La velocidad de la animación es uniforme en todo el recorrido.
- *ease*: La velocidad se acelera al inicio, luego se retarda un poco y se acelera al final de nuevo.
- *ease-in*: La animación empieza lentamente y va aumentando progresivamente.
- *ease-out*: La animación empieza muy rápida y va descendiendo progresivamente.
- *ease-in-out*: La animación empieza y acaba lentamente, y es en la parte central del recorrido donde la velocidad es más rápida.

# Transiciones

## Punto inicial y final

Obviamente para tener una transición necesitamos unos valores de inicio y otros de fin.

- Los valores de inicio son los que el sistema (o nuestro CSS) defina.
- Los valores finales son cualquier otros que acontezcan por alguna razón. Por ejemplo, un *hover* o un cambio de valor de una propiedad mediante JS por haber realizado alguna acción

# Compatibilidad

Tanto las transformaciones como las transiciones pueden tener problemas de compatibilidad entre diferentes navegadores.

- Para alcanzar la mayoría de ellos es recomendable utilizar los prefijos de cada navegador (-webkit-, -moz-, -o-, -ms-)
- Es conveniente conocer usar una tabla de compatibilidad para saber qué y donde podemos usar cada propiedad

<http://caniuse.com/>