DISEÑO DE INTERFACES WEB UD-2

ANIMACIONES

ÍNDICE

[TRANSFORMACIONES Y TRANSICIONES 3](#_Toc147419448)

[A. TRASLACIONES 3](#_Toc147419449)

[B. ROTACIÓN 4](#_Toc147419450)

[C. ESCALADO 4](#_Toc147419451)

[D. SESGO 5](#_Toc147419452)

[E. AGRUPACIONES 6](#_Toc147419453)

[USO DE PSEUDOCLASES 6](#_Toc147419454)

[TRANSICIONES 7](#_Toc147419455)

# ANIMACIONES

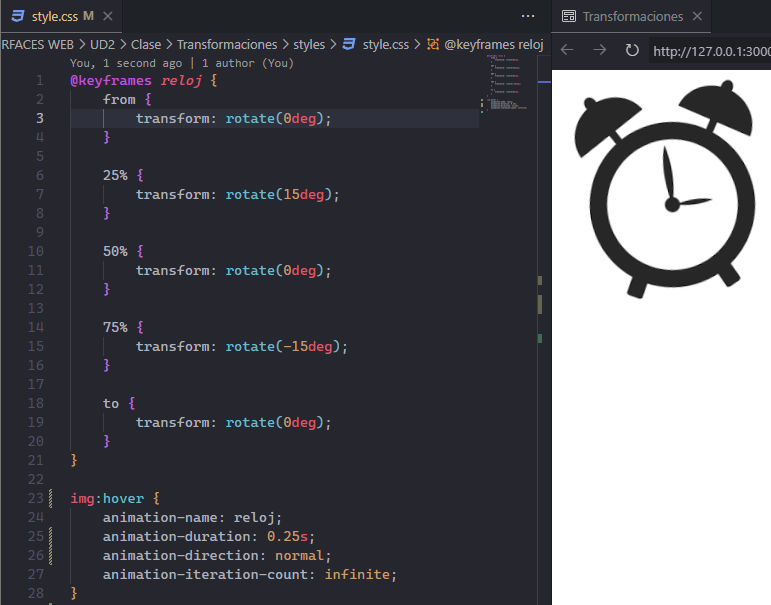
Las transformaciones devuelven elementos estáticos, que mediante las pseudoclases como hover podemos hacerlas dinámicas.

Las transiciones aportan más dinamismo, pero con las animaciones podemos conseguir mejores efectos.

Se puede considerar una animación a una agrupación de transiciones.

## ANIMACIONES ESTÁTICAS

Una animación se compone de keyframes (fotogramas clave), que son imágenes estáticas de la animación en momentos determinados. Los keyframes que siempre deben existir son el inicial y el final de la animación. Se declaran así, para una imagen de un reloj, por ejemplo:



En la animación lo que se va a conseguir es un balanceo del reloj imitando al movimiento que tienen cuando suenan.

En los porcentajes del keyframe podemos utilizar cualquier valor (entre 0 y 100%) e incluso hacer intervalos tipo (65%, 75%). Hay que tener en cuenta que el from es el 0% y el to es el 100%

En las propiedades de la animación tenemos:

* animation-name: Es el nombre de la animación y, por tanto, del keyframe que hayamos creado.
* animation-duration: Duración de la animación.
* animation-direction: La dirección que tendrá la animación.
  + Normal: hará que vaya de from a to y en el siguiente ciclo empezará de nuevo por from.
  + Alternate: hará que vaya de from a to y luego volverá del to al from.
  + Reverse: hará que vaya del to al from y en el siguiente ciclo empezará de nuevo por el to.
  + Alternate-reverse: hará que vaya del to al from y luego volverá del from al to.
* animation-iteration-count: Es el número de repeticiones que tendrá la animación.

## ANIMACIONES CON JAVASCRIPT

Otra forma de crear animaciones en nuestro HTML es a través de eventos de JS, por ejemplo, vamos a añadir un botón a nuestro HTML el cual actuará como disparador de un evento que cambie la clase del img.

Texto

Descripción generada automáticamente

Cuando se cargue la página se buscará el elemento con id “run”, en nuestro caso nuestro botón, y se le asignará un escuchador, que cuando se active lanzará la función accRun().

Esta función buscará el elemento con id “reloj”, en nuestro caso, nuestra imagen de reloj, y le asignará la clase “animaReloj”, que será en la que tengamos configurada la animación:

Vamos a meter otra imagen en nuestra ventana y vamos a hacer que cuando acabe la animación del primer reloj empiece la del segundo:

Texto

Descripción generada automáticamente

Se añade un escuchador sobre el primer reloj, para que cuando termine la animación (“animationend”) se lance la función accRun2, que localiza el elemento con id “reloj2” y le añade la clase animaReloj, iniciando la animación del segundo reloj.

## ANIMACIONES CON PSEUDOELEMENTOS

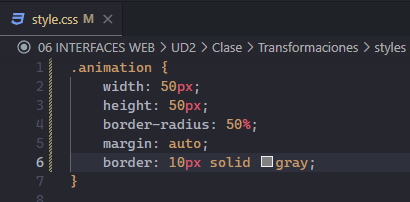
Vamos a crear una animación de una circunferencia.

Para crearla, partimos de un bloque div de clase animación:

Texto

Descripción generada automáticamente

Y le damos el siguiente estilo a esa clase:



Y con esto generaremos el siguiente objeto en pantalla:

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

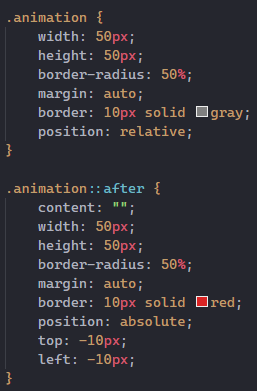
Primero hacemos el div cuadrado igualando ancho y alto y luego le damos un border-radius del 50% para hacerla un círculo.

Cuando definimos el pseudoelemento “after” queremos añadir un círculo igual, pero para que se visualice tenemos que añadir contenido al after (por eso ponemos content: “”)

Luego tenemos que ajustar el posicionamiento de ambas reglas: en el principal lo ponemos como relative para que sea el ancestro del after cuando pongamos éste como posición absoluta:

Por último, tenemos que ajustar el top y left del after para compensar el border del principal, que está a 10.

Icono

Descripción generada automáticamente

Vamos ahora a hacer transparente el borde del after y ponemos sólo rojo el cuadrante superior:

Icono

Descripción generada automáticamenteTexto

Descripción generada automáticamente

Ahora para mover el cuadrante a lo largo del borde creamos unos keyframes:

Una captura de pantalla de un celular

Descripción generada automáticamente

Imagen que contiene Círculo

Descripción generada automáticamenteY al final del after metemos el código de la animación:

Texto

Descripción generada automáticamente

El mismo resultado se podría obtener incluyendo un div anidado de clase “animacion2” y con un estilo similar al de animación, y con display: inline.

Pero se hace con before y after porque toda la animación se define en el css (se centraliza todo ahí) y en el html sólo se necesita un div. Y tampoco hace falta cambiar el display del elemento, que puede resultar confuso.