

## 1. ANIMACIONES

### ✓ FOTOGRAMA CLAVE

Valores iniciales y finales que debe tener la animación CSS. Siempre conocemos en qué punto vamos a empezar y en cual vamos a terminar la animación, así como su duración.

El código sería el siguiente:

#### 1º) Definir animación con sus atributos.

```
#clase{ (También se puede aplicar a etiqueta)
  animation-name: 'nombre-fotograma-clave';
  animation-duration: 45s;
  animation-iteration-count: 10;
}
```

Los atributos de estilo para esta capa que se ven en el código anterior son los siguientes:

- **animation-name:** el nombre del fotograma clave.
- **animation-duration:** la duración de la animación.
- **animation-iteration-count:** las veces que se repite.

Otros atributos:

- **animation-direction:** esta propiedad define el sentido de la animación. Si especificamos “alternate”, se realizará en la dirección inversa.
- **animation-delay:** propiedad que nos indica el momento en el que comenzará la animación. Si el valor es 0 se ejecuta en cuanto se carga la página.
- **animation:** esta propiedad combina las anteriores de una forma resumida.

Siguiendo el orden: **name duration direction count delay**

#### 2º) Definir la animación con fotograma clave.

```
@keyframes 'nombre_fotograma_clave' {
  especificar el movimiento que realiza el elemento con
    from { posición inicial}
    to {posición final}
  o definir los fotogramas con %
}
```

- **Con from to:**

```
@keyframes 'movimiento-diagonal' {
  from {
    left: 0px;
    top: 0px;
  }
  to {
    left: 100px;
    top: 100px;
  }
}
```

Ej1Animacion0.html  
Ej1Animacion1.html

- **Con porcentajes**

El elemento se mueve desde la posición 0,0 hasta la posición 100,100 dentro del div en el que se encaje.

```
@keyframes 'fotograma_con_porcentajes' {  
  0% {  
    left: 100px;  
  }  
  40% {  
    left: 150px;  
  }  
  60% {  
    left: 75px;  
  }  
  100% {  
    left: 100px;  
  }  
}
```

Esta animación estaría compuesta de 4 fotogramas clave, el porcentaje es en el momento de la animación en el que va a producirse ese fotograma y los px son la longitud y la alineación donde se colocaría el fotograma dentro del div en que se encaje.

### **Ej1Animacion2.html**

Se pueden añadir las propiedades:

- **animation-timing-function:** se aplica entre los fotogramas clave, no sobre toda la animación y describe como progresa la animación a lo largo de un ciclo.

Los valores posibles para la propiedad animation-timing-function:

- ease : Define un efecto de animación con un comienzo lento, luego rápido y finalmente termina lento (cuando no definimos la función elige esta por defecto)
- linear : Define un efecto de animación con la misma velocidad de inicio a fin.
- ease-in : Define un efecto de animación con un comienzo lento.
- ease-out : Define un efecto de animación con un final lento.
- ease-in-out : Define un efecto de animación con un comienzo lento y un final lento.

Probar los ejemplos 1 y 2 con distintos valores de animation-timing-function.

## 2. TRANSICIONES

### ✓ TRANSITION (UNA PROPIEDAD)

Las transiciones en css3 permiten modificar el valor de una propiedad de un elemento HTML en forma gradual durante un tiempo determinado de un estado inicial a un estado final.

La sintaxis más simple para definir una transición de una propiedad es:

```
Elemento {  
    transition: [nombre de propiedad] [duración de la transición] ;  
}
```

El nombre de la propiedad puede ser un atributo (width, color, background...) o una transformación (se verá en el punto 3)

La propiedad varía cuando el elemento tiene otro comportamiento. Por ejemplo al señalar (hover)

### ✓ TRANSITION (VARIAS PROPIEDADES)

```
Elemento {  
    transition: [nombre de propiedad] [duración de la transición],  
        [nombre de propiedad] [duración de la transición],  
        [nombre de propiedad] [duración de la transición] ;  
}
```

**Ej2Animacion 1TransicionSimple.html**

### ✓ FUNCIONES DE TRANSICIÓN

```
Elemento {  
    transition: [nombre de propiedad] [duración de la transición] [función de  
    transición];  
}
```

Los valores posibles que podemos especificar son:

ease : Define un efecto de transición con un comienzo lento, luego rápido y finalmente termina lento (cuando no definimos la función de transición elige esta por defecto)

linear : Define un efecto de transición con la misma velocidad de inicio a fin.

ease-in : Define un efecto de transición con un comienzo lento.

ease-out : Define un efecto de transición con un final lento.

ease-in-out : Define un efecto de transición con un comienzo lento y un final lento.

### ✓ TIEMPO DE DEMORA EN INICIAR LA TRANSICIÓN

```
Elemento {  
    transition: [nombre de propiedad] [duración de la transición]  
        [función de transición] [tiempo de inicio];  
}
```

**Ej2Animacion 2TransicionFunciones.html**

### 3. TRANSICIONES Y TRANSFORMACIONES

*transform: función de transformación*

La función de transformación puede ser:

- rotate(valor)
- translate(valorx.valory)
- scale(valorx,valory)

Si valorx es igual a valory solo se pone uno.

#### **Ej3Animacion 1Transform.html**

Se añade en Ej1Animacion0 la transformación:

```
transform: rotate(180deg);
```

#### **Ej3Animación 2TransicionTransform.html**

Se añaden en Ej3Animacion\_2TransicionFunciones.html las siguientes

transformaciones:

```
transform: scale(1.5) rotate(180deg);
```

## 4. TRANSFORMACIONES 2D

```
@keyframes nombre {  
    transform: función de transformación(valor(es));  
}
```

### ✓ ROTACIÓN (TRANSFORM: ROTATE)

```
@keyframes nombre { transform: rotate(valor);}
```

La función de rotación se llama rotate y tiene un parámetro que indica la cantidad de grados a rotar. La rotación es en el sentido de las agujas del reloj. Podemos indicar un valor negativo para rotar en sentido antihorario.

```
@keyframes gira {  
    transform: rotate(180deg);  
}
```

#### Ej4Animacion Rotate

### ✓ ESCALADO (TRANSFORM: SCALE)

Otra función de transformación 2D es el escalado, esta función permite agrandar o reducir el tamaño del elemento. Sintaxis que podemos utilizar es la siguiente:

```
@keyframes nombre { transform: scale(valorx ,valory);}
```

Con valor menor que 1 decrece y con valor mayor que 1 crece.

Un recuadro escalado con 20% menos de ancho y 20% más de alto

```
@keyframes escalar {  
    transform: scale(0.8 , 1.2);  
}
```

#### Ej4Animacion Scale

### ✓ TRANSLACIÓN (TRANSFORM: TRASLATE)

La función translate permite desplazar un elemento HTML indicando una medida (en porcentaje, píxeles, em, etc.) la sintaxis es:

```
@keyframes nombre { transform: translate(valorx ,valory);}
```

Si queremos trasladar 25 píxeles en "x" y 10 píxeles en "y"

```
@keyframes trasladar {  
    transform: translate(25px,10px);  
}
```

#### Ej4Animacion Translate