

# Unidad 6: Animación en SVG

## Animación en CSS

### **@keyframes** *miAnimacion*{}

Declara los pasos que tendrá una animación, que aplicaremos luego a un elemento. Cada paso se indica entre llaves {}, y puede emplear palabras clave como **from**{ } (desde) y **to**{ } (hasta), o usar porcentajes para indicar más pasos, como **0%**{ } **20%**{ } ...

### **animation:** *nombreAnimacion duración ...;*

Propiedad abreviada para asignar una animación a un elemento, en la que podemos encadenar varios valores de animación: nombre duración aceleración retraso repetición dirección *fill-mode play-state*;

### **animation-name:** *nombreAnimacion;*

Asignación a un elemento de una animación definida anteriormente mediante **@keyframes**.

### **animation-duration:** *0s;*

Duración de la animación, en segundos (s) o milisegundos (ms).

### **animation-timing-function:** *ease;*

Aceleración de la animación. Por defecto es **ease**. Puedes consultar las curvas de velocidad por defecto y crear una propia en <http://cubic-bezier.com>

### **animation-timing-function:** *steps(5);*

La posibilidad de fraccionar la animación en varios pasos y que no necesariamente exista transición entre ellos. Es útil para realizar una animación de tipo *sprites*, entre otras cosas.

### **animation-delay:** *0s;*

Retraso en ejecutar la animación. Es útil para sincronizar animaciones.

### **animation-iteration-count:** *1;*

Número de repeticiones de la animación. Una animación continua se consigue con el valor **infinite**.

### **animation-direction:** *normal;*

Dirección en la que será reproducida la animación. Algunos valores posibles son **reverse** (se reproduce al revés) o **alternate** (cuando ha terminado, se reproduce en

dirección contraria). Este último valor es interesante para crear una animación que se repita y que resulte más fluida.

**animation-fill-mode: none;**

Por defecto, un elemento animado vuelve a su estado inicial cuando la animación ha terminado. Con esta propiedad, podemos asignarle el valor **forwards** para que mantenga los estilos del último paso de la animación.

**animation-play-state: running;**

Una animación está activa (**running**) por defecto, pero mediante esta propiedad podría pausarse (**paused**) en momentos concretos.

## ***Animación de dibujado***

Animación que aprovecha dos propiedades del stroke para dar la impresión de que se auto-dibuja. Conviene preparar el dibujo adecuadamente para conseguir un buen efecto, ya

**stroke-dasharray="100"**

Atributo que provoca un borde discontinuo y controla la longitud de los tramos de dicho borde. Un **stroke-dasharray** lo bastante largo rodeará un dibujo con un solo tramo, dando la impresión de no existir discontinuidad.

En CSS sería **stroke-dasharray: 100px;**

**stroke-dashoffset="100"**

Atributo que controla el punto en el que empieza el **stroke-dasharray**, dejando un espacio sin borde equivalente al valor que demos. Un valor lo bastante amplio cubrirá todo el dibujo, dando la impresión de no existir ningún borde. Este es el valor que animaremos, y a medida que el valor se acerque a 0, tendremos la impresión de que el elemento se dibuja.

En CSS sería **stroke-dashoffset: 100px;**

En este artículo se dan algunos consejos para conseguir un mejor resultado en las animaciones de dibujado, para SVG diseñados desde Illustrator: <http://valhead.com/2017/03/03/three-illustrator-tricks-for-better-svg-stroke-animations>

## Eventos CSS para crear interacción

En los ejemplos de interacción sencilla se han utilizado eventos de CSS. Estos eventos tienen efecto en el SVG si está incorporado en la web de manera adecuada, como vimos en la Unidad 4.

### **:hover**

Este evento CSS se lanza cuando el usuario pase el ratón por encima del elemento.

### **:checked**

Este evento CSS se lanza cuando el usuario activa un **input** de dos estados (**radio** o **checkbox**). Se puede emplear un selector de adyacencia (+) para manipular el elemento que venga a continuación de dicho **input**.

## SMIL

Este método de animación no es recomendable si nos preocupa el soporte entre navegadores (es preferible JavaScript). Aún así, tiene algunas ventajas interesantes, como la animación a lo largo de un trazo o el morphing entre formas.

La forma de emplearlo es introducir las etiquetas con la información de la animación en la etiqueta a animar, por ejemplo,

```
<circle> <!--etiquetas de animación --> </circle>.
```

```
<animate attributeName="..." from="0" to="100" dur="3s"
repeatCount="indefinite" />
```

Animación del atributo indicado en **attributeName** para el elemento que contiene la etiqueta **<animate>**, teniendo como punto inicial el atributo **from** y punto final en el atributo **to**. La duración está definida en el atributo **dur** y que se repite tantas veces como indique el atributo **repeatCount**.

```
<animateMotion path="..." dur="3s" repeatCount="indefinite" />
```

Animación que se realiza a través de un trazo definido en el atributo **path**.

Puedes encontrar más información sobre SMIL en [https://developer.mozilla.org/es/docs/Web/SVG/SVG\\_animation\\_with\\_SMIL](https://developer.mozilla.org/es/docs/Web/SVG/SVG_animation_with_SMIL)