

Animaciones CSS

Las animaciones CSS3 permiten animar la transición entre un estilo CSS y otro. Las animaciones constan de dos componentes: un estilo que describe la animación CSS y un conjunto de fotogramas que indican su estado inicial y final, así como posibles puntos intermedios en la misma.

Las animaciones CSS tienen tres ventajas principales sobre las técnicas tradicionales de animación basada en scripts:

1. Son muy fácil usar para animaciones sencillas, puedes hacerlo incluso sin tener conocimientos de Javascript.
2. La animación se muestra correctamente, incluso en equipos poco potentes. Animaciones simples realizadas en Javascript pueden verse mal (a menos que estén muy bien echas). El motor de renderizado puede usar técnicas de optimización como el "frame-skipping" u otras técnicas para mantener que la animación se vea tan suave como sea posible.
3. Al ser el navegador quien controle la secuencia de la animación, permitimos que optimice el rendimiento y eficiencia de la misma, por ejemplo, reduciendo la frecuencia de actualización de la animación ejecutándola en pestañas que no estén visibles.

Configurando la animación

Para crear una secuencia de animación CSS, tu estilizarás el elemento que quieras animar con la propiedad `animation` y sus sub-propiedades. Con ellas podemos no solo configurar el ritmo y la duración de la animación sino otros detalles sobre la secuencia de la animación. Con ellas **no** configuramos la apariencia actual de la animación, para ello disponemos de `@keyframes` como describiremos más adelante .

Las subpropiedades de `animation` son:

`animation-delay`

Tiempo de retardo entre el momento en que el elemento se carga y el comienzo de la secuencia de la animación.

animation-direction

Indica si la animación debe retroceder hasta el fotograma de inicio al finalizar la secuencia o si debe comenzar desde el principio al llegar al final.

animation-duration

Indica la cantidad de tiempo que la animación consume en completar su ciclo (duración).

animation-iteration-count

El número de veces que se repite. Podemos indicar `infinite` para repetir la animación indefinidamente.

animation-name

Especifica el nombre de la regla `@keyframes` que describe los fotogramas de la animación.

animation-play-state

Permite pausar y reanudar la secuencia de la animación

animation-timing-function

Indica el ritmo de la animación, es decir, como se muestran los fotogramas de la animación, estableciendo curvas de aceleración.

animation-fill-mode

Especifica qué valores tendrán las propiedades después de finalizar la animación (los de antes de ejecutarla, los del último fotograma de la animación o ambos).

Definiendo la secuencia de la animación con fotogramas

Una vez configurado el tiempo de la animación, necesitamos definir su apariencia. Esto lo haremos estableciendo dos fotogramas más usando la regla `@keyframes`. Cada fotograma describe cómo se muestra cada elemento animado en un momento dado durante la secuencia de la animación. Desde que se define el tiempo y el ritmo de la animación, el fotograma usa `percentage` para indicar en qué momento de la secuencia de la animación

tiene lugar. 0% es el principio, 100% es el estado final de la animación. Debemos especificar estos dos momentos para que el navegador sepa dónde debe comenzar y finalizar; debido a su importancia, estos dos momentos tienen alias especiales: `from` y `to`.

Además, puedes, opcionalmente, incluir fotogramas que describan pasos intermedios entre el punto inicial y final de la animación.

Ejemplo

```
p {
  animation-duration: 3s;
  animation-name: slidein;
}

@keyframes slidein {
  from {
    margin-left: 100%;
    width: 300%
  }

  to {
    margin-left: 0%;
    width: 100%;
  }
}
```

El estilo del elemento `<p>` especifica, a través de la propiedad `animation-duration`, que la animación debe durar 3 segundos desde el inicio al fin y que el nombre de los `@keyframes` que definen los fotogramas de la secuencia de la animación es "slidein".

Si queremos añadir algún estilo personalizado sobre el elemento `<h1>` para usarlo en navegadores que no soporten animaciones CSS también podemos incluirlos. En nuestro ejemplo, no queremos ningún otro estilo personalizado diferente al efecto de la animación.

Los fotogramas se definen usando la regla `@keyframes`. En nuestro ejemplo, tenemos solo dos fotogramas. El primero de ellos sucede en el 0% (hemos usado su alias `from`). Aquí, configuramos el margen izquierdo del elemento, poniéndolo al 100% (es decir, en el borde derecho del elemento contenedor), y su ancho al 300% (o tres veces el ancho del elemento contenedor). Esto hace que en el primer fotograma de la animación tengamos el encabezado fuera del borde derecho de la ventana del navegador.

El segundo (y último) fotograma sucede en el 100% (hemos usado su alias `to`). Hemos puesto el margen derecho al 0% y el ancho del elemento al 100%. Esto produce que el encabezado, al finalizar la animación, esté en el borde derecho del área de contenido.

Añadiendo otro fotograma

Vamos a añadir otro fotograma a la animación de nuestro ejemplo anterior. Pongamos que queremos que el tamaño de fuente del encabezado aumente a medida que se mueve durante un tiempo y que después disminuye hasta su tamaño original. Esto es tan sencillo como añadir este fotograma:

```
75% {  
  font-size: 300%;  
  margin-left: 25%;  
  width: 150%;  
}
```

Esto le dice al navegador que en el 75% de la secuencia de la animación, el encabezado tiene un margen izquierdo del 25%, un tamaño de letra del 200% y un ancho del 150%.

Haciendo que se repita

Para hacer que la animación se repita, solo hay que usar la propiedad `animation-iteration-count` e indicarle cuántas veces debe repetirse. En nuestro caso, usamos `infinite` para que la animación se repita indefinidamente:

```
p {  
  animation-duration: 3s;  
  animation-name: slidein;  
  animation-iteration-count: infinite;  
}
```

Moviendolo hacia adelante y hacia atrás

Hemos hecho que se repita, pero queda un poco raro que salte al inicio de la animación cada vez que ésta comienza. Queremos que se mueva hacia adelante y hacia atrás en la pantalla. Esto lo conseguimos fácilmente indicando que `animation-direction` es `alternate`:

```
p {  
  animation-duration: 3s;  
  animation-name: slidein;  
  animation-iteration-count: infinite;  
  animation-direction: alternate;  
}
```