



APRENDERAPROGRAMAR.COM

ANIMATION CSS. NAME,
DURATION, DELAY, FILL-
MODE, ITERATION-
COUNT, DIRECTION,
TIMING-FUNCTION, PLAY-
STATE. EJEMPLOS
(CU01065D)

Sección: Cursos

Categoría: Tutorial básico del programador web: CSS desde cero

Fecha revisión: 2029

Resumen: Entrega nº65 del Tutorial básico: "CSS desde cero".

Autor: César Krall

ANIMATION CSS

Mediante la regla CSS `@keyframes` podemos definir una animación con un nombre y los diferentes estados por los que pasará el elemento HTML que va a ser animado durante el tiempo que dure la animación. Conocida esta regla, vamos a ver cómo aplicarla a un elemento o grupo de elementos mediante el uso de un selector tipo id o class y la aplicación de las propiedades relacionadas con animation CSS.



PROPIEDAD ANIMATION-NAME

Mediante esta propiedad indicamos el nombre de animación definida en una regla `@keyframes` que va a ser aplicada a un elemento seleccionado mediante un selector CSS. El valor por defecto para esta propiedad es none (implica que no se aplicará ninguna animación). Si se desean aplicar varias animaciones, se especificarán sus nombres separadas por comas.

Ejemplo: `.pubMov{ animation-name: animacion1musk; }` esta regla indica que a los elementos con `class = "pubMov"` se les aplicará la animación de nombre `animacion1musk` que habrá sido definida en una regla `@keyframes`.

Otro ejemplo: `.pubMov{ animation-name: animacion1musk, animacion7kim; }` esta regla indica que a los elementos con `class = "pubMov"` se le aplicarán dos animaciones: la que lleva por nombre `animacion1musk` y la que lleva por nombre `animacion7kim`. Ambas deberán haber sido definidas en reglas `@keyframes`.

Esta propiedad puede no ser reconocida por los navegadores antiguos o requerir del uso de prefijos específicos para algunos navegadores actuales. Consulta en Mozilla Developer Network para conocer si debes aplicar prefijos.

PROPIEDAD ANIMATION-DURATION

Mediante esta propiedad indicamos el tiempo en segundos (s) o milisegundos (ms) que llevará completar un ciclo de la animación. La animación habrá sido definida en una regla `@keyframes` y va a ser aplicada a un elemento seleccionado mediante un selector CSS. El valor por defecto para esta propiedad es 0s (implica que no se aplicará ninguna animación ya que la duración es nula). Si se desean aplicar varias animaciones, se especificarán sus duraciones separadas por comas, siendo cada valor correspondiente a la animación cuyo nombre figura en el mismo orden en la propiedad `animation-name`.

Ejemplo: `.pubMov{ animation-name: animacion1musk; animation-duration: 5s;}` esta regla indica que a los elementos con `class = "pubMov"` se les aplicará la animación de nombre `animacion1musk` que habrá sido definida en una regla `@keyframes` y que esta animación deberá completarse en 5 segundos.

Otro ejemplo: `.pubMov{ animation-name: animacion1musk, animacion7kim; animation-duration: 5s, 250ms;}` esta regla indica que la primera animación deberá completarse en 5 segundos y la segunda en 250 milisegundos.

Esta propiedad puede no ser reconocida por los navegadores antiguos o requerir del uso de prefijos específicos para algunos navegadores actuales. Consulta en Mozilla Developer Network para conocer si debes aplicar prefijos.

PROPIEDAD ANIMATION-ITERATION-COUNT

Mediante esta propiedad indicamos el número de veces que debe repetirse la animación. La animación habrá sido definida en una regla `@keyframes` y va a ser aplicada a un elemento seleccionado mediante un selector CSS. El valor por defecto para esta propiedad es 1 (implica que la animación se ejecutará una sola vez y se detendrá). Sus valores posibles son cualquier número igual o superior a cero, admitiéndose decimales (por ejemplo se admite, 1, 2, 3, 3.1, 3.2, 3.5, 10, 20, etc.) ó `infinite`. Si se usa la palabra clave `infinite` la animación se repetirá indefinidamente sin parar.

Si se desean aplicar varias animaciones, se especificarán sus repeticiones separadas por comas, siendo cada valor correspondiente a la animación cuyo nombre figura en el mismo orden en la propiedad `animation-name`.

Ejemplo: `.pubMov{ animation-name: animacion1musk; animation-duration: 5s; animation-iteration-count:infinite;}` esta regla indica que a los elementos con `class = "pubMov"` se les aplicará la animación de nombre `animacion1musk` que habrá sido definida en una regla `@keyframes` y que esta animación deberá completarse en 5 segundos y que una vez completada volverá a repetirse y así continuamente.

Otro ejemplo: `.pubMov{ animation-name: animacion1musk, animacion7kim; animation-duration: 5s, 250ms; animation-iteration-count:infinite, 7;}` esta regla indica que la primera animación se repetirá continuamente mientras que la segunda se ejecutará 7 veces y se detendrá.

Esta propiedad puede no ser reconocida por los navegadores antiguos o requerir del uso de prefijos específicos para algunos navegadores actuales. Consulta en Mozilla Developer Network para conocer si debes aplicar prefijos.

PROPIEDAD ANIMATION-DIRECTION

Mediante esta propiedad indicamos cómo debe ejecutarse una animación: hacia delante (del principio al final), hacia atrás (del final al principio), una vez hacia delante y otra hacia detrás, etc. La animación habrá sido definida en una regla `@keyframes` y va a ser aplicada a un elemento seleccionado mediante

un selector CSS. El valor por defecto para esta propiedad es **normal** (implica que la animación se ejecutará hacia delante). Sus valores posibles son cualquiera de estas palabras clave:

- **normal**: la animación se ejecutará hacia delante. Si se repite, cuando vuelve a empezar parte de la situación inicial.
- **reverse**: la animación se ejecutará hacia detrás. Si se repite, cuando vuelve a empezar parte de la situación inicial.
- **alternate**: la animación se ejecutará una vez en un sentido y otra vez en otro, comenzando hacia delante, luego hacia detrás y así sucesivamente el número de repeticiones especificado.
- **alternate-reverse**: la animación se ejecutará una vez en un sentido y otra vez en otro, comenzando hacia detrás, luego hacia delante y así sucesivamente el número de repeticiones especificado.

Si se desean aplicar varias animaciones, se especificarán sus direcciones separadas por comas, siendo cada valor correspondiente a la animación cuyo nombre figura en el mismo orden en la propiedad animation-name.

Esta propiedad puede no ser reconocida por los navegadores antiguos o requerir del uso de prefijos específicos o no ser reconocida por algunos navegadores actuales. Consulta en Mozilla Developer Network para conocer si debes aplicar prefijos o si la propiedad será reconocida por un navegador concreto.

PROPIEDAD ANIMATION-DELAY

Análoga a la propiedad transition-delay. Indica el tiempo en segundos (s) o milisegundos (ms) que debe retrasarse el inicio de la animación respecto a lo que sería el momento normal para el comienzo. Ejemplo: animation-delay: 3s;

PROPIEDAD ANIMATION-TIMING-FUNCTION

La animación habrá sido definida en una regla @keyframes y va a ser aplicada a un elemento seleccionado mediante un selector CSS. El valor por defecto para esta propiedad es **ease** (implica que la animación se ejecutará con un comienzo rápido, luego velocidad constante y final lento.).

Si se desean aplicar varias animaciones, se especificarán sus timing-functions separadas por comas, siendo cada valor correspondiente a la animación cuyo nombre figura en el mismo orden en la propiedad animation-name.

PROPIEDAD CSS animation-timing-function	
Función de la propiedad	Permite definir la velocidad a la que transcurre la animación respetando la duración establecida. A efectos prácticos, permite que el usuario vea la animación como progresiva a velocidad constante o con cierta aceleración.
Valor por defecto	ease
Aplicable a	Todos los elementos.
Valores posibles para esta propiedad	ease : comienzo rápido, luego velocidad constante y final lento.
	ease-in : al revés que ease.
	ease-out : similar a ease con final a velocidad constante.
	ease-in-out : comienzo lento y progresión a velocidad constante.
	linear : progresión a velocidad constante.
	step-start : salto de la situación inicial a la final al comienzo.
	step-end : salto de la situación inicial a la final al final.
	steps (numeroDePasos, end) [El parámetro numeroDePasos establece el número de pasos intermedios entre la situación inicial y la final, lo que equivale a que la animación se vea como "pequeños saltos".]
	cubic-bezier (valor0a1, valor0a1, valor0a1, valor0a1) [Esta forma de definición permite especificar la curva temporal de forma matemática para definir formas de cambio personalizadas]
inherit (se heredan las características del elemento padre).	
Ejemplos aprenderaprogramar.com	<pre>.myContainer { animation-timing-function:ease;} .myContainerSP { animation-timing-function:steps(6, end);}</pre>

Esta propiedad puede no ser reconocida por los navegadores antiguos o requerir del uso de prefijos específicos para algunos navegadores actuales. Consulta en Mozilla Developer Network para conocer si debes aplicar prefijos.

PROPIEDAD ANIMATION-PLAY-STATE

Esta propiedad permite detener una animación que se está ejecutando (ponerla en pausa). Su valor por defecto es running y sus dos valores posibles son:

- **running**: la animación está en ejecución
- **paused**: la animación está en pausa

Esta propiedad puede no ser reconocida por los navegadores antiguos o requerir del uso de prefijos específicos para algunos navegadores actuales. Consulta en Mozilla Developer Network para conocer si debes aplicar prefijos.

Ejemplo: `.pubMov:hover{ animation-play-state:paused; }` con esta regla damos lugar a que la animación se detenga si el usuario pone el puntero del ratón sobre el elemento animado.

PROPIEDAD ANIMATION-FILL-MODE

Esta propiedad permite definir cómo debe comenzar y cómo debe quedar un elemento que tiene una animación. Su valor por defecto es `none` (implica que el elemento comenzará y quedará en el estado que tenía antes de que comenzara la animación) y sus valores posibles son:

- **none**: el elemento comenzará y quedará en el estado previo a la animación.
- **forwards**: el elemento quedará en el estado final de la animación.
- **backwards**: el elemento se pondrá en el estado indicado para el comienzo de la animación inmediatamente y esperará en ese estado hasta que se cumpla el tiempo indicado en `animation-delay`. Una vez se cumpla ese tiempo la animación continuará la ejecución desde ese estado inicial.
- **both**: combinación de las dos opciones anteriores.

Esta propiedad puede no ser reconocida por los navegadores antiguos o requerir del uso de prefijos específicos para algunos navegadores actuales. Consulta en Mozilla Developer Network para conocer si debes aplicar prefijos.

Ejemplo: `.pubMov:hover{ animation-fill-mode: forwards; }` con esta regla damos lugar a que cuando la animación se detenga quede en el último estado definido (el correspondiente a `to` ó `100%` dentro de la regla `@keyframes`).

PROPIEDAD SHORTAND ANIMATION

Esta propiedad permite definir las propiedades `animation-name`, `animation-duration`, `animation-timing-function`, `animation-delay`, `animation-iteration-count`, `animation-direction` y `animation-fill-mode` en una sola regla CSS.

Ejemplo: `.pubMov{ animation: animacion 5s 2 alternate linear forwards; }` equivale al conjunto de reglas:

```
animation-name:animacion; animation-duration: 5s; animation-iteration-count:2; animation-direction:alternate; animation-timing-function:linear; animation-fill-mode: forwards;
```

EJEMPLOS DE APLICACIÓN

Escribe este código y comprueba los resultados en tu navegador. Recuerda que hay navegadores que no soportan estas propiedades. Consulta en Mozilla Developer Network el soporte que ofrecen los navegadores y sus versiones.

```

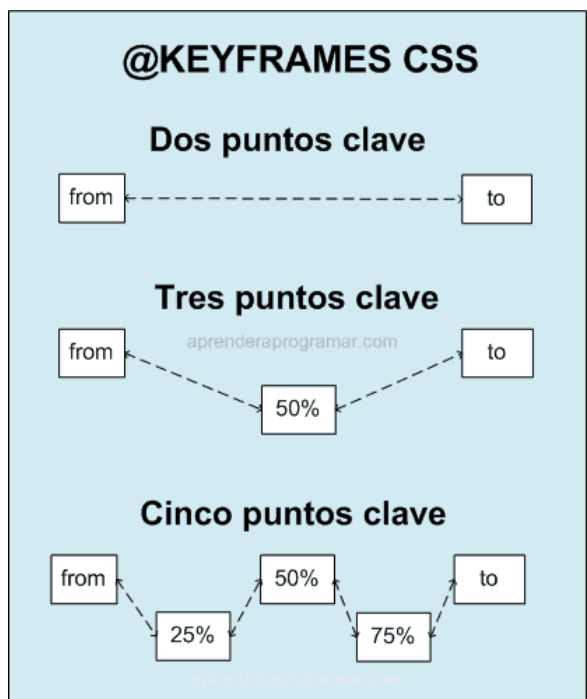
<html>
<head>
<title>Portal web - aprenderaprogramar.com</title> <meta charset="utf-8">
<style type="text/css">

    @keyframes animacion{
        from { margin-left: 10%; color: red;}
            25% { margin-left:22.5%; margin-top:10%; }
            50% { margin-left:35%; margin-top:0%;}
            75% { margin-left:47.5%; margin-top:10%; }
        to { margin-left: 60%; }
    }

    #bloquePub{ text-align:center; position:relative; width:250px;
        padding: 15px; font-family:arial; font-size:22px;
        font-weight:bold; background-color: yellow; }
    img {padding: 10px;}
    .pubMov{ animation-name:animacion; animation-duration: 5s;
        animation-iteration-count:2; animation-direction:alternate;
        animation-timing-function:linear; animation-fill-mode: forwards; }

    .pubMov:hover{ animation-play-state:paused; }
</style>
</head>
<body>
    <div class="pubMov" id="bloquePub" style="border-radius: 40px;">aprenderaprogramar.com
    </a>
    </div>
</body>
</html>
    
```

El resultado a obtener debe ser similar al que se indica como "Cinco puntos clave" en esta figura:



El elemento con un texto y una imagen debe partir de la zona izquierda de la pantalla, ir progresivamente hacia la derecha bajando primero hasta la posición 25%, luego subiendo hasta 50%, luego bajando de nuevo hasta 75% y luego subiendo hasta 100%. A continuación, debe continuar invirtiendo el movimiento, es decir, realizar lo que hizo anteriormente pero al revés hasta volver a la posición inicial.

Prueba a cambiar el número de puntos clave y otros parámetros y observa las diferencias. Si no logras visualizar la animación es probable que se deba a un problema de compatibilidad del navegador o necesidad de uso de prefijos. Consulta en los foros aprenderaprogramar.com si no logras resolverlo.

EJERCICIO

Busca en internet (página web o blog) una animación CSS y ejecútala en tu navegador. Escribe por separado el código HTML y el código CSS. Describe paso a paso qué es lo que indica cada fragmento de código CSS. ¿Es necesario usar prefijos de navegador para lograr su ejecución en otros navegadores?

Para comprobar si tu código y respuestas son correctas puedes consultar en los foros aprenderaprogramar.com.

Próxima entrega: CU01066D

Acceso al curso completo en aprenderaprogramar.com -- > Cursos, o en la dirección siguiente:
http://aprenderaprogramar.com/index.php?option=com_content&view=category&id=75&Itemid=203