



APRENDERAPROGRAMAR.COM

EFFECTO ANIMACIÓN CSS.
DIFERENCIA ENTRE
TRANSICIÓN Y
ANIMACIÓN.
@KEYFRAMES O
FOTOGRAFÍAS O ESTADOS
CLAVE. (CU01064D)

Sección: Cursos

Categoría: Tutorial básico del programador web: CSS desde cero

Fecha revisión: 2029

Resumen: Entrega nº64 del Tutorial básico: "CSS desde cero".

Autor: César Krall

ANIMATION CSS

Ya hemos visto que usando las propiedades transform y transition con hover podemos crear efectos dinámicos sobre una web. Sin embargo, la animación está limitada a que se produzca al posicionar el usuario el ratón sobre un elemento o usar otros recursos. CSS ha introducido nuevas técnicas que permiten elaborar animaciones más sofisticadas que pueden colaborar a no tener que recurrir a tecnologías como Javascript ó Flash.



CONCEPTO DE ANIMACIÓN CSS

El concepto de animación es un concepto que tiene cierta similitud con el concepto de transición CSS, pero que es más amplio y ofrece más posibilidades. Una animación es un efecto dinámico que cambia las propiedades de un elemento en el tiempo y que puede tener varios estados intermedios especificados. Una animación requiere al menos de dos estados: el estado inicial (from) y el estado final (to). No obstante, puede tener otros estados intermedios adicionales.

A continuación vemos las diferencias entre transición y animación:

	Transición	Animación
Uso básico	Cambio de estado inicial a final cuando usuario posiciona mouse encima de elemento	Cambio de estado inicial a otro final, pasando por varios estados intermedios diferentes si se desea y de ejecución espontánea si se desea
Ejecución espontánea fácil de codificar	No	Sí
Repetición espontánea durante cierto tiempo fácil de codificar	No	Sí
Repetición continua fácil de codificar	No	Sí
Ejecución hacia delante y hacia detrás fácil de codificar	No	Sí
Puede requerir uso de prefijos en ciertos navegadores	Sí	Sí

@KEYFRAMES

Para definir una animación empezaremos definiendo las propiedades que va a tener el elemento en cada uno de los estados de la animación mediante el uso de la regla CSS @keyframes.

Para ello se usa la siguiente sintaxis:

```
@keyframes nombreDeLaAnimacion{  
    from {propiedad1: valor1; propiedad2: valor2; ... propiedadN: valorN; }  
    valor%1 { propiedad1: valor; propiedad2: valor; ... propiedadN: valor; }  
    valor%2 { propiedad1: valor; propiedad: valor2; ... propiedad: valorN; }  
    ...  
    valor%N { propiedad1: valor; propiedad: valor2; ... propiedad: valorN; }  
    to {propiedad1: valor1; propiedad2: valor2; ... propiedadN: valorN; }  
}
```

Las propiedades definidas en from { ... } definen el estado inicial o situación de partida. Las propiedades definidas en to { ... } definen el estado final o situación de terminación. Estos dos estados son de declaración obligatoria.

Entre los estados from y to pueden existir un número indeterminado de estados intermedios que representan la situación en momentos intermedios durante la ejecución de la animación. La animación va pasando así por los estados definidos hasta llegar al estado final.

El ejemplo más básico puede ser tener sólo un estado inicial y un estado final. Sería este caso:

```
@keyframes animacion{  
    from { margin-left: 10%; }  
    to { margin-left: 60%; color:red }  
}
```

En este ejemplo estamos definiendo que el elemento HTML al que se le aplique la animación que tiene por nombre "animacion" pasará de tener un margin-left del 10%, es decir, estar al lado izquierdo, a tener un margin-left del 60%, es decir, estar al lado derecho de su contenedor. Además en el lado izquierdo tendrá su color por defecto mientras que en lado derecho tomará color rojo. Aquí hemos definido básicamente un movimiento de izquierda a derecha.

Ahora otra definición con un punto clave intermedio:

```
@keyframes animacion{  
    from { margin-left: 10%; }  
        50% { margin-left:35%; margin-top:10%; }  
    to { margin-left: 60%; }  
}
```

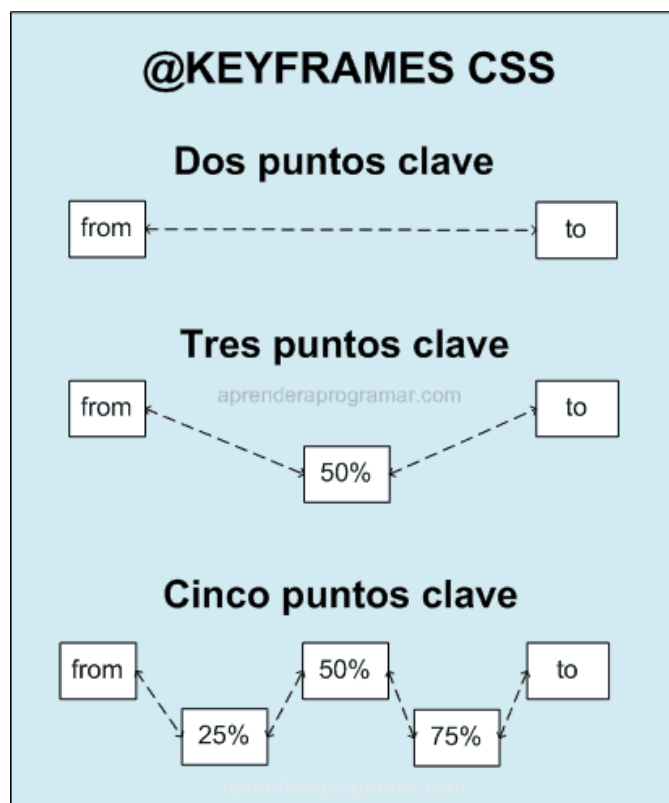
Aquí hemos indicado que cuando la transición esté al 50% de su ejecución el elemento debe estar a medio camino entre la izquierda y la derecha (entre el 10% y el 60% el punto medio resulta el 35%) y además que en ese momento debe tener un desplazamiento hacia abajo debido al margin-top. Hemos definido un movimiento de izquierda a derecha que va bajando hasta alcanzar su posición más baja en el centro y luego sube hasta llegar a su posición en el lado derecho.

Ahora otra definición con más puntos clave:

```
@keyframes animacion{
  from { margin-left: 10%; }
    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%; }
```

Con esta definición el elemento partirá de arriba a la izquierda, se irá desplazando hacia la derecha y bajará, luego subirá, luego volverá a bajar y terminará arriba a la derecha. Hemos definido un movimiento de “sube y baja” de arriba a derecha.

Para entender gráficamente lo que suponen estas definiciones podemos ver este esquema:



Nosotros hemos usado como ejemplo que un elemento se va moviendo, pero ten en cuenta que animación no tiene por qué implicar movimiento. Animación implica cambio de propiedades, con lo

cual un elemento HTML puede estar quieto y presentar una animación debido a que va cambiando su color, su tamaño de fuente, su fondo, etc. y no es obligatorio que exista movimiento. De hecho, un efecto bastante habitual es hacer brillar un texto, hacer que su color o efecto de gradiente de fondo vaya cambiando, etc.

La regla @keyframes 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.

Una vez hemos definido los keyframes pasaremos a la definición de los parámetros que definen la animación mediante las propiedades animation-name, animation-duration, animation-delay, animation-fill-mode, animation-iteration-count, animation-direction y animation-timing-function.

EJERCICIO

Estudia el siguiente código CSS y responde a las cuestiones planteadas:

```
@keyframes anime{
  0%{ color: #f00; font-size: 10px; top: 10px;}
  25%{color: #A52A2A; font-size: 90px; left: 100px;}
  50%{ top: 56px; color: #000; font-size: 90px; }
  100%{color: #A52A2A; font-size: 90px; left: 0; }
}
```

- a) ¿Cuál es el nombre de la animación? ¿Cuántos puntos clave define?
- b) ¿Qué ocurrirá (cuál será el cambio de propiedades) durante la animación?
- c) ¿Es equivalente usar 0% en lugar de from? ¿Y 100% en lugar de to?

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

Próxima entrega: CU01065D

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