

UD02 - Animaciones CSS

Angel Berlanas

November 6, 2019

Contents

1 Animaciones CSS	1
1.1 ¿Qué son las Animaciones CSS?	1
1.2 Propiedades	1
1.2.1 animation-name	2
1.2.2 animation-duration	2
1.2.3 animation-iteration-count	2
1.2.4 animation-timing-function	2
1.2.5 @keyframes	3
1.2.6 Un ejemplo completo	3
1.2.7 Ejercicio 13 : ¡Estan vivos!¡Vivooooos!	4

1 Animaciones CSS

1.1 ¿Qué son las Animaciones CSS?

Las animaciones CSS3 permiten animar la transición entre un estilo CSS y otro, son muy similares a las **transiciones** que vimos en clases anteriores.

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.

1.2 Propiedades

A continuación vamos a ver las diferentes propiedades que pueden tener las **Animaciones CSS**, como vereís se trata de conceptos muy similares a los que están descritos en los apartados anteriores.

1.2.1 animation-name

Se trata de la propiedad que establece el nombre de la regla (**@keyframes**) que describe los fotogramas de la aplicación.

Pueden especificarse más de uno, en ese caso irán separados por `~,~`.

1.2.2 animation-duration

Especificado en *segundos* lo que dura la animación **completa**.

1.2.3 animation-iteration-count

Indica cuantas veces debe realizarse la animación.

Los valores posibles son:

- `infinite`
- `<number>`

Donde `<number>` establece un número de iteraciones, se pueden poner decimales, y se aplicará ese porcentaje de la animación.

```
animation-iteration-count: 2.3;
```

Realizará la animación 2 veces completas y **un tercio** de la misma.

1.2.4 animation-timing-function

Se trata del mismo concepto que vimos en las transiciones, *cómo cambia en el tiempo* la propia animación, acelerándose o frenándose según se especifique.

Este parámetro puede ser establecido en cada uno de los **@keyframes** que tenga la animación.

Y aquí también podemos usar **cubic-bezier** como posibles valores:

Valores
ease-in
ease-out
ease-in-out
linear
cubic-bezier

1.2.5 @keyframes

@keyframes es la parte en la que programamos la animación, indicando mediante una serie de palabras *clave* y los porcentajes de la animación.

```
@keyframes pixar {  
  
    from {  
top:100px;  
animation-timing-function:ease-in;  
    }  
    50%{  
        top:125px;  
        animation-timing-function:ease-out;  
    }  
    to {  
top:150px;  
    }  
}
```

1.2.6 Un ejemplo completo

```

.elemento {
    width: 142px;
    height: 142px;
    background-color: #00AA66;
    position: relative;
    margin: 0 auto;
    animation-name: saltarin;
    animation-duration: 2s;
    animation-iteration-count: infinite;
}

@keyframes saltarin {
    from {
    top: 150px;
    animation-timing-function: ease-out;
    }
    25% {
    top: 50px;
    animation-timing-function: ease-in;
    }
    50% {
    top: 200px;
    animation-timing-function: ease-out;
    }
    75% {
    top: 75px;
    animation-timing-function: ease-in;
    }
    to {
    top: 150px;
    }
}

```

1.2.7 Ejercicio 13 : ¡Estan vivos!;Vivooooos!

La investigación de los pergaminos ha concluido con éxito. Los *Seres de Sedefkar* están despertando y ya son capaces de realizar pequeñas tareas.

Utilizando dos *papiros* que se ha encontrado en la Biblioteca de la Universidad de Miskatonic, que están escritos en JS y CSS. Los investigadores deben recomponer los fragmentos perdidos que permitirán indicar a los difer-

entes *Seres de Sedefkar* cuál será su tarea, ya que si no los controlaríamos los *Seres de Sedefkar* se unirán bajo la voluntad de Cthulhu y acabarán con la cordura de los investigadores.

Sobre los pergaminos anteriores añadiremos dos áreas, que al hacer click sobre ellas, nos permitirán indicarle la acción sobre los *Seres de Sedefkar*.

- Se establezcan dos *glifos de órdenes* en la parte superior del tablero, al lado del botón que hacer surgir las cajas. El primero hará girar al *Ser de Sedefkar* y el segundo hará un recorrido en V y volverá al origen.
- Si el *Ser de Sedefkar* está visible ha de hacer caso, en caso contrario, la orden se pierde.
- Si el *Ser de Sedefkar* ya está ejecutando una orden, debe dejar de ejecutarla y pasar a realizar la nueva orden.
- Si se hace click fuera de un *Ser de Sedefkar* la orden también se pierde.
- **(Opcional)** Establecer una tercera orden que hace el *Ser de Sedefkar* deje de hacer *cosas*.