UTN.BA
UNIVERSIDAD TECNOLÓGICA NACIONAL
FACULTAD REGIONAL BUENOS AIRES

PROGRAMADOR WEB INICIAL: FRONT END DEVELOPER

CSS

Universidad Tecnológica Nacional - Facultad Regional Buenos Aires
Medrano 951 (C1179AAQ) C.A.B.A. Tel: (54 11) 4867-7511

Temario:

- Animando con CSS3
- Propiedad animation
- Propiedad transform.
- Transformando elementos en 3D
- Propiedad transition.

1. Animando con CSS3

Una de las características más interesantes que nos brinda CSS3 es la posibilidad de crear nuestras propias animaciones a los elementos del sitio sin la necesidad de programas ni plugins.

Anteriormente, para poder animar necesitábamos instalar programas o hacer uso de otros lenguajes como JavaScript para lograr algunas animaciones. En la actualidad, las propiedades de animación no solamente nos permiten darle movimiento a los elementos de una manera más simple, sino que también nos dan la seguridad de la compatibilidad con gran parte de los distintos tipos de navegador utilizando un lenguaje amigable y conocido.

Tenemos que tener en cuenta que las siguientes propiedades nos permiten no solo ser utilizadas de forma individual sino también podemos combinarlas dentro de un mismo elemento para lograr el efecto que deseemos.

1.2 Propiedad animation

Esta propiedad nos permite crear la animación para elementos de HTML estableciendo distintos estadios de esta durante el tiempo que deseemos que dure.

Para poder definirla, necesitamos el uso de la palabra clave ***@keyframe.*** Esta palabra nos definirá tanto el nombre de la animación (lo que nos va a permitir poder reutilizarla en los elementos que deseemos) como la lista de estilos que va a tener a lo largo del tiempo el elemento y en qué tiempo de la animación va a tenerlos.

Pero entonces, ¿cómo uso esta propiedad y @keyframe?

Para ello, necesitamos primero definir con ***@keyframe*** el nombre de la animación y qué propiedades va a recibir el elemento a lo largo de lo que dure la animación. Podemos hacerlo utilizando **porcentajes** donde indicamos el estadio inicial, estadios intermedios y el final; o bien indicando con las palabras claves **from** y **to** para indicar el estadio inicial y final.

Sintaxis

```
@keyframe nombre_animacion{
    0%{
        propiedad_del_elemento_en_dicha_instancia
    }
    25%{
        propiedad_del_elemento_en_dicha_instancia
    }
    50%{
        propiedad_del_elemento_en_dicha_instancia
    }
```

```
100%{

propiedad_del_elemento_en_dicha_instancia

                }

            }
```

Una vez definido el @keyframe, ya podemos utilizar la propiedad animation con sus variantes para poder invocar la misma en los elementos.

animation-name:

Esta propiedad de ***animation*** nos sirve para invocar el nombre de la animación que definimos en la @keyframe en el elemento a utilizar.

animation-duration:

Esta propiedad de ***animation*** nos sirve para indicar el tiempo que queremos que dure la animación. Esta propiedad utiliza unidades en segundos (s).Tener en cuenta que si no definimos la duración de la animación, esta no va a ocurrir ya que por defecto la duración de la animación es 0 segundos (0s).

animation-delay:

Esta propiedad de ***animation*** nos sirve para indicar el tiempo de demora que debe tener antes de comenzar la animación. Esta propiedad utiliza la unidad de segundos y tiene como valor por defecto de 0 segundos (0s).

animation-fill-mode:

Esta propiedad de ***animation*** nos permite indicarle si, al finalizar la animación, el elemento debe volver a su estado original o quedarse en el estado final definido de la animación.

Esta propiedad puede tomar como valores: **`none`** (valor por defecto donde no aplica estilos ni antes ni después de su ejecución)**,** **`forwards`** (indica que, al finalizar la animación, el elemento quedará con las propiedades definidas en el estado final de la animación)**,** **`backwards`** (indica que, al finalizar la animación, el elemento quedará con las propiedades definidas en el estado inicial de la animación)**,** **`both`** (indica que, antes de iniciar la animación, tendrá las propiedades del estado inicial definido en el @keyframe y que, al finalizar la animación, quedará con las propiedades definidas en el estado final de dicha @keyframe).

animation-direction:

Esta propiedad de ***animation*** nos permite indicarle qué debe hacer al finalizar la animación. Puede tomar como valores: **`normal`** (valor por defecto que indica que al finalizar la animación vuelva al estado original)**,** **`alternate`** (indica que al finalizar la animación invierta su dirección) **,`reverse`** (indica que cada animación se reproduzca al revés, es decir de final a inicio)**,** **`alternative-reverse`** (es una combinación de los valores *altérnate* y *reverse* donde la animación inicia desde su estado final hacia el estado inicial y al finalizar invierte su dirección)**.**

animation-interation-count:

Esta propiedad de ***animation*** nos indica la cantidad de veces que queremos que la animación se repita. Puede tomar valores numéricos y también la palabra ***infinite*** si queremos que se repita constantemente.

animation:

Esta propiedad funciona como una abreviación de todas las propiedades anteriormente definidas

Con esta propiedad podemos en una misma línea definir todas las propiedades de animation siempre separando con un espacio entre cada uno de los valores.

1.2 Propiedad transform

Esta propiedad de CSS nos permite modificar o transformar un elemento en términos bidimensionales y tridimensionales. Nos permite poder trasladar, rotar, escalar o incluso inclinar elementos.

Métodos de transform

translate(tX, tY):

Nos permite trasladar el elemento en 2D y puede tomar valores de longitud y de porcentaje. Al definir este método, debemos darle la distancia en la que se va a trasladar en el eje x **(tX)** y la distancia en la que se trasladara en el eje y **(tY).**

rotate(deg):

Este método nos permite girar el elemento. Podemos especificar los grados en que
queremos que dicho elemento gire. Utilizamos como unidad los grados (**deg**) y girará en sentido horario.

scale(sX, sY) / scale(sXY):

Método que nos permite escalar en dos dimensiones al elemento. Este método toma valores numéricos sin unidad y podemos definir tanto lo que queremos que escale en el eje x **(sX)** como en el eje

y **(sY)**. En caso que solamente definimos un único eje, se asume que dicho valor corresponde tanto al eje x como al y **(sXY).**

scaleX(sX):

Este método se utiliza cuando solamente queremos que escale el eje X y puede tomar valores numéricos sin unidad.

scaleY(sY):

Este método se utiliza cuando solamente queremos que escale el eje Y y puede tomar valores numéricos sin unidad.

skew(sX,sY):

Este método nos permite inclinar al elemento en 2D indicándole los grados de inclinación que deseamos en el eje x **(sX)** e y **(sY)**. En caso de que solamente definamos uno de ellos, inclinará el elemento en el eje x únicamente. Como se puede deducir de la definición de este método, utiliza como unidad grados (deg).

1.3 Transformando elementos en 3D

Cuando hablamos de elementos tridimensionales, nos referimos a un tercer eje que se suma al plano: el eje z.
Este eje es el que va a dar el efecto de si un elemento se encuentra más cerca o más lejos del espectador. Para ello, lo que tenemos en cuenta es una perspectiva, la cual definirá la parte visual donde un elemento que queramos que dé el efecto de encontrarse más cerca será de un tamaño más grande que el elemento que se encuentre más lejos.

Para poder definir esta perspectiva, en CSS contamos con la propiedad ***perspective.***

Esta propiedad la definimos en el elemento padre del elemento que queremos que reciba el efecto y puede tomar valores de longitud ya que indica la distancia que tiene el elemento al eje “z'. Usualmente se utiliza la unidad pixeles.

1.4 Propiedad transition

Esta propiedad nos sirve para indicar cómo queremos que se produzca el cambio entre dos estados de un elemento.

Para definirlo, debemos indicar la propiedad que produce el cambio y la duración.Este último utiliza como unidad los segundos **(s)** y tenemos que tener en cuenta que si no definimos la duración este tomará su valor por defecto que es 0 segundos **(0s)**, lo que provoca que no se produzca el efecto.

Sintaxis

```
transition: nombre_de_la_propiedad  duración;
```

2. Bibliografia

https://developer.mozilla.org/es/docs/Web/CSS/@keyframes
https://developer.mozilla.org/es/docs/Web/CSS/transform
https://css-tricks.com/almanac/properties/t/transform/
https://www.w3schools.com/cssref/css3_pr_animation.asp
https://www.w3schools.com/css/css3_animations.asp
https://developer.mozilla.org/es/docs/Web/CSS/animation-delay
https://css-tricks.com/almanac/properties/a/animation/
https://www.w3schools.com/css/css3_2dtransforms.asp
https://developer.mozilla.org/es/docs/Web/CSS/transition
https://www.w3schools.com/css/css3_transitions.a