

Animaciones en CSS

¿Qué son las animaciones CSS?

Una animación permite que un elemento cambie gradualmente de un estilo a otro. Adicional permite cambiar las propiedades de CSS tantas veces como desee

Básicamente una animación CSS se define como el la transición gradual de unos estilos a otros en los elementos del html. Es decir, en el CSS vas a poder definir una animación mediante los estilos de partida y los estilos que quieras al final de la animación, y el navegador se va a encargar de hacer el paso de unos estilos a otros de forma gradual.

Para empezar a crear las animaciones, simplemente crea en el CSS una regla especial llamada keyframes con la regla `@keyframes` y dentro definiremos las distintas posiciones de nuestra animación. La primera será al 0% y la última al 100% (en lugar de 0% y 100% podremos usar `from` y `to`):

Ejemplo 1

```
@keyframes intro
{
  from { opacity: 0;
        transform: scale(10) rotate(-150deg);
        }
  to { opacity: 1;
       transform: scale(1);
       }
}

.elemento {
  width: 100px;
  height: 100px;
  background-color: red;
  animation-name: Clase;
  animation-duration: 4s;
}
```

► [El resultado en Codepen](#)

Ejemplo 2

```
@keyframes Clase {
  0%{
    background-color: palegreen;
  }
  25%{
    background-color:palevioletred;
  }
}
```

```

    }
    50%{
        background-color:#000;
    }
    75%{
        background-color:rgb(136, 19, 19);
    }
    100%{
        background-color: palegreen;
    }
}

```

En este ejemplo hemos creado una animación llamada ejemplo con clase. En este ejercicio el elemento cambiara de color sugun el grado de ejecución que se encuentre la animación

Para hacer uso de la animacion debemos invocar la animacion al interior de los estilos de un elemento

```

.elemento {
    width: 100px;
    height: 100px;
    background-color: red;
    animation-name: Clase;
    animation-duration: 4s;
}

```

subpropiedades de las animaciones:

animation-delay: determina el tiempo hasta que empieza la animación

animation-direction: especifica si la animación mantendrá la dirección, la alternará en cada repetición, o si la resetea al punto inicial.

animation-duration: marca el tiempo que durará la animación.

animation-iteration-count: dice las veces que se va a repetir la animación.

animation-name: especifica el identificador de la animación.

animation-play-state: permite pausar una animación.

animation-timing-function: funciona igual que con las transiciones.

animation-fill-mode: especifica si los valores de la animación se aplicarán al elemento animado una vez termine o antes de empezar

Vamos a probar una más... [powerpoint](#)

Documentación completa de las sub propiedades de animation en la MDN: [CSS Animations](#)

Ejemplos de animaciones (más o menos curradas)

[Pure CSS Animated Checkbox](#)

[CSS Animate \(y ha escrito un post sobre cómo se hace\)](#)

[CSS Animation Material Design](#)
[CSS Animation with steps\(\)](#)
[Mr JeellyFish pure CSS animation](#)
[WWDC 2015 CSS](#)
[Cloudy Spiral CSS animation](#)

[Transforms en CSS](#)

Las transformaciones CSS nos permiten rotar, torcer, escalar o desplazar los elementos de nuestra página web. Este tipo de propiedades de CSS3 son muy interesantes para convertir el lenguaje de hojas de estilo en un sistema capaz de realizar todo tipo de efectos visuales.

Tipos de transformaciones

Las transformaciones que podemos aplicar son las siguientes:

scale: modifica el tamaño de los elementos. Translate: cambia la posición del elemento. Izquierda, derecha, arriba o abajo. Rotate: gira o rota los elementos en grados. skew: distorsiona los elementos según el ángulo en grados. Matrix: mueve o transforma los elementos con precisión de píxel.

► [Ver en Codepen](#)

[¿Cómo se construye una transición?](#)

Las transiciones se especifican con la propiedad transition que es un atajo (shortcut) de: transition-property la propiedad sobre la que queremos aplicar la transición

transition-duration la duración de la transición en segundos o milisegundos

transition-timing-function: una función de tiempo para determinar cómo aplicar la transición, hay unas palabras clave para algunas funciones.

transition-delay: un tiempo de retardo antes de empezar la transición No hace falta especificarlas todas y podemos usar el atajo o no. En los ejemplos lo usamos con la propiedad, la función de tiempo y la duración de la transición:

```
.caja1{
  background-color: #C0392B;
  transition: 1s linear;
}
.caja1:hover{
  background-color: #C0334B;
  transform: scale(0.5);
}
```

