

Unidad 8: Transiciones y animaciones

Transformaciones

transform: scale()

La función `scale` de la propiedad `transform` permite aumentar o reducir el tamaño del elemento al que se aplica. Dentro de los paréntesis se indica un valor, en base a 1, según el cual se escalará el elemento: si el valor es menor que 1 reducirá su tamaño y, si es mayor, lo aumentará.

transform: scaleX()

La función `scaleX` permite escalar un elemento solo en el eje X, es decir, horizontalmente.

transform: scaleY()

La función `scaleY` permite escalar un elemento solo en el eje Y, es decir, verticalmente.

transform: rotate()

La función `rotate` permite rotar un elemento, indicando los grados de rotación (deg).

transform: translate()

La función `translate` permite desplazar un elemento en el eje X y el eje Y, dando dos valores separados por un espacio.

transform: translateX()

La función `translateX` permite desplazar un elemento en el eje X.

transform: translateY()

La función `translateY` permite desplazar un elemento en el eje Y.

transform: skewX()

La función `skewX` permite deformar un elemento un número de grados (señalados en deg) en el eje X.

transform: skewY()

La función `skewY` permite deformar un elemento un número de grados (señalados en deg) en el eje Y.

transform-origin: coordenadas

La función `transform-origin` permite cambiar el origen (en el eje X y el eje Y) a partir del cual se produce una transformación (por defecto, el centro).

Transiciones

transition: propiedad duración velocidad;

La propiedad `transition` permite indicar una transición en un cambio de estado (normalmente, un `:hover`), indicando la propiedad (por defecto, todas), la duración de la transición (expresado en segundos, `s`, o milisegundos, `ms`) y la velocidad (o aceleración).

transition: ... retraso;

La propiedad `transition` puede contener un cuarto valor que indicaría el retraso en iniciarse, expresado en `s` o `ms`.

Tipos de velocidad/aceleración

La velocidad (o aceleración) por defecto en las transiciones es `ease-in-out`, más lento al principio y al final. Otros valores posibles son `ease-in` (más lento al principio), `ease-out` (más lento al final), `linear` (velocidad constante) o una curva de aceleración, que se expresa con la función `cubic-bezier()` y cuatro valores en base 1, separados por comas. Puedes ver más sobre esta función en <http://cubic-bezier.com/>

Animaciones

@keyframes miAnimacion{}

Una animación se describe primero (antes de ser asignada a un elemento) con la regla `@keyframes` y un nombre para la animación. Dentro de las llaves, indicaremos los pasos que seguirá la animación, indicados por las palabras `from` y `to` (si solo tiene dos pasos) o con porcentajes que indican el progreso de la animación.

Ejemplo:

```
@keyframes miAnimacion{
  0%{
    width: 10px;
  }
  60%{
    width: 10px;
  }
```

```
100%{  
  width: 10px;  
}  
}
```

animation: miAnimacion duraci3n repeticiones retraso velocidad alternate;

Esta propiedad nos permite asignar a un elemento la animaci3n definida con @keyframes. Los valores que acepta (separados por espacios) son el nombre de la animaci3n, la duraci3n de la animaci3n (en s o ms), las repeticiones de la animaci3n (infinite para que se repita continuamente), el retraso en el inicio de la animaci3n (si se desea), y la velocidad o aceleraci3n. Si se desea que una vez finalizada la animaci3n, se repita en sentido contrario, se puede a1adir el valor alternate.