

ANIMACIONES, TRANSFORMACIONES Y TRANSICIONES EN CSS

Con las propiedades para animaciones, transformaciones y transiciones es posible crear muchos interesantes efectos sin necesidad de JavaScript, uso de frameworks o de la librería jQuery. La combinación de HTML con CSS nos permitirá crear sitios optimizados para SEO y con soporte nativo en la mayoría de navegadores web.

Transformaciones en CSS

El efecto de una transformación en CSS consiste en aplicar un cambio de apariencia geométrica a cualquier elemento. Esto proporciona una notable capacidad de control visual que anteriormente no era posible en lo que respecta a HTML o a las hojas de estilo.

La propiedad CSS para realizar transformaciones se denomina *transform* (esta propiedad requiere el uso de prefijos propietarios para garantizar la compatibilidad entre navegadores). La propiedad transform permite hacer los siguientes tipo de transformaciones:

- Rotación
- Transformación de escala
- Distorsión
- Traslación

Función	Descripción
rotate(angulo)	Especifica una rotación 2D en el ángulo especificado en el parámetro sobre el origen del elemento.
scale(sx,sy)	Especifica una operación de escala 2D con el vector de escala [sx,sy] descrita por los dos parámetros.
scaleX(sx)	Especifica una operación de escala con el vector de escala [sx,1], donde sx se indica como parámetro.
scaleY(sy)	Especifica una operación de escala con el vector de escala [1,sy], donde sy se indica como parámetro.
skew(angX,angY)	Especifica una transformación de deformación a lo largo de los ejes X e Y. El primer parámetro de ángulo especifica la deformación en el eje X. El segundo parámetro de ángulo especifica la deformación en el eje Y.
skewX(angulo)	Especifica una operación de deformación a lo largo del eje X según el ángulo indicado.
skewY(angulo)	Especifica una operación de deformación a lo largo del eje Y según el ángulo indicado.
translate(tx,ty)	Especifica una traslación 2D según el vector [tx,ty] descrito por los dos parámetros
translateX(tx)	Especifica una traslación de cantidad determinada en la dirección X.
translateY(ty)	Especifica una traslación de cantidad determinada en la dirección Y.

Nota: Todas estas propiedades requieren el uso de prefijos propietarios para garantizar la compatibilidad entre navegadores.

Múltiples transformaciones

A veces podría resultar útil realizar sobre un elemento varias transformaciones al mismo tiempo. Para obtener una propiedad **transform** combinada, solo tenemos que separar cada función a aplicar con un espacio.

Considere el siguiente ejemplo en el que se escala, rota y desplaza en el eje X un 3º elemento div:

```
div:nth-child(3) {  
    transform: scale(1.25) rotate(30deg) translate(40px,0px);  
}
```

Transiciones en CSS

Mientras que las transformaciones son una herramienta fuerte e interesante para el desarrollador, la posibilidad de animar los mismos efectos es algo aún más emocionante, es en este momento donde la propiedad **transition** entra en acción.

Las transiciones en CSS permiten modificar el valor de una propiedad de un elemento HTML en forma gradual durante un tiempo determinado de un estado inicial a un estado final.

La sintaxis más simple para definir una transición de una propiedad es:

```
Elemento {  
    transition: [nombre_propiedad] [duración_transición] ;  
}
```

En el siguiente ejemplo se transiciona la rotación de un div para que esta rotación se realice de forma sutil y gradualmente:

```
div {  
    display: inline-block;  
    padding: 1em;  
    margin: 1em;  
    background-color: #666;  
    color: #fff;  
    transition: transform 2s;  
}  
  
div:nth-child(1):hover {  
    background-color: #3F51B5;  
    transform: rotate(30.5deg);  
}
```

Transiciones de múltiples propiedades

También podemos hacer transiciones de múltiples propiedades, para ello indicamos cada transición separada por coma:

```
Elemento {  
    transition: [nombre_propiedad] [duración_transición],  
               [nombre_propiedad] [duración_transición],  
               [nombre_propiedad] [duración_transición] ;  
}
```

También es posible transicionar todas las propiedades de un elemento mediante la palabra clave **"all"**.

```
div {  
    display: inline-block;  
    padding: 1em;  
    margin: 1em;  
    background-color: #666;  
    color: #fff;  
    transition: all 3s;  
}
```

Funciones de transición

Un tercer parámetro opcional de la propiedad transition es indicar una "función de transición" que nos permite seleccionar la velocidad durante la transición:

```
Elemento {  
    transition: [nombre_propiedad] [duración_transición] [función_de_transición];  
}
```

Los valores posibles que podemos especificar son:

- **ease** : Define un efecto de transición con un comienzo lento, luego rápido y finalmente termina lento (cuando no definimos la función de transición elige esta por defecto)
- **linear** : Define un efecto de transición con la misma velocidad de inicio a fin.
- **ease-in** : Define un efecto de transición con un comienzo lento.
- **ease-out** : Define un efecto de transición con un final lento.
- **ease-in-out** : Define un efecto de transición con un comienzo lento y un final lento.



Tiempo de demora en iniciar la transición

El cuarto parámetro opcional de la propiedad transition es indicar un tiempo de espera hasta que se inicie la transición:

```
Elemento {  
    transition: [nombre_propiedad] [duración_transición]  
               [función de transición] [tiempo de inicio];  
}  
Es decir indicamos la cantidad de milisegundos o segundos hasta que se inicia el proceso de transición.
```

Sintaxis alternativa

Hemos visto que a la propiedad transition podemos indicarle hasta cuatro parámetros. Los dos primeros son obligatorios y los otros son opcionales. CSS dispone de otras cuatro funciones para indicar cada uno de estos valores en forma independiente:

```
Elemento {  
    transition-property: [nombre de propiedad];  
    transition-duration: [duración de la transición];  
    transition-timing-function: [función de transición];  
    transition-delay: [tiempo de inicio];  
}
```

Animaciones en CSS

Las animaciones en CSS nos permiten hacer cosas que con las transiciones no alcanzamos y nos quedamos cortos. La sintaxis básica para una animación es la siguiente:

```
Elemento {  
    animation-name: [nombre de la animación];  
    animation-duration: [tiempo de duración];  
}  
@ keyframes [nombre de la animación] {  
    from {  
        [propiedades y valores del estado inicial de la animación]  
    }  
    to {  
        [propiedades y valores del estado final de la animación]  
    }  
}
```

animation-iteration-count

La propiedad **animation-iteration-count** permite controlar el número de repeticiones que tendrá la animación.

```
Elemento {  
    animation-iteration-count: [cantidad de veces a repetir la animación o  
    "infinite"];  
}
```

Si queremos que la animación se repita solo tres veces luego indicamos en la propiedad animation-iteration-count dicho valor:

```
animation-iteration-count: 3;
```

En cambio si queremos que se repita siempre luego especificamos el valor "infinite":

```
animation-iteration-count: infinite;
```

animation-direction

La propiedad animation-direction indica como debe repetirse la animación. Esta propiedad puede asumir dos valores:

- **normal:** con lo que la animación cada vez que finaliza comienza desde el principio.
- **Alternate:** cuando finaliza la animación comienza desde el final hasta el principio.

```
Elemento {  
  animation-direction: ["normal" o "alternate"];  
}
```

animation-timing-function y animation-delay

Similar a las transiciones disponemos de dos propiedades para definir la función de transición y el tiempo que debe esperar para comenzar la animación:

```
Elemento {  
  animation-timing-function: [función de transición];  
  animation-delay: [tiempo de demora para iniciar la animación];  
}
```

Los valores posibles para la propiedad animation-timing-function son exactamente las mismas funciones disponibles para las transiciones, es decir: ease, linear, ease-in, ease-out y ease-in-out.

Definición de más de 2 keyframes

Hasta ahora hemos indicado en la animación solo 2 keyframes (el inicial y el final), pero para animaciones más complejas es posible que necesitemos más de 2 keyframes, para esto tenemos la siguiente sintaxis:

```
@keyframes [nombre de la animación] {  
  0%{  
    [propiedades y valores]  
  }  
  [valor en porcentaje]{  
    [propiedades y valores]  
  }  
  [valor en porcentaje]{  
    [propiedades y valores]  
  }  
  100%{  
    [propiedades y valores]  
  }  
}
```