CSS – Transformaciones

¿Qué es una transformación?: consiste en modificar los atributos de un objeto (HTML) para cambiar sus propiedades de lugar, ángulo, escala y deformación.

1) Desplazamiento:

Es el desplazamiento de un objeto, respecto a la posición de su eje. La función por defecto nos permite especificar en qué dirección se tiene que mover. Pero lo podemos especificar para un solo eje.

```
.translacion {

     transform:translate (100px, 10px;);
     -o-transform:translate (100px, 10px;); /* Para Opera */
     -ms-transform:translate (100px, 10px;); /* Para IE */
     -moz-transform:translate (100px, 10px;); /* Para Firefox */
     -webkit-transform:translate (100px, 10px;); /* Para Safari y Chrome
*/
}
```

Si solo quisiéramos usar la translación en un eje, podríamos hacerlo usando la instrucción transform:translateX, en el caso del eje X. Igual para los demás.

Ejemplo:

```
<!DOCTYPE html>
<html lang="es">
     <head>
          <title>Transformación de desplazamiento</title>
          <meta charset="utf-8">

          <style type="text/css">

               div{
                    background-color: red;
                    width: 150px;
                    height: 150px;
                    transition: all 2s ease;
               }

               div:hover{

                    transform: translate(250px, 250px);
               }
```

```
          </style>

     </head>
     <body>
          <div></div>
     </body>
</html>
```

2) Rotación:

Permite girar un objeto sobre su eje. Se define en grados, y al igual que las translaciones, permiten especificar un eje en concreto.

```
.rotacion {
     transform:rotate(30deg);
     -o-transform:rotate(30deg); /* Para Opera */
     -ms-transform:rotate(30deg); /* Para IE */
     -moz-transform:rotate(30deg); /* Para Firefox */
     -webkit-transform:rotate(30deg); /* Para Safari y Chrome */
}
```

Ejemplo:

```
<!DOCTYPE html>
<html lang="es">
     <head>
          <title>Transformación rotación</title>
          <meta charset="utf-8">

          <style type="text/css">

               div{
                    background-color: red;
                    width: 150px;
                    height: 150px;
                    transition: all 2s linear;

               }

               div:hover{

                    transform: rotate(360deg);
               }

          </style>

     </head>
```

```
     <body>

          <div></div>

     </body>
</html>
```

3) Escalado:

Cambia el tamaño de un objeto, respecto al original. Tiene dos parámetros, el eje X y el eje Y. Si ponemos el mismo valor en ambos casos, haremos un escalado. Pero si ponemos diferentes valores, haremos una deformación.

```
.escalado {
     transform:scale(1.5,1.5);
     -o-transform:scale(1.5,1.5); /* Para Opera */
     -ms-transform:scale(1.5,1.5); /* Para IE */
     -moz-transform:scale(1.5,1.5); /* Para Firefox */
     -webkit-transform:scale(1.5,1.5); /* Para Safari y Chrome */
}
```

Vemos que al poner un factor de 1.5, el objeto es un 50% mayor que el original.

IMPORTANTE!: Recordad que la escala se realiza desde el eje central del objeto.

Ejemplo:

```
<!DOCTYPE html>
<html lang="es">
     <head>
          <title>Tranformación escala</title>
          <meta charset="utf-8">
          <style type="text/css">

               div{
                    background-color: red;
                    height: 150px;
                    width: 150px;
                    display: inline-block;
                    /*NOMBRES DE TRANSICIONES
                    -linear
                    -ease
                    -ease-in
                    -ease-out
                    -ease-in-out*/
                    transition: all 1s ease;
```

```
                }

                div:hover{
                      /*background-color: blue;*/
                      transform: scale(1.2);

                }


           </style>
      </head>
      <body>

           <div></div>


      </body>
</html>
```

4) Deformación:

Hemos visto que haciendo un escalado con valores diferentes para sus ejes, se podía conseguir una deformación. Pero tenemos también otra función para deformar.

```
.deformar {
     transform:skew(30deg,15deg);
     -o-transform:skew(30deg,15deg); /* Para Opera */
     -ms-transform:skew(30deg,15deg); /* Para IE */
     -moz-transform:skew(30deg,15deg); /* Para Firefox */
     -webkit-transform:skew(30deg,15deg); /* Para Safari y Chrome */
}
```

En este ejemplo se hace la deformación a partir de los dos ejes. En el eje X, se deforma 30 grados, mientras que en el Y se deforma 15.

Ejemplo:

```
<!DOCTYPE html>
<html>
      <head>
           <title>Transformación Sesgar</title>
           <meta charset="utf-8">
```

```
          <style type="text/css">

               div{

                    background-color: red;
                    width: 150px;
                    height: 150px;
                    transition: all 1s ease;

               }

               div:hover{
                    transform: skew(25deg,40deg);
               }

          </style>

     </head>
     <body>
          <div></div>
     </body>
</html>
```

CSS – Transiciones

Las transiciones permiten que los cambios de valores en las propiedades de un elemento sucedan de una forma gradual durante un periodo de tiempo determinado, es decir, nos permite suavizar el paso de un estado a otro de la interfaz, muy similar a una “animación”. Un ejemplo claro es el cambio de color del fondo de un botón cuando llevamos el ratón sobre él (hover), con una transición conseguimos que el cambio de color no se haga de forma brusca, sino que se produzca de forma gradual.

Agregando Transiciones con la propiedad Transition

Para aplicar una transición a un elemento tenemos que utilizar la propiedad transition. Aunque está propiedad ya la podemos considerar un estándar, es recomendable el uso de prefijos para que funcione en versiones antiguas de los navegadores.

Parámetros

Transition-property: Especifica el nombre de la propiedad CSS sobre el que se va a realizar el efecto de transición. Utilizamos ‘all’ si

queremos que se aplique sobre todas las propiedades o ‘’none’ para ninguna. Debemos tener en cuenta que no todas las propiedades css son compatibles con las transiciones

Transition-duration: Como su nombre indica, especifica la duración de la transición en segundos o milisegundos (ejemplo: 2s, 150ms).

Transition-timing-function: Especifica la curva de velocidad de la transición, es decir, si queremos que mantenga una velocidad constante, o un comienzo o final rápido, lento, etc. En concreto podemos utilizar los siguientes:

1) Linear: Mantiene la misma velocidad de principio a fin.
2) Ease: Comienzo lento, luego rápido y termina lento.
3) Ease-in: Comienza lento, y después mantiene velocidad.
4) Ease-out: Mantiene velocidad con un final lento.
5) Ease-in-out: Comienzo y fin lentos, muy similar a ease sólo que este último empieza más rápido de lo que termina.
6) Cubic-bezier: Este es un poco más complejo, no voy a entrar en detalle, pero podemos configurar la curva de velocidad a nuestro gusto utilizando la función cúbica de Bezier. Acepta cuatro valores entre 0 y 1.

Transition-delay: Especifica el tiempo de retardo en segundos o milisegundos antes del comienzo de la transición.

Estos parámetros se suelen poner en una sola línea en el orden en el que los he especificado:

Ejemplo:

```
transition:  all 2s ease 1s;
```

También podemos concatenar varias propiedades a la vez:

Ejemplo:

```
transition: background-color 1s ease, height 2s ease-in 1s, margin-bottom 1s linear 3s;
```

(Nota: si no queremos retardo, directamente no lo ponemos).

Aquí podemos ver un ejemplo complejo de transición con retardo:

```
<!DOCTYPE html>
<html>
     <head>
          <title>Transformación compleja</title>
```

```
            <meta charset="utf-8">

            <style type="text/css">

                  #transicion{
                        margin:30px auto;
                        width:200px;
                        height:200px;
                        background-color:#16a085;
                        color:#16a085;
                        line-height:300px;
                        font-size:25px;
                        text-align:center;
                  }

                  #transicion:hover{
                        width:400px;
                        height:400px;
                        border-radius:60%;
                        background-color:#2c3e50;
                        cursor:pointer;
                        padding:30px;
                        box-shadow:0px 2px 20px #000;
                        color:#fff;

                        -webkit-transition: width 1s ease,height 1s
ease,background-color 2s ease-out, border-radius 2s ease-in-out, padding
1s ease 2s, box-shadow 1s ease 2s, color 3s linear;
                        -moz-transition: width 1s ease,height 1s
ease,background-color 2s ease-out, border-radius 2s ease-in-out, padding
1s ease 2s, box-shadow 1s ease 2s, color 3s linear;
                        -o-transition: width 1s ease,height 1s
ease,background-color 2s ease-out, border-radius 2s ease-in-out, padding
1s ease 2s, box-shadow 1s ease 2s, color 3s linear;
                        -ms-transition: width 1s ease,height 1s
ease,background-color 2s ease-out, border-radius 2s ease-in-out, padding
1s ease 2s, box-shadow 1s ease 2s, color 3s linear;
                        transition: width 1s ease,height 1s
ease,background-color 2s ease-out, border-radius 2s ease-in-out, padding
1s ease 2s, box-shadow 1s ease 2s, color 3s linear;

                  }


            </style>

      </head>
      <body>
            <div id="transicion"></div>
      </body>
</html>
```

Conclusiones:

En los ejemplos anteriores podemos observar cómo hemos conjuntado la acción de las transformaciones con las transiciones para que éstas se produzcan de manera suave.

CSS3 nos permite una forma muy sencilla y comprensible de realizar transformaciones a objetos. Los únicos inconvenientes son, como siempre, la compatibilidad entre navegadores, y tener que poner todos los prefijos en las instrucciones, para hacer las funciones compatibles. Recordad que si no ponemos los prefijos (-o-, -ms-, -moz-, -webkit-), no funcionará en ese navegador. O muy probablemente no funcione. Siempre depende de la versión que utilicéis.