



Diseño Web Responsive HTML5 y CSS3

Unidad 7: Tipografías, Íconos y Animación con CSS3



Indice

Unidad 7: Tipografías, Íconos y Animación con CSS3

Animación CSS3

Transform





Objetivos

Que el alumno logre:

- Conocer e implementar los nuevos elementos incorporados en CSS3 para optimizar el diseño.





Animaciones CSS

Una de las características más impresionantes de CSS 3 es la posibilidad de crear animaciones de los elementos de la página sin utilizar programas o plugins.

Todo ello nos abre infinitas posibilidades que antes solo estaban disponibles para los programadores Javascript o diseñadores con Flash.

Como todos sabemos, hasta el momento, las animaciones en las páginas web se realizaban utilizando diversas tecnologías accesorias, más allá del simple HTML o CSS.

El primer sistema que alcanzó gran popularidad para realizar una animación de elementos bastante fluida fue la tecnología Flash y luego lo acompañaron algunos otros sistemas como Silverlight, de características similares. Sin embargo, todo esto son tecnologías propietarias, que requieren la instalación de un plugin para funcionar en el navegador, lo que impide que sean universales, por mucha aceptación que hayan llegado a tener.

Paralelamente existen varios otros soportes para animación que sí forman parte de las tecnologías de creación de páginas web universales, pero que no llegan ni de lejos a las posibilidades de animación que podríamos desear. Nos referimos a los GIF animados, que tanto se utilizaron en los comienzos del desarrollo web, así como a Javascript que también permite hacer animaciones a base de cambiar atributos CSS de manera progresiva a lo largo de un tiempo.

Con CSS 3 viene una nueva forma de realizar animaciones totalmente novedosas y que resultará mucho más sencilla que el uso que podemos conocer con Javascript. Pero lo que es más importante, que soporta muchos más tipos de animación que hasta ahora estaban reservados a tecnologías como Flash, como pueden ser rotaciones, ampliaciones y reducciones del tamaño vectoriales, etc.

Esto no se queda ahí, ya que además se han implementado una ciertas interacciones con el usuario y que se consiguen únicamente con CSS 3. Además, todo ello sin tener que utilizar ningún lenguaje de programación, lo que puede resultar mucho más agradable y al alcance de los desarrolladores menos técnicos.

Ventajas de las animaciones CSS 3

Las animaciones CSS permiten hacer muchas de las cosas que antes teníamos reservadas sólo al uso de tecnologías supletorias, que no hacían más que incrementar la dificultad del desarrollo, limitar su compatibilidad entre distintos tipos de usuarios y plataformas, así como los requisitos de conocimientos del desarrollador para poder incorporarlas.



Por tanto, una de las ventajas es que nos podemos olvidar de Flash si queremos hacer elementos con dinamismo en nuestra web. Dejar a Flash de lado además implica que no tenemos que preocuparnos por el posicionamiento de la página que tantos dolores de cabeza provoca cuando nuestra web esta creada enteramente en Flash.

Sin embargo, las ventajas más importantes serían la compatibilidad y la facilidad de implementación, al usar un lenguaje que ya resulta familiar para el desarrollador. La compatibilidad viene dada por el uso de un sistema abierto y regulado por el W3C, al que todos los navegadores tarde o temprano se adaptarán. Y la facilidad de desarrollo porque sólo trabajaremos en nuestros sitios con el lenguaje CSS y no existirá la necesidad de dominar otros lenguajes de programación como ocurría con Flash.

Inconvenientes de las animaciones CSS

También existen algunas desventajas al trabajar con animaciones en CSS. Lo cierto es que la mayor que se podría destacar es sólo circunstancial, debido al poco soporte que existe actualmente a esta utilidad. Tenemos dos principales inconvenientes.

Las animaciones CSS no son admitidas por los muchos navegadores (Ninguna utilidad para animación con CSS 3 se puede utilizar en Internet Explorer y en Firefox algunas cosas ya podemos ver que funcionan a medias, pero aún le queda largo camino por recorrer).

Consume bastantes recursos de máquina para producir las animaciones.

También podremos encontrar que existe alguna dificultad a la hora de la programación, pero no más de la que encontraríamos si tuviésemos que utilizar otros lenguajes o tecnologías distintos de CSS.

Finalmente, volvemos a remarcar que, debido a la imposibilidad de ver los resultados en todos los clientes web, al menos por el momento, deberemos utilizar navegadores basados en Webkit, como son Safari o Google Chrome (siempre en su versión más actualizada).

Algunos conceptos necesarios para comprender las animaciones en CSS.

Fotograma clave

Los *fotogramas claves* son valores iniciales y finales que debe tener la animación CSS. Estas localizaciones, en teoría, las sabemos a ciencia cierta, es decir, siempre conocemos en qué punto vamos a empezar y en cual vamos a terminar la animación, así como su duración. Pero podemos crear otros fotogramas clave, no



solamente los de inicio y fin, que correspondan con puntos intermedios del movimiento. Las reglas que determinan estos valores es lo que llamamos fotogramas clave dentro de CSS.

Su sintaxis sería algo así:

```
@keyframes 'nombre_fotograma_clave' {
```

```
    puntodelKeyframe {  
        atributos iniciales;  
    }
```

```
    puntodelKeyframe {  
        nuevos atributos;  
    }
```

```
    puntodelKeyframe {  
        últimos atributos;  
    }
```

```
}
```

En el código real se traduciría así:

```
@keyframes 'animacion' {
```

```
    0% {  
        left: 100px;  
    }
```

```
    40% {  
        left: 150px;  
    }
```

```
    60% {  
        left: 75px;  
    }
```

```
    100% {  
        left: 100px;  
    }
```

```
}
```



Esta animación estaría compuesta de 4 fotogramas clave, el porcentaje es en el momento de la animación en el que va a producirse ese fotograma y los px son la longitud y la alineación donde se colocaría el fotograma dentro del DIV en que se encaje.

El código de dicho DIV sería el siguiente:

```
div {  
  animation-name: 'nombre-fotograma-clave';  
  animation-duration: 45s;  
  animation-iteration-count: 10;  
}
```

Los atributos de estilo para esta capa que se ven en el código anterior son los siguientes:

animation-name: el nombre del fotograma clave.

animation-duration: la duración de la animación.

animation-iteration-count: la veces que se repite.

Propiedades sobre la animación aplicables en el DIV

Además de las propiedades que vimos en el párrafo anterior, tenemos otra serie de atributos que se pueden aplicar a la animación y que se colocan en el DIV.

Esta sería una lista de las propiedades adicionales, aplicables para definir las animaciones que especificamos en el DIV:

animation-timing-function: se aplica entre los fotogramas clave, no sobre toda la animación y describe como progresa la animación a lo largo de un ciclo.

animation-direction: esta propiedad define el sentido de la animación. Si especificamos “alternate” y los ciclos de interacción son impares, la animación irá en la dirección normal, si no, se realizará en la dirección inversa

animation-delay: propiedad que nos indica el momento en el que comenzará la animación. Si el valor es 0 se ejecuta en cuanto se carga la página.

animation: esta propiedad combina las anteriores de una forma resumida.



Código completo para una animación CSS

A continuación veremos un código CSS donde estamos definiendo una animación:

```
div {  
  animation-name: 'movimiento-diagonal';  
  animation-duration: 5s;  
  animation-iteration-count: 10;  
}  
  
@keyframes 'movimiento-diagonal' {  
  from {  
    left: 0;  
    top: 0;  
  }  
  
  to {  
    left: 100px;  
    top: 100px;  
  }  
}
```

Este ejemplo lo que nos mostraría sería una animación en la que se mueve un elemento de la esquina inferior izquierda a la esquina superior derecha, ese movimiento va a tardar 5 segundos y se va a repetir 10 veces.

(NOTA: por el momento este atributo solo funciona en Chrome y Safari, por lo tanto a las propiedades debemos agregarle -webkit-)

Animación de texto

Lo primero que tenemos que hacer es crear nuestros fotogramas clave, para ello utilizamos el siguiente código en nuestra hoja de estilos:

```
@-webkit-keyframes movimiento-diagonal {  
  from {  
    left: 0px;  
  }  
}
```




```
to {  
  left: 100px;  
}  
}
```

A través de este código le indicamos el inicio y el fin de nuestra animación.

Ahora crearemos un estilo con selector de id, para darle forma a nuestro div:

```
#anim {  
  -webkit-animation-name: movimiento-diagonal;  
  -webkit-animation-duration: 3s;  
  -webkit-animation-iteration-count: infinite;  
  -webkit-animation-direction: alternate; /*para que vuelva a su posicion inicial */  
  width: 100px;  
  background-color: Teal;  
  color: #fff;  
  position: relative;  
  padding: 2px;  
}
```

En la primera línea le damos el *nombre a la animación*, que tiene que ser el mismo que el del fotograma clave.

En la segunda le damos una *duración* de 3 segundos, es decir, la animación tardará en hacer el recorrido sólo 3 segundos.

En la tercera le decimos que lo *repita* infinitas veces.

La propiedad *-webkit-animation-direction: alternate* hace que el texto, una vez que haga el recorrido, vuelva a su posición inicial realizando el camino inverso.

Y por último le damos un *ancho*, y *color de fondo* y de *texto*, así como una *posición relativa*, ya que de lo contrario no funcionaría nuestra animación.

(Recuerden que por el momento estos atributos son válidos solamente en Safari y Chrome)



Atributo Transform

Un atributo que podemos combinar con las animaciones en CSS, es el atributo *transform*.

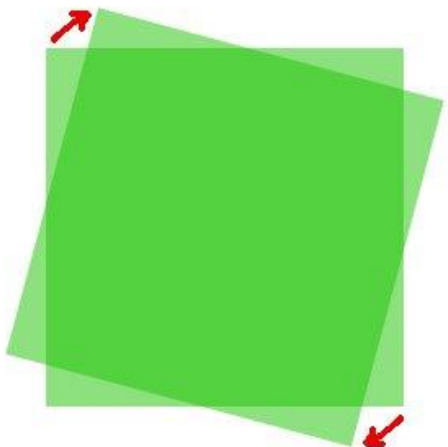
El atributo *transform* nos permite, como su propio nombre indica, transformar un elemento. Su sintaxis es la siguiente:

```
transform: tipo(cantidad);
```

El valor tipo puede tomar cuatro valores, y cada uno de ellos realiza una función diferente:

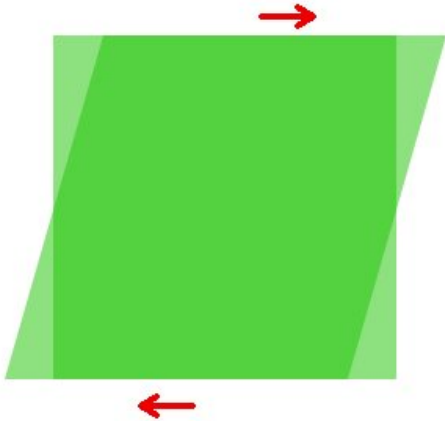
Rotate. Nos permite girar los elementos un número de grados. La sintaxis es:

```
transform: rotate(25deg);
```



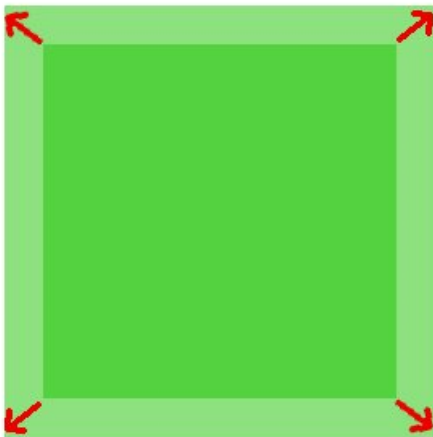
Skew. Podemos inclinar un elemento tanto en coordenadas X como Y. El valor se expresa en grados y la sintaxis es la siguiente:

```
/*transform: skew(gradosX, gradosY);*/  
transform: skew(15deg, 3deg);
```



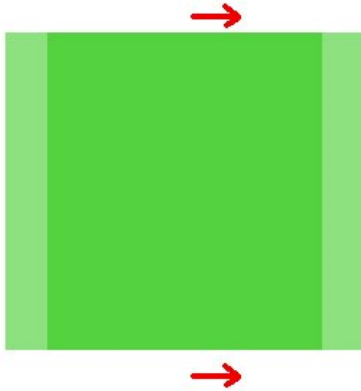
Scale. Con este tipo podremos escalar nuestro elemento tanto en X como en Y en una cantidad expresada en tantos por uno:

```
/*transform: scale(escalaX,escalaY);*/  
transform: scale(1.5,0.6);*/
```



Translate. Podemos desplazar el elemento tanto en X como en Y.

```
/*transform: translate(desplazamientoX, desplazamientoY);*/  
transform: translate(12px, 19px);
```



Se pueden aplicar diferentes transformaciones a un mismo elemento simplemente escribiéndolas de manera consecutiva:

```
transform: scale(1.6) skew(10deg) translate(5px) rotate(12deg);
```



Resumen

En esta Unidad...

Trabajamos con Tipografías, Íconos y Animación con CSS3.

En la próxima Unidad...

Comenzaremos a trabajar con Responsive Design.