

Gradientes

CSS3 ha agregado la opción de crear gradientes (fondos en degradé) sin la necesidad de usar imágenes.

Los gradientes en CSS son de dos tipos: lineales (`linear-gradient`) o radiales (`radial-gradient`). En el gradiente lineal, la transformación de color va avanzando línea a línea, mientras que en el radial, la transformación de color se produce debido a que sucesivos círculos concéntricos van cambiando de color.

La propiedad que utilizamos para realizar gradientes lineales, se llama `linear-gradient` y esta se agrega al atributo `background`, que ya vimos con anterioridad.

Esta propiedad maneja dos opciones de parámetros, podemos elegir el punto de inicio o sea si queremos que lo aplique arriba, abajo, a la derecha o a la izquierda de nuestra caja o podemos elegir los grados de inclinación que queremos que tenga nuestro gradiente.

Los puntos de inicio pueden ser `top`, `right`, `left` o `bottom`.

Entonces, supongamos que queremos aplicarle a nuestra caja un gradiente que va de negro a gris desde la parte de arriba de la misma. La sintaxis sería la siguiente:

```
div { background: linear-gradient (to top, #000, #ccc) }
```

Si lo que queremos es aplicar un gradiente de 90°, entonces la sintaxis sería así:

```
div { background: linear-gradient (90deg, #000, #ccc) }
```

Algunas cosas a tener en cuenta, la propiedad `linear-gradient` no genera un color de fondo, sino una imagen sin dimensiones específicas, esta se va a adaptar automáticamente para cubrir todo el espacio disponible.

Por otro lado, no todos los navegadores soportan esta propiedad.

Transformaciones

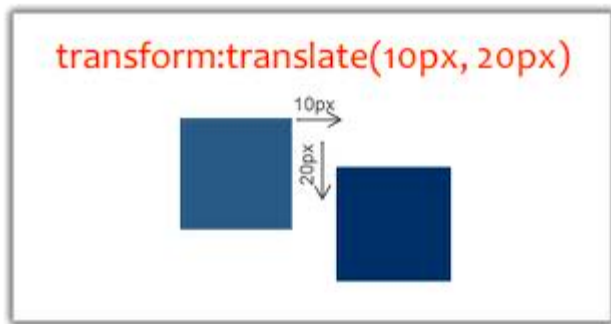
Las transformaciones CSS ofrecen la posibilidad de modificar el desplazamiento, escala, rotación, sesgo de elementos y en combinación con las transiciones nos permite crear animaciones css modificando gradualmente sus propiedades.

De las transformaciones de CSS3 en 2D, las más usadas son:

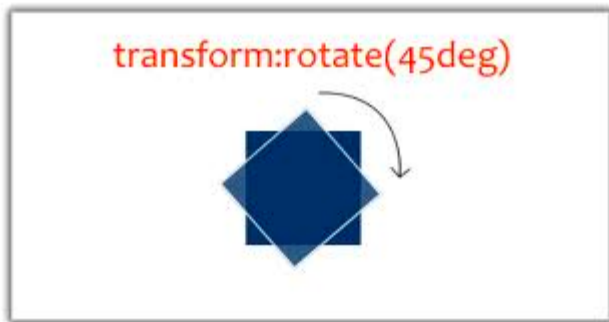
- **Rotate:** Rotate te permite rotar un elemento dándole un ángulo de giro en grados.
- **Scale:** Scale te permite escalar un elemento, toma valores positivos y negativos y se le pueden poner decimales.
- **Translate:** Translate nos permite trasladar un elemento a la vez en el eje de las X y de las Y, dándole las coordenadas iniciales y finales.
- **Trasladar:** Esta propiedad de CSS3 no hace propiamente una transición, sino que hace que un elemento pase de estar en una posición a estar en otra. Con las transiciones y las animaciones de CSS3 vamos a poder darle un efecto visual de movimiento.

Ejemplo:

```
transform: translate(10px, 20px);
```



Rotación: Se puede aplicar tanto a elemento inline como a elementos de bloque.

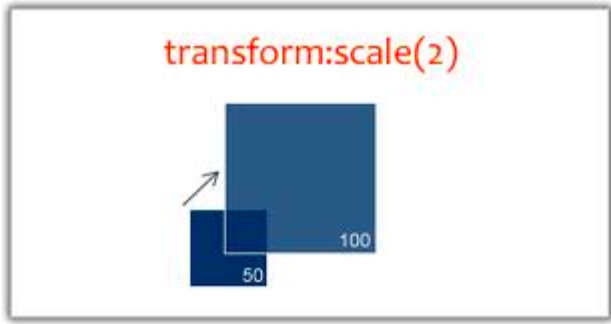


```
.ejemplo { transform: rotate (45deg); }
```

Los grados se marcan en positivo si el elemento se rota en el sentido de las agujas del reloj y en negativo si es al revés.

Por defecto, el punto de referencia que toma como eje para hacer la rotación es el centro, pero también se puede especificar otro punto.

Escalar: Otro punto interesante va a ser escalar un elemento con CSS3.



Si el valor es positivo, el elemento se hace más grande, y si es negativo se hace más pequeño.

Transiciones

Las transiciones permiten controlar la velocidad a la que se van a modificar las propiedades de un elemento inclusive se pueden aplicar curvas de aceleración permitiendo de esta forma crear animaciones.

Animaciones

Una animación tiene lugar en el tiempo, con lo que vamos a tener que tomar diferentes puntos dentro de un fragmento de tiempo para especificar que sucede en cada uno de ellos. Es lo que se llaman **keyframes**. Para los que ya han trabajado en Flash u otros programas de animación ya conocen el concepto. En cada keyframe podemos introducir un cambio y la suma de estos cambios es lo que va a dar lugar a la animación final.

Para indicar la duración de una animación utilizaremos la propiedad `animation-duration`.

La velocidad de la animación estará gestionado por la propiedad `animation-timing-function`.

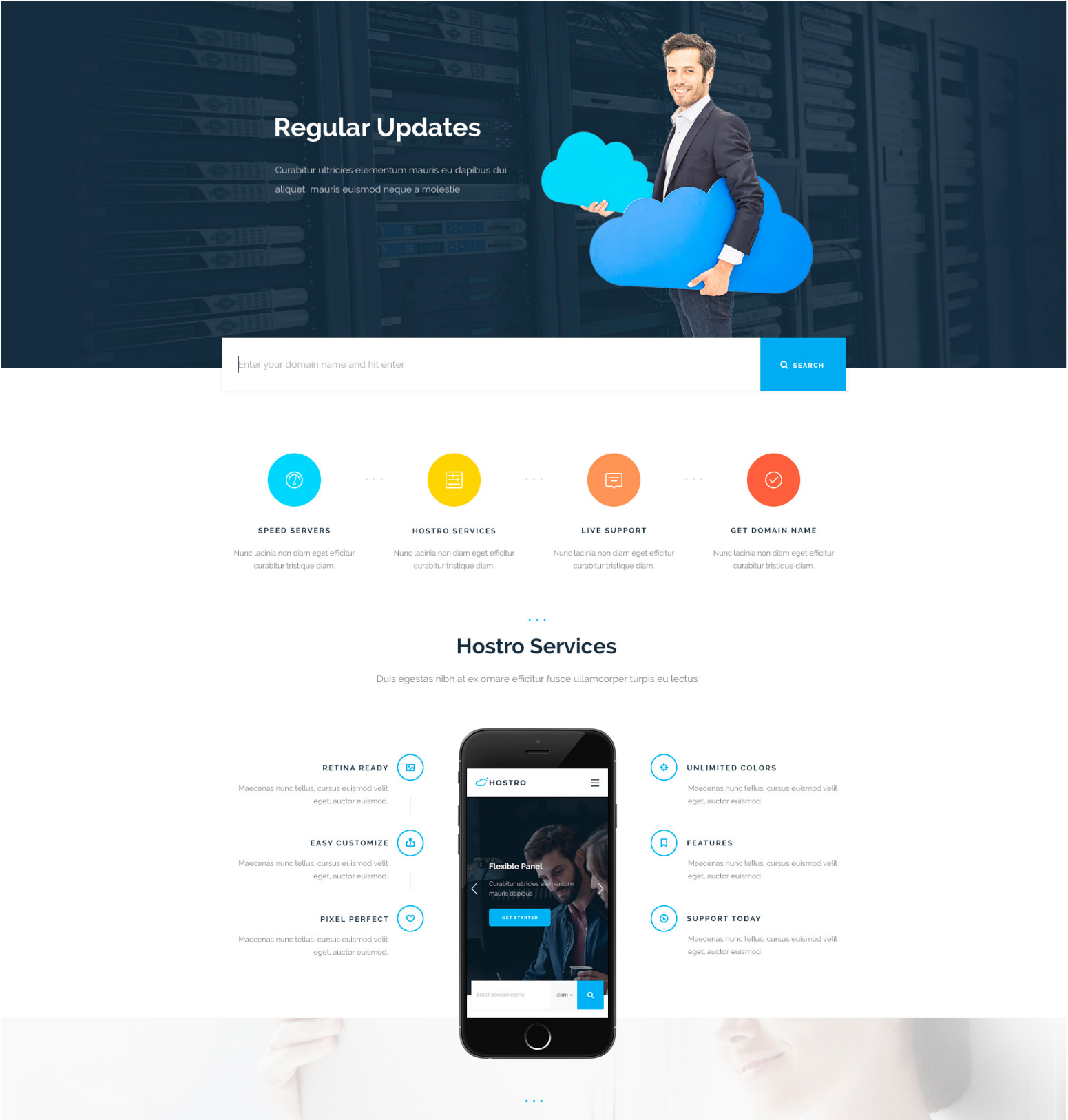
También en las animaciones de CSS3 podemos especificar un tiempo de espera antes de iniciar la animación con la propiedad `animation-delay`.

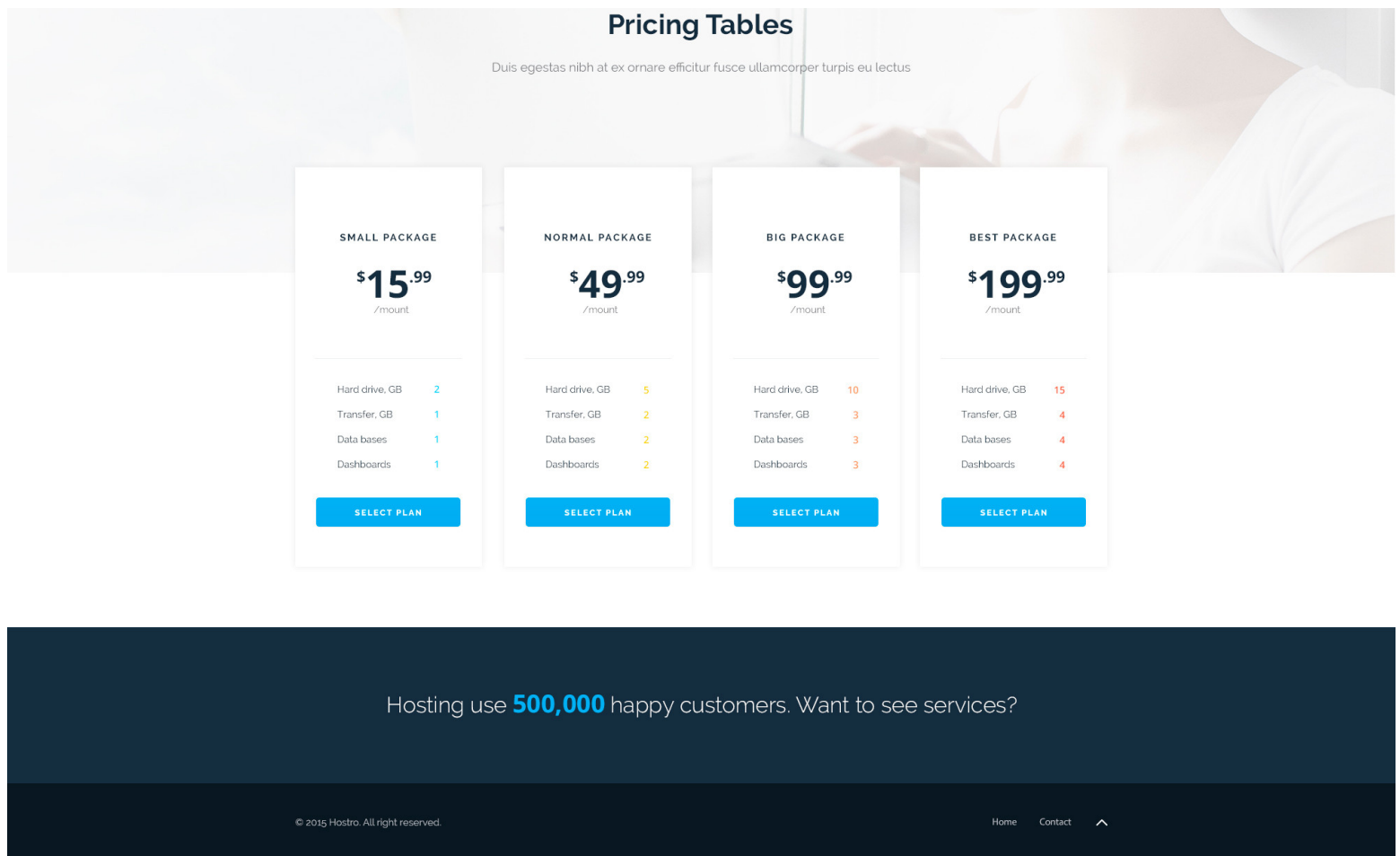
Con la propiedad `animation-direction` podremos indicarle que la propiedad se haga en sentido inverso al habitual, es decir, que empiece por el final.

Para aprender más sobre esto, pueden ver <http://daneden.github.io/animate.css/>

Trabajo Práctico 5 CSS

Descripción:





1. Abajo del primer input hay cuatro iconos, llamarlos con la clase iconos y en cada uno seguir las instrucciones:
2. Cuando sucede el evento hover hacer que los iconos aumente gradualmente de tamaño con las propiedades:
 - a. Transition on hover: 1.3s ease-in-out;
 - b. Transformación de escala on hover: 1.25

Objetivo:

- Tener una noción básica de animaciones con CSS3.
Olvidate de Flash y de los Gifs animados al estilo: <http://www.homerswebpage.com/>

Requerimientos:

1. Haber realizado los anteriores tp.
2. Aplicar las propiedades correspondientes.
3. Llamarlos con la clase iconos (.iconos) dentro de un div.
4. Subir la resolución de tu trabajo a tu repositorio de Github en la subcarpeta “CSS TP5”.

Bonus:

1. Investiga y aplica otras subpropiedades animation como por ejemplo:
 - a. animation-delay
 - b. animation-direction entre otros

Recursos:

- [Animation CSS](#)
- [Usando Animaciones con CSS](#)

Resolución

```
.iconos:hover{
  -webkit-transform: scale(1.35);
  transform: scale(1.35)
}
.iconos {
  -webkit-transition: 1.3s ease-in-out;
  -moz-transition: 1.3s ease-in-out;
  -o-transition: 1.3s ease-in-out;
  transition: 1.3s ease-in-out;
}
```