LAS NUEVAS REGLAS

- Con CSS3 no solo definimos la apariencia y el formato .
- Con el fin de reducir el uso de código JS y estandarizar las carácterísticas más apreciadas CSS3 no se ocupa solo de diseño y estilos web, sino **tambien de la forma y el movimiento**
- La especificación CSS3 se presenta en módulos que proporcionan una especificación estandar para todos los aspectos implicados en la presentación viusal del documento: esquinas redondeadas y sombras….transformaciones y reorganización de elementos

Border radius

- Varias formas de declararlo:
  border-radius: 20px
  border radius: 20px 20px 20px 20px
  border radius: 20px 20px

- Tambien es posible modificar las curvas, proporcionando nuevos valores separados por una barra inclinada (/). Los valore sa la izquierda de la barra reprentan el radio horizontal y los valores a la derecha, el radio vertical: la combinación de estos valores genera una elipse

```
body {
  text-align: center;
}
#mainbox {
  display: block;
  width: 500px;
  margin: 50px auto;
  padding: 15px;
  border: 1px solid #999999;
  background: #FFFFFF;

  border-radius: 20px / 10px;
}
#title {
  font: bold 36px verdana, sans-serif;
}
```

Box-shadow

- Box-shadow: rgb(150,150,150) 5px 5px;

Esta propiedad necesita por lo menos tres valores : color, (en rgb o hexadecimal, como lo hicimos antes para otras propiedades.

Los dos siguientes valores expresado en pixeles, son el desplazamiento de la sombra (primero horizontal y luego vertical).
Admiten números positivos y negativos

Si añadimos un 4 valor podremos establecer la distancia de desenfoque y hacer que la sombra tenga una apariencia más real

Si añadimos un 5 valor. (la palabra clave **inset.** Convierte la sombra externa en una sombra interior que proporciona un efecto de profundidad a la caja)

box-shadow: rgb(150,150,150) 5px 5px 10px inset;

Text-Shadow

- Los 4 parámetros del anterior

```
body {
  text-align: center;
}
#mainbox {
  display: block;
  width: 500px;
  margin: 50px auto;
  padding: 15px;
  border: 1px solid #999999;
  background: #FFFFFF;
}
#title {
  font: bold 36px verdana, sans-serif;
  text-shadow: rgb(150, 150, 150) 3px 3px 5px;
}
```

@font-face

1. Primero importamos la tipografía en formato ttf y lo guardamos en nuestra carpeta raíz.
2. Con la siguiente sentencia relacionamos la fuente con nuestra web (ponemos el nombre que quieras)
3. @font-face {
   font-family:'mifuente'
   Src: url('font.ttf')
4. Aplicar la fuente como siempre mediante el font face

EJECICIO. TIPOGRAFIA

```
body {
  text-align: center;
}
#mainbox {
  display: block;
  width: 500px;
  margin: 50px auto;
  padding: 15px;
  border: 1px solid #999999;
  background: #FFFFFF;
}
#title {
  font: bold 36px MyNewFont, verdana, sans-serif;
}
@font-face {
  font-family: 'MyNewFont';
  src: url('font.ttf');
}
```

Linear-gradient

- Degradado lineal
- Sintaxis linear-gradient(top, transparent, red)
- Valores:
  - Primero (ej Top) posición inicio
  - Segundo (ej red) color inicial
  - Tercer (ej blu) color final
- Primer valor Esquinas: top right……
- Prefijo to: invierte el gradiente

EJECICIO. GRADIENTE LINEAL UTILIZANDO 2 ESQUINAS (TOP RIGHT...)

```
background: -webkit-linear-gradient(top right, #FFFFFF, #666666);
background: -moz-linear-gradient(top right, #FFFFFF, #666666);
```

EJECICIO. CREAR UN DEGRADADO CON 30 GRADOS

```
background: -webkit-linear-gradient(30deg, #FFFFFF, #666666);
background: -moz-linear-gradient(30deg, #FFFFFF, #666666);
```

EJECICIO. CREAR UN DEGRADADO LINEAL MULTICOLOR

```
background: -webkit-linear-gradient(top, #000000, #FFFFFF, #999999);
background: -moz-linear-gradient(top, #000000, #FFFFFF, #999999)
```

EJECICIO. DEGRADADO LINEAL CON TRANSPARENCIA

```
body {
  text-align: center;
  background: url(http://www.formasterminds.com/content/bricks2.jpg);
}
#mainbox {
  display: block;
  width: 500px;
  margin: 50px auto;
  padding: 15px;
  border: 1px solid #999999;

  background: -webkit-linear-gradient(top, transparent, #666666);
  background: -moz-linear-gradient(top, transparent, #666666);
}
#title {
  font: bold 36px verdana, sans-serif;
}
```

EJECICIO. DEGRADADO LINEAL CON PORCENTAJE

```
background: -webkit-linear-gradient(top, #FFFFFF 50%, #666666 90%);
background: -moz-linear-gradient(top, #FFFFFF 50%, #666666 90%);
```

RADIAL GRADIENT

- Degradado radial
- Sintaxis (radial-gradient (center, elipse, red, blue)
- Primer valor: posicion inicio (en pixeles o porcentaje o palabra clave-center, top…)
- Excepto por el parámetro de forma **(circulo o elipse)**, el resto de la funcion es exáctamente igual que lineal gradient
- La posición puede ser personalizada y podemos utilizar vrios colores con el segundo valor de color-stp para determinar el límite de cada uno de ellos

EJECICIO. DEGRADADO RADIAL

```
background: -webkit-radial-gradient(center, ellipse, #FFFFFF, #000000);
background: -moz-radial-gradient(center, ellipse, #FFFFFF, #000000);
```

EJECICIO. Multicolor degradado radial

```
background: -webkit-radial-gradient(30px 50px, ellipse, #FFFFFF 50%, #666666 70%, #999999 90%);
background: -moz-radial-gradient(30px 50px, ellipse, #FFFFFF 50%, #666666 70%, #999999 90%)
```

rgba

- R.Rojo
- G green
  B blue
  A transparencia

EJECICIO. Sombra con transparencia

```
#title {
  font: bold 36px verdana, sans-serif;
  text-shadow: rgba(0,0,0,0.5) 3px 3px 5px;
}
```

hsla

- Tono (expresa de 0 a 360 (360 son rojos, 120 verdes y 240 azules)
- Saturacion(0% escala de grises, 100% completamente saturado,)
- Luminosidad (igual que saturacion.Y 50% perfectamente iluminado)
- opacidad

EJECICIO. HSLA

```
#title {
  font: bold 36px verdana, sans-serif;
  text-shadow: rgba(0,0,0,0.5) 3px 3px 5px;
  color: hsla(120, 100%, 50%, 0.5);
}
```

OUTLINE

- Para incluir un desplazamiento (2º borde separado del borde del elemento)
- 2PX DASHED #000000
  outline-offset: 15px
- Añadimos a los estilos asignados originalmente a la caja de la cabecera, un borde exterior de 2px con un desplazamiento de 15 px.
- La porpiedad outline tiene características similares y utiliza los mismos parámetros de border
  la propiedad outline-offset solo tiene un valor en pixeles

EJECICIO. Outline. Doble linea borde

```
#outline {
  display: block;
  width: 500px;
  margin: 50px auto;
  padding: 15px;
  border: 1px solid #999999;
  background: #FFFFFF;

  outline: 2px dashed #000000;
  outline-offset: 15px;
}
```

BORDER IMAGE

- Tanto para border como para outline, se complementan con la propiedad border image para introducir una imagen en borde (la imagen debe tener 29 px)
  sintaxis border-image url() 29 stretch
  - Stretch(distribuir la imagen en una sola pieza)
  - Round (calcula la longitud y estira las piezas)
  - Repeat (se repite tantas veces como sea necesario para cubrir la pieza)

BACKGROUND

- La propiedad background ha incorporado nuevos parámetros y características tales como el tamaño, la posición o el área de pintura, y puede incluir varias imágenes

  Nueva sintaxis: background: color imagen posición tamaño repetir origen clip documento-adjunto
- Forma abreviada de mencionar las siguientes propiedades:
  - Bacground-color: asigna un color de fondo a un elemento. El valor puede ser hexadecimal , números decimales utilizando la función rgb() por ejemplo rgb (255, o, o), o una palabra clave como yellow, red….

BACKGROUND

- Background-position: Declara la posición de partida de una imagen de fondo. Los valores se pueden especificar en píxeles o porcentajes utilizando una combinación de las siguientes palabras clave: center, left, right, top bottom
- Background-size: Declara el tamaño de la imagen de fondo. Los valores se pueden indicar como porcentaje, como pixeles o por las palabras clave cover (escala la imagen hasta ajustarla al elemento en anchura y altura) y contain (escala la imagen para encajarla entera)
- Background-repeat: Determina la forma en la que la imagen de fondo se distribuye utilizando cuatro palabras clave: repeat, repeat-x, repeat-y y no-repear

Background

- Fondo-clup: Declara el área a pintar utilizando 3 palabras clave: border-box, padding-box y content-box. La primera muestra la imagen hasta el borde de la caja, la segunda hasta relleno de caja y la tercera solo hasa contenido de caja.
- Background-origin:utiliza las mismas palabras clave que Background-clip para ajustar la imagen respecto al borde, el relleno o el contenido de la caja.
- Background-attachment: Esta propiedad determina si la imagen está fija o se desplaza con el resto de los elementos y para ello utiliza las plabras clave scroll y fixed. La primera es el valor por defecto y hace que la imagen se desplace con la página y la segunda,fixed, fija la imagen de fondo en el lugar en el que fue declarada

EJECICIO. BACGROUND

```
#mainbox {
  display: block;
  width: 500px;
  margin: 50px auto;
  padding: 15px;
  border: 1px solid #999999;

  background: #CCCCCC url("bricks2.jpg") 10px 10px / 510px 55px
no-repeat scroll border-box;
}
```

Columnas

Distribuir un contenido en varias columnas mediante una simple regla

Column-count:2

Column-gap:10px

Column-rule: 1px solid #000000

Column-count: Declara el número de columnas a crear

Column-gap: declara la diferencia entre las columnas

Column-rule especifica una división visible entre las columnas que aparecen en pantalla

Columnas

- Otras propiedades disponibles para personalizar las columnas:

  Column-width: seclara un ancho específico para las columnas (valores: auto o una longitud válida en CSS)

  Column-span: Se aplica a los elementos dentro de la caja y determina si el elemento se colocará en el inerior de una colmna o será distribuido en varias columnas

  Valores: all: todas las columnas

  none: aplicado por defecto

EJECICIO. GENERAR COLUMNAS

```
#mainbox {
  display: block;
  width: 500px;
  margin: 50px auto;
  padding: 15px;
  border: 1px solid #999999;
  background: #FFFFFF;

  column-count: 2;
  column-gap: 10px;
  column-rule: 1px solid #000000;
}
```

FILTROS

- Para añadir efectos a una imagen: filter
- Sintaxix filter: blur(5px);
- Valores:
  - Blur: produce un efecto de desenfoque. Necesita un valor en píxeles de 1px a 10px
  - Grayscale Convierte gradualmente los colores del elemento a una escala de grises. Se necesita un número decimal de 0.1 a 1
  - Drop-shadow (x,y,tamaño,color) . Aplica una sombra simple al elemento

    Atributos x, y determinan la ditancia entre la sombra y el elemento, el atributo tamaño especifica las dimensiones de la sombra y color declara su color

- Sepia (valor) Proporciona a los colores del elemento un tono sepia (necesita un número decimal de 0.1 a 1)
- Brightness (valor): cambia la luminosidad del elemento (necesita un número decimal de 0.1 a 10)
- Contrast (valor): Cambia el contraste del elemento. Necesita un número ecimasl de 0.1 a 10)
- Hue-rotate (valor) Aplica un giro a la tonalidad del elemento. Necesita un valor en grados de 1 a 360 deg
- Invert: invierte los colores del elemento y produce un negativo. Necesita un numero decimal de 0.1 a 1
- Saturate. Satura los colores del elemento. Necesita un numero decimal de 0.1 a 1
- Opacity: produce un efecto de opacidad. Necesita un numero decimal de 0 a 1(donde 0 establece transparencia completa y 1 opacidad completa)

EJECICIO. APLICAR UN FILTRO

```
#mainbox {
  display: block;
  width: 500px;
  margin: 50px auto;
  padding: 15px;
  border: 1px solid #999999;

  background: #CCCC99
url("bricks2.jpg");
  -webkit-filter: blur(5px);
```

Transformar

- La propiedad trasnform puede aplicar cuatro transformaciones básicas a un objeto:
  1. **Scale**
  2. **Rotate**
  3. **Skew**
  4. **Translate**

Transform:scale

- Acepta valores enteros y decimales
  - Entre 0 y 1 reducen el elemento, 1 mantiene las proporciones originales y los valores mayores que 1 aumentan las dimensiones.
  - Usando valores negativos se obtiene un resultado interesante
- Usa dos parámetros:
  - X para la escala horizontal
  - Y para la escala vertical

(en caso de proporcionar solo un valor, este se aplica a ambos parámetros)

EJECICIO. ESCALAR

```
#mainbox {
  display: block;
  width: 500px;
  margin: 50px auto;
  padding: 15px;
  border: 1px solid #999999;
  background: #FFFFFF;

  -webkit-transform: scale(2);
  -moz-transform: scale(2);
}
```

Transform:ROTATE

- Gira en sentido de las agujas del reloj.
  Valor especificado en grados (deg)
  transform: rotate(30deg)

EJECICIO. ROTAR UN ELEMENTO

```
#mainbox {
  display: block;
  width: 500px;
  margin: 50px auto;
  padding: 15px;
  border: 1px solid #999999;
  background: #FFFFFF;

  -webkit-transform: rotate(30deg);
  -moz-transform: rotate(30deg);
}
```

Transform:skew cambia la simetría del elemento en ambas dimensiones utilizando también valores en grados **deg**

EJECICIO. SESGAR UN ELEMENTO

```
#mainbox {
  display: block;
  width: 500px;
  margin: 50px auto;
  padding: 15px;
  border: 1px solid #999999;
  background: #FFFFFF;

  -webkit-transform: skew(20deg);
  -moz-transform: skew(20deg);
}
```

Transform translate Similar a las propiedades top y left, . mueve el elemento a una nueva ubicación en la pantalla

EJECICIO. MOVER DE SITIO UN ELEMENTO

```
 mainbox {
 display: block;
 width: 500px;
 margin: 50px auto;
 padding: 15px;
 border: 1px solid #999999;
 background: #FFFFFF;

 -webkit-transform: translate(100px);
 -moz-transform: translate(100px);
}
```

TRASNFORMAR TODO UN ELEMENTO

- Para transformas todo en un mismo elemento. (propiedad transform compuesta) solo tiene que separar las funciones con un espacio: translateY(100px) rotate(45deg) scale(0.3);

EJECICIO. Mover, escalar y rotar un documento

```
  mainbox {
  display: block;
  width: 500px;
  margin: 50px auto;
  padding: 15px;
  border: 1px solid #999999;
  background: #FFFFFF;

  -webkit-transform: translateY(100px) rotate(45deg) scaleX(0.3);
  -moz-transform: translateY(100px) rotate(45deg) scaleX(0.3);
}
```

TRANSFORMACIONES DINÁMICAS

- Podemos sacar ventaja de las combinacion de transformaciones y pseudo clases para convertir nuestro documento en una aplicación dinámica.
  #conendor:hover{transform:rotate(5deg);
  }
- Resultado cada vez que el puntero del raton esta sobre la cabecera, transform le hace girar 5° y cuando el puntero del ratón está sobre la cabeceera regrasa a su posición anterior. De esta manera se logra una animacion básica pero útil utilizano solo propiedades CSS

EJECICIO. RESPONDER A UNA ACCION DEL USUARIO

```
#contenedor{
  display: block;
  width: 500px;
  margin: 50px auto;
  padding: 15px;
  border: 1px solid #999999;
  background: #FFFFFF;
}
#contenedor:hover{
  -webkit-transform: rotate(5deg);
  -moz-transform: rotate(5deg);
}
```

TRASNFORMACIONES 3D

- Transform perspective(500px) rotate3D (0,1,0, 45deg)
- Excepto perspective, el resto de las funciones 3d disponible spara la propiedead transform sonsimilares a las de los efectos 2D:

Perspective. Añade profundida a la escena, haciendo los elementos m´sa grandes cuando están más cerca del espectador

Trasnlate 3d(x,y,z) mueve el elemento a una bueva posocion en el espacio 3d. Necesita 3 valoer en pixeles para los ejer x,Y y Z.

Scale3d(x,y,z). Proporciona una nueva escala en el espacio 3D. Necesita 3 valores en números decimales para establecer la ecala de los ejes XY Z . Como en las trasnformaciones 2D, un valor igual a 1 manteiene la escala original

TRASNFORMACIONES 3D

- ROTAR 3D. HACER GIRAR EL elemento de acuerdo con los valores proporcionados para cada eje y angulo. Hay que indicar los valore de los ejes en números decimales y el ángulo en grados (por ejemplo 30dg) Los valores para los ejes determina un vector para la rotación, por lo que lo realmente importante es la relacion etre los valores y no los valores en simismos ej rotate 3D(5,2,6,30deg) producira el mimo efexto que rotate3d(50,20,60,30deg) debido a que el vector resultante es el mismo

Transformaciones 3D

- Hay otras porpiedades disponibles que permiten conseguir un efecto más ralista
  - Perspective. Igual quela funcion perspective() pero funciona en un caja padre. Crea una caja conenedore y aplica el ejecto de perspectiva a sus elementos hijos
  - Perspective-origin: cambia las oordenadas x y del espectador . Necesita dos valores en pixeles o porcentaje, o las palabras clave center, left, rigut , top y bottom. Los valores por defecto son 50% 50%
  - Backface-visibiliy determina si la parte trasera de un elemento será visible o no
    tiene dos valores:visible o hidden (visible es el valor por defecto)

Efecto tridimensional a titulo

```
#mainbox  {
  display:  block;
   width:  500 px;
   margen:  50 px  auto;
   padding:  15 px;
 borde:  1 px  sólido  # 999999;
   background:  #FFFFFF;

  - webkit-transform:  perspectiva ( 500 px )
rotate3D ( 0 , 1 , 0 , 45 grados );
  }
```

Tridimensional y que gira en 3D

```
#mainbox  {
  display:  bloque;
   width:  500 px;
   margen:  50 px  auto;
   padding:  15 px;
   fronteriza:  1 px  sólido  # 999999;
   background:  #FFFFFF;
  - webkit-transform:  perspectiva ( 500 px ) translate3d ( 0 , 0 , - 300 px )
ROTATE3D ( 0 , 1 , 0 , 50 grados );
  -
```

Declaracion de un inicio diferente

```
body { text-align: centro;
webkit-perspectiva: 500 px;
webkit-perspectiva-origen: 50 % 90 %;}
#contenedor { display: block;width: 500 px; margen: 50 px auto;padding: 15 px;borde: 1 px sólido # 999999;
background: #FFFFFF;
-webkit-transform: ROTATE3D ( 0 , 1 , 0 , 135 grados );}
#title { fuente: negrita 36 px verdana , sans-serif;}
```

Transiciones

- Una animacion requiere una transición entre las dos etapas del proceso.
- La propiedad transition fue creada para suavizar el cambio y así s, mágicamente, rear el resto de los pasos implícitos en el movimiento. Al añadir esa propiedad forzamos al navegador a manejar la situacion, recrar estos pasoso invisible y generar una suave transicion de un estado a otro.

Sintaxis: transition: transform 1s ease-i-out 0.5s;

#contenedor:hover {

Trasnform:perespective(500px) rotate3d(0,1,0, 360deg);

La propiedad transtion puede tomar hata cuatro parámetros separados por espacios.

El primer valor es la propiedad que se usará para crear la transicion (transform en el ejemplo) .

Es necesario porque aunque varias propiedades pueden cambiar al mismo tiempo, probablement solo será necesario crear pasos para una de ellas

transiciones

- El segundo parámetro establece el tiempo en el que la transicion ejecutará , desde la posicion inicial hasta la posicon final (un segundo en nuestro ejemplo).
- El tercer parámetro puede ser alguna de estas cinco palabras clare: ease,linear, ease-in, ease-out o ease.in.out.
- Estas palabras clave determinan cómo se llegará a cabo la transicon, sobre la base de una cruva bezier.
- Cada palabra vlave representa una cruva bexier diferente y la única manera de saber cual es la mejor par aun transicion , es realizando una prueba en la pantalla

transiciones

- El último parámetro de la propiedad de trasition es el retardo. Indica cuanto tiempo se tardará la transicion para comenzar.

  Tambien es posible utilizar la palabra clave all para generar una transicion de todas las propiedades que cambian en un elemento o odeclarar varias propiedades separándolas por comas

Vamos a crear una rotacion

#contenedor { display: block;width: 500 px; margen: 50 px auto; padding: 15 px;border: 1 px sólido # 999999;

background: #FFFFFF;

webkit-transición: - webkit-transform 1 s ease-in-out 0,5 s;

}

#contendor : hover {

- webkit-transform: perspectiva (500 px) rotate3D (0 , 1 , 0 , 360deg);

}

Animaciones

- En una transicion tenemos dos pasos: el principio y el final
- Una verdadera animacion exige la declcaracion de varias tramas, como en una película y CSS3 permite hacerlo con la fución @keyframes y la propiedad animation. ´la funcion define la animacion y la propiedd proporciona los parámetros de configuracion,
- Los fotogramas son definidos por la funcion @keyframes usando valores porcentuales y agrupando los estilos para cada fotograma con llaves.

Ejemplo animacion

```
body  { text-align:  centro; }
#contenedor  { display:  bloque;width:  500 px;margin:  50 px
auto;padding:  15 px;border:  1 px  sólido  # 999999;
   animation:  mianimacion  1s  facilidad-in-out  0 s  infinita
normales  ninguno;}
keyframes  mianimacion  {
 0% { background:  #FFFFFF; }
 50% {  background:  # FF0000;}
 100% {  background:  #FFFFFF; } }
#titulo  { fuente:  negrita  36 px  verdana , sans-serif;}
```

Animaciones

- El valor 0% define el primer cuadro o el inicio de la película, mientras el 100% define el ultimo cuadro o el final. Estos valores pueden ser ignorados o también sustituidos con las palabras clave **from(0%) y to (100%).**
- La animación puede comenzar en cualquier momento y pueden declararse todos los cuadros que se quiera.
- La animación comienza en 20% y termina a 80% con un total de cinco fotogramas.
- Las propiedades de cada fotograma indican el estilo del elemento en cada paso del proceso de animación pero no como se desarrollará la animación.

Ejercicio 2 animacion

- keyframes mianimacion {

20% { background: #FFFFFF;}

35% { transformar: escala (0.5); background: # FFFF00;}

50% { transformar: escala (1.5);background: # FF0000; }

65% { transformar: escala (0.5); background: # FFFF00; }

80% { background: #FFFFFF;}

}

Animacion

- Para configurarla está la propiedad animation. Esta propiedad permite aplicar varias porpiedades disponibles para el contrlar el proceso de animacion:
- **Animation-name** proporciona el nombre que se utiliza para crear los fotogramas en la funcion @**keyframe** y conectar así de forma efextiva el fotograma con la configuracion de la animacion. Se puede utilizar para configurar varias animaciones a la vez declarando los nombres separados por comas

Animacion

- **Animation-duration**: Se utiliza para declarar cuanto tiempo durará cada ciclo de la animación. El valor debe ser especificado en segundos
- **Animation-timing-function** Funciona exactamente igua lque en la propiedad de transición (que estudiamos antes). Determina como se llevará a cabo el proceso de animación a partir de una curva Bezier declarada por las palabras clave ease, linear, ease-in, ease-out y ease-in-out
- **Animation-delay** retrasa el inicio de la animacion. Debe ser declarada en segundos , con un valor por defecto de 1

Animacion

- **Animation-iteration-count**. Declara el número de veces que se ejecuta la animacion. Necesita un número entero para el valor o la palabra clave **infinite** que ejecuta la animación indefinidamente. El valor por defecto es 1
- **Animation-direction:** Establece la dirección de la animación .

Puede tomar 4 valores: normal, reverse, alternate y alternate-reverse. Por defecto se aplica el valor normal, Que no produce ningún cambio
el segundo valor reverse, invierte la dirección de la animación, mostrando los fotogramas en la dirección opuesta en que hayan sido declaradas
el valor alternate mezcla los cilos de animacion, reporduciendo los que tienen un indice impar en dirección normal y el resto en dirección opuesta. Por ultimo el valor alternate reverse simplemente hace los mismo que alternate pero al revés

ANIMACION

- Animaton fill mode: Esta porpiedad puede tomar 4 vlores para deifnir la forma en la que la animacion va a interactua con los estilos del elemento:

None valor por defecto y no afecta a lo estilos del elemento.

El valor fowards mantiene el estilo del elemento con las propiedades aplicadas en el último fotograma de la animación (que se define con el valor 100% o las palabras clave to), mientras que backwards aplica el estilo del primer cuadro (definido por el valor o% o la palabra clave from) tan pronto como se define la animacion, es decir antes de que se ejecute

Finalmente el valor both produce ambos efectos

Como hemos visto en nuestro ejemplos, es posible declarar todas estas propiedades a la vez utilizando la propiedad animation y las siguiente sintaxis
animation: name duration timming-funmtion delay itetation-counto direction fill-mode (EL ORDEN ES IMPORTANTE)