# 2\_20-21-22-23 Filtros, Transformaciones, Transiciones y Animaciones

# FILTROS

Para añadir efectos a una imagen u otro elemento

|  |  |  |  |
| --- | --- | --- | --- |
| **Función** | **Significado** | **Valor** | **Descripción** |
| **grayscale** | Escala de blanco y negro | porcentaje | 0% = original, 100% = B&N |
| **blur** | Grado de difuminado | píxeles | Radio de desenfoque (en píxeles) |
| **sepia** | Grado de color sepia | porcentaje | 0% = original, 100% = sepia |
| **saturate** | Grado de saturación | porcentaje | 0% = B&N, 100% = original, permite >100% |
| **opacity** | Grado de transparencia | porcentaje | 0% = invisible, 100% = visible |
| **brightness** | Brillo | porcentaje | 0 = negro, 100% = original, permite >100% |
| **contrast** | Contraste | porcentaje | 0 = gris, 100% = original, permite >100% |
| **hue-rotate** | Rotación de color (matiz) | grados | 0 ó 360deg = original |
| **invert** | Invertir | porcentaje | 0% original, 100% = invertido |
| **drop-shadow** | Sombra idéntica | x y blur color | (Ver apartado de sombras) |

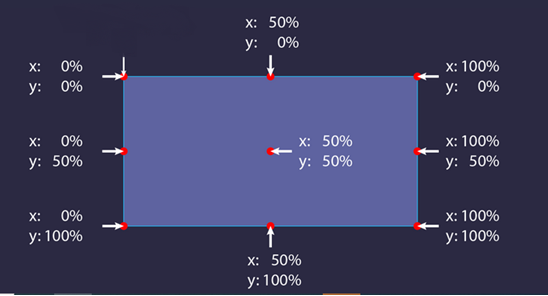
Propiedad: filter: filtro1 filtro 2 … ;

EJEMPLO:filter: contrast(175%) brightness(3%);

EJEMPLO: 1Filtros

# TRANSFORMACIONES

1. **ORIGEN DE LAS TRANSFORMACIONES:**

****

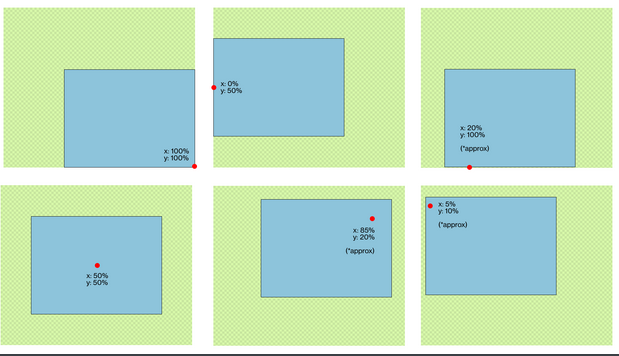
Por defecto center center (50% 50%)

Si se quiere cambiar: transform-origin: pos-x pos-y (pos-z) donde los valores pueden ser:

[x% y% (z%)]

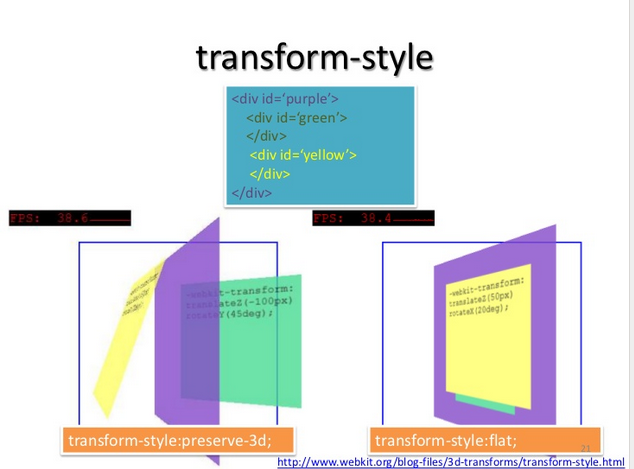
[XpxYpx (Zpx)]

[left|center|right] | [top|center|bottom]



1. **MODIFICAR TRATAMIENTO HIJOS**

transform-style: flat | preserve-3d



Efecto de transform-style

IMAGEN1

* Caja azul: detransform-style:flat
* Caja morada:-webkit-transform-style: preserve-3d;
* Amarilla y verde: Permite que las cajas hijasde la moradas (preserve-3d), amarilla y verde, vivan en un espacio 3D compartido y, por lo tanto, parezcan alejarse de la caja púrpura en virtud de sus transformaciones.

IMAGEN2:

Si ponemosen la cajamorada: transform-style: flat, sushijossecolapsandentro de él. Su -webkit-transform se aplica (se rotan, giran, deforman,…) pero ahora es solo un efecto de pintura y no se ve afectado por la perspectiva del contenedor.

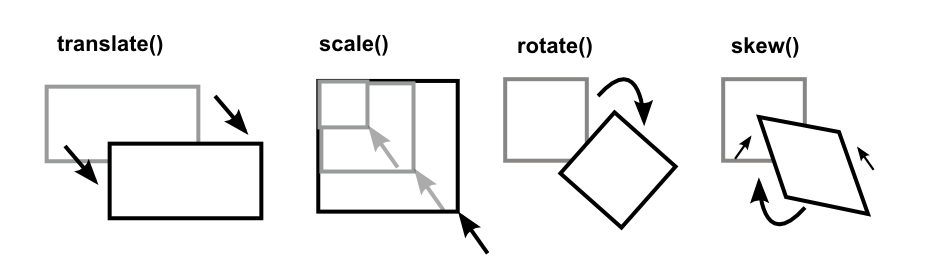
<https://codepen.io/yoksel/pen/xgzreO>

EJEMPLO: transform-style

1. **PROPIEDAD TRANSFORM**

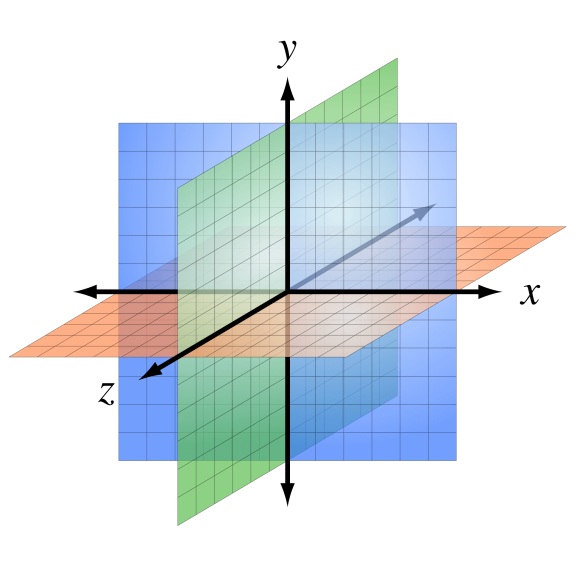
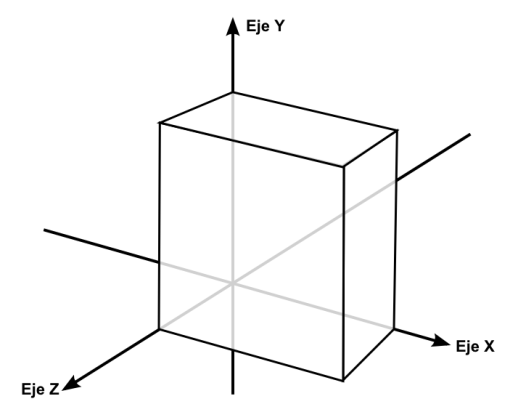
Para definir las transformaciones a aplicar.transform: funcion1 función 2 …

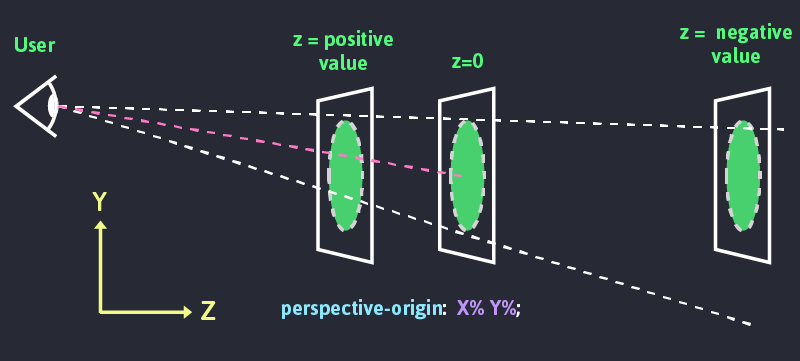
## 2D

****

|  |  |
| --- | --- |
| **Transformaciones** | **Descripción** |
| **translate(x,y)** | **Traslada el elemento una distancia de x horizontalmente y de y verticalmente.** |
| **translateX(x)** | **Traslada el elemento una distancia de x horizontalmente.** |
| **translateY(y)** | **Traslada el elemento una distancia de y verticalmente.** |
| **scale(fx,fy)** | **Reescala el elemento a un nuevo tamaño con factor fx horizontal y factor fy vertical.** |
| **scaleX(fx)** | **Reescala el elemento a un nuevo tamaño con factor fx horizontal.** |
| **scaleY(fy)** | **Reescala el elemento a un nuevo tamaño con factor fy vertical.** |
| **skew(degx,degy)** | **Establece un ángulo para una deformación 2D** |
| **skewX(degx)** | **Establece un ángulo para una deformación 2D respecto al eje X** |
| **skewY(degy)** | **Establece un ángulo para una deformación 2D respecto al eje Y** |
| **rotate(deg)** | **Establece una rotación 2D de deg grados.** |
| **rotateX(xdeg)** | **Establece una rotación 2D de xdeg grados sólo para el eje horizontal X.** |
| **rotate(ydeg)** | **Establece una rotación 2D de ydeg grados sólo para el eje vertical Y.** |

## 3D





Para activar el espacio 3D, un elemento necesita perspectiva. 2 opciones

1. Utilizar perspectiva con tranform

transform: perspective(400px) rotateY(45deg);

1. Utilizar propiedad perpective:

.scene-blue {perspective: 400px;}

<http://ds-overdesign.com/transform/matrix3d.html>

<https://makingcss.com/transform>

|  |  |
| --- | --- |
| **Transformaciones** | **Descripción** |
| **translateX(x)** | Traslada el elemento una distancia de x horizontalmente. |
| **translateY(y)** | Traslada el elemento una distancia de y verticalmente. |
| **translateZ(z)** | Traslada el elemento una distancia de z en el eje de profundidad. |
| **translate3d(x, y, z)** | Establece una translación 3D, donde aplica los parámetros a cada eje. |
| **scaleX(fx)** | Re-escala el elemento a un nuevo tamaño con factor fx horizontal. |
| **scaleY(fy)** | Re-escala el elemento a un nuevo tamaño con factor fy vertical. |
| **scaleZ(fz)** | Re-escala el elemento a un nuevo tamaño con factor fz de profundidad. |
| **scale3d(fx, fy, fz)** | Establece un escalado 3D, donde aplica los factores a cada eje. |
| **rotateX(xdeg)** | Establece una rotación 2D de xdeg grados sólo para el eje horizontal X. |
| **rotate(ydeg)** | Establece una rotación 2D de ydeg grados sólo para el eje vertical Y. |
| **rotateZ(zdeg)** | Establece una rotación 2D de zdeg grados sólo para el eje de profundidad Z. |
| **rotate3d(x, y, z, deg)** | Establece una rotación 3D, donde aplica a cada eje el número de grados deg. |
| **perspective(n)** | Establece una perspectiva 3D |
| **perspective-origin(x y)** | Establece el origen de la perspectiva |
| **transform-style** | Indica qué hacen los hijos de un elemento en 3D: flat (no conserva 3D)|preserve-3d; |

# TRANSICIONES

Variaciones progresivas

* **transition-property**: determina las propiedades a las que va afectar la transición, cuando dichas propiedades cambien de valor. (No hace falta. Aplicará lo que venga en transform)
* **transition-duration**: el tiempo empleado en la transición.
* **transition-delay**: retraso en iniciarse la transición.
* **transition-timing-function**: cómo se desarrolla la transición en el tiempo.
  + Linear: mantiene la misma velocidad de principio a fin.
  + Ease: comienzo lento, luego rápido y termina lento.
  + Ease-in: comienza lento, y después mantiene velocidad.
  + Ease-out: mantiene velocidad con un final lento.
  + Ease-in-out: comienzo y fin lentos, muy similar a ease sólo que este último empieza más rápido de lo que termina.
  + Cubic-bezier: Configurar la curva de velocidad a nuestro gusto utilizando la función cúbica de Bezier. Acepta cuatro valores entre 0 y 1.
* **transition**: atajo para especificar todos los valores (por defecto: all 0s ease 0s)

transition: propiedad duración timing retraso, propiedad2 duraciontiming retraso,…

EJEMPLO: 3Transiciones

# ANIMACIONES

* Fotogramas claves: inicio y final

Las intermedias se añadirán según cuántas tenemos o según el tiempo en el que queremos que ocurran:

* + Tantos por ciento % y la transformación a aplicar.
  + from { posición incial} to {posición final}
* Propiedades:
  + Definición:

@keyframes nombre-animacion

Fotogramas

* EJEMPLO de from to:

@keyframestraslado

{/\*Posicion inicial\*/

from {

left: 0;top: 0;}

to {/\*Posición final\*/

left: 100px; top: 100px;}

}

* EJEMPLO con %:

{ /\*Defino el número de fotogramas por %. En este caso 7 fotogramas. Y dentro de cada uno de ellos lo que va a ocurrir\*/

@keyframes gira1

{

0%{transform: rotate(0deg) }

15%{transform: rotate(20deg)}

30%{transform: rotate(40deg)}

45%{transform: rotate(60deg)}

60%{transform: rotate(80deg)}

80%{transform: rotate(100deg)}

100%{transform: rotate(120deg)} }

* + Configuración propiedades:
* **animation-name: nombre de la animación (definida en @keyframe)**
* **animation-duration:** la duración de la animación.
* **animation-iteration-count:**las veces que se repite.infinite| <numero>
* **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.
  + Linear: mantiene la misma velocidad de principio a fin.
  + Ease: comienzo lento, luego rápido y termina lento.
  + Ease-in: comienza lento, y después mantiene velocidad.
  + Ease-out: mantiene velocidad con un final lento.
  + Ease-in-out: comienzo y fin lentos, muy similar a ease sólo que este último empieza más rápido de lo que termina.
  + Cubic-bezier: Configurar la curva de velocidad a nuestro gusto utilizando la función cúbica de Bezier. Acepta cuatro valores entre 0 y 1.
* **animation-direction:** establece la dirección de la animación
  + normal: por defecto
  + reverse: el recorrido lo hace al revés
  + alternate:
    - la iteracción de indice impar--> normal
    - la iteracción de indice par --> al revés
  + reverse alternate: al revés que alternate
* **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-play-state:**paused|running.
* **animation-fill-mode**: pueden darse varios valores separados por comas**.** Cada uno de los valores se aplicará en el mismo orden en que se ha declarado en la propiedad 'animation'

none| forwards |backwards|both

* + none:La animación no aplicará los estilos antes ni después de su ejecución
  + forwards: El objeto sobre el que se aplica la animación quedará con los valores y estilos que le aplique el último keyframe de la ejecución de la animación. El último valor dependerá del valor de [animation-direction](https://developer.mozilla.org/es/docs/Web/CSS/animation-direction) y [animation-iteration-count](https://developer.mozilla.org/es/docs/Web/CSS/animation-iteration-count):

| **animation-direction** | **animation-iteration-count** | ultimo keyframe encontrado |
| --- | --- | --- |
| normal | even o odd | 100% o to |
| reverse | even o odd | 0% ofrom |
| alternate | even | 0% ofrom |
| alternate | odd | 100% oto |
| alternate-reverse | even | 100% oto |
| alternate-reverse | odd | 0% ofrom |

* + backwards:Coge los valores del primer keyframe, antes de que la animación empiece durante el animation-delay.

| **animation-direction** | primer keyframe |
| --- | --- |
| normal o alternate | 0%orfrom |
| reverse o alternate-reverse | 100%orto |

* + both: La animación seguirá las reglas de las opciones forwards y backwards, extendiendo las propiedades de la animación en ambas direcciones.

EJEMPLO:

<https://www.w3schools.com/cssref/tryit.asp?filename=trycss3_animation-fill-mode3>

3animation-fill-mode

* **animation:**Atajo:
  + [ <'animation-name'> || <'animation-duration'> || <'animation-timing-function'> || <'animation-delay'> || <'animation-iteration-count'> || <'animation-direction'> || <'animation-fill-mode'> ]
  + [, [<'animation-name'> || <'animation-duration'> || <'animation-timing-function'> || <'animation-delay'> || <'animation-iteration-count'> || <'animation-direction'> || <'animation-fill-mode'>]]