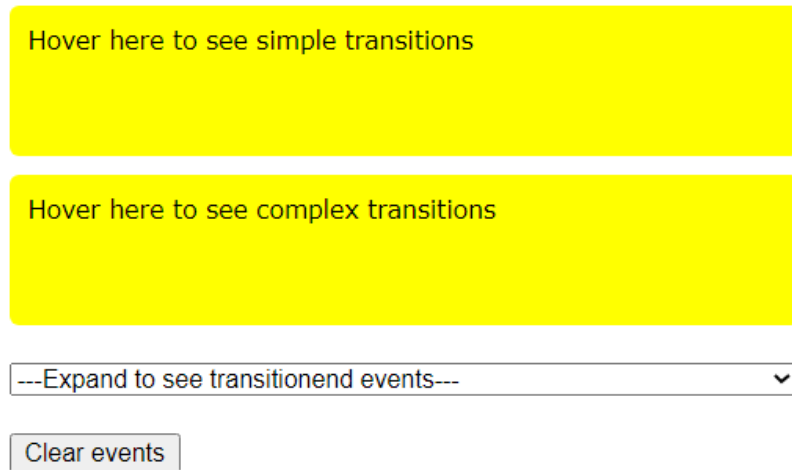


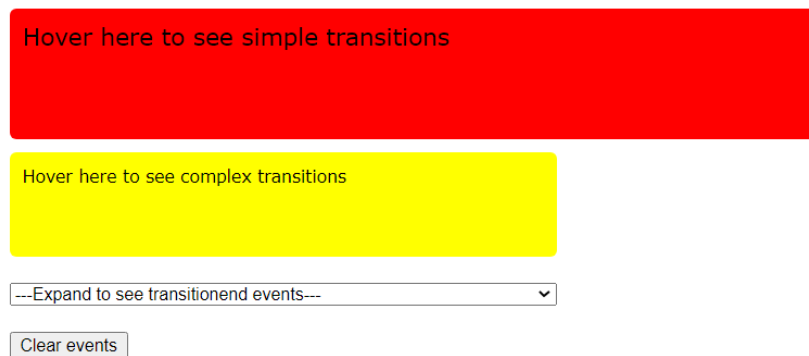
Aplicando las transiciones de CSS

Aplicando transiciones a elementos HTML

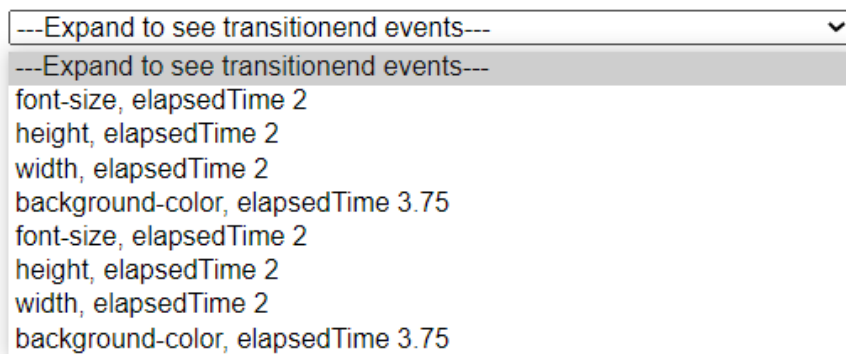
Partimos de la siguiente pantalla



Cuando pasamos el ratón sobre alguno de los dos rectángulos podemos ver la transición que se produce



Y en el combo podemos ver los eventos que se producen



Realizando transformaciones 2D

Translate x positive amount.
transform: translate(60px);

Translate x negative amount.
transform: translate(-60px);

Translate x positive percentage.
transform: translate(20%);

Translate x negative percentage.
transform: translate(-20%);

TranslateX.
transform: translateX(10px);

TranslateY.
transform: translateY(10px);

Translate x and y.
transform: translate(400px, 20px);

Cada rectángulo demuestra cómo realizar una transformación de escala 2D utilizando las funciones `scale()`, `scaleX()`, o `scaleY()`.

Realizando rotaciones 2D

Rotate 10 degrees clockwise (about center-point).
transform: rotate(10deg);

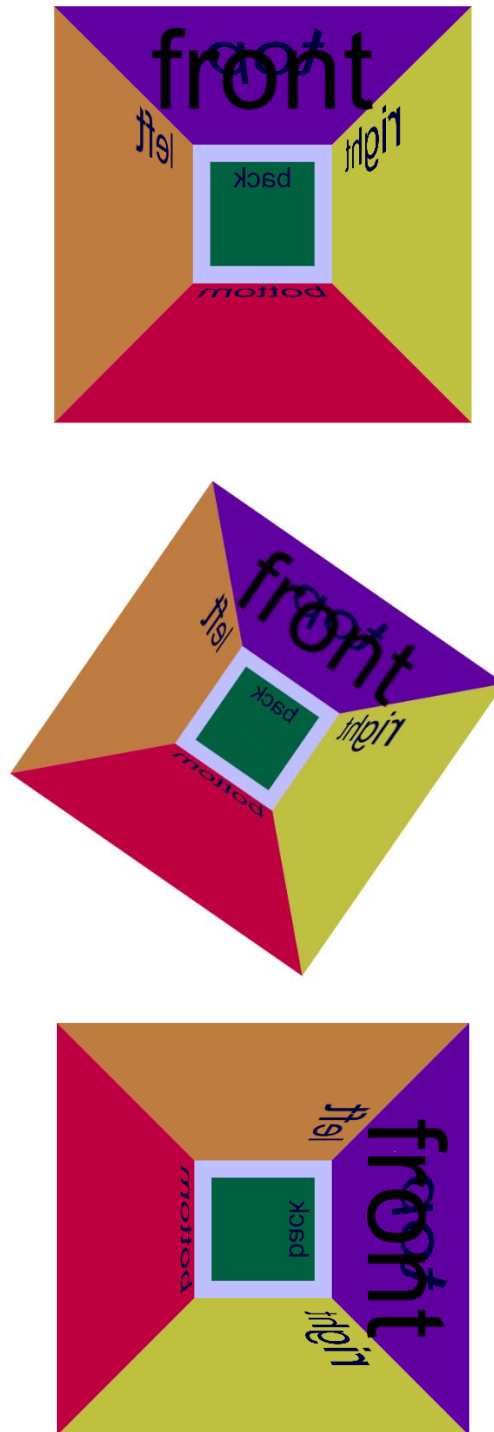
Rotate 10 degrees anticlockwise (about center-point).
transform: rotate(-10deg);

Rotate 0.5 radians anticlockwise (about center-point).
transform: rotate(-0.5rad);

Translate and then rotate (about top-left corner).
transform: translate(100px, 100px) rotate(10deg);
transform-origin: left top;

Cada rectángulo demuestra cómo realizar una operación de inclinación 2D utilizando las funciones `skew()`, `skewX()`, o `skewY()`.

Realizando transformaciones 3D



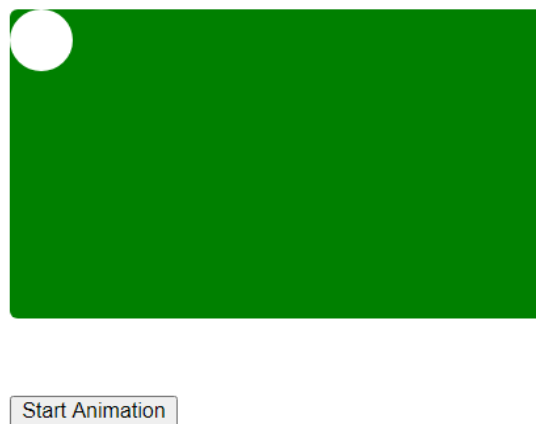
Las transformaciones vistas en las imágenes anteriores se realizan cuando se pasa el ratón por encima de la primera imagen. Una vez el ratón sale de la imagen, ésta vuelve a su posición inicial. Hay que tener en cuenta las siguientes reglas de CSS:

- #contenedor: Especifica una perspectiva para todos los elementos hijos del elemento contenedor, y una transición de cinco segundos para las transformaciones.

- #contenedor: hover: Especifica una rotación de 90 grados cuando el usuario coloca el ratón sobre el elemento contenedor.
- #rightFace: Transforma el elemento rightFace en el espacio 3D, de modo que aparezca en el lado derecho del cubo.
- #leftFace: Transforma el elemento leftFace en un espacio tridimensional, de modo que aparezca en el lado izquierdo del cubo.
- #topFace: Transforma el elemento topFace en espacio 3D, de modo que aparezca en la parte superior del cubo.
- #bottomFace: Transforma el elemento bottomFace en espacio 3D, de modo que aparezca en la parte inferior del cubo.
- #backFace: Transforma el elemento backFace en el espacio tridimensional, de modo que aparezca en la parte posterior del cubo.
- #frontFace: Transforma el elemento frontFace en el espacio 3D, de modo que aparezca en la parte delantera del cubo. El color de fondo es parcialmente transparente.

Implementación de animación de fotogramas

Partimos de la siguiente imagen

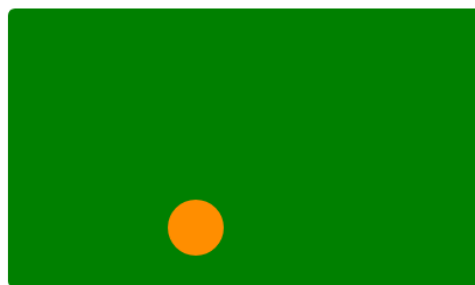


- Después de tres segundos, la bola empieza a moverse en diagonal en la mesa de billar. El color de la mesa de billar también cambia a azul y aparece un mensaje en la parte inferior de la página para indicar la hora de inicio de la animación.
- A medida que avanza la animación, la bola parece rebotar en los lados de la mesa de billar y el color de la bola varía de blanco a amarillo, luego a naranja, luego a rojo y finalmente a púrpura.
- Cuando la bola llega a la esquina inferior derecha de la mesa de billar, aparece un mensaje para indicar que la primera iteración de la animación se ha completado. Comienza la siguiente

Para obtener este efecto hay que tener en cuenta las siguientes reglas de CSS:

- #pooltable: Especifica la apariencia inicial de la mesa de billar.
- #pooltable.animate: Especifica un color diferente para la mesa de billar durante una animación. Hay un código JavaScript en otra parte del documento que agrega programáticamente la clase de animación al elemento de la mesa de billar cuando una animación comienza a hacer que la mesa de billar se ponga azul durante una animación.
- Bola: Especifica la apariencia inicial de la bola.
- @-ms-keyframes ballmovimiento: Define una animación de fotogramas clave llamada movimiento de la bola. El primer conjunto de reglas especifica el color original y la ubicación de la bola. Cada conjunto de reglas subsiguiente simula la bola golpeando uno de los lados de la mesa de billar y hace que la bola cambie de color durante cada parte de su recorrido. El último conjunto de reglas especifica el color y la ubicación final de la bola.
- #ball.animate: Aplica la animación del fotograma clave de movimiento de la bola a una bola cuando la bola tiene la clase de animación.

En la siguientes vamos viendo cómo van produciendo los efectos descritos anteriormente.



Start Animation

