



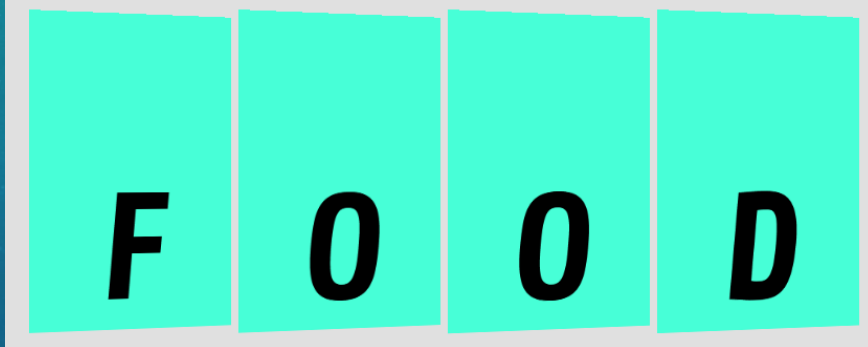
ANIMACIÓN CSS

ALICIA ACOSTA TERNERO

HTML

En la estructura de html contamos con un div principal que es el container, donde se recoge toda la animación.

Dentro del container encontramos cuatro div hijos, con la clase panel. Estos paneles serán cada uno de los rectángulos.



```
<body>  
  <div class="container">  
    <div class="panel">...  
  </div>  
  <div class="panel">...  
  </div>  
  <div class="panel">...  
  </div>  
  <div class="panel">...  
  </div>  
</body>  
</html>
```

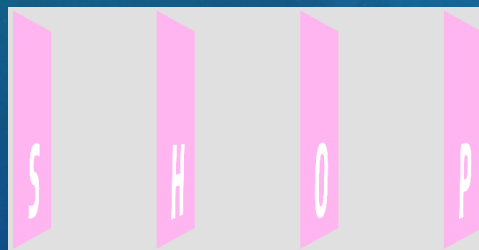
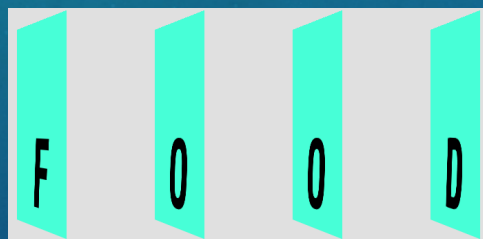
```
<body>  
  <div class="container">  
    <div class="panel">  
      <div class="panel-inner">  
        <div class="panel-front">  
          <p>F</p>  
        </div>  
        <div class="panel-back">  
          <p>S</p>  
        </div>  
      </div>  
    </div>  
    <div class="panel">  
      <div class="panel-inner">  
        <div class="panel-front">  
          <p>O</p>  
        </div>  
        <div class="panel-back">  
          <p>H</p>  
        </div>  
      </div>  
    </div>  
    <div class="panel">  
      <div class="panel-inner">  
        <div class="panel-front">  
          <p>O</p>  
        </div>  
        <div class="panel-back">  
          <p>O</p>  
        </div>  
      </div>  
    </div>  
    <div class="panel">  
      <div class="panel-inner">  
        <div class="panel-front">  
          <p>D</p>  
        </div>  
        <div class="panel-back">  
          <p>P</p>  
        </div>  
      </div>  
    </div>  
  </div>  
</body>
```


HTML

En cada uno de los paneles encontramos otro div “padre”, con la clase compartida de panel-inner

Dentro de cada ‘panel-inner’ encontramos dos div, uno con la clase ‘panel-front’ y otro con la clase ‘panel-back’. Cada uno contiene a su vez un párrafo (<p>), que contiene una letra.

Este ‘panel-front’ y ‘panel-back’ hace referencia a las dos caras del rectángulo, de manera que los párrafos del ‘panel-front’ formarán una palabra y los del ‘panel-back’ otra distinta, para que cuando rote el rectángulo se pueda ver ambas caras.



```
<div class="panel">
  <div class="panel-inner">
    <div class="panel-front">
      <p>F</p>
    </div>
    <div class="panel-back">
      <p>S</p>
    </div>
  </div>
</div>
```

```
<body>
  <div class="container">
    <div class="panel">
      <div class="panel-inner">
        <div class="panel-front">
          <p>F</p>
        </div>
        <div class="panel-back">
          <p>S</p>
        </div>
      </div>
    </div>
    <div class="panel">
      <div class="panel-inner">
        <div class="panel-front">
          <p>O</p>
        </div>
        <div class="panel-back">
          <p>H</p>
        </div>
      </div>
    </div>
    <div class="panel">
      <div class="panel-inner">
        <div class="panel-front">
          <p>O</p>
        </div>
        <div class="panel-back">
          <p>O</p>
        </div>
      </div>
    </div>
    <div class="panel">
      <div class="panel-inner">
        <div class="panel-front">
          <p>D</p>
        </div>
        <div class="panel-back">
          <p>P</p>
        </div>
      </div>
    </div>
  </div>
</body>
```

CSS

En la etiqueta html se establece el color del fondo, el tamaño de la letra, su fuente y que el ancho sea el total de la página.

En la clase container le damos un margen arriba, y alineamos las letras en dos frases horizontalmente, centrado y en línea.

FOOD

SHOP

F

S

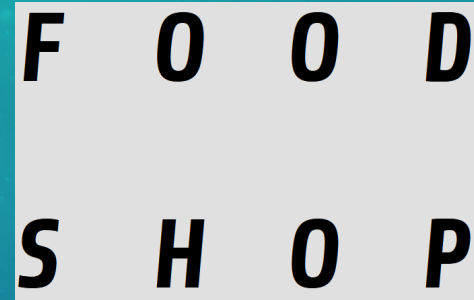
O

```
html {  
  box-sizing: border-box;  
  font-family: "Share", sans-serif;  
  font-style: italic;  
  font-size: 9em;  
  background-color: #e0e0e0;  
}  
.container {  
  /* margin: auto; */  
  /* width: 1000px; */  
  display: flex;  
  justify-content: center;  
  flex-wrap: nowrap;  
  margin-top: 15vh;  
}
```


CSS

En la clase panel asignamos a cada “rectángulo” una altura y anchura, así como la separación en el plano z, y que su color de fondo sea transparente, de forma que se verá el color de fondo del html cuando se rote en la animación.

Para la clase ‘panel-inner’ padre de las clases hijas ‘panel-front’ y ‘panel-back’, le daremos una posición relativa y un tamaño del 100% y lo alinearemos todo al centro. También se le dará la propiedad ‘transform-style’ y asignándole a las hijas una posición en el espacio el 3d, y la propiedad ‘animation’, dónde se le dirá que la animación flip gire sobre sí mismo con una duración de 4 segundos, de forma línea e infinitamente. Esta clase será la encargada de asignar la animación.



F O O D
S H O P

```
.panel {  
  background-color: transparent;  
  width: 200px;  
  height: 300px;  
  perspective: 1000px;  
}  
  
.panel-inner {  
  position: relative;  
  width: 100%;  
  height: 100%;  
  text-align: center;  
  /* transition: transform 0.75s; */  
  transform-style: preserve-3d;  
  animation: flip 4s linear forwards infinite;  
  animation-direction: normal;  
}  
  
.panel-front,  
.panel-back {  
  position: absolute;  
  width: 100%;  
  height: 100%;  
  -webkit-backface-visibility: hidden;  
  backface-visibility: hidden;  
}
```

CSS

Las propiedades que se le dan a las clases 'panel-front' y 'panel-back' serán el tamaño que ocuparán dentro de sus respectivos div, que será del 100%. También se le da una posición absoluta haciendo que puedan superponerse una tarjeta sobre otra. Y por último la propiedad 'backface-visibility' que determina si la cara posterior está visible o escondida, en este caso se determina como escondida, de forma que ninguna de las dos caras de las tarjetas se encuentren durante la rotación.

Se repite las clases 'panel-front' y 'panel-back', esta vez para determinar el color de fondo que tendrá cada una, y el color de la letra. En el 'panel-back' además se le añade la propiedad 'transform: rotateY' que definirá la rotación, su dirección, su eje (horizontal) y establecerá cual cara será la primera en aparecer ('panel-back') y en el grado de la rotación donde se hará el cambio de una tarjeta a otra, que en este caso es a 180 grados.



```
.panel-front,
.panel-back {
  position: absolute;
  width: 100%;
  height: 100%;
  -webkit-backface-visibility: hidden;
  backface-visibility: hidden;
}

.panel-front {
  background-color: #4682b4;
  color: black;
}

.panel-back {
  background-color: #800080;
  color: white;
  transform: rotateY(180deg);
  /* padding: 5px; */
}
```


CSS

Por último tenemos el `@keyframes`, que nos permite controlar los pasos de la rotación, estableciendo keyframes a lo largo de la animación.

En este caso lo usamos en animación flip donde le decimos que deberá rotar en su eje horizontal Y, de 0 grados a 360, es decir, un giro completo.

Por lo que con la animación flip determinada en la clase 'panel-inner', le decimos a todas las tarjetas que rotarán en una duración de 4 segundos, de forma infinita, hacia la derecha en el eje horizontal, escondiéndose una de las caras a cada 180 grado y rotando entre sí un giro completo de 360 grados.

```
.panel-inner {
  position: relative;
  width: 100%;
  height: 100%;
  text-align: center;
  /* transition: transform 0.75s; */
  transform-style: preserve-3d;
  animation: flip 4s linear forwards infinite;
  animation-direction: normal;
}

.panel-front,
.panel-back {
  position: absolute;
  width: 100%;
  height: 100%;
  -webkit-backface-visibility: hidden;
  backface-visibility: hidden;
}

.panel-front {
  background-color: #4682b4;
  color: black;
}

.panel-back {
  background-color: #800080;
  color: white;
  transform: rotateY(180deg);
  /* padding: 5px; */
}

.panel-front img {
  width: 300px;
  height: 300px;
}

@keyframes flip {
  from {
    transform: rotateY(0deg);
  }
  to {
    transform: rotateY(360deg);
  }
}
```