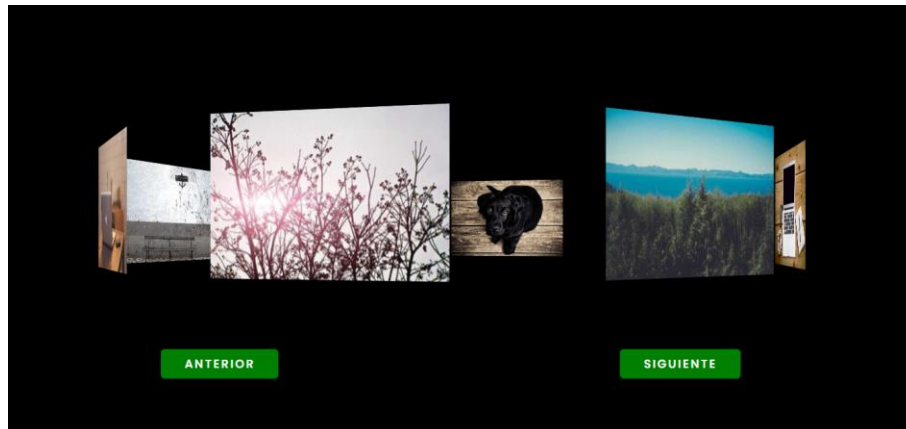


## Galería 3D Rotativa de Imágenes



Aprendices a continuación realizaremos una galería de imágenes en 3D que rotará y nos permitirá utilizar varias propiedades de nuestro JS. Vamos con el código HTML:

```
12 <body>
13
14 <!-- las imagenes las tomaremos de picsum -->
15
16 <div class="image-container">
17   <span style="--i: 1">
18     
19   </span>
20   <span style="--i: 2">
21     
22   </span>
23   <span style="--i: 3">
24     
25   </span>
26   <span style="--i: 4">
27     
28   </span>
29   <span style="--i: 5">
30     
31   </span>
32   <span style="--i: 6">
33     
34   </span>
35   <span style="--i: 7">
36     
37   </span>
38   <span style="--i: 8">
39     
40   </span>
41 </div>
42 <div class="buttons">
43   <button class="btn prev">Anterior</button>
44   <button class="btn next">Siguiente</button>
45 </div>
46
47 <script src="index.js"></script>
48 </body>
49
50 </html>
```



@hdtoledo

1. `<div class="image-container">...</div>` Esto crea un contenedor `<div>` con la clase "image-container". Se utiliza para agrupar y contener las imágenes dentro de él.
2. `<span style="--i: 1">...</span>` Este elemento `<span>` se utiliza como un contenedor para una imagen. El estilo en línea `style="--i: 1"` establece una variable CSS personalizada `--i` con el valor 1. Esto puede ser utilizado posteriormente para aplicar estilos a cada imagen de manera individual.
3. `` Este elemento `<img>` representa una imagen con la atributo `src` que contiene la URL de la imagen. En este caso, se utiliza el servicio "Lorem Picsum" para obtener imágenes de demostración. La URL específica se compone del dominio "<https://picsum.photos/>" seguido del identificador de la imagen ("id") y las dimensiones de la imagen en píxeles (300 de ancho y 200 de alto).

Las líneas 2 a 8 siguen el mismo patrón que se explicó anteriormente, pero con diferentes valores en las variables `--i` y en los identificadores de las imágenes.

En resumen, este código HTML crea un contenedor de imágenes que utiliza el servicio "Lorem Picsum" para cargar y mostrar imágenes de demostración. Cada imagen está contenida dentro de un elemento `<span>` con un estilo en línea que establece una variable CSS personalizada `--i` con un valor numérico. Esto permite aplicar estilos a cada imagen de manera individual si se utiliza CSS adicional.

Vamos con nuestro CSS:

```
1  @import url("https://fonts.googleapis.com/css2?family=Poppins:wght@100;200;300;400;500;600;700;800;900&display=swap");
2
3  * {
4    margin: 0;
5    padding: 0;
6    box-sizing: border-box;
7    font-family: "Poppins", sans-serif;
8  }
9
10 body {
11   display: flex;
12   flex-direction: column;
13   align-items: center;
14   justify-content: center;
15   height: 100vh;
16   background-color: black;
17   overflow: hidden;
18 }
19
20 .image-container {
21   position: relative;
22   width: 200px;
23   min-height: 200px;
24   transform-style: preserve-3d;
25   transform: perspective(1000px) rotateY(0deg);
26   transition: all 1s ease-in-out;
27 }
28
29 .image-container span {
30   position: absolute;
31   top: 0;
32   left: 0;
33   width: 100%;
34   transform: rotateY(calc(var(--i) * 45deg)) translateZ(400px);
35 }
```



@hdtoledo

```

37 .image-container span img {
38   position: absolute;
39   left: 0;
40   top: 0;
41   width: 100%;
42 }
43
44 .buttons {
45   position: relative;
46   width: 80%;
47 }
48
49 .btn {
50   position: absolute;
51   bottom: -80px;
52   background-color: green;
53   border: none;
54   color: white;
55   padding: 10px 30px;
56   font-size: 15px;
57   font-weight: 600;
58   letter-spacing: 2px;
59   text-transform: uppercase;
60   border-radius: 5px;
61   cursor: pointer;
62 }
63
64 .btn:hover {
65   filter: brightness(1.5);
66 }
67
68 .btn:active {
69   transform: scale(0.95);
70 }
71
72 .prev {
73   left: 20%;
74 }
75
76 .next {
77   right: 20%;
78 }
79

```

### 1. Importación de la fuente de Google Fonts:

- Se importa la fuente "Poppins" con diferentes pesos y estilos para su uso en el documento.

### 2. Estilos generales para todos los elementos:

- Se establece un margen y relleno de 0 para todos los elementos (**margin: 0; padding: 0;**).
- Se utiliza la propiedad **box-sizing** con el valor **border-box** para que el tamaño total de un elemento incluya el relleno y el borde.
- Se aplica la fuente "Poppins" como la fuente predeterminada para todos los elementos del documento.

### 3. Estilos para el cuerpo (**body**):

- Se establece el cuerpo como un contenedor flexible (**display: flex**) con dirección de columna (**flex-direction: column**), lo que permite alinear verticalmente los elementos hijos.
- Los elementos se alinean en el centro tanto horizontal como verticalmente (**align-items: center; justify-content: center;**).
- La altura del cuerpo se establece en el 100% de la altura de la ventana gráfica (**height: 100vh**).



@hdtoledo

- El color de fondo se establece en negro (**background-color: black**).
  - El desbordamiento (overflow) del cuerpo se oculta, lo que significa que no se mostrarán barras de desplazamiento si el contenido es más grande que el tamaño del cuerpo (**overflow: hidden**).
4. Estilos para el contenedor de imágenes (**.image-container**):
- Se establece la posición del contenedor como relativa (**position: relative**).
  - Se establece un ancho fijo de 200 píxeles (**width: 200px**).
  - Se establece una altura mínima de 200 píxeles (**min-height: 200px**).
  - Se utiliza el estilo **preserve-3d** para mantener la representación en 3D de los elementos secundarios.
  - Se establece una transformación inicial que aplica una perspectiva de 1000 píxeles (**transform: perspective(1000px)**) y una rotación en el eje Y de 0 grados (**rotateY(0deg)**).
  - Se aplica una transición suave a todas las propiedades del contenedor con una duración de 1 segundo (**transition: all 1s ease-in-out**).
5. Estilos para los elementos **span** dentro del contenedor de imágenes (**.image-container span**):
- Se establece la posición de los elementos **span** como absoluta (**position: absolute**).
  - Los elementos se posicionan en la esquina superior izquierda del contenedor (**top: 0; left: 0**).
  - Se establece un ancho del 100% para ocupar todo el ancho disponible (**width: 100%**).
  - Se aplica una transformación que rota los elementos **span** en el eje Y, con un ángulo calculado en base a una variable `--i` multiplicada por 45 grados (**transform: rotateY(calc(var(--i) \* 45deg))**).
  - También se aplica una traslación en el eje Z para alejar los elementos del espectador (**translateZ(400px)**).
6. Estilos para las imágenes dentro de los elementos **span** (**.image-container span img**):
- Se establece la posición de las imágenes como absoluta (**position: absolute**).
  - Las imágenes se ajustan a la esquina superior izquierda de los elementos **span** (**left: 0; top: 0**).
  - Se establece un ancho del 100% para que las imágenes ocupen todo el ancho disponible.
7. Estilos para el contenedor de botones (**.buttons**):
- Se establece la posición del contenedor como relativa (**position: relative**).
  - Se establece un ancho del 80% para que el contenedor ocupe el 80% del ancho disponible.



@hdtoledo

8. Estilos para los botones (**.btn**):

- Se establece la posición de los botones como absoluta (**position: absolute**).
- Los botones se sitúan fuera del contenedor en la parte inferior, dejando un espacio de -80 píxeles (**bottom: -80px**).
- Se establece un fondo verde para los botones (**background-color: green**).
- Se eliminan los bordes (**border: none**).
- El texto de los botones se muestra en color blanco (**color: white**).
- Se aplica un relleno de 10 píxeles vertical y 30 píxeles horizontal (**padding: 10px 30px**).
- Se establece un tamaño de fuente de 15 píxeles (**font-size: 15px**).
- Se establece un peso de fuente de 600 (**font-weight: 600**).
- Se aplica un espaciado de letras de 2 píxeles (**letter-spacing: 2px**).
- El texto se muestra en mayúsculas (**text-transform: uppercase**).
- Se establece un radio de borde de 5 píxeles (**border-radius: 5px**).
- Se cambia el cursor al puntero al pasar el ratón sobre los botones (**cursor: pointer**).

9. Estilos para el estado de desplazamiento (hover) de los botones (**.btn:hover**):

- Cuando se pasa el ratón sobre los botones, se aplica un filtro de brillo aumentado (**filter: brightness(1.5)**), lo que hace que los botones sean más brillantes.

10. Estilos para el estado activo de los botones (**.btn:active**):

- Cuando se activa un botón (por ejemplo, al hacer clic), se aplica una transformación de escala reducida (**transform: scale(0.95)**), lo que hace que el botón se reduzca ligeramente de tamaño.

11. Estilos para el botón "anterior" (**.prev**) y el botón "siguiente" (**.next**):

- Se establece la posición de los botones "anterior" y "siguiente" como absoluta y se colocan a la izquierda y a la derecha del 20% del contenedor, respectivamente (**left: 20%** y **right: 20%**).

Estos estilos combinados crean una galería de imágenes con efecto de rotación en 3D y botones de navegación que se deslizan desde la parte inferior cuando se activan.

Ahora vamos con nuestro JS:



@hdtoledo

```

1  const imageContainerEl = document.querySelector(".image-container");
2  const prevEl = document.querySelector(".prev");
3  const nextEl = document.querySelector(".next");
4
5  let x=0;
6  let timer=0;
7
8  prevEl.addEventListener("click", ()=>{
9      x = x + 45;
10     clearTimeout(timer);
11     updateContainer();
12 })
13
14 nextEl.addEventListener("click", ()=>{
15     x = x - 45;
16     clearTimeout(timer);
17     updateContainer();
18 })
19
20 function updateContainer(){
21     imageContainerEl.style.transform = `perspective(1000px) rotateY(${x}deg)`;
22
23     timer = setTimeout(() => {
24         x = x - 45;
25         updateContainer();
26     }, 2000);
27 }
28
29 updateContainer();

```

### 1. Seleccionar elementos del DOM:

- **imageContainerEl**: Selecciona el elemento del DOM con la clase "image-container", que representa el contenedor de imágenes.
- **prevEl**: Selecciona el elemento del DOM con la clase "prev", que representa el botón "anterior".
- **nextEl**: Selecciona el elemento del DOM con la clase "next", que representa el botón "siguiente".

### 2. Variables:

- **x**: Almacena el valor de la rotación en grados del contenedor de imágenes.
- **timer**: Almacena el identificador del temporizador para la animación automática de rotación.

### 3. Event listeners para los botones:

- **prevEl.addEventListener("click", ()=> { ... })**: Agrega un event listener al botón "anterior" que se activa cuando se hace clic en él. En la función de devolución de llamada, se incrementa el valor de **x** en 45 grados, se cancela cualquier temporizador activo y se llama a la función **updateContainer()** para actualizar la transformación del contenedor de imágenes.
- **nextEl.addEventListener("click", ()=> { ... })**: Agrega un event listener al botón "siguiente" que se activa cuando se hace clic en él. En la función de devolución de llamada, se decrementa el valor de **x** en 45 grados, se cancela cualquier temporizador activo y se llama a la función **updateContainer()** para actualizar la transformación del contenedor de imágenes.



@hdtoledo

4. Función **updateContainer()**:

- Actualiza la transformación del contenedor de imágenes (**imageContainerEl.style.transform**) aplicando el valor de rotación actual almacenado en la variable **x**.
- Establece un temporizador utilizando **setTimeout()** para realizar la rotación automática cada 2 segundos. Dentro de la función de temporizador, se decrementa el valor de **x** en 45 grados y se llama recursivamente a **updateContainer()** para actualizar la transformación del contenedor de imágenes nuevamente después de 2 segundos.

5. Llamada inicial a **updateContainer()**:

- Al final del código, se llama a **updateContainer()** para iniciar la rotación automática del contenedor de imágenes.

En resumen, este código agrega interactividad a la galería de imágenes al permitir al usuario navegar hacia atrás y hacia adelante mediante los botones "anterior" y "siguiente". También incluye una animación automática que rota el contenedor de imágenes cada 2 segundos.



@hdtolledo