

Exercicio 71-B: Crear un slider de imaxes con JavaScript

Obxectivos: Aprender a crear un slider de imaxes con JavaScript. Comprender o funcionamento básico dos sliders. Practicar os conceptos básicos de JavaScript.

Pasos a seguir:

1. Crear o HTML

O primeiro paso é crear o HTML que conterá o *slider*. Para iso, crearemos un elemento *div* coa clase *slider*. Dentro deste elemento, engadiremos as imaxes que queremos amosar dentro doutro *div* coa clase ***slider__contenedor***.

HTML:

```
<div class="slider">
  <div class="slider__contenedor">
    
    
    
  </div>
  <button id="prev" onclick="prevSlide()">&#10094;</button>
  <button id="siguiente" onclick="nextSlide()">&#10095;</button>
</div>
```

2. Engadir os estilos CSS

A continuación, engadiremos os estilos CSS necesarios para que o *slider* teña un aspecto axeitado. Para iso, definiremos as seguintes propiedades:

- A anchura e a altura do slider.
- A aliñación das imaxes.
- As marxes e os recheos das imaxes.

CSS:

```
.slider {
  width: 100%;
  height: 100%;
  position: relative;
```

```

    overflow: hidden;
}
.slider__contenedor {
    display: flex;
    transition: transform 0.5s ease-in-out;
}
.slider__slide {
    min-width: 100vw;
    box-sizing: border-box;
}

```

3. Engadir a funcionalidade JavaScript

O último paso é engadir a funcionalidade JavaScript que fará que o slider funcione. Para iso, crearemos unha función que se encargue de cambiar a imaxe que se amosa no slider.

- ***let currentIndex = 0;*** : Declárase unha variable *currentIndex* e inicialízase en 0. Esta variable úsase para facer un seguimento do índice actual do slide que se está amosando.
- ***const slides = document.querySelectorAll('.slider__slide');*** Selecciónanse todos os elementos HTML coa clase CSS *.slider__slide* e almacénanse na variable *slides*. Isto supón que cada un destes elementos representa un *slide* no carrusel.
- ***const totalSlides = slides.length;*** Almacénase a cantidade total de *slides* na variable *totalSlides* utilizando a propiedade *length* da colección de nodos obtida con *querySelectorAll*.
- ***const sliderContainer = document.querySelector('.slider__contenedor');*** Selecciónase o contedor do slider e almacénase na variable *sliderContainer*. Este contedor probablemente sexa o elemento que se moverá para cambiar os slides.
- ***function mostrarSlide(index) { ... }*** Defínese unha función chamada ***mostrarSlide*** que toma un parámetro *index*. Esta función encárgase de amosar o *slide* correspondente ao índice pasado como argumento.
 - Dentro da función, verifícase se o índice é menor que 0. Nese caso, establécese *currentIndex* ao último *slide*.
 - Se o índice é maior ou igual ao total de *slides*, establécese *currentIndex* ao primeiro *slide*.

- Se o índice está dentro do rango válido, establécese **currentIndex** ao valor proporcionado.
- Cálculase o valor de traslación (*translateValue*) multiplicando - *currentIndex* polo ancho de cada *slide* (asumindo que cada *slide* ocupa o 100% do ancho do contedor).
- Actualízase a propiedade *transform* do contedor do *slider* para amosar o *slide* correspondente.
- **function seguinteSlide() { ... }:** Defínese unha función chamada seguinteSlide que avanza ao seguinte slide chamando á función mostrarSlide co índice do seguinte slide.
- **function previaSlide() { ... }:** Defínese unha función chamada previaSlide que retrocede ao slide anterior chamando á función mostrarSlide co índice do slide anterior.
- **setInterval(seguinteSlide, 3000);:** Establécese un intervalo que chama á función seguinteSlide cada 3000 milisegundos (3 segundos), o que crea un efecto de reprodución automática do slider.

O resultado debería ser este:

```
let currentIndex = 0;

const slides = document.querySelectorAll('.slider__slide');
const totalSlides = slides.Length;
const sliderContainer = document.querySelector('.slider__contenedor');

function mostrarSlide(index) {
  if (index < 0) {
    currentIndex = totalSlides - 1;
  } else if (index >= totalSlides) {
    currentIndex = 0;
  } else {
    currentIndex = index;
  }

  const translateValue = -currentIndex * 100 + '%';
  sliderContainer.style.transform = 'translateX(' + translateValue + ')';
}

function siguienteSlide() {
```

```
    mostrarSlide(currentIndex + 1);
}

function previaSlide() {
    mostrarSlide(currentIndex - 1);
}

// Autoplay
setInterval(siguienteSlide, 3000);
```

4. Probamos o slider

Agora, podemos probar o slider nun navegador web. Se todo foi ben, o slider debería funcionar correctamente. Ao facer clic no botón Seguinte, a imaxe que se amosa no slider debería cambiar.

Melloras (opcional)

O *slider* que creamos é bastante básico. Podemos melloralo engadindo as seguintes funcionalidades:

- Controis a partir de indicadores (círculos por exemplo) para indicar a posición.
- Opción para deter a reprodución automática.
- Opción para cambiar a velocidade de reprodución automática.

Tamén podemos engadir máis funcionalidades, como a posibilidade de **amosar texto ou vídeos** en vez de imaxes ou ademais de imaxes.

Entrega: Unha vez completado todo **subiremos a carpeta *Exercicio71-B*** ao espazo para entrega de exercicios que temos no OneDrive de Microsoft Teams.

Ejercicio 71-B: Crear un slider de imágenes con JavaScript

Objetivos: Aprender a crear un *slider* de imágenes con JavaScript. Comprender el funcionamiento básico de los *sliders*. Practicar los conceptos básicos de JavaScript.

Pasos a seguir:

1. Crear el HTML

El primer paso es crear el HTML que contendrá el *slider*. Para ello, crearemos un elemento *div* con la clase *slider*. Dentro de este elemento, añadiremos las imágenes que queremos mostrar dentro de otro *div* con la clase *slider__contenedor*.

HTML:

```
<div class="slider">
  <div class="slider__contenedor">
    
    
    
  </div>
  <button id="prev" onclick="prevSlide()">&#10094;</button>
  <button id="siguiente" onclick="nextSlide()">&#10095;</button>
</div>
```

2. Añadir los estilos CSS

A continuación, añadiremos los estilos CSS necesarios para que el *slider* tenga un aspecto adecuado. Para ello, definiremos las siguientes propiedades:

- El ancho y la altura del *slider*.
- La alineación de las imágenes.
- Los márgenes y los rellenos de las imágenes.

CSS:

```
.slider {
  width: 100%;
  height: 100%;
  position: relative;
  overflow: hidden;
}
```

```

.slider__contenedor {
  display: flex;

  transition: transform 0.5s ease-in-out;
}

.slider__slide {
  min-width: 100vw;

  box-sizing: border-box;
}

```

3. Añadir la funcionalidad JavaScript

El último paso es añadir la funcionalidad JavaScript que hará que el slider funcione. Para ello, crearemos una función que se encargue de cambiar la imagen que se muestra en el slider.

- ***let currentIndex = 0;*** Se declara una variable *currentIndex* y se inicializa en 0. Esta variable se usa para hacer un seguimiento del índice actual del *slide* que se está mostrando.
- ***const slides = document.querySelectorAll('.slider__slide');*** Se seleccionan todos los elementos HTML con la clase CSS *.slider__slide* y se almacenan en la variable *slides*. Esto supone que cada uno de estos elementos representa un *slide* en el carrusel.
- ***const totalSlides = slides.length;*** Se almacena la cantidad total de *slides* en la variable *totalSlides* utilizando la propiedad *length* de la colección de nodos obtenida con *querySelectorAll*.
- ***const sliderContainer = document.querySelector('.slider__contenedor');*** Se selecciona el contenedor del *slider* y se almacena en la variable *sliderContainer*. Este contenedor probablemente sea el elemento que se moverá para cambiar los slides.
- ***function mostrarSlide(index) { ... };*** Se define una función llamada *mostrarSlide* que toma un parámetro *index*. Esta función se encarga de mostrar el slide correspondiente al índice pasado como argumento.
 - Dentro de la función, se verifica si el índice es menor que 0. En ese caso, se establece *currentIndex* al último *slide*.
 - Si el índice es mayor o igual al total de *slides*, se establece *currentIndex* al primer *slide*.
 - Si el índice está dentro del rango válido, se establece *currentIndex* al valor proporcionado.

- Se calcula el valor de traslación (*translateValue*) multiplicando *-currentIndex* por el ancho de cada *slide* (asumiendo que cada slide ocupa el 100% del ancho del contenedor).
- Se actualiza la propiedad *transform* del contenedor del slider para mostrar el *slide* correspondiente.

• **function siguienteSlide() { ... }:** Se define una función llamada siguienteSlide que avanza al siguiente slide llamando a la función mostrarSlide con el índice del siguiente slide.

• **function previaSlide() { ... }:** Se define una función llamada previaSlide que retrocede al slide anterior llamando a la función mostrarSlide con el índice del slide anterior.

• **setInterval(siguienteSlide, 3000);:** Se establece un intervalo que llama a la función siguienteSlide cada 3000 milisegundos (3 segundos), lo que crea un efecto de reproducción automática del slider.

El resultado debería ser este:

```
let currentIndex = 0;

const slides = document.querySelectorAll('.slider__slide');
const totalSlides = slides.Length;
const sliderContainer = document.querySelector('.slider__contenedor');

function mostrarSlide(index) {
  if (index < 0) {
    currentIndex = totalSlides - 1;
  } else if (index >= totalSlides) {
    currentIndex = 0;
  } else {
    currentIndex = index;
  }

  const translateValue = -currentIndex * 100 + '%';
  sliderContainer.style.transform = 'translateX(' + translateValue + ')';
}

function siguienteSlide() {
  mostrarSlide(currentIndex + 1);
}
```

```
function previaSlide() {  
    mostrarSlide(currentIndex - 1);  
}  
  
// Autoplay  
setInterval(siguienteSlide, 3000);
```

4. Probamos el slider

Ahora, podemos probar el *slider* en un navegador web. Si todo ha ido bien, el *slider* debería funcionar correctamente. Al hacer clic en el botón Siguiente, la imagen que se muestra en el *slider* debería cambiar.

Mejoras (opcional)

El slider que hemos creado es bastante básico. Podemos mejorarlo añadiendo las siguientes funcionalidades:

- Controles a partir de indicadores (círculos por ejemplo) para indicar la posición.
- Opción para detener la reproducción automática.
- Opción para cambiar la velocidad de reproducción automática.

También podemos añadir más funcionalidades, como la posibilidad de **mostrar texto o vídeos en vez de imágenes** o además de imágenes.

Entrega:

Una vez completado todo subiremos la carpeta **Ejercicio71-B** al espacio para entrega de ejercicios que tenemos en el OneDrive de Microsoft Teams.