

Aula 9

# Imagens dinâmicas

Unidade

**Página web: utilizando a responsividade  
em aplicações com HTML e CSS**

**Parte 1**

# O que vamos aprender?



Consultar a documentação API.



Inicializar o Swiper.



Definir parâmetros do Swiper.

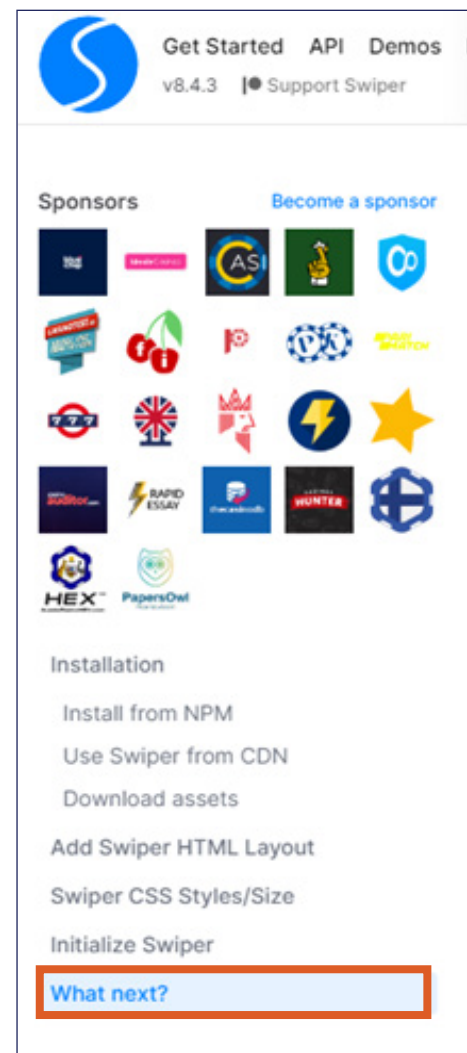
Let's look on list of all available parameters:

Parameter	Type	Default	Description
<code>autoplay</code>	boolean	true	Use to enable/disable autoplay. To disable autoplay, set the value to false.
<code>autoplayDelay</code>	boolean	true	Use to enable/disable autoplay delay. To disable autoplay delay, set the value to false.
<code>autoplaySpeed</code>	boolean	true	Use to enable/disable autoplay speed. To disable autoplay speed, set the value to false.
<code>autoplayTimeout</code>	boolean	true	Use to enable/disable autoplay timeout. To disable autoplay timeout, set the value to false.
<code>autoplayType</code>	boolean	true	Use to enable/disable autoplay type. To disable autoplay type, set the value to false.
<code>autoplayUpdateOnImagesLoaded</code>	boolean	true	Use to enable/disable autoplay update on images loaded. To disable autoplay update on images loaded, set the value to false.
<code>autoplayUpdateOnProgress</code>	boolean	true	Use to enable/disable autoplay update on progress. To disable autoplay update on progress, set the value to false.
<code>autoplayUpdateOnTransitionEnd</code>	boolean	true	Use to enable/disable autoplay update on transition end. To disable autoplay update on transition end, set the value to false.
<code>autoplayUpdateOnVisible</code>	boolean	true	Use to enable/disable autoplay update on visible. To disable autoplay update on visible, set the value to false.
<code>autoplayUpdateOnWindowResize</code>	boolean	true	Use to enable/disable autoplay update on window resize. To disable autoplay update on window resize, set the value to false.
<code>autoplayUpdateOnWindowScroll</code>	boolean	true	Use to enable/disable autoplay update on window scroll. To disable autoplay update on window scroll, set the value to false.
<code>autoplayUpdateOnWindowZoom</code>	boolean	true	Use to enable/disable autoplay update on window zoom. To disable autoplay update on window zoom, set the value to false.
<code>autoplayUpdateOnWindowFocus</code>	boolean	true	Use to enable/disable autoplay update on window focus. To disable autoplay update on window focus, set the value to false.
<code>autoplayUpdateOnWindowBlur</code>	boolean	true	Use to enable/disable autoplay update on window blur. To disable autoplay update on window blur, set the value to false.
<code>autoplayUpdateOnWindowClose</code>	boolean	true	Use to enable/disable autoplay update on window close. To disable autoplay update on window close, set the value to false.
<code>autoplayUpdateOnWindowOpen</code>	boolean	true	Use to enable/disable autoplay update on window open. To disable autoplay update on window open, set the value to false.
<code>autoplayUpdateOnWindowMove</code>	boolean	true	Use to enable/disable autoplay update on window move. To disable autoplay update on window move, set the value to false.
<code>autoplayUpdateOnWindowResize</code>	boolean	true	Use to enable/disable autoplay update on window resize. To disable autoplay update on window resize, set the value to false.
<code>autoplayUpdateOnWindowScroll</code>	boolean	true	Use to enable/disable autoplay update on window scroll. To disable autoplay update on window scroll, set the value to false.
<code>autoplayUpdateOnWindowZoom</code>	boolean	true	Use to enable/disable autoplay update on window zoom. To disable autoplay update on window zoom, set the value to false.
<code>autoplayUpdateOnWindowFocus</code>	boolean	true	Use to enable/disable autoplay update on window focus. To disable autoplay update on window focus, set the value to false.
<code>autoplayUpdateOnWindowBlur</code>	boolean	true	Use to enable/disable autoplay update on window blur. To disable autoplay update on window blur, set the value to false.
<code>autoplayUpdateOnWindowClose</code>	boolean	true	Use to enable/disable autoplay update on window close. To disable autoplay update on window close, set the value to false.
<code>autoplayUpdateOnWindowOpen</code>	boolean	true	Use to enable/disable autoplay update on window open. To disable autoplay update on window open, set the value to false.
<code>autoplayUpdateOnWindowMove</code>	boolean	true	Use to enable/disable autoplay update on window move. To disable autoplay update on window move, set the value to false.

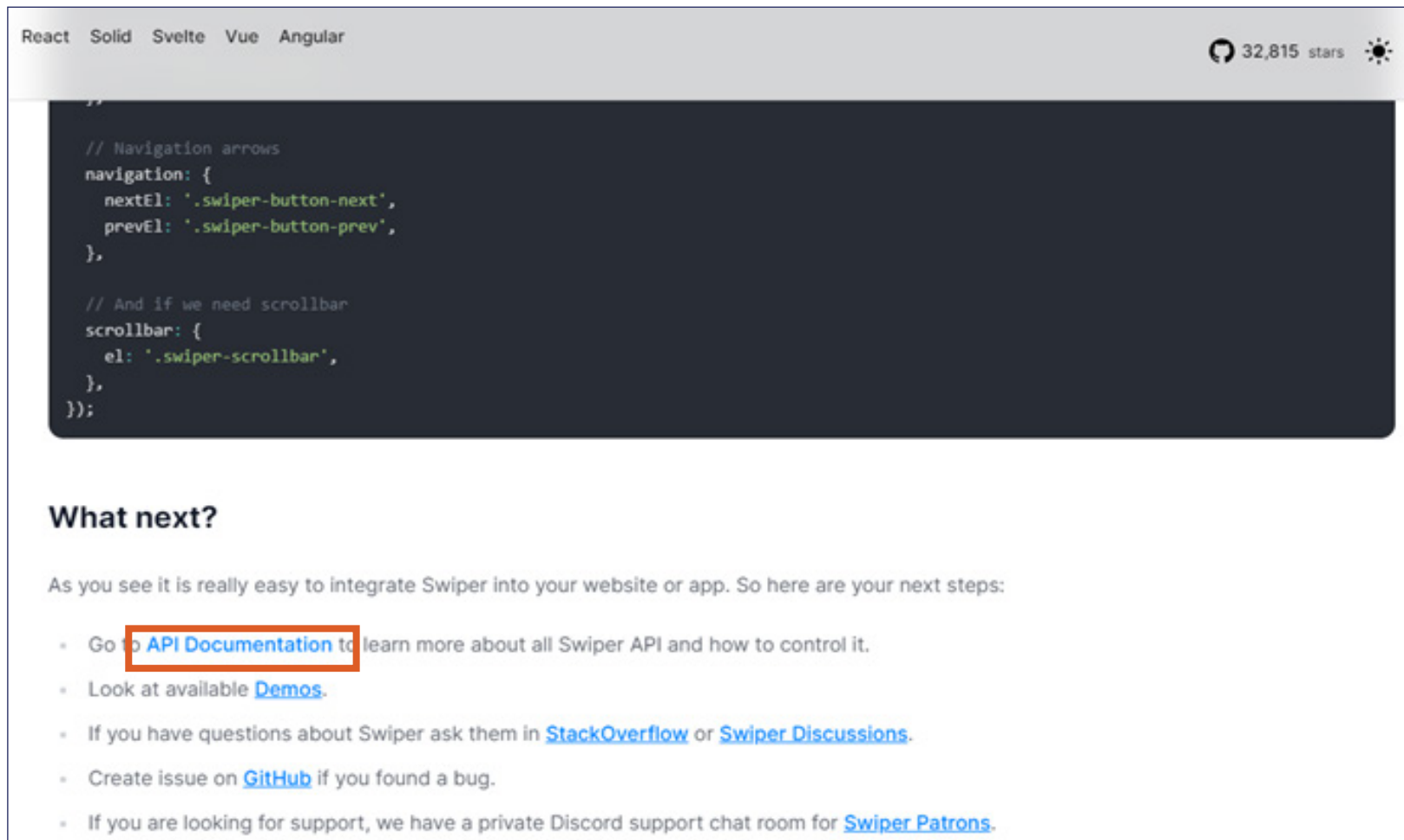
Clique e acompanhe a aula na Alura

# Corrigindo os bugs (erros de código)

Importamos a biblioteca Swiper e criamos o carrossel da AluraBooks, porém ele ainda não funciona. Nesta aula, vamos aprender a inicializar o Swiper e a configurar alguns parâmetros do Javascript consultando a documentação completa da API, disponível no site do Swiper. Acesse o site por meio do link <https://swiperjs.com/>. Em seguida, clique em *Whats next?*.



No menu que abrir, clique em *API Documentation* no centro da tela.



The screenshot shows the Swiper.js website interface. At the top, there is a navigation bar with links for React, Solid, Svelte, Vue, and Angular. On the right side of the bar, it displays '32,815 stars' and a sun icon. Below the navigation bar, there is a dark-themed code editor showing the following JavaScript code:

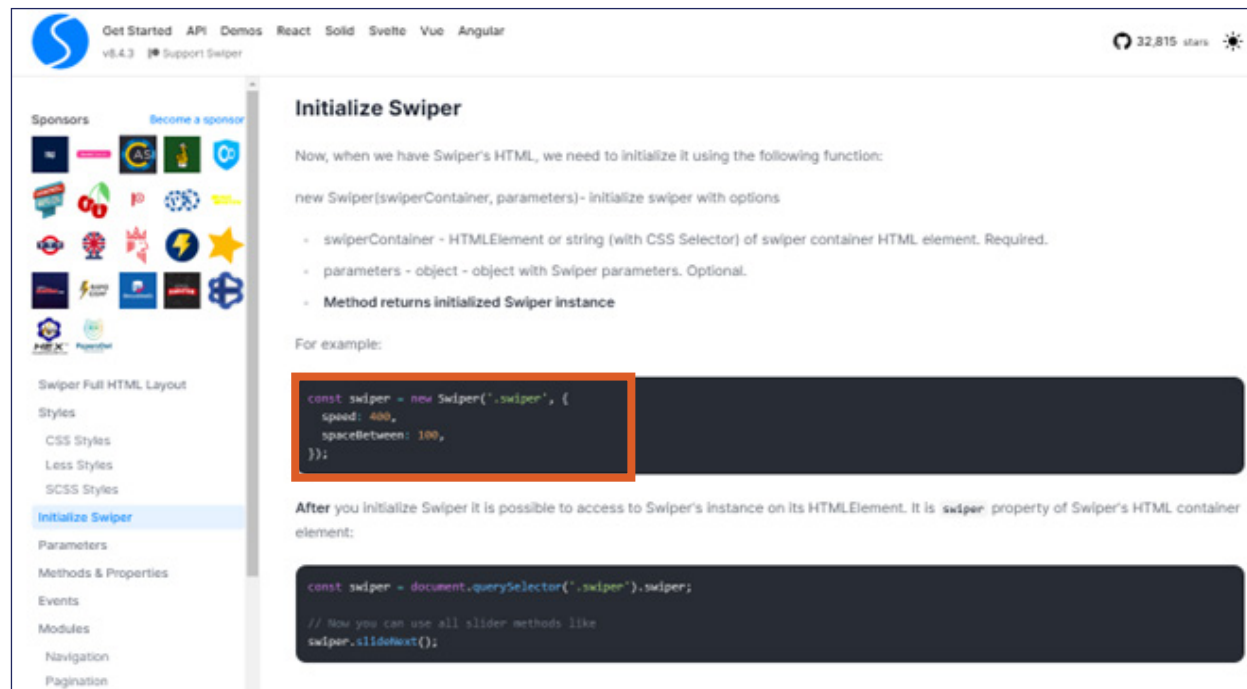
```
// Navigation arrows
navigation: {
  nextEl: '.swiper-button-next',
  prevEl: '.swiper-button-prev',
},

// And if we need scrollbar
scrollbar: {
  el: '.swiper-scrollbar',
},
});
```

Below the code editor, the section 'What next?' is displayed. It contains the text: 'As you see it is really easy to integrate Swiper into your website or app. So here are your next steps:'. Below this text is a list of five steps:

- Go to **API Documentation** to learn more about all Swiper API and how to control it.
- Look at available [Demos](#).
- If you have questions about Swiper ask them in [StackOverflow](#) or [Swiper Discussions](#).
- Create issue on [GitHub](#) if you found a bug.
- If you are looking for support, we have a private Discord support chat room for [Swiper Patrons](#).

Na página que abrir, teremos acesso a toda a documentação do Swiper. Vamos seleccionar a opção *Initialize Swiper* no menu esquerdo, e copiar todo o código do primeiro bloco de exemplo.



The screenshot shows the Swiper.js documentation website. The left sidebar contains a menu with the following items: Sponsors, Styles, CSS Styles, Less Styles, SCSS Styles, **Initialize Swiper** (highlighted), Parameters, Methods & Properties, Events, Modules, Navigation, and Pagination. The main content area is titled 'Initialize Swiper' and explains that Swiper needs to be initialized. It lists the parameters for the `new Swiper()` function: `swiperContainer` (required) and `parameters` (optional). A code block shows an example of initializing Swiper with a speed of 400 and space between of 100. Below this, another code block shows how to access the Swiper instance via `document.querySelector('.swiper').swiper` and use methods like `swiper.slideNext()`.

Get Started API Demos React Solid Svelte Vue Angular  
v8.4.3 Support Swiper 32,815 stars

Sponsors Become a sponsor

Initialize Swiper

Now, when we have Swiper's HTML, we need to initialize it using the following function:

new Swiper(swiperContainer, parameters) - initialize swiper with options

- swiperContainer - HTMLElement or string (with CSS Selector) of swiper container HTML element. Required.
- parameters - object - object with Swiper parameters. Optional.
- Method returns initialized Swiper instance

For example:

```
const swiper = new Swiper('.swiper', {  
  speed: 400,  
  spaceBetween: 100,  
});
```

After you initialize Swiper it is possible to access to Swiper's instance on its HTMLElement. It is `swiper` property of Swiper's HTML container element:

```
const swiper = document.querySelector('.swiper').swiper;  
  
// Now you can use all slider methods like  
swiper.slideNext();
```

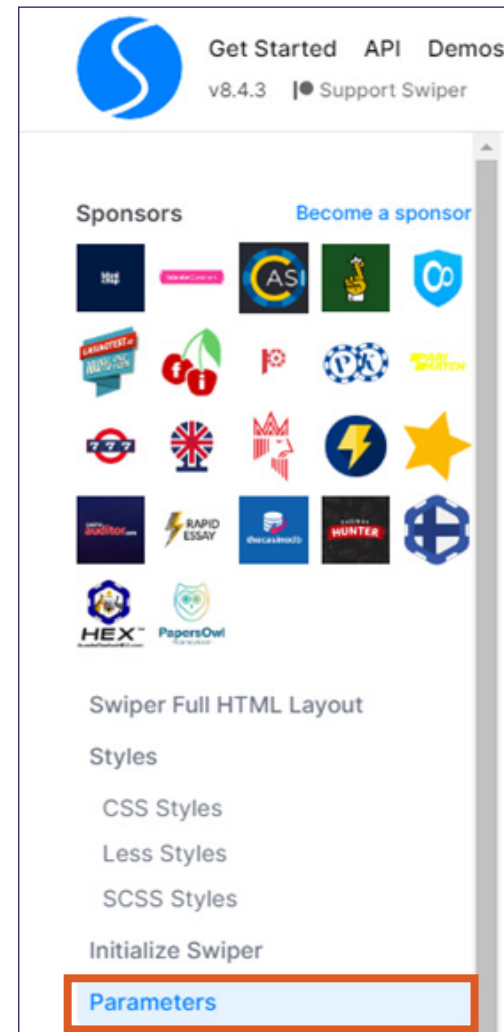
Vamos criar uma nova tag **<script>**, antes do fechamento do **<body>**, e colar todo o código de inicialização do Swiper dentro, no arquivo *index.html*.

```
<script src="https://cdn.jsdelivr.net/npm/swiper@8/swiper-bundle.min.js"></script>
<script>
  const swiper = new Swiper('.swiper', {
    speed: 400,
    spaceBetween: 100,
  });
</script>
</body>
```

Temos dois parâmetros, o **speed** e o **spaceBetween**. O **speed** refere-se à velocidade entre a troca de imagens do carrossel, enquanto o **spaceBetween** refere-se ao espaçamento entre cada slide. Vamos remover o parâmetro **speed**, pois este faz sentido apenas quando temos uma troca automática das imagens, que não é o caso aqui. Como estamos construindo nosso projeto pensando em mobile-first, vamos alterar o valor de **spaceBetween** para 10, pois 100 ficaria muito grande em uma tela de celular.

```
<script>
  const swiper = new Swiper('.swiper', {
    spaceBetween: 10,
  });
</script>
```

Podemos consultar os parâmetros disponíveis clicando na opção *Parameters*, no menu esquerdo do site do Swiper.





No código, em **breakpoints** temos o parâmetro **slidesPerView**, onde podemos definir quantas imagens podemos visualizar por vez. Vamos adicionar esse parâmetro em nosso carrossel e definir seu valor para 3.

```
<script>
  const swiper = new Swiper('.swiper', {
    spaceBetween: 10,
    slidesPerView: 3,
  });
</script>
```

O resultado na página da AluraBooks deve ficar da seguinte maneira:



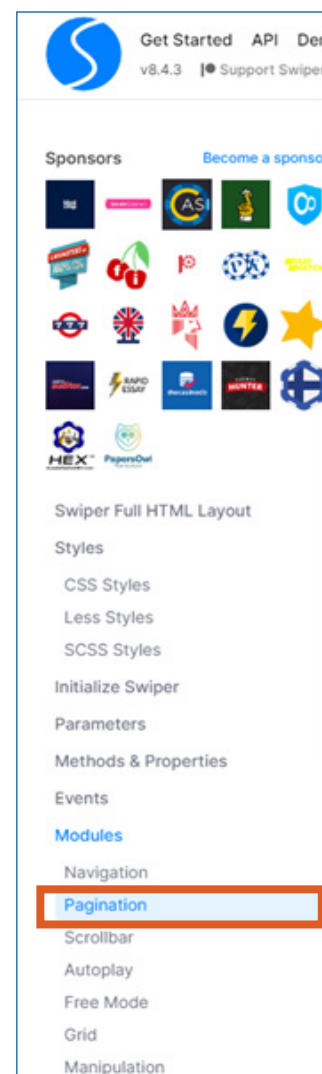
Para que as imagens não fiquem uma em cima da outra, vamos definir uma largura de 100% para elas. Acrescente o seguinte código no arquivo *carrossel.css*.

```
.swiper-slide img {  
  width: 100%;  
}
```



Vamos ocultar as setas que passam as imagens do carrossel, pois elas aparecerão apenas quando visualizarmos nosso site por um computador, e não pelo celular. Para isso, podemos atribuir às classes correspondentes um **display: none**.

```
.swiper-button-prev,  
.swiper-button-next {  
  display: none;  
}
```

Vamos acrescentar a paginação em nosso carrossel. Para isso, no site Swiper podemos consultar a guia *Pagination*.



Copie o código da descrição (*Description*).

 32,815 stars 

---

Description

---

Object with pagination parameters or boolean `true` to enable with default settings.

```
const swiper = new Swiper('.swiper', {
  pagination: {
    el: '.swiper-pagination',
    type: 'bullets',
  },
});
```

Cole o bloco referente à paginação dentro do nosso script.

```
<script>
  const swiper = new Swiper('.swiper', {
    spaceBetween: 10,
    slidesPerView: 3,
    pagination: {
      el: '.swiper-pagination',
      type: 'bullets',
    },
  });
</script>
```

Agora vamos posicionar a tag **<div>** referente à paginação acima da **<div>** referente ao carrossel.

```
<div class="swiper">  
  <!-- Additional required wrapper -->  
  <!-- If we need pagination -->  
  <div class="swiper-pagination"></div>  
  <div class="swiper-wrapper">
```

Por fim, vamos definir uma nova posição e margem para a paginação ser exibida corretamente, conforme vemos no projeto Figma, no arquivo *carrossel.css*.

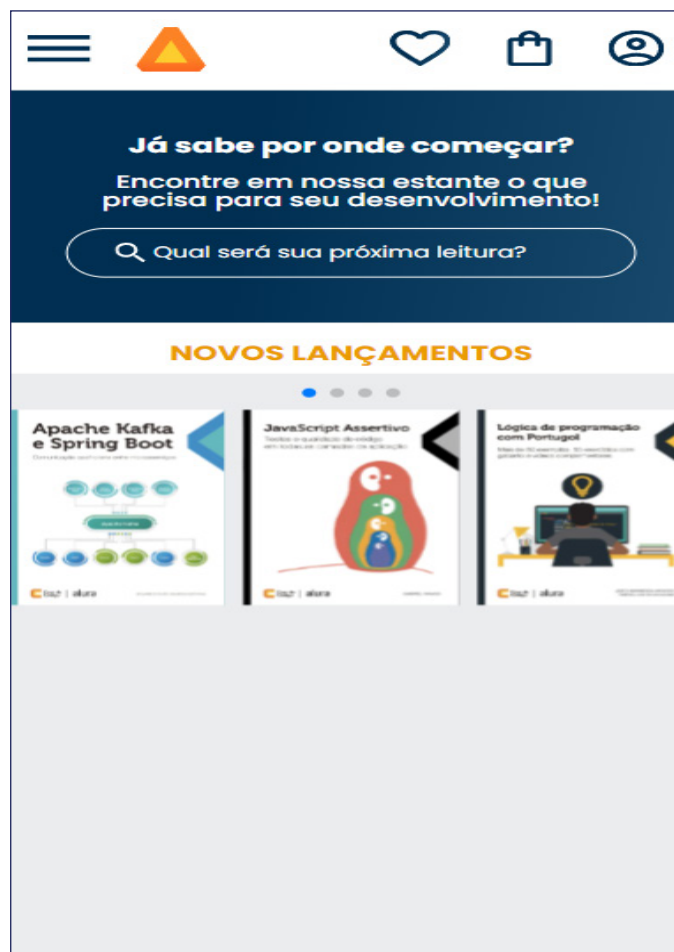
```
.swiper-pagination {  
  position: initial;  
  margin: 0.5em 0;  
}
```

Podemos observar que temos uma barra de rolagem indesejada na parte de baixo do carrossel. Para removê-la, basta eliminar a linha a seguir, no arquivo *index.html*.

```
<!-- If we need scrollbar -->  
<div class="swiper-scrollbar"></div>
```



Como resultado, teremos:



# Desafio

Nesta aula, finalizamos o carrossel do AluraBooks utilizando uma biblioteca Javascript externa chamada Swiper.

A biblioteca Swiper possui diversos parâmetros. O desafio será encontrar quais são os principais tipos dos dados que esses parâmetros recebem, em geral.

Avalie este material acessando o link: <https://forms.gle/EcEZdj59zGS9QTWY9>