

06

Continuando a refatoração

Transcrição

Em `carousel.js`, no trecho de código abaixo, um detalhe em relação ao JS é que colocamos o `this` pensando em pegar o elemento que foi clicado, no caso o botão (`btn`):

```
// Percorre todos os botoes controladores
btws.forEach(function(btn) {
  btn.addEventListener('click', function() {

    // btn.href = 'javascript:void(0)'
    noticias.forEach(function(noticia) {
      noticia.style.display = 'none';

      if ( this.getAttribute('data-sliderItem') === noticia.getAttribute('data-noticia') ) {
        noticia.style.display = 'block'
      }
    });
  });

```

Porém, antes dele alcançar a parte de cima, temos o `forEach()`, ou seja, quando se usa o `this` no `forEach()`, este é o objeto de interação. Verificamos portanto se `noticia` possui o atributo `data-sliderItem` que, não sendo encontrado, ocasionará em falha.

Para que isso seja ignorado e seja pego o `this` que nos importa - o do `addEventListener` -, usaremos o `bind()`:

```
// btn.href = 'javascript:void(0)'
noticias.forEach(function(noticia) {
  noticia.style.display = 'none';

  if ( this.getAttribute('data-sliderItem') === noticia.getAttribute('data-noticia') ) {
    noticia.style.display = 'block'
  }
});

.bind(this));

```

Salvaremos o arquivo, voltaremos ao navegador e testaremos. Ao chegarmos à parte do carrossel e apertarmos "Enter", nenhum conteúdo é mostrado. O que será que está acontecendo? Por algum motivo, em `carousel.js`, ele não está caindo no `if()` que colocamos no trecho acima.

Isto é tão verdade que, ao trocarmos `this.getAttribute('data-sliderItem') === noticia.getAttribute('data-noticia')` por `true`, todas as notícias são mostradas, uma embaixo da outra. Provavelmente ocorreu algum erro no HTML, vamos conferi-lo.

Esquecemos de colocar o identificador `data-noticia` nas `li`s da `listaDeArtigos`, o que corrigiremos:

```
<ol class="listaDeArtigos" aria-labelledby="tituloBlog">

  <li id="new0" class="listaDeArtigos-item" data-noticia="0">
    <!-- ... -->
  </li>

  <li id="new1" class="listaDeArtigos-item" data-noticia="1">
    <!-- ... -->
  </li>

  <li id="new2" class="listaDeArtigos-item" data-noticia="2">
    <!-- ... -->
  </li>
<!-- ... -->
</ol>
```

Com isto, verificamos se quando o usuário clicar ou utilizar o "Enter" no link referente a `new2`, o resultado de `data-sliderItem` bate com o `data-noticia` correspondente. Se este for o caso, deve-se usar o `display: block`, enquanto o restante permanecerá com `display: none`.

Vamos conferir o comportamento do carrossel salvando tudo e voltando ao navegador, atualizando-o. Testando com o NVDA, a chamada para a notícia é lida corretamente, e com "Shift + Tab", o foco volta para um dos botões anteriores, "Assinar plano".

É possível criar vários componentes com CSS, os quais normalmente faríamos com JS, no entanto, isto pode afetar negativamente em relação à acessibilidade.

Para quem gosta e está acostumado a usar, a questão da acessibilidade do Bootstrap é decente, pelo menos no que concerne ao carrossel, e se você tiver interesse em conhecer, lembre-se sempre de testar antes.