

Refatorando

Transcrição

Antes de melhorarmos o bloco de `if()`, notaremos que ao clicarmos nos *bullets* do carrossel, acontece um deslocamento da tela. Para algumas pessoas isto pode ser meio chato, e a forma mais prática para ajeitarmos isto é pegar seu `href` e aplicar um ajuste. Em `carousel.js`, teremos então:

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

    btn.href = 'javascript:void(0)'
```

Salvaremos, voltaremos ao navegador e o atualizaremos. Aparentemente, tudo funciona como gostaríamos, inclusive testando com o "Tab" e o leitor de telas, e o deslocamento da tela também deixou de ocorrer.

Entretanto, ao apertarmos "Enter", o NVDA não lê mais nada. O que acontece é que retiramos um comportamento padrão do link, e isto impacta negativamente no nosso código. Portanto, teremos que remover o que acabamos de acrescentar. Vamos deixá-lo comentado:

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

//      btn.href = 'javascript:void(0)'
```

Retornaremos ao navegador e atualizaremos a página. Poderíamos utilizar o `event.preventDefault()`, mas basicamente daria na mesma, pois ele não deixaria o link funcionar. O deslocamento da tela, então, continuará acontecendo.

Vimos o `javascript:void()` porque às vezes teremos que abrir mão de um comportamento que não é o ideal, do que seria esperado, para deixar a acessibilidade do site mais decente. Às vezes acontece o oposto, o que não é recomendado. É importante a discussão com seu time de desenvolvimento, pois às vezes você quer deixar sua aplicação 100% acessível mas o cliente acha inaceitável, por algum motivo.

Agora, vamos melhorar o bloco dos `if()` s. Da maneira em que está, teremos um problema se quisermos acrescentar mais uma notícia, já que isto causará a necessidade de alterarmos manualmente o HTML e o JS. Uma ideia é usarmos o JavaScript para contar as notícias e fazer o `display: block` ou `display: none` de acordo com o `data-sliderItem`, ou seja, do botão que foi clicado.

Vamos lidar com isso deletando os blocos de `if()` s em `carousel.js`, focando no primeiro `if()`, de que deletaremos a condição `=== '0'`. Como queremos pegar uma lista de todas as notícias, usaremos o `document.querySelectorAll()`.

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

```
});
```

```
document.querySelector('.listaDeArtigos-slider-item .escondeVisualmente').remove();  
this.append(indicadorSlideAtual);
```

Salvaremos e voltaremos ao `carousel.js` para continuarmos:

```
// btn.href = 'javascript:void(0)'  
document.querySelectorAll('.listaDeArtigosItem').forEach(function(noticia) {  
    noticia.style.display = 'none';  
  
    if ( this.getAttribute('data-sliderItem') === noticia.getAttribute('data-noticia')) ) {  
        noticia.style.display = 'block'  
    }  
  
});
```

```
document.querySelector('.listaDeArtigos-slider-item .escondeVisualmente').remove();  
this.append(indicadorSlideAtual);
```

Lembre-se de que o código pode ser feito de diversas maneiras para alcançar um mesmo resultado, e de que neste curso estamos focando em acessibilidade.

Falando um pouquinho sobre performance em JavaScript, reparem que o `querySelectorAll()` está contido em um `forEach()`, ou seja, o `querySelectorAll()` será dado no documento inteiro, em busca das classes, para cada botão que tivermos.

Vamos, então, criar uma variável fora deste `forEach()`, substituindo `document.querySelectorAll('.listaDeArtigos-item')` do trecho acima por `noticias`, e inserindo-o junto das demais variáveis, assim:

```
// Variaveis  
  
var btns = document.querySelectorAll('.listaDeArtigos-slider-item');  
var noticias = document.querySelectorAll('.listaDeArtigos-item');  
  
var new0 = document.querySelector('#new0');  
var new1 = document.querySelector('#new1');  
var new2 = document.querySelector('#new2');  
  
new0.style.display = 'block'
```

Observação: neste vídeo é incluído o atributo customizável `data-noticia` para cada item de `listaDeArtigos` no arquivo HTML, porém o código sofreu alterações em vídeos posteriores. Fique tranquilo que seu código funcionará bem da mesma forma, conforme veremos adiante. Qualquer dúvida, nos falamos no [fórum](https://cursos.alura.com.br/forum/curso-acessibilidade-web-front-end-parte-2/) (<https://cursos.alura.com.br/forum/curso-acessibilidade-web-front-end-parte-2/>).

