

Fechar modal

Transcrição

Ao abrirmos o modal com o clique no botão "Receber destaques por email", conforme navegamos com a tecla "Tab", há um momento em que o site insiste em deixar o elemento `video` como focável.

Em `inert.js`, a variável `_focusableElementsString` serve basicamente para selecionar todos os elementos focáveis. Se futuramente houver outro elemento focável no HTML6 ou HTML7, basta incluí-lo.

Recolocaremos `video`, então, como se vê abaixo:

```
/** @type {string} */  
var _focusableElementsString = ['a[href]', 'area[href]', 'input:not([disabled])', 'select:not([disal
```

Voltaremos ao navegador para testar a alteração, que funcionará perfeitamente.

Sabemos que o `inert.js` foi copiado do Github, mas se formos utilizar vídeos em nossas páginas, precisaremos fazer esta pequena modificação no código, incluindo o elemento `video` na listagem dos focáveis.

Outra solução, não muito tradicional, é usar o `display: none` em `dialog.js`, para quando os controles do vídeo forem removidos. E em `fechandoDialog()`, usaríamos o `display: block`. Funcionaria, mas prefiro editar o `inert.js`, por motivos de organização e boas práticas.

Para resolver a questão do "xX botão", em `index.html` iremos à parte do botão de classe `dialogNewsletter-fechar`, com um "X" entre as tags de abertura e fechamento de `button`, o qual removeremos.

Nos cursos anteriores, vimos que todo elemento necessita de um conteúdo. Sendo assim, acrescentaremos no lugar do "X" a função deste botão, que pode ser algo como "Fechar modal":

```
<button class="dialogNewsletter-fechar" type="button">  
  Fechar modal  
</button>
```

Em seguida, veremos como o NVDA se comporta diante desta alteração - ele exibe "xFechar modal". De onde vem esse **x**? Vamos voltar ao Brackets e analisar melhor a classe `dialogNewsletter-fechar` por meio do atalho "Ctrl + Shift + O". Encontraremos isto:

```
.dialogNewsletter-fechar:before {  
  content: 'x';  
  color: #666;  
  width: 32px;  
  height: 32px;  
  display: block;  
  position: absolute;  
  top: 0;  
  left: 0;
```

```
z-index: 2;
font-size: 20px;
font-weight: bold;
}
```

O front-end que fez este site colocou o "x" com o pseudoelemento `:before` (poderia ser o `:after`). O que acontece é que se o `content` está sendo criado via CSS, ainda assim o NVDA lê.

Para solucionarmos este problema, poderíamos usar uma imagem no lugar do `x`, puxando por `url(..img/fechar.svg)`, por exemplo. Ao fazermos isto e testarmos no navegador, o NVDA lê "Fechar modal", que é o conteúdo do botão, porém perdemos o efeito de *hover*, da passagem do mouse.

Uma opção seria usar duas imagens e acrescentar o trecho abaixo em `dialogNewsletter.css`:

```
.dialogNewsletter-fechar:hover:before {
  content: url(..img/fechar-hover.svg);
}
```

Porém isso seria um pouco mais trabalhoso. Percebam que começamos a nos engessar muito, pois dependemos de outros profissionais e ferramentas, enquanto poderíamos resolver via código. Vamos manter o `x` no CSS de `dialogNewsletter-fechar:before`.

Queremos que o botão tenha algo como uma etiqueta, um rótulo indicando o fechamento da janela, e que o NVDA, no caso, ignore o conteúdo e leia apenas o rótulo. Em `index.html` teremos, portanto:

```
<button class="dialogNewsletter-fechar" type="button" aria-label="Fechar janela modal">
  Fechar modal
</button>
```

Feito isto, salvaremos o arquivo e voltaremos ao navegador para teste. Dessa vez, ao navegarmos com a tecla "Tab", o que surge ao chegarmos ao pop up é a leitura "Fechar janela modal".

Sempre que houver problemas deste tipo em seus projetos, em que visualmente é preciso um conteúdo, mas o leitor de tela necessita de outro, o uso de `aria-label` é a melhor solução.

Existem outras maneiras de resolver isto, "gambiarras" tais como a que sugerimos, de usar duas imagens, mexermos nos pseudoelementos, enfim, *n* jeitos diferentes.

Usando-se `aria-label`, reparem que o conteúdo entre as tags `<button>` foi ignorado, isto é, ele não importa. Adiante, seguiremos trabalhando na acessibilidade desta janela!