

Foco inconveniente

Transcrição

Conforme vamos navegando com "Shift + Tab" na nossa página, tocamos em discussões recorrentes sobre se o foco deveria chegar à aba do navegador ou não. Pessoalmente acho que do jeito que está agora funciona bem. A partir do momento em que damos mais quatro "Tab"s, o foco vai para o fim da página, nos controles do elemento de um vídeo que aparece ali - o leitor de tela acusa "Tela inteira botão". Fica um pouco estranho.

A solução que proponho, uma espécie de "ajuste técnico" que encontrei, envolve o vídeo. Ele se encontra no conteúdo fora da *dialog*, o qual, ao surgir, possui o `inert` criado, mas por algum motivo os controles do HTML5 em relação ao elemento vídeo se sobrepõem ao `inert`.

Para tentarmos resolver isto, temos a opção de abrir `inert.js` e indicar que o elemento `video` também será não focável, acrescentando-o no código abaixo:

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

Feito isso, voltaremos ao navegador, que atualizaremos e testaremos com o leitor de tela.

Como os controles criados pelo HTML5 não são exatamente do elemento `video`, e sim elementos criados dentro de outros, ou seja, aos quais não obtemos acesso via CSS, a alteração que fizemos não adianta em nada.

Deixaremos o código como estava anteriormente, da forma como copiamos do Github e, quando abrirmos a *dialog*, estes controles serão removidos. Em `dialog.js`, então, teremos:

```
// Quando abrir a dialog...  
btnAbreDialog.addEventListener('click', function() {  
  dialog.classList.add('dialogNewsletter--aberto');  
  document.querySelector('.dialogNewsletter-label').focus();  
  conteudoForaDialog.inert = true;  
  
  document.querySelector('video').removeAttribute('controls');  
});
```

Salvaremos o arquivo e voltaremos ao navegador para verificar o que foi feito. Quando usamos o "Tab" para navegar na tela, passando pelo modal, veremos que os controles do vídeo no fim da página sumiram. Ao retornarmos para a tela da página - fechando o modal -, no entanto, os controles permanecem inexistentes. Precisaremos resolver isso.

Copiaremos o trecho de código que acabamos de digitar e o colaremos na parte de fechamento da *dialog*, como se vê abaixo:

```
function fechandoDialog() {  
  document.activeElement.blur();  
  dialog.classList.remove('dialogNewsletter--aberto');  
  conteudoForaDialog.inert = false;  
  btnAbreDialog.focus();
```

```
document.querySelector('video').setAttribute('controls');  
});
```

Para não repetirmos `document.querySelector('video')` em ambos os trechos (de abertura e fechamento de *dialog*), criaremos uma variável:

```
var video = document.querySelector('video');
```

E substituiremos as linhas que se repetem com `video.removeAttribute('controls')` e `video.setAttribute('controls', true)`, sem esquecer de passar o segundo parâmetro em `setAttribute`. Em uma linha removemos os controles, e na outra estamos colocando.

Testaremos novamente no navegador, e desta vez tudo funciona como gostaríamos: os controles do vídeo saem quando o modal está na tela, e quando este é fechado, os controles voltam normalmente. Com esta alteração, todo o foco do elemento está na *dialog*. Trata-se de um ajuste técnico que resolve a nossa questão, e melhora em muito a acessibilidade do site.

A seguir lidaremos com algo que acontece e desconhecemos no modal, o qual o NVDA lê como

```
botão  
xX  
botão
```