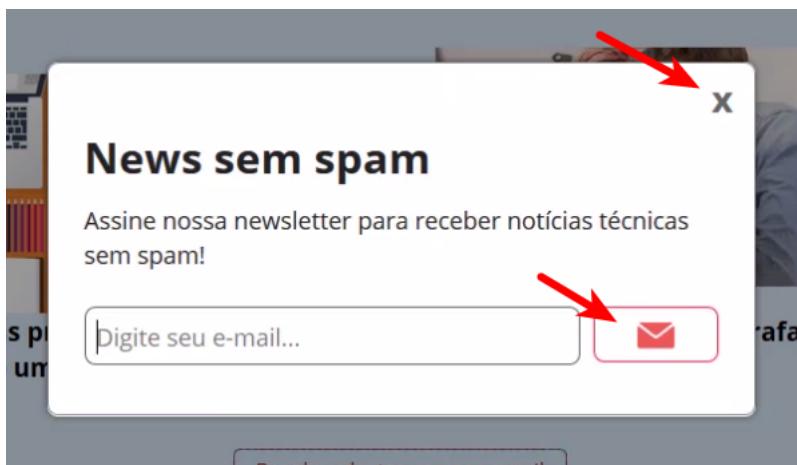


03

Tirando o foco

Transcrição

No formulário do modal (`dialogNewsletter-form`) no Brackets, temos dois botões, um para assinatura da newsletter e outro para fechar o modal:



Porém, como é que o browser vai saber qual botão submeterá o formulário? Da forma que isto aparece no código no momento, ele terá que "adivinar". Para termos certeza, indicaremos os tipos dos botões:

```
<button class="dialogNewsletter-btn" type="submit">
    Assinar newsletter
</button>
<button class="dialogNewsletter-fechar" type="button">
    X
</button>
```

Isso deixará nosso código melhor. Em seguida atualizaremos a página e testaremos o foco, que está na parte de trás do modal. No futuro, espero muito que isso se resolva simplesmente com o uso de `role="dialog"`, e que tudo que estiver além dessa `dialog` não seja focável. Infelizmente atualmente ainda precisamos dar um jeito.

E o jeito que temos é um atributo chamado `inert`, cuja [lib \(`https://github.com/WICG/inert`\)](https://github.com/WICG/inert) usaremos, um JavaScript puro e não tão difícil de se trabalhar. Seria muito legal se conseguíssemos simplesmente inclui-lo no nosso HTML, diretamente, e esta é a ideia geral desta biblioteca.

Iremos implementá-la no nosso projeto. Clicaremos em `inert.js` da pasta `dist`, copiaremos o código, voltaremos ao Brackets, abriremos um novo projeto, colaremos e salvaremos, em "Projeto > js" com o nome `inert.js` também.

Caso prefira, pode-se copiar a versão minificada (`inert.min.js`), pois não mexeremos nesta lib. Tudo que alterarmos será em nosso arquivo `.js`.

Não poderemos nos esquecer de chamar o arquivo recém salvo no HTML, com `<script src="js/inert.js"></script>`, após a tag de fechamento do `<footer>` e antes da tag de fechamento do `<body>`.

Lembrando que ele precisa ficar antes do `<script>` que se refere à `dialog`, pois alguns de seus recursos serão utilizados ali, assim:

```
<script src="js/carousel.js"></script>
<script src="js/inert.js"></script>
<script src="js/dialog.js"></script>
```

Caso você não tenha interesse em utilizar uma biblioteca, será necessário selecionar todos os elementos focáveis, tais como links, `select`s, `input`s e afins via JavaScript, e retirar manualmente seus focos. Uma vez que este curso não tem como prioridade ensinar JavaScript, optaremos por utilizar a biblioteca, pela praticidade.

Cuidado: normalmente, com o Bootstrap, por exemplo, pode ocorrer dele ser chamado primeiro e depois o jQuery, o que não ficará certo.

Quando nos deparamos com o `inert` em algum elemento, o foco será totalmente perdido, ou seja, o elemento deixará de ser focável. Testaremos acessando `index.html` e procurando o formulário na parte inferior do site.

Incluiremos então o atributo na seção inteira da classe `secaoContato`:

```
<section class="secaoContato" inert="true">
```

Ao lidarmos com uma classe, costumamos pensar que trata-se de algo que podemos mudar no CSS, e que isto não afetará em nada a funcionalidade da página. Mas neste caso qualquer mudança acabará afetando, sim. Portanto, para prevenirmos isto, deixaremos `conteudoForaDialog` como `id`, e não `class`:

```
<div id="conteudoForaDialog">
```

Precisaremos colocar este `inert` no nosso arquivo, no conteúdo fora da `dialog`, não quando a página for carregada, e sim quando a `dialog` for aberta:

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

Testando no navegador com o NVDA, verificaremos que tudo funciona conforme gostaríamos! Desta vez o foco está na aba, passando às informações da página ao lado da barra de endereços do navegador e para a barra de endereços.

Quando fechamos a página, porém, o foco não é encontrável em lugar algum, e não conseguimos clicar em nada. Isso acontece porque o `inert` ainda está presente no HTML.

Ao inspecionarmos o elemento no navegador para confirmar, encontraremos `aria-hidden="true" inert=""`, isto é, ele existe, em `conteudoForaDialog`.

Então, quando a `dialog` for fechada, será necessário remover o `inert`. Colocaremos o `conteudoForaDialog` dentro as variáveis de `dialog.js`:

```
//Variáveis
var conteudoForaDialog =
document.querySelector('#conteudoForaDialog');
```

Isto ajudará o browser também, evitando que ele rode um `querySelector` duas vezes. Sendo assim, teremos, em `dialog.js`:

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

No mesmo arquivo, já temos uma função que realiza algumas operações quando a `dialog` é fechada, então vamos fazer o mesmo com o nosso modal, que em vez de `true` será `false`:

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

O que precisamos fazer com a `dialog` em si?

Se voltarmos ao navegador para testar apertando o "Enter" na janelinha e, com o lado direito do mouse, selecionar "Inspecionar Elemento (Q)", veremos que o `inert` se encontra ali, funcionando conforme esperado, porém o `false` que acabamos de colocar não está sendo usado.

Uma das maneiras de o removermos dali é usar o valor booleano em vez de `string`, o que também faremos com `true`, simplesmente removendo o apóstrofo de ambos.

Vamos testar novamente na página, navegando com a tecla "Tab" para ver que o foco se encontra, no caso, apenas na interface do navegador e da `dialog`. Verificaremos o que acontece em `inert=""` do inspecionador de elementos, ao fecharmos o modal.

Agora sim, ele foi removido, e conseguimos navegar tranquilamente por meio do teclado. Quando fechamos a `dialog`, o que ocorre com o foco? Ele volta ao conteúdo principal, o que não é muito interessante, já que o usuário estava lá embaixo quando acessou a janelinha.

As boas práticas de acessibilidade indicam que quando o foco for para alguma janela de pop up ou similar, ao voltar à página ele deve estar no último local acessado. Isto é, vamos voltar à `dialog.js` para informar que quando o modal for fechado queremos que o botão (`btnAbreDialog`) fique focado.

Atualizaremos a página, navegaremos usando o "Tab" até chegarmos à parte de abrir o pop up, o qual fecharemos em seguida. O foco voltou ao botão "Receber destaque por email".

Agora o nosso site está começando a ficar interessante, em relação à `dialog`.

