

03

Checkbox com focus

Transcrição

Vamos mexer no *checkbox*, pois o usuário de um leitor de telas não consegue saber o que quer dizer o `x` em

`x Receber destaque por email` caixa de seleção marcado

Inicialmente, desmarcaremos o *checkbox* e acessaremos o CSS para confirmar a existência de alguma regra em relação a ele. Basicamente, precisaremos incluir uma borda quando o usuário alcançar o *checkbox*, por exemplo, quando o `hover` ou o `focus` estiverem ali.

Queremos certas características visíveis quando o `contatoCampo-checkbox`, irmão direto do `contatoCampo-label`, estiver com o `hover`. E para que a navegação com o "Tab" também funcione corretamente, incluiremos um `focus`:

```
.contatoCampo-checkbox:hover + .contatoCampo-label:before,  
.contatoCampo-checkbox:focus + .contatoCampo-label:before {  
    border: 4px solid #333;  
}
```

Se testarmos o código acima, veremos que tudo funciona como esperado. Porém, não queremos que a borda destinada à mensagem fique animada, aparentemente mais grossa. Na classe `contatoCampo-label:before`, poderíamos deixar `transition: background-color .2s ease-in, color .2s ease-in`.

Voltaremos ao navegador para testar com o "Tab" e, ao passarmos pelo *checkbox* e usarmos a barra de espaço do teclado, ele é marcado. Como não queremos que ele tenha borda quando marcado, incluiremos `border: 0` ou `border-color: #015fbe` em `.contatoCampo-checkbox:checked + .contatoCampo-label:before`. Logo abaixo desta linha, incluiremos `line-height: .8` também.

Ainda assim, quando damos um "Tab", o campo de mensagem dá uma expandida. Isto é, quando colocamos foco ou similar em algum elemento com borda, suas medidas aumentam um pouquinho. Provavelmente teremos que diminuir a largura e a altura do box... Mas isto seria trabalhoso.

Um recurso interessante no Brackets quando lidamos com cor é selecionar o valor hexadecimal e usar o atalho "Ctrl + E" para abrir uma paleta de cores.

A forma como o browser precisa calcular as medidas de um elemento deve levar em consideração sua borda. Faremos isto desta forma:

```
.contatoCampo-checkbox:hover + .contatoCampo-label:before,  
.contatoCampo-checkbox:focus + .contatoCampo-label:before {  
    border: 4px solid #333;  
    box-sizing: border-box;  
}
```

Lembrando que precisaremos fazê-lo no elemento todo, portanto incluiremos a linha `box-sizing: border-box` em `contatoCampo-label:before`.

Agora sim, sabemos onde está o foco e o hover. Neste caso, poderíamos até remover o hover e manter apenas o foco, pois reparem que, como se trata de um campo de preenchimento, talvez não faça muito sentido. De qualquer forma, isto acaba sendo subjetivo. Vamos manter como está.

Precisamos melhorar ainda mais a acessibilidade e a usabilidade do nosso site, uma vez que o NVDA continua lendo o `checkbox` com o `x`, que vem justamente de `content: 'x'` em `contatoCampo-checkbox:checked + .contatoCampo-label:before`.

De que forma manteremos isto, ao passo em que deixaremos uma opção de desmarcar ou marcar um texto, algo assim?

Existem algumas maneiras de resolvemos isso, cada qual com suas vantagens e desvantagens. Poderíamos usar uma imagem, com `url`, mas o NVDA não lerá o conteúdo porque não é possível colocar um `alt` em uma imagem de um `background` no CSS.

Por não ser tão ideal trabalharmos com imagens neste momento, tentaremos outra opção, inicialmente mantendo `content: ''`, isto é, sem conteúdo. Outro ponto: como poderíamos criar algo que indique que esta caixa se encontra marcada?

Pode-se utilizar o próprio `x`, no entanto ele se relaciona muito a **fechar** algo. Assim, poderemos experimentar o símbolo de `check`. Isto envolve algum ajuste ou "maracutaia". O CSS ficará assim:

```
.contatoCampo-checkbox:checked + .contatoCampo-label:after {  
    content: '';  
    width: 10px;  
    height: 5px;  
    display: block;  
    /* background-color: #000; */  
    border-bottom: 2px solid #fff;  
    border-left: 2px solid #fff;  
    transform: rotate(-45deg);  
    position: absolute;  
    z-index: 2;  
    left: -3px;  
    top: 5px;  
}
```

Como definimos a `position` como `absolute`, seu elemento pai (no caso, `contatoCampo-label`) deve ser definido como `position: relative`:

```
.contatoCampo-label {  
    position: relative;  
}
```

Isto torna a usabilidade do site muito melhor. O HTML semântico, quando bem usado no seu código, faz com que o próprio NVDA indique a marcação (ou não) de um `checkbox`, por exemplo.

No fim desta aula, disponibilizarei um post ensinando a fazer uma `div` que parece um `checkbox` acessível.

O que importa para o cliente final não é se o código CSS ou HTML está com "gambiarras", ou de que forma você está fazendo as implementações. No fim das contas, o que é relevante é se o site está usável, com valor para a empresa e para o usuário.

