

03

Começando o responsivo

Transcrição

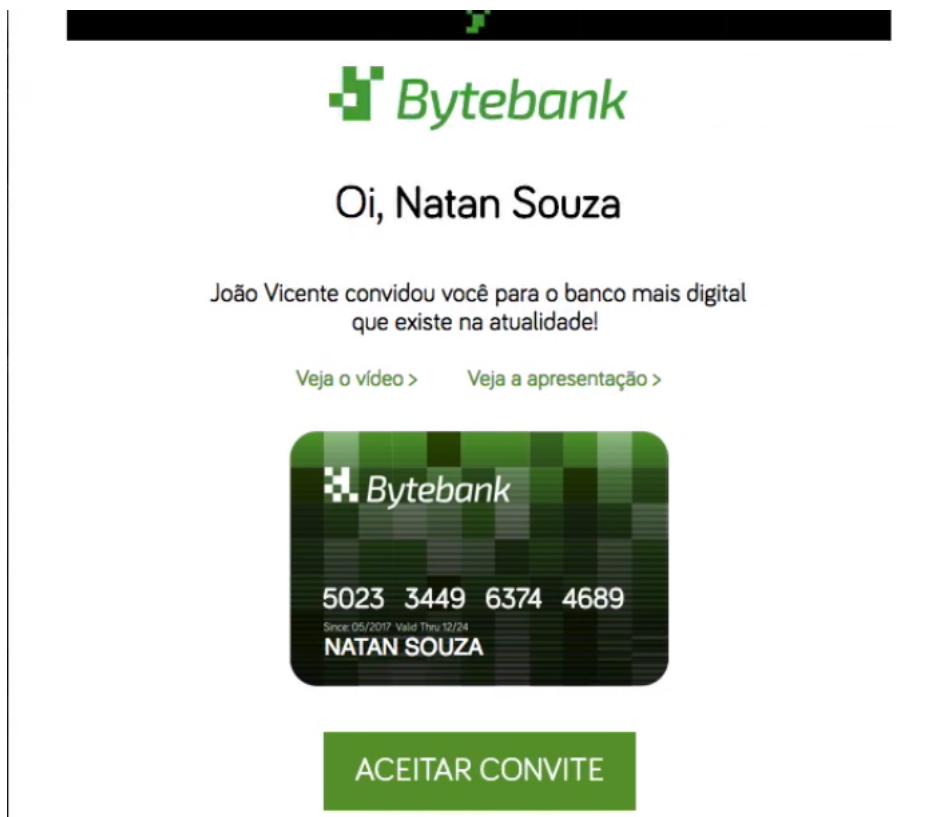
A lista de pré-requisitos do curso inclui o de *Web Design Responsivo*, no qual são abordados *media queries*, unidades de medida e UX.

Nosso e-mail ainda não está responsivo; qual seria o primeiro ponto a ser trabalhado? O ideal seria que a mensagem ocupasse a tela inteira, ou seja, 100% do espaço. Logo, a primeira alteração será na largura da tabela, cujo valor do `width` passará de 600 para 90%. Não usaremos o valor completo para o layout não se chocar com as quinas.

```
<center>
  <table align="center" cellpadding="0" cellspacing="0" border="0" width="90%">

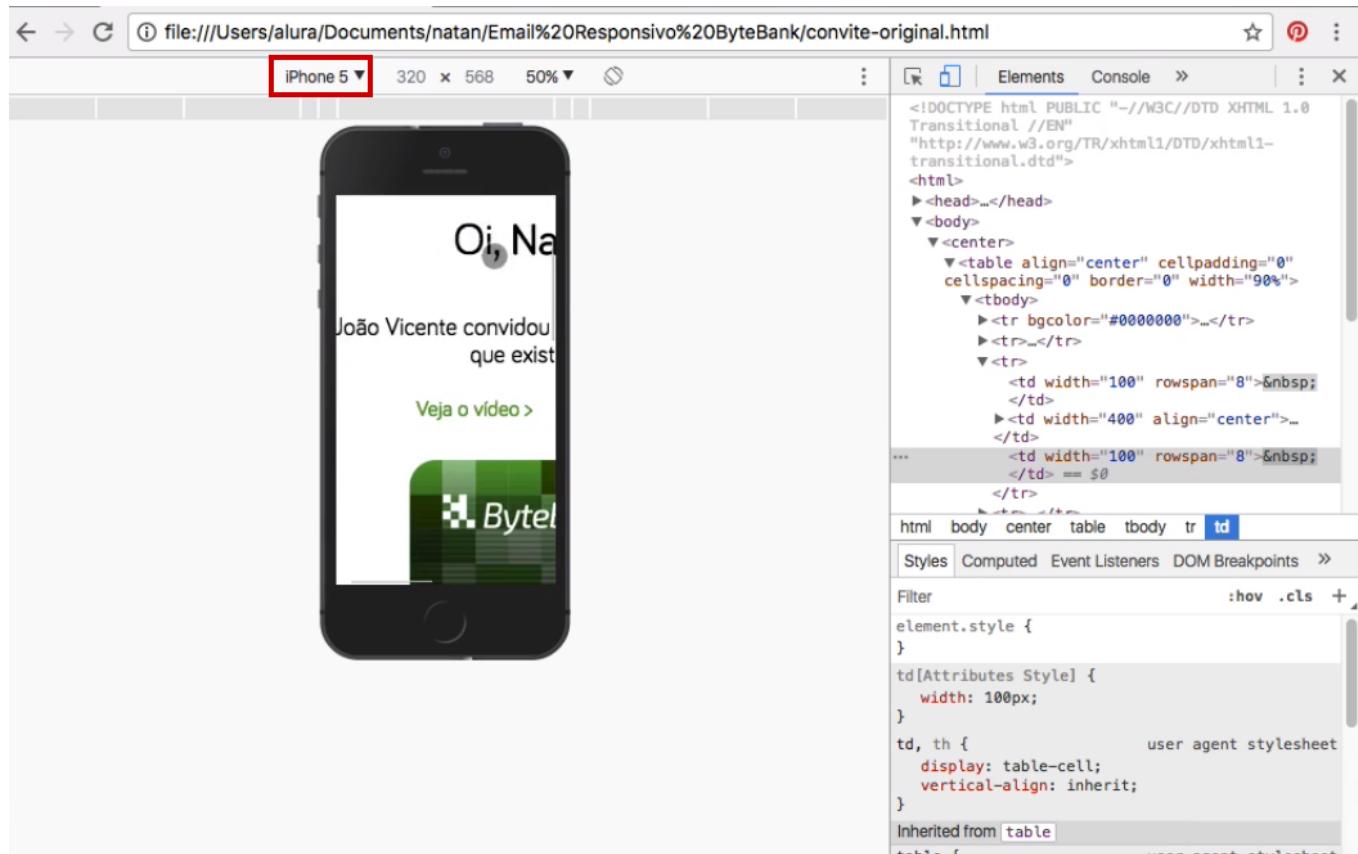
    <tbody>
      <tr bgcolor="#00000000">
        <td colspan="3" align="center" style="padding: 5px;">
          
        </td>
      </tr>
    <!-- . . . -->
  </center>
```

A partir desta alteração, o e-mail ficará da seguinte maneira:



Porém, apenas esta alteração não faz com que o e-mail seja responsivo e, se ele for aberto no celular, os textos ainda estarão pequenos. Apesar de ser uma newsletter simples, com uma coluna, as vantagens ainda ficarão lado a lado na parte inferior.

A seguir, abriremos a opção de **Inspeccionar** do navegador e acessaremos o "DevTools", que nos oferece a opção de visualizarmos o e-mail em diferentes dispositivos.



Se selecionarmos o iPhone 5, por exemplo, a fonte aparece legível, no entanto o usuário precisa rolar a tela para visualizar a mensagem. De qualquer forma, as vantagens oferecidas pela empresa se posicionará uma embaixo da outra. Da maneira como está atualmente, isto ainda prejudica bastante a experiência do usuário.

Ao alterarmos a largura do e-mail, o nome do cliente acabou se alinhando para fora dos limites da imagem do cartão.

Bytebank

Oi, Natan Souza

João Vicente convidou você para o banco mais digital que existe na atualidade!

[Veja o vídeo >](#) [Veja a apresentação >](#)

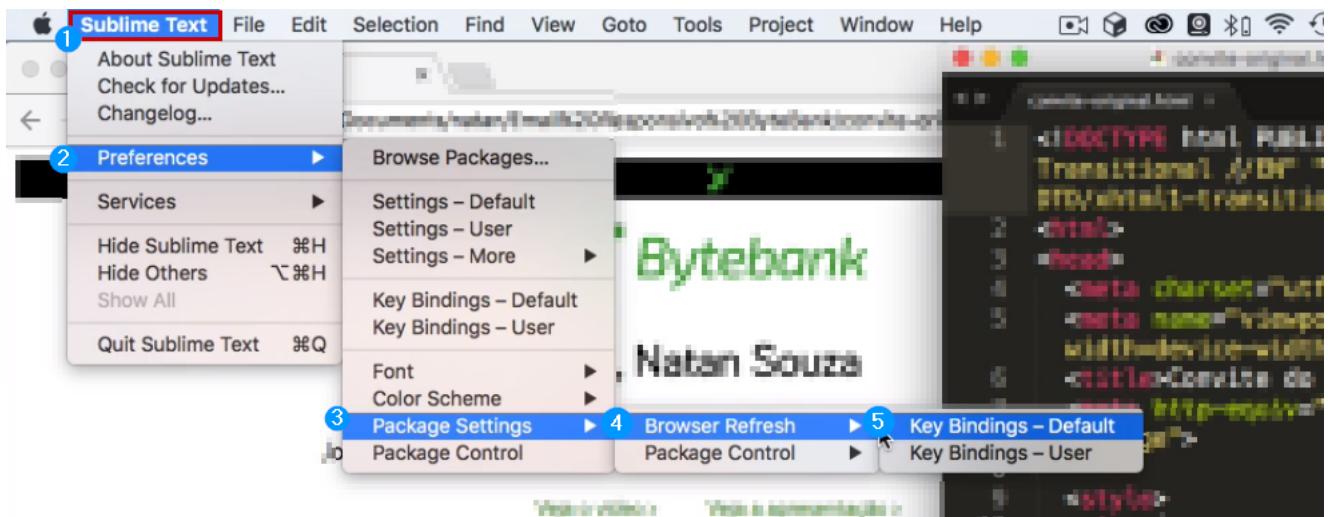
NATAN SOUZA

ACEITAR CONVITE

Poderíamos resolver esse problema com um pequeno truque, no caso, definindo um `padding` para os dois lados. Porém devemos evitar ao máximo o uso de CSS!

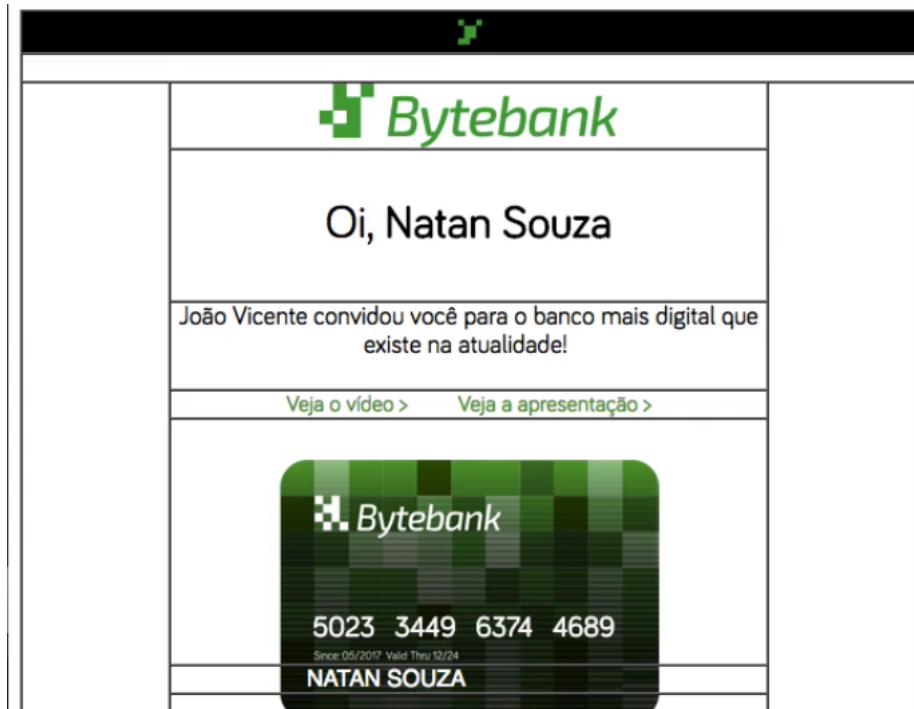
Até agora, para visualizarmos as alterações feitas no e-mail, precisamos atualizar constantemente o navegador, a menos que você tenha um *Browser sync*. Se você estiver utilizando o Dreamweaver, existe o modo "Live", em que aquilo que codificamos vai sendo replicado na tela automaticamente.

Caso esteja usando o Sublime, existe um plugin que poderá ser baixado se você já tiver o *Package Control*. Para acessar o plugin **Browser refresh**, clicaremos em "Sublime Text" no menu superior, e depois selecionaremos "Preferences > Package Settings > Browser Refresh > Key Bindings".

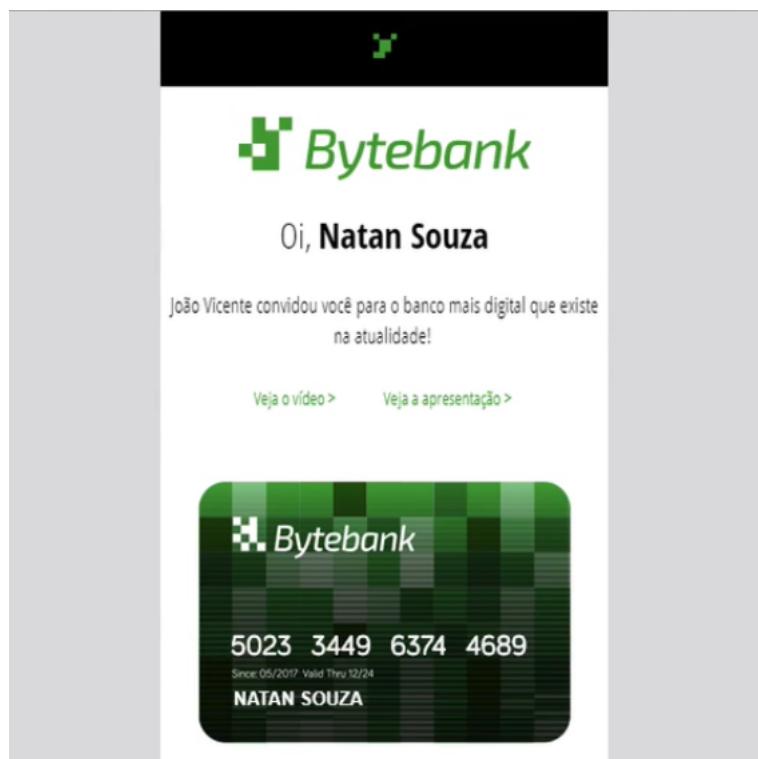


Com o plugin configurado, quando fizermos qualquer modificação no código e salvarmos o arquivo com "Control + Shift + R", a página será alterada automaticamente no código. Desta forma, ganhamos tempo no nosso trabalho. Se você tiver instalado um *live preview*, conseguirá visualizar rapidamente as alterações.

Para a resolução dos problemas surgidos na página após a alteração da altura, é preciso entender como funcionam os espaçamentos. Adicionaremos o valor `1` para a borda, e desta forma visualizaremos a tabela novamente:



O trabalho do `padding` está sendo realizado pelas tags `td`s, referentes a cada coluna. O espaçamento da altura foi definido com o atributo `rowspan`, que acopla diversas células e cria uma *single column* (coluna única). Queremos um design responsivo, e se analisarmos a versão final do e-mail, não precisamos de um espaço vazio e grande nas laterais.



À medida em que reduzimos o tamanho da tela, queremos que o e-mail marketing também se reduza proporcionalmente. Sendo assim, esta `td` não é necessária.

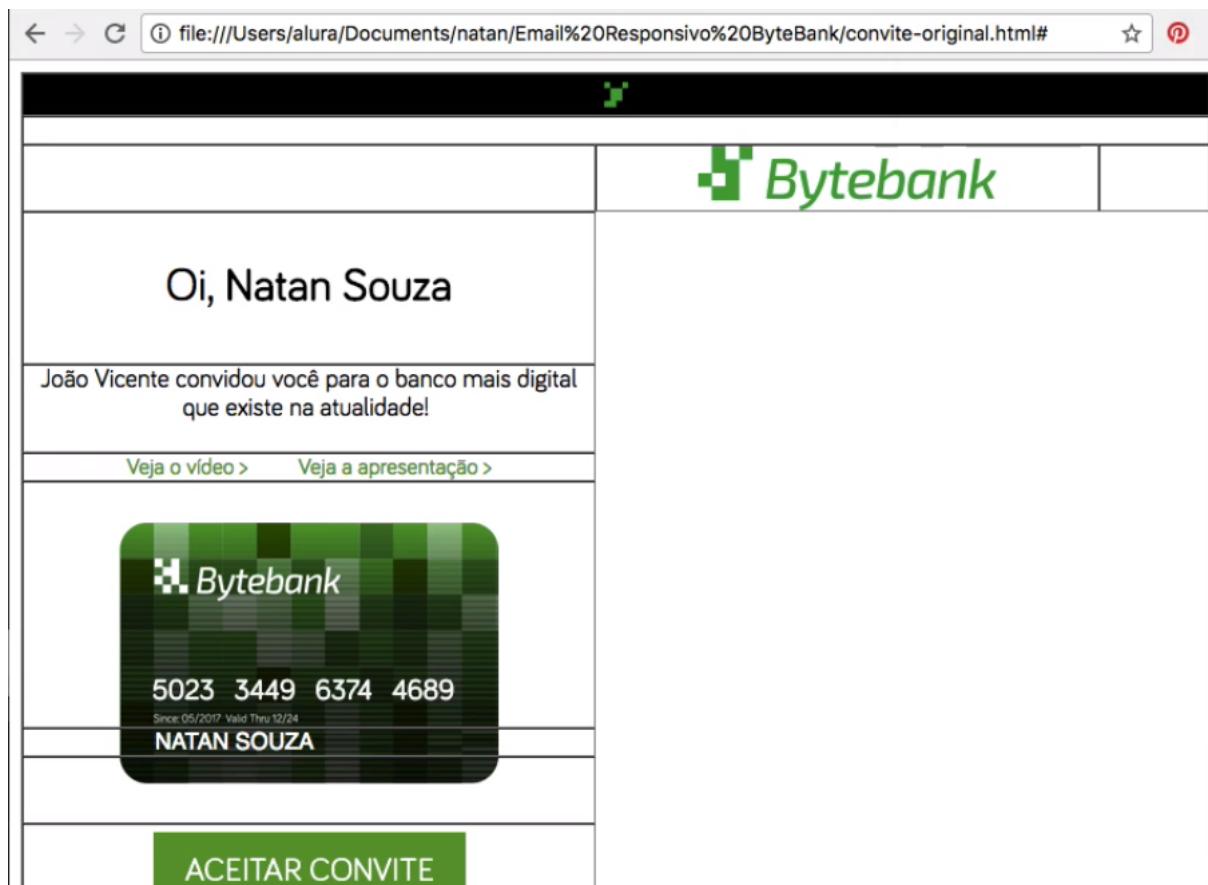
Como melhoraremos o respiro interno em um elemento? Podemos adicionar o `padding`, mas ele não poderá ser incluído como um atributo do HTML por ser, na verdade, uma propriedade do CSS. Tanto o `rowspan` como `colspan` (relacionado com as colunas), não são recomendados na criação de e-mail marketing responsivo.

Então, apagaremos todas as referências do `rowspan` no código, incluindo seus valores, como no trecho abaixo:

```
<tr>
  <td width="100">&ampnbsp</td>
  <td width="400" align="center">
    
  </td>
  <td width="100">&ampnbsp</td>
</tr>
```

Com o atalho "command + D", pode-se selecionar todos os trechos iguais em um código. Podemos utilizar esse comando para selecionar e apagar todos os `rowspan` de uma vez.

No browser, veremos o resultado das alterações.



O e-mail ficou "quebrado". Apesar de sabermos que não é bom usar os atributos `rowspan` e `colspan`, quando aquele foi removido, o código deixou de funcionar corretamente. Sua estrutura foi pensada para um e-mail com tamanho fixo, então teríamos que refazê-lo praticamente do zero.

Para não perdemos os textos e nomes de imagens, vamos apenas deixar comentado o trecho da tag `<center>`, localizado abaixo da abertura do `<body>` até seu fechamento, tomando cuidado para remover os comentários que estão dentro deste código.

```
<!-- <center>
<table align="center" cellpadding="0" cellspacing="0" border="0" width="90%">
  <tbody>
    <tr bgcolor="#00000000">
```

```

        <td colspan="3" align="center" style="padding: 5px;">
            
        </td>
    </tr>
<!--...-->
</center> -->

```

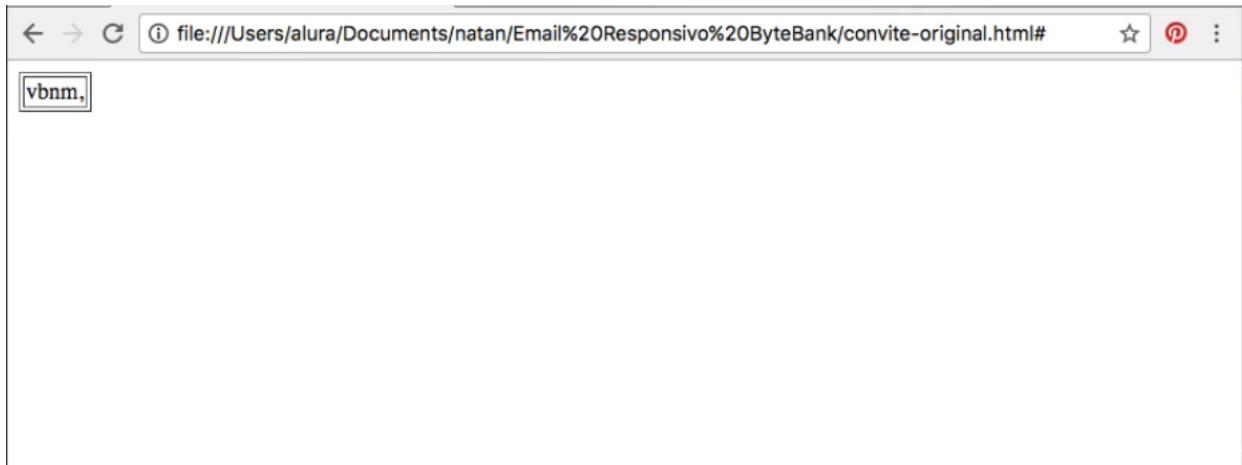
Ficaremos com a página em branco no browser. A seguir, começaremos a construir o e-mail responsivo, abaixo da tag `body`, em que adicionaremos uma nova tabela:

```

<body>
    <table border="1">
        <tr>
            <td>
                vbnm,
            </td>
        </tr>
    </table>

```

Para visualizarmos a borda, incluiremos o `border` com valor `1`, e conseguiremos perceber a existência da tabela no navegador.



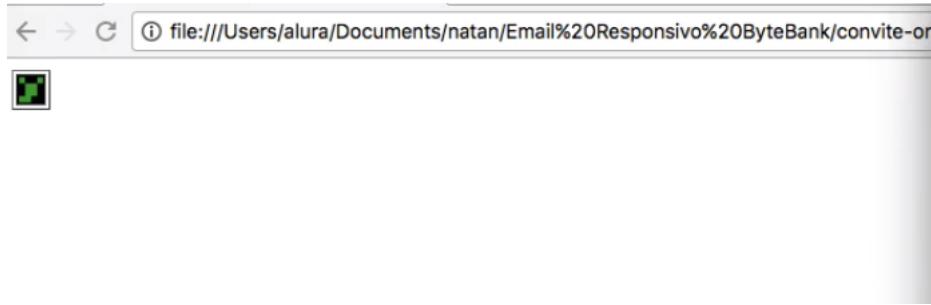
Em seguida, vamos conferir o layout da versão final. Incluiremos uma linha preta, na primeira linha, adicionando o `bgcolor` na `<tr>`, enquanto na `<td>` guardaremos uma imagem.

```

<body>
    <table border="1">
        <tr bgcolor="#000000">
            <td>
                
            </td>
        </tr>
    </table>
<!--...-->
</body>

```

No browser, veremos um pequeno quadrado com a imagem dentro.



O próximo passo será adicionar uma largura (`width`) de 90% .

```
<table border="1" width="90%">
  <tr bgcolor="#000000">
    <td>
      
    </td>
  </tr>
</table>
```



Observe que no caminho da imagem, incluímos a pasta `images` e puxamos o arquivo `Top_Logo.png` . Temos também o atributo `alt` , e sobre o qual nos aprofundaremos mais a seguir.

E como colocaremos a linha no centro da página? Eu adicionaria uma classe no estilo do arquivo, por exemplo, `.top-logo` :

```
<style>
  img {
    display: block;
  }

  .top-logo {
    margin: 0 auto;
    width: 10px;
  }
</style>
```

Em seguida, adicionaria o `class` no `img` .

```
<table border="1" width="90%">
  <tr bgcolor="#000000">
    <td>
      
    </td>
  </tr>
</table>
```

Porém, teríamos um problema. Será que o Gmail dá suporte para `margin` ou para `auto`? Não sabemos. Descobriremos acessando o site [Campaign monitor](https://www.campaignmonitor.com/css/) (<https://www.campaignmonitor.com/css/>), mais especificamente, a parte sobre suporte para CSS. Nele, encontraremos um guia que nos indica o que funciona ou não. Buscaremos nele, quem nos oferece suporte para margem:

The screenshot shows a table from campaignmonitor.com/css/ comparing CSS properties across different email clients. The columns represent Outlook 2007/10/13+, Outlook 03/Express/Mail, iPhone/iOS 7/iPad, Outlook.com, Apple Mail 6.5, Yahoo! Mail, Google Gmail, and Android 4 (Gmail). The rows list properties like border, border-color, border-image, border-radius, box-shadow, height, margin, padding, width, max-width, and min-width. Margins are highlighted in blue. The 'margin' row shows varying support: Outlook clients support it, while Outlook.com does not (indicated by a red 'x'). Other clients like Apple Mail, Google Gmail, and most mobile clients support it.

Style Element	Outlook 2007/10/13 +	Outlook 03/Express/Mail	iPhone/iOS 7/iPad	Outlook.com	Apple Mail 6.5	Yahoo! Mail	Google Gmail	Android 4 (Gmail) +
border	✓	✓	✓	✓	✓	✓	✓	✓
border-color	✗	✗	✓	✓	✓	✓	✓	✗
border-image	✗	✗	✓	✗	✓	✗	✗	✗
border-radius	✗	✗	✓	✓	✓	✗	✓	✗
box-shadow	✗	✗	✓	✓	✓	✗	✗	✗
height	✗	✓	✓	✓	✓	✓	✓	✓
margin	✓	✓	✓	✗	✓	✓	✓	✓
padding	i	✓	✓	✓	✓	✓	✓	✓
width	i	✓	✓	✓	✓	✓	✓	✓
max-width	✗	✓	✓	✓	✓	✓	✓	✓
min-width	✗	✓	✓	✓	✓	✓	✓	✓

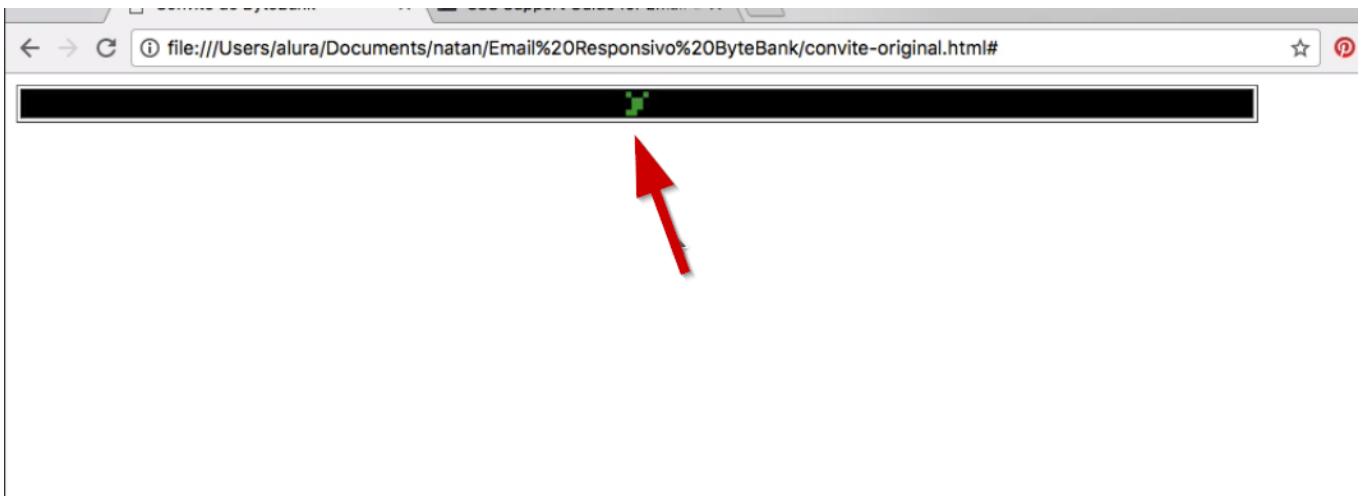
Vemos que o Outlook.com não dá suporte para margem, se queremos dar suporte para esse email client, não usaremos o `margin`. Porém, sem margem, nosso elemento nunca poderá ser centralizado desta forma. Temos que encontrar uma maneira de dependermos menos do CSS, se queremos centralizar o conteúdo da `<td>`, usaremos o atributo `align="center"`.

```

<table border="1" width="90%">
  <tr bgcolor="#000000">
    <td align="center">
      
    </td>
  </tr>
</table>

```

Agora, a imagem ficará centralizada na linha.



Conseguimos!