

03

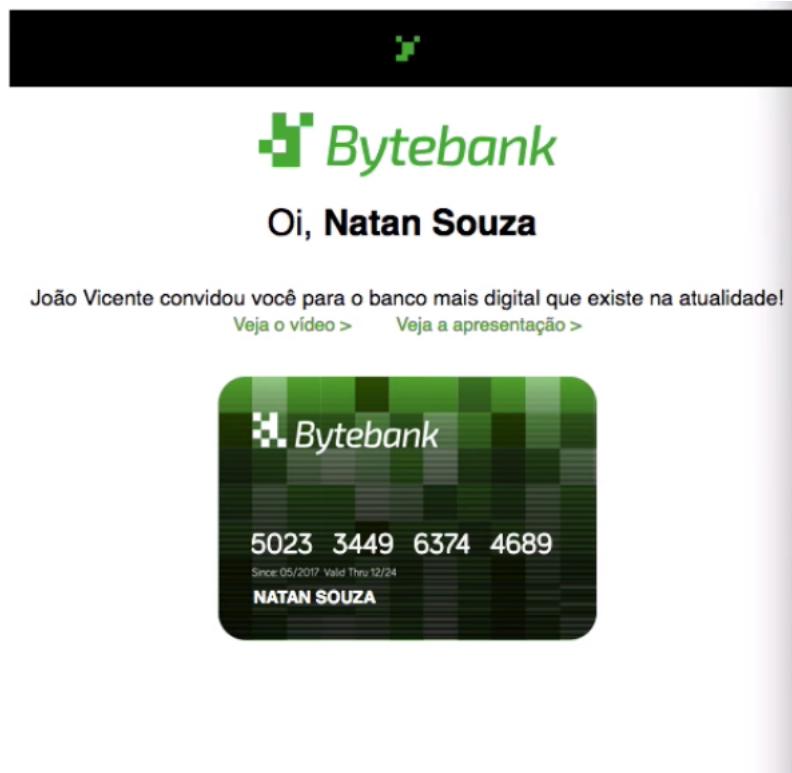
Overview

Transcrição

Uma curiosidade em nosso código ao adicionarmos o valor `1` para o atributo `border`, na tag `<table>`:

```
<body>
  <table border="1" width="90%" cellspacing="0" cellpadding="0">
    <tr bgcolor="#000000">
      <td align="center" class="cabecalho-logo">
        
      </td>
    </tr>
  </table>
```

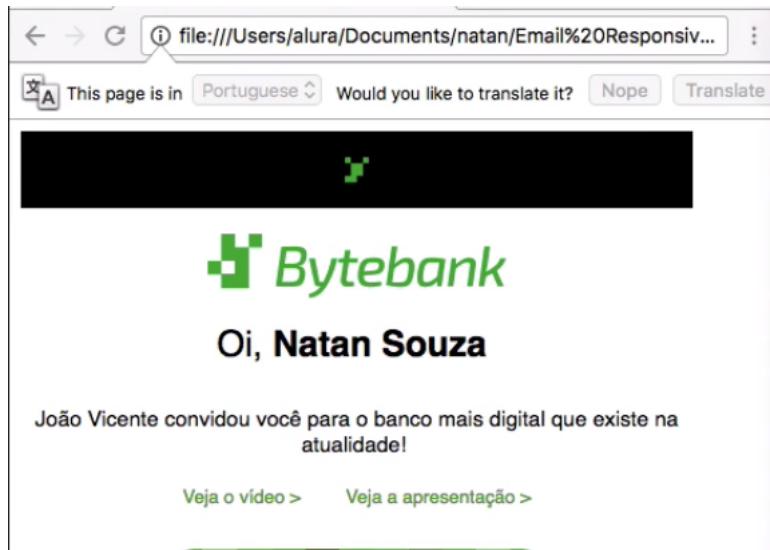
Colocaremos o valor para conseguirmos enxergar os limites, caso ele seja removido, ficará da seguinte maneira:



Sem as bordas, é notável como o texto do convite fica bem próximo dos links. Vamos incluir o espaço com a tag `
`.

```
<tr>
  <td align="center">
    <span style="font-size: 15px; font-family: sans-serif">
      João Vicente convidou você para o banco mais digital que existe na atualidade!
      <br><br>
      </span>
    </td>
  </tr>
```

Desta forma, o texto ganha espaçamento em relação aos links:



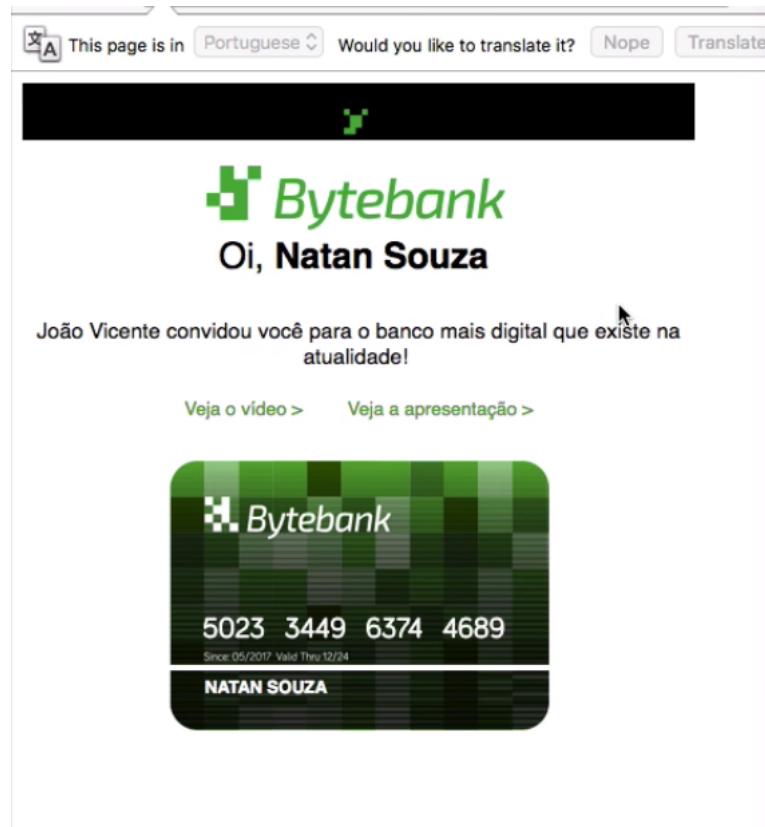
Dentre os pré-requisitos deste curso, há [E-mail Marketing](https://www.alura.com.br/curso-online-e-mail-marketing) (<https://www.alura.com.br/curso-online-e-mail-marketing>), de que vimos a parte introdutória e entendemos o porquê de usarmos o `display: block` na tag `<style>` no HTML do convite inicial.

```
<style>
  img {
    display: block;
  }
  .link-veja {
    font-family: sans-serif;
    color: #47a138;
    font-size: 13px
  }
</style>
```

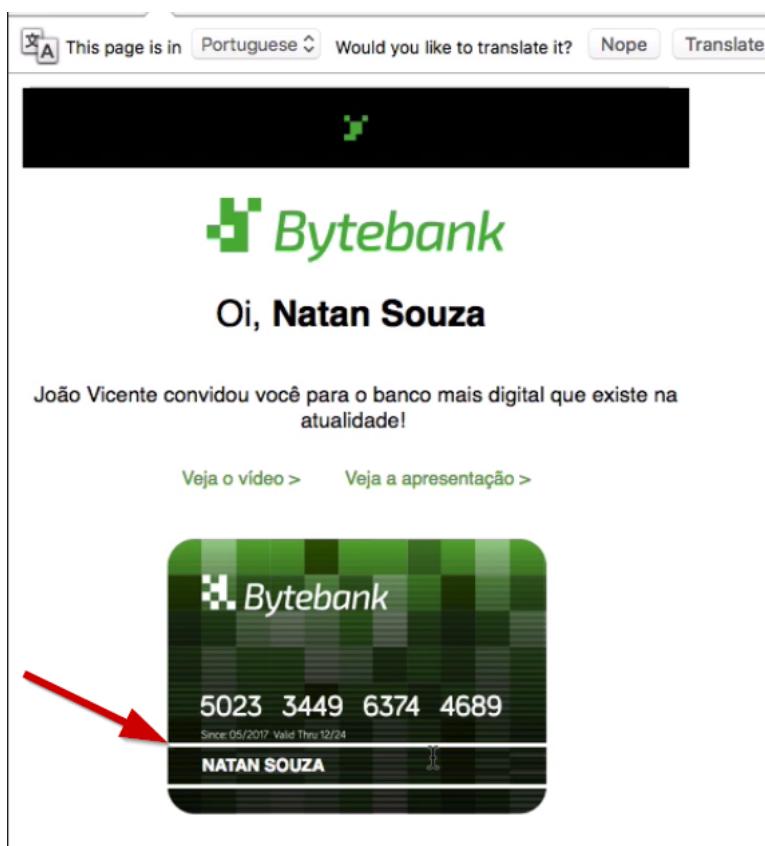
Para entendermos melhor o que ele faz, vamos comentar a linha do `display`:

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

Muitos navegadores e e-mail clients colocam imagens ao redor das imagens, então, se não usarmos `display: block`, a imagem ficará quebrada, como podemos ver abaixo:



Obteremos um problema semelhante se não adicionarmos o `cellspacing` :



Isto acontece pois a célula tem um respiro interno e um espaço acima. Quando adicionamos o `cellspacing`, "resetamos" esse padrão.

O que vimos até aqui? Falamos como é relevante ignorarmos algumas boas práticas, apesar de doer meu coração dizê-lo. Isto porque queremos dar suporte para usuários que estão usando emails clients muito antigos, então um código com

"gambiarras" acabará sendo uma solução. Para sabermos quais propriedades funcionam, poderemos consultar no [Campaign Monitor](https://www.campaignmonitor.com/css/) (<https://www.campaignmonitor.com/css/>).

O próprio MailChimp tem uma lista com as tags e propriedades que suporta. Descobriremos que nosso código HTML já terá problemas com Gmail no futuro.

The screenshot shows a section titled "Email Client CSS Support" under "Resources". Below it is a heading "`<style>` & `<link>`". There are three tabs: "Browser" (selected), "Desktop", and "Mobile". A table follows, comparing four email clients: Outlook.com, Yahoo!, Gmail, and AOL. The table rows represent different placement of `<style>` and `<link>` elements. Green checkmarks indicate support, while red X's indicate non-support.

	Outlook.com	Yahoo!	Gmail	AOL
<code><style></code> within <code><head></code>	✓	✓	✓	✓
<code><style></code> within <code><body></code>	✓	✓	✗	✓
<code><link></code> within <code><head></code>	✗	✗	✗	✗
<code><link></code> within <code><body></code>	✗	✗	✗	✗

No entanto, seguiremos a filosofia "problemas posteriores serão resolvidos posteriormente".

Vimos que algumas propriedades funcionam e outras não, mostramos onde buscar referência e como pesquisar informações. Vimos também que criar layouts com apenas uma coluna facilita bastante nosso trabalho. É possível fazê-lo no formato **multicolumna**, porém, para o e-mail marketing responsivo, é mais tranquilo trabalharmos da forma mais simples possível.

Mais adiante trabalharemos com as outras partes do layout, começando pelo botão "Aceitar convite".