



8 STEPS TO CODE KILLER **RESPONSIVE** **EMAILS**

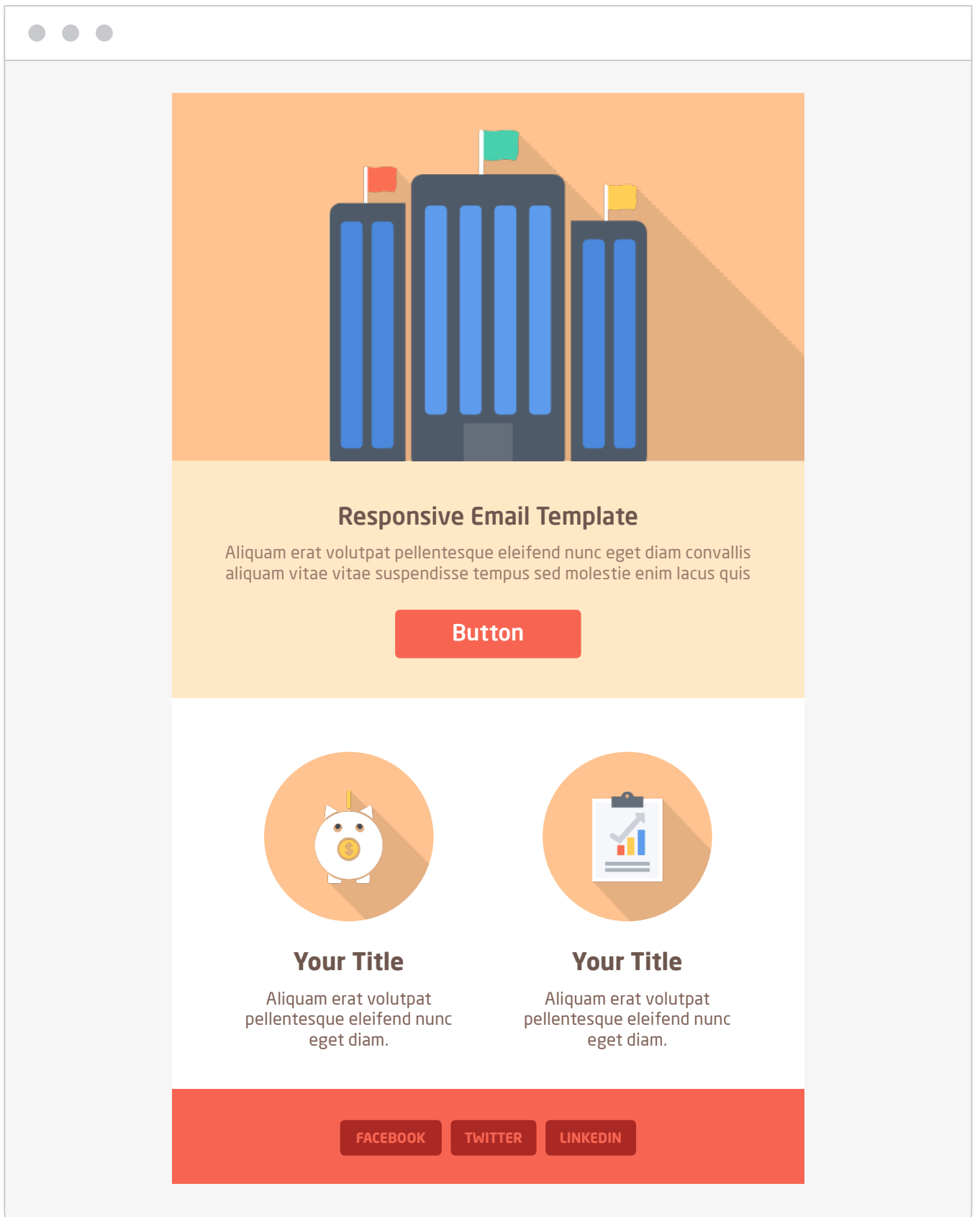
THAT WILL MAKE YOUR
EMAILS BEAUTIFUL

BUILD RESPONSIVE EMAIL **STEP BY STEP**

Steps to create a simple responsive email template. (fluid image, main content, two columns section and footer)

Learn to create flawless emails step by step with the proper use of HTML, CSS, and Media Queries.

But this is only possible if you keep attention to details. One simple line in your code can sabotage all your efforts in attain good results across all email clients. To help you deal with all the frustration that comes with coding HTML email we've created this guide with simple steps to follow when coding email design.



1 - DOCUMENT SET UP

For every email template that you code, you'll need to set up your document base structure the same way. Starting with a Doctype will inform the program what type of document it is and what HTML tags, set of rules and CSS to expect. Usually is used to distinguish versions of HTML languages, such as HTML or XHTML. If you're going to use Media Queries, you'll need to include the viewport meta tag on your email's <head>. This will tell the browser to set the viewable area to the width of the device's screen. After the <head> tag, you'll need a body, where you'll place your content.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-  
transitional.dtd">  
<html xmlns="http://www.w3.org/1999/xhtml">  
  <head>  
    <meta charset="utf-8">  
    <meta name="viewport" content="width=device-width">  
  </head>  
  <body>  
    <!-- HTML Content Here -->  
  </body>  
</html>
```

2 - CSS RESETS

As email clients render your email in different ways, there is some hacks that can help to prevent some of them. You can reset some general styles and reset some client specific styles. You should place this resets in the end of your <body> instead of putting it in the <head> because some email clients strip it out of the email.

```
(...)
</body>
<style type="text/css">
  /* GENERAL STYLE RESETS */
  body #bodyTable {height: 100% !important; width: 100% !important; margin: 0; padding: 0; -webkit-
font-smoothing: antialiased;}
  img, a img {border: 0; height: auto; line-height: 100%; outline: none; text-decoration: none;} /* Ensure
that images don't have borders, text-decoration and unwanted spacings */
  table, td {border-collapse: collapse !important;} /* Ensure that tables and cells don't have any extra
spacing by specifying that borders should be collapsed*/
  /* CLIENT SPECIFIC RESETS */
  .ReadMsgBody {width:100%;} .ExternalClass{width:100%;} /* Force Hotmail to display emails at full
width by targeting classes that the service adds */
  .ExternalClass, .ExternalClass p, .ExternalClass span, .ExternalClass font, .ExternalClass td,
.ExternalClass div {line-height: 100%;} /* Force Hotmail to display normal line spacing by targeting classes
that the service adds*/
  #outlook a {padding:0;} /* Force Outlook to provide a "view in browser" message */
  body, table, td, a {-webkit-text-size-adjust:100%; -ms-text-size-adjust:100%;} /* Prevent WebKit and
Windows mobile changing default text sizes */
  table, td {mso-table-lspace:0pt; mso-table-rspace:0pt;} /* Remove spacing between tables in Outlook
2007 and up */
  img {-ms-interpolation-mode:bicubic;} /* Allow smoother rendering of resized image in Internet
Explorer */
  /* iOS BLUE LINKS */
  .apple-link a { color:#999999 !important; text-decoration: none !important; } /* Handling iOS blue links
*/
</style>
</html>
```

The first line on General Style Resets has the purpose to avoid that email clients add padding or spaces on your email design. As some email clients strip the `<body>` you'll need to create a class for your body table (step 3) and call it on the general style resets.

```
(...)  
<body>  
  <table class="bodyTable" width="100%" border="0" cellspacing="0" cellpadding="0" valign="top"  
  bgcolor="#f4f4f4">  
    (...)  
  </table>  
</body>  
(...)
```

3 - BODY & MAIN TABLE

To start, create a table 100% wide that works as the body to some email clients that strip out the style from `<body>` tag. It's important to set all table's cellpadding and cellspacing to '0' to avoid unwanted spaces. Also, unless you want your tables to have a 1px black border, don't forget to set it to '0'. The attribute valign will help to ensure the content inside this table will float to the top of the email.

```
(...)  
<body>  
  <table class="bodyTable" width="100%" border="0" cellspacing="0" cellpadding="0" valign="top"  
  bgcolor="#f4f4f4">  
    <tr>  
      <td align="center"></td>  
    </tr>  
  </table>  
</body>  
(...)
```

Due to all different supports given by email clients, building a responsive email it's not an easy task. To make a fluid email you need to create it out of percentages by setting max-widths or fixed-widths that turn flexible with media queries. Unfortunately both methods don't have full support. For instance max-width is not supported by Outlook 2007/10/13, Lotus Notes, AOL Desktop and Windows Mobile. On the other hand, media queries aren't supported by Gmail (both browser and mobile app), Outlook (browser and desktop) Windows Live Mail and AOL Mail. So we need to combine techniques in order to make our email responsive in most inboxes.

To get around the different support of email clients, we're going to base our structure both on percentages, fixed-widths and max-widths. Inside your body table you're going to place three cells in a row. Using the HTML attribute, set the width for the center TD and leave the other two cells empty. The center TD is where you'll put your email structure while the other two td cells have the purpose to give space around your email content. Email clients tend to ignore empty cells, so you'll need to place a " " which is used to represent a non-breaking space.

```
(...)  
<body>  
  <table class="bodyTable" width="100%" border="0" cellspacing="0" cellpadding="0" valign="top"  
  bgcolor="#f4f4f4">  
    <tr>  
      <td align="&nbsp;"></td>  
      <td width="600" align="center" >!-- Tables Content Here --></td>  
      <td align="&nbsp;"></td>  
    </tr>  
  </table>  
</body>  
(...)
```


4 - FLUID IMAGE & MAIN CONTENT

Inside the main table, on the 600 wide td, we're going to add our first table which will allocate a fluid image. Set its width to 100% and align it to center. Style the table with a "max-width" of 600. This is needed to some email clients that doesn't respect the td width such as Apple Mail. To make your image fluid, give it's <td> a fix width and a percentage width to the and set it's height to "auto" to allow the image to resize according to the browser's width.

Don't forget to give your image an "alt text" and to add the "display: block;" style command to avoid gaps around images.

```
(...)
<body>
  <table class="bodyTable" width="100%" border="0" cellspacing="0" cellpadding="0" valign="top"
  bgcolor="#f4f4f4">
    <tr>
      <td align="&nbsp;"></td>
      <td width="600" align="center" >
        <!-- ===== Fluid Image ===== -->
        <table width="100%" align="center" border="0" cellspacing="0" cellpadding="0" valign="top"
        bgcolor="#f4f4f4" style="max-width: 600px;" >
          <tr>
            <td width="600" align="center">
            </td>
          </tr>
        </table>
        <!-- ===== End Fluid Image ===== -->
      </td>
    </tr>
  </table>
  <td align="&nbsp;"></td>
</body>
(...)
```

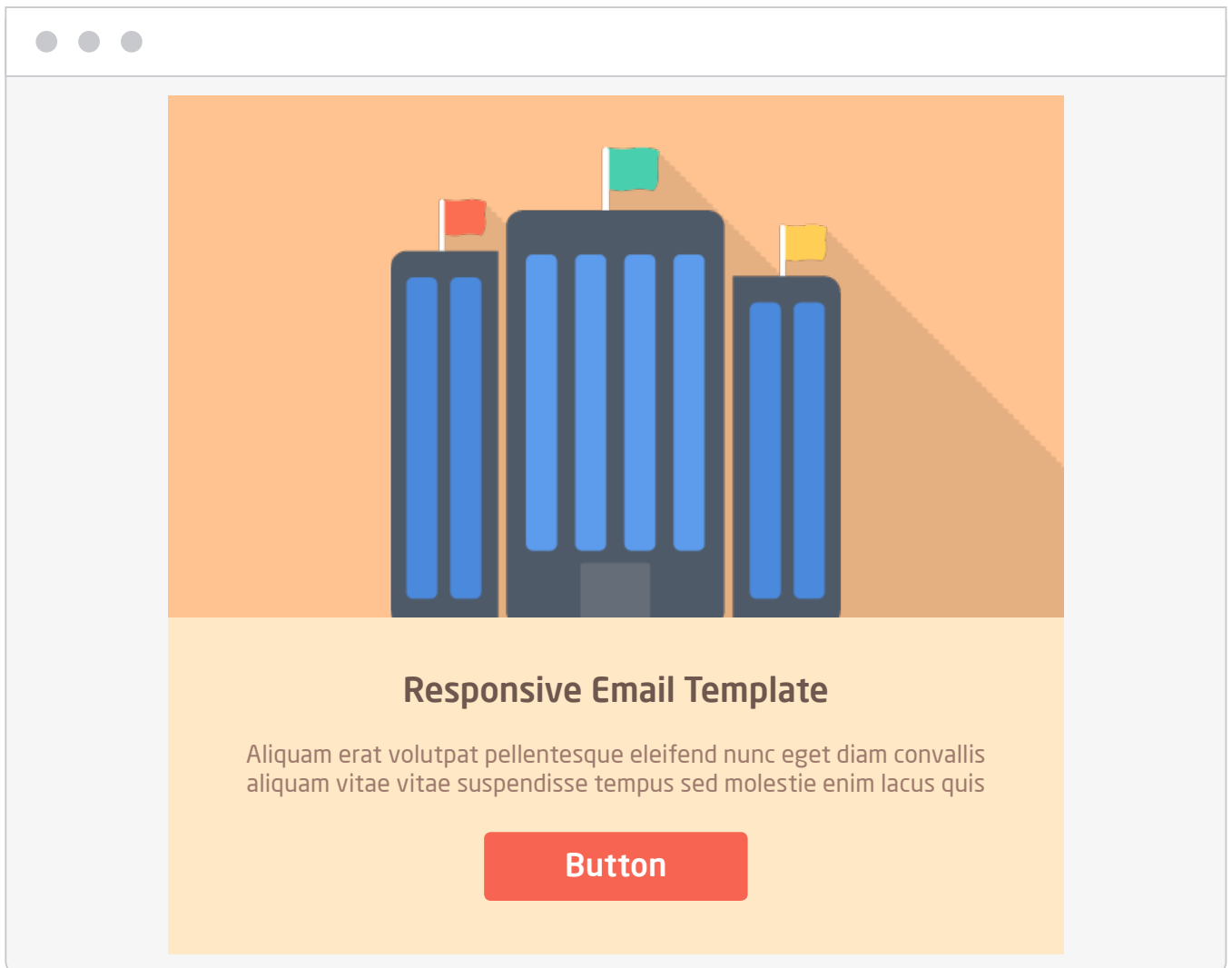
Our second table will contain three main rows - Title, Description and Button. Give empty space between the elements with empty rows and remember to always set the height on the <td> (you can also set spacing between content with padding on each <td> or for the whole table)

```

<!-- ===== End Fluid Image ===== -->
<!-- ===== Main Content ===== -->
<table width="100%" align="center" border="0" cellspacing="0" cellpadding="0" valign="top"
bgcolor="#ffe8c4" style="max-width: 600px;" >
  <tr>
    <td height="40"></td>
  </tr>
  <tr>
    <td align="center" style="padding: 0px 40px 0px 40px; font-family: Helvetica, arial, sans-serif; font-size:
20px; text-transform: uppercase; font-weight: bold; color: #6f574e;">Responsive Email Template</td>
  </tr>
  <tr>
    <td height="20"></td>
  </tr>
  <tr>
    <td align="center" style="padding: 0px 40px 0px 40px; font-size: 16px; font-family: Helvetica, arial, sans-
serif; text-align: center; line-height: 25px; color: #9f7b6d">Aliquam erat volutpat pellentesque eleifend nunc eget
diam convallis aliquam vitae vitae suspendisse tempus sed molestie enim lacus quis </td>
  </tr>
  <tr>
    <td height="20"></td>
  </tr>
  <tr>
    <td align="center">
      <a href="#" style="text-decoration: none; display: inline-block; padding: 13px 50px 13px 50px; font-
size: 20px; font-family: Helvetica, arial, sans-serif; line-height: 20px; border: none; background-color: #f76552;
color: #ffffff; border-radius: 6px; -webkit-border-radius: 6px; -moz-border-radius: 6px;"> Button </a>
    </td>
  </tr>
  <tr>
    <td height="40"></td>
  </tr>
</table>
<!-- ===== End Main Content ===== -->

```

(...)

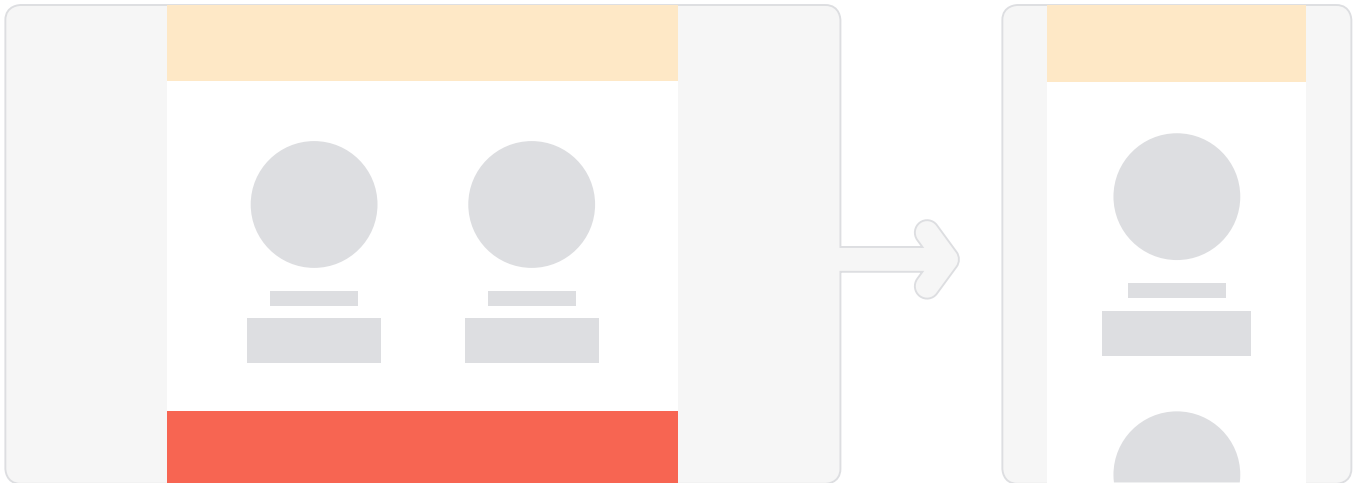


5 - 2 COLUMNS SECTION

To build this section, we're going to build two columns side by side, that when in mobile devices will become stacked columns without needing media queries. Each one will have an image, title and text.

Start by placing two 50% wide tables inside a 100% wide table.

For this technic to work on some email clients, you'll need to had some fixes to the <td> of your wrapper table and had some conditional code that you can see here: <https://www.campaignmonitor.com/blog/post/4240/creating-a-centred-responsive-design-without-media-queries>



Set each table width to 300px on inline style. Add also "inline-block" to allow the columns to become stacked depending on the browsers window width.

```

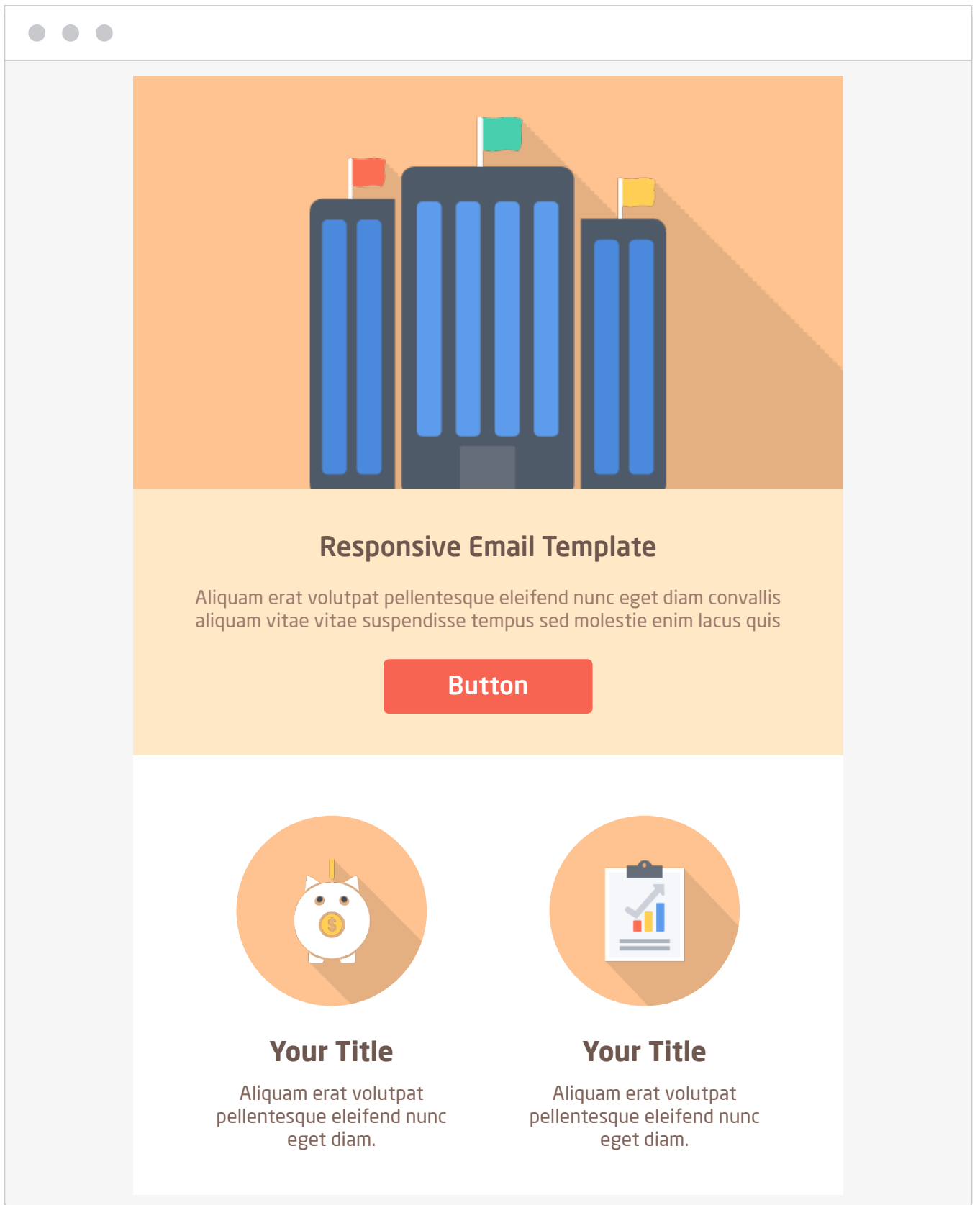
<!-- ===== End Main Content ===== -->
<!-- ===== 2 Columns Section ===== -->
<table width="100%" align="center" border="0" cellspacing="0" cellpadding="0" valign="top"
bgcolor="#f0ecec" style="max-width: 600px;" >
  <tr>
    <td align="center" style="text-align: center; vertical-align: top; font-size: 0;">
      <table align="center" width="50%" border="0" cellspacing="0" cellpadding="0" valign="top"
style="width: 300px; display: inline-block;">
        <tr>
          <td></td>
        </tr>
      </table>
      <table align="center" width="50%" border="0" cellspacing="0" cellpadding="0" valign="top"
style="width: 300px; display: inline-block;">
        <tr>
          <td></td>
        </tr>
      </table>
    </td>
  </tr>
</table>
<!-- ===== End 2 Columns Section ===== -->

```

(...)

Inside each 50% wide table, place some rows to allocate your content:

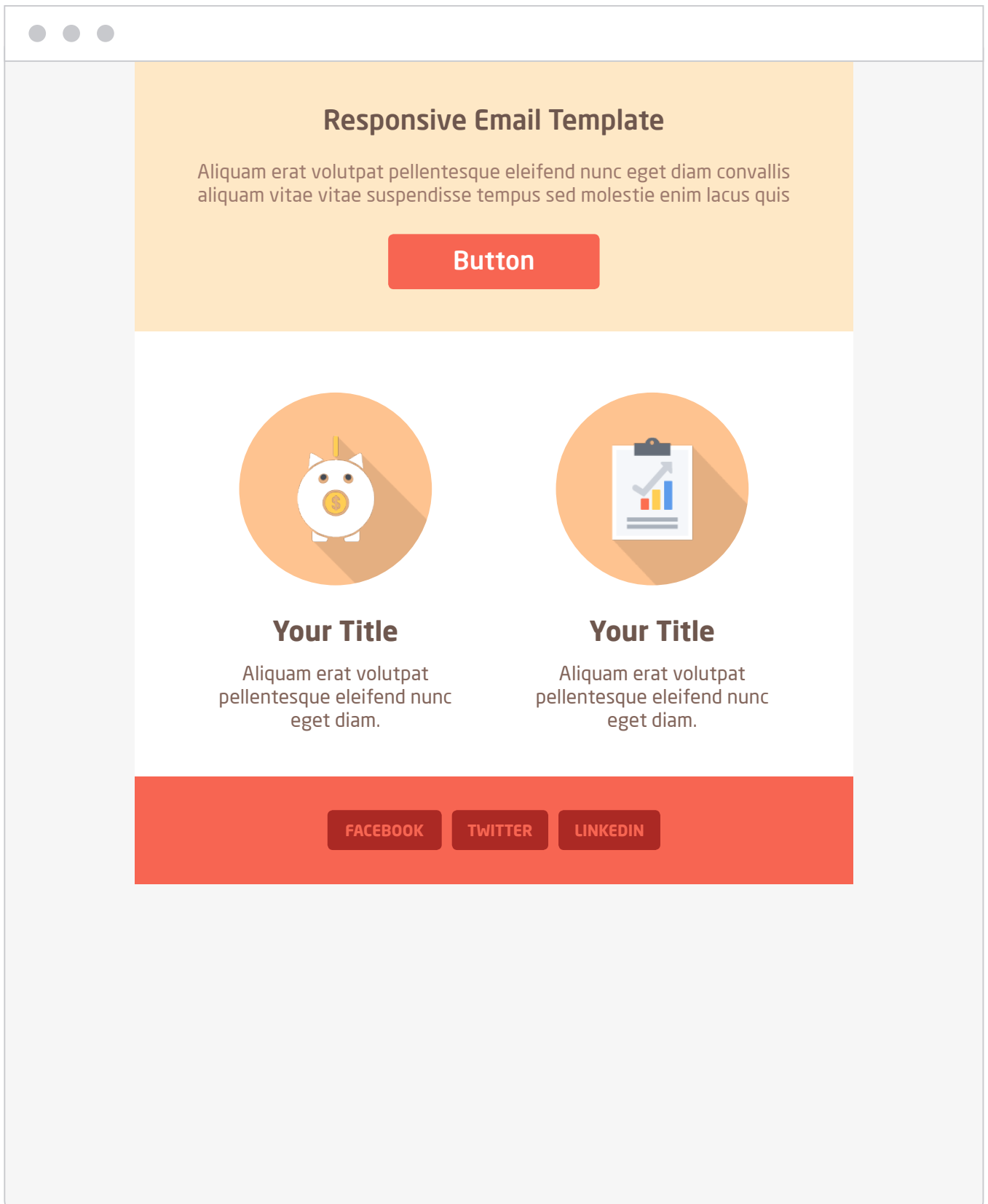
```
(...)
    <table align="center" width="50%" border="0" cellspacing="0" cellpadding="0" valign="top"
style="width: 300px; display: inline-block;">
        <tr>
            <td height="40"></td>
        </tr>
        <tr>
            <td align="center"></td>
        </tr>
        <tr>
            <td height="20"></td>
        </tr>
        <tr>
            <td align="center" color="#9f7b6d" style="padding: 0px 40px 0px 40px; font-size: 20px;
font-family: Helvetica, arial, sans-serif; text-align: center; font-weight: bold; color: #9f7b6d;"> Your Title</
td>
        </tr>
        <tr>
            <td align="center" color="#9f7b6d" style="padding: 0px 40px 0px 40px; font-size: 16px;
font-family: Helvetica, arial, sans-serif; text-align: center; line-height: 25px; color: #6f574e;"> Aliquam erat
volutpat pellentesque eleifend nunc eget diam. </td>
        </tr>
        <tr>
            <td height="30"></td>
        </tr>
    </table>
(...)
```



6 - FOOTER

Footer follows the same structure as shown before. Just Place a new table 100% wide, set it's max-width and place your content inside.

```
(...)
<!-- ===== 2 Columns Section ===== -->
<!-- ===== Footer ===== -->
<table width="100%" align="center" border="0" cellspacing="0" cellpadding="0" valign="top"
bgcolor="#f76552" style="max-width: 600px;" >
  <tr>
    <td height="20"></td>
  </tr>
  <tr>
    <td align="center">
      <!-- Button -->
      <a href="#" style="background-color:#c63623; color:#ff9a8d; text-decoration: none;
font-weight: bold; padding: 12px 10px 10px 10px; font-size: 12px; font-family: Helvetica, arial, sans-serif;
text-transform: uppercase; border-radius: 3px;">Facebook</a>
      <a href="#" style="background-color:#c63623; color:#ff9a8d; text-decoration: none;
font-weight: bold; padding: 12px 10px 10px 10px; font-size: 12px; font-family: Helvetica, arial, sans-serif;
text-transform: uppercase; border-radius: 3px;">Twitter</a>
      <a href="#" style="background-color:#c63623; color:#ff9a8d; text-decoration: none;
font-weight: bold; padding: 12px 10px 10px 10px; font-size: 12px; font-family: Helvetica, arial, sans-serif;
text-transform: uppercase; border-radius: 3px;">LinkedIn</a>
    </td>
  </tr>
  <tr>
    <td height="20"></td>
  </tr>
</table>
<!-- ===== End Footer ===== -->
(...)
```



7 - MAKE IT RESPONSIVE

The first step towards making your email responsive is to include the viewport meta tag on your <head>. This will tell the browser to set the email viewable area to the width of the device's screen. So if the screen's width is 480px, the browser window will be 480px, rather than showing your email content zoomed out to fit the screen.

```
(...)  
<head>  
  <meta charset="utf-8">  
  <meta name="viewport" content="width=device-width">  
</head>  
(...)
```

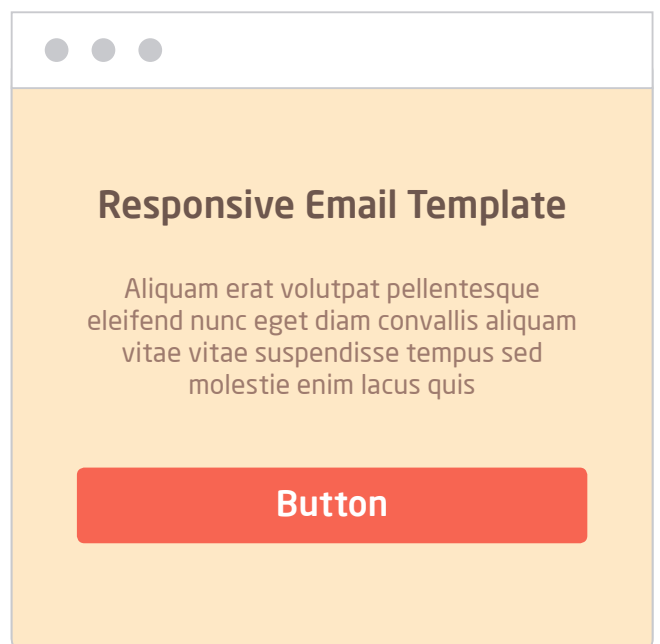
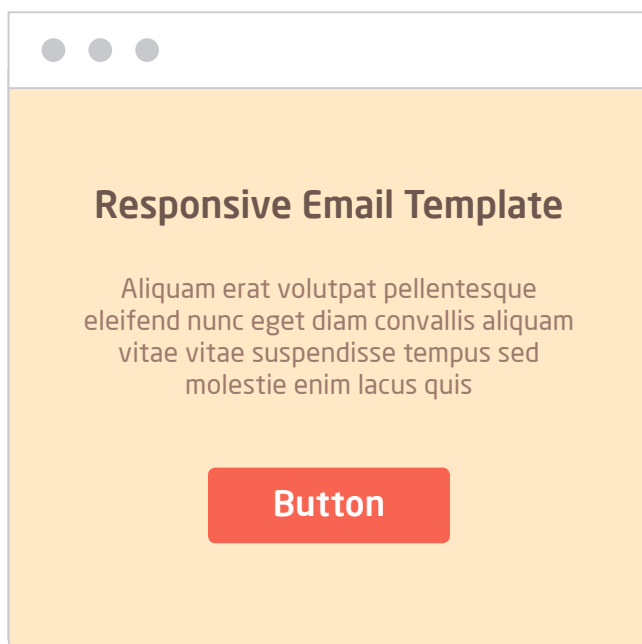
Now we can continue by adding our media queries which is a component of CSS. Place it on the email <style> tag next to the CSS Resets (step 2). Media queries will detect the screen size of the device where the email is being opened and then "turn on" different set of rules based on that screen size. There is no specific rule to determine breaking points. It depends on your individual email design. Just start developing for small screen size and as you resize your browser window, to make it bigger, see when your layout starts to look weird or unstructured. Regardless of the resolution this is where you should create your first breakpoint.

```
(...)  
</body>  
  <style type="text/css">  
    (...)  
    /*MOBILE STYLES */  
    @media screen and (max-width: 600px) {  
      .class {style rules}  
    }  
  </style>  
</html>  
(...)
```

Let's start to optimize our button to mobile devices. We need to make it wider so it becomes easier to tap. Add a class to your button and then place it on your media querie so you can make the nedded alterations.

```
(...)  
<tr>  
  <td align="center"><a class="mobile-button" href="#" style="text-decoration: none; display: inline-block; padding: 13px 50px 13px 50px; font-size: 20px; font-family: Helvetica, arial, sans-serif; line-height: 20px; border:none; background-color: #f76552; color: #ffffff;"> Button </a> </td>  
</tr>  
(...)
```

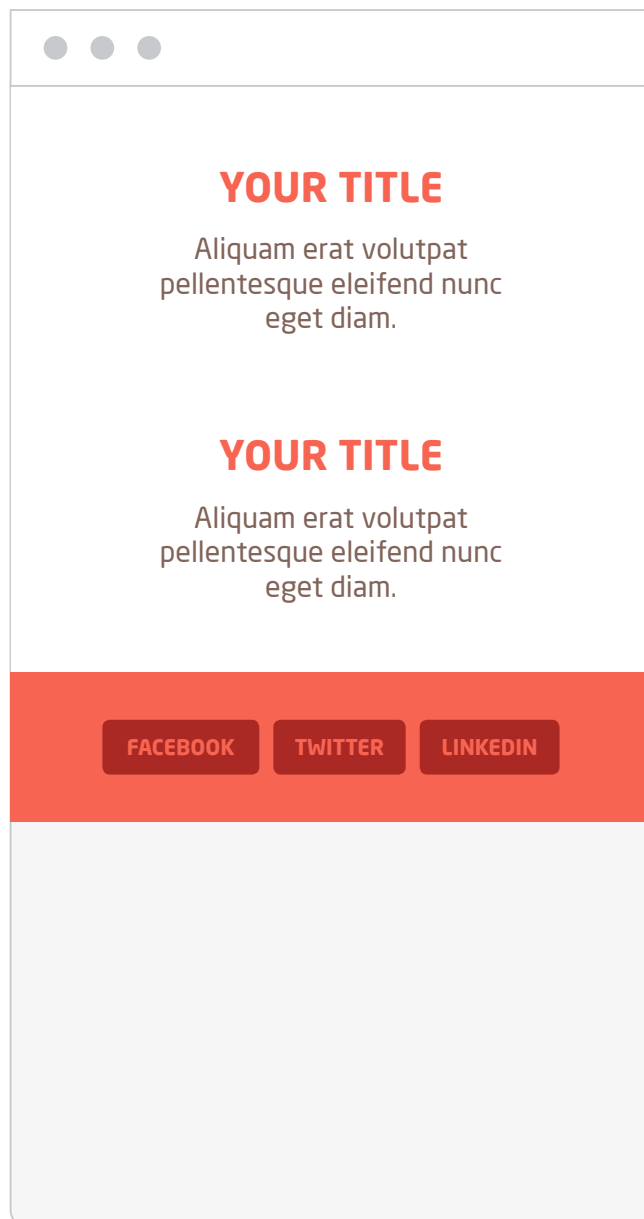
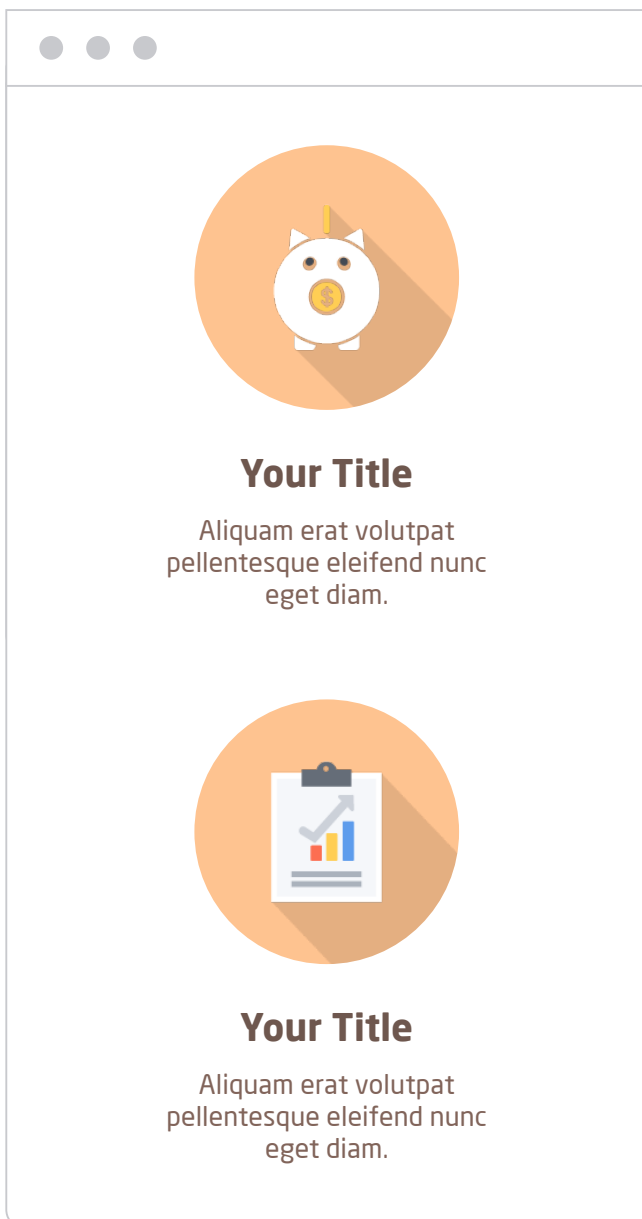
```
(...)  
@media screen and (max-width: 600px) {  
  a[class="mobile-button"] {  
    width: 60% !important;  
    display: block !important;  
  }  
}  
(...)
```



Now let's make some alterations on our main-section. Let's hide the images in mobile with "display:none" so that users don't have to scrol too much. Also let's make the text more proiminent by inscreasing the font-size and change it's color. Don't forget to add classes to the respective elements.

```
(...)
<tr>
  <td align="center" class="mobile-hide"></td>
</tr>
<tr>
  <td height="20"></td>
</tr>
<tr>
  <td align="center" class="mobile-text" color="#9f7b6d" style="padding: 0px 40px 0px
  40px; font-size: 20px; font-family: Helvetica, arial, sans-serif; text-align: center; font-weight: bold; color:
  #9f7b6d;"> Your Title</td>
</tr>
(...)
```

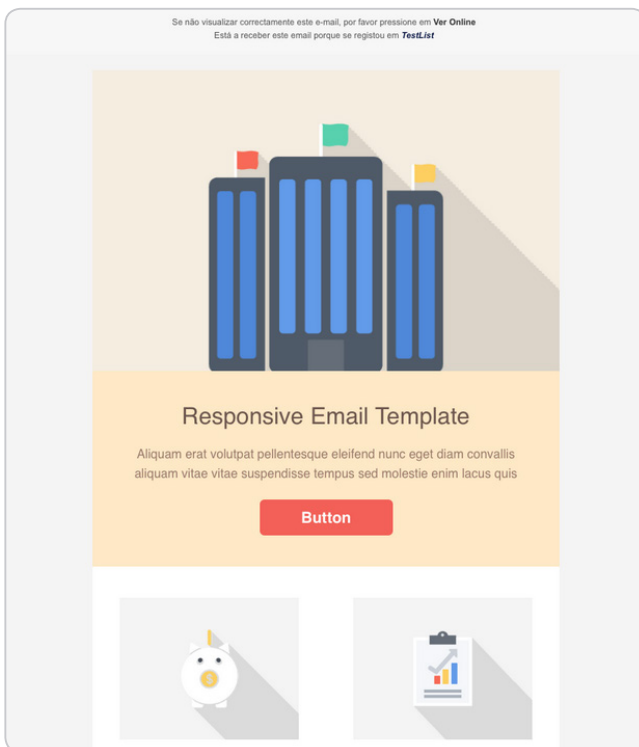
```
(...)
@media screen and (max-width: 600px) {
  a[class="mobile-button"] {
    width: 60% !important;
    display: block !important;
  }
  td[class="mobile-hide"] {
    display: none !important;
  }
  td[class="mobile-text"] {
    font-size: 25px !important;
    text-transform: uppercase !important;
    color: #f76552 !important;
  }
}
(...)
```



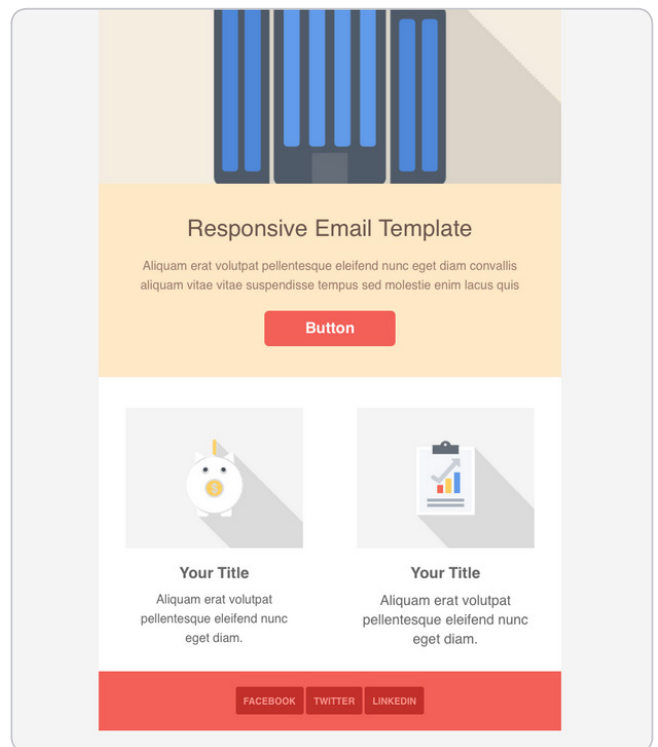
8 - TESTING

When your over with coding your email campaign, always test how your email renders in different email clients and browsers. For example, Outlook and Gmail renders your HTML and CSS differently. This will help you to look for bugs and main alterations you didn't plan. There are some platforms that helps you with this process. For example, Litmus let's you preview your campaign across 30+ email clients and devices with screenshots.

LITMUS TESTINGS DESKTOP CLIENTS

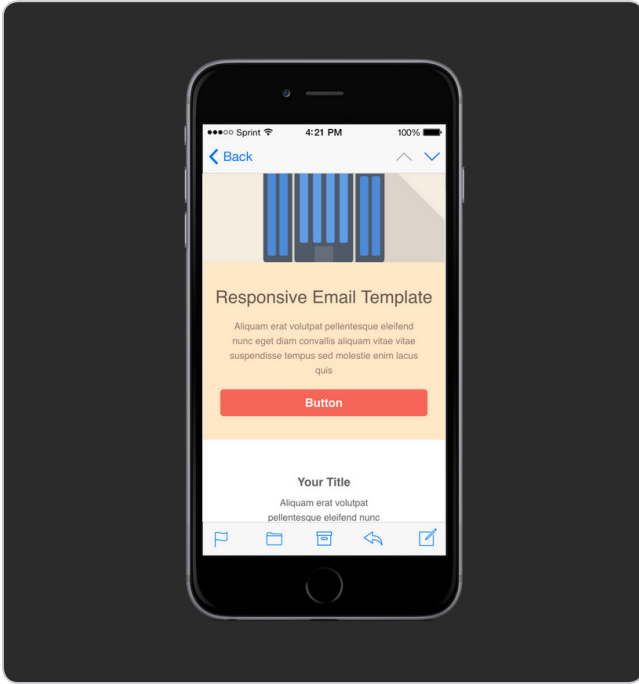


Apple mail 7

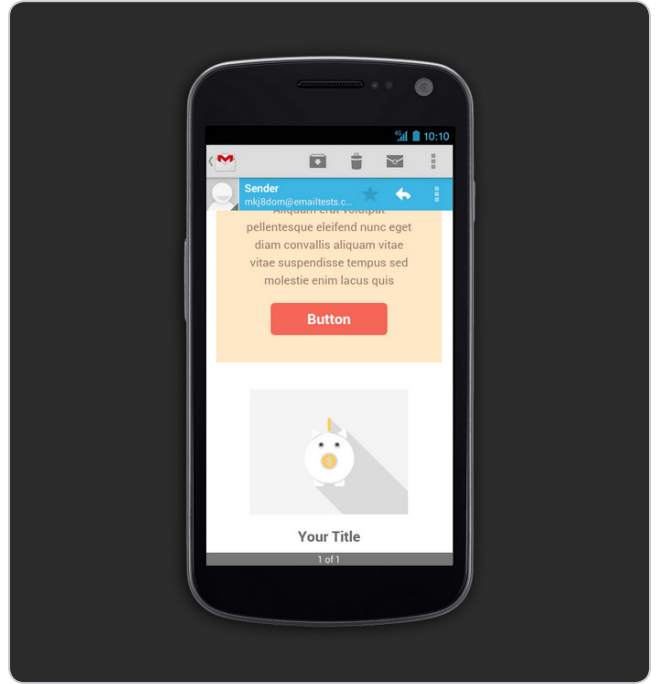


Outlook 2011

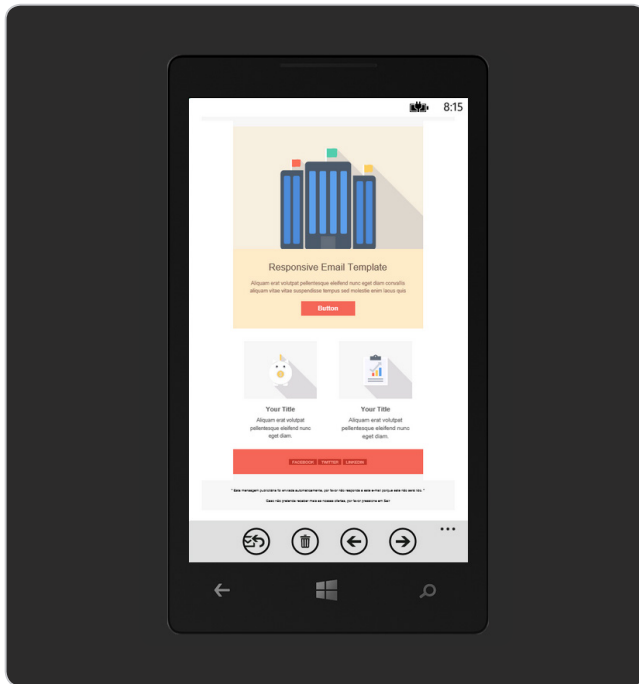
MOBILE CLIENTS



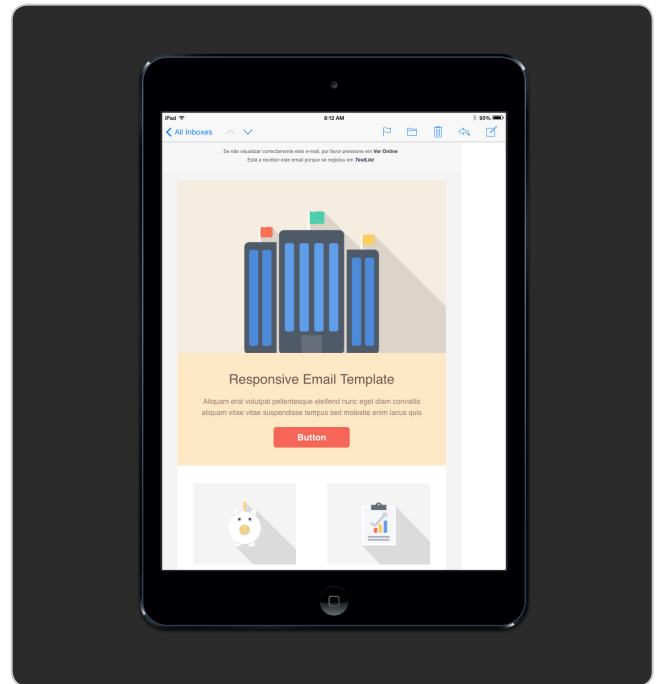
iPhone 6 Plus



Gmail App (Android)

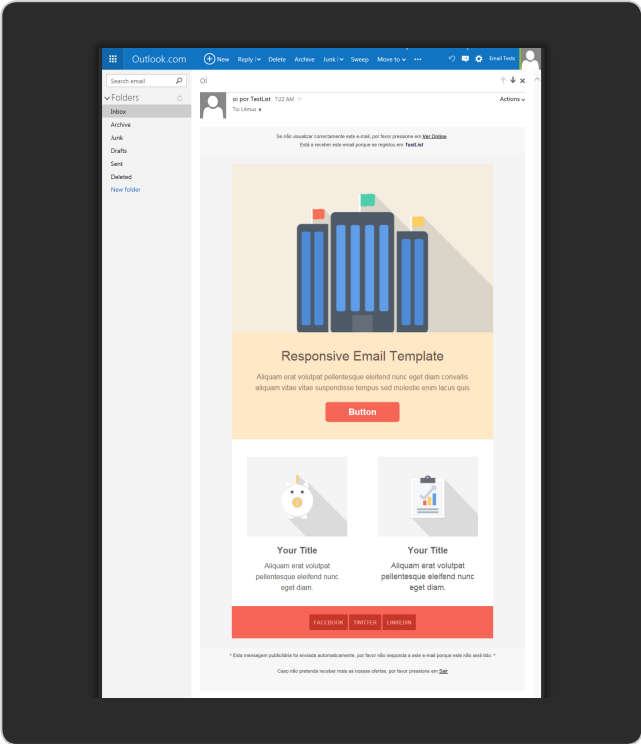


Windows Phone 8

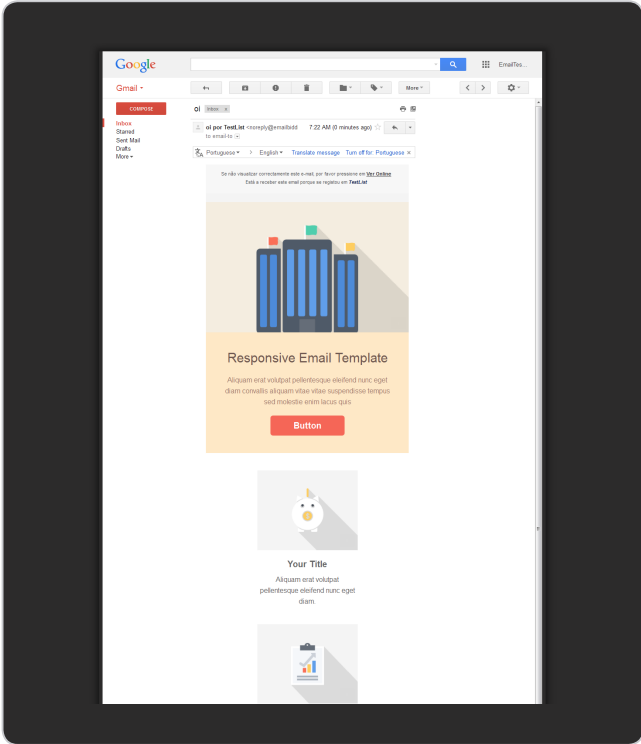


iPad Mini

WEB-BASED CLIENTS



Outlook.com (Explorer)



Gmail (Firefox)



GET IN TOUCH AND FIND OUT
HOW YOU CAN POWER-UP YOUR
EMAIL MARKETING

PORTUGAL

Rua Alfredo Allen, n° 455/461, 4200-135 Porto . Portugal
P: +351 300 401 582

FRANCE

60, boulevard du Maréchal Joffre, 92340 Bourg-la-Reine . France
P: +33 1 70 68 97 37

ITALY

P.le Cadorna 10 , 20123 Milano . Italy
P: +39 02 56 56 64 75

www.emailbidding.com