

HTML TIPS FOR DESIGNING YOUR EMAIL

HTML TIPS FOR DESIGNING YOUR EMAIL

We have over 700 predesigned email templates to use in our system, so whether you are just getting started or are an experienced user, there are many options for you.

This guide was created with our more advanced users in mind, those who prefer to use our Freeform email editor to create their emails and are comfortable writing HTML. Before we get to the tips, the most important thing to keep in mind when you are designing HTML for emails is that emails are not websites. Remember the Prince song “1999”? Today we are going to code like its 1999. Yep, email programs are still living in the past and are not as sophisticated as browsers, so keep your code simple.

7 Tips For HTML Emails

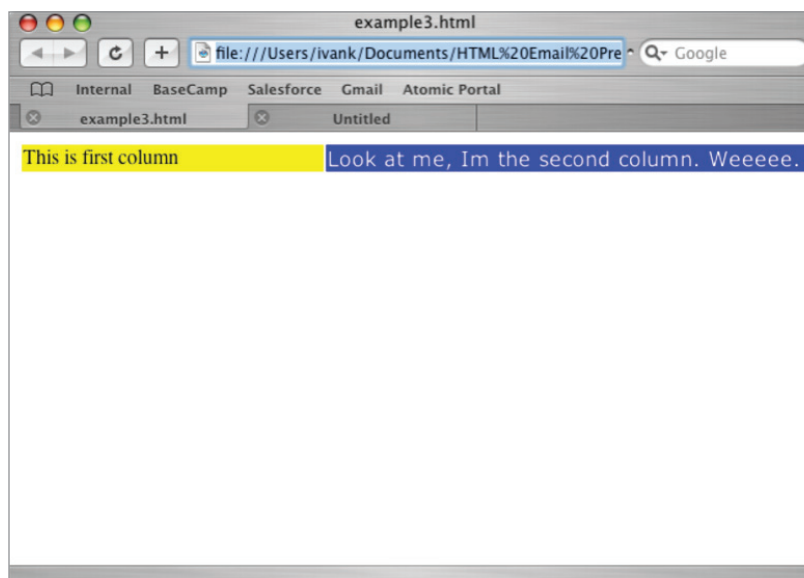
1 Use Tables

Email programs use different HTML rendering tools, so keep your code simple.

Use tables for the layout and positioning of your email; table nesting will give you much better results across more email programs.

For example, to create a two-column, newsletter-style email, you’ll want to create a table for the header, a table for the content section and a table for the footer. Then use HTML table attributes to control how the tables are displayed.

```
<table align="center"><tr>
<td width="250" bgcolor="yellow"> This is the first column. </td>
<td width="400" bgcolor="blue"> <font face="Verdana" color="#ffffff">
Look at me, I'm the second column. Weeeee.</font></td></tr></table>
```



Wrap these three tables in a container table and set the width to 100% (this will come in handy in Tip 3).

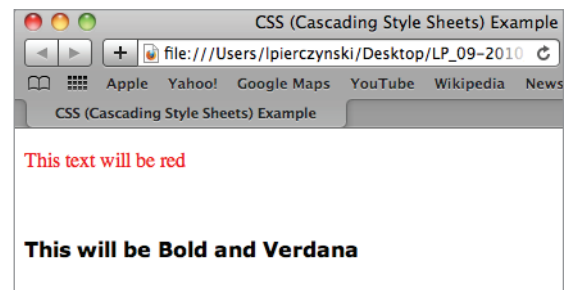
2 CSS

Now that you have your nested tables set up, you can start designing the style for your content. While CSS (Cascading Style Sheets) is great and pretty much necessary for websites, it does not work well for HTML emails. Most email programs, whether they're online, browser-based ones like Yahoo and Gmail, or desktop clients, like Outlook, will ignore CSS in the <head> tag. **The best and most reliable approach for emails is inline CSS; don't use external stylesheets.**

```
<p style="color:#ff0000"> This text will be red</p>

<p style="font-family: Verdana; font-weight: bold;"> This will be Bold
and Verdana</p>
```

Gmail is known for stripping out CSS; if you want to test what your email will look like without it, try sending a test to a Gmail account.



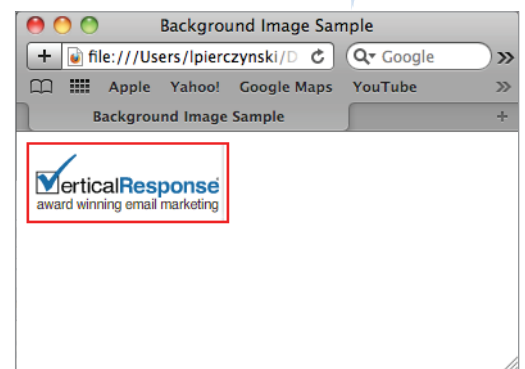
3 Background

Using background images is very popular for websites and to some extent emails. However, there is so much inconsistency between which email programs will actually display them that it's best to not use them at all. If you wish to use some background images anyway, even though we don't recommend it, use the HTML table attribute **background=** instead of CSS.

```
<table><tr>
<td bgcolor="#ff0000" background="http://www.mywebsite.com/images/
someimage.gif">&nbsp;  </td>
</tr></table>
```

Ensure you test your email across different email programs to see how it will look with or without the background images. Also, if you have text over the image be sure the color of the font is still readable over your background color if the image is not rendered.

This seems like a great time to talk about background color, since we just mentioned it. As we hinted in **Tip 1**, if you wrap your tables in a container table, you can use this to specify a background color for your email without it being stripped out by email programs.



4 Images

Almost every email program out there has a feature that hides images by default. Images are not displayed until the recipient allows them to load. This means you need to ensure that your email conveys the important information through the text of the email, not just the images. We like to suggest an 80/20 rule of thumb: 80% text to 20% images in your email. This is not a law though; if your content needs something more like 60/40 that will be fine too, just keep it balanced.

Fun Marketing Tip! Because of how images are handled by email programs, a great way to convey information to your recipients is to use **Alt tags** for your images. Instead of seeing a broken image icon, your recipients will see whatever text you add to the **Alt tag**.

Thinking of using image maps? Since images are frequently turned off, at least when an email is first loaded, most links will not be clickable in many email programs. It's usually better to not use this technique in HTML emails.

This brings up one last thing for images: if you are thinking of slicing up your image (see **Tip 6**) because image maps won't work, Gmail has recently made some changes that could affect how it's rendered.

To avoid any problems code your images like this:

```

```

Adding the style tag, in bold in the sample code, will help image slices appear correctly in Gmail and some other email programs.

5 Good Code

On a website, using Flash can add a lot to your visitors experience on your site. In an email, it makes the file size larger, causes potential spam filter problems, or just doesn't work. In addition to Flash, it's a good idea to avoid using Javascript, ActiveX and even videos in your emails for these same reasons. Your best bet is to link to a site where your recipient can see these elements rather than to try to embed them in an email.

One more thing - every tag has to be closed and nested correctly!

```
<p>This is a paragraph, where <b>these words are bold</p></b> Wrong!
```

```
<p>This is a paragraph, where <b>these words are bold</b></p> Right!
```

```
<br> This will work but...
```

```
<br/> This is the correct way
```

6 Remember Spam Filters

Ah spam filters, the bane of many an email marketer. Your HTML code can cause your email to go to a spam folder just as much as using F*R*E*E in the subject line will. Keep a balance of images and text; emails containing only images tend to be filtered to spam folders. Remember the 80/20 rule of thumb in **Tip 4**? Spam filters are another reason to follow this rule. Also, make sure any links you include in your email go to good, reputable websites; a bad URL can also help land your email in the spam folder. **Always use:**

```
<HTML>
  <body></body>
</HTML>
```

tags to open and close your email HTML code. And close all your tags! Dirty HTML code will flag an email as spam pretty quickly.

7 Best Practices: things to keep in mind as you set up your email

Keep the width around 600 pixels – This will ensure your recipient can read the email easily and comfortably. For anyone reading the emails on a mobile device this width should scale down well for the size of the window on the device and still make the email readable.

Beware of WYSIWYG editors – Many WYSIWYG editors will add strange and unnecessary code to your email causing problems with rendering in email programs, and even landing it in the spam folder. This includes code from a Word document; it is not all HTML code and will cause a lot of issues with your email.

Test your email! – You can send tests to your own email accounts or use a program like Email On Acid (www.emailonacid.com) to test how it will be rendered in various programs.

All images must be hosted and referenced by absolute urls in your source code (i.e. `` rather than ``).

Your HTML source code file should not exceed 45KB.

The use of HTML forms in your email message is not recommended; many email clients will not render forms correctly.

Avoid using HTML frames, layering or absolute positioning.

Do not rely on CSS inheritance. Write your CSS specifically (i.e. `element#idname p {}`) as opposed to just `p {}`.

Text links need explicit inline styling to render as desired in most webmail clients.

For example: `VerticalResponse`

Avoid commenting within the CSS, as this could cause problems with some webmail clients.

So there you have it, 7 easy steps of tips to create HTML for emails. These will help your email work successfully across most email programs and should keep your email out of the spam folder. Now get ready to code like its 1999 and keep it simple.