

Use tables for layout

Because clients like Gmail and Outlook 2007 have poor support for `float`, `margin` and `padding`, you'll need to use tables as the framework of your email. While nested tables are widely supported, consistent treatment of `width`, `margin` and `padding` within table cells is not. For the best results, keep the following in mind when coding your table structure.

Set the width in each cell, not the table

When you combine `table` widths, `td` widths, `td` padding and CSS padding into an email, the final result is different in almost every email client. The most reliable way to set the width of your table is to set a width for each cell, not for the table itself.

```
<table cellspacing="0" cellpadding="10" border="0">
<tr>
  <td width="80"></td>
  <td width="280"></td>
</tr>
</table>
```

Never assume that if you don't specify a cell width the email client will figure it out. It won't. Also avoid using percentage based widths. Clients like Outlook 2007 don't respect them, especially for nested tables. Stick to pixels. If you want to add padding to each cell, use either the `cellpadding` attribute of the table or CSS padding for each cell, but never combine the two.

Err toward nesting

Table nesting is far more reliable than setting left and right margins or padding for table cells. If you can achieve the same effect by table nesting, that will always give you the best result across the buggier email clients.

Use a container table for body background colors

Many email clients ignore background colors specified in your CSS or the `<body>` tag. To work around this, wrap your entire email with a 100% width table and give that a background color.

```
<table cellspacing="0" cellpadding="0" border="0" width="100%">
<tr>
  <td bgcolor="#000000">
    Your email code goes here.
  </td>
</tr>
</table>
```

You can use the same approach for background images too. Just remember that some email clients don't support them, so always provide a fallback color.

Avoid unnecessary whitespace in table cells

Where possible, avoid whitespace between your `<td>` tags. Some email clients (ahem, Yahoo! and Hotmail) can add additional padding above or below the cell contents in some scenarios, breaking your design for no apparent reason.

CSS and general font formatting

While some email designers do their best to avoid CSS altogether and rely on the dreaded `` tag, the truth is many CSS properties are well supported by most email clients. See this comprehensive [list of CSS support](#) across the major clients for a good idea of the safe properties and those that should be avoided.

Always move your CSS inline

Gmail is the culprit for this one. By stripping the CSS from the `<head>` and `<body>` of any email, we're left with no choice but to move all CSS inline. The good news is this is something you can almost completely automate. Free services like [Premailer](#) will move all CSS inline with the click of a button. I recommend leaving this step to the end of your build process so you can utilize all the benefits of CSS.

Avoid shorthand for fonts and hex notation

A number of email clients reject CSS shorthand for the `font` property. For example, never set your font styles like this.

```
p {
  font:bold 1em/1.2em georgia,times,serif;
}
```

Instead, declare the properties individually like this.

```
p {
  font-weight: bold;
  font-size: 1em;
  line-height: 1.2em;
  font-family: georgia,times,serif;
}
```

While we're on the topic of fonts, I recently tested every conceivable variation of `@font-face` across the major email clients. The results were dismal, so unfortunately it's web-safe fonts in email for the foreseeable future.

When declaring the `color` property in your CSS, some email clients don't support shorthand hexadecimal colors like `color:#f60;` instead of `color:#ff6600;`. Stick to the longhand approach for the best results.

Paragraphs

Just like table cell spacing, paragraph spacing can be tricky to get a consistent result across the board. I've seen many designers revert to using double `
` or DIVs with inline CSS margins to work around these shortfalls, but recent testing showed that paragraph support is now reliable enough to use in most cases (there was a time when Yahoo! didn't support the paragraph tag at all).

The best approach is to set the `margin` inline via CSS for every paragraph in your email, like so:

```
p {  
    margin: 0 0 1.6em 0;  
}
```

Again, do this via CSS in the head when building your email, then use [Premailer](#) to bring it inline for each paragraph later.

If part of your design is height-sensitive and calls for pixel perfection, I recommend avoiding paragraphs altogether and setting the text formatting inline in the table cell. You might need to use table nesting or `cellpadding` / CSS to get the desired result. Here's an example:

```
<td width="200" style="font-weight:bold; font-size:1em; line-height:1.2em;  
font-family:georgia, 'times', serif;">your height sensitive text</td>
```

Links

Some email clients will overwrite your link colors with their defaults, and you can avoid this by taking two steps. First, set a default color for each link inline like so:

```
<a href="http://somesite.com/" style="color:#ff00ff">this is a link</a>
```

Next, add a redundant span inside the a tag.

```
<a href="http://somesite.com/" style="color:#ff00ff"><span  
style="color:#ff00ff">this is a link</span></a>
```

To some this may be overkill, but if link color is important to your design then a superfluous span is the best way to achieve consistency.

Images in HTML emails

The most important thing to remember about images in email is that they won't be visible by default for many subscribers. If you start your design with that assumption, it forces you to keep things simple and ensure no important content is suppressed by image blocking.

With this in mind, here are the essentials to remember when using images in HTML email:

Avoid spacer images

While the combination of spacer images and nested tables was popular on the web ten years ago, image blocking in many email clients has ruled it out as a reliable technique today. Most clients replace images with an empty placeholder in the same dimensions, others strip the image altogether. Given image blocking is on by default in most email clients, this can lead to a poor first impression for many of your subscribers. Stick to fixed cell widths to keep your formatting in place with or without images.

Always include the dimensions of your image

If you forget to set the dimensions for each image, a number of clients will invent their own sizes when images are blocked and break your layout. Also, ensure that any images are correctly sized before

adding them to your email. Some email clients will ignore the dimensions specified in code and rely on the true dimensions of your image.

Avoid PNGs

Lotus Notes 6 and 7 don't support 8-bit or 24-bit PNG images, so stick with the GIF or JPG formats for all images, even if it means some additional file size.

Provide fallback colors for background images

Outlook 2007 has no support for background images (aside from [this hack](#) to get full page background images working). If you want to use a background image in your design, always provide a background color the email client can fall back on. This solves both the image blocking and Outlook 2007 problem simultaneously.

Don't forget alt text

Lack of standards support means email clients have long destroyed the chances of a semantic and accessible HTML email. Even still, providing alt text is important from an image blocking perspective. Even with images suppressed by default, many email clients will display the provided alt text instead. Just remember that some email clients like Outlook 2007, Hotmail and Apple Mail don't support alt text at all when images are blocked.

Use the display hack for Hotmail

For some inexplicable reason, Windows Live Hotmail adds a few pixels of additional padding below images. A workaround is to set the display property like so.

```
img {display:block;}
```

This removes the padding in Hotmail and still gives you the predicable result in other email clients.

Don't use floats

Both Outlook 2007 and earlier versions of Notes offer no support for the float property. Instead, use the align attribute of the `img` tag to float images in your email.

```

```

If you're seeing strange image behavior in Yahoo! Mail, adding align="top" to your images can often solve this problem