FREE eBook

LEARNING email

Free unaffiliated eBook created from **Stack Overflow contributors.**



Table of Contents

About
Chapter 1: Getting started with email
Remarks2
Examples2
HTML and Plain Text
Email Clients and Rendering Engines5
The Basics (Tables and Inline CSS)6
Coding emails for all email clients
Chapter 2: Bulletproof Buttons
Introduction11
Remarks
Examples
Buttons.cm Button
Padding-Based Buttons
Border-Based Buttons
Padding + Border-Based Buttons14
Bullet-proof button with spacers14
Chapter 3: CSS/HTML Support
Introduction16
Examples
Visual Decorations
Chapter 4: HTML email composition and Outlook intricacies
Examples
Background images
Full width table cell background images
Tiled full width table cell background images18
Specified width table cell background images20
Creating spacers
Outlook can sometimes add a bit of spacing on the left and right side of a element that ca
Image Resizing

Chapter 5: HTML Email Reset	23
Introduction	23
Examples	
Resetting Email CSS in	24



You can share this PDF with anyone you feel could benefit from it, downloaded the latest version from: email

It is an unofficial and free email ebook created for educational purposes. All the content is extracted from Stack Overflow Documentation, which is written by many hardworking individuals at Stack Overflow. It is neither affiliated with Stack Overflow nor official email.

The content is released under Creative Commons BY-SA, and the list of contributors to each chapter are provided in the credits section at the end of this book. Images may be copyright of their respective owners unless otherwise specified. All trademarks and registered trademarks are the property of their respective company owners.

Use the content presented in this book at your own risk; it is not guaranteed to be correct nor accurate, please send your feedback and corrections to info@zzzprojects.com

Chapter 1: Getting started with email

Remarks

HTML email is the use of a subset of HTML and CSS to format an email message like a web page using colors, graphics, table columns and links.

Useful Links:

- Things I've Learned About Building & Coding HTML Email Templates
- Coding Your Emails
- Email Design Guides
- MailChimp Resources
- Professional Email Design (Book)

Examples

HTML and Plain Text

HTML email is the use of a subset of HTML and CSS to format an email message like a web page using colors, graphics, table columns and links.

When you send an email it's important to send both plain text and HTML. You do this by sending your email as multi-part MIME. Most email service providers have tools to automatically construct the MIME for you. Some will generate a plain text version based on your HTML version, and there are third party tools as well.

Example of a text-only email:

```
# EMAIL TITLE GOES HERE
Lorem ipsum dolor sit amet, consectetur adipiscing elit. Morbi commodo elit sed quam auctor,
ut facilisis odio aliquam. In hac habitasse platea dictumst.
Learn more: http://www.website.com/landing-page-1
----
## Subheader goes here
Lorem ipsum dolor sit amet, consectetur adipiscing elit. Morbi commodo elit sed quam auctor,
ut facilisis odio aliquam. In hac habitasse platea dictumst.
Learn more: http://www.website.com/landing-page-2
----
Footer and Sender information
```

Unsubscribe here: http://www.website.com/unsubscribe



. This content will only be displayed in the Apple Watch (and any other clients that support this MIME type).

Email Clients and Rendering Engines

Email clients use different rendering engines to render HTML emails:

- Apple Mail, Outlook for Mac, Android Mail and iOS Mail use WebKit
- Outlook 2000/02/03 use Internet Explorer 6
- Outlook 2007/10/13 use Microsoft Word
- Web clients use their browser's respective engine (e.g. Safari uses WebKit, Chrome uses Blink)

Some clients will also add their own styles on top of yours e.g. Gmail sets all fonts to font-family: arial, sans-serif;, which could impact how an email renders.

Prior to 2016, all Gmail products stripped out <link> tags, any CSS inside <style> tags, and any other CSS that isn't inlined. In late 2016, Gmail began supporting the <style> tag and mediaqueries in most of its products. Here is an update as of 23-Nov-2016: **Gmail Client**

Desktop webmail

Mobile webmail

Gmail Android app (Gmail account)

Gmail Android app (Gmailified account)

Gmail Android app (POP/IMAP account)

Gmail iOS app

Inbox by Gmail (webmail)

https://riptutora.com/y Gmail (iOS)

-based CSS layout doesn't work in email as it does on the web. Major email clients either offer no support for things like floats and flexbox, or mangle it in different ways. <div>s also have positioning and box model issues in different clients, particularly Outlook. There are a few techniques to code an email using only <div>s, but it's safer to stick with tables for layout.

```
<!DOCTYPE html>
<html lang="en">
 <head>
    <meta charset="UTF-8">
    <title>Hello!</title>
  </head>
  <body>
    <h1>Hello World!</h1>
        This is a simple paragraph.
        </body>
</html>
```

Inline CSS

Applying a style inline gives it priority over styles further away (such as webmail client styles), and also works around the email clients that strip out CSS from the head or external CSS files. Inline CSS is the best way to ensure consistent display in every email client.

```
<!DOCTYPE html>
<html lang="en">
  <head>
     <meta charset="UTF-8">
     <title>Hello!</title>
  </head>
  <body style="background: #000000;">
     <table cellspacing="0" cellpadding="0" border="0" width="600" role="presentation"
style="margin: auto; background: #ffffff;">
        <h1 style="font-weight: bold; color: #444444; font-family: Georgia,</pre>
serif;">Hello World!</h1>
          This
is a simple paragraph.
          </body>
</html>
```

You have a couple of options for inlining CSS:

- Write CSS inline as you go
- Use local snippets in your editor
- Use a web-based CSS inliner like this one
- Use a programmatic CSS inliner like this one
- · Use a build process like this one or a templating language
- Let your email service provider handle the inlining for you (if they support it)

Coding emails for all email clients

Coding method used: Hybrid/Spongy

It has been forever a myth that div's can not be used in emails. There are email clients (unlike outlook) that can render div's properly. The example below will illustrate how an email can be coded that will work on Gmail app (with updates not rolled out yet), Samsung device's and other email clients that don't read media queries.

Introducing Outlook conditional statements

Outlook conditional statements are very useful when it come to rendering emails or showing something specific like fall backs for outlook.

```
<!--[if (gte mso 9)|(IE)]>
<![endif]-->
```

The above code reads if greater than microsoft outlook 9 or IE

```
      Outlook
      2000
      -
      Version
      9

      Outlook
      2002
      -
      Version
      10

      Outlook
      2003
      -
      Version
      11

      Outlook
      2007
      -
      Version
      12

      Outlook
      2010
      -
      Version
      14

      Outlook
      2013
      -
      Version
      15
```

Versions for conditional statements listed.

Starting a hybrid email template

Each step has been explained in a way that it should be easy for anyone with basic HTML knowledge to understand.

First we start with a wrapper table which will span all the way across the screen and with a class of container.

```
        (tr>
        </ttr>
```

After that we add in a media query for email clients that dont read max width but read the CSS in the header. The media query will be targeting all screens and showing the container at 700 pixels width.

```
@media only screen and (max-width : 700px) {
.container{width:700px;}
}
```

Next we add an outlook conditional statement that keeps the table (with class container) to be at a width of 700 pixels.

```
<!--[if (gte mso 9) | (IE)]>
  <![endif]-->
     <table class="container" width="100%" border="0" cellspacing="0" cellpadding="0"
style="width: 100%; max-width: 700px;">
         <td valign="top" bgcolor="#FFFFFF" style="padding:0px;text-align: center;
vertical-align: top; font-size: 0px;">[CONTENT GOES HERE]
          <!--[if (gte mso 9)|(IE)]>
       <![endif]-->
```

The above code should now hold your template in outlook at 700px wide.

Now for the columns. Below code will create two equal columns on the template both at 350px wide.

```
<!--[if (gte mso 9) | (IE)]>
>
   <![endif]-->
<div style="width: 350px; display: inline-block; vertical-align: top;">
[COLUMN CONTENT HERE] 
 </div>
<!--[if (gte mso 9) | (IE)]>
<![endif]-->
```

```
<div style="width: 350px; display: inline-block; vertical-align: top;">

[COLUMN CONTENT HERE]
</d>
```

After this the limit is only your imagination or the designer. When designs are done it is important to be involved in the wire framing stage so there are no suprises once the design is in coding stage.

Note:

- Retina images will need images set at the element level not on its style <img src="" width=""
 ...
- You can still choose to do in-line CSS or you can choose to CSS in head only **if** all your clients support CSS in the head.

Read Getting started with email online: https://riptutorial.com/email/topic/4990/getting-started-withemail

Chapter 2: Bulletproof Buttons

Introduction

Buttons can be one of the most frustrating parts of email development, but they *can* be built with code instead of images and still display well in all major email clients. When built in HTML/CSS, a button will display in all email clients even with images off. Both the content and style of each button can be quickly edited in HTML/CSS template.

Remarks

Progressive Enhancement with Media Queries

For clients that support media queries, any of the examples above can be progressively enhance the button.

```
Eg.@media screen and (-webkit-min-device-pixel-ratio: 0) { /* Insert -webkit-targeted CSS here */ }
```

Add more horizontal padding in Outlook

Conditionally adding non-breaking space on each side of a button link can increase the horizontal padding for Outlook. Especially useful for border-based buttons.

```
`<!--[if mso]>&nbsp;&nbsp;<![endif]-->Link Text<!--[if mso]>&nbsp;&nbsp;<![endif]-
->`
```

Examples

Buttons.cm Button

Campaign Monitor built a nifty tool that generates code for buttons in HTML email. It uses a cominbation os CSS and VML (for Microsoft Outlook) to display background images or patterns in most major email clients.

```
radius:3px;color:#fffffff;display:inline-block;font-family:sans-serif;font-size:13px;font-
weight:bold;line-height:38px;text-align:center;text-decoration:none;width:200px;-webkit-text-
size-adjust:none;mso-hide:all;">Stack Overflow</a></div>
```

Will display something like this:



note: Since the clickable hyperlink in Outlook is inside a VML tag, some email service providers have trouble tracking clicks on this link.

```
<v:roundrect xmlns:v="urn:schemas-microsoft-com:vml" xmlns:w="urn:schemas-microsoft-
com:office:word" href="http://www.stackoverflow.com" style="height:38px;v-text-
anchor:middle;width:200px;" arcsize="8%" strokecolor="#7ea613" fill="t">
```

Padding-Based Buttons

This technique relies on padding at the table cell level to structure the button, and both HTML attributes and CSS to style the button. No $_{\text{VML}}$ means simpler code, but no background images in Outlook.

Stack Overflow

The drawback of padding-based-buttons is that the entire area of the button is not clickable, only the link within the table cell is.

Border-Based Buttons

This technique relies on adding thick borders to the link itself to build the button's CTA. Using borders is universally understood by email clients, but limit button appearance solid colors.

Stack Overflow

The link tag is set to be a block-level element and that borders are used to provide the padding, so the entire button is hoverable + clickable, even in older desktop clients.

The drawback of border-based-buttons is that Outlook will reduce the size of the borders by a small amount and, as mentioned above, background images are not supported.

Padding + Border-Based Buttons

This technique uses a combination of border-based and padding-based buttons, styling the link with both padding and at least a solid 1px border. The background color needs to be applied to the instead of the in this instance because Outlook does recognize horizontal padding on the tag (since it's not a block level tag).



The drawback of padding+border-based-buttons is that it separates the styling between a and <a> tag, so it can be difficult to maintain. But only one button is used, specific dimensions are not required, and background images can be supported.

Bullet-proof button with spacers

This button derives from this example by Email on Acid. It is entirely code-based, so it will display without images downloaded, and the entire button is hoverable + clickable.

Additionally, this example also includes spacers to help control how much vertical space appears before and after the button.

```
<!-- Top Spacer : BEGIN -->
```

```
<t.r>
       
       <!-- Top Spacer : END -->
    <!-- Button : BEGIN -->
    <td align="center" bgcolor="#0077CC" width="150" style="-moz-border-radius:
4px; -webkit-border-radius: 4px; border-radius: 4px;">
           <a href="http://www.stackoverflow.com" style="padding: 10px; width:150px;</pre>
display: block;text-decoration: none; border:0;text-align: center; font-weight: bold; font-
size: 14px; font-family: sans-serif; color: #ffffff; background: #0095FF; border: 1px solid
#0077CC; -moz-border-radius: 4px; -webkit-border-radius: 4px; border-radius: 4px; line-
height:17px;" class="button_link">Stack Overflow</a>
          <!-- Button : END -->
    <!-- Bottom Spacer : BEGIN -->
     
      <!-- Bottom Spacer : END -->
```

Stack Overflow

Read Bulletproof Buttons online: https://riptutorial.com/email/topic/8182/bulletproof-buttons

Chapter 3: CSS/HTML Support

Introduction

Not every css property or HTML tag is supported by all email clients and here we can keep track of it & possible work around.

Examples

Visual Decorations

CSS properties which add visual decorations but are not supported can be replaced with image tag.

For example: border-radius is not supported in Yahoo! Mail, Outlook 2007/10/13 +, Outlook 03/Express/Mail & Android 4 (Gmail) +

To work around this we can add images with border radius in them i.e buttons with left & right side as image:

Here is a list of property that should be avoided and used as image instead:

border-radius, text-shadow, box-shadow, text-fill-color & text-fill-stroke

Read CSS/HTML Support online: https://riptutorial.com/email/topic/9357/css-html-support

Chapter 4: HTML email composition and Outlook intricacies

Examples

Background images

Full width table cell background images

```
<html xmlns:v="urn:schemas-microsoft-com:vml">
   <head>
   <stvle>
   v:* { behavior: url(#default#VML); display: inline-block; }
   </style>
   </head>
   <body>
      <center>
      <td bgcolor="#dddddd" style="background-
image:url('http://placekitten.com/g/500/300');background-repeat:no-repeat;background-
position:center; background="http://placekitten.com/g/500/300" width="100%" height="300">
              <!--[if gte mso 9]>
                 <v:rect xmlns:v="urn:schemas-microsoft-com:vml" fill="true" stroke="false"</pre>
style="mso-width-percent:1000;height:300px;">
                    <v:fill type="frame" src="http://placekitten.com/g/500/300"
color="#fffffff" />
                 </v:rect>
             <![endif]-->
              <td
height="5"><table border="0" cellpadding="0"
cellspacing="0">
              <t.r>
             <td bgcolor="#ffff99" style="background-
image:url('http://placekitten.com/g/500/300');background-repeat:no-repeat;background-
position:center;" background="http://placekitten.com/g/500/300" width="100%" height="300">
                 <!--[if gte mso 9]>
                    <v:rect xmlns:v="urn:schemas-microsoft-com:vml" fill="true"
stroke="false" style="mso-width-percent:1000;height:300px;">
                        <v:fill type="frame" src="http://placekitten.com/g/500/300"
color="#fffffff" />
                    </v:rect>
                 <![endif]-->
              </center>
```

• Tested in Outlook 2010 thru Windows 7 VMWare on OSX 10.9.2



Tiled full width table cell background images

```
<html xmlns:v="urn:schemas-microsoft-com:vml">
       <head>
           <style>
               v:* { behavior: url(#default#VML); display: inline-block; }
           </style>
       </head>
       <body>
       <center>
       <td bgcolor="#dddddd" style="background-
image:url('http://placekitten.com/g/500/300');background-repeat:no-repeat;background-
position:center;" background="http://placekitten.com/g/500/300" width="100%" height="300">
                   <!--[if gte mso 9]>
                       <v:rect xmlns:v="urn:schemas-microsoft-com:vml" fill="true"
stroke="false" style="mso-width-percent:1000;height:300px;">
                           <v:fill type="tile" src="http://placekitten.com/g/500/300"
```

• Tested in Outlook 2010 thru Windows 7 VMWare on OSX 10.9.2

VMware Fu	ision F	ile Edit	View	Virtual Machine	Window	Help	and the second
000							🖻 win8
	2						
🖂 📙 🌖	0 4 7	-				Untitled Test HTML	- Message (HTML)
File	Message	Developer					
If there a	re problems	with how th	is message	e is displayed, click here	to view it in a	web browser.	
From:	test@puts	mail.com	-				
To:	davidcondi	rey@gmail.cor	m				
Cc:							
Subject:	Untitled T	est HTML					

Specified width table cell background images

Here is an example, two rows, the first row has 3 columns with 3 separate background images, the second row spans all the way across as one background image.

```
<table width="600" border="0" cellpadding="0" cellspacing="0" style="border-collapse:
collapse;">
       <td style="width: 300px; height: 80px; background-image:
url('http://placekitten.com/g/300/80');">
           <!--[if gte mso 9]>
               <v:image xmlns:v="urn:schemas-microsoft-com:vml" id="theImage"
style='behavior: url(#default#VML); display: inline-block; position: absolute; width: 300px;
height: 80px; top: 0; left: 0; border: 0; z-index: 1; src="http://placekitten.com/g/300/80"
/>
               <v:shape xmlns:v="urn:schemas-microsoft-com:vml" id="theText" style='behavior:
url(#default#VML); display: inline-block; position: absolute; width: 300px; height: 80px; top:
-5; left: -10; border: 0; z-index: 2; '>
               <div>
           <![endif]-->
               <table width="300" border="0" cellspacing="0" cellpadding="0" style="border-
collapse: collapse;">
                   <td height="80" align="center" valign="top" style="color:#ffffff;font-
size:20px;">
                          <span>Text</span>
                       <!--[if gte mso 9]>
               </div>
               </v:shape>
           <![endif]-->
           <td style="width: 100px; height: 80px; background-image:
url('http://placekitten.com/g/100/80');">
           <!--[if gte mso 9]>
               <v:image xmlns:v="urn:schemas-microsoft-com:vml" id="theImage"
style='behavior: url(#default#VML); display: inline-block; position: absolute; width: 100px;
height: 80px; top: 0; left: 0; border: 0; z-index: 1; ' src="http://placekitten.com/g/100/80"
/>
               <v:shape xmlns:v="urn:schemas-microsoft-com:vml" id="theText" style='behavior:
url(#default#VML); display: inline-block; position: absolute; width: 100px; height: 80px; top:
-5; left: -10; border: 0; z-index: 2; '>
               <div>
           <![endif]-->
               <table width="80" border="0" cellspacing="0" cellpadding="0" style="border-
collapse: collapse;">
                   <span>Text</span>
                      <!--[if gte mso 9]>
               </div>
               </v:shape>
           <![endif]-->
```

```
<td style="width: 200px; height: 80px; background-image:
url('http://placekitten.com/g/200/100');">
           <!--[if gte mso 9]>
               <v:image xmlns:v="urn:schemas-microsoft-com:vml" id="theImage"
style='behavior: url(#default#VML); display: inline-block; position: absolute; width: 200px;
height: 80px; top: 0; left: 0; border: 0; z-index: 1; ' src="http://placekitten.com/g/200/100"
/>
               <v:shape xmlns:v="urn:schemas-microsoft-com:vml" id="theText" style='behavior:
url(#default#VML); display: inline-block; position: absolute; width: 200px; height: 80px; top:
-5; left: -10; border: 0; z-index: 2; '>
               <div>
           <![endif]-->
               <table width="200" border="0" cellspacing="0" cellpadding="0" style="border-
collapse: collapse;">
                   <td height="80" align="center" valign="top" style="color:#ffffff;font-
size:20px;">
                           <span>Text</span>
                       <!--[if gte mso 9]>
               </div>
               </v:shape>
           <![endif]-->
            </t.d>
       </t.r>
    <table width="600" border="0" cellpadding="0" cellspacing="0" style="border-collapse:
collapse;">
       <t.r>
           <td style="width: 600px; height: 150px; background-image:
url('http://placekitten.com/g/600/150');">
           <!--[if gte mso 9]>
               <v:image xmlns:v="urn:schemas-microsoft-com:vml" id="theImage"
style='behavior: url(#default#VML); display: inline-block; position: absolute; width: 600px;
height: 150px; top: 0; left: 0; border: 0; z-index: 1;' src="http://placekitten.com/g/600/150"
/>
               <v:shape xmlns:v="urn:schemas-microsoft-com:vml" id="theText" style='behavior:
url(#default#VML); display: inline-block; position: absolute; width: 600px; height: 150px;
top: -5; left: -10; border: 0; z-index: 2; '>
               <div>
            <![endif]-->
               <table width="600" border="0" cellspacing="0" cellpadding="0" style="border-
collapse: collapse;">
                   <td height="150" align="center" valign="top"
style="color:#ffffff;font-size:20px;">
                           <span>Text</span>
                       </t.d>
                   <!--[if gte mso 9]>
               </div>
               </v:shape>
           <![endif]-->
```

Tested in Outlook 2010 thru Windows 7 VMWare on OSX 10.9.2



Creating spacers

Some email clients (notably Windows Desktop Outlook) collapse tables with no content, even if there are dimensions or padding.

To prevent these clients from collapsing the spacer table, you can add an invisible <code> </code>; that serves as content. Then zero out the character's <code>font-size</code> and <code>line-height</code> so it doesn't impact the <code>s</code> dimensions. The height (or width) attribute is now the only thing affecting the spacer's dimensions.

It was common practice to use a 1x1 transparent .gif as a spacer, but this method is deprecated.

Outlook can sometimes add a bit of spacing on the left and right side of a element that can cause some layout-related headaches. By using the vendor-specific mso-table-Ispace and mso-table-rspace CSS properties, you can be rid of those spaces and continue on to tackle the million other problems caused by Outlook.

```
table{
    mso-table-lspace:Opt;
    mso-table-rspace:Opt;
}
```

Image Resizing

Using width or height tags to resize images in your markup can create a problem in Internet Explorer browsers. If your reader is viewing an email inbrowser, and that email happens to have fluid images in it, they'll look pretty ugly as they resize. Using -ms-interpolation-mode:bicubic; ensures that your images look a little better.

```
img{
    -ms-interpolation-mode:bicubic;
}
```

Read HTML email composition and Outlook intricacies online: https://riptutorial.com/email/topic/5377/html-email-composition-and-outlook-intricacies

Chapter 5: HTML Email Reset

Introduction

As with web development, it's good to provide some reset CSS to the emails. As when building a web page, adding a CSS reset to an email helps to normalize how the code gets rendered across multiple email clients (since each has it's own default styles and way of interpreting code).

Examples

Resetting Email CSS in

Element Spacing