How To Code HTML Emails

How HTML email works, basic concepts, best practices, tips and tricks

Want to learn how to code your own HTML email campaigns? You’ve probably googled all kinds of web pages that give you countless "what works, what doesn’t" charts. They tell you which CSS definitions break, how Lotus Notes never renders HTML properly, and how Outlook can’t send email campaigns right.

But instead of focusing on specific tactics, let’s go over some fundamental principles.

Prerequisites

One thing we stress is that in order to code your own HTML email, you really, really, really need to know how to code HTML. You should be able to code web pages “from scratch” without the help of any WYSIWYGs (like Front Page, or even DreamWeaver). If you’re that good, then you really don’t need to worry about a million little rules (like what CSS definitions work in this email program, but not in that email program). Just being able to understand "the fundamentals" will save you a lot of time and frustration.

Not an expert coder? No worries. Most email marketing services provide you with built-in templates that you can use instead. MailChimp actually gives you an HTML Email Designer that you can use to create as many HTML emails as you want, without even knowing how to code.

HTML Email In A Nutshell

An HTML email is nothing but a web page. That’s it. Sorry if you thought there was more to it than that. So if you can code your own web page, you can code your own HTML email templates. There is a little catch, though. You have to code like it’s 1996 (explanation later).

HTML Email: Proper Delivery via Multipart/Alternative MIME

What makes HTML email so hard for most people is not the designing and coding part. It’s actually delivering it properly. You can’t just attach your HTML file and images to an email and send it. When people open the HTML file, your images will all be broken (because their email program has stored them in a temporary folder somewhere on your hard drive.

Even if you do manage to get your HTML email to display properly, there will be some recipients who can’t (or won’t) view HTML email in their email programs. So you need to send your HTML email along with a plain-text alternative version of your message. Then, your recipients’ email programs will automatically determine which format to display. In the early days, email marketing services would call this "HTML email sniffers." As in, "our solution comes with sniffers that automatically detect which version to display for your recipients." The truth is, the "sniffer" is on the recipient side, not the sender. It was just easier to explain that way. Nowadays, just about everybody can receive HTML emails. Some still choose not to.
How To Code HTML Emails | MailChimp.com

Coding Your HTML Emails: Fundamental Principles

Like I said earlier, coding an HTML email is basically like coding a web page. Only you've got to do it the old fashioned way. Remember back in the 90's when there were no WYSIWYGs yet, and you had to code everything by hand? Remember the Internet Explorer vs. Netscape wars? Remember how you had to test your work over and over and over again? HTML email is a lot like that. Times 10.

1. The most important thing (if you wish to preserve your sanity) is to keep it simple. Focus on your message, not your craftsmanship.

2. Images should be posted on your publicly accessible web server. In your code, use absolute paths to point to them. Attachments are often stored in randomly named temporary cache folders by some email programs. They also hog a lot of bandwidth (especially when they bounce) so attachments are not the way to go. Tip: One common mistake is to post your image files on an intranet, extranet, password-protected server, a secure server that's extremely slow, or a free hosting site (images my not show due to bandwidth issues). Post images on your fast, public-accessible web server.

3. TABLES and SHIM.GIFs are your friend. Keep it simple, because email programs use different HTML rendering engines. Some of them use Internet Explorer, Firefox, or in the case of Outlook 2007, Microsoft Word (what a headache!). Some of them use their own proprietary renderers (Lotus used to do that, which is why you'll find lots of old rants out there about Lotus and HTML email problems).

4. Don't code your emails too wide. Most people view messages in their preview panes, which are narrow and small. Don't ask me if you should code for 1024x768 screens or 800x600. Puh-lease. This is email we're talking about here. The preview pane in AOL 9 is less than 200 pixels wide, my friend. Think small. The templates we design at MailChimp are never more than 600 pixels wide, or they're fluid-width.

5. Test how it renders. Your email will display differently in all the different email programs out there. So testing is a must. And we're not talking about IE, Firefox, and Safari. We're talking about Outlook, Outlook 2007, Outlook 2003, Outlook Express, Thunderbird, Apple Mail, Eudora, Lotus, Gmail, YahooMail, AOL, AOL Web, Hotmail, MSN, Comcast, Earthlink, and on and on and on. How to cope? Keep it simple. TABLES. 600 pixels wide max. You can also test how your email renders in major programs with the MailChimp Inbox Inspector.

6. Webmail services are deceptively tricky. Email services that are browser-based, like Gmail, YahooMail, Hotmail, etc., will strip out your DOCTYPE, BODY, and HEAD tags. Makes sense---they don't want your code to potentially override theirs. Anything you'd normally code inside those tags (bgcolors, embedded CSS, JavaScript, background music files, etc) will also get stripped.

7. Think like a spam filter. You have to consider spam filters and spam firewalls when you code. It's pretty obvious that spammy words like "Viagra" or "V14GR4" will get you spam filtered. But spam filters also look for stuff like, "do you have too many images, and not enough text?" If you're sloppy with your code, you'll look like a sloppy spammer. See: How spam filters work.

8. 99% of your CSS won't work (especially not in the browser-based email services like Gmail, YahooMail, etc.). That means no CSS-positioning, DIVs, etc. Before you ask me for a detailed list of what works and what doesn't, just remember this (it'll save us all a lot of time): When your email is opened in Gmail, for example, the CSS in your HTML email could potentially override the CSS on the rest of their page. So they disable a lot of it. Inline CSS is safer, and plain-old FONT tags are safest (code like it's 1996, remember?). Or check out MailChimp's CSS inliner tool.
How Email Open and Click Tracking Works

If you’re sending HTML email, chances are you’re doing it for marketing of some sort. If that’s the case, you probably want to track opens and clicks. This functionality is built right into email marketing solutions like MailChimp, so you really don’t have to do anything. But it helps to know how it works.

To track an “open” in an HTML email, you’d embed a tiny, 1x1 pixel transparent .GIF at the bottom of the message. Some people call this a “tracker image” or “web beacon.” Actually, you probably don’t want it to be 1x1 pixels, because that’s a little too common. Some anti-virus packages automatically block images that are 1x1, because they know it’s a beacon. Anyways, whenever your recipient opens his email, the tracker image is downloaded from your server (and you can count that as an open). This is why most people will tell you that you can’t track opens unless it’s an HTML email, and why open tracking must be taken with a grain of salt. That’s only partly true.

It’s true that only HTML email can have embedded tracking images. But you can track “opens” from a plain-text email too. MailChimp lets you enable click-tracking in your plain-text emails. Whenever someone clicks a link in their plain-text email (or if they leave images turned off while viewing your HTML email) we can track that as an “open.”

That brings us to click tracking. You’ve probably already figured this one out. Just use redirect scripts. MailChimp automatically converts all links in your emails to point to our redirect scripts. When a recipient clicks your link, they’re redirected from our server to your original URL (in the blink of an eye) and we track the click. That helps us generate nice little campaign reports telling you which links your recipients clicked most.

Proper List Management

I can’t just teach you how to code and deliver HTML emails without telling you at least a little bit about proper list management. If you’ve made it this far, you’re probably the type of do-it-yourselfer who is managing your own email list in some database on your server. Right? That’s fine, but if you don’t manage your list properly after every campaign you send, you could get blacklisted by ISPs and anti-spam groups.

Here’s what you need to know about managing your email lists:

- First of all, don’t send mass email marketing to any lists unless the recipients gave you permission. The key word there was “marketing” not “transactional email” or “personal messages.”
- After every campaign you send, you will get bounce backs. There are two overall types of bounces. A "soft bounce" means the recipient was "temporarily unavailable" (account overquota, server down for maintenance, etc). A “hard bounce” means the email was undeliverable. The account has expired, it never existed, etc.
- You should remove hard bounces immediately, because if you keep sending emails to a server, after that server has told you repeatedly “hey buddy, that email doesn’t exist here!” they will most likely start blocking your future messages (and telling other servers out there what a jerk you are, and that you’re probably an infected zombie).
- If an email soft bounces 3 campaigns in a row, they say you should clean it from your list. That’s best practice, But we’ve found that if you send daily emails, you’re going to lose A LOT of subscribers while they’re on vacations that last more than 3 days. Even if your list is huge, and you can afford to lose a few subscribers like that, you’ll probably get annoyed at all the “hey, I’m not getting your emails anymore” complaints.
- Unsubscribe requests should be handled immediately.
- For more information on proper list management, check out these suggestions from MAPS.

By the way, you can setup a list in MailChimp, and we’ll manage it for you (removing bounces, handling unsubscribes, etc). Then, you can totally customize the signup form and opt-in process to match your branding. Or, use our API to pass opt-in data directly from your website to MailChimp.

The Future of HTML Email

Over the years, we’ve seen lots of technology come and go that could “threaten the future of HTML email." Stuff like blacklists, anti-virus programs that block web beacons, aggressive spam filters, email programs that turn images off by default, Outlook 2007, yadda yadda. But HTML email is here to stay. Instead of obsessing over all the email programs and coding tricks and CSS hacks and Flash compatibility, just keep your designs simple and focus on your message. That being said, there are some things you will need to keep your eye on in the near future:

- More and more people are switching to browser-based email services, like YahooMail and Gmail. Those guys are using AJAX and other cool stuff to make their interfaces behave just like your desktop program. And heck, why not put the burden of spam filtering on some hosted solution, instead of your own computer, right? Those services have varying levels of HTML email and CSS support. And they need to make a buck, too. Think about the cost of...
managing millions (billions?) of email accounts, their bandwidth, their storage, etc. Their spam filters will get more and more aggressive and creative. They’re already starting to require authentication, and will likely gravitate more and more to certified email programs like SenderScore, Habeas, and Goodmail. If you’re a big sender, you’ll need a dedicated IP address and you’ll need to seek certification soon.

- More and more people are using mobile devices to check their email. Blackberries and cell phones render HTML email in strange ways. They often strip out everything but text and a few images (and display them vertically, too). Does that mean we’ll see a shift back to plain-text?
- Not really. Apple’s iPhone will display web pages in full, and let you “zoom in” to specific areas on the screen. It’ll probably handle HTML emails beautifully. See, with email marketing, it’s always two steps back, then one step forward.
- Email Reputation and Certification will be very important soon. Large email senders will want to seek Goodmail, or SenderScore, or Habeas certification. I can’t predict how prevalent they’ll be or how soon (this article has some info on that) but if you plan to do a lot of email marketing, start studying up on those services now.

Related Articles
- Can I provide my own HTML for a campaign?
- HTML code for footer tags
- My emails are being delivered as broken HTML code or an attachment.