



# GUIDE TO CODE KILLER **RESPONSIVE EMAILS**

THAT WILL MAKE YOUR  
EMAILS BEAUTIFUL



## Create flawless emails 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 best practices to follow and advices on how to go around the most common problems.



*Keep your emails without a scratch,  
while they go through all inboxes, towards your happy readers.*

# BEST PRACTICES TO **CODE RESPONSIVE EMAILS**

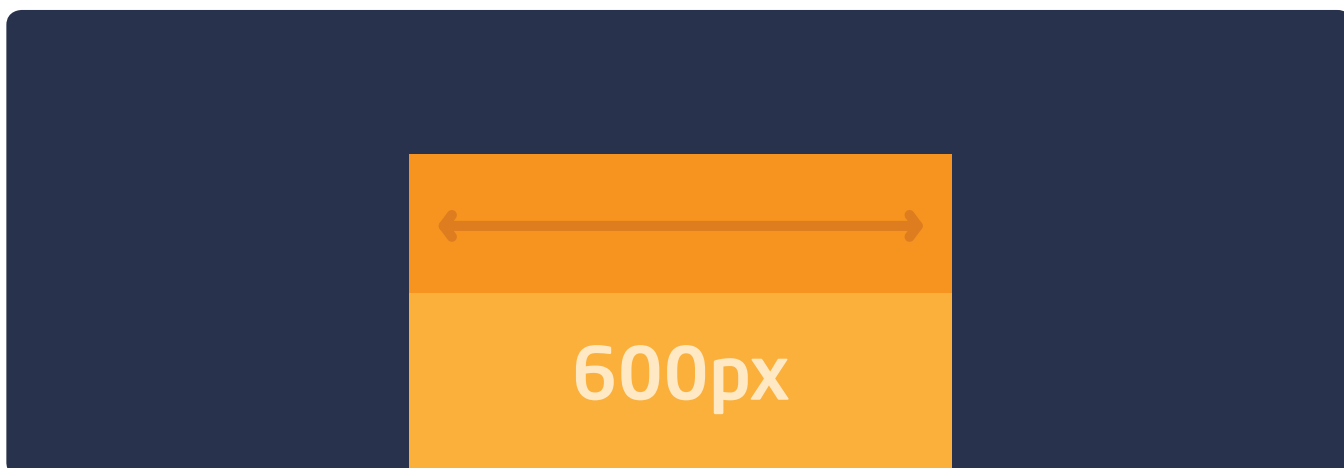
*Here are listed some best practices that you can work with to create your own optimized email.*

**DO NEVER FORGET THE MOST IMPORTANT THING:  
ALWAYS TEST YOUR DESIGNS**

## 1 - KEEP EMAIL WIDTH AROUND 600 PIXELS

Emails are opened in a variety of clients. Many people view them in the preview window which has a limited size that depends on the email client and the size of the screen. To limit your layout's width you should wrap your content in a container table with a fixed width.

We recommend designing your emails with the standard width which is around 600 pixels. This way you're optimising your emails to display the full content in the preview panel.



## 2 - ABOUT TABLES

### USE TABLES, NOT DIVS

For your layout to render correctly across all email clients, you have to use HTML tables, which offers the most consistent support.

Email coding defies what most web designers use nowadays like divs, sections, nav, footer or other tags for building webpages. Using tables for web development was the old-school way. We need to go back in time because most email clients have poor support for the use of <div> tags.

*“Design your email like it’s 1999.”*

### NESTING

When coding your table structure keep in mind that some email clients have poor support for margin, padding, positioning, and float. In order to solve this we advise you to nest all your content and tables inside a structural table. Also it helps to assure that your elements adapt to a shrinking or expanding screen as one, instead of individually.

```
<body>
  <table border="0" cellpadding="0" cellspacing="0" width="100%" class="className">
    <tbody>
      <tr>
        <td align="center">
          <table border="0" cellpadding="20" cellspacing="0" width="600" class="className">
            <tbody>
              <tr>
                <td align="center"> Your text here. </td>
              </tr>
            </tbody>
          </table>
        </td>
      </tr>
    </tbody>
  </table>
</body>
```

## SET WIDTHS

In order to achieve better results you have to specify widths for both tables and cells. Otherwise email clients may assume some strange values. Whether you use pixels or percentages will depend on what's best for your layout structure. For example if you nest your tables based on percentages, then when you adjust the width of the parent element, everything will adapt accordingly.

```
<table width="600">
  <tr>
    <td width="250"> Column 1 </td>
    <td width="100"> Gutter </td>
    <td width="250"> Column 2 </td>
  <tr>
</table>
```

```
<table width="100%">
  <tr>
    <td width="250"> Column 1 </td>
    <td width="100"> Gutter </td>
    <td width="250"> Column 2 </td>
  <tr>
</table>
```

```
<table width="600">
  <tr>
    <td width="40%"> Column 1 </td>
    <td width="20%"> Gutter </td>
    <td width="40%"> Column 2 </td>
  <tr>
</table>
```

## BACKGROUND COLOR TABLE

For setting a background color in your HTML email you'll need to wrap all your email structure into a table 100% wide where you can set a background color. The traditional way of assigning the background color to the <body> tag won't work in many email clients. To achieve better results specify the color you want with "bgcolor" or "background-color".

## 3 - CSS

### DON'T USE EXTERNAL STYLESHEETS

External Stylesheets aren't a wise choice because some email clients will take them out of the <head> and <body> of any email. So your best option is to use CSS inline.

### YOU MUST USE ALWAYS INLINE STYLES

Many clients do not support CSS classes. Instead of creating classes you have to use inline CSS. Everything that needs to be stylized has to be specified on individual styles placed on <td> or in the <table> tag, such as font type, color and size. Despite all the repeated code, this is the safest way to ensure that your email renders as you want, preserving the consistency and structure of your layout.

MailChimp has a CSS Inliner Tool that automatically inline your email's CSS.  
<http://templates.mailchimp.com/resources/inline-css/>

```
<td style="font-size: 20px; font-weight: bold; font-family: Arial, sans-serif; line-height:40px; color: #333333">Lorem Ipsum</td>
```

### DON'T USE SHORTCUTS

Support for shorthand CSS in font, padding and defining hex values is generally inconsistent. They won't render properly. You need to write each style property separately, like font-size, color, font-family, etc. Use the full six digits for HEX color codes and avoid RGB colors. For best results go for the extended approach.

```
<td style="font-style: italic; font-size: 14px; font-family: Verdana; color: #333333">
```



```
<td style="font: italic 14px Verdana; color: #333">
```



## HEADINGS & PARAGRAPHS

Heading tags should be avoided as they have inherent styling which renders differently across the different email platforms. The use of paragraph tags can also reveal some inconsistencies across email clients with spacing and other formation. Avoid them and place the text within `<td>` cells instead and apply inline styling to control the text.

`<td style="font-size: 14px; font-family: Verdana; color: #333333"> Text </td>` ✓

`<h1 style="font-family: Verdana; color: #333333"> Text </h1>` ✗

## 4 - IMAGES

### ALT TAGS

Some email clients do not automatically load images, this means that they are disabled by default. You can get around that situation by giving them an alt text.

When images are suppressed, most email clients will display the defined alt text instead, providing some context for the empty area. For even better results, you can use inline styles on the `img <td>` such as font style, color and size. This will allow you to have more control of your overall email design when images are off.

### HOSTING

To include images on your email template, first you need to host them on a public accessible server. This way your images will be available on the web so that your email readers can see them. Then you need to insert absolute urls, into your HTML code, with your image location in order to display them properly.

absolute image url - `` ✓

relative image url - `` ✗



## DIMENSION & SIZE

Before adding images to your email, confirm that they have the correct size that you want them to display. This will prevent some clients that tend to ignore your specified size and resize them to it's real dimensions. Also make sure that the file size is as small as possible, while maintaining their quality, to prevent long loadings.

Declaring their width and height will ensure that some email clients won't resize them when the images are turned off by default, keeping your layout structure as you planned.

## BACKGROUND IMAGES

Avoid using background images, usually their stripped out by some email applications.

Background images aren't supported in Outlook 2007 and Outlook 2010, Yahoo! Mail, Hotmail and Gmail. Avoid them or try to use a background color as a backup for when background images are disabled.

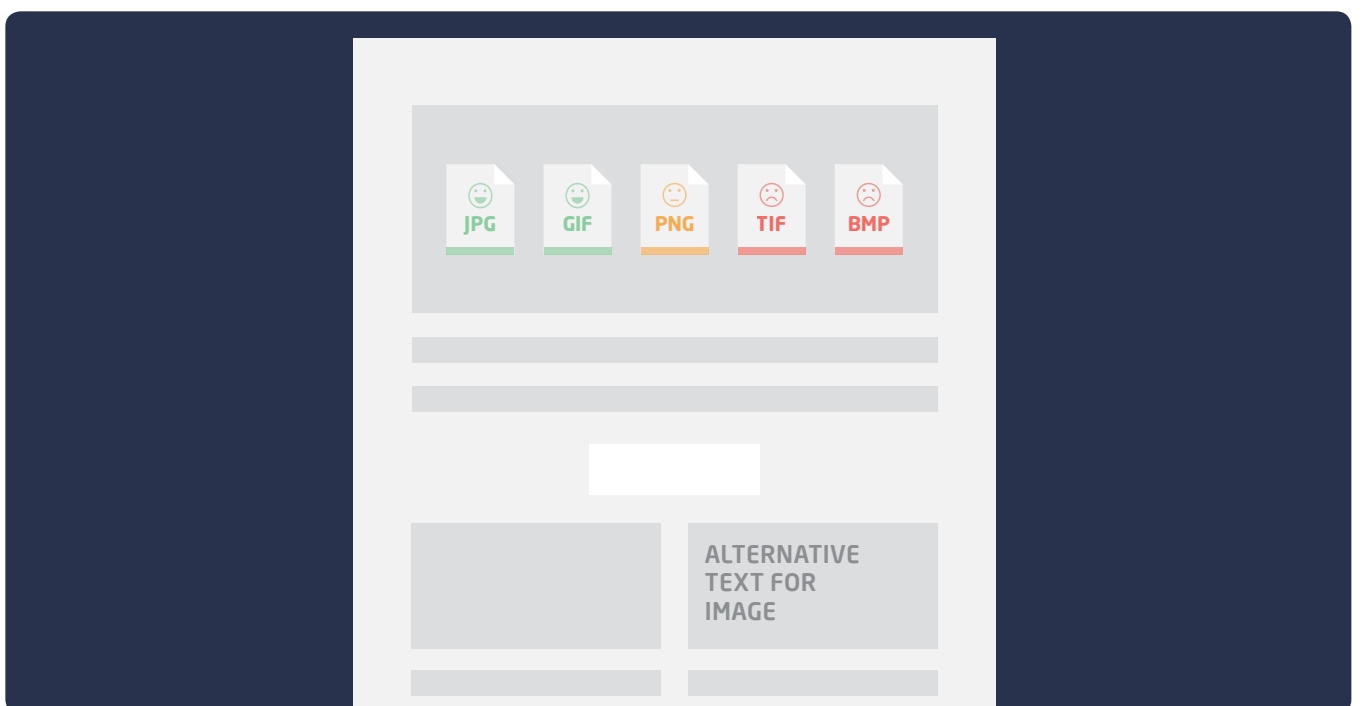
However, Campaign Monitor has developed a way around this problem. It'll allow you to add a background image to a table cell while working in most major email clients.

<http://backgrounds.cm/>

## SUPPORTED FORMATS

JPG and GIF are the best image formats to use in your emails, once they're supported by all email clients. You can also use PNG, but keep in mind that it's not supported by all web browsers and on early versions of Lotus Notes.

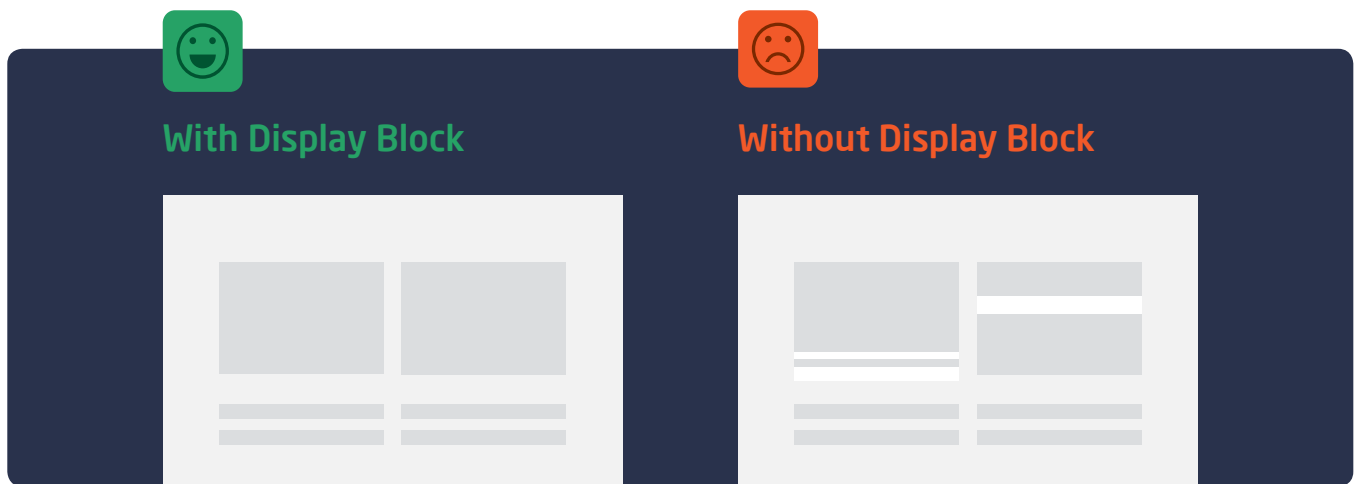
Avoid using TIF and BMP image formats once most email clients don't support them.



## DISPLAY BLOCK

For your image to render properly you should add the "display: block;" style command. Some email clients requires it to be specified. Otherwise they can create gaps around images.

```
<td align="left" class="main-image"> <a href="#" style="display: block;">  </a> </td>
```



## DON'T USE FLOATS

Some email clients don't support the float property. Instead, use the align attribute to float your layout elements.

```
<td align="left" class="main-image"> <a href="#" style="display: block;"></a></td>
```

## 5 - JAVASCRIPT, FLASH & VIDEO

Email clients have limited support for Flash, TypeKit, JavaScript or other dynamic scripts. So stick with the plain HTML and CSS.

Avoid using forms, or search boxes as they'll probably get disabled by email clients. If you want to show some animation, music or make a survey, you should guide your readers to the CTA button that will lead them to your landing page, where you can include what you wish.

Although, it turns out that some email clients are starting to give support to HTML5 video, including Apple Mail, Entourage 2008, MobileMe and the iPhone. If you want to know how to embed a video on your email you should check campaign monitor technique with code samples:

<https://www.campaignmonitor.com/blog/post/2905/html5-and-video-in-email/>

## 6 - FONTS

The best method, to achieve reliable results on most email clients, when it comes to typography, is to use web fonts. It allows you to use fonts that aren't installed on the computer's reader by including them in your CSS. Campaign Monitor testings, in 2012, showed that Google Web Fonts is the best option available. Some email clients don't have support for web fonts, so make sure that after you've specified your font, you provide a widely-used alternative font such as Helvetica, Arial or a generic sans serif.

To include a web font in your email, we recommend the HTML approach, using `<link>` to pull in your chosen font.

The font of each block of text must be defined separately and inline. If your fonts were specified in the `<head>` of your email, some email clients will take that out, leaving you with their own default fonts.

```
<link href="http://fonts.googleapis.com/css?family=FontName" rel="stylesheet" type="text/css"/>
```

```
<td style="font-family: FontName, Arial, sans-serif; font-size: 16px; font-weight: normal; line-height: 30px; color: #333333;"> Some text here. </td>
```

## 7 - EMAIL FILE SIZE

Don't forget to check your overall email file size. It can affect your message's deliverability and how it looks once it gets to the inboxes. For example, Gmail will render the first 102K of your email and clip off the remain.

Email on Acid did some testings to find witch is the maximum size for emails to go through spam filters without problems. Their conclusion was that emails under 100KB successfully passed through spam filters. Images do not interfere towards this total count. So pay more attention to the email's final size than image's size.

## 8 - MAKE IT RESPONSIVE

### WHAT IS RESPONSIVE

Responsive Email Design optimizes your campaign to display differently according to the screen size of a wide range of devices. With the big rising of email opens on mobile - around 53% - it's important to design a layout that works well in different factors. A study from MailChimp revealed that responsive design is responsible for a 15% increase in clicks for mobile users. This happens because in small screen sizes, if the email isn't responsible, the design breaks down, the email is zoomed to fit, sometimes requiring horizontal scrolling. Also text is often difficult to see and links hard to click.

With Responsive Design you can change things such as hierarchy, navigation, colors and layout, also add padding, change or hide content, enlarge fonts and scale images and buttons, depending on the screen width that the email is open.

To make this possible, responsive emails uses media query, witch is a special set of CSS styles, that act like conditional statements or dynamic rules.

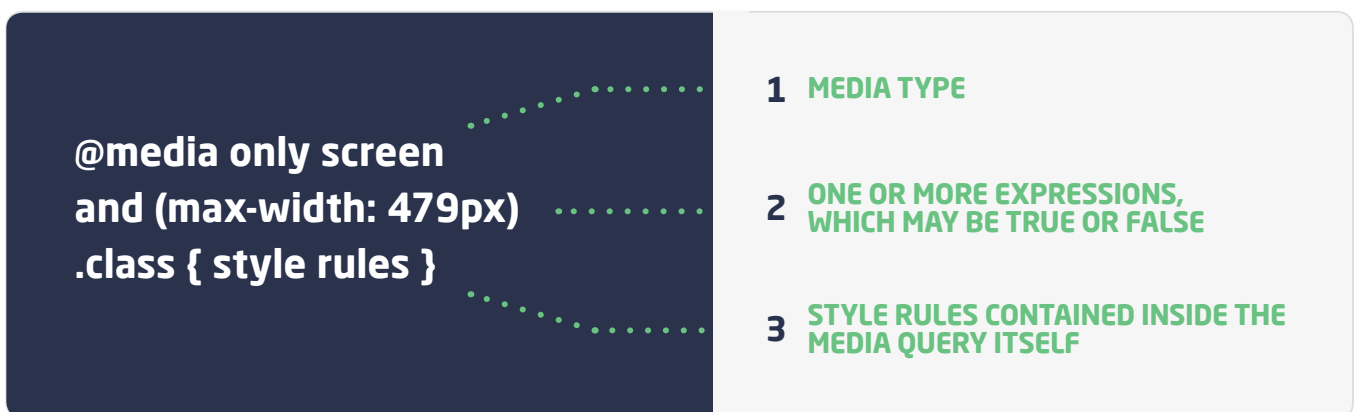
With the help of responsive design, you can simplify your message and make it easier to read and interact with on the small screens of mobile devices. Responsive email design offers benefits such as increased usability, higher read rates, along with better performance and click-through rates.



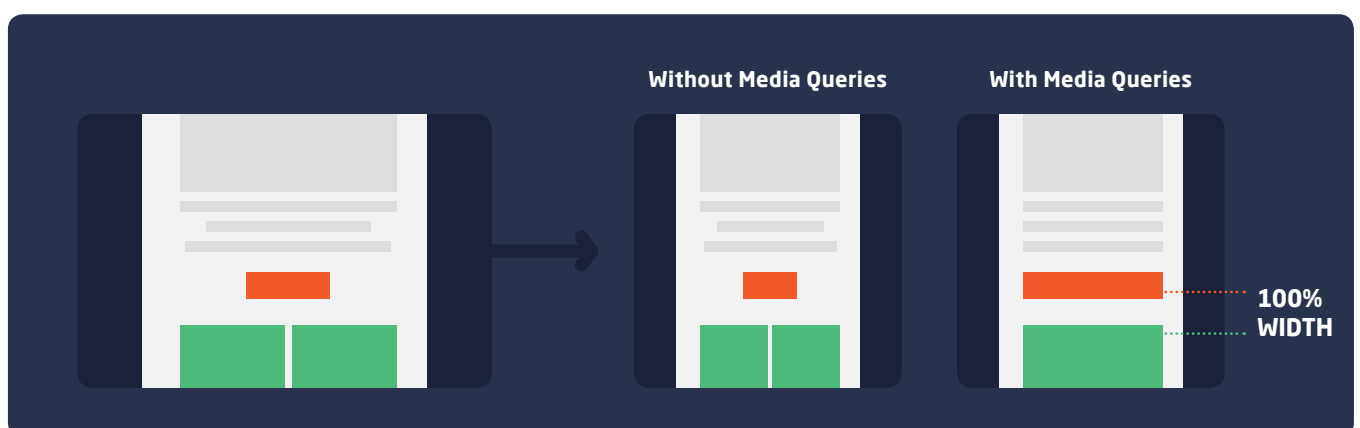
## HOW MEDIA QUERIES WORK?

A media query (@media) is a component of cascading style sheets (CSS). It allows your email content to render in different ways according to the screen resolution. This happens because Media Queries acts like a switch for triggering a set of CSS for a specific set of rules. Media queries detect the screen size of a device and then “turn on” different sets of rules based on that screen size.

### MEDIA QUERIES ARE COMPOSED BY THREE PARTS:



If the media type specified in the media query matches the type of device where the email is being displayed, then all expressions in the media query become true. When this happens, the corresponding style rules are applied overlaying the rules defined on your HTML body. You can insert the media query's styles within the email <style> tag. You can specify as many styles as you want for each screen size.



The style rules have to be specified by targeting class names or specific elements such as an entire table or just a table cell which style changes should affect. In the case below, the element (td) and the class "main-header" used in the media query must be according to those used in the table's td.

```
<body>
  <table border="0" cellpadding="0" cellspacing="0" width="600">
    <tbody>
      <tr>
        <td class="main-header" style="font-size: 14px;"> This is a the header </td>
      </tr>
    </tbody>
  </table>
</body>

<style type="text/css">
  @media screen and (max-width: 600px){
    td[class="main-header"]{
      font-size: 19px !important;
    }
  }
</style>
```

## DON'T FORGET THE VIEWPORT META TAG

For Media Queries to work, you'll have to include the viewport meta tag on your email <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.

```
<meta name="viewport" content="width=device-width">
```

## !IMPORTANT

Media Queries are used to override inline styles. To determine which one the browser will render, you need a way to make its priority clear. This can be achieved using the !important declaration.

```
@media screen and (max-width: 600px) {  
  td[class="main-header"] {  
    font-size: 20px !important;  
  }  
}
```

## BREAKPOINTS

Breakpoints are used to define the screen resolution at which email clients receive instructions to change from one email layout to another. This is usually established by one, or more, specific widths, in pixels.

There is no rules for defining the number of breakpoints. It depends on your individual email design and how you want to optimize the experience for your readers. But to make your email responsive you'll need at least one breakpoint, with a minimum of 414px, which is the width of the new iPhone 6 Plus.

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.

### MOST COMMON BREAKPOINTS:

XL Screen – > 1920px

Desktop & Laptops – 1024px

Tablets – 768px to 1024px

Smartphones – 320px to 480px

## MEDIA QUERIES SUPPORT

Keep in mind that Media Queries aren't supported by all email clients. HTML email will display differently depending on which application is used to view the email. For example, Gmail will not show the responsive version of your layout on any platform.

In order to get around this issue, we advise you to make your emails mobile friendly from scratch. Keep your email structure simple with fluid layouts and pay attention to details, such as fonts and buttons sizes.

If you're comfortable with advanced coding techniques, take a look at the "hybrid coding approach". It is based on making the email out of flexible percentages. Learn more about how to implement this approach here: <http://labs.actionrocket.co/the-hybrid-coding-approach>.



### Not supported by:

Gmail app (iOS + Android)

Mailbox (iOS + Android)

Inbox Gmail app (iOS + Android)

Outlook.com (Android Browser)

Android Outlook Exchange via native client

Yahoo! Mail (Android Browser)

Android Yahoo! Mail app

Windows Phone 7

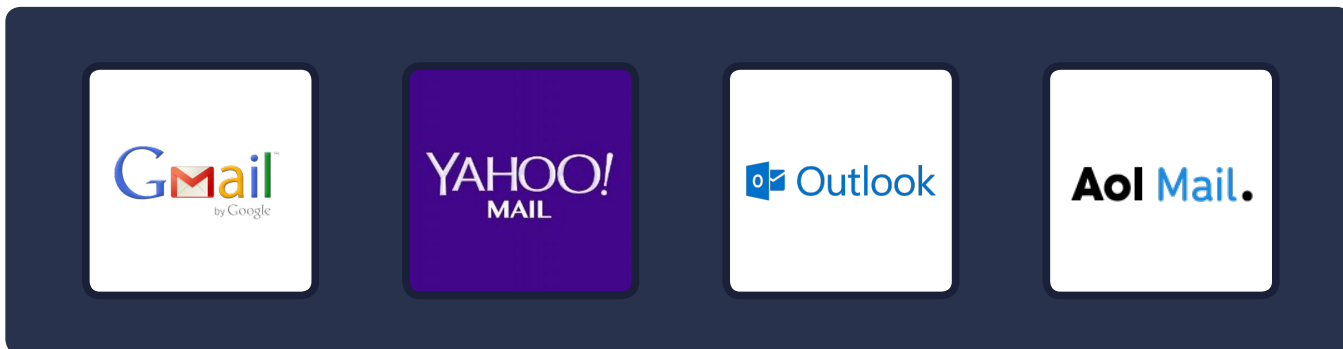
Gmail (Android Browser)

Windows Phone 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.





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](http://www.emailbidding.com)