

RESPONSIVE EMAIL DESIGN





© SELLIGENT

The contents of this manual cover material copyrighted by Selligent.

This manual cannot be reproduced, in part or in whole, or distributed or transferred by means electronic or mechanical, or photocopied, without the prior written consent of a representative from Selligent.

Glossary

TERM	DESCRIPTION
<u>Responsive design</u>	An approach aimed at crafting websites/e-mails to provide an optimal viewing experience (easy reading and navigation with a minimum of resizing, panning, and scrolling) across a wide range of devices.
<u>Metatag</u>	An HTML tag that is used to store information about a page but is not displayed in the browser.
<u>Viewport</u>	The viewport is a virtual area used by the browser/e-mail rendering engine to determine how content is scaled and sized when it is initially rendered on the current screen.
<u>Media Query</u>	Consists of a media type and at least one expression that limits the style sheets' scope by using media features, such as width, height and color. It lets the presentation of content be tailored to a specific range of output devices without having to change the content itself.
<u>Initial-scale</u>	This property will control the zoom level when the page is first loaded.
<u>Maximum- & minimum-scale</u>	This property will control how users are allowed to zoom in or out.

Table of contents

1. Purpose	4
2. Email design	5
2.1 WRONG APPROACH	5
2.1.1 SCALING	5
2.1.2 SKINNY DESIGN	5
2.1.3 UNDERSTAND YOUR AUDIENCE	6
2.2 EMAIL LAYOUT	6
2.2.1 STICK TO A SINGLE COLUMN	6
2.2.2 EXAMPLES OF HOW IT SHOULD BE DONE	7
2.3 MOBILE BEST PRACTICE	8
2.4 DOCTYPE	8
2.5 VIEWPORT RENDERED UNUSABLE	9
2.6 MEDIA QUERIES	10
2.6.1 MEDIA QUERY SUPPORT	11
2.7 THE IMPORTANCE OF “!IMPORTANT”	12
2.8 RESET THE BODY	13
2.9 RESIZE FONTS	13
2.10 HYPERLINKS	13
2.11 LIQUEFY YOUR DESIGN	14
2.12 SLIM DOWN YOUR CONTENT	15
2.12.1 THINK IN PACKAGES	17
2.13 MULTIPLE COLUMNS ON ONE SCREEN	18
2.13.1 REMOVING A SPACER TD	19
2.14 RESPONSIVE IMAGES	19
2.15 TEST YOUR RESPONSIVE DESIGN	20

1. Purpose

The future of email marketing lies in the ability to adapt to your surroundings. Everywhere we look we see smartphones, tablets... These smaller devices each have different screen resolutions compared to the traditional desktop clients. When your email is not optimized for representation on a smaller screen your mobile device will try its best to visualize the email by introducing a scrollbar or scaling the content. In both cases the mobile device is trying to fit what was not intended to be displayed on a smaller screen. The art of optimizing an email for representation on multiple screen sizes is called responsive email design, and it is the main focus of this guideline document.

Mobile devices have almost won the race over desktop devices as preferred device to read email on. In september 2013 Litmus reported that 47% of all emails opened are opened on a mobile device. If you know that half of your audience isn't reading your email as it was intended, the need to change becomes apparent. And change can be good for business too, because:



MOBILE USERS WILL DO ANYTHING AND EVERYTHING DESKTOP USERS DO. PROVIDED IT'S PRESENTED IN A USABLE WAY.

We know that 63% of Americans and 41% of europeans would close or even delete an email that isn't optimized. so we need to adapt and change our emails to keep the reader's focus.

We need to present our content in a usable way. Meaning important buttons should be displayed larger so that users can click on them with ease. Images should be smaller in file size so that download time is reduced on a mobile connection. These and many others are examples of what we need to improve.

In this guideline document we attempt to give you some tips, tricks and best practices for setting up effective responsive email templates.



2. Email design

So what exactly does responsive design entail? It's a set of techniques and principles applied to email design that includes media queries, fluid layouts and images, and customized copy making for emails suitable for both desktop and mobile devices. Responsive techniques allow designers to hide, stack, expand/collapse or modify content to optimize email content display on smaller screens. It also includes simple considerations to make emails easier for mobile users to consume, such as large buttons for easy tapping, layouts that follow a hierarchy for content so they work even without image display enabled...

Creating responsive emails can be done in many ways depending on the final result you wish to achieve. You could choose for the fixed design option where you're transforming all the pixels to the correct size. For example when having a table width of 600 pixels and you would like to have it fit the 320 pixels screen resolution of a smartphone you will transform all the 600 pixels to 320 pixels and so on.

The other option is to choose for a liquid design option, which is more flexible than the fixed design option. Just like in the fixed design, the pixels are going to be transformed, but this time to percentages. So a 600 pixel design will be transformed to a 100% design which will fit the screen of the device automatically.

2.1 WRONG APPROACH

2.1.1 SCALING

Scaling allows creating a complete fluid design which will adapt to the screen size. However, since text size, images, etc. are not adjusted, the email will be exactly the same as the desktop version - only smaller.

Users will have problems reading the text, clicking the buttons and will probably close the email before reading it. This method is not user-friendly and is not recommended. However this method should work on every device because you are working with a fluid design.

2.1.2 SKINNY DESIGN

Skinny design is basically a design small enough to fit a mobile screen; it will remain in this state for the desktop users. This will result in a very minimalistic desktop email, it is a bit user-friendlier than the previous method but it's still not a real responsive design.

However in this case the desktop users will be limited and potential important (extra) information will never be shown to the user because there is simply not enough space to put this information.

2.1.3 UNDERSTAND YOUR AUDIENCE

From the two methods above scaling is the least “wrong” but as mentioned still not a “true” mobile design. However, since this method will work everywhere, you should still consider this when selecting a strategy. It is important to understand your audience and which environments are most frequently used to view your emails. Once you have this data you can determine your investment into your mobile strategy.

However, as a best practice, you should use BOTH. A scalable layout that turns responsive. The rest of this document is focused on this ideology. We will take the best of both worlds and try to combine them into a responsive email design.

2.2 EMAIL LAYOUT

When starting from scratch it’s important to keep the responsive part in mind at all times, especially when you’re still in the design stages. There is nothing more important than a design that is optimized for desktop and mobile, this will make the upcoming process a lot easier.

A good way of working is to first think about a mobile design and create the desktop version based upon this design. Working the other way around will limit your options for the mobile users. Try to keep your mobile version as sleek and minimalistic as possible. The more images and special designs you use the harder it will be to have a decent email for mobile and desktop.

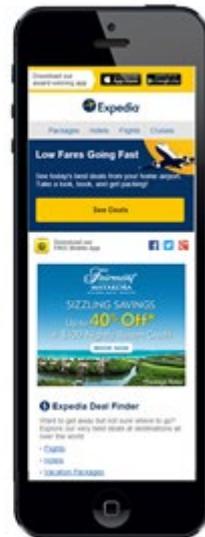
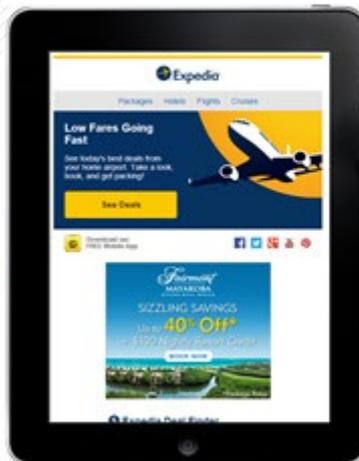
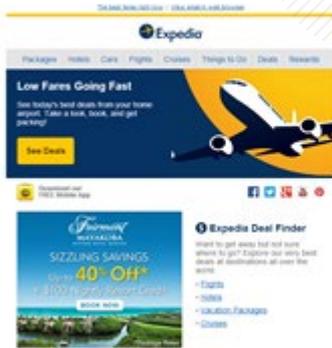
2.2.1 STICK TO A SINGLE COLUMN

To make the process of optimizing your newsletter as painless as possible, you should opt for a single column design. Because you need to rely on nested tables these days, it makes it very difficult to re-order columns or stack blocks of content on top of each other. This is relatively easy with CSS, but nearly impossible given the rigidity of HTML tables.

However, this doesn’t mean that the multi-column designs won’t work on mobile devices; it will just be harder to make them work on desktop and mobile. If you do go for a two- or three-column design, you still have the option of simply hiding certain columns altogether. If you’re using a column for secondary content that isn’t important, this might be a great option for your mobile version anyway.

More information on how you can work with multiple columns can be found further in this document.

2.2.2 EXAMPLES OF HOW IT SHOULD BE DONE



2.3 MOBILE BEST PRACTICE

- **Clear and concise content**: small screens mean it's more important than ever to engage the user as efficiently as possible.
- **Single-column layout**: simplicity is key, as mentioned above. Layout no wider than 640px will degrade gracefully. A single column ensures no content will be completely lost.
- **Large call-to-action (CTA)**: don't punish fat fingers! Apple's iOS Human Interface Guidelines recommend a minimum 'tappable' target area of 44x44 points. This will however differ from device to device.
- **Generous font sizes**: make sure your message can be read easily.
- **Pre-header**: another key area when it comes to visibility in the inbox. Try to avoid simply displaying 'view in browser' text.
- **Left-aligned text**: there are a number of reasons for aligning important elements to the left-hand side of the content area. eye tracking research suggests that western users focus the majority of their attention on the left-hand side of email content. This is hardly surprising since we read text from left to right. Certain operating systems, notably Android, will not scale content to fit the screen, therefore displaying only the left half of an email. from an ergonomic perspective, the majority of users will find it easier to interact with elements in the bottom left/middle of their hand-held screen.
- **Vertical hierarchy**: reduced screen real estate places more credence than ever in the idea of 'the fold'. significant CTAs should be placed as near to the top as possible; if they are not seen immediately, they will probably not be seen at all.
- **Use images carefully**: don't assume that images will be seen. The iPhone's native email app will display images by default but many clients won't. If your email doesn't contain enough text it might also be picked up by a spam filter.

2.4 DOCTYPE

Some email clients are known to strip out the DOCTYPE such as Hotmail and Gmail. They automatically insert the XHTML 1.0 strict DOCTYPE. It's not a bad idea to put it in whilst building your email, to avoid conflicts with Hotmail/Gmail in the testing phase.



```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//en" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
```

2.5 VIEWPORT RENDERED UNUSABLE

The viewport is a virtual area used by the browser/email rendering engine to determine how content is scaled and sized when it is initially rendered on the current screen. The viewport metatag was originally created by Apple; then it was implemented and continued by others. However for responsive emails the viewport metatag will only give you the illusion of control, browser support for this metatag is spotty and email client support is even worse.



EXAMPLE

```
<meta name="viewport" content="width=device-width, initial-scale=1.0, maximum-scale=1.0," />
```

Below you can find a list with viewport testing results on different mobile clients:

DEVICE	ALLOWS ZOOM BY DEFAULT	INITIAL-SCALE	MAXIMUM-SCALE	NOTES
<u>Android</u>	✓	✓	✗	Initial-scale seems to block the user from zooming out. Without the initial-scale, users can zoom out to approx. 5X. With and without the meta tag, it scaled up to the same max dimension (approx. 2X). seems like maximum-scale isn't supported.
<u>iPhone</u>	✓	✗	✗	This device operated the same regardless of viewport settings. Its maximum zoom is much higher than on Android (approx. 5X).
<u>Kindle Fire</u>	✓	✓	✓	Setting the initial-scale and maximum-scale to 1.0 basically disables the zooming functions for this device. setting the maximum-scale controls the maximum zoom-in and the maximum zoom-out.
<u>Xoom</u>	✓	✗	✓	Setting the maximum-scale to 1.0 basically disables the zooming functions for this device. setting the maximum-scale controls the maximum zoom-in and the maximum zoom-out. This device does not respond to media queries with the meta tag.
<u>iPad</u>	✓	✗	✗	This device operated the same regardless of viewport settings. Its maximum zoom is much higher than the Android (approx. 5X).
<u>BlackBerry</u>	⊘	⊘	⊘	Emails that are sent with <meta name="viewport" content="..." /> and anything in the content attribute appear to be blank when viewed on blackberry.

2.6 MEDIA QUERIES

MEDIA QUERIES

Media queries are the biggest part of responsive design. They will allow you to transform your email to a slim version on mobile devices. Media queries is a method also used by web designers to make their website responsive and they can also be used for email layouts. However, because media queries are a part of the style tag, not every email client will accept and show them, which will cause your mobile version to fail (not show).

Different types of mobile devices can have different types of media queries. This means that you can target different devices depending on their screen resolution. The following list will provide you with a couple of media query examples:

- **Smartphones – Portrait & landscape**: @media screen and (min-device-width : 320px) and (max-device-width : 480px)
- **Smartphones - Landscape**: @media screen and (min-width : 321px)
- **Smartphones - Portrait**: @media screen and (max-width : 320px)
- **iPads – Portrait & landscape**: @media screen and (min-device-width : 768px) and (max-device-width : 1024px)
- **iPads – Landscape**: @media screen and (min-device-width : 768px) and (max-device-width : 1024px) and (orientation : landscape)
- **Smartphones – Portrait**: @media screen and (min-device-width : 768px) and (max-device-width : 1024px) and (orientation : portrait)
- **Desktops and laptops**: @media screen and (min-width : 1224px)
- **Large screens**: @media screen and (min-width : 1824px)
- **iPhone 4**: @media screen and (-webkit-min-device-pixel-ratio : 1.5), screen and (min-device-pixel-ratio : 1.5)

There are a lot of different media queries that cover a whole lot more mobile devices than the list above. However creating a responsive design that will work 100% on every device is very hard to do. The best way to avoid setting up a whole list of media queries is to look at the most popular devices, or even better look at what the most popular devices are amongst your users. This way you have an idea on which devices should absolutely be targeted.



NOTE

Some of the new devices have a larger screen than the standard smartphones, which means that they could slip through the meshes of the net.

2.6.1 MEDIA QUERY SUPPORT

However not every default email client on mobile devices will support media queries, which means that unless you have a complete device list with the email app of your customers you will not be able to create an email that will be responsive on every mobile device.

DEFAULT DEVICE EMAIL CLIENTS

CLIENT	MEDIA QUERY SUPPORT
Amazon Kindle Fire	
Amazone Kindle Fire HD	
Android 2.1 Eclair	
Android 2.2+	
Apple iPhone	
Apple iPad	
Apple iPod Touch	
BlackBerry OS 5	
BlackBerry OS 6+	
BlackBerry Playbook	
Microsoft Windows Mobile 6.1	
Microsoft Windows Mobile 7	
Microsoft Windows Mobile 7.5	
Microsoft Windows Mobile 8	
Microsoft Surface	
Palm Web OS 4.5	
sSamsung Galaxy S3+	

3RD-PARTY EMAIL CLIENTS

CLIENT

MEDIA QUERY SUPPORT

Microsoft Outlook Exchange 3rd party app (Android)



Gmail mobile app (all platforms)



Yahoo! Mail mobile app (all platforms)



GENERAL SUPPORT

OS	PREVIEW TEXT	MEDIA QUERIES	IMAGES	ALT TEXT	SCALE
Android 4.0 (Gmail)					
Android 4.0 (email)					
BlackBerry OS 6					
iOS 6					
Windows Phone 7.5					

= sort of

2.7 THE IMPORTANCE OF “!IMPORTANT”

Whatever CSS you write within the media tag should end with “!important”. This will ensure that any inline style will be overwritten by the CSS in the style tag. This is the method we use to create two different layout styles for mobile and desktop users.

However, not every email client is that fond of the style tag, take Gmail for example. Gmail will remove any style tag inside of the email regardless of its position (inside the header, body). so in this case you will have to take peace with the fact that Gmail will always show users the desktop version.

2.8 RESET THE BODY

Make sure you zero out the padding and margins on the body, if you don't the document will render with a white border around the content in ios mail. You then disable automatic text size adjustment on some mobile clients with `-webkit-text-size-adjust:none` and `-ms-text-size-adjust:100%`, which stops devices from reducing/increasing text sizes based on their defaults.



```
<body bgcolor="#ffff" topmargin="0" leftmargin="0" marginheight="0" marginwidth="0" style="height: auto; padding:0; margin:0; -webkit-text-size-adjust:none; -ms-text-size-adjust:100%;">
```

2.9 RESIZE FONTS

Smaller screens can cause the text to become unreadable because they are too small to read, the text should transform just like the rest of your design depending on the screen resolution. Making sure that users can read your email without any effort and providing a user-friendly email. Try to keep the proportions right, for instance a text with the H1 tags should be bigger than a normal P tag text.



```
*[class].transf_tekst{ font-size:17px !important; line-height:21px !important; }
```

You could also do this on the entire body of the email if you don't like to give every P or SPAN tag a specific class.

SIZE MATTERS

Links and buttons should have a minimum target area of 44 x 44 pixels, as per Apple guidelines. nothing is more unusable than clouds of tiny links on touchscreen devices. suggested is a font size between 17-21 pixels; the usual font size is about 19 pixels.

2.10 HYPERLINKS

Just as with text, a smaller screen can make it difficult for users to click on a hyperlink. You don't want the users to zoom in to be able to click on a hyperlink or button. The best practice here is to make the buttons and hyperlinks larger. The larger the button, the easier it is to click on it. If possible a full-width button is never a bad idea.

The easiest way to do this is by using pure CSS hyperlinks and buttons. When using images you will have to do an image swap of some sort for every button to have this effect. While a CSS button can be changed fully by changing class on mobile devices.

EXAMPLE



```
a[class=transf_btn]{
  display:block !important;
  font-size:14px !important;
  font-weight:bold !important;
  padding:6px 4px 8px 4px !important;
  line-height:18px !important;
  background:#dddddd !important;
  border-radius:5px !important;
  margin:10px auto;width:70%;
  text-align:center; color:#111 !important;
  text-decoration:none;
  text-shadow:#fff 1px 0 0;
}
```

2.11 LIQUEFY YOUR DESIGN

As mentioned before in this document, there are different methods for creating a responsive layout. The easiest method consists of creating a liquid design which will adapt to the width of the device it is displayed on. This implies using percentages to control your layout.

This method ensures a shorter CSS instead of calculating the amount of pixels for each resolution and converting them accordingly. both are acceptable of course, and a combination of the two can be handy from time to time.

In the example below you can see the CSS used to liquefy a table from a fixed width to a variable width.



```
table[class=transf_100],
img[class=transf_100]{
  width:100% !important;
  height:auto !important;
}
```

OR



```
*[class.transf_100{
  width:100% !important;
  height:auto !important;
}
```

By setting the width to 100% you ensure that the layout will cover the whole width of the device, no matter the resolution. The height will be proportional to the width because it is set to "auto".

You still have to declare the width of the table for desktop clients. You can do this by setting the width attribute of the table. Don't set it as an extra in the style attribute because this will always overwrite the CSS class in the style tag (unless you are using "!important").



```
<table border="0" cellpadding="0" cellspacing="0" width="660" class="transf_100">
  <tr>
    <td align="center"> Custom text goes here </td>
  </tr>
</table>
```

When not using the liquefy method, you are depending on the fixed design method which can be done as followed:



```
.w320 {
  width: 320px !important;
}

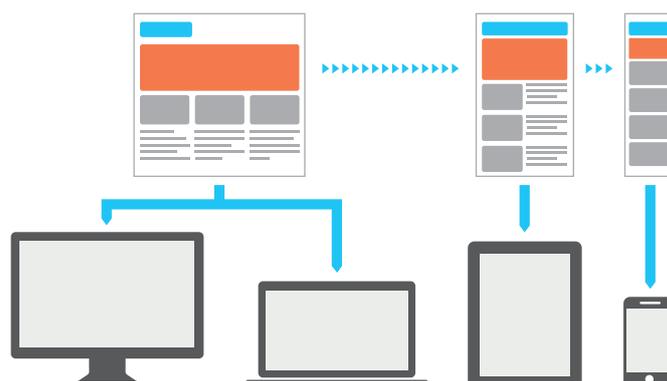
<table border="0" cellpadding="0" cellspacing="0" width="660" class="w320">
  <tr>
    <td align="center"> Custom text goes here </td>
  </tr>
</table>
```

The disadvantage of this method is that the table will now remain 320 pixels on mobile devices. So even if the screen is smaller the table remains the same.

2.12 SLIM DOWN YOUR CONTENT

Most emails nowadays are filled with tons of information on all sorts of different things. However not every piece of information in the email is of the same importance; some are more important than others. When creating your responsive design you should keep this in mind and make a selection of the "most important" information in the email. Mobile emails tend to look crowded when there is too much information.

The main objective of a mobile version should be to ensure it is as easy to scan through and as readable as possible. This often means that some of the secondary content in your newsletter should be hidden to keep the focus on what's most important.



By applying a simple hide class to any image, paragraph or entire table that isn't top priority, it is automatically hidden from the mobile version. Here are some of the elements that can be considered unworthy for the mobile version:

- **Links to the web version and preference center:** traditionally you might put high priority on things like a link to view the email in a browser for a mobile device. But by optimizing the design, the version you're sending is actually the best possible.
- **Additional footer content:** links to Flickr, Twitter, Facebook, etc.
- **Month of edition**
- **Social sharing links:** this is a tough call, but in order to "Like" on Facebook, the subscriber would need to be logged to Facebook on his phone (instead of just using the Facebook app). Same goes for sharing on Twitter or using the "Tell a friend" feature. These are easy on a desktop, but possibly too complicated on a mobile device.
- ...

You can hide unwanted content by adding a simple line in your media query.



```
table[class=nok_mobile],
td[class=nok_mobile],
img[class=nok_mobile],
span[class=nok_mobile],
div[class=nok_mobile]{
  display:none !important;
}
```

Or if you would like it to be available for every kind of tag you could use the following piece of code:



```
*[class].nok_mobile{
  display:none !important;
}
```

In the HTML of the email you can use this piece of CSS code for the parts you want to hide. For instance if you have a table with a custom text that doesn't have to be shown on a mobile device, you can do it like this:



```
<table border="0" cellpadding="0" cellspacing="0" width="660" class="nok_mobile">
  <tr>
    <td align="center"> Custom text goes here </td>
  </tr>
</table>
```

2.12.1 THINK IN PACKAGES

When slimming down your design for mobile users you should stop thinking in pages and start thinking in packages. Consider every part of your email as a package you can select to show or not show. So to give an example, these could be packages:

- Header
- Navigation
- Top Article
- ...

Packaging gives you more freedom in designing your email. However you also have the possibility to hide parts inside a package if needed. Creating a content hierarchy is a good exercise to select the correct content to show in the responsive design version.

LOOSE DESIGN

Normally when creating an email you would put everything in fixed tables and use widths and heights as much as possible to make sure that the design remains the same in the different email clients. But in the case of responsive designs you want the tables to be able to move a bit around when you're displaying the email on a mobile client. You will need to do a combination of fixed width/height but keep the tables floating.

Using this method will make it easier to code your design into HTML. Trying to convert a fixed design into a responsive design is very hard and (in most cases) won't work out as desired. So you will create a loose design. Below you can find an example of a table structure:

FIXED

```
<table width="100%" border="0" cellspacing="0" cellpadding="0">
  <tr>
    <td>
      <td>
        <table cellpadding="0" cellspacing="0" width="600">
          <tr>
            <td width="600"> Webversion </td>
          </tr>
          <tr>
            <td width="600">
              <table cellspacing="0" cellpadding="0" width="600">
                <tr>
                  <td width="600"> content </td>
                </tr>
              </table>
            </td>
          </tr>
        </table>
      </td>
    </tr>
  </table>
```



LOOSE

```
<table width="100%" border="0" cellspacing="0" cellpadding="0">
  <tr>
    <td>
      <table cellpadding="0" cellspacing="0" width="600" class="transf_100">
        <tr>
          <td width="600"> Webversion </td>
        </tr>
      </table>
      <table cellspacing="0" cellpadding="0" width="600">
        <tr>
          <td width="600"> content </td>
        </tr>
      </table>
    </td>
  </tr>
</table>
```



2.13 MULTIPLE COLUMNS ON ONE SCREEN

Most emails today consist of more than 1 column. This can cause problems when converting the layout to a responsive design. It is always better to have one column for the mobile user because every extra column will result in a smaller display.

This does not mean that it is not allowed to have multiple columns in the desktop version. But you need to reduce them to one column for the mobile version. Luckily mobile clients allow us to use the float technique without any issue. By using this technique you can float a table td to the left side of the screen.



Floating the right TD to the left side will create the idea that you are viewing a one-column email while in fact it can consist of multiple columns. But that's not the only thing: you also need to set the width of this TD to 100% so that it can fill the full screen and not keep the pixels from the desktop version.

```
td[class=transf_2column]{
  width:100% !important;
  float:left !important;
}
```



In the HTML code you will also have a table with a width set to 100% and a certain amount of TDs with the "transf_2column" class.



```
<table border="0" cellpadding="0" cellspacing="0" width="660" class="transf_100">
  <tr>
    <td class="transf_2column" width="330" valign="top"></td>
    <td class="transf_2column" width="330" valign="top"></td>
  </tr>
</table>
```

2.13.1 REMOVING A SPACER TD

Sometimes a TD is used between two content TDs to generate a space between them. This is something that is not recommended in the mobile version because it will create too much whitespaces between both content TDs that are now one below the other. You can solve this issue by removing the spacer TD between the content TDs like this:



STYLE

```
td[class=transf_2column]{
  width:100% !important;
  float:left !important;
}
*[class].nok_mobile{
  display:none !important;
}
```



HTML

```
<table border="0" cellpadding="0" cellspacing="0" width="660" class="transf_100">
  <tr>
    <td class="transf_2column" width="310" valign="top"></td>
    <td class="nok_mobiles" width="40" valign="top"></td>
    <td class="transf_2column" width="310" valign="top"></td>
  </tr>
</table>
```

2.14 RESPONSIVE IMAGES

When working with images on mobile devices, you need to keep in mind that most devices will be on a connection such as 3G for instance, and not on a wireless router of some sort. This causes mobile devices to load certain items slower than on a computer with a direct internet connection.

People don't like to wait for their emails to load, which will be the case when using the same image size for the mobile and for the desktop versions. To counter this, you can swap the images to a smaller version when the email is received on the mobile device.

This means you have to do two things: you need to hide the existing image (desktop version) and replace it with a smaller image. There are a couple of different methods to do this, but not all of them work on the different clients. The following method however is a bit more secure and is very quick to implement.

You need to add a rule to the style tag (inside the media query) to define the content of a certain image tag. Which can be done like this:



```
img[id:"ID_OF_IMAGE"] {  
  Content:url("URL_TO_MOBILE_IMAGE") !important;  
}
```

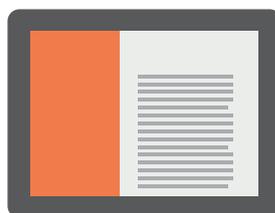
The HTML code would be something like this:



```

```

This technique is not only used to display a smaller sized version of the original image but can also be used to show a completely different image on mobile devices. You could target your mobile device users even more by having target-specific images for these users.



2.15 TEST YOUR RESPONSIVE DESIGN

You can use one of the following tools to test your mobile design:

- mattkersley.com/responsive: this website allows you to enter a URL with your responsive design and it will give you a preview of the design in different sizes.
- resizemybrowser.com: not all developers have all the different devices to test their design on. This website will rescale your browser to the desired resolution.
- pxtoem.com: convert your pixel into percentages/ems. This is always better when using responsive design.
- www.mrss.com/wp-content/uploads/2012/04/12NTC_Mobile_Email.pdf: PDF providing more information about mobile emails.

Sources

www.emailmonday.com/mobile-email-usage-statistics

litmus.com/blog/mobile-opens-hit-record-high-of-47

www.digitalglare.com.au/news_what-is-responsive-email_190.html

www.webdesignerdepot.com/2013/06/responsive-html-email-design/

econsultancy.com/be/blog/63427-responsive-email-design-10-great-examples

www.emailonacid.com/blog/details/C13/emailology_viewport_metatag_rendered_unusable

litmus.com/blog/connections-2012-key-takeaway-design-for-all-inboxes

css-tricks.com/snippets/css/media-queries-for-standard-devices/

www.campaignmonitor.com/guides/mobile/

stephen.io/mediaqueries/

litmus.com/blog/the-how-to-guide-to-responsive-email-design-infographic/full-view

www.returnpath.com/resource/email-in-motion/