Introducción

En este tema  hablaremos sobre buenas prácticas para el comienzo del desarrollo de plantillas emails. Las bases del desarrollo para email como los comentarios de las secciones, la estructura distintos elementos, y veremos un plantilla que nos servirá como modelo para diferentes dispositivos y cliente de email.

Una buena práctica para empezar a desarrollar nuestras plantillas HTML es comentar en el código las secciones en las que se divide un email.

El email se divide en las siguientes secciones:

* Cabecera
* Cuerpo o Hero (imagen o gráfico llamativo)
* Footer

Los comentarios podrían quedar de la siguiente manera:

<!DOCTYPE html>

<html lang="es" dir="ltr">

<head>

<meta charset="utf-8">

<title></title>

</head>

<body>

<!-- Aquí va mi header -->

<!--Aquí va el cuerpo -->

<!-- Footer -->

</body>

</html>

La estructura de las plantillas de los emails html están basadas en tablas. Cada sección mencionada anteriormente es una columna en una tabla HTML.

Una estructura básica sería la siguiente:

<!DOCTYPE html>

<html lang="es" dir="ltr">

<head>

<meta charset="utf-8">

<title></title>

</head>

<body>

<table>

<tr>

<td></td>

</tr>

</table>

</body>

</html>

Uno de los errores más comunes y que se no se debe hacer en el desarrollo de HTML para emails es utilizar la propiedad colspan. Lo que colspan nos permite es poder unir filas dentro de una columna sin necsidad de afectar a la columna anterior.

Por ejemplo lo siguiente sería problemático:

<table>

<tr>

<td></td>

</tr>

<tr>

<td></td>

<td colspan="2"></td>

</tr>

</table>

No se utilizará esta propiedad ya que Outlook no puede interpretarla.

Otro elemento que se utiliza habitualmente es el elemento span. Es un elemento inLine , es decir que va estar alineado con el resto de los elementos. Y se usa normalmente para introducir texto.

Por ejemplo en inline se utiliza este elemento como headline o como un título.

<span>Título</span>

Se utilizan otros elementos con menos frecuencia como los div. Obviamente se utilizan elementos img ya que un email sin imágenes no sería muy llamativo.

<!DOCTYPE html>

<html lang="es" dir="ltr">

<head>

<meta charset="utf-8">

<title></title>

</head>

<body>

<table>

<!-- Aquí va el header -->

<tr>

<td>

<span>Título</span>

</td>

</tr>

<!-- El código sigue en la siguiente diapositiva -->

<!-- Aquí va el cuerpo -->

<tr>

<td>

<span>Body</span>

<div class="">

<img src="" alt="">

</div>

</td>

</tr>

<!-- Footer -->

<tr>

<td><span>Footer</span></td>

</tr>

</table>

</body>

</html>

Vamos examinar detenidamente la plantilla.

Podemos ver que empezamos nuestro documento con el DOCTYPE. Para email es recomendable no usar el DOCTYPE de HTML5.

La razón es porque hay unos clientes de email como yahoo o como outlook que todavía necesitan este tipo de información para mostrar el email.

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML

1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

Uno de los elementos más importantes de cualquier plantilla emails responsive es el meta con la propiedad name con valor viewport. Sin esto el email no va a poder ser interpretado en un dispositivo móvil.

En pocas palabras lo que permite esto es indicarle al navegador que interpreta este código que tome en cuenta el tamaño de la pantalla del dispositivo. Este elemento meta sirve para cualquier tipo de dispositivo ya sea un móvil o una tablet.

<meta name="viewport" content="width=device-width, initial-scale=1">

<meta name="viewport" content="width=device-width, initial-scale=1">

A continuación vamos a hablar de las buenas prácticas con los estilos CSS.

El siguiente bloque importante es el CSS, este es el único CSS que vamos a necesitar en la parte superior del código. La mayoría del CSS lo vamos implementar inline.

La primera línea:

\* { -webkit-text-size-adjust:none; -ms-text-size-adjust:none; max-height:1000000px;}

Se utiliza para que iphone no aumente el tamaño de las fuentes.

Esta línea en particular:

#outlook a {padding:0;}

No deja que Outlook añada algún tipo de espacio alrededor de los enlaces.

Lo mismo para estás otras líneas CSS:

.ReadMsgBody { width: 100%; }

.ExternalClass { width: 100%; }

.ExternalClass \* { line-height: 100%; }

Son para clientes online como Yahoo, Gmail y AOL.

Por último está línea:

.ios\_geo a { color:#1c1c1c !important; text-decoration:none !important; }

Ayuda a prevenir que los teléfonos iOS conviertan partes del texto como teléfonos, fechas, dominios en enlaces.

La siguiente parte del CSS es la media query. Esta parte es muy importante para los emails responsive.

El límite de ancho que se va a dar a la estructura del código es de 600px como ya vimos en el anterior tema. Usando 600 podemos cubrir gran parte dispositivos móviles y tabletas.

/\* responsive styles \*/

@media only screen and (max-width: 600px) {

Respecto a los estilos globales, esta línea en particular:

.hide{ display:none !important; display:none;}

Esconde ciertos elementos en pantallas pequeñas.

.blockwrap{ display:block !important; }

Esta permite montar ciertos elementos.

.showme{ display:block !important; width: auto !important;

overflow: visible !important; float: none !important;

max-height:inherit !important; max-width:inherit !important;

line-height: auto !important; margin-top:0px

!important; visibility:inherit !important;}

Esta permite esconder elementos de pantallas grandes a pantallas pequeñas, en otras palabras podemos esconder elementos en Outlook y dejarlos que se vean otros clientes.

\*[class].movedown{ display: table-footer-group !important;}

\*[class].moveup{ display: table-header-group !important; }

Estas dos líneas van a permitir que podamos mover los elementos arriba o abajo de la tabla.

El siguiente bloque tiene que ver con todo lo relaciona al texto y las fuentes.

\*[class].textright{ text-align: right !important; }

\*[class].textleft{ text-align: left !important; }

\*[class].textcenter{ text-align: center !important; }

\*[class].font27{ font-size: 27px !important; font-weight:normal !important;

line-height:27px !important; }

La última línea es para ajustar el tamaño de la fuente en pantallas de teléfonos móvil.

Por último se visualizan las reglas CSS del ancho y del alto. Vamos a utilizar 10px de alto y 320.

/\* anchos y altos \*/

\*[class].h10{ height:10px !important; }

\*[class].w320{ width:320px !important; }

La línea del body tiene una serie de reglas CSS inline aplicadas. Hacemos esto para prevenir que Gmail o Yahoo traten de modificar algunos espacios dentro del body.

<body style="margin:0; padding:0;" topmargin="0" leftmargin="0" marginheight="0"

marginwidth="0">

La regla comentada se trata de un código condicional y es exclusivamente para Outlook, esta regla previene que Outlook trate de aumentar el tamaño de los spacers.

Los spacers son imágenes que se usan porque miden 1px por 1px y son transparentes. Es buena técnica usar estos spacers para definir ciertas medidas en los emails.

<!--[if gte mso 15]>

<style type="text/css" media="all">

tr { font-size:1px; mso-line-height-alt:0; mso-margin-top-alt:1px; }

</style>

<![endif]-->

Y finalmente tenemos la tabla contenedora, que tiene un ancho de 100%. Usando está tabla podemos añadir un color de fondo con la propiedad bgcolor.

<table width="100%" border="0" cellspacing="0" cellpadding="0" bgcolor="#fff">

De esta forma todo el ancho y el alto del email va a ser de color blanco.

Y la fila contenedora que es donde se implementará todo el contenido del email.

<table width="100%" border="0" cellspacing="0" cellpadding="0" bgcolor="#fff">

<tr>

<td align="center" valign="top">

<!-- Empieza el contenido -->

<!-- Final de contenido -->

</td>

</tr>

</table>