Email HTML - 最佳实践

目录
概述
关键原则
HTML
使用较旧、更简单的 HTML 标准
表格作为基础
避免使用有问题的元素
CSS
CSS 支持限制
始终使用内联样式
谨慎使用 style 标签
支持的和不支持的 CSS
布局技巧
单列设计
使用表格的多列布局
列堆叠技术
响应式设计
移动优先策略
媒体查询
流式混合方法
排版
邮件安全字体
Web 字体
文本格式最佳实践
图像
图像指南
背景图像
电子邮件客户端特定设置
Outlook (Windows)

Gmail

Apple Mail/iOS

可访问性

语义结构

文本替代

颜色和对比度

导航和链接

测试工具和服务

概述
电子邮件的设计和开发需要一种与传统 Web 开发不同的方法。本指南概述了确保您的 HTML 电子邮件在所有主要电子邮件客户端中正确显示的最佳实践。

关键原则

宽度: 保持电子邮件宽度在 600-640px 之间 -
这可确保您的电子邮件在所有设备上正确显示，而不会在较小的屏幕上出现水平滚动条。
文件大小 : 保持总电子邮件大小低于 100KB - 较大的电子邮件可能会被 Gmail
和其他提供商裁剪，并且在移动设备上加载时间更长。
表格: 使用表格进行布局（尽管在 Web 上已被弃用） - 电子邮件客户端对 CSS
的支持不一致，但对表格的渲染非常可靠。
测试: 在多个电子邮件客户端和设备上进行测试 - 电子邮件客户端以不同的方式渲染 HTML
，测试有助于识别和修复渲染问题。

HTML

使用较旧、更简单的 HTML 标准

使用 HTML 4.01 或 XHTML 1.0 - 许多电子邮件客户端使用过时的渲染引擎，不支持现代 HTML5
功能。
避免在主要结构中使用 HTML5 元素 - 诸如 <section>、 <article> 和 <aside>
之类的元素在所有电子邮件客户端中都未得到支持。

- 始终使用小写字母表示标签和属性 -
  这可确保最大的兼容性，并防止在严格的客户端中出现渲染问题。
- 始终为属性值使用引号 - 未加引号的属性可能导致某些电子邮件客户端出现解析错误。
- 关闭所有标签，即使是自闭合标签也应使用尾随斜杠 (`<br />`) - 这可防止在期望 XHTML
  语法风格的客户端中出现渲染问题。

表格作为基础

- 使用嵌套表格进行布局，而不是基于 div 的布局 - 表格在具有较差 CSS
  支持的电子邮件客户端中提供一致的结构。
- 设置明确的单元格内边距、间距和尺寸 -
  这可防止在不同的电子邮件客户端中出现不一致的间距和布局。
- 在表格元素和单元格上声明宽度 - 双重声明宽度可以提高渲染的一致性，尤其是在 Outlook 中。
- 使用 `align` 和 `valign` 属性，而不是 CSS 等效项 - 这些 HTML 属性在电子邮件客户端中的支持比
  CSS 定位更好。

```
<table border="0" cellpadding="0" cellspacing="0" width="100%">
    <tr">
        <td align="center" valign="top">
            <!-- 内容在此处 -->
        </td>
    </tr>
</table>
```

避免使用有问题的元素

- 不使用 JavaScript - 大多数电子邮件客户端出于安全原因会移除 JavaScript
  ，因此任何依赖于它的功能都将失效。
- 不使用表单（虽然某些客户端支持） - 表单的支持非常不一致；许多客户端会禁用或移除表单元素。
- 不使用 iframe - 这些通常会被大多数电子邮件客户端出于安全原因移除。
- 不使用背景图像（如果必要，可以使用替代方案） - 许多电子邮件客户端默认禁用背景图像。
- 不嵌入音频 /视频（而是链接到托管内容） - 直接嵌入的支持较差；链接到外部内容更可靠。

CSS

CSS 支持限制

- 对于所有关键部分，请使用内联 CSS - 许多电子邮件客户端会移除 `<style>` 标签或完全忽略它们。

- 避免使用 CSS 简写属性（使用 `margin-top` 代替 `margin`） - 某些电子邮件客户端只能识别单个属性，而无法识别简写。
- 避免使用 CSS 定位属性 (`position`, `float`, `clear`) - 这些支持较差，可能会导致布局问题。
- 避免使用高级选择器（坚持使用元素、类和 ID 选择器） - 复杂的选择器通常在具有有限 CSS 支持的电子邮件客户端中无法正常工作。

始终使用内联样式

```
<p style="font-family: Arial, sans-serif; font-size: 16px; line-height: 24px; color:
#333333;">
    您的内容在此处
</p>
```

谨慎使用 style 标签

- 仅用于特定于电子邮件客户端的技巧
- 谨慎使用媒体查询（支持程度各不相同）
- 始终包含 `type` 属性：`<style type="text/css">`

支持的和不支持的 CSS

支持良好：

- `font-family`, `font-size`, `color` - 基本文本样式在电子邮件客户端中的支持较为一致。
- `text-align`, `line-height` - 文本对齐和间距属性在大多数客户端中工作良好。
- `width`, `height`, `padding`, `margin` (谨慎使用) - 基本的盒子模型属性在使用时效果良好。 Outlook 中， `margin` 的效果可能不佳。
- `border`, `background-color` - 这些视觉属性在大多数电子邮件客户端中都具有良好的支持。

支持较差或不一致：

- Flexbox, Grid - 现代布局系统在许多电子邮件客户端中不受支持，尤其是 Outlook。
- 定位属性 (`position`, `display`) - 这些可能会导致电子邮件在某些客户端中显示不同或破坏布局。
- CSS 动画 - 这些通常会被大多数电子邮件客户端移除或忽略。
- 许多伪类和伪元素 - 支持不一致； Outlook 和某些 Webmail 客户端会完全忽略它们。

布局技巧

单列设计

- 在所有客户端中都最可靠 - 简单的布局具有较少的故障点。
- 建议宽度: 600px - 这可以在大多数预览窗口和移动屏幕中显示。
- 在移动设备上效果良好，无需响应式设计 - 单列布局自然适应窄屏，几乎没有问题。

使用表格的多列布局

```
<table border="0" cellpadding="0" cellspacing="0" width="600">
    <tr>
        <td width="300" valign="top">
            <!-- Column 1 content -->
        </td>
        <td width="300" valign="top">
            <!-- Column 2 content -->
        </td>
    </tr>
</table
```

列堆叠技术

- 在支持的设备上使用媒体查询 - 媒体查询允许在支持它们的客户端上，使列在移动设备上堆叠显示。
- 使用 MSO条件评论针对 Outlook - 这些专门针对 Outlook， Outlook需要特殊的响应式布局处理。
- 考虑对于不支持媒体查询的客户端采用混合 /自适应方法 - 这种技术创建的布局即使在没有媒体查询支持的情况下也能合理地适应。

```
<!--[if mso]>
<table border="0" cellpadding="0" cellspacing="0" width="600">
<tr><td width="300" valign="top"><![endif]-->
<div style="display: inline-block; width: 300px; vertical-align: top;">
    <!-- Column 1 content -->
</div>
<!--[if mso]></td><td width="300" valign="top"><![endif]-->
<div style="display: inline-block; width: 300px; vertical-align: top;">
    <!-- Column 2 content -->
</div>
<!--[if mso]></td></tr></table><![endif]-->
```

响应式设计

移动优先策略

- 首先为小屏幕设计 - 移动设备的邮件打开率通常高于桌面设备，因此移动优化至关重要。
- 尽可能使用比例宽度 - 这允许内容根据不同的屏幕尺寸自动适应。
- 使用合适的字体大小（最小 14px） - 较小的字体在移动设备上难以阅读，可能需要缩放。
- 确保触摸目标至少为 44x44像素 - 这符合可访问性指南，并使触摸屏上的交互更容易。

媒体查询

支持有限但重要:

```
<style type="text/css">
    @media screen and (max-width: 480px) {
        .mobile-full-width {
            width: 100% !important;
            height: auto !important;
        }

        .mobile-font {
            font-size: 16px !important;
        }
    }
</style>
```

流式混合方法

使用 max-width和 MSO条件，可在支持和不支持媒体查询的客户端上工作:

```
<!--[if mso]>
<table width="600" cellpadding="0" cellspacing="0" border="0"><tr><td>
<![endif]-->
<div style="width:100%; max-width:600px; margin:0 auto;">
    <!-- Content -->
</div>
```

```
<!--[if mso]>
</td></tr></table>
<![endif]-->
```

排版

邮件安全字体

始终使用包含广泛支持的字体的字体堆栈：

```
font-family: Arial, Helvetica, sans-serif;
font-family: 'Helvetica Neue', Helvetica, Arial, sans-serif;
font-family: 'Times New Roman', Times, serif;
font-family: Verdana, Geneva, sans-serif;
font-family: Georgia, Times, 'Times New Roman', serif;
font-family: 'Courier New', Courier, monospace;
```

Web 字体

- 支持有限 - 仅在使用适当的备选字体时使用 - 许多电子邮件客户端不支持 Web 字体，因此备选字体至关重要。
- 最可靠的支持: Apple Mail、iOS Mail 和 Android 上的 Google 字体 - 这些客户端能够一致地渲染 Web 字体。
- Outlook、Gmail 和 Yahoo! 通常不支持 Web 字体 - 这些流行的客户端将显示您的备选字体。

```
<style>
    @import url('https://fonts.googleapis.com/css2?family=Open+Sans&display=swap');
</style>

<p style="font-family: 'Open Sans', Arial, sans-serif;">您的文本</p>
```

文本格式最佳实践

- 将基本字体大小设置为 14-16px - 较小的字体可能难以阅读，尤其是在移动设备上。
- 行高应为字体大小的 1.4-1.6 倍 - 适当的间距可以提高可读性，并防止文本显得拥挤。
- 将大段文本分成小块 - 将内容分解成小块可以提高可读性和用户对电子邮件的参与度。

将文本左对齐以获得更好的可读性 - 这可以创建一致的阅读体验，并防止两侧出现不规则的边缘。
确保文本和背景之间具有足够的对比度 - 这可以提高所有用户的可读性和可访问性。

图像

图像指南

始终包含 alt 属性 - 这在图像被阻止时提供文本替代方案，并提高可访问性。
为所有图像设置明确的宽度和高度 -
这可以防止图像加载时出现布局偏移，并在图像被阻止时保持结构。
保持图像文件大小小 (针对 Web 优化) - 较大的图像会增加加载时间，并可能超出文件大小限制。
考虑当图像被阻止时会发生什么 -
许多电子邮件客户端默认会阻止图像，因此您的电子邮件即使在没有图像的情况下也应该仍然易于理
解。

<img src="https://example.com/image.jpg" alt="图像描述" width="600" height="400" style="display: block; width: 100%; max-width: 600px; height: auto;" border="0">

背景图像

支持有限；始终提供备用背景颜色：

<table background="https://example.com/bg.jpg" bgcolor="#f7f7f7" width="600" cellpadding="0" cellspacing="0" border="0">
<tr">
<td">
内容
</td>
</tr>
</table>

电子邮件客户端特定设置

Outlook (Windows)

- 使用 Word 渲染引擎 (MSO) - 这解释了 Outlook 的独特渲染特性以及有限的 CSS 支持。
- 需要 VML 来实现背景图像和圆角 - 标准的 CSS 方法对于这些效果在 Outlook 中不起作用。
- 使用 MSO 条件性评论来包含特定于 Outlook 的代码 - 这允许您专门针对 Outlook 版本。

```
<!--[if mso]>
    Outlook 相关的代码
<![endif]-->
```

Gmail

- 移除样式标签和头部部分 - 所有关键的样式必须使用内联样式。
- 在某些情况下，移除 class 和 ID 属性 - 不要依赖这些属性来实现关键的样式或功能。
- 始终使用内联样式 - 这是在 Gmail 中样式化内容的唯一可靠方法。
- 限制邮件大小（如果太大则会裁剪） - 超过 102KB 的邮件会被裁剪，并显示一个 “查看完整消息 ” 的链接。

Apple Mail/iOS

- 最佳的渲染能力 - 这些客户端支持现代 CSS 和 HTML 标准。
- 支持大部分现代 CSS - 如果主要面向 Apple 用户，可以使用 flexbox 和 grid 等特性。
- 良好的媒体查询支持 - 响应式设计在这些平台上可以可靠地工作。

可访问性

语义结构

- 尽可能使用语义化的 HTML (`p` , `h1` , `h2` 等) - 这可以提高屏幕阅读器的解析能力和整体可访问性。
- 为布局表格添加 `role="presentation"` - 这告诉屏幕阅读器该表格仅用于布局，而不是用于数据展示。
- 包含正确的标题结构 - 这会创建一个逻辑的文档大纲，帮助屏幕阅读器用户浏览。
- 为装饰性元素使用 `aria-hidden="true"` - 这可以防止屏幕阅读器朗读纯粹的视觉元素。

```
<table role="presentation" border="0" cellpadding="0" cellspacing="0">
```

文本替代

- 始终为图像使用描述性的替代文本 - 这可以确保使用屏幕阅读器的用户能够理解图像的内容和目的。

为装饰性图像设置空的替代文本 - 这可以防止屏幕阅读器朗读那些不提供信息的图像。
考虑在禁用图像时邮件的外观 -
许多用户和邮件客户端会阻止图像，因此内容应该在没有图像的情况下也能正常工作。

颜色和对比度

保持正常文本的最小对比度比为 4.5:1 - 这符合 WCAG
可访问性标准，并可以提高所有用户的可读性。
使用实际文本，而不是图像中的文本 -
图像中的文本无法被屏幕阅读器访问，并且在不同的设备上缩放效果不佳。
不要仅仅依靠颜色来传达信息 - 这可以适应有色觉障碍的用户。

导航和链接

使链接易于识别 - 独特的样式可以帮助用户识别可点击的元素。
使用描述性的链接文本（避免 "点击这里 "）-
描述性的链接可以提供上下文，并对使用屏幕阅读器的用户更友好。
确保触摸目标之间有足够的间距 - 这可以防止在移动设备上意外点击，并帮助有运动障碍的用户。

测试工具和服务

使用专业的邮件测试服务，例如 Litmus和 Email on Acid
。这些平台提供全面的跨多个邮件客户端和设备的测试。
尽可能在真实的邮件客户端中进行测试。虽然测试服务很有价值，但实际测试可以发现自动化测试可
能无法检测到的问题。
检查在桌面和移动客户端中的渲染效果。移动邮件的打开率通常高于桌面，因此移动测试至关重要。
在图像被禁用的情况下进行测试。许多邮件客户端默认会阻止图像，因此请确保您的邮件在没有图像
的情况下也能被阅读和正常使用。

检查垃圾邮件过滤器得分。使用诸如 Mail-Tester 之类的工具来识别邮件内容中潜在的垃圾邮件触发因素。
验证 HTML。使用专门用于邮件的验证器，以在发送之前发现潜在的渲染问题。

版本号 #2
由 壳壳虫 创建于 2026-03-09 11:53:58 UTC
由 壳壳虫 更新于 2026-03-09 11:53:58 UTC