HTML NEWSLETTER AND MJML

WHY IS CREATING HTML NEWSLETTERS SO WEIRD?

- Many render engines with different capabilities
- Rarely get updated for new requirements
- Webmailer are picky and remove a lot of styles

BASIC LAYOUT TECHNIQUES

- Use deeply nested tables for all layouts
- Use inline CSS for general styling and style-tags for responsive overrides
- Don't use body tag for styling
- Avoid html-attributes like cellpadding or cellspacing
- Avoid padding and width/height on the same element
- Avoid floats or modern layout methods
- Use conditional commands to target Outlook
- Font-tags, center-tags are not necessary anymore

EXAMPLE

`template`-ticketyboo.html

COMMON EMAIL CLIENTS

OUTLOOK

- Since Outlook 2007 Word is used as render engine
- Engine hasn't been updated since then
- Support for margin, padding, width and height spotty
- No media queries or other modern CSS-features
- No support for background images
- Outlook for Mac uses Webkit

OUTLOOK.COM

- Uses render engine of browser
- Didn't support CSS margins for a while
- Background images don't repeat
- Media queries support was supposed to be introduced in 2019
- Can be targeted in CSS via `[owa] .test {}`

GMAIL

- Support of style-tags in the head since late 2016
- No support for style-element in the body
- Messages larger than 1024kb are clipped
- Doesn't support attribute selectors
- Ignores style-elements larger than 16,384 characters (used to be 8,192)
- Removes entire style-tag when encountering an error
- Numbers and URLs are automatically converted to links

APPLE MAIL

- Very good support due to Webkit rendering engine
- Dark mode is not applied to HTML emails automatically
- Media queries can be used to add support for dark mode `@media (prefers-color-scheme: dark)`
- Numbers and URLs are automatically converted to links

STATISTICS

- 4 of 5 used email clients are either from Apple or Google (Apple Mail and Gmail)
- Standalone Outlook much more popular than Outlook.com
- https://www.kevinmandeville.com/blog/email-client-market-share-trends-2018

TOOLS

CSS-INLINER

- Convert style-tags to inline styles
- No support for media queries
- Usually applied before deployment
- Circumvent issues of lack of support of embedded or external styles
- Popular tools: Premailer, PutsMail, MailChimp Inliner

SENDING TOOLS

- Can be used to send the unmodified html source code to email addresses
- Usually require email verification to avoid abuse
- Some services provide APIs for automated testing
- Popular tools: MailGun, SendGrid

FRAMEWORKS & BOILERPLATE TEMPLATES

- MJML
- Foundation for Emails 2 (formerly Zurb Ink)
- Cerberus
- Blueprints by Mailchimp

TESTING

- Litmus Email ($99 per month)
- Email on Acid ($86 per month)
- Testi@ ($0/$20 per month)

MJML

BASICS

- Open source framework for responsive HTML mails
- First release in January 2016
- Uses `Mailjet Markup Language` DSL
- Comes with library of components out of the box
- Supports creation of custom components
- Used by The New York Times and Ryanair (as of 2018)

TECHNICAL DETAILS

- Version 4 was rewritten in vanillaJS
- Plugins for common editors like VSCode
- Supported by module bundlers like Webpack and task runners like Gulp
- Provides quality of live improvements like watch tasks and linter
- Can be used with React and template engines
- Can be used standalone via npx

ADVANTAGES

- Fixes a lot of the common issues with render engines
- Automatically inlines styles during compilation and adds Outlook specific overrides
- Gives errors about incorrect usage at compile time
- Allows splitting up into partials/components
- MJML DSL is fairly close to HTML
- Mobile first

DISADVANTAGES

- No support for variables, loops, template inheritance etc.
- Requires use of template engines to enable parameterization
- Imposes restrictions on email design
- No way to configure render engine support
- Outdated guides and tips that reference MJMl version 3 can be confusing

BUILDING BLOCKS

BUILDING BLOCKS - BODY

```
<mj-body>
   ...
</mj-body>
```

- Acts similar to body-tag in HTML
- Sets background color and max-width of email
- Used to be called container in MJML 3

BUILDING BLOCKS - SECTION

```
<mj-section>
   ...
</mj-section>
```

- Similar to rows in grid systems like Bootstrap or Foundation
- Cannot contain sections themselves
- Multiple sibling sections are okay

BUILDING BLOCKS - COLUMN

```
<mj-column>
  ..
</mj-column>
```

- Similar to columns in grid systems like Bootstrap or Foundation
- Must be a direct children of sections
- Cannot contain other columns or sections
- Fill up row automatically, e.g. 1 column 100% width, 2 columns 50% width etc.
- Columns become stacked on smaller viewports

CREATING A BASIC GRID LAYOUT

```
<mj-section>
  <mj-column>
    Full-width column
  </mj-column>
</mj-section>
<mj-section>
  <mj-column>
    Column 50%
  </mj-column>
  <mj-column>
    Column 50%
  </mj-column>
</mj-section>
```

USING IMAGES

```
<mj-image
  width="250px"
  src="image.png"
  padding="0"
  align="center"
  href="#"
  fluid-on-mobile
/>
```

- Helper to deal with images
- Images can be used as links
- SrcSet is supported
- Can be full width on mobile and fixed width on desktop

STYLING ELEMENTS - INLINE STYLES

```
<mj-text color="red" font-size="16px">
  Cat
</mj-text>
```

- Not reusable
- Inline style override global styles
- Make changes more difficult

STYLING ELEMENTS - GLOBAL STYLES

```
<mj-attributes>
  <mj-class name="font-size-base" font-size="16px" />
  <mj-class name="color-primary" color="red" />
</mj-attributes>

<mj-text mj-class="font-size-base color-primary">
  Cat
</mj-text>
```

- Reusable
- Easily changeable
- Get inlined during compilation
- Can be overridden by inline styles

OTHER ELEMENTS

- Buttons
- Hero component
- Media elements
- Navbar
- Accordion
- Social icons
- Data tables

WORKING WITH PARTIALS

```
// base
<mj-body background-color="#4e4096">
  <mj-include path="./child-component.mjml" />
</mj-body>



// child-component.mjml
<mjml>
  <mj-body>
    <mj-text>
      Hello
    </mj-text>
  </mj-body>
</mjml>
```

- Watch task also works on partials since v4
- No way to pass variables to a partial

PARAMETERIZE MJML TEMPLATES

- Not possible out of the box
- Can be achieved via template engines like handlebars, twig etc.
- Generate MJML markup via template engine and then compile to HTML
- Parameters are handled by template engine when MJML markup is generated