

# Våra tekniska riktlinjer för dynamiskt innehåll

Här nedan har vi samlat våra tekniska guidelines för att skicka så säkert och tillgängligt innehåll som möjligt.

## General Style Rules

### Security

To protect the integrity of the receiver; external assets and tracking using pixels or script are prohibited.

### Context

The HTML Document will be consumed on a variety of clients and factors like screen size will vary. It is therefore important that the layout of the document makes the most whatever frame it is given. This should be a responsive document, not a “website”.

## General Meta Rules

### Encoding

Use UTF-8 (no BOM).  
Make sure your editor uses UTF-8 as character encoding, without a byte order mark.  
Specify the encoding in HTML documents via `<meta charset="utf-8">`.  
(More on encodings and when and how to specify them can be found at [Handling character encodings in HTML and CSS.](#))

## HTML Style Rules

### Document type

Use HTML5.  
HTML 5 (HTML syntax) is preferred for all HTML documents:  
`<!DOCTYPE html>`.

### HTML Validity

Use valid HTML.  
Use tools such as the [W3C HTML validator](#) to test.

<pre>&lt;!-- Recommended --&gt; &lt;meta charset="utf-8"&gt; &lt;title&gt;Test&lt;/title&gt; &lt;article&gt;This is only a test.&lt;/article&gt;</pre>	<pre>&lt;!-- Not allowed --&gt; &lt;title&gt;Test&lt;/title&gt; &lt;article&gt;This is only a test.</pre>
--	---

### Hyperlink Security & Reliability

HTML Content sent in Kivra is archived and can't be updated or “fixed” at a later point including links. Over time this increases the risk of “dead links”.  
it's recommended that the message of the HTML Content doesn't depend on a link working.

Using `<a>` tags without `rel="noreferrer noopener"` is prohibited for security and integrity.

<pre>&lt;!-- Recommended --&gt; &lt;a   href="https://example.com/support"   target="_blank"   rel="noreferrer noopener"&gt;   Example &lt;/a&gt;</pre>	<pre>&lt;!-- Not allowed --&gt; &lt;a href="https://example.com/support"&gt;   Example &lt;/a&gt; &lt;a   href="https://example.com/support"   target="_blank"&gt;   Example &lt;/a&gt;</pre>
---	---

### Multimedia Fallback

Provide alternative contents for multimedia.

For multimedia, such as images, video, animated objects via [canvas](#), make sure to offer alternative access. For images that means use of meaningful alternative text ([alt](#)) and for video and audio transcripts and captions if available.

Providing alternative contents is important for accessibility reasons: A blind user has few cues to tell what an image is about without [@alt](#), and other users may have no way of understanding what video or audio contents are about eiter.

(For images whose [alt](#) attributes would introduce redundancy, and for images whose purpose is purely decorative which you cannot immediately use CSS for, use no alternative text, as in `alt=""`.)

<pre>&lt;!-- Recommended --&gt; &lt;img   src=""   alt="Spreadsheet screenshot"&gt;</pre>	<pre>&lt;!-- Not allowed --&gt; &lt;img   src=""&gt;</pre>
---	--

## CSS Style Rules

### CSS Validity

Use valid CSS where possible.  
Unless dealing with CSS validator bugs or requiring proprietary syntax, use valid CSS code.  
Use tools such as the [W3C CSS validator](#) to test.

### Define styles in head

Define CSS using the `<style>` tag inside the head of the document.  
External assets using the `<link>` tag are prohibited for security and reliability.

<pre>&lt;!-- Recommended --&gt; &lt;!doctype html&gt; &lt;head&gt;   &lt;style&gt;     body {       background-color: #fff;     }   &lt;/style&gt; &lt;/head&gt;</pre>	<pre>&lt;!-- Not allowed --&gt; &lt;!doctype html&gt; &lt;head&gt;   &lt;link     href="https://example.com/style.css"     rel="stylesheet"&gt; &lt;/head&gt;</pre>
--	---

### Define inline styles

HTML Content in Kivra allows inline styles using the `style=""` attribute.

### Using assets in CSS

External assets using [url\(https://\)](#) are prohibited for security and reliability, instead include assets using the ["data" URL scheme](#) which consists of `data:[<mediatype>];base64,data`.

Tools like <https://b64.io/> can be used to make conversion of assets easier.

<pre>&lt;!-- Recommended --&gt; &lt;style&gt;   .header {     background-image:       url();   } &lt;/style&gt;</pre>	<pre>&lt;!-- Not allowed --&gt; &lt;style&gt;   .header {     background-image:       url(https://example.com/image.png);   } &lt;/style&gt;</pre>
---	--

## AMP

### Resources

Examples:  
<https://amp.dev/documentation/examples/?format=email>

### AMP for Content in Kivra

SCRIPTS  
To avoid unnecessary traffic from the user to a third party, Kivra provides a mirror of Google Fonts hosted at <https://amp-cdn.kivra.com>.

<pre>&lt;!-- Recommended --&gt; &lt;script src="https://amp-cdn.kivra.com/v0.js"   async crossorigin&gt; &lt;/script&gt; &lt;script src="https://amp-cdn.kivra.com/v0/amp-accordion-0.1.js"   custom-element="amp-accordion"   async crossorigin&gt; &lt;/script&gt;</pre>	<pre>&lt;!-- Not allowed --&gt; &lt;script src="https://cdn.ampproject.org/v0.js"   async&gt; &lt;/script&gt; &lt;script src="https://cdn.ampproject.org/v0/amp-accordion-0.1.js"   custom-element="amp-accordion"   async&gt; &lt;/script&gt;</pre>
--	--