# Dev Portal — CSP Header Size Reduction: Replace Inline CSS with Bootstrap Utilities

Date: 2025-08-19

## 1. Context & Problem

Inline CSS in Drupal content (nodes, blocks, Layout Builder text) forces us to add a unique sha256 hash for each style snippet to the Content-Security-Policy (CSP) header under style-src. As pages accumulate varied inline styles, the header grows, risking proxy/server header-size limits and degrading cacheability. We will eliminate inline CSS in authorable content and standardize on Bootstrap 5 utility classes.

## 2. Target State

• Authorable content contains no style="…" attributes.  
• Writers apply presentation via Bootstrap utility classes only.  
• CKEditor 5 UI exposes a curated Styles dropdown; pasting HTML with inline styles is sanitized.  
• CSP style-src contains only site CSS origins (no dynamic hashes for content).

## 3. Drupal Implementation Plan

### 3.1 Enable Bootstrap utilities in both editor and frontend

Ensure the theme loads Bootstrap 5 utilities in the WYSIWYG iframe as well as frontend so authors see accurate formatting.

### 3.2 CKEditor 5 Configuration

• Remove any buttons that introduce inline styles (e.g., text color picker, background color, font size if they emit style attributes).  
• Enable a Styles dropdown with class mappings (CKEditor 5 Style plugin).

Example classes to expose:

Typography: fw-bold, fst-italic, text-start|center|end, text-uppercase  
Spacing: m-0..5, mt-\*, mb-\*, p-0..5, pt-\*, pb-\*  
Display: d-block, d-inline, d-inline-block, d-none  
Sizing: w-25|50|75|100, h-auto  
Colors (theme-limited): text-body, text-muted, bg-body, bg-light

### 3.3 Text Format (Filtered HTML) settings

Use the HTML Restrictor filter to disallow the style attribute and allow class selectively. Also restrict allowed tags to a safe subset (p, h1–h6, a, ul/ol/li, img, figure/figcaption, table, etc.).

Suggested attribute policy:

• Disallow: style  
• Allow class on: p, span, div, h1–h6, img, figure, table, thead, tbody, tr, td, th, ul, ol, li, a  
• Allow href/target/rel on a; src/alt on img

### 3.4 Example config snippets

Text format YAML (illustrative):

filter\_html\_restrict\_settings:  
 allowed\_tags: 'p h2 h3 h4 h5 h6 a ul ol li figure figcaption img table thead tbody tr td th div span code pre blockquote'  
 allowed\_attributes:  
 a: ['href', 'target', 'rel', 'class']  
 img: ['src', 'alt', 'class']  
 p: ['class']  
 div: ['class']  
 span: ['class']  
 table: ['class']  
 td: ['class']  
 th: ['class']  
 disallowed\_attributes: ['style']

CKEditor 5 Styles presets (illustrative JSON):

editor:  
 styles:  
 definitions:  
 - name: 'Lead paragraph'  
 element: 'p'  
 classes: ['lead']  
 - name: 'Muted text'  
 element: 'span'  
 classes: ['text-muted']  
 - name: 'Center text'  
 element: 'p'  
 classes: ['text-center']  
 - name: 'Tight bottom margin'  
 element: 'p'  
 classes: ['mb-1']

### 3.5 Content migration (remove inline styles)

Run a one-time script to scan nodes for style="…" and replace with equivalent Bootstrap classes. Start with highest-traffic content. Maintain a mapping to minimize manual edits.

Inline CSS → Bootstrap mapping (starter set):

• margin-bottom: 0; → mb-0

• margin-top: 0; → mt-0

• margin: 0; → m-0

• padding: 0; → p-0

• text-align: center; → text-center

• text-align: right; → text-end

• font-weight: bold; → fw-bold

• font-style: italic; → fst-italic

• display: none; → d-none

• display: inline-block; → d-inline-block

• width: 100%; → w-100

Keep a fallback utility CSS file for any one-off cases that don't map neatly, then refactor those within two sprints.

### 3.6 QA & Validation

• Automated: a crawler that fails on presence of style="…" in rendered body.  
• Header size check: capture CSP header bytes across top 50 URLs and compare before/after.  
• Accessibility: verify color-contrast and semantics preserved.

## 4. CSP Policy Update

Once inline styles are removed from content, drop per-page style hashes. Suggested directive:  
style-src 'self' https://fonts.googleapis.com;

If the theme injects critical inline styles, prefer moving them into a stylesheet. If unavoidable, use a single nonce for server-rendered inline style blocks (kept minimal).

## 5. Rollout Plan

Phase 1: Editor + filters in lower env; docs to Tech Writers.  
Phase 2: Content migration for top sections; enable CSP without style hashes on those paths.  
Phase 3: Site-wide enablement; monitoring and quick fix window.

## 6. Success Metrics

• 0 inline style attributes in authorable content.  
• Avg CSP header size < 4 KB across top pages.  
• No CSP style-src hash entries required for content pages.  
• No regressions in page layout.

## 7. Appendices

A. Drush ideas:  
• Audit: `drush sqlq "SELECT nid FROM node\_\_body WHERE body\_value LIKE '%style=%' LIMIT 50;"`  
• Config export/import after text format changes: `drush cex -y`, `drush cim -y`

B. Authoring quick reference (examples):  
• Center text: add class `text-center` to the paragraph.  
• Tighten spacing: use `mb-1`, `mt-0`, etc.  
• Hide element: `d-none` (use sparingly).