

SASS Mixin für typografische Kontrolle in Bricks Builder

Ich nutze den BricksBuilder und dort kann ich dank SASS (durch das Plugin Advanced Themer) Mixins verwenden. Also dort gibt es ja folgende Syntax:

```
%root% {  
  color: red;  
}
```

Kann du mir ein Mixin schreiben, das ich einfach in beliebige Klassen einfügen kann und das folgendes macht?:

Es soll 1. einen Parameter entgegen nehmen (1, 2 oder 3):

1. Wörter brechen typografisch korrekt um, so wie es in der gerade aktiven Sprache (auf der Website) korrekt ist. D.h. also "Mund-schenk" beispielsweise (im Deutschen).
2. Es finden keine Umbrüche mitten in Wörtern statt (nur ganze Wörter), außer die gesamte Zeile reicht für ein einziges Wort nicht aus, dann soll es wie bei 1. umbrochen werden.
3. Umbrüche finden gar nicht mehr automatisch statt. Nur manuelle Umbrüche (z.B. `
`) funktionieren.

Dann soll das Mixing auch noch unabhängig vom gewählten Parameter-Wert festlegen, dass spracheabhängige typografische Zeichen gesetzt werden: Beispielsweise im Deutschen statt `"` dann `„“`, statt `'` dann `‚‘`, statt `--` dann `–`, usw.

Diese Regeln sollen aber intelligent gesetzt werden, d.h. z.B. nicht in `pre`-Elementen usw.

Hier ist ein umfassendes Mixin für deine Anforderungen:

```
// =====  
// TYPOGRAFIE MIXIN  
// =====  
// Parameter:  
// 1 = Silbentrennung (spracheabhängig korrekt)  
// 2 = Nur ganze Wörter (Fallback auf Silbentrennung)  
// 3 = Keine automatischen Umbrüche
```

```

// =====

@mixin typo-wrap($mode: 2) {
  // =====
  // GEMEINSAME EINSTELLUNGEN (immer aktiv)
  // =====

  // Sprache für Anführungszeichen und Trennung erkennen
  // Browser nutzt das lang-Attribut des HTML-Elements

  // Typografische Anführungszeichen aktivieren
  // Deutsch: „" und ‚‘
  // Englisch: "" und ''
  // Französisch: «» und ‹›
  // etc. - Browser wählt automatisch basierend auf lang=""
  quotes: auto;

  // Typografische Optimierungen
  font-variant-numeric: oldstyle-nums; // Optional: Mediävalziffern
  font-kerning: auto;
  font-feature-settings: "kern" 1, "liga" 1, "calt" 1;

  // Witwen und Waisen vermeiden (moderne Browser)
  text-wrap: pretty;

  // =====
  // UMBRUCH-MODI
  // =====

  @if $mode == 1 {
    // MODUS 1: Silbentrennung (sprachabhängig)
    // Wörter werden an korrekten Silbengrenzen getrennt
    hyphens: auto;
    -webkit-hyphens: auto;
    -ms-hyphens: auto;
    hyphenate-limit-chars: 6 3 2; // Min. Wortlänge, vor Trennung, nach Trennung
    hyphenate-limit-lines: 2; // Max. aufeinanderfolgende getrennte Zeilen
    hyphenate-limit-zone: 8%; // Zone am Zeilenende für Trennung
    word-wrap: break-word;
  }
}

```

```
overflow-wrap: break-word;
word-break: normal;

} @else if $mode == 2 {
  // MODUS 2: Ganze Wörter bevorzugen
  // Umbruch nur zwischen Wörtern, Silbentrennung nur als Fallback
  hyphens: auto;
  -webkit-hyphens: auto;
  -ms-hyphens: auto;
  overflow-wrap: break-word;
  word-wrap: break-word;
  word-break: normal;
  // Strengere Trennregeln
  hyphenate-limit-chars: 12 6 4; // Nur lange Wörter trennen
  hyphenate-limit-lines: 1;
  hyphenate-limit-zone: 20%; // Größere Zone = weniger Trennungen

} @else if $mode == 3 {
  // MODUS 3: Keine automatischen Umbrüche
  // Nur manuelle Umbrüche (<br>, \A) funktionieren
  hyphens: none;
  -webkit-hyphens: none;
  -ms-hyphens: none;
  white-space: normal;
  word-break: keep-all;
  overflow-wrap: normal;
  word-wrap: normal;
}

// =====
// AUSNAHMEN für Code-Elemente
// =====

pre,
code,
kbd,
samp,
var,
tt,
```

```
[class*="code"],
[class*="pre"],
.wp-block-code,
.wp-block-preformatted {
    // Keine typografischen Ersetzungen in Code
    quotes: none;
    hyphens: none;
    -webkit-hyphens: none;
    font-feature-settings: normal;
    font-variant-numeric: normal;
    // Code darf umbrechen wenn nötig
    white-space: pre-wrap;
    word-break: break-all;
    overflow-wrap: break-word;
}
}

// =====
// ZUSATZ: Typografische Zeichen per CSS
// =====
// Für echte Zeichenersetzung (-- → -) benötigst du
// serverseitiges Processing (PHP/JS), aber CSS kann
// helfen mit Anführungszeichen:

// =====
// SHORTHAND MIXINS
// =====

@mixin typo-hyphenate {
    @include typo-wrap(1);
}

@mixin typo-whole-words {
    @include typo-wrap(2);
}

@mixin typo-no-break {
    @include typo-wrap(3);
}
```

Anwendung in Bricks Builder

```
// Beispiel 1: Fließtext mit Silbentrennung
%root% {
  @include typo-wrap(1);
}

// Beispiel 2: Überschriften - ganze Wörter
%root% {
  @include typo-wrap(2);
}

// Beispiel 3: Buttons/Labels - keine Umbrüche
%root% {
  @include typo-wrap(3);
}

// Oder mit Shorthand:
%root% {
  @include typo-hyphenate;    // = Modus 1
  @include typo-whole-words; // = Modus 2
  @include typo-no-break;    // = Modus 3
}
```

Wichtige Hinweise

1. HTML-Sprachattribut setzen

Damit die automatische Silbentrennung funktioniert, muss das `lang`-Attribut gesetzt sein:

```
<html lang="de"> <!-- Deutsch -->
<html lang="en"> <!-- Englisch -->
<html lang="de-CH"> <!-- Schweizerdeutsch -->
```

2. Gedankenstriche (-- → -)

CSS kann **keine Textzeichen ersetzen**. Für automatische Ersetzung von `--` zu `–` benötigst du:

Option A: JavaScript (clientseitig)

```
// In Bricks unter "Page Settings > Custom Code > Body Scripts"
document.querySelectorAll('p, h1, h2, h3, h4, h5, h6, li, td, th, blockquote')
  .forEach(el => {
    if (!el.closest('pre, code, script, style')) {
      el.innerHTML = el.innerHTML
        .replace(/--/g, '–')
        .replace(/\.\.\.\/g, '…');
    }
  });
```

Option B: WordPress-Filter (serverseitig, empfohlen)

```
// In functions.php oder Code-Snippets Plugin
add_filter('the_content', 'typografie_korrektur', 99);
add_filter('the_title', 'typografie_korrektur', 99);

function typografie_korrektur($content) {
  $replacements = [
    '–' => '–',    // Gedankenstrich
    '—' => '—',    // Geviertstrich
    '…' => '…',    // Auslassungspunkte
    '(c)' => '©',
    '(r)' => '®',
    '(tm)' => '™',
  ];
  return str_replace(
    array_keys($replacements),
    array_values($replacements),
    $content
  );
}
```

3. Browser-Support

Feature	Chrome	Firefox	Safari	Edge
---------	--------	---------	--------	------

hyphens: auto	☐	☐	☐	☐
quotes: auto	☐	☐	☐	☐
hyphenate-limit-*	△	☐	☐	△
text-wrap: pretty	☐	☐	☐	☐

Die Basis-Funktionalität wird von allen modernen Browsern unterstützt!

Revision #3

Created 2026-05-18 15:18:37 UTC by art10m

Updated 2026-05-18 16:04:08 UTC by art10m