

Highlight-text Quarto Extension

Quarto Extension

Mickaël CANOUIL, *Ph.D.*

This is a Quarto extension that allows to highlight text in a document for various formats: HTML, LaTeX, Typst, Docx, PowerPoint, Reveal.js, and Beamer.

Installation

```
quarto add mcanouil/quarto-highlight-text
```

This will install the extension under the `_extensions` subdirectory.

If you're using version control, you will want to check in this directory.

Usage

To use the extension, add the following to your document's front matter:

```
filters:  
- highlight-text
```

Then you can use the span syntax markup to highlight text in your document.

Basic Syntax

The extension supports both British and American English spelling for colour attributes:

```
[Red]{colour="#b22222" bg-colour="#abc123"} # UK spelling  
[Blue]{color="#0000ff" bg-color="#abc123"} # US spelling
```

Shorter Syntax

You can use abbreviated attribute names ([v1.1.1](#)):

```
[Red text]{fg="#b22222"}  
[Red background]{bg="#abc123"}  
[White on Red]{fg="ffffff" bg="#b22222"}  
[Text with solid border]{bc="#0000ff"}  
[Text with dashed border]{bc="#b22222" bs="dashed"}  
[Text with dotted border]{bc="#00aa00" border-style="dotted"}
```

Supported attributes:

- **Foreground (text) colour:** ink, fg, colour, or color
- **Background colour:** paper, bg, bg-colour, or bg-color
- **Border colour:** bc, border-colour, or border-color
- **Border style:** bs or border-style (values: solid, dashed, dotted, double; defaults to solid)

Font Colour

```
[Red text]{colour="#b22222"}
```

Red text

```
[Blue text]{color="#0000ff"}
```

Blue text

```
[Green text (ink alias)]{ink="#008000"}
```

Green text (ink alias)

Background Colour

```
[Red background]{bg-colour="#b22222"}
```

Red background

```
[Blue background]{bg-color="#0000ff"}
```

Blue background

```
[Yellow background (paper alias)]{paper="#ffff00"}
```

Yellow background (paper alias)

Font and Background Colour

```
[White text, Red background]{  
  fg="#ffffff" bg="#b22222"  
}
```

White text, Red background

```
[White text, Blue background]{  
  colour="#ffffff" bg-colour="#0000ff"  
}
```

White text, Blue background

```
[Black text, Orange background (ink/paper aliases)]{  
  ink="#000000" paper="#ffa500"  
}
```

Black text, Orange background (ink/paper aliases)

Border Colour

```
[Text with solid border]{bc="#b22222"}
```

Text with solid border

```
[Text with dashed border]{bc="#0000ff" bs="dashed"}
```

Text with dashed border

```
[Text with dotted border]{border-colour="#00aa00" border-style="dotted"}
```

Text with dotted border

```
[Text with double border]{bc="#8b008b" bs="double"}
```

Text with double border

Border with Background Colour

```
[Red border, yellow background]{bc="#b22222" bg="#ffff00"}
```

Red border, yellow background

```
[Blue border, white on red]{bc="#0000ff" fg="#ffffff" bg="#b22222"}
```

Blue border, white on red

More Examples

```
[text [with a link](https://quarto.org/)]{  
  color="#ffffff" bg-color="#00ffff"  
}
```

text with a link

Using Brand Colours

Define colours once in `_brand.yml` and reference them throughout your documents (v1.1.0):

```
color:  
  palette:  
    red: "#b22222"  
    custom-blue: "#0000ff"  
    primary: "#abc123"
```

Reference these colours directly by name:

```
[Red text]{fg="red"}  
[Custom background]{bg="custom-blue"}  
[Primary highlight]{bg="primary"}
```

i Note

The old `brand-color.` prefix syntax (e.g., `colour="brand-color.red"`) is deprecated but still supported ([v1.4.0](#)). You'll see a warning when using it. Use the colour name directly instead: `colour="red"`.

Light and Dark Theme Support

With Quarto CLI 1.7.28, you can define different colours for light and dark themes ([v1.2.0](#)):

Option 1: Define themes in document front matter:

```
brand:  
  light:  
    color:  
      palette:  
        fg: "#ffffff"  
        bg: "#b22222"  
  dark:  
    color:  
      palette:  
        fg: "#b22222"  
        bg: "#ffffff"
```

Option 2: Use external `_brand.yml` file:

```
brand:  
  light: _brand.yml  
  dark: _brand-dark.yml
```

Then reference theme-aware colours:

```
[This text adapts to theme]{fg="fg" bg="bg"}
```

i Note

Only HTML formats support dynamic light/dark mode switching. Other formats will use the light mode colours if available, or fall back to dark mode colours otherwise, unless specified otherwise.

Brand Colours Examples

Using colours defined in the **brand** section of this document's front matter:

```
[Light: White/Red | Dark: Red/White]{fg="fg" bg="bg"}
```

Light: White/Red | Dark: Red/White

Deprecated Syntax (Still Supported)

The old **brand-color.** prefix syntax still works but shows a deprecation warning:

```
[Using deprecated syntax]{colour="brand-color.fg" bg-colour="brand-color.bg"}
```

Using deprecated syntax

i Note

Use the colour name directly without the **brand-color.** prefix (e.g., **fg="fg"** instead of **colour="brand-color.fg"**).

Block-Level Highlighting

You can highlight entire blocks using div syntax.

```
::: {fg="#ffffff" bg="#0000ff"}
```

This is a block-level highlighted section.

It can contain multiple paragraphs, lists, and other block elements.

- Item one.
- Item two.

All content within this div is highlighted.

```
:::
```

This is a block-level highlighted section. It can contain multiple paragraphs, lists, and other block elements.

- Item one.
- Item two.

All content within this div is highlighted.

Block with Border

```
:::: {bc="#b22222" bg="#ffffcc"}
```

This block has a red solid border with a light yellow background.

```
:::
```

This block has a red solid border with a light yellow background.

```
:::: {bc="#0000ff" bg="#f0f0f0" bs="dashed"}
```

This block has a blue dashed border with a light grey background.

```
:::
```

This block has a blue dashed border with a light grey background.

```
:::: {bc="#00aa00" bs="dotted"}
```

This block has a green dotted border without background.

```
:::
```

This block has a green dotted border without background.

```
:::: {ink="#ffffff" paper="#800080"}
```

This block uses ink/paper aliases with white text on purple background.

```
:::
```

This block uses ink/paper aliases with white text on purple background.

Block with Brand Colours

```
:::: {fg="fg" bg="bg"}
```

This block uses brand colours that adapt to light/dark themes.

In light mode, this appears as white text on red background.

In dark mode, this appears as red text on white background.

```
:::
```

This block uses brand colours that adapt to light/dark themes.
In light mode, this appears as white text on red background.
In dark mode, this appears as red text on white background.

Limitations

LaTeX/PDF Output

The LaTeX `\colorbox` command does not support line wrapping for highlighted text with background colours. Long highlighted text may overflow or break awkwardly.

For inline highlighting: Use the `par=true` attribute to add `\parbox{\linewidth}` (Xe-LaTeX and PDFLaTeX only):

```
[Long text with background]{colour="#b22222" bg-colour="#abc123" par=true}
```

For block-level highlighting: Automatic line wrapping is enabled for all engines. Block divs automatically use `\parbox` for non-LuaLaTeX engines.

Best solution: Use LuaLaTeX as your PDF engine for proper line wrapping with the `lua-ul` package:

```
format:
pdf:
  pdf-engine: lualatex
```

i Note

LuaLaTeX is the default PDF engine in Quarto CLI 1.8.25.

Example without workaround (inline):

```
[Long text example without workaround]{colour="#b22222" bg-colour="#abc123"}
```

LaTeX `\colorbox` command does not support wrapping/line breaks in the text to be highlighted. This means

Example with `par=true` (inline):

```
[Long text example with par=true]{colour="#b22222" bg-colour="#abc123" par=true}
```

LaTeX `\colorbox` command does not support wrapping/line breaks in the text to be highlighted. This means that the above example will not work well in LaTeX output when using XeLaTeX or PDFLaTeX engines.

Example with block-level highlighting:

```
::: {colour="#b22222" bg-colour="#abc123"}
LaTeX ``\colorbox`` command does not support wrapping/line breaks in the text to be highlighted.
Block divs automatically handle line wrapping for all PDF engines.
:::
```

LaTeX `\colorbox` command does not support wrapping/line breaks in the text to be highlighted.
Block divs automatically handle line wrapping for all PDF engines, including XeLaTeX and PDFLaTeX.

PowerPoint Output

Links are not supported in highlighted text in PowerPoint output.

Border colour is not supported in PowerPoint output.

Word Output

Links are not supported in highlighted text in Word output.