

Typst Render

Quarto Extension

Mickaël CANOUIL, *Ph.D.*

A Quarto filter extension that compiles ```{typst}` code blocks to images (PNG, SVG, or PDF) using the Typst binary bundled with Quarto. This makes Typst diagrams, figures, tables, and equations usable across all output formats (HTML, PDF via LaTeX, DOCX, RevealJS, and more).

When the output format is Typst, blocks pass through natively without image conversion.

Installation

```
quarto add mcanouil/quarto-typst-render@0.5.0
```

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

If you are 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:
  - typst-render
```

For cross-referencing support, use timing control:

```
filters:
  - path: typst-render
    at: pre-quarto
```

Then write Typst code blocks in your document:

```

{typst}
#set text(size: 16pt)
Hello from *Typst*!

```

Basic Usage

A simple Typst block rendered as an image:

```

{typst}
#set text(size: 16pt)
Hello from *Typst*!
This is rendered as an image in non-Typst outputs.

```

Hello from **Typst**! This is rendered as an image in non-Typst outputs.

Per-Block Options

Use comment+pipe syntax (`//| key: value`) at the top of the code block to set per-block options.

Page Dimensions

Control the width, height, and margin of the rendered image page. These options set the Typst page dimensions for image compilation (non-Typst output formats). In native Typst output, `width` and `height` are ignored; `margin` is applied as block `inset`.

```

{typst}
//| width: 12cm
//| height: auto
//| margin: 1em
#set text(size: 14pt, fill: eastern)
#align(center)[
  #box(
    fill: luma(240),
    inset: 1em,
    radius: 4pt,
  )[A styled box with custom dimensions.]
]

```

A styled box with custom dimensions.

Image Format and DPI

Set the output image format and resolution:

```
```{typst}
//| format: png
//| dpi: 288
#set text(size: 14pt)
#align(center)[High-resolution PNG output.]
```
```

High-resolution PNG output.

Background Colour

Set a fill colour for the rendered image page. In native Typst output, the colour is applied as block fill.

```
```{typst}
//| background: luma(245)
//| width: 8cm
//| margin: 1em
#set text(size: 14pt)
#align(center)[
 Content with a light grey background.
]
```
```

Content with a light grey
background.

Cross-Referencing

Use `/// label:` and `/// cap:` for Quarto cross-references. Generic `cap` and `alt` options work with any label prefix. Prefix-specific variants (e.g., `fig-cap`, `tbl-alt`) override the generic options when the label matches. Any Quarto cross-reference type is supported (`fig-`, `tbl-`, `lst-`, etc.), including custom types.

See Figure 1 for a captioned figure.

```
```{typst}
/// label: fig-typst-shapes
/// cap: "A simple Typst diagram with shapes."
/// alt: "A blue circle and orange rectangle arranged horizontally."
#set text(size: 12pt)
#align(center)[
 #stack(dir: ltr, spacing: 1em)[
 #circle(radius: 1cm, fill: blue.lighten(60%))
][
 #rect(width: 2cm, height: 2cm, fill: orange.lighten(60%))
]
]
```
```

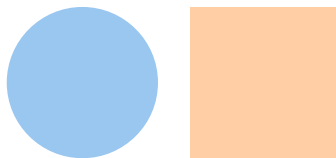


Figure 1: A simple Typst diagram with shapes.

Echo and Eval Control

The `echo` and `eval` options control whether source code is displayed and whether the block is compiled.

Echo: Source Code Alongside Output

Show the source code followed by the rendered output with `echo: true`:

```
#set text(size: 14pt)
$ sum_(k=0)^n k = (n(n+1))/2 $
```

$$\sum_{k=0}^n k = \frac{n(n+1)}{2}$$

Echo: Fenced Source Code

Use `echo: fenced` to display source code wrapped in fenced code block markers (````{typst}`), including any comment+pipe options (except `echo` itself). This mirrors Quarto's native `echo: fenced` behaviour for computational cells.

```
```{typst}
#set text(size: 14pt)
$ \int_0^\infty e^{-x} dx = 1 $
```
```

$$\int_0^\infty e^{-x} dx = 1$$

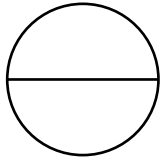
Eval: Source Only (No Compilation)

Display the source code without compiling with `eval: false`:

```
```{typst}
//| eval: false
#import "@preview/cetz:0.4.2": canvas, draw
#canvas({
 draw.circle((0, 0), radius: 1)
 draw.line((-1, 0), (1, 0))
})
```
```

With `eval: true` (the default), the block is compiled:

```
```{typst}
//| eval: true
#import "@preview/cetz:0.4.2": canvas, draw
#canvas({
 draw.circle((0, 0), radius: 1)
 draw.line((-1, 0), (1, 0))
})
```
```



Hidden Block

Setting `eval: false` and `echo: false` hides the block entirely. Nothing is rendered.

Include: Suppress All Output

Setting `include: false` suppresses all output (both source and rendered image). The block is completely hidden from the document, regardless of other options.

Output: Suppress Rendered Output

Setting `output: false` skips compilation and suppresses the rendered image. When combined with `echo: true`, only the source code is shown.

```
#set text(size: 14pt)
This source is visible, but the rendered image is suppressed.
```

Output Location (Reveal.js)

Input Variables

Pass key-value pairs to Typst code via `sys.inputs`. Global inputs are set in YAML; per-block inputs override them using comma-separated syntax.

```
```{typst}
//| input: greeting=Hello,subject=World
#set text(size: 16pt)
#sys.inputs.at("greeting"), #sys.inputs.at("subject")!
```
```

Hello, World!

Multi-Page Output

When Typst produces multiple pages, all pages are included by default. Use `pages` to select specific pages and `layout-ncol` to arrange them in columns.

All Pages Side by Side

```
```{typst}
//| layout-ncol: 2
//| width: 8cm
//| height: 6cm
#set text(size: 14pt)
#align(center + horizon)[
 Page 1: First page content.
]
#pagebreak()
#align(center + horizon)[
 Page 2: Second page content.
]
```
```

Page 1: First page content.

First Page Only

```
```{typst}
//| pages: 1
//| width: 8cm
//| height: 6cm
#set text(size: 14pt)
```

```
#align(center + horizon)[
 Page 1: This page is shown.
]
#pagebreak()
#align(center + horizon)[
 Page 2: This page is hidden.
]

```

**Page 1:** This page is shown.

## External File Rendering

Render an external `.typ` file instead of inline code:

```
```{typst}
//| file: _extensions/typst-render/_example.typ
#set text(size: 14pt)
#align(center)[
  This content was loaded from an external .typ file.
]
---
```

This content was loaded from an external `.typ` file.

Preamble

Typst code can be prepended before each block using the `preamble` option. This is useful for setting global styles or importing packages.

Inline Preamble

```
extensions:
  typst-render:
    preamble: '#set text(font: "Libertinus Serif")'
```

File-Based Preamble

Any value ending in `.typ` is treated as a file path:

```
extensions:
  typst-render:
    preamble: "preamble.typ"
```

Social Card

A repository social card built entirely in Typst, rendered as a high-resolution PNG.

```
```{typst}
//| label: fig-social-card
//| cap: "Social card for quarto-typst-render."
//| format: png
//| dpi: 288
//| width: 1280pt
//| height: 640pt
//| margin: 0pt

#let gradient-bg = gradient.linear(
 rgb("#1a1a2e"),
 rgb("#16213e"),
 rgb("#0f3460"),
 angle: 135deg,
)

#let accent = rgb("#e94560")
#let text-primary = rgb("#eaeaea")
#let text-secondary = rgb("#a0a0b0")
#let surface = rgb("#1a1a2e").lighten(10%)

#set page(fill: gradient-bg)
#set text(font: "Inter", fill: text-primary)

#place(
```

```

 top + left,
 dx: -40pt,
 dy: -40pt,
 circle(radius: 200pt, fill: accent.transparentize(92%)),
)
 #place(
 bottom + right,
 dx: 80pt,
 dy: 80pt,
 circle(radius: 300pt, fill: rgb("#0f3460").transparentize(60%)),
)

 #align(horizon)[
 #pad(x: 72pt)[
 #grid(
 columns: (1fr, auto),
 column-gutter: 48pt,
 align: (left + horizon, right + horizon),
 [
 #block(
 fill: accent,
 radius: 6pt,
 inset: (x: 14pt, y: 8pt),
) [
 #text(
 size: 14pt,
 weight: "bold",
 tracking: 1.5pt,
 fill: white,
) [QUARTO EXTENSION]
]

 #v(20pt)

 #text(size: 48pt, weight: "bold", tracking: -0.5pt)[
 typst-render
]

 #v(12pt)

 #text(size: 20pt, fill: text-secondary, weight: "regular")[
 Compile Typst code blocks to images#linebreak()
 for any Quarto output format.
]
]
]
]
]

```

```

 #v(28pt)

 #text(size: 16pt, fill: text-secondary)[
 #text(fill: accent, weight: "bold")[mcanouil]
 #h(4pt)
 #text(fill: text-secondary.transparentize(50%))[/]
 #h(4pt)
 quarto-typst-render
]
],
)
]
```

```

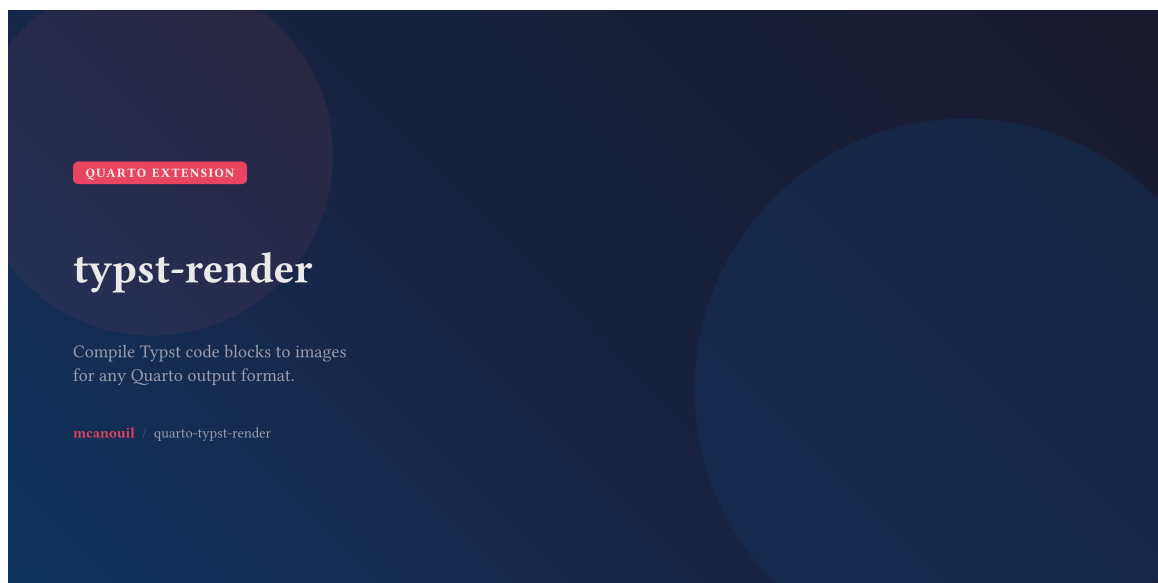


Figure 2: Social card for quarto-typst-render.

Global Configuration

Configure the filter globally in your document YAML. Most options can be set globally and overridden per block using comment+pipe syntax (`///| key: value`). See [Global-Only Options](#) for options that cannot be overridden per block.

```

extensions:
  typst-render:
    dpi: "288"
    margin: "1em"

```

```
format: png
cache: true
```

Options

| Option | Type | De-fault | Description |
|------------------------------|----------------|--------------------|--|
| <code>format</code> | string | (auto) | Image format: <code>png</code> , <code>svg</code> , <code>pdf</code> . |
| <code>dpi</code> | string | "144" | Pixels per inch (PNG only). |
| <code>width</code> | string | "auto" | Page width for image compilation (ignored in Typst output). |
| <code>height</code> | string | "auto" | Page height for image compilation (ignored in Typst output). |
| <code>margin</code> | string | "0.5em" | Page margin for image compilation; block <code>inset</code> in Typst output. |
| <code>background</code> | string | "none" | Page fill for image compilation; block <code>fill</code> in Typst output. |
| <code>preamble</code> | string | "" | Typst code or path to a <code>.typ</code> file prepended before user code. |
| <code>cache</code> | boolean string | <code>true</code> | Cache compiled images. Use <code>"clean"</code> to also remove stale cache files. |
| <code>input</code> | object | (none) | Key-value pairs passed as <code>--input</code> flags to Typst CLI. |
| <code>file</code> | string | (none) | Path to external <code>.typ</code> file to render. |
| <code>echo</code> | boolean string | <code>false</code> | Show Typst source code alongside output (<code>true</code> , <code>false</code> , <code>fenced</code>). |
| <code>eval</code> | boolean | <code>true</code> | Compile Typst code to image. |
| <code>include</code> | boolean | <code>true</code> | Include block in output. Set <code>false</code> to suppress entirely. |
| <code>output</code> | boolean | <code>true</code> | Show rendered output. Set <code>false</code> to skip compilation. |
| <code>output-location</code> | string | (none) | Output placement in Reveal.js (<code>fragment</code> , <code>slide</code> , <code>column</code> , <code>column-fragment</code>). |
| <code>classes</code> | string | (none) | Space-separated CSS classes on the output image (e.g., <code>r-stretch</code>). |
| <code>pages</code> | string | "all" | Pages to include from multi-page output: <code>all</code> , <code>1</code> , <code>1-3</code> , <code>2,5</code> , <code>3-</code> . |
| <code>layout-ncol</code> | string | (none) | Number of columns for arranging multi-page output. Omit for vertical stack. |

Any unknown option with a string value is forwarded as an HTML attribute on the output image element (e.g., `/// style: "max-height: 300px;"`). Values that look like booleans (`true/false`) must be quoted to be forwarded (e.g., `/// data-lazy: "true"`).

Global-Only Options

These options can only be set in the document YAML and cannot be overridden per block.

| Option | Type | Default | Description |
|---------------------------|--------------|---------------------|---|
| <code>root</code> | string | (project directory) | Root directory for Typst compilation. |
| <code>font-path</code> | string array | (none) | Path or list of paths to directories containing additional fonts. |
| <code>package-path</code> | string | (none) | Path to a directory containing Typst packages (offline/reproducible). |

Per-Block Cross-Referencing Options

| Option | Type | Description |
|---------------------------------|--------|---|
| <code>label</code> | string | Quarto cross-ref label (e.g., <code>fig-x</code> , <code>tbl-y</code> , <code>lst-z</code>). |
| <code>cap</code> | string | Caption text for the labelled block. |
| <code>alt</code> | string | Alternative text for accessibility. |
| <code><prefix>-cap</code> | string | Prefix-specific caption (e.g., <code>fig-cap</code>). Overrides <code>cap</code> for matching labels. |
| <code><prefix>-alt</code> | string | Prefix-specific alt text (e.g., <code>fig-alt</code>). Overrides <code>alt</code> for matching labels. |

Auto-Selected Image Format

| Output Format | Default Image Format |
|-----------------|-----------------------|
| HTML / RevealJS | <code>svg</code> |
| LaTeX / Beamer | <code>pdf</code> |
| Typst | (native pass-through) |
| DOCX / PPTX | <code>png</code> |
| Other | <code>png</code> |

Echo/Eval Behaviour

| eval | echo | Result |
|--------------------|---------------------|--|
| <code>true</code> | <code>false</code> | Image only (default). |
| <code>true</code> | <code>true</code> | Source code block + image below. |
| <code>true</code> | <code>fenced</code> | Fenced source code block (with markers) + image. |
| <code>false</code> | <code>true</code> | Source code listing only. |

| <code>eval</code> | <code>echo</code> | Result |
|--------------------|---------------------|---|
| <code>false</code> | <code>fenced</code> | Fenced source code listing only (with markers). |
| <code>false</code> | <code>false</code> | Nothing rendered (hidden block). |

The `include` and `output` options take precedence over the `eval/echo` matrix:

- `include: false` hides the entire block regardless of `eval/echo` settings.
- `output: false` skips compilation and shows only the source code (if `echo` is enabled).