



Mastering Quarto CLI

From Authoring to Publishing

Mickaël CANOUIL, Ph.D.^{1*}

Saturday, the 7th of February, 2026



Affiliations

¹Independent Contractor, Lille, France

ORCID

Mickaël CANOUIL, Ph.D.: [0000-0002-3396-4549](https://orcid.org/0000-0002-3396-4549)

*Corresponding author: pro@mickael.canouil.dev

Table of Contents

| | |
|---|---|
| 1. Overview | 3 |
| 2. Workshop Objectives | 3 |
| 3. Prerequisites | 4 |
| 4. Program Sessions | 4 |
| 4.1. Part 1 | 4 |
| 4.1.1. Session 1: Introduction & Setup | 4 |
| 4.1.2. Session 2: Authoring Essentials | 5 |
| 4.1.3. Session 3: Embedding Computations & Code | 5 |
| 4.2. Part 2 | 5 |
| 4.2.1. Session 4: Formats & Projects | 5 |
| 4.2.2. Session 5: Formatting & Branding | 6 |
| 4.2.3. Session 6: Publishing | 6 |
| 4.2.4. Session 7: Closing & Next Steps | 7 |

1. Overview

This comprehensive workshop provides an immersive journey through the Quarto ecosystem, designed to transform participants from newcomers to confident practitioners capable of creating sophisticated, reproducible documents and interactive publications.

Through seven structured sessions, attendees will master the complete Quarto workflow from initial setup and Markdown fundamentals to advanced theming, multi-format publishing, and professional deployment strategies. The workshop emphasises practical, hands-on learning with real-world applications, covering project architecture, computational integration across R/Python/Julia, advanced formatting techniques, and modern publishing workflows.

By the end of this workshop, participants will have developed the expertise to create polished, professional publications that seamlessly integrate narrative content with executable code whilst maintaining reproducibility across multiple output formats.

2. Workshop Objectives

- Establish a solid foundation in Quarto fundamentals, including installation, project types, and the command-line interface.
- Master authoring essentials through Markdown fundamentals, Quarto-specific features, and YAML configuration.
- Integrate computational workflows by embedding R, Python, and Julia code with proper execution control and caching strategies.
- Develop professional publications using advanced project structures, multi-format optimisation, and custom formatting techniques.
- Implement cohesive branding and visual identity across formats using Bootstrap theming, `brand.yml` systems, and Pandoc templating.
- Deploy projects confidently through various publishing platforms including Quarto Pub, GitHub Pages, and automated workflows.

3. Prerequisites

Prior to the workshop, please ensure you have the following installed on your computer.

- Quarto CLI 1.8.* or higher: <https://quarto.org/docs/get-started>
- Computing languages (one or more):
 - ▶ R 4.5.* or higher: <https://cran.r-project.org>
 - ▶ Python 3.13.* or higher: <https://www.python.org/downloads/>
 - ▶ Julia 1.12.* or higher: <https://julialang.org/downloads/>

Alternatively, you can use GitHub Codespaces:



[Open in GitHub Codespaces](#)



```
./.devcontainer/setup.sh --what <all|r|python|julia>
```

4. Program Sessions

4.1. Part 1

4.1.1. Session 1: Introduction & Setup

- Welcome & Overview
 - ▶ Workshop objectives and participant goals.
 - ▶ Quarto overview: purpose, strengths, and reproducibility benefits.
 - ▶ What Quarto is and how it works with Pandoc.
 - ▶ Brief look at the [Quarto Guide](#).
- Installation & Environment Setup
 - ▶ Installing Quarto (installer, Homebrew, Chocolatey).
 - ▶ Verify installation with `quarto check`.
 - ▶ Quarto as a command-line interface.
- Quarto Projects
 - ▶ Project types: default, website, blog, book, manuscript.
 - ▶ Creating projects with `quarto create project`.
 - ▶ Writing with various editors (VS Code, RStudio, Jupyter, etc.).

4.1.2. Session 2: Authoring Essentials

- Markdown Fundamentals
 - ▶ Creating structured documents: headings, lists, links, and text formatting.
 - ▶ Best practices for writing clean, reproducible content in Quarto.
 - ▶ Working with code blocks, tables, and mathematical expressions.
- Quarto Markdown
 - ▶ Callout blocks, shortcodes, and Mermaid diagrams.
 - ▶ Cross-references for figures, tables, and custom elements.
 - ▶ Code annotations and advanced formatting features.
- YAML Configuration
 - ▶ Document headers and multi-format output.
 - ▶ Metadata inheritance and project configuration.
 - ▶ Extensions and customisation options.

4.1.3. Session 3: Embedding Computations & Code

- Computing Environments
 - ▶ Setup R, Python, and Julia for Quarto.
 - ▶ Engine configuration and code cell structure.
 - ▶ Understanding multi-language workflows.
- Execution Options
 - ▶ Code cell options using comment + pipe syntax.
 - ▶ Control code visibility, evaluation, and output.
 - ▶ Global execution settings in YAML.
- Cache and Freeze
 - ▶ Cache for development speed vs freeze for collaboration.
 - ▶ Understanding Jupyter and Knitr caching systems.
 - ▶ CLI commands for cache and freeze management.
- Parameters
 - ▶ Creating dynamic document variations.
 - ▶ Defining parameters in Python and R.
 - ▶ Command-line parameter passing and YAML files.

4.2. Part 2

4.2.1. Session 4: Formats & Projects

- Quarto Projects
 - ▶ Project types and architecture: default, website, blog, manuscript, book.
 - ▶ Project type decision framework based on publishing goals.

- ▶ Understanding project structure and configuration files.
- Multiple Formats
 - ▶ Multi-format output optimisation with format-specific options.
 - ▶ Conditional content by format using content-visible/hidden classes.
 - ▶ Cross-format resource management and figure optimisation.
- Advanced Project Features
 - ▶ Shared metadata and configuration with `_metadata.yml`.
 - ▶ Pre and post-render scripts for workflow automation.
 - ▶ Project profiles for different deployment scenarios.
 - ▶ Environment variables and resource management.

4.2.2. Session 5: Formatting & Branding

- Bootstrap Theming & Customisation
 - ▶ Bootstrap 5 integration and built-in theme selection.
 - ▶ Progressive customisation from YAML options to light CSS.
 - ▶ Theme layering and CSS custom properties.
- Unified Branding with `Brand.yml`
 - ▶ `Brand.yml` systems for cross-format consistency.
 - ▶ Colour palettes, typography, and logo management.
 - ▶ Integration with Bootstrap and format-specific options.
- Pandoc Templating & Extensions
 - ▶ Understanding Pandoc's templating syntax fundamentals.
 - ▶ Variables, conditionals, and loops for dynamic content.
 - ▶ Developing brand extensions for organisational reuse.
- Typst Customisation
 - ▶ Typst templates and partials for PDF branding.
 - ▶ Template structure and `brand.yml` integration.
 - ▶ Creating custom Typst formats with consistent visual identity.

4.2.3. Session 6: Publishing

- Publishing Options Overview
 - ▶ Comparison of Quarto Pub, GitHub Pages, Netlify, and other hosting services.
 - ▶ Understanding benefits and limitations of each platform.
 - ▶ Choosing the right publishing method for different use cases.
- Quarto Pub and GitHub Pages
 - ▶ Step-by-step Quarto Pub deployment workflow.
 - ▶ Three methods for GitHub Pages: docs folder, `quarto publish`, and GitHub Actions.
 - ▶ Understanding `_publish.yml` and repository configuration.
- Code Execution Strategies
 - ▶ Local execution with `freeze` vs CI execution.

- ▶ Environment management and dependency handling.
- ▶ Version control best practices for published projects.
- Advanced Topics
 - ▶ Custom domains configuration.
 - ▶ Automated deployment workflows.
 - ▶ Troubleshooting common publishing issues.

4.2.4. Session 7: Closing & Next Steps

- Recapitulation:
Summarise the day's key learning outcomes.
- Resource Sharing:
Direct participants to further Quarto documentation and tutorials.
- Feedback:
Request feedback to refine future sessions and identify areas for advanced exploration.