Formats & Projects

Session 4

Mickaël CANOUIL, *Ph.D.*

pro@mickael.canouil.dev

Independent Contractor

Saturday, the 7th of February, 2026

 | License: CC BY-NC-SA 4.0

- Session 4: Formats & Projects
- Session Objectives
- Learning Objectives

Session Overview

 | License: CC BY-NC-SA 4.0

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.
- **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.
- **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.

Session Objectives

This session explores Quarto's project architecture and format ecosystem, covering project types, multi-format optimisation, and template systems for scalable workflows.

 | License: CC BY-NC-SA 4.0

Learning Objectives

 | License: CC BY-NC-SA 4.0

Learning Objectives

By the end of this session, participants will be able to:

 | License: CC BY-NC-SA 4.0

Learning Objectives

By the end of this session, participants will be able to:

- Select appropriate project types for different publishing scenarios.

 | License: CC BY-NC-SA 4.0

Learning Objectives

By the end of this session, participants will be able to:

- Select appropriate project types for different publishing scenarios.
- Configure multi-format outputs with format-specific optimisations.

 | License: CC BY-NC-SA 4.0

Learning Objectives

By the end of this session, participants will be able to:

- Select appropriate project types for different publishing scenarios.
- Configure multi-format outputs with format-specific optimisations.
- Use template systems for rapid project initiation.

 | License: CC BY-NC-SA 4.0

Learning Objectives

By the end of this session, participants will be able to:

- Select appropriate project types for different publishing scenarios.
- Configure multi-format outputs with format-specific optimisations.
- Use template systems for rapid project initiation.
- Implement collaborative workflows with proper project structure.

 | License: CC BY-NC-SA 4.0

Learning Objectives

By the end of this session, participants will be able to:

- Select appropriate project types for different publishing scenarios.
- Configure multi-format outputs with format-specific optimisations.
- Use template systems for rapid project initiation.
- Implement collaborative workflows with proper project structure.
- Understand format capabilities and limitations.

 | License: CC BY-NC-SA 4.0

Quarto Projects

- Project Types and Architecture
- Project Type Decision Framework
- Default Project Structure
- Website Projects
- Blog Projects (website + listing)
- Manuscript Projects
- Book Projects

 | License: CC BY-NC-SA 4.0

Project Types and Architecture

 | License: CC BY-NC-SA 4.0

Project Types and Architecture

Recall the project types introduced in Session 1. Here we examine their configuration and architecture in more detail.

 | License: CC BY-NC-SA 4.0

Project Types and Architecture

Recall the project types introduced in Session 1. Here we examine their configuration and architecture in more detail.

Bash

```
quarto create project
```

 | License: CC BY-NC-SA 4.0

Project Types and Architecture

Recall the project types introduced in Session 1. Here we examine their configuration and architecture in more detail.

Bash

```bash
quarto create project
```

BASH

```
? Type
❯ default
  website
  blog
  manuscript
  book
  confluence
```

 | License: CC BY-NC-SA 4.0

Project Type Decision Framework

Choose your project type based on your publishing goals:

Scenario	Project Type	Key Features
Multiple unrelated documents	`default`	Flexible structure, shared metadata
Single research paper	`manuscript`	Academic formatting, citations, submission-ready
Public documentation/blog	`website` / `blog`	Navigation, search, responsive design
Long-form content	`book`	Cross-chapter references, multiple outputs

 | License: CC BY-NC-SA 4.0

Default Project Structure

The most flexible option for diverse content:

YAML

```yaml
project:
  title: "Default"
```

 | License: CC BY-NC-SA 4.0

Website Projects

Optimised for web publishing:

YAML

```yaml
project:
  type: website

website:
  title: "Website"
  navbar:
    left:
      - href: index.qmd
        text: Home
      - about.qmd

format:
  html:
    theme:
      - cosmo
      - brand
    css: styles.css
    toc: true
```

Blog Projects (website + listing)

Designed for regular posts and updates:

YAML

```yaml
project:
  type: website

website:
  description: "A blog built with Quarto"
  site-url: https://your-website-url.example.com # you must change this appropriately for RSS feeds to work properly
  title: "Blog"
  navbar:
    right:
      - about.qmd
      - icon: github
        href: https://github.com/
      - icon: bluesky
        href: https://bsky.app/
format:
  html:
    theme:
      - cosmo
      - brand
    css: styles.css


```

Manuscript Projects

Optimised for academic publishing:

YAML

```yaml
project:
  type: manuscript

manuscript:
  article: index.qmd

format:
  html:
    comments:
      hypothesis: true
  docx: default
  jats: default

  # (other formats)
  # pdf: default

execute:
  freeze: true
```

Book Projects

For comprehensive, structured content:

YAML

```yaml
project:
  type: book

book:
  title: "Book"
  author: "Norah Jones"
  date: "2/7/2026"
  chapters:
    - index.qmd
    - intro.qmd
    - summary.qmd
    - references.qmd

bibliography: references.bib

format:
  html:
    theme:
      - cosmo
      - brand
  pdf:
    documentclass: scrreprt
```

- Multi-Format Output Optimisation
- Conditional Content by Format
- Cross-Format Resource Management

Multiple Formats

 | License: CC BY-NC-SA 4.0

Multi-Format Output Optimisation

Balance capability with compatibility by configuring format-specific options that leverage each output type's strengths.

YAML

```yaml
title: "Multi-Format Document"
fig-width: 6
fig-height: 4
format:
  html:
    theme: cosmo
    code-fold: true
    code-tools: true
    toc: true
    fig-width: 8
    fig-height: 6
  pdf:
    documentclass: article
    geometry: margin=1in
    fontsize: 11pt
    colorlinks: true
  typst:
    brand-mode: dark
  revealjs:
    brand-mode: dark
    fig-width: 10
    fig-height: 6
```

(2)

(1)

Multi-Format Output Optimisation

Balance capability with compatibility by configuring format-specific options that leverage each output type's strengths.

YAML

```yaml
title: "Multi-Format Document"
fig-width: 6
fig-height: 4
format:
  html:
    theme: cosmo
    code-fold: true
    code-tools: true
    toc: true
    fig-width: 8
    fig-height: 6
  pdf:
    documentclass: article
    geometry: margin=1in
    fontsize: 11pt
    colorlinks: true
  typst:
    brand-mode: dark
  revealjs:
    brand-mode: dark
    fig-width: 10
    fig-height: 6
```

2

1 Setting format-specific optio

Multi-Format Output Optimisation

Balance capability with compatibility by configuring format-specific options that leverage each output type's strengths.

```yaml
title: "Multi-Format Document"
fig-width: 6
fig-height: 4
format:
  html:
    theme: cosmo
    code-fold: true
    code-tools: true
    toc: true
    fig-width: 8
    fig-height: 6
  pdf:
    documentclass: article
    geometry: margin=1in
    fontsize: 11pt
    colorlinks: true
  typst:
    brand-mode: dark
  revealjs:
    brand-mode: dark
    fig-width: 10
    fig-height: 6
```

Setting shared and format-specific options.

2

1

Multi-Format Output Optimisation

Balance capability with compatibility by configuring format-specific options that leverage each output type's strengths.

YAML

```yaml
title: "Multi-Format Document"
fig-width: 6
fig-height: 4
format:
  html:
    theme: cosmo
    code-fold: true
    code-tools: true
    toc: true
    fig-width: 8
    fig-height: 6
  pdf:
    documentclass: article
    geometry: margin=1in
    fontsize: 11pt
    colorlinks: true
  typst:
    brand-mode: dark
  revealjs:
    brand-mode: dark
    fig-width: 10
    fig-height: 6
```

(2)

(1)

Conditional Content by Format

 | License: CC BY-NC-SA 4.0

Conditional Content by Format

Control what content appears in different output formats.

 | License: CC BY-NC-SA 4.0

Conditional Content by Format

Control what content appears in different output formats.

Key Classes

- `.content-visible` - Show content only for specific formats.
- `.content-hidden` - Hide content from specific formats.

 | License: CC BY-NC-SA 4.0

Conditional Content by Format

Control what content appears in different output formats.

Key Classes

- `.content-visible` - Show content only for specific formats.
- `.content-hidden` - Hide content from specific formats.

Conditions

- `when-format` / `unless-format` - Control visibility based on output format.
- `when-meta` / `unless-meta` - Control visibility based on metadata.
- `when-profile` / `unless-profile` - Control visibility based on profile.

 | License: CC BY-NC-SA 4.0

Conditional Content by Format

 | License: CC BY-NC-SA 4.0

Conditional Content by Format

Syntax

 | License: CC BY-NC-SA 4.0

Conditional Content by Format

Syntax

- MARKDOWN

```
::: {.content-visible when-format="html"}
HTML-only content
:::
```

Conditional Content by Format

Syntax

- MARKDOWN

```
::: {.content-visible when-format="html"}
HTML-only content
:::
```

- MARKDOWN

```
[Typst/PDF only text]{.content-visible when-format="typst"}
```

 | License: CC BY-NC-SA 4.0

Conditional Content by Format

Syntax

- MARKDOWN

```
::: {.content-visible when-format="html"}
HTML-only content
:::
```

- MARKDOWN

```
[Typst/PDF only text]{.content-visible when-format="typst"}
```

- MARKDOWN

````
```{.python .content-visible when-format="html"}
print("Only shows in HTML")
```
````

Conditional Content by Format

Syntax

- MARKDOWN

```
::: {.content-visible when-format="html"}
HTML-only content
:::
```

- MARKDOWN

```
[Typst/PDF only text]{.content-visible when-format="typst"}
```

- MARKDOWN

````
```{.python .content-visible when-format="html"}
print("Only shows in HTML")
```
````

- MARKDOWN

````
::: {.content-visible when-format="html"}
```{python}
print("Only shows in HTML but is evaluated in all formats")
```
:::
````

Cross-Format Resource Management

 | License: CC BY-NC-SA 4.0

Cross-Format Resource Management

Optimise images and assets for different outputs:

 | License: CC BY-NC-SA 4.0

Cross-Format Resource Management

Optimise images and assets for different outputs:

YAML

```yaml
format:
  html:
    fig-format: svg      # Vector graphics for web
    fig-dpi: 150
  pdf:
    fig-format: pdf      # Vector graphics for print
    fig-dpi: 300
  typst:
    fig-format: png      # Raster for compatibility
    fig-dpi: 150
```

 | License: CC BY-NC-SA 4.0

- Shared Metadata and Configuration
- Resource Management
- Pre and Post Render Scripts
- Environment Variables
- Variables
- Project Profiles

Advanced Project Features

 | License: CC BY-NC-SA 4.0

Shared Metadata and Configuration

 | License: CC BY-NC-SA 4.0

Shared Metadata and Configuration

Use `_metadata.yml` for common settings to be shared by documents in a directory:

 | License: CC BY-NC-SA 4.0

Shared Metadata and Configuration

Use `_metadata.yml` for common settings to be shared by documents in a directory:

subdirectory/_metadata.yml

```yaml
format:
  html:
    theme: custom
execute:
  echo: false
  warning: false
```

 | License: CC BY-NC-SA 4.0

Resource Management

 | License: CC BY-NC-SA 4.0

Resource Management

Organise assets that needs to be copied to the output directory:

 | License: CC BY-NC-SA 4.0

Resource Management

Organise assets that needs to be copied to the output directory:

_quarto.yml

```yaml
project:
  resources:
    - "data/*.csv"
    - "images/"
    - "!temp/"       # Exclude temporary files
```

 | License: CC BY-NC-SA 4.0

Pre and Post Render Scripts

 | License: CC BY-NC-SA 4.0

Pre and Post Render Scripts

Automate workflows before and after rendering.

 | License: CC BY-NC-SA 4.0

Pre and Post Render Scripts

Automate workflows before and after rendering.

Configuration

_quarto.yml

```yaml
project:
  pre-render:
    - data-prep.py
    - generate-content.R
  post-render:
    - cleanup.sh
    - deploy.ts
```

 | License: CC BY-NC-SA 4.0

Pre and Post Render Scripts

 | License: CC BY-NC-SA 4.0

Pre and Post Render Scripts

Key Features

 | License: CC BY-NC-SA 4.0

Pre and Post Render Scripts

Key Features

- Scripts run from **project root directory**.

 | License: CC BY-NC-SA 4.0

Pre and Post Render Scripts

Key Features

- Scripts run from **project root directory**.
- **Re-computes metadata** after pre-render (can generate new files).

 | License: CC BY-NC-SA 4.0

Pre and Post Render Scripts

Key Features

- Scripts run from **project root directory**.
- **Re-computes metadata** after pre-render (can generate new files).
- Supports **shell commands**, **Python**, **R**, **TypeScript**, **Lua**.

 | License: CC BY-NC-SA 4.0

Pre and Post Render Scripts

Environment Variables Available

- `QUARTO_PROJECT_RENDER_ALL` - “1” if rendering entire project.
- `QUARTO_PROJECT_OUTPUT_DIR` - Output directory path.
- `QUARTO_PROJECT_INPUT_FILES` - List of input files (pre-render).
- `QUARTO_PROJECT_OUTPUT_FILES` - List of output files (post-render).

 | License: CC BY-NC-SA 4.0

Pre and Post Render Scripts

Conditional Execution Example

PYTHON

```python
# Only run expensive operations on full renders
import os
if os.environ.get("QUARTO_PROJECT_RENDER_ALL") == "1":
    # Run expensive data processing
    process_large_dataset()
```

 | License: CC BY-NC-SA 4.0

Environment Variables

 | License: CC BY-NC-SA 4.0

Environment Variables

Manage project environment variables with files and profiles:

 | License: CC BY-NC-SA 4.0

Environment Variables

Manage project environment variables with files and profiles:

- `_environment` - Default environment variables for all renders.
- `_environment.local` - Local overrides (auto-ignored by Git).
- `_environment-{profile}` - Profile-specific variables.
- `_environment.required` - Documentation of required variables.

 | License: CC BY-NC-SA 4.0

Variables

 | License: CC BY-NC-SA 4.0

Variables

Insert dynamic content with shortcodes:

- `{{< var variable_name >}}` - From `_variables.yml` file.
- `{{< meta field_name >}}` - From document YAML metadata.
- `{{< env ENV_VARIABLE >}}` - From environment variables.

Variables

Insert dynamic content with shortcodes:

- `{{< var variable_name >}}` - From `_variables.yml` file.
- `{{< meta field_name >}}` - From document YAML metadata.
- `{{< env ENV_VARIABLE >}}` - From environment variables.

Markdown

MARKDOWN

```
1 - {{< meta title >}}
2 - {{< meta subtitle >}}
```

 | License: CC BY-NC-SA 4.0

Variables

Insert dynamic content with shortcodes:

- `{{< var variable_name >}}` - From `_variables.yml` file.
- `{{< meta field_name >}}` - From document YAML metadata.
- `{{< env ENV_VARIABLE >}}` - From environment variables.

Markdown

MARKDOWN

```
1 - {{< meta title >}}
2 - {{< meta subtitle >}}
```

Output

- Formats & Projects
- Session 4

Project Profiles

 | License: CC BY-NC-SA 4.0

Project Profiles

Adapt projects for different scenarios with configuration variants.

 | License: CC BY-NC-SA 4.0

Project Profiles

Adapt projects for different scenarios with configuration variants.

What Are Profiles?

 | License: CC BY-NC-SA 4.0

Project Profiles

Adapt projects for different scenarios with configuration variants.

What Are Profiles?

- Different configurations for the same project (production, development, basic/advanced versions).

 | License: CC BY-NC-SA 4.0

Project Profiles

Configuration Files

- `_quarto.yml` - Base configuration.
- `_quarto-{profile}.yml` - Profile-specific config that merges with base.
- `_environment-{profile}` - Profile-specific environment variables.

 | License: CC BY-NC-SA 4.0

Project Profiles

Activation

BASH

```bash
# Environment variable
export QUARTO_PROFILE=production
quarto render

# Command line
quarto render --profile production

# Multiple profiles
quarto render --profile production,advanced
```

 | License: CC BY-NC-SA 4.0

Project Profiles

Content Targeting

MARKDOWN

```
::: {.content-visible when-profile="advanced"}
Advanced content only
:::

::: {.content-hidden unless-profile="basic"}
Hidden for basic profile
:::
```

 | License: CC BY-NC-SA 4.0

Project Profiles

Profile Groups & Defaults

YAML

```yaml
profile:
  group: [basic, advanced]
  default: development
```

 | License: CC BY-NC-SA 4.0

- Exercise Overview
- Part 1: Create a Project Structure
- Part 2: Multi-Format Configuration
- Part 3: Conditional Content
- Part 4: Test Multiple Formats
- Part 5: Add Project Features
- Success Criteria

Hands-On Exercise: Building Your First Project

 | License: CC BY-NC-SA 4.0

Exercise Overview

Objective: Transform your computational portfolio from Session 3 into a structured project and explore multiple output formats.

Example Code: Exercises

BASH

```bash
tar -xzf "04-exercises.tar.gz" -C "04-formats-projects"
```

Part 1: Create a Project Structure

Convert your portfolio into a project:

Website Project | Default Project | Book Project

Create a `_quarto.yml` file:

YAML

```yaml
project:
  type: website

website:
  title: "My Learning Portfolio"
  navbar:
    left:
      - href: index.qmd
        text: Home
      - about.qmd

format:
  html:
    theme: cosmo
    toc: true
```

Part 1: Create a Project Structure

Convert your portfolio into a project:

Website Project | **Default Project** | Book Project

Create a `_quarto.yml` file:

YAML

```yaml
project:
  title: "My Portfolio Collection"

format:
  html:
    theme: cosmo
    toc: true
  typst: default
```

Part 1: Create a Project Structure

Convert your portfolio into a project:

Website Project | Default Project | **Book Project**

Create a `_quarto.yml` file:

YAML

```yaml
project:
  type: book

book:
  title: "My Quarto Journey"
  chapters:
    - index.qmd
    - portfolio.qmd
    - reflection.qmd

format:
  html:
    theme: cosmo
```

Part 2: Multi-Format Configuration

Add multiple output formats to your project:

Basic Multi-Format | Advanced Configuration | Presentation Format

Add to your `_quarto.yml`:

YAML

```yaml
format:
  html:
    theme: cosmo
    code-fold: true
  typst:
    margin:
      top: 2cm
      bottom: 2cm
  docx: default
```

Part 2: Multi-Format Configuration

Add multiple output formats to your project:

Basic Multi-Format | **Advanced Configuration** | Presentation Format

Add format-specific options:

YAML

```yaml
format:
  html:
    theme: cosmo
    code-fold: true
    fig-width: 8
    fig-height: 6
  typst:
    fig-width: 6
    fig-height: 4
  pdf:
    geometry:
      - margin=1in
    fontsize: 11pt
```

Part 2: Multi-Format Configuration

Add multiple output formats to your project:

Basic Multi-Format | Advanced Configuration | Presentation Format

Add slides to your outputs:

YAML

```yaml
format:
 html: default
 revealjs:
   theme: dark
   fig-width: 10
   fig-height: 6
```

 | License: CC BY-NC-SA 4.0

Part 3: Conditional Content

Add format-specific content to your documents:

Format-Specific Sections | Code Visibility by Format | Format-Specific Messages

Add to your content:

MARKDOWN

```
::: {.content-visible when-format="html"}
This interactive content only appears in HTML output.
:::

::: {.content-visible when-format="typst"}
This formatted text appears only in Typst/PDF output.
:::

::: {.content-hidden when-format="revealjs"}
This detailed content is hidden in presentations.
:::
```

 | License: CC BY-NC-SA 4.0

Part 3: Conditional Content

Add format-specific content to your documents:

Format-Specific Sections | **Code Visibility by Format** | Format-Specific Messages

Control code display:

MARKDOWN

```
```{python .content-visible when-format="html"}
# Interactive code for web
import plotly.express as px
# Create interactive plot
```

```{python .content-visible unless-format="html"}
# Static plot for print formats
import matplotlib.pyplot as plt
# Create static plot
```
```

 | License: CC BY-NC-SA 4.0

Part 3: Conditional Content

Add format-specific content to your documents:

Format-Specific Sections | Code Visibility by Format | Format-Specific Messages

Add dynamic text:

MARKDOWN

```
1 [[View the interactive version online](link-to-html)]{.content-visible when-format="typst"}
2
3 [*This is the web version with interactive features*]{.content-visible when-format="html"}
```

Part 4: Test Multiple Formats

Render your project in different formats:

Single Format | All Formats | Preview Mode

Try rendering to specific formats:

BASH

```bash
quarto render --to html
quarto render --to typst
quarto render --to docx
```

 | License: CC BY-NC-SA 4.0

Part 4: Test Multiple Formats

Render your project in different formats:

Single Format | All Formats | Preview Mode

Render all configured formats:

```bash
quarto render
```

 | License: CC BY-NC-SA 4.0

Part 4: Test Multiple Formats

Render your project in different formats:

Single Format | All Formats | Preview Mode

Use preview for development:

BASH

```bash
quarto preview
```

 | License: CC BY-NC-SA 4.0

Part 5: Add Project Features

Enhance your project with additional features:

Shared Metadata | Resource Management | Environment Variables

Create `_metadata.yml`:

YAML

```yaml
author: "Your Name"
date: today
execute:
  echo: true
  warning: false
```

Part 5: Add Project Features

Enhance your project with additional features:

Shared Metadata | **Resource Management** | Environment Variables

Add to `_quarto.yml` :

YAML

```yaml
project:
  resources:
    - "data/"
    - "images/"
    - "*.csv"
```

 | License: CC BY-NC-SA 4.0

Part 5: Add Project Features

Enhance your project with additional features:

Shared Metadata | Resource Management | Environment Variables

Create `_environment` file:

TXT

```
QUARTO_PYTHON=/path/to/python
API_KEY=your-key-here
```

 | License: CC BY-NC-SA 4.0

Success Criteria

You've successfully completed the exercise if you can:

- Create a properly structured project with `_quarto.yml`.
- Configure at least two different output formats.
- Add format-specific content that shows/hides appropriately.
- Successfully render your project to multiple formats.
- Test that conditional content works correctly.

 | License: CC BY-NC-SA 4.0

Mickaël CANOUIL, *Ph.D.*

Independent Contractor

pro@mickael.canouil.dev

Website	GitHub	LinkedIn	Bluesky	Mastodon
mickael.canouil.fr	@mcanouil	MickaelCanouil	@mickael.canouil.fr	@MickaelCanouil

 | License: CC BY-NC-SA 4.0