Quarto Basics

Table of contents

Write, cite, and render with Quarto.

Chapter 1

What is Quarto?

Quarto is a publishing system for technical docs. You write a `.qmd` file (Markdown + optional code), and Quarto renders it to **HTML**, **PDF**, slides, and more. By default it renders to **HTML** unless you specify otherwise.

Chapter 2

Install & verify

- Install Quarto: https://quarto.org/docs/get-started/
- Verify in a terminal:

```
quarto check
quarto --version
```

PDF note: PDF output requires a TeX engine. The easiest path is TinyTeX.

```
quarto install tinytex
```

(Or run `mccourse_setup()` in R to install prerequisites automatically.)

Chapter 3

Your first document

Create `report.qmd`:

```
---
title: "My Report"
format: html   # default is html if omitted
---

# Hello, Quarto

This is **Markdown**. Add text, lists, images, and code blocks.
```

Render to HTML:

```
quarto render report.qmd --to html
```

Chapter 4

Course patterns: Documents vs. Book

In this course you'll use Quarto in two modes:

1. **Weekly documents** (one file → one output)

- Lives under folders like `week_01/`
- You open a `.qmd` and render it directly

2. **Journal as a Quarto book** (many chapters → one book)

- Lives in `journal/`
- Each entry is a chapter in `journal/entries/`
- The book's table of contents is defined in `journal/_quarto.yml`

4.1 Minimal book `_quarto.yml` (journal)

```yaml
project:
  type: book
  output-dir: _book

book:
  title: "Course Journal"
  author: "YOUR NAME"
  date: today
  downloads: [pdf, epub]
  chapters:
    - index.qmd
    # --- auto-managed: do not edit below ---
    # BEGIN AUTO
    # END AUTO
    # --- end auto-managed ---

format:
  html:
    toc: true
    number-sections: true
  pdf:
    toc: true
    number-sections: true
```

> The `scripts/update_chapters.R` script rewrites the lines between `BEGIN AUTO` and `END AUTO` to include all dated files under `entries/` (sorted by filename). Don't edit that block manually.

Render the whole journal book from inside `journal/`:

```
quarto render
```

Outputs appear in `journal/_book/`.

Chapter 5

PDF-first and multi-format options

You can request specific formats in the document YAML:

```
---
title: "PDF-first example"
format: pdf
---
```

Or enable multiple formats with options:

```
---
title: "PDF & HTML"
format:
  pdf:
    toc: true
    number-sections: true
  html:
    toc: true
    code-fold: true
---
```

Chapter 6

Render from RStudio or VS Code

- **RStudio**: open a `.qmd` and click **Render**.
- **VS Code**: install the Quarto extension; use **Quarto: Render** or run `quarto render` in the integrated terminal.

Chapter 7

Project basics (render only what you intend)

Quarto projects let you render a folder of documents with shared settings and control *which* files render:

```yaml
# site/_quarto.yml (example for a website/docs project)
project:
  type: website
  output-dir: _site
  render:
    - index.qmd
    - pages/**/*.qmd
```

That whitelist prevents Quarto from scanning and rendering the entire repository.

Chapter 8

Execution control: cache & freeze (optional)

- **Cache** stores chunk results so long computations aren't re-run.
- **Freeze** reuses previously rendered results during project renders:

```yaml
execute:
  freeze: auto
```

Add this to the project's `_quarto.yml` (site or book) to stabilize outputs until sources change.

Chapter 9

Journal chapter template (prompt-only)

Each journal chapter is created from `entry_template.qmd` and contains **three prompts** plus a **Response** section. Students answer **one** prompt only. The template includes an **automatic word count**:

- **MC 451:** 250–300 words
- **MC 501:** 450–500 words

Rendering the chapter shows the current count and whether it's in range.

> New chapters are created by running `source("scripts/new_journal_entry.R")` inside `journal/`. The script injects prompts and updates `_quarto.yml`.

Chapter 10

Common options

- **Table of contents**: `toc: true`
- **Code folding** (HTML): `code-fold: true`
- **Section numbering**: `number-sections: true`
- **PDF fine-tuning**: `geometry`, `documentclass`, `keep-tex`, `include-in-header`, etc.

Chapter 11

Troubleshooting quick hits

- **"No TeX installation detected"** → run `quarto install tinytex` or `mccourse_setup()`.
- **Whole repo renders unintentionally** → set `project.render` (see above).
- **New journal chapter not appearing** → commit both the new `entries/YYYY-MM-DD.qmd` **and** `journal/_quarto.yml`, then re-render.
- **Word count out of range** → ensure only the **Response** section contains your text; target the correct range for your course.