Designing a Course With Quarto

Norah Jones

2025-12-09

Table of contents

1 Introduction

This book is a practical, end-to-end guide for educators, instructional designers, and academic technologists who want to modernize how they develop, organize, and deliver course materials. It brings together version control, reproducible documents, Quarto publishing, and Blackboard Ultra integration into a unified workflow that is scalable, maintainable, and future-proof.

The goal of this book is not only to show you *what* tools to use, but also *how* these tools work together to form a complete course-production pipeline—one that enhances instructional quality while reducing long-term maintenance effort.

To achieve this, the book is organized into several core skill domains that build on one another.

1.1 Tool Setup Essentials

Before creating any course materials or writing your first Quarto document, you will learn how to properly set up your development environment. This chapter guides you through installing and configuring:

- Git, Git LFS, Github CLI, and GitHub
- Antigravity IDE (Google's VS Code–based environment with agents and extensions)
- Quarto CLI
- Python 3.12
    - virtual environments
    - pip, setuptools, wheel
    - pandas, numpy, openpyxl, and matplotlib
- R, RTools (Windows), and renv
- Pandoc and Graphviz
- Core Antigravity plugins for Jupyter, Quarto, Python, and remote development

By the end of this section, you will have a fully functional authoring environment where Quarto, Git, and Blackboard workflows run reliably.

This chapter ensures **every user starts with a stable, consistent toolset**, preventing configuration issues later in the course.

1.2 Understanding the Authoring Layer: Markdown, LaTeX, and Quarto Syntax

Before building full websites, it is essential to understand the authoring languages that Quarto builds upon. This chapter explains:

- The basics of Markdown formatting
- Extensions used by Quarto (fenced divs, callouts, code blocks, metadata)
- How LaTeX is used within Quarto for mathematical expressions, typesetting, and PDF output
- How citations, cross-references, and bibliographies work
- Code cell execution in Python, R, and other languages
- How Quarto merges authoring, computation, and formatting
- Differences between Markdown, Jupyter notebooks, `.Rmd`, and `.qmd`

This chapter demystifies the underlying structure of Quarto and gives users confidence in writing documents that are clean, readable, and publishable.

Users will come away understanding not only *syntax*, but also the *philosophy* behind reproducible authoring.

1.3 Setting Up Git, GitHub, and a Version-Controlled Workspace

This chapter teaches the fundamentals of version control and how to use GitHub as your central repository for course materials. You will learn how to:

- Create a GitHub account
- Configure Git identity and SSH keys
- Create your first repository (`username.github.io`)
- Clone, branch, commit, push, and pull changes
- Use GitHub as a publishing platform and collaboration space
- Apply version control to Quarto documents, slides, and assets

By the end of this chapter, users will understand how to maintain a clean, traceable version history for their course, and how to safely experiment and update materials without losing work.

1.4 Building a Quarto Website Structure from Scratch

Once the tools and authoring skills are in place, this chapter walks through creating a complete Quarto site, including:

- Initializing new projects
- Creating navigation, pages, and assets
- Organizing modules, slides, assignments, and labs
- Using Reveal.js for presentations
- Embedding media, figures, and interactive elements
- Structuring a microsite that mirrors a course syllabus

By the end of this chapter, readers will have a fully functional course website that is clean, navigable, and professional.

1.5 Designing a Blackboard Ultra Course Structure

This chapter teaches how to translate your Quarto website into a Blackboard Ultra course shell. Users will learn how to:

- Create ultra-style learning modules
- Embed HTML content exported from Quarto
- Add learning materials, assignments, rubrics, and media
- Optimize Blackboard structure for student navigation
- Use Blackboard as the delivery layer on top of your Quarto-generated content

The result is a Blackboard course that mirrors your Quarto site, offering a consistent student experience across platforms.

1.6 Integrating Quarto Deliverables into Blackboard Ultra

This final integration chapter teaches how to:

- Export Quarto pages, slides, and content as HTML
- Insert HTML into Blackboard Ultra pages
- Embed external web content
- Manage updates through GitHub synchronization
- Maintain consistency between Blackboard and your Quarto repository

Users finish with a fully connected system: Quarto as the authoring engine, GitHub as the repository, and Blackboard Ultra as the delivery interface.

2 What You Will Achieve by the End of This Book

By progressing through the chapters, you will develop the ability to:

- Build, maintain, and publish an entire course website
- Use version control confidently and professionally
- Write reproducible, portable instructional materials
- Integrate your website into a university LMS
- Maintain a clean, scalable course development workflow

This book aims to transform your approach to digital course design—making your materials more organized, more professional, and significantly easier to maintain across semesters.

3 Tool Setup

This chapter provides a step-by-step recipe for installing and configuring all tools required for building course materials with Quarto, managing content with GitHub, and integrating outputs into Blackboard Ultra.

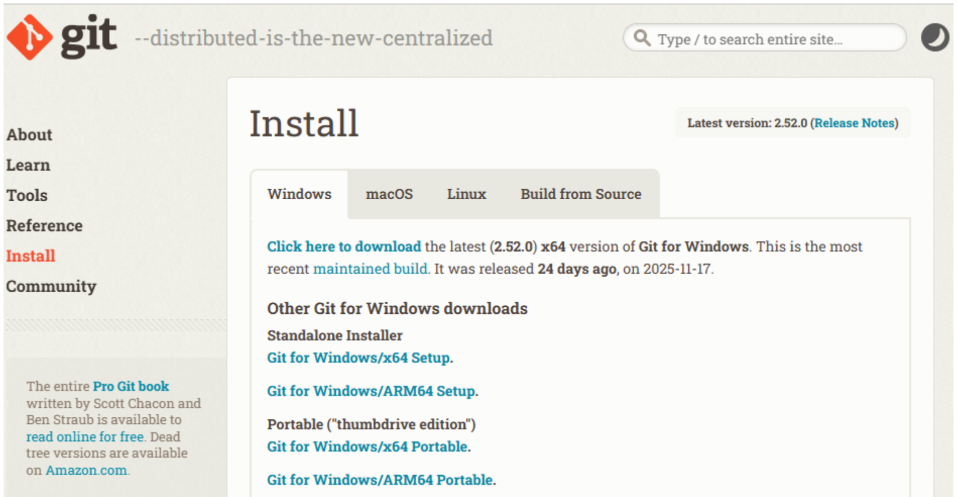
While GitHub is strongly recommended for version control and long-term course management, users who prefer to work entirely on local machines or cloud-synced drives (Google Drive, OneDrive, Dropbox, iCloud Drive) may skip the repository creation steps and proceed directly to **Chapter 3**.

However, the GitHub-based workflow—especially when paired with Antigravity or VS Code—is highly preferred for reproducibility, collaboration, and maintaining a clean version history of your teaching materials.

This recipe assumes no prior experience. Each step includes instructions, verification checks, and placeholders for screenshots.

4 Install Git

Git is the version control system used throughout the recommended workflow. You can choose the platform from the installation on the Git website.



Install

Latest version: 2.52.0 (Release Notes)

Click here to download the latest (2.52.0) x64 version of Git for Windows. This is the most recent maintained build. It was released 24 days ago, on 2025-11-17.

Other Git for Windows downloads

Standalone Installer

Git for Windows/x64 Setup.

Git for Windows/ARM64 Setup.

Portable ("thumbdrive edition")

Git for Windows/x64 Portable.

Git for Windows/ARM64 Portable.

The entire Pro Git book written by Scott Chacon and Ben Straub is available to read online for free. Dead tree versions are available on Amazon.com.

Figure 4.1: Installing Git - Linux Screenshot Placeholder

4.1 Instructions

1. Install Git using your operating system:

   1. macOS:

```
brew install git
```

   2. Windows: Download https://git-scm.com/download/win
   3. Linux:

```
sudo apt update && sudo apt install git -y
```

4.2 Verification

```
git --version
```

---

5 Step 2: Install GitHub CLI

GitHub CLI (gh) enables authentication and repository operations directly from the terminal.

5.0.1 Instructions

Download and install from: https://cli.github.com/

Verify installation:

```
gh --version
```

5.0.2 Screenshot Placeholder

```
![GitHub CLI Installation](images/toolsetup-ghcli.png)
```

---

6 Step 3: Create a GitHub Account

A GitHub account stores repositories, manages version history, and integrates directly with Antigravity or VS Code.

6.0.1 Instructions

1. Visit https://github.com/signup
2. Create an account
3. Choose a professional username

6.0.2 Screenshot Placeholder

```
![GitHub Signup Page](images/toolsetup-github-signup.png)
```

---

7 Step 4: Create Your GitHub Pages Repository

(Optional for users who plan to work only on local/cloud drives)

This step creates your personal website:

```
https://yourusername.github.io
```

If you prefer to work without GitHub, skip to **Chapter 3**. However, the GitHub-based workflow is strongly recommended.

7.0.1 Instructions

1. Go to https://github.com/new
2. Set repository name:

   ```
   yourusername.github.io
   ```

3. Select Public
4. Add README and .gitignore
5. Create repository

7.0.2 Screenshot Placeholder

```
![Create GitHub Pages Repo](images/toolsetup-ghpages.png)
```

---

8 Step 5: Configure Git Identity and SSH Keys

(Required only if using GitHub workflow)

8.1 5.1 Configure identity

```
git config --global user.name "Your Name"
git config --global user.email "you@example.com"
```

8.2 5.2 Generate an SSH key

```
ssh-keygen -t ed25519
```

8.3 5.3 Add the public key to GitHub

```
cat ~/.ssh/id_ed25519.pub
```

Add it to https://github.com/settings/keys

8.4 5.4 Test authentication

```
ssh -T git@github.com
```

8.4.1 Screenshot Placeholder

![SSH Key Added](images/toolsetup-sshkey.png)

---

9 Step 6: Clone Your GitHub Pages Repository

(Skip this step if not using GitHub)

9.0.1 Instructions

Clone repository:

```
git clone git@github.com:yourusername/yourusername.github.io.git
cd yourusername.github.io
```

Create a sample file:

```
echo "Welcome to my GitHub Pages site." >> index.md
```

Stage and commit:

```
git add .
git commit -m "Initial commit"
```

Push:

```
git push origin main
```

Pull:

```
git pull
```

9.0.2 Screenshot Placeholder

`![Local Repo Clone](images/toolsetup-clone.png)`

---

10 Step 7: Install Quarto

Quarto is the publishing framework used in this book.

10.0.1 Instructions

Download from: https://quarto.org/docs/download/

Verify:

```
quarto --version
```

10.0.2 Screenshot Placeholder

`![Quarto Installation](images/toolsetup-quarto.png)`

---

11 Step 8: Install Pandoc

Pandoc is required for document format conversion.

11.0.1 Instructions

- macOS:

  ```
  brew install pandoc
  ```

- Linux:

  ```
  sudo apt install pandoc -y
  ```

Windows installer already includes Pandoc.

11.0.2 Screenshot Placeholder

`![Pandoc Installation](images/toolsetup-pandoc.png)`

---

12 Step 9: Install Graphviz

Graphviz supports diagrams and visual structures.

12.0.1 Instructions

- macOS:

  ```
  brew install graphviz
  ```

- Linux:

  ```
  sudo apt install graphviz -y
  ```

- Windows: Download https://graphviz.org/download/

12.0.2 Screenshot Placeholder

```
![Graphviz Installation](images/toolsetup-graphviz.png)
```

---

13 Step 10: Install Python 3.12

13.0.1 Instructions

Download from: https://www.python.org/downloads/

Select "Add Python to PATH" during installation.

13.0.2 Verification

```
python3 --version
```

13.0.3 Screenshot Placeholder

`![Python Installation](images/toolsetup-python.png)`

---

14 Step 11: Install R and RTools

14.0.1 Instructions

Install R: https://cran.r-project.org/

Windows users install RTools: https://cran.r-project.org/bin/windows/Rtools/

14.0.2 Screenshot Placeholder

`![R Installation](images/toolsetup-r.png)`

---

15 Step 12: Install Antigravity IDE

Antigravity is the preferred IDE for this workflow because it integrates:

- GitHub
- Jupyter
- Python
- Quarto
- Remote development
- AI agents
- Extension ecosystem

15.0.1 Instructions

Download: https://antigravity.google/

Sign in with Google or GitHub.

15.0.2 Screenshot Placeholder

```
![Antigravity IDE Installer](images/toolsetup-antigravity.png)
```

---

16 Step 13: Install Required Antigravity Extensions

The following extension groups must be installed:

16.0.1 Jupyter Extensions

- Jupyter
- Jupyter Notebook Renderers
- Jupyter Keymap
- Jupyter Cell Tags
- Jupyter PowerToys
- Jupyter Slide Show

16.0.2 Quarto Extensions

- Quarto
- Markdown Preview Enhanced
- Markdown All-in-One
- Quarto Wingman
- Quarto Wizard
- Markdown Code Runner
- Pandoc Citer

16.0.3 Python Extensions

- Python Extension Pack
- Pylance
- Python Debugger
- autopep8
- isort
- Pylint
- MagicPython

16.0.4 Remote Development Extensions

- Remote – SSH
- Remote Explorer
- GitHub Repositories
- Dev Containers
- Remote Repositories
- Remote – Tunnels

16.0.5 Screenshot Placeholder

`![Antigravity Extensions Installed](images/toolsetup-extensions.png)`

---

17 Step 14: Create Python and R Environments (Optional)

17.1 Python virtual environment

```
python3 -m venv .venv
```

Activate:

macOS/Linux:

```
source .venv/bin/activate
```

Windows:

```
.venv\Scripts\activate
```

Install packages:

```
pip install numpy pandas jupyter matplotlib quarto
```

17.2 R environment using renv

```
install.packages("renv")
renv::init()
install.packages(c("tidyverse", "knitr", "rmarkdown"))
renv::snapshot()
```

17.2.1 Screenshot Placeholder

`![Environment Setup](images/toolsetup-env.png)`

18 Step 15: Verification Checklist

Check that each core tool is installed:

18.0.1 Git

```
git --version
```

18.0.2 Quarto

```
quarto --version
```

18.0.3 SSH

```
ssh -T git@github.com
```

18.0.4 Python

```
python3 --version
```

18.0.5 R

```
version
```

18.0.6 Screenshot Placeholder

```
![Verification Outputs](images/toolsetup-verification.png)
```

---

19 Step 16: Next Steps

If you chose not to create a GitHub repository, proceed directly to **Chapter 3**, where you will learn the authoring fundamentals of Markdown, LaTeX, and Quarto.

If you completed the GitHub setup, you now have a fully configured development environment ready for:

- Authoring (Chapter 3)
- Building Quarto websites (Chapter 4)
- Connecting content to Blackboard Ultra (Chapters 5–6)

20 Introduction

21

22

23 Summary

In summary, this book has no content whatsoever.

References