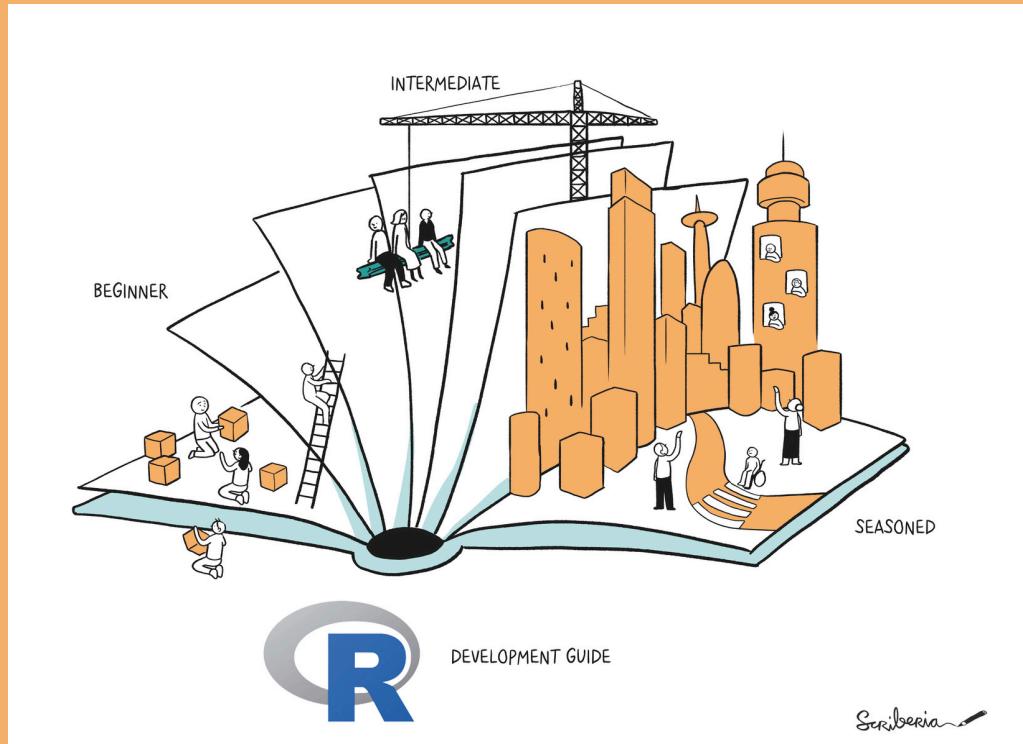


R Development Guide



R Development Guide. This illustration is created by Scriberia with The Turing Way community, used under a CC-BY 4.0 licence. DOI: <https://zenodo.org/records/13882307>

collaboratively authored by the
R Contribution Working Group

Building Reproducible and Interactive Analytics Courses

Table of contents

| | |
|---|-----------|
| 1 Designing an Online Course: Where Do I Start? | 4 |
| 2 Introduction | 5 |
| 3 What This Book Offers | 6 |
| 3.1 For BU Faculty Use Only | 6 |
| 4 Building a Quarto Microsite Structure | 7 |
| 5 Quarto Microsite Builder — Step-by-Step Guide for Educators | 8 |
| 5.1 1. Create the Base Quarto Website Using the CLI (Recommended) | 8 |
| 5.2 2. Replace the Default Pages With Your Course Pages | 9 |
| 5.3 3. Create Module Folders (M0–M8) | 9 |
| 5.4 4. Inside Each Module Folder, Add the Standard File Structure | 10 |
| 5.5 5. Update <code>_quarto.yml</code> to Configure the Microsite | 10 |
| 5.6 6. Recommended Templates for Educators | 11 |
| 5.6.1 Lecture Notes Template | 11 |
| 5.6.2 Presentation Template (Reveal.js) | 12 |
| 5.6.3 Tutorial Template | 12 |
| 5.6.4 Lab Template | 13 |
| 5.6.5 Assignment Template | 13 |
| 5.6.6 Project Step Template | 13 |
| 5.7 7. Build an Instructor TODO Website | 14 |
| 5.8 8. Render the Website | 14 |
| 6 Introduction to Git and GitHub | 16 |
| 7 Overview | 17 |
| 8 Learning Objectives | 18 |
| 9 Understanding Version Control | 19 |
| 9.1 What is Version Control? | 19 |
| 9.2 Git: A Distributed Version Control System | 19 |
| 9.3 GitHub: A Platform for Collaboration | 19 |

| | |
|--|-----------|
| 10 Setting Up Git | 20 |
| 10.1 Installing Git | 20 |
| 10.2 Configuring Git | 20 |
| 11 Creating a GitHub Account | 21 |
| 12 Initializing a Git Repository | 22 |
| 12.1 Creating a New Repository on GitHub | 22 |
| 12.2 Cloning the Repository Locally | 22 |
| 13 Making Your First Commit | 23 |
| 14 Exploring GitHub's Web Interface | 24 |
| 15 Summary | 25 |
| 16 Summary | 26 |
| References | 27 |

1 Designing an Online Course: Where Do I Start?

2 Introduction

As data and computation continue to reshape disciplines across the university, faculty are increasingly called upon to integrate code, collaboration, and scalable infrastructure into their teaching. Yet, despite this growing demand, instructors often lack structured guidance on how to develop and maintain reproducible instructional materials across tools, platforms, and modalities.

Teach Reproducibly is a faculty-facing resource developed by the **Metropolitan College**, in partnership with **BU ELIVE Services** and **BU Virtual**, to support instructors in building robust, version-controlled, and scalable course content for use in **Boston University's Blackboard Ultra environment**.

This book provides practical guidance for instructors teaching in data-rich and computation-heavy domains — including analytics, computer science, statistics, engineering, public health, and beyond — who are committed to developing high-quality, maintainable instructional workflows.

3 What This Book Offers

Rather than focusing solely on curriculum content, this book emphasizes **how content is developed, maintained, and shared reproducibly**. It introduces foundational tools and design principles that support instructional reuse, transparency, and collaboration, including:

- Literate programming environments with **Quarto**
- Source control and team workflows using **Git** and **GitHub**
- Collaborative assignments via **GitHub Classroom**
- Cloud-based computing with **AWS Academy** and **VS Code.dev**
- Browser-native runtime environments such as **WebR** and **Pyodide**
- Seamless integration with **Blackboard Ultra** and BU's LMS ecosystem

This resource supports faculty in developing modular instructional pipelines, enabling version-controlled teaching materials, reproducible research-in-practice demonstrations, and seamless cloud-backed student exercises. Each module aligns with BU's pedagogical standards and technical infrastructure.

3.1 For BU Faculty Use Only

This resource is intended for **internal instructional development and support** within Boston University. It is not licensed for external reproduction, distribution, or commercial use.

For inquiries, access to editable templates, or instructional support, please contact:

- BU ELIVE Services: elive@bu.edu
- BU Virtual Instructional Design: buv@bu.edu
- Metropolitan College Academic Affairs: metacad@bu.edu

4 Building a Quarto Microsite Structure

Designing an Quarto-based Course

Here is the **updated educator-friendly step-by-step guide**, now explicitly integrating the **Quarto CLI command**:

`quarto create project website <name-of-project>` This ensures educators can start from the correct Quarto project scaffold immediately.

Everything below is written as a clean instruction manual educators can follow without needing prior technical knowledge.

5 Quarto Microsite Builder — Step-by-Step Guide for Educators

A reusable workflow for creating professional, consistent course websites with modules 0–8.

5.1 1. Create the Base Quarto Website Using the CLI (Recommended)

Educators should begin by using the built-in Quarto command:

```
quarto create project website <name-of-project>
```

Example:

```
quarto create project website CloudAnalyticsCourse
```

This automatically creates:

```
CloudAnalyticsCourse/
  _quarto.yml
  index.qmd
  styles.css  (depending on template)
```

Now open the folder in VS Code, RStudio, or any preferred editor.

5.2 2. Replace the Default Pages With Your Course Pages

In the root of the project, create the standard site pages:

```
index.qmd  
syllabus.qmd  
deliverables.qmd  
schedule.qmd  
todo.qmd      # Instructor-only checklist
```

Purpose of Each Page

- **index.qmd** → Course homepage
 - **syllabus.qmd** → Course policies
 - **deliverables.qmd** → Assignments & grading
 - **schedule.qmd** → Module or weekly timeline
 - **todo.qmd** → Instructor checklists
-

5.3 3. Create Module Folders (M0–M8)

In your project folder, add:

```
M0/  
M1/  
M2/  
M3/  
M4/  
M5/  
M6/  
M7/  
M8/
```

Nine modules total.

5.4 4. Inside Each Module Folder, Add the Standard File Structure

For example, inside **M1/** create:

```
M1_LN1.qmd      # Lecture Note 1
M1_LN2.qmd      # Lecture Note 2
M1_P1.qmd       # Presentation 1
M1_P2.qmd       # Presentation 2
M1_T1.qmd       # Tutorial 1
M1_T2.qmd       # Tutorial 2
M1_Lab1.qmd     # Lab 1
M1_Lab2.qmd     # Lab 2
M1_A.qmd        # Assignment
M1_Proj.qmd     # Project step
M1_highlights.qmd # Summary landing page
M1L1_figures/
M1L2_figures/
```

Copy this structure for **every module** from 0–8, adjusting the naming prefix.

5.5 5. Update `_quarto.yml` to Configure the Microsite

Replace the default file created by Quarto CLI with a course-oriented version:

```
project:
  type: website

website:
  title: "Course Name"
  navbar:
    left:
      - href: index.qmd
        text: Home
      - href: syllabus.qmd
        text: Syllabus
      - href: deliverables.qmd
        text: Deliverables
      - href: schedule.qmd
```

```
text: Schedule
- text: Modules
  menu:
    - text: Module 0
      href: M0/M0_highlights.qmd
    - text: Module 1
      href: M1/M1_highlights.qmd
    - text: Module 2
      href: M2/M2_highlights.qmd
    - text: Module 3
      href: M3/M3_highlights.qmd
    - text: Module 4
      href: M4/M4_highlights.qmd
    - text: Module 5
      href: M5/M5_highlights.qmd
    - text: Module 6
      href: M6/M6_highlights.qmd
    - text: Module 7
      href: M7/M7_highlights.qmd
    - text: Module 8
      href: M8/M8_highlights.qmd

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

This creates a clean navigation system, anchored around module highlights pages.

5.6 6. Recommended Templates for Educators

Educators can copy/paste these into any file.

5.6.1 Lecture Notes Template

```
---  
title: "Module X - Lecture Y"  
format: html  
---
```

```
# Learning Objectives
```

```
# Content
```

```
# Examples
```

```
# Summary
```

5.6.2 Presentation Template (Reveal.js)

```
---  
title: "Module X - Presentation Y"  
format: revealjs  
---
```

```
# Slide 1
```

```
---  
# Slide 2
```

5.6.3 Tutorial Template

```
---  
title: "Module X - Tutorial Y"  
format: html  
---
```

```
# Steps
```

```
# Exercises
```

5.6.4 Lab Template

```
---  
title: "Module X - Lab Y"  
format: html  
---  
  
# Lab Instructions  
  
# Tasks  
  
# Submission Notes
```

5.6.5 Assignment Template

```
---  
title: "Module X - Assignment"  
format: html  
---  
  
# Overview  
  
# Requirements  
  
# Rubric
```

5.6.6 Project Step Template

```
---  
title: "Module X - Project Step"  
format: html  
---  
  
# Goal of This Step  
  
# Required Deliverables  
  
# Integration With Final Project
```

5.7 7. Build an Instructor TODO Website

Create todo.qmd:

```
---
```

```
title: "Instructor TODO Checklist"
format: html
---  
  
# Course-Level Tasks
- [ ] Update syllabus
- [ ] Customize schedule
- [ ] Adjust grading/deliverables  
  
# Module Tasks (Repeat for Each Module)
## Module 1
- [ ] Lecture Note 1
- [ ] Lecture Note 2
- [ ] Presentation 1
- [ ] Presentation 2
- [ ] Tutorial 1
- [ ] Tutorial 2
- [ ] Lab 1
- [ ] Lab 2
- [ ] Assignment
- [ ] Project Step
- [ ] Highlights Page
```

Educators can check tasks off during development.

5.8 8. Render the Website

Once content is ready, educators run:

`quarto render`

Or click **Render Website** if using RStudio.

6 Introduction to Git and GitHub

7 Overview

In the realm of modern education, especially within data-centric disciplines, the ability to manage and collaborate on code is paramount. Git and GitHub serve as foundational tools that facilitate version control, collaborative development, and reproducible research. This chapter aims to introduce these tools, providing educators with the skills necessary to integrate them into their teaching workflows.

8 Learning Objectives

By the end of this chapter, you will be able to:

- Understand the concepts of version control and its significance in educational contexts.
- Differentiate between Git and GitHub and comprehend their respective roles.
- Install and configure Git on your local machine.
- Create a GitHub account and set up your user profile.
- Perform basic Git operations: initializing a repository, committing changes, and pushing to GitHub (GitHub 2024).

9 Understanding Version Control

9.1 What is Version Control?

Version control is a system that records changes to a file or set of files over time, enabling you to recall specific versions later. In educational settings, this allows instructors to:([arxiv.org][2])

- Track the evolution of course materials.
- Collaborate with colleagues on curriculum development.
- Revert to previous versions of teaching resources when necessary.([docs.github.com][3])

9.2 Git: A Distributed Version Control System

Git is a distributed version control system that allows multiple contributors to work on a project simultaneously without interfering with each other's work. Each contributor has a complete copy of the repository, including its history, on their local machine. This setup enhances collaboration and ensures data integrity.([docs.github.com][4])

For more information, refer to the official Git documentation: [About Git](#).([docs.github.com][4])

9.3 GitHub: A Platform for Collaboration

GitHub is a cloud-based platform that hosts Git repositories, providing tools for collaboration, code review, and project management. In an educational context, GitHub can be used to:

- Distribute assignments and course materials.
- Facilitate student collaboration on projects.
- Manage feedback and revisions efficiently.([docs.github.com][5], [en.wikipedia.org][6])

Learn more about GitHub here: [About GitHub](#).([docs.github.com][7])

10 Setting Up Git

10.1 Installing Git

To begin using Git, you need to install it on your local machine.

- **Windows:** Download the installer from [Git for Windows](#) and follow the setup instructions.
- **macOS:** Install Git using Homebrew:

```
brew install git
```

- **Linux:** Use your distribution's package manager. For example, on Ubuntu:

```
sudo apt-get install git
```

Detailed installation instructions can be found here: [Set up Git](#).([docs.github.com][8])

10.2 Configuring Git

After installation, configure your Git username and email address. These details will be associated with your commits.

```
git config --global user.name "Your Name"  
git config --global user.email "your_email@bu.edu"
```

To verify your configuration:

```
git config --list
```

11 Creating a GitHub Account

To utilize GitHub's features, create an account:

1. Navigate to [GitHub's Sign Up Page](#).
2. Enter your email address, create a username and password.
3. Follow the prompts to complete the setup process.

After account creation, you can personalize your profile by adding a profile picture, bio, and other details.

12 Initializing a Git Repository

12.1 Creating a New Repository on GitHub

1. Log in to your GitHub account.
2. Click on the “+” icon in the top-right corner and select “New repository”.
3. Provide a repository name, e.g., `teaching-materials`.
4. Optionally, add a description.
5. Choose the repository’s visibility (public or private).
6. Initialize the repository with a README file.
7. Click “Create repository”.([docs.github.com][1])

For a step-by-step guide, refer to: [Quickstart for repositories](#).([docs.github.com][1])

12.2 Cloning the Repository Locally

To work on your repository locally:

1. Navigate to your repository on GitHub.
2. Click on the “Code” button and copy the URL.
3. Open your terminal and run:

```
git clone https://github.com/yourusername/teaching-materials.git
```

4. Navigate into the cloned directory:

```
cd teaching-materials
```

13 Making Your First Commit

Let's add a new file to your repository:

1. Create a new file named `lesson1.md` and add some content.
2. Stage the file for commit:

```
git add lesson1.md
```

3. Commit the file with a message:

```
git commit -m "Add Lesson 1 materials"
```

4. Push the changes to GitHub:

```
git push origin main
```

Your new file is now part of the repository on GitHub.

14 Exploring GitHub's Web Interface

GitHub's web interface provides various features to manage your repository:

- **Code:** View and manage your files.
- **Issues:** Track tasks, enhancements, and bugs.
- **Pull Requests:** Propose changes and collaborate with others.
- **Actions:** Automate workflows.
- **Projects:** Organize and prioritize your work.

For a comprehensive overview, visit: [Hello World - GitHub Docs.](#)([docs.github.com][9])

15 Summary

In this chapter, we've covered the essentials of Git and GitHub, focusing on their applications in educational settings. By setting up Git, creating a GitHub account, and performing basic operations, you're now equipped to manage and collaborate on teaching materials effectively.

Note: For visual learners, GitHub provides illustrative guides and tutorials to further enhance understanding. It's recommended to explore these resources to reinforce the concepts covered in this chapter.

16 Summary

In summary, this book has no content whatsoever.

References

GitHub. 2024. “Quickstart for Repositories - GitHub Docs.” <https://docs.github.com/en/repositories/creating-and-managing-repositories/quickstart-for-repositories>.