

# GITHUB

A Guide to Modern Version Control

Compiled and Edited by:  
Sylvanity Dev Team  
<https://sylvanity.eu>

1<sup>st</sup> Edition, July 2025

Copyright © 2025 Sylvanity B.V. All rights reserved.

No part of this publication may be reproduced, distributed, or transmitted in any form or by any means, including photocopying, recording, or other electronic or mechanical methods, without the prior written permission of the Sylvanity B.V., except in the case of brief quotations embodied in critical reviews and certain other noncommercial uses permitted by copyright law.

ISBN: 9798293708055

<https://www.sylvanity.eu/>

# Contents

<b>Preface</b>	<b>vii</b>
<b>I The Foundation</b>	<b>1</b>
<b>1 Introduction to Version Control and GitHub</b>	<b>3</b>
1.1 Git vs. GitHub: Understanding the Distinction . . . . .	3
1.2 What is Version Control? . . . . .	4
1.3 A Brief History of Git . . . . .	7
1.4 More Than Just Code Storage . . . . .	8
1.5 Setting Up Your GitHub Account . . . . .	10
<b>2 Getting Started with Git</b>	<b>13</b>
2.1 Installing Git on Your Machine . . . . .	13
2.2 First-Time Git Configuration . . . . .	15
2.3 Your First Local Repository . . . . .	16
2.4 Understanding the .gitignore File . . . . .	18
2.5 Viewing Your Project’s History . . . . .	19
2.6 Undoing Mistakes and Recovering History . . . . .	22
<b>II Core GitHub Workflows</b>	<b>25</b>
<b>3 Working with Remote Repositories</b>	<b>27</b>
3.1 Creating Your First GitHub Repository . . . . .	27
3.2 The <code>git remote</code> Command . . . . .	28
3.3 Pushing Changes to GitHub . . . . .	29
3.4 Cloning Existing Repositories . . . . .	30

3.5	Working with Multiple Remotes . . . . .	31
3.6	Keeping Your Local Repository in Sync . . . . .	32
3.7	Remote-Tracking Branches . . . . .	32
3.8	Advanced Remote Management . . . . .	33
<b>4</b>	<b>The Heart of Collaboration: Branching and Merging</b>	<b>37</b>
4.1	What are Branches? . . . . .	37
4.2	Essential Branching Commands . . . . .	38
4.3	Merging Branches . . . . .	39
4.4	Resolving Merge Conflicts . . . . .	41
4.5	When to Rebase Instead of Merge . . . . .	42
4.6	Branch Naming Conventions . . . . .	42
4.7	Short-Lived Branches and Trunk-Based Development . . . .	42
4.8	Cherry-Picking Commits Across Branches . . . . .	43
4.9	Visualising Branches . . . . .	43
<b>III</b>	<b>Advanced GitHub and Collaboration</b>	<b>45</b>
<b>5</b>	<b>The Power of Pull Requests</b>	<b>47</b>
5.1	What is a Pull Request? . . . . .	47
5.2	Creating Your First Pull Request . . . . .	48
5.3	Reviewing Pull Requests . . . . .	49
5.4	Merging a Pull Request . . . . .	50
5.5	Using Pull Request Templates . . . . .	51
<b>6</b>	<b>Effective Team Collaboration</b>	<b>53</b>
6.1	Organizations and Teams . . . . .	53
6.2	Using GitHub Issues for Project Management . . . . .	54
6.3	GitHub Projects: Kanban-Style Boards . . . . .	55
6.4	Writing a Great README . . . . .	56
<b>7</b>	<b>Advanced Git Techniques</b>	<b>58</b>
7.1	Interactive Rebase: Rewriting History . . . . .	58
7.2	Stashing Changes . . . . .	59
7.3	Cherry-Picking Commits . . . . .	60
7.4	Git Tags for Releases . . . . .	61
7.5	Managing Large Files and Huge Repositories . . . . .	62
7.6	Hunting Down Bugs with <code>git bisect</code> . . . . .	65

<b>IV</b>	<b>Automation and Beyond</b>	<b>69</b>
<b>8</b>	<b>Automating Your Workflow with GitHub Actions</b>	<b>71</b>
8.1	Introduction to GitHub Actions . . . . .	71
8.2	Continuous Integration . . . . .	72
8.3	Using Actions from the Marketplace . . . . .	74
8.4	Continuous Deployment (CD) . . . . .	74
8.5	Matrix Builds for Multi-Platform Confidence . . . . .	75
8.6	End-to-End Build . . . . .	77
<b>9</b>	<b>The Broader GitHub Ecosystem</b>	<b>80</b>
9.1	GitHub Pages: Hosting Your Static Site . . . . .	80
9.2	GitHub Gists: Sharing Snippets . . . . .	81
9.3	Exploring GitHub: Discovering Projects and People . . . .	82
9.4	GitHub Codespaces . . . . .	83
9.5	Keeping Dependencies Secure with Dependabot . . . . .	85
9.6	Choosing a License for Your Project . . . . .	87
9.7	GitHub Packages and the Container Registry . . . . .	90
9.8	Automated Security with Code and Secret Scanning . . . .	92
9.9	The GitHub CLI: Bringing GitHub to Your Terminal . . . .	94
9.10	Extending GitHub with the REST and GraphQL APIs . . . .	96
<b>V</b>	<b>Conclusion</b>	<b>101</b>
<b>10</b>	<b>Your Journey with GitHub Continues</b>	<b>103</b>
<b>A</b>	<b>Git Command Quick Reference</b>	<b>105</b>
A.1	Setup and Configuration . . . . .	105
A.2	Creating and Cloning Repositories . . . . .	106
A.3	Basic Workflow . . . . .	107
A.4	Branching and Merging . . . . .	107
A.5	Working with Remotes . . . . .	108
A.6	Advanced Commands . . . . .	109
A.7	Undoing and Recovery . . . . .	110
A.8	Stashing and Shelving . . . . .	111
A.9	Remote Management . . . . .	112
A.10	Large Repositories . . . . .	112
A.11	Tagging and Releases . . . . .	113

A.12 Aliases and Shortcuts . . . . .	113
A.13 GitHub CLI Commands . . . . .	114
<b>B Glossary</b>	<b>115</b>

# Preface

In the rapidly evolving landscape of technology, some tools become so foundational that they transition from being a specialized skill to a universal necessity. Version control, and specifically the ecosystem built around Git and GitHub, has firmly cemented its place in this category. For programmers and software developers, proficiency in these tools is no longer a "nice-to-have" but an absolute prerequisite for professional work. It is the bedrock of modern software development, enabling collaboration, ensuring quality, and providing a safety net for innovation. This book is dedicated to mastering this essential skill, guiding you from the fundamental concepts to advanced workflows.

The advent of artificial intelligence, particularly large language models and generative AI, has not diminished the importance of version control; it has amplified it. We are entering an era of hybrid creativity, where humans and AI agents collaborate on complex projects. Whether it's a programmer working alongside an AI coding assistant that suggests and writes code, or a writer using a text-based chatbot to draft a document, the nature of collaboration is changing.

In this new paradigm, the ability to meticulously track changes becomes more critical than ever. When an AI agent modifies a codebase, a manuscript, or any set of files, the human collaborator must be able to see precisely what has been altered. They need the power to review, accept, or reject these changes with confidence. Version control systems like Git provide the perfect mechanism for this human-AI interaction. They offer a transparent, auditable history of every change, regardless of whether it was made by a human or a machine. This makes Git and GitHub indispensable not just for programmers, but for anyone who collaborates with AI on text-based projects.

Throughout this book, you may encounter terms that seem unfamiliar

at first. We have included a comprehensive glossary at the end of this book to serve as a quick reference. If a term is introduced that you don't immediately recognize, it is likely defined in the glossary and will be discussed in greater detail in a later chapter.

We have strived to create a comprehensive and error-free guide to help you navigate this world. However, the nature of writing, much like software development, is iterative. Should you find any errors, or if you have feedback or suggestions for improvement, we would be grateful to hear from you. Please do not hesitate to reach out to us at [info@sylvanity.eu](mailto:info@sylvanity.eu). Your input is invaluable as we continue to refine this work.