

# Introduction to Git and GitHub

by Google

## About this Course

In this course, you'll learn how to keep track of the different versions of your code and configuration files using a popular version control system (VCS) called Git. We'll also go through how to setup an account with a service called GitHub so that you can create your very own remote repositories to store your code and configuration.

Throughout this course, you'll learn about Git's core functionality so you can understand how and why it's used in organizations. We'll look into both basic and more advanced features, like branches and merging. We'll demonstrate how having a working knowledge of a VCS like Git can be a lifesaver in emergency situations or when debugging. And then we'll explore how to use a VCS to work with others through remote repositories, like the ones provided by GitHub.

By the end of this course, you'll be able to store your code's history in Git and collaborate with others in GitHub, where you'll also start creating your own portfolio!

In order to follow along and complete the assessments, you'll need a computer where you can install Git or ask your administrator to install it for you.

**Show less**

<b>Basic Info</b>	Course 3 of 6 in the <a href="#">Google IT Automation with Python Specialization</a>
<b>Level</b>	Beginner
<b>Language</b>	English, <b>Subtitles:</b> Arabic, French, Portuguese (European), Italian, Vietnamese, German, Russian, Spanish <a href="#">Volunteer to translate subtitles for this course</a>
<b>How To Pass</b>	Pass all graded assignments to complete the course.
<b>User Ratings</b>	4.8 stars

## Syllabus

---

### WEEK 1

## Introduction to Version Control

In this module, you'll be introduced to the concept of version control, which will make managing and rolling back your code look super easy. You'll learn how to differentiate between files and the tools at your disposal to make this happen. Next, you'll be introduced to Git and how you can leverage that platform to improve your coding abilities. Once you've got a grasp on what Git is, you'll install it and start using it to create and clone code repositories. Last up, you'll deep dive into Git in order to get more familiar with the different tools and commands it has to offer.

### Show less




16 videos, 6 readings, 3 practice quizzes

1. **Video:** Course Introduction
2. **Reading:** Welcome to the Course
3. **Discussion Prompt:** Meet & Greet
4. **Video:** Intro to Module 1: Version Control
5. **Video:** Keeping Historical Copies
6. **Video:** Diffing Files
7. **Video:** Applying Changes
8. **Video:** Practical Application of diff and patch
9. **Practice Quiz:** Practice Quiz: Before Version Control
10. **Reading:** diff and patch Cheat Sheet
11. **Video:** What is version control?
12. **Video:** Version Control and Automation
13. **Video:** What is Git?
14. **Reading:** More Information About Git
15. **Video:** Installing Git
16. **Video:** Installing Git on Windows (Optional)
17. **Reading:** Installing Git
18. **Practice Quiz:** Practice Quiz: Version Control Systems
19. **Video:** First Steps with Git
20. **Video:** Tracking Files
21. **Video:** The Basic Git Workflow
22. **Video:** Anatomy of a Commit Message
23. **Reading:** Initial Git Cheat Sheet
24. **Practice Quiz:** Practice Quiz: Using Git

25. **Video:** Module 1 Wrap Up: Intro to Version Control
26. **Reading:** How to Log in to Qwiklabs
27. **Discussion Prompt:** Discussion Prompt

[Show less](#)

 **Graded:** Qwiklabs Assessment: Introduction to Git

## WEEK 2

### Using Git Locally

In this module, you'll dive into advanced Git interactions by skipping the staging area for small code changes, and understand how Git uses the HEAD alias to represent checked-out snapshots. Next, you'll explore how to move and remove files before finally getting a cheat sheet to help you in your Git explorations! The next step of your learning will include how to undo changes before committing and how to amend commits once they're submitted. Finally, you'll be able to identify errors in commits that were submitted a while back. In the final section of this module, you'll explore the concept of branching and merging. You'll learn what a branch is, how to create one, and how they work in harmony with you and your code. Once you're comfortable with branching, you'll dive into merging, how it works with branched data, and how to deal with merge conflicts.


**Show less**

 14 videos, 3 readings, 3 practice quizzes

1. **Video:** Intro to Module 2: Using Git Locally
2. **Discussion Prompt:** Discussion Prompt
3. **Video:** Skipping the Staging Area
4. **Video:** Getting More Information About Our Changes
5. **Video:** Deleting and Renaming Files
6. **Reading:** Advanced Git Cheat Sheet
7. **Practice Quiz:** Practice Quiz: Advanced Git Interaction
8. **Video:** Undoing Changes Before Committing
9. **Video:** Amending Commits
10. **Video:** Rollbacks
11. **Video:** Identifying a Commit
12. **Reading:** Git Revert Cheat Sheet
13. **Practice Quiz:** Practice Quiz: Undoing Things
14. **Video:** What is a branch?

15. **Video:** Creating New Branches
16. **Video:** Working with Branches
17. **Video:** Merging
18. **Video:** Merge Conflicts
19. **Reading:** Git Branches and Merging Cheat Sheet
20. **Practice Quiz:** Practice Quiz: Branching & Merging
21. **Video:** Module 2 Wrap Up: Using Git Locally
22. **Discussion Prompt:** Discussion Prompt

[Show less](#)

 **Graded:** Qwiklabs Assessment: Merging Branches in Git

## WEEK 3

### Working with Remotes

In this module, you'll be introduced to GitHub and learn how it works with Git. You'll create new repositories and clone those repositories onto your computer. Next, we'll explain what a remote repository is, how we can work with them, and how we can host them. You'll get familiar with commands like modify, stage, and commit, which will be used for local changes, as well as the fetch command, which can pull any changes from remote repositories. Our final lesson will focus on learning about conflicts. This will allow you to explore the concepts of pull-merge-push workflows, pushing remote branches and rebasing your changes.


**Show less**

 13 videos, 3 readings, 3 practice quizzes

1. **Video:** Intro to Module 3: Working with Remotes
2. **Discussion Prompt:** Discussion Prompt
3. **Video:** What is GitHub?
4. **Video:** Basic Interaction with GitHub
5. **Reading:** Basic Interaction with GitHub Cheat-Sheet
6. **Practice Quiz:** Practice Quiz: Introduction to GitHub
7. **Video:** What is a remote?
8. **Video:** Working with Remotes
9. **Video:** Fetching New Changes
10. **Video:** Updating the Local Repository

11. **Reading:** Git Remotes Cheat-Sheet
12. **Practice Quiz:** Practice Quiz: Using a Remote Repository
13. **Video:** The Pull-Merge-Push Workflow
14. **Video:** Pushing Remote Branches
15. **Video:** Rebasing Your Changes
16. **Video:** Another Rebasing Example
17. **Video:** Best Practices for Collaboration
18. **Reading:** Conflict Resolution Cheat Sheet
19. **Practice Quiz:** Practice Quiz: Solving Conflicts
20. **Video:** Module 3 Wrap Up: Working with Remotes
21. **Discussion Prompt:** Discussion Prompt

[Show less](#)

 **Graded:** Qwiklabs Assessment: Introduction to Github

## WEEK 4

### Collaboration

In this module, you'll continue to explore the collaboration tools available in Git. You'll learn about the tools that are available to help improve the quality of your code and to better track your code. This includes an overview of pull requests and how the typical workflow of a pull request looks like on GitHub. Next, you'll dive into how you can squash changes in your code. We'll finish up by providing you with a cheat sheet on fork and pull requests. Next up, we'll cover what code reviews are and what the code review workflow looks like. Then, you'll learn about how to use code reviews on GitHub. The final lesson of this module will focus on managing projects. We'll take a rundown of best practices on managing projects and how to manage collaboration within those projects. We'll explore different ways of tracking issues and finish up by discussing the concept of continuous integration with your projects.


**Show less**

 14 videos, 3 readings, 3 practice quizzes

1. **Video:** Intro to Module 4: Collaboration
2. **Video:** A Simple Pull Request on GitHub
3. **Video:** The Typical Pull Request Workflow on GitHub
4. **Video:** Updating an Existing Pull Request
5. **Video:** Squashing Changes
6. **Reading:** Git Fork and Pull Request Cheat Sheet

7. **Practice Quiz:** Practice Quiz: Pull Requests
8. **Video:** What are code reviews?
9. **Video:** The Code Review Workflow
10. **Video:** How to Use Code Reviews in GitHub
11. **Reading:** More Information on Code Reviews
12. **Practice Quiz:** Practice Quiz: Code Reviews
13. **Video:** Managing Collaboration
14. **Video:** Tracking Issues
15. **Video:** Continuous Integration
16. **Reading:** Additional Tools
17. **Practice Quiz:** Practice Quiz: Managing Collaboration
18. **Video:** Module 4 Wrap Up: Collaboration
19. **Video:** Congratulations!
20. **Discussion Prompt:** Your Learner Journey
21. **Video:** Sneak Peek of the Next Course

[Show less](#)

 **Graded:** Qwiklabs Assessment: Pushing Local Commits to Github

[View Less](#)

## How It Works

### General

#### How do I pass the course?

To earn your Course Certificate, you'll need to earn a passing grade on each of the required assignments—these can be quizzes, peer-graded assignments, or programming assignments. Videos, readings, and practice exercises are there to help you prepare for the graded assignments.

#### What do start dates and end dates mean?

Once you enroll, you'll have access to all videos, readings, quizzes, and programming assignments (if applicable). If you choose to explore the course without purchasing, you may not be able to access certain assignments. If you don't finish all graded assignments before the end of the course, you can reset your deadlines. Your progress will be saved and you'll be able to pick up where you left off.

What are due dates? Is there a penalty for submitting my work after a due date?

Within a course, there are suggested due dates to help you manage your schedule and keep coursework from piling up. Quizzes and programming assignments can be submitted late without consequence. However, it is possible that you won't receive a grade if you submit your peer-graded assignment too late because classmates usually review assignment within three days of the assignment deadline.

Can I re-attempt an assignment?

Yes. If you want to improve your grade, you can always try again. If you're re-attempting a peer-graded assignment, re-submit your work as soon as you can to make sure there's enough time for your classmates to review your work. In some cases you may need to wait before re-submitting a programming assignment or quiz. We encourage you to review course material during this delay.

**Show less**