Introduction

Welcome to Version Control!

Version control is critical to maintaining software and enabling scalability solutions. Any programming project that requires multiple files can benefit from this best practice. Version control enables teams to have collaborative workflows while also enhancing the software development life cycle. This course introduces the basics of publishing, retrieving, branching, and cloning.

This course focuses on the version control system Git. Git can be used to create and manage repositories both locally and remotely. In the learning material for this course, they will use a popular platform for remote repository hosting called GitHub. In the task you complete for your performance assessment you will be asked to connect to a remote repository called GitLab. All of the commands and work that you study within our learning resource will work the same with both GitHub and GitLab as they both use Git as their underlying technology.

Similarly, there are many ways to initiate a Git command. In this course, you will primarily be using a command line interface (CLI). We recommend using GitBash for Windows and Terminal for MacOS. You will also be shown how an integrated development environment (IDE) may be used with a built in CLI. Most modern IDEs have built in version control integration. You will likely work with a variety of IDEs during your time as a student and in your career. We recommend that you familiarize yourself with these features any time you begin working with a new IDE.

Prerequisites

This course has no prerequisites.

Course Competencies

This course guide will help you to demonstrate the following competency:

Competency 4030.2.1: Implements Version Control

The learner implements version control processes and solutions that maintains source code.

‹ Previous Next ›

Learning Resources

Version Control

Automatically Enrolled Resources

You can access the learning resources in this course by clicking on the links provided throughout the course. You may be prompted to log in to the WGU student portal to access the resources.

Learning Resources

The following learning resources are essential to your learning throughout this course.

Git Essential Training (2023) by LinkedIn Learning

Git Intermediate Techniques (2018) – "Tagging" by LinkedIn Learning

Microsoft Office 365 for WGU Students to access your free student Microsoft 365 account.

Visual Studio Code to download Visual Studio Code, which will allow you to follow along with the activities in the LinkedIn videos.

Supplemental Learning Resources

The following supplemental resources provide additional information and examples to further extend your knowledge about Git.

Additional Resources for D197 includes information provided to you by the instructors, including templates and webinars.

Getting Started with Visual Studio Code and Building HTML Websites – Code Academy

Git Tutorial – W3Schools

Introduction to Git for GitLab projects – WGU Udemy resource

Course Organization and Navigation

This course is organized into multiple sections: Git Introduction, Using Git, Installing Git, Using the Remote Repository, Working with Files, Important Git Concepts, and Tagging Files. Each section contains multiple lessons and a section quiz.

You can expect to spend approximately 40 hours completing this course.

Follow this suggested path to help you complete the course in the recommended time frame.

Steps	Steps Toward Completing the Performance Assessment	Completion Time (from start to finish)
1	Review all the lessons within the course material.	1 week
2	Watch the webinars under "Important Webinars" in the D197 Additional Resources.	3 days
3	Set up a repository on GitLab.	1 day
4	Review "Important Documents" in the D197 Additional Resources.	1 day
5	Complete the requirements for the performance assessment.	1 week
6	Review the "Common Reasons Submissions Don't Pass"	N/A

Version Control

7	Make an appointment with your instructor to review your performance assessment before your first submission.	N/A
8	Submit your performance assessment.	Total: 2–3 weeks

Previous Next

2.1 Introduction

This section will explore the foundational aspects of Git, the distributed version control system that has become an industry standard for software development. Git provides robust features that allow multiple developers to work together, providing options for branching, merging, and versioning code. Understanding Git is essential for individual developers and teams aiming for a streamlined and efficient development process. You will cover key Git commands, the concept of repositories, and the importance of branching strategies. By the end of this section, you will have a solid understanding of Git's role in version control and be ready to use it in your own projects.

‹ Previous

Next ›

Lesson 2.2: Git Basics

This lesson is about getting you comfortable with the basic functionality of Git, right from the command line. You do not need previous experience with Git or any particular programming language; you will primarily work with simple text and MD files. The goal is to give you the necessary Git commands so you can start managing your projects more efficiently.

Watch

"Get started with Git" (0:38) by LinkedIn Learning

"What you should know" (0:59) by LinkedIn Learning

Attributions

Forbes, B. (2023). *Git Essential Training*. LinkedIn Learning. https://www.linkedin.com/learning/git-essential-training-19417064/get-started-with-git?u=2045532

Previous Next

Lesson2.3: Why Do You Want to Use Git

This lesson will explore the why behind using Git, focusing on its practical applications rather than the technical details. You will start by learning how Git acts as a powerful tool for version control, allowing you to maintain a clean and organized code base. Next, you will explore how Git simplifies code sharing and collaboration, eliminating email attachments and multiple cluttered file definitions. Finally, this lesson will discuss the important role Git plays in the open-source community, where it enables multiple users to contribute to a project while maintaining code integrity. By the end of this lesson, you will have a broader understanding of why Git is not only useful but essential in modern software development.

Watch

"Git for version Control" (3:33) by LinkedIn Learning

"Git to share code" (1:57) by LinkedIn Learning

"Git to collaborate" (1:32) by LinkedIn Learning

"Open source" (3:00) by LinkedIn Learning

Quiz

Complete the LinkedIn Learning "Chapter Quiz."

Attributions

Forbes, B. (2023). *Git Essential Training*. LinkedIn Learning. https://www.linkedin.com/learning/git-essential-training-19417064/get-started-with-git?u=2045532

‹ Previous Next ›

3.1 Introduction

Think of Git as your trusted partner on your coding adventures. Git is a version control system that allows you to manage your source code history effectively. It helps you save various snapshots of your project, so you can backtrack if you need to make revisions. With Git, you can also collaborate seamlessly with other developers. Gone are the days of emailing code files back and forth or losing hours of work because you forgot to save. Git provides a robust framework for tracking changes, reverting to previous versions, and collaborating with others.

‹ Previous Next ›

Version Control

Lesson3.2: How Does Git Work?

Git operates on a distributed version control model, setting it apart from centralized systems. When you initialize a Git repository, you create a local repository complete with a history of commits. As you make changes to your code, these can be staged for a commit, which will then be added to your local repository's history. With commands such as git add and git commit, you can create snapshots of your project at any given time. These snapshots can be shared with other developers or even pushed to a remote repository for collaborative work. The beauty of Git lies in its flexibility and depth, empowering you to manage your projects effectively.

Watch

"Use Git locally" (1:51) by LinkedIn Learning

"Use a Git provider" (1:04) by LinkedIn Learning

"Distributed version control" (1:36) by LinkedIn Learning

"How to start working with Git" (1:11) by LinkedIn Learning

Quiz

Complete the LinkedIn Learning "Chapter Quiz."

Attributions

Forbes, B. (2023). *Git Essential Training*. LinkedIn Learning. https://www.linkedin.com/learning/git-essential-training-19417064/get-started-with-git?u=2045532

Previous Next

4.1 Introduction

Git is a version control system that has become an industry standard for developers. Whether you are working solo or in a team, Git allows you to keep track of changes, collaborate more effectively, and resolve conflicts in your code. With support for multiple platforms like Linux, Windows, and macOS, Git offers versatility. It can be used straight from the command line, or via various graphical clients. Understanding Git is crucial for versioning your projects and for contributing to collaborative endeavors in the software development world.

‹ Previous Next ›

Lesson4.2: Install and Configure Git

To commence your journey with Git, visit the official website, git-scm.com, and download the installer that matches your operating system. The installation process is relatively straightforward, and while Git includes options to customize, the default settings are adequate for beginners. After installation, you will need to set up your username and email in Git. This is done using git config commands in the terminal. Once installation and setup are complete, you are ready to start using Git.

Watch

"Install Git on Windows" (4:39) by LinkedIn Learning

"Install Git on Linux" (1:09) by LinkedIn Learning

"Install Git on MacOS" (1:18) by LinkedIn Learning

"Git GUI clients" (1:47) by LinkedIn Learning

"Optional: Install Visual Studio Code" (2:21) by LinkedIn Learning

"Configure Git" (2:44) by LinkedIn Learning

Quiz

Complete the LinkedIn Learning "Chapter Quiz."

Attributions

Forbes, B. (2023). *Git Essential Training*. LinkedIn Learning. https://www.linkedin.com/learning/git-essential-training-19417064/get-started-with-git?u=2045532

Lesson4.2: Install and Configure Git

To commence your journey with Git, visit the official website, git-scm.com, and download the installer that matches your operating system. The installation process is relatively straightforward, and while Git includes options to customize, the default settings are adequate for beginners. After installation, you will need to set up your username and email in Git. This is done using git config commands in the terminal. Once installation and setup are complete, you are ready to start using Git.

Watch

"Install Git on Windows" (4:39) by LinkedIn Learning

"Install Git on Linux" (1:09) by LinkedIn Learning

"Install Git on MacOS" (1:18) by LinkedIn Learning

"Git GUI clients" (1:47) by LinkedIn Learning

"Optional: Install Visual Studio Code" (2:21) by LinkedIn Learning

"Configure Git" (2:44) by LinkedIn Learning

Quiz

Complete the LinkedIn Learning "Chapter Quiz."

Attributions

Forbes, B. (2023). *Git Essential Training*. LinkedIn Learning. https://www.linkedin.com/learning/git-essential-training-19417064/get-started-with-git?u=2045532

‹ Previous Next ›

5.1 Introduction

Using a remote repository is an essential practice in modern software development, enabling efficient collaboration and code management. Whether you are interacting with GitHub, GitLab, or Bitbucket, the principle remains consistent: you have a centralized, cloud-based storage space for your code. This lesson assists you in establishing a remote repository and pushing your local code to it for shared access.

‹ Previous Next ›

Lesson5.2: Push Your Code with Git

Once you have made changes to your local code base and are ready to share them, the next step involves pushing the code to a remote repository. To accomplish this, begin by staging your changes using git add; then use git commit to capture a snapshot of your modifications. Finally, execute git push to update the remote repository with your local changes. If you are pushing for the first time, you may be prompted for your log-in credentials. A successful push confirms that your local code has been securely transferred to the remote repository.

Watch

"Set up a remote repository" (2:30) by LinkedIn Learning

"Clone the remote repository" (1:21) by LinkedIn Learning

"Create a file and stage it" (1:27) by LinkedIn Learning

"Commit a file" (1:08) by LinkedIn Learning

"Push the file to the remote repository" (0:57) by LinkedIn Learning

"The git folder" (1:37) by LinkedIn Learning

"Initialize a repository locally and sync it to the remote repository" (2:33) by LinkedIn Learning

"Challenge: Push your first code" (0:41) by LinkedIn Learning

"Solution: Push your first code" (0:40) by LinkedIn Learning

Quiz

Complete the LinkedIn Learning "Chapter Quiz."

Attributions

Forbes, B. (2023). *Git Essential Training*. LinkedIn Learning. https://www.linkedin.com/learning/git-essential-training-19417064/get-started-with-git?u=2045532

‹ Previous Next ›

6.1 Introduction

Git is a powerful tool designed to help software developers manage and track changes in their codebase. It offers a range of commands and functionalities that provide insights into the status of a repository, allowing users to interact with files, view changes, and maintain a clean history. One of the fundamental commands in Git is "git status," which provides a snapshot of the current state of the repository, indicating any changes made, files added or deleted, and suggestions on potential next steps.

‹ Previous Next ›

Version Control

Lesson6.2: Make Changesto Files

When working with Git, making changes to files is a common task. Whether you are adding new files, editing existing ones, or deleting them, Git provides a great way to track these modifications. After creating or editing a file, use "git status" to show the file's current state, such as whether it is untracked or staged for commit. To commit a new file or change to an existing one, you can use the "git add" command followed by "git commit." If you are unsure about the changes made, "git diff" offers a detailed view of modifications across files. Additionally, tools such as Visual Studio Code integrate seamlessly with Git, offering a graphical user interface to visualize and manage changes, making the process even more intuitive.

Watch

"Git status" (2:17) by LinkedIn Learning

"Edit a file and view changes" (2:48) by LinkedIn Learning

"Make use of the GUI of Visual Studio Code" (2:45) by LinkedIn Learning

"View commit history" (3:28) by LinkedIn Learning

"Delete files" (1:28) by LinkedIn Learning

"Rename files" (2:04) by LinkedIn Learning

"Working with folders" (2:34) by LinkedIn Learning

"Undo your changes" (2:28) by LinkedIn Learning

"Look back in Git history" (1:53) by LinkedIn Learning

"Revert to an old state" (2:06) by LinkedIn Learning

"Challenge: Make a change and revert" (1:04) by LinkedIn Learning

"Solution: Make a change and revert" (2:01) by LinkedIn Learning

Quiz

Complete the LinkedIn Learning "Chapter Quiz."

Attributions

Forbes, B. (2023). *Git Essential Training*. LinkedIn Learning. https://www.linkedin.com/learning/git-essential-training-19417064/get-started-with-git?u=2045532

Previous Next

Version Control

7.1 Introduction

Git is an essential tool for software developers, allowing for efficient version control and collaboration. The ability of Git to track changes, manage branches, and resolve conflicts makes it indispensable in modern software development. This section will delve into some of the fundamental concepts of Git, providing insights into its functionality and best practices.

Previous Next

Version Control

Lesson7.2: Important Concepts in Git

Git offers a plethora of features to streamline coding workflows. One such feature is the .gitignore file, which instructs Git to overlook specific files or directories, such as log files or local settings. This ensures only relevant code changes are tracked. Branching in Git allows software developers to create parallel versions of code, facilitating simultaneous development without affecting the main production code. Pull requests play a pivotal role in merging these branches, ensuring code reviews and quality checks. Commit messages in Git serve as a historical record, with best practices suggesting concise, present-tense descriptions of changes.

Watch

"Ignoring files" (2:27) by LinkedIn Learning

Opens in new tab

"Git branches" (1:59) by LinkedIn Learning

"Git commit messages" (3:46) by LinkedIn Learning

"Getting out of trouble" (3:36) by LinkedIn Learning

More on resolving merge conflicts from GitHub

More on resolving merge conflicts from GitLab

Please note, although some of the videos in the list demonstrate how to edit, please do not edit the code or files provided to you for this course.

Quiz

Complete the LinkedIn Learning "Chapter Quiz."

Attributions

Forbes, B. (2023). *Git Essential Training*. LinkedIn Learning. https://www.linkedin.com/learning/git-essential-training-19417064/get-started-with-git?u=2045532

‹ Previous Next ›

WGU

ADA Accommodation

8.1 Introduction

Git provides a way to mark important points of a repository's history through a feature known as "tagging." Tagging stands out as a pivotal tool for software developers seeking to organize and reference their codebase efficiently. While Git is renowned for tracking changes and managing versions, tagging offers a nuanced approach to marking specific moments or milestones within a project. This feature allows software developers to label and recall particular commits, ensuring that significant points in the project's history are easily identifiable and accessible.

‹ Previous Next ›

Lesson 8.2: Tagging Files

Tagging in Git allows users to mark specific commits with a simple, human-readable label, making it easier to navigate and identify significant milestones or versions in a project. Unlike branches, which can have an evolving history with additional commits, tags are static; they point to a specific commit and remain unchanged. This makes them ideal for marking release points, such as version 1.0 or 2.0.

Watch

"Create tags" (4:10) by LinkedIn Learning

"List tags" (4:30) by LinkedIn Learning

"Delete tags" (1:03) by LinkedIn Learning

"Push tags to a remote" (3:40) by LinkedIn Learning

Opens in new tab

"Check out tags" (5:33) by LinkedIn Learning

Attributions

Skoglund, K. (2018). *Git Intermiediate Techniques: Tagging*. LinkedIn Learning. https://www.linkedin.com/learning/git-intermediate-techniques/create-tags?u=2045532

Previous Next

9.1 Conclusion

As this course concludes, reflect on the journey you have undertaken. Git, with its myriad of functionalities, can be revolutionary for software developers, streamlining workflows and enhancing collaboration. While the topics covered in this course provide a solid foundation, it is worth noting that the world of Git is vast, and there is much more to explore.

Watch

"Next steps" (0:47) by LinkedIn Learning

Attributions

Forbes, B. (2023). *Git Essential Training*. LinkedIn Learning. https://www.linkedin.com/learning/git-essential-training-19417064/get-started-with-git?u=2045532

Previous Next