WEB AGE SOLUTIONS

Powered by Axcel Learning

Lecture Book

WA3662 Introduction to Command Line, Git, and IDEs

Version 1.0.1

Web Age Solutions, LLC
1 California Street Suite 2900
San Francisco, CA 94111
https://www.webagesolutions.com

USA: 1-877-517-6540, email: getinfousa@webagesolutions.com
Canada: 1-877-812-8887 toll free, email: getinfo@webagesolutions.com

WEB AGE SOLUTIONS

Introduction to Command Line

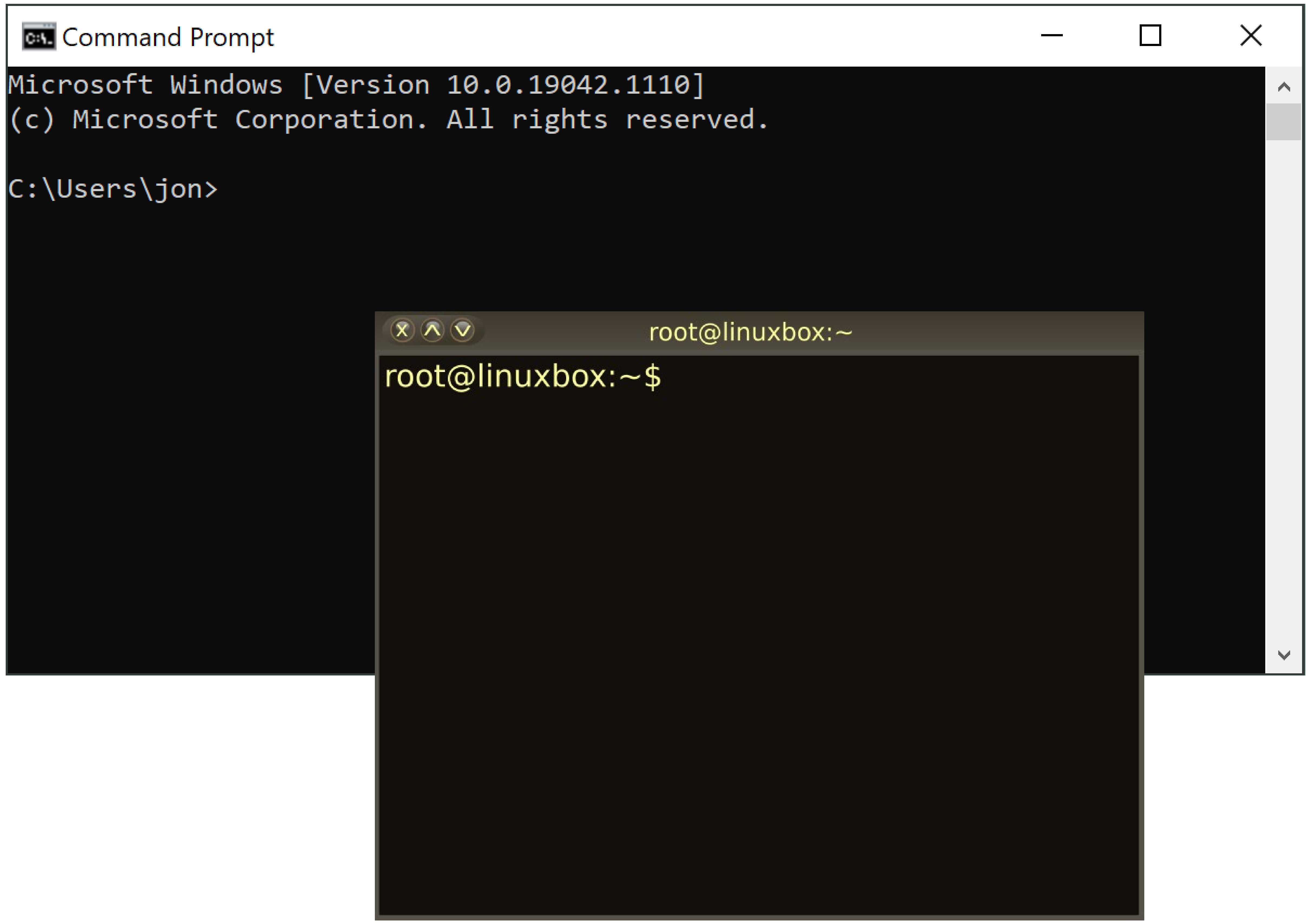
1.1 Command Lines and Terminals

The Command line or Terminal:

- Is a text interface for the computer
- Execute commands, runs scripts, returns text output.
- Uses include File navigation, program execution, system configuration and administration
- Supports reliable and repeatable operations through scripting

Common Types include:

- Command Prompt (Windows),
- Terminal (Linux/macOS),
- PowerShell, Bash, … (cross platform)



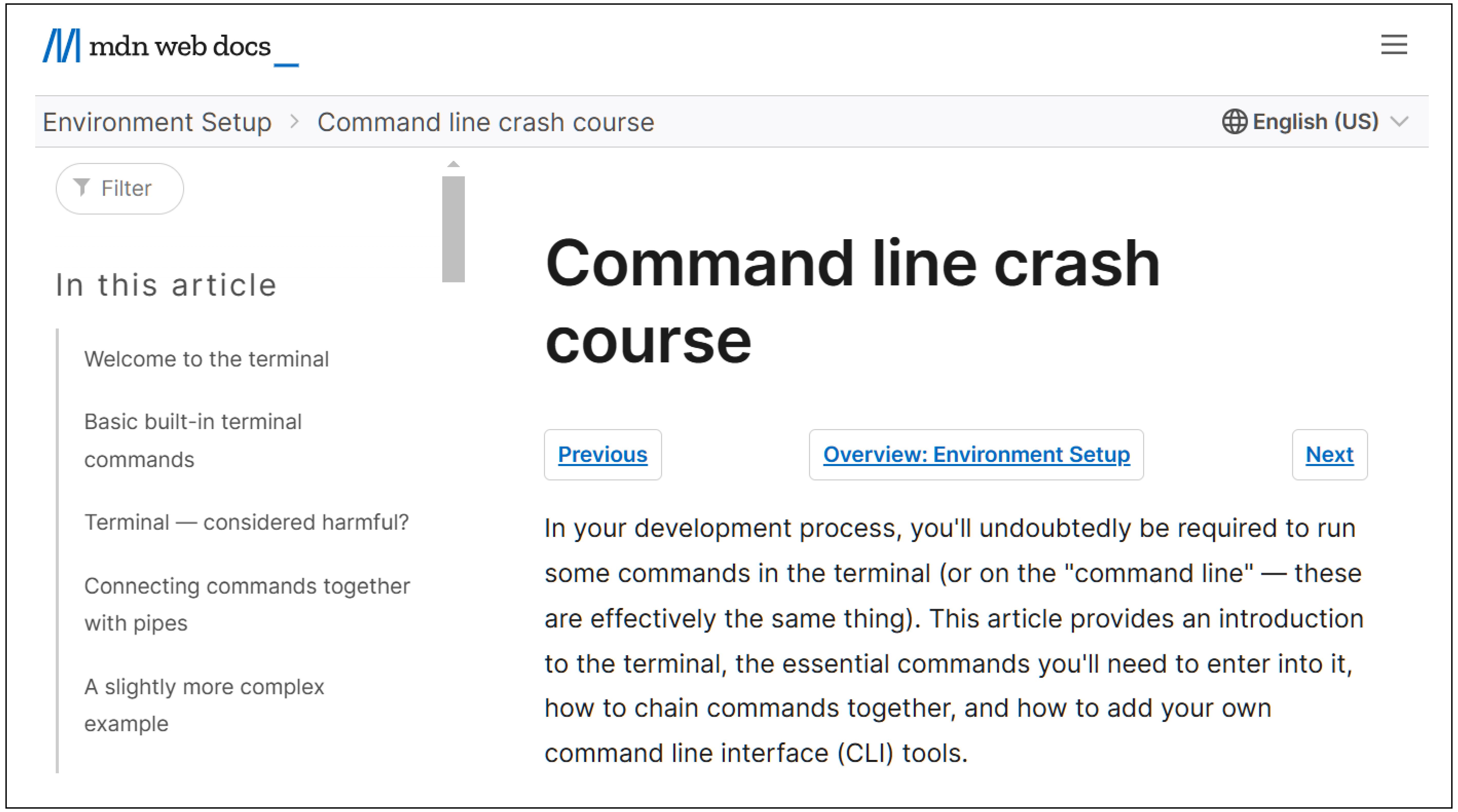
```
Microsoft Windows [Version 10.0.19042.1110]
(c) Microsoft Corporation. All rights reserved.

C:\Users\jon>
```

```
root@linuxbox:~$
```

1.2 Command line crash course

Article

https://developer.mozilla.org/en-US/docs/Learn_web_development/Getting_started/Environment_setup/Command_line


mdn web docs
Environment Setup > Command line crash course
English (US)
Filter
In this article
Welcome to the terminal
Basic built-in terminal commands
Terminal — considered harmful?
Connecting commands together with pipes
A slightly more complex example
Previous
Overview: Environment Setup
Next

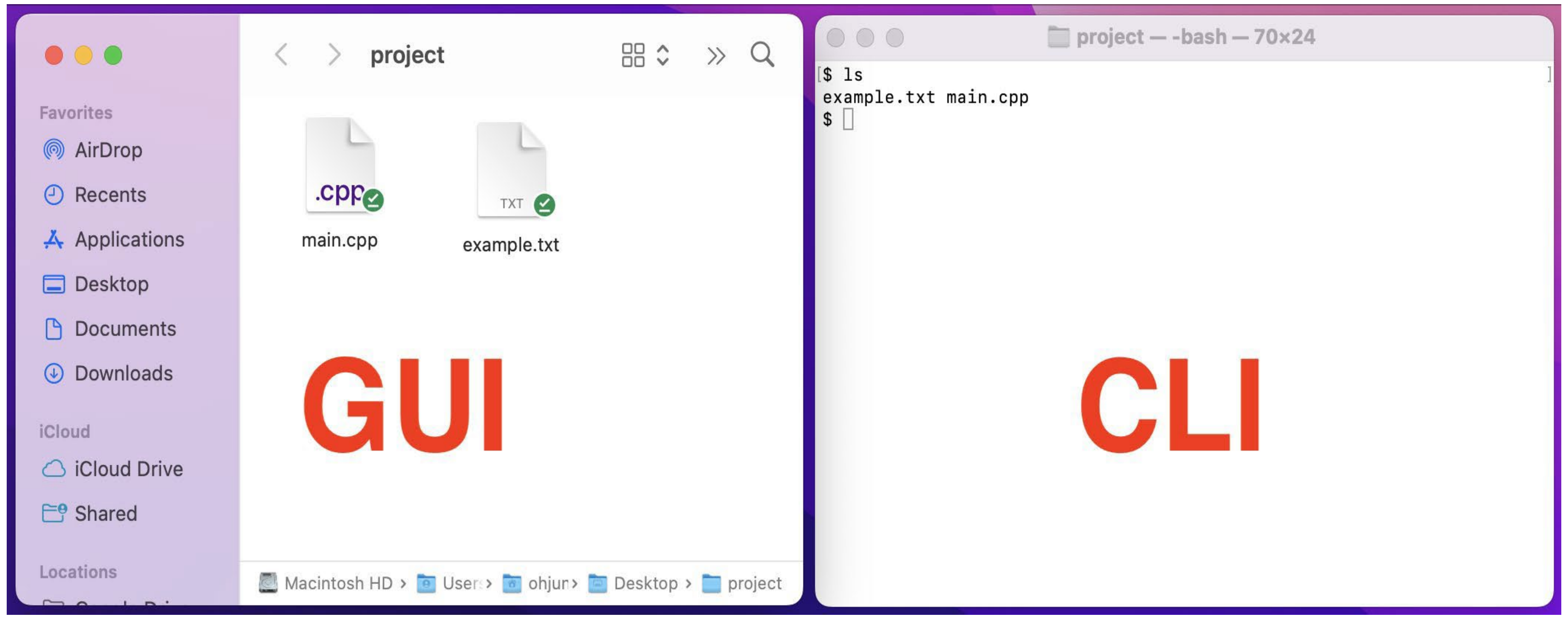
Command line crash course

In your development process, you'll undoubtedly be required to run some commands in the terminal (or on the "command line" — these are effectively the same thing). This article provides an introduction to the terminal, the essential commands you'll need to enter into it, how to chain commands together, and how to add your own command line interface (CLI) tools.

WEB AGE SOLUTIONS

1.3 What is the command line?

- The window, which is usually called the **command line** or **command-line interface**, is a text-based application for viewing, handling, and manipulating files on your computer. It allows you to interact with the computer's operating system.
- The command line works by typing commands against a prompt, which then gets passed to the operating system of the computer that runs these commands.
- It's much **like Windows Explorer** or Finder on the Mac, but **without the graphical interface.**
- Other names for the command line are**: cmd, CLI, prompt, console or terminal**. (Generally, you'll find these terms used interchangeably. **)**

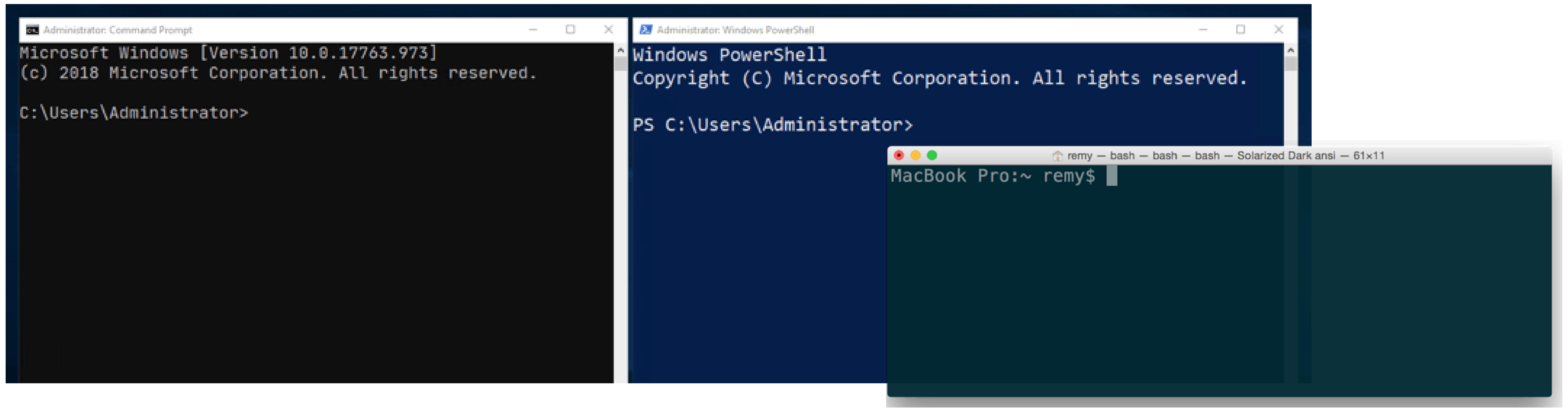


```
$ ls
example.txt main.cpp
$
```

1.4 How to Access the Terminal

How to get access to the terminal **depends on your operating system.**

- **Linux/Unix -** Linux/Unix systems have a terminal available by default, **listed among your Applications**.
- **macOS -** macOS has a system called Darwin that sits underneath the graphical user interface. Darwin is a Unix-like system, which provides the terminal, and access to the low-level tools.
- The terminal is **available on macOS at Applications/Utilities/Terminal**.
- **Windows -** As with some other programming tools, using the terminal (or command line) on Windows has traditionally not been as simple or easy as on other operating systems. But things are getting better.
    - Windows has traditionally had its own terminal-like program called **cmd** ("the command prompt") for a long time. It is equivalent to the old-style Windows DOS prompt.


Administrator: Command Prompt
Administrator: Windows PowerShell
remy — bash — bash — bash — Solarized Dark ansi — 61×11

```
Microsoft Windows [Version 10.0.17763.973]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Users\Administrator>
```

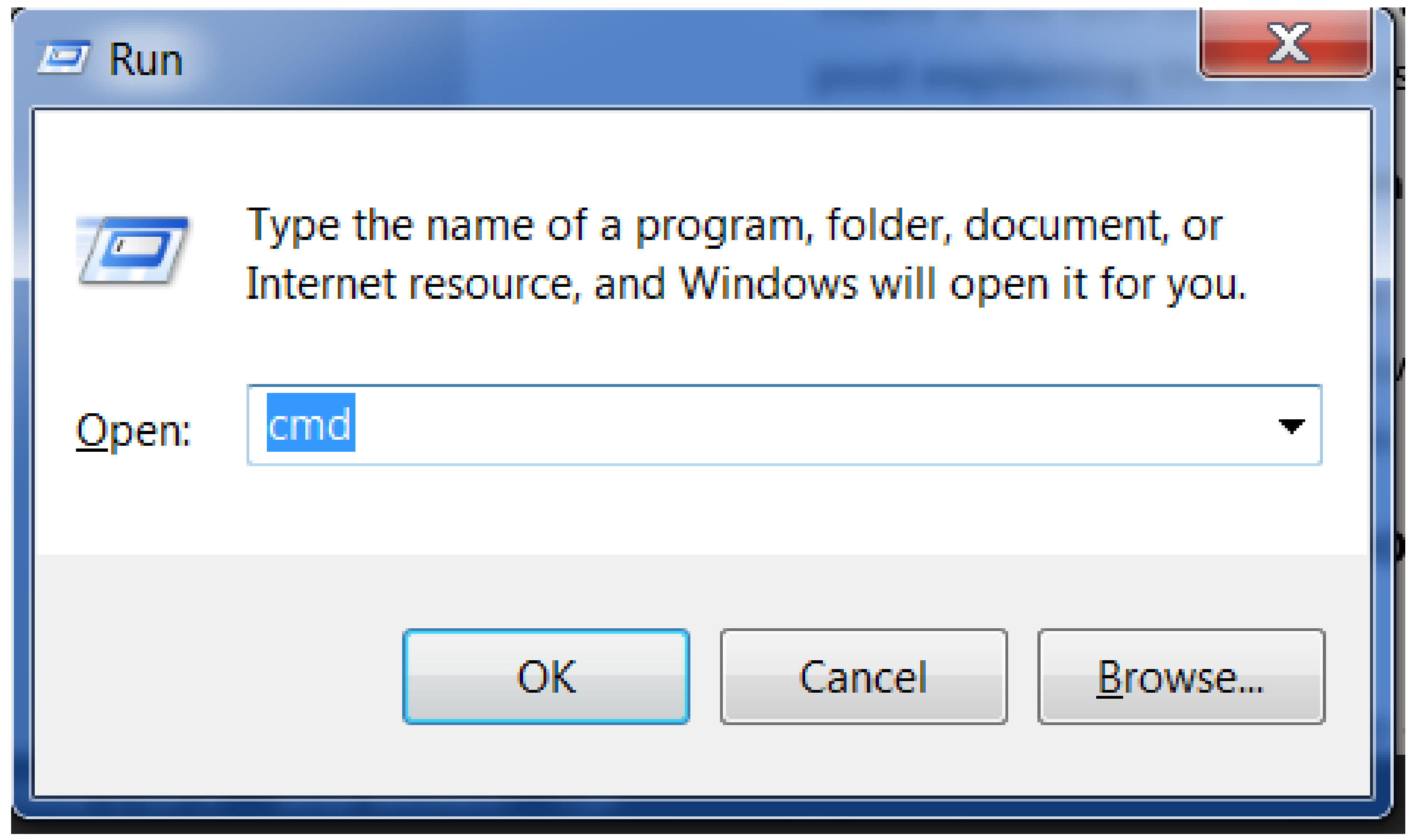
```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

PS C:\Users\Administrator>
```

```
MacBook Pro:~ remy$
```

1.5 Open the command-line interface – Windows

Depending on your version of Windows and your keyboard, you can use one of the following to open a command window :

- Go to the Start menu or screen and enter "Command Prompt" in the search field.
- Go to Start menu → Windows System → Command Prompt.
- Go to Start menu → All Programs → Accessories → Command Prompt.
- Go to the Start screen, hover your mouse in the lower-left corner of the screen, and click the down arrow that appears (on a touch screen, instead flick up from the bottom of the screen). The Apps page should open. Click on Command Prompt in the Windows System section.
- Hold the special Windows key on your keyboard and press the "X" key. Choose "Command Prompt" from the pop-up menu.
- Hold the Windows key and press the "R" key to get a "Run" window. Type "cmd" in the box and click the OK key,


Run

Type the name of a program, folder, document, or Internet resource, and Windows will open it for you.

Open: cmd

1.6 Basic Built-In Terminal Commands

- These are a few of the things you will need to be able to do with the command line.
- Navigate your computer's file system along with base-level tasks such as create, copy, rename, and delete.

Move around your directory structure: **cd**

Create directories: **mkdir**

Copy files or directories: **cp**

Move files or directories: **mv**

Delete files or directories: **rm**

Download files found at specific URLs: **curl**

Search for fragments of text inside larger bodies of text: **grep**

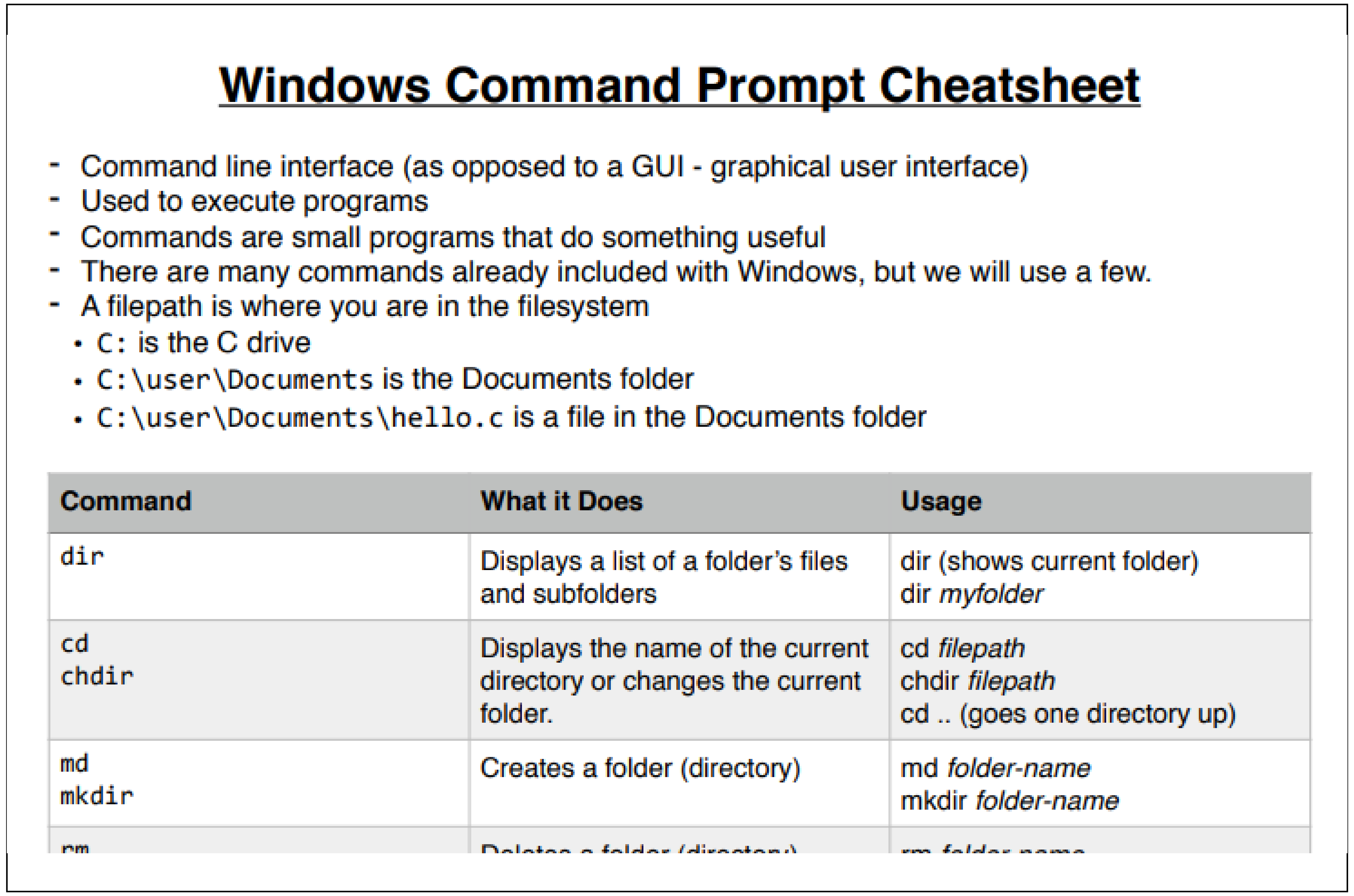
View a file's contents page by page: **less, cat**

Manipulate and transform streams of text (for example changing all the instances of <div>s in an HTML file to <article>): **awk, tr, sed**

1.7 Command Prompt Cheat Sheet

Reference

https://www.cs.columbia.edu/~sedwards/classes/2015/1102-fall/Command Prompt Cheatsheet.pdf



Windows Command Prompt Cheatsheet

- Command line interface (as opposed to a GUI - graphical user interface)
- Used to execute programs
- Commands are small programs that do something useful
- There are many commands already included with Windows, but we will use a few.
- A filepath is where you are in the filesystem
  - `C:` is the C drive
  - `C:\user\Documents` is the Documents folder
  - `C:\user\Documents\hello.c` is a file in the Documents folder

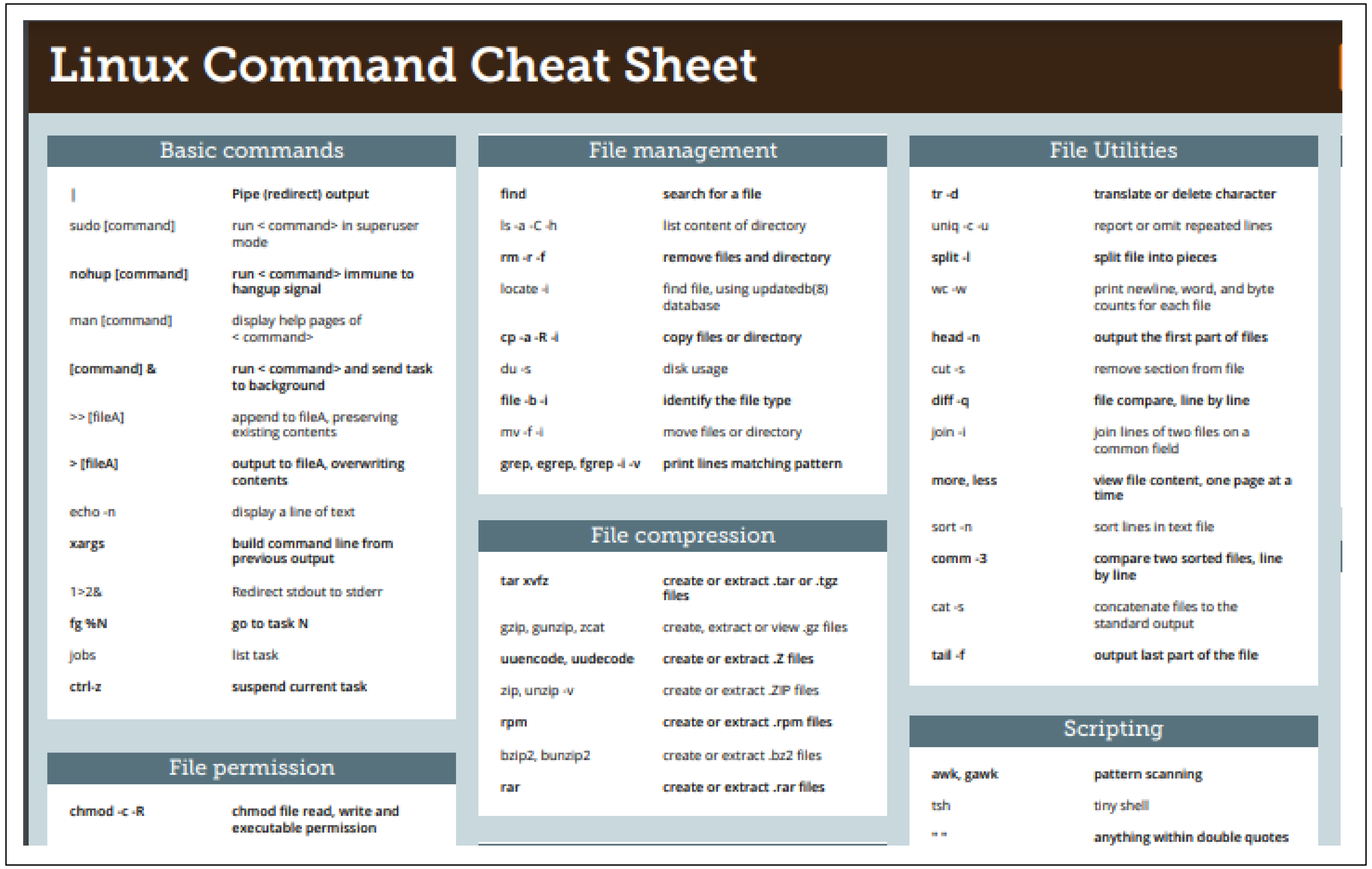
Command	What it Does	Usage
`dir`	Displays a list of a folder's files and subfolders	dir (shows current folder) dir *myfolder*
`cd` `chdir`	Displays the name of the current directory or changes the current folder.	cd *filepath* chdir *filepath* cd .. (goes one directory up)
`md` `mkdir`	Creates a folder (directory)	md *folder-name* mkdir *folder-name*
`rm`	Deletes a folder (directory)	rm *folder-name*

WEB AGE SOLUTIONS

1.8 Linux Command Cheat Sheet

Reference

https://www.loggly.com/wp-content/uploads/2015/05/Linux-Cheat-Sheet-Sponsored-By-Loggly.pdf



Linux Command Cheat Sheet

Basic commands

|	Pipe (redirect) output
sudo [command]	run < command> in superuser mode
nohup [command]	run < command> immune to hangup signal
man [command]	display help pages of < command>
[command] &	run < command> and send task to background
>> [fileA]	append to fileA, preserving existing contents
> [fileA]	output to fileA, overwriting contents
echo -n	display a line of text
xargs	build command line from previous output
1>2&	Redirect stdout to stderr
fg %N	go to task N
jobs	list task
ctrl-z	suspend current task

File permission

chmod -c -R	chmod file read, write and executable permission

File management

find	search for a file
ls -a -C -h	list content of directory
rm -r -f	remove files and directory
locate -i	find file, using updatedb(8) database
cp -a -R -i	copy files or directory
du -s	disk usage
file -b -i	identify the file type
mv -f -i	move files or directory
grep, egrep, fgrep -i -v	print lines matching pattern

File compression

tar xvfz	create or extract .tar or .tgz files
gzip, gunzip, zcat	create, extract or view .gz files
uuencode, uudecode	create or extract .Z files
zip, unzip -v	create or extract .ZIP files
rpm	create or extract .rpm files
bzip2, bunzip2	create or extract .bz2 files
rar	create or extract .rar files

File Utilities

tr -d	translate or delete character
uniq -c -u	report or omit repeated lines
split -l	split file into pieces
wc -w	print newline, word, and byte counts for each file
head -n	output the first part of files
cut -s	remove section from file
diff -q	file compare, line by line
join -i	join lines of two files on a common field
more, less	view file content, one page at a time
sort -n	sort lines in text file
comm -3	compare two sorted files, line by line
cat -s	concatenate files to the standard output
tail -f	output last part of the file

Scripting

awk, gawk	pattern scanning
tsh	tiny shell
" "	anything within double quotes

Summary

- ✓ Have fun with your tasks this week!
- ✓ Remember you can always ask questions in the Basecamp message board. If you run into trouble, or need help, reach out and I can help.
- ✓ Be sure to remember which parts of the course were the most difficult.
- ✓ We can review those at our next Monday session.

WEB AGE SOLUTIONS

Git

1.1 What is Git?

- Git
  - Git is a version control system that intelligently tracks changes in files.
  - Git is particularly useful when you and a group of people are all making changes to the same files at the same time

    https://docs.github.com/en/get-started/start-your-journey/about-github-and-git#about-git

- GitHub
  - GitHub is a cloud-based platform where you can store, share, and work together with others to write code.

    https://docs.github.com/en/get-started/start-your-journey/about-github-and-git#about-github

    https://github.com/

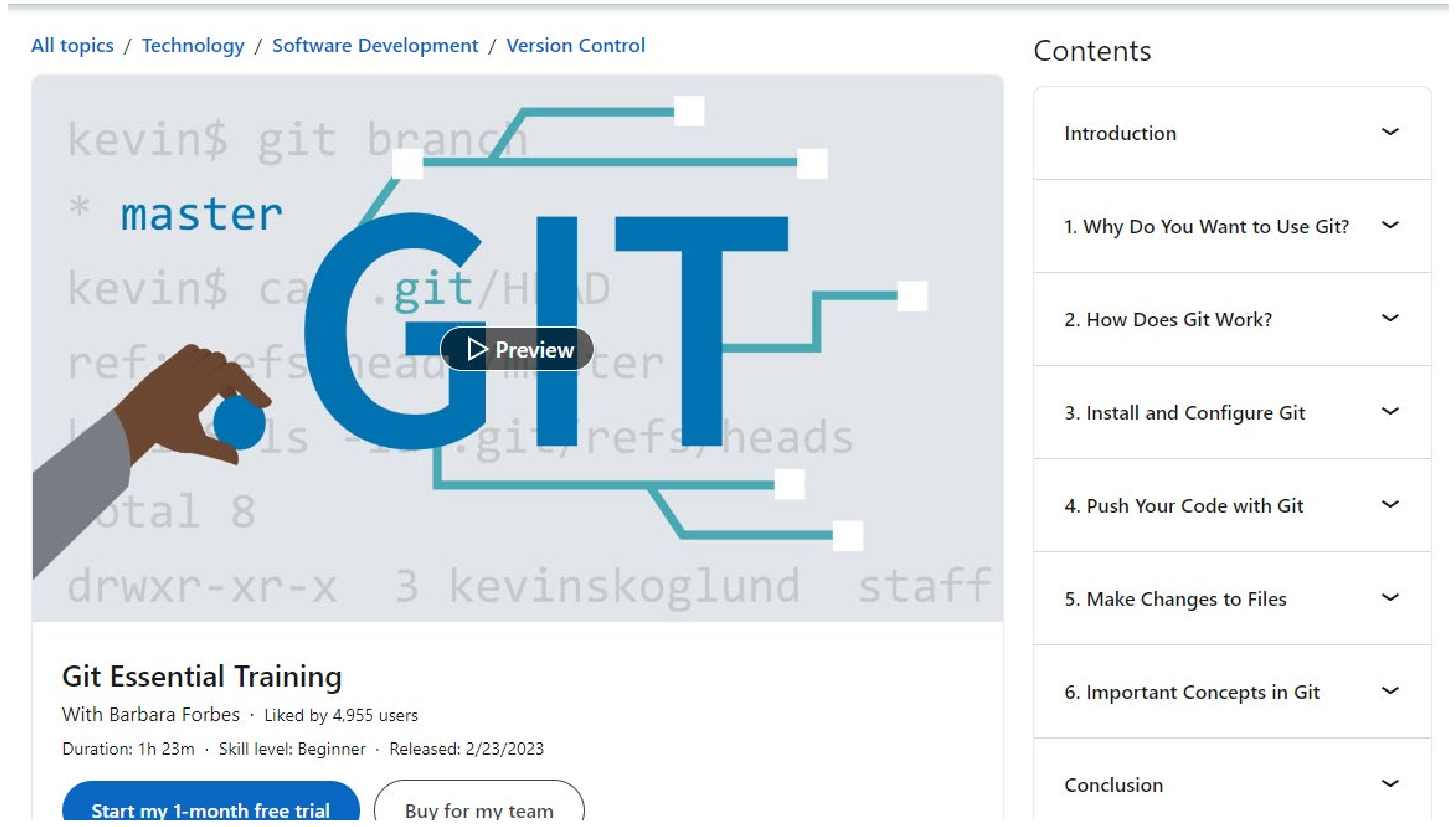
- Collaborative working, one of GitHub's fundamental features, is made possible by the open-source software, Git, upon which GitHub is built.

  https://marklodato.github.io/visual-git-guide/index-en.html

1.2 This Week Study Overview

- For all the coursework this week, use our Lab machine and/or your personal desktop.

- Git Study
  - Study Git Essential Training by Barbara Forbes
    https://www.linkedin.com/learning/git-essential-training-the-basics-2019


All topics / Technology / Software Development / Version Control

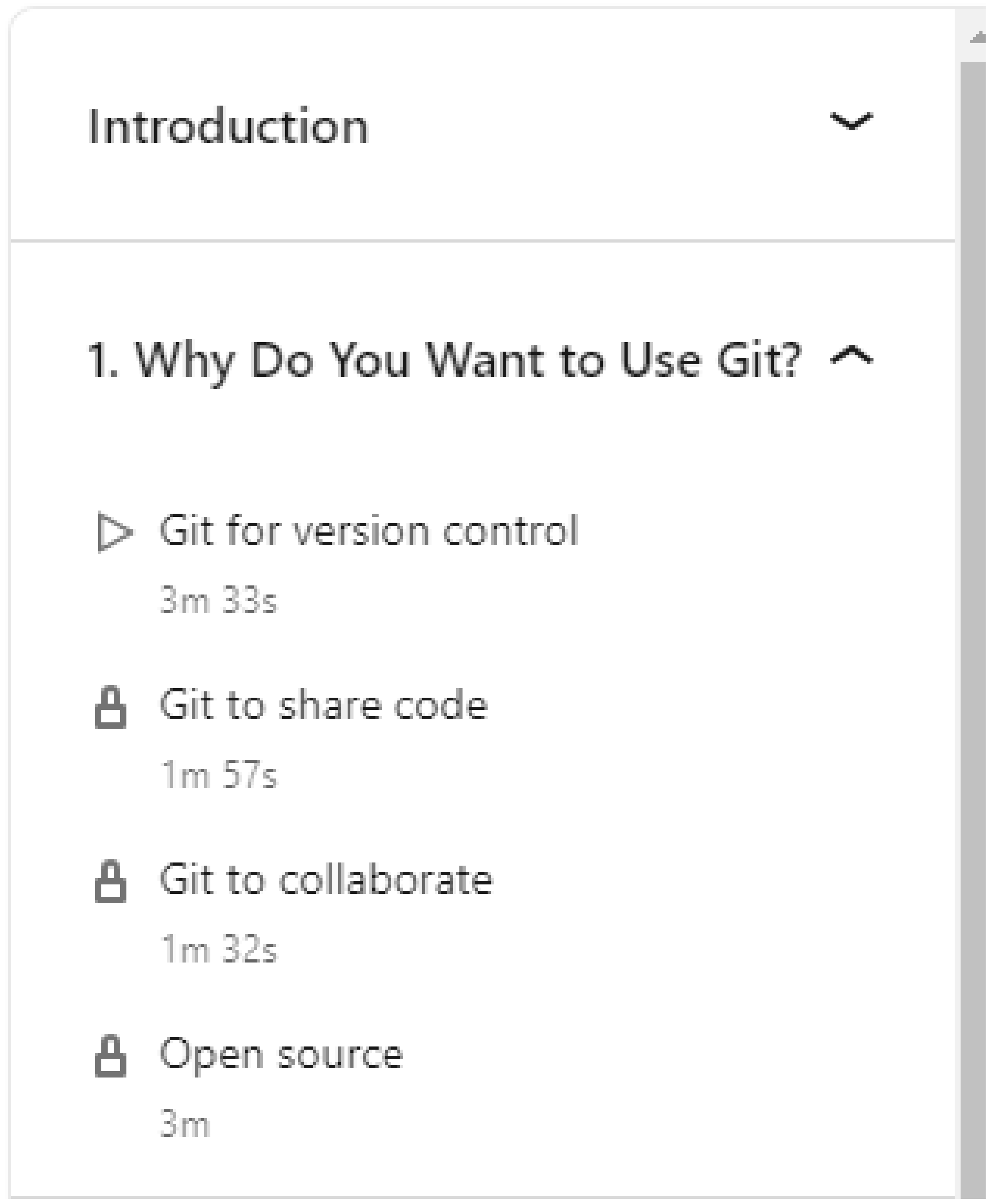
Git Essential Training

With Barbara Forbes · Liked by 4,955 users

Duration: 1h 23m · Skill level: Beginner · Released: 2/23/2023

Contents

1.3 Git Essentials Training Overview



Introduction

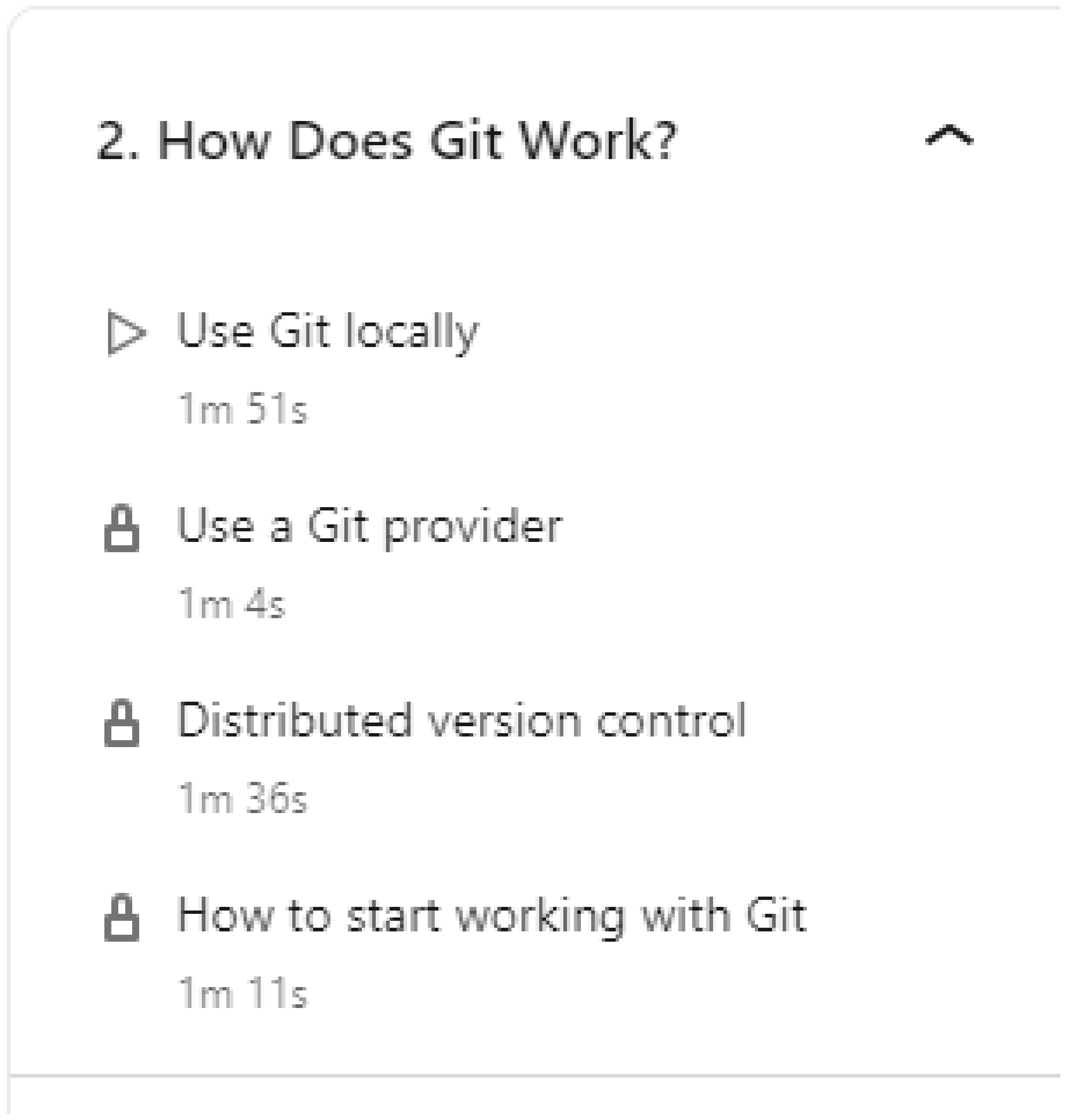
1. Why Do You Want to Use Git?

Git for version control
3m 33s

Git to share code
1m 57s

Git to collaborate
1m 32s

Open source
3m



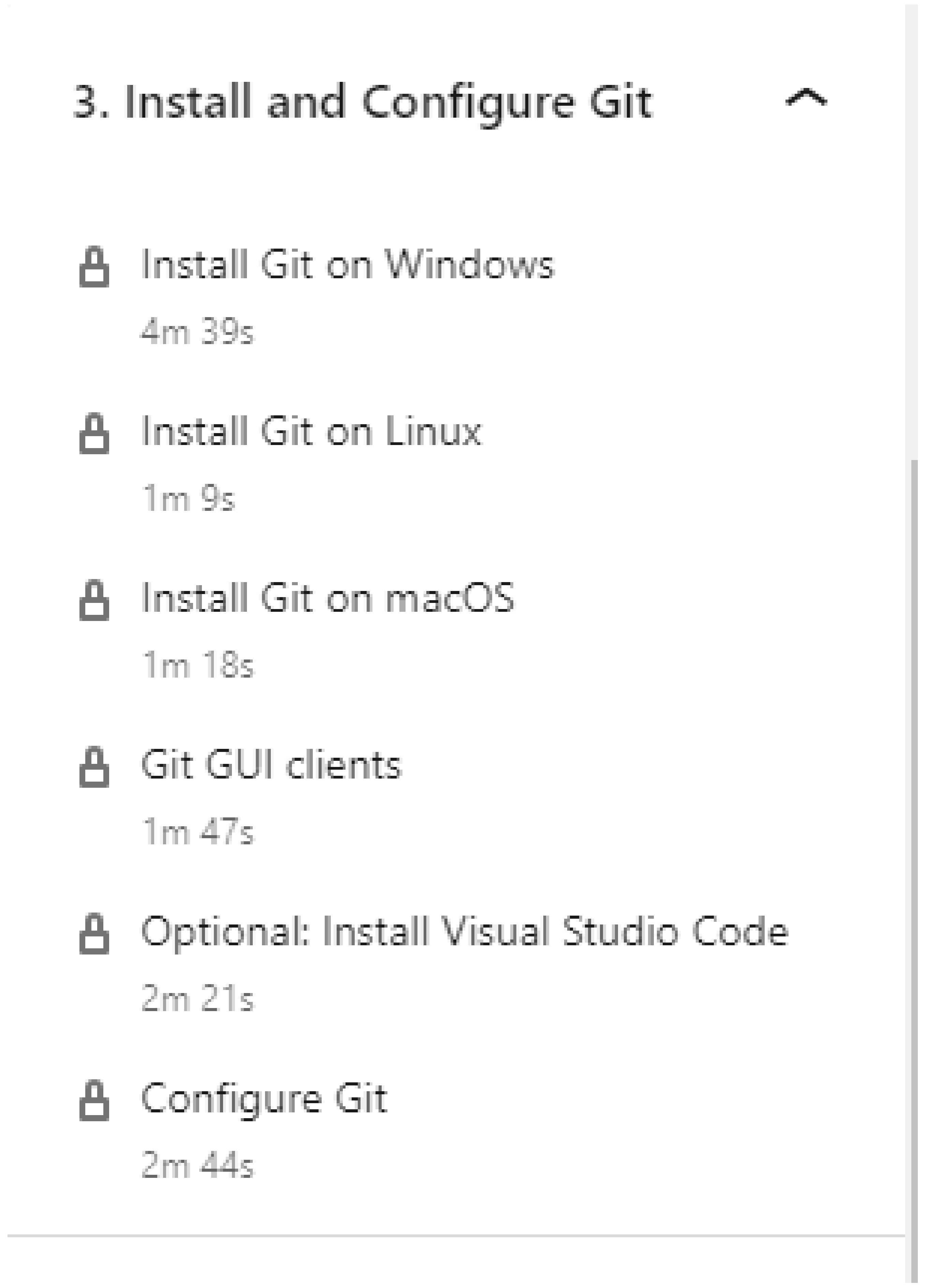
2. How Does Git Work?

Use Git locally
1m 51s

Use a Git provider
1m 4s

Distributed version control
1m 36s

How to start working with Git
1m 11s



3. Install and Configure Git

Install Git on Windows
4m 39s

Install Git on Linux
1m 9s

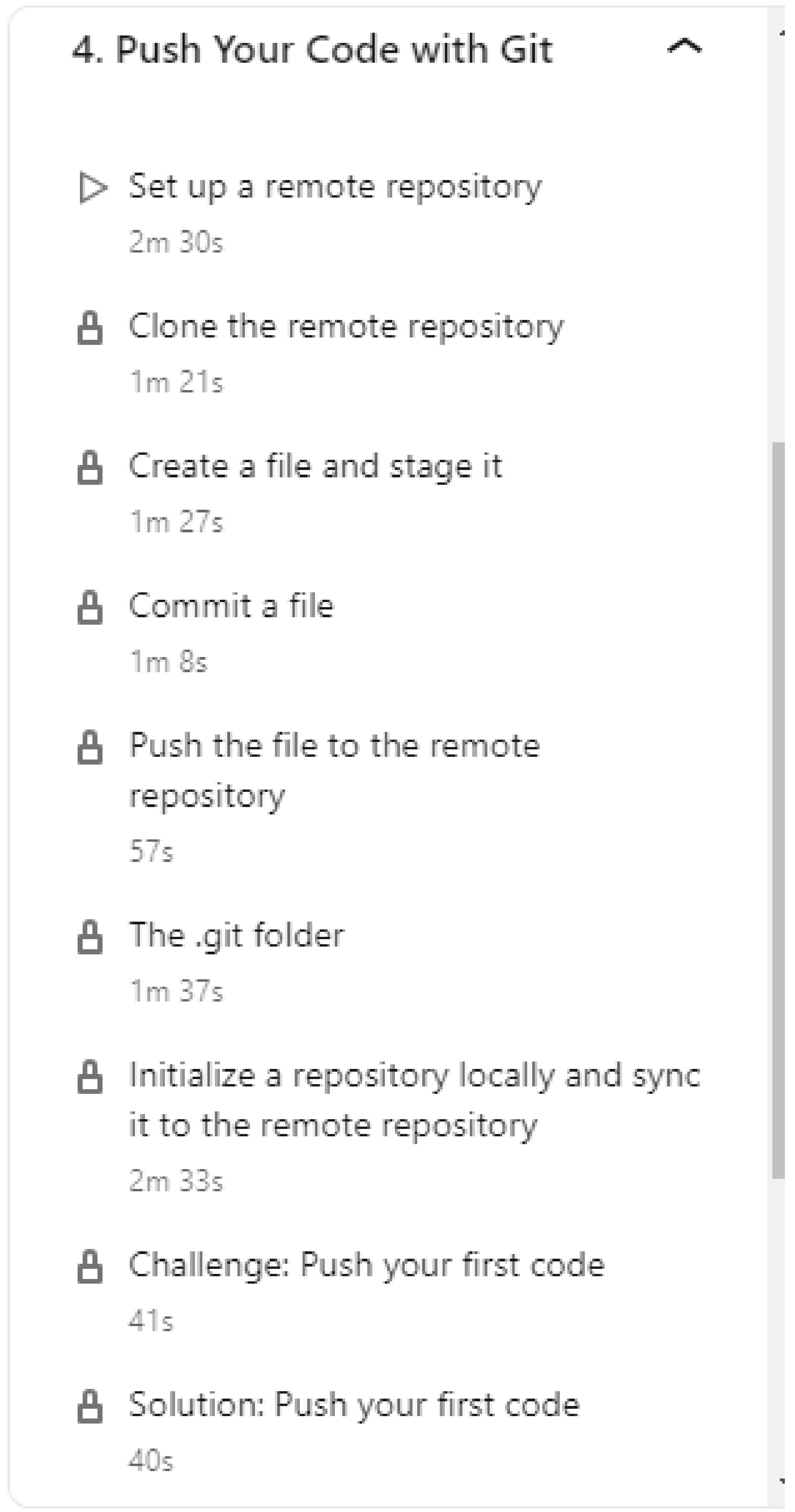
Install Git on macOS
1m 18s

Git GUI clients
1m 47s

Optional: Install Visual Studio Code
2m 21s

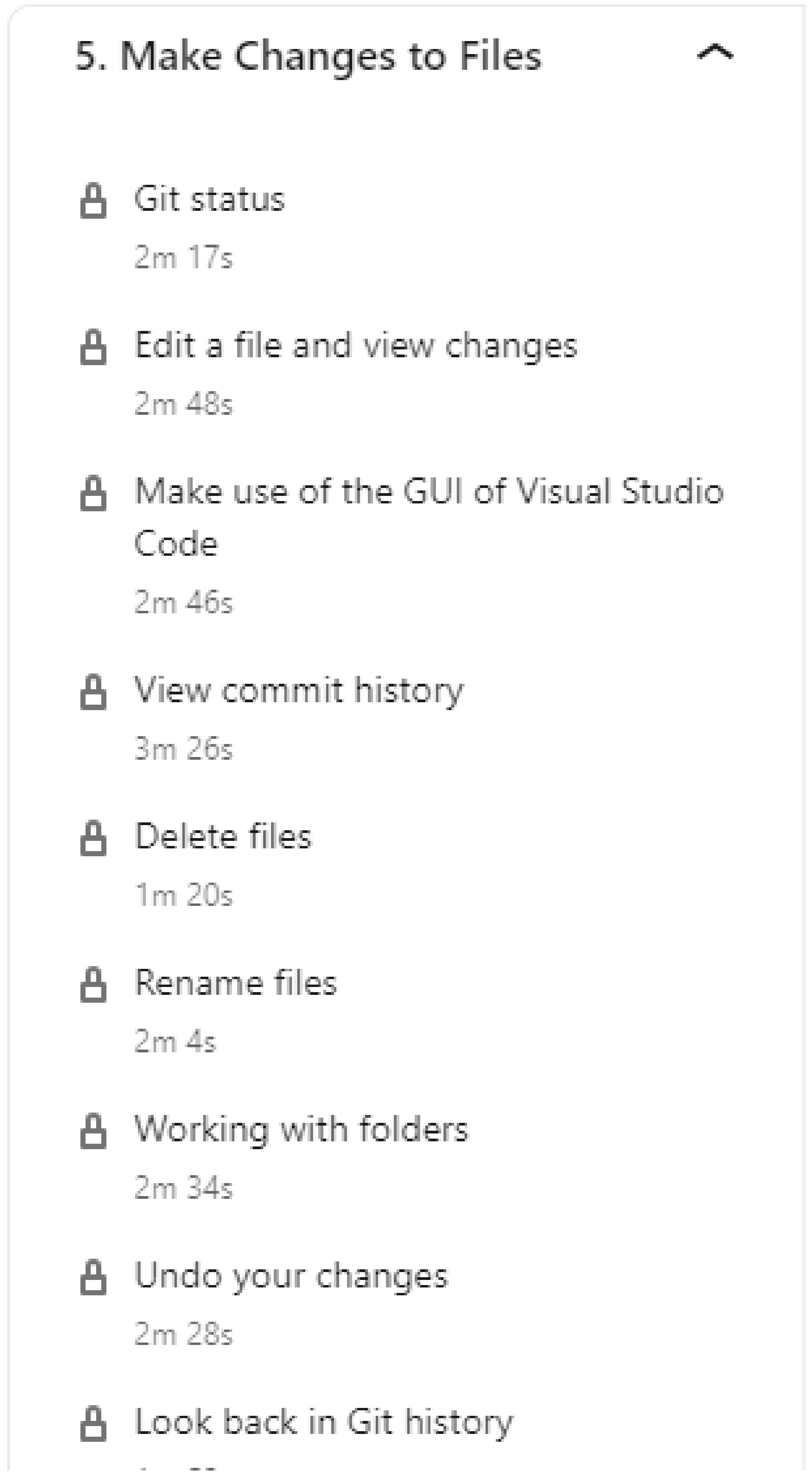
Configure Git
2m 44s

1.3 Git Essentials Training Overview



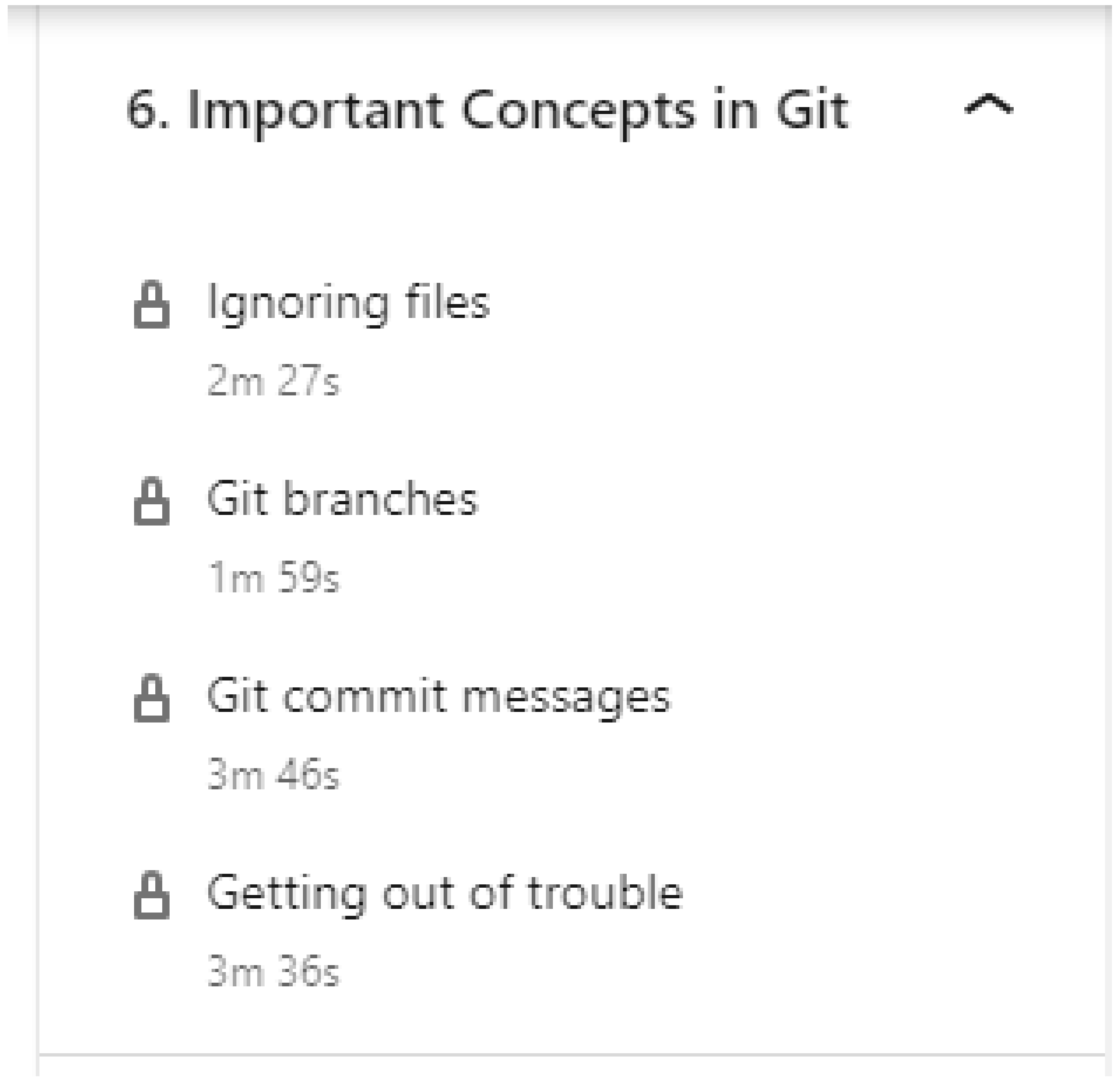
4. Push Your Code with Git

- Set up a remote repository
  2m 30s
- Clone the remote repository
  1m 21s
- Create a file and stage it
  1m 27s
- Commit a file
  1m 8s
- Push the file to the remote repository
  57s
- The .git folder
  1m 37s
- Initialize a repository locally and sync it to the remote repository
  2m 33s
- Challenge: Push your first code
  41s
- Solution: Push your first code
  40s



5. Make Changes to Files

- Git status
  2m 17s
- Edit a file and view changes
  2m 48s
- Make use of the GUI of Visual Studio Code
  2m 46s
- View commit history
  3m 26s
- Delete files
  1m 20s
- Rename files
  2m 4s
- Working with folders
  2m 34s
- Undo your changes
  2m 28s
- Look back in Git history



6. Important Concepts in Git

- Ignoring files
  2m 27s
- Git branches
  1m 59s
- Git commit messages
  3m 46s
- Getting out of trouble
  3m 36s

1.4 Git Essential Training Quizzes

Chapter 1- Why Do You Want to Use Git?

7 quiz questions

Chapter 2- How Does Git Work?

7 quiz questions

Chapter 3- Install and Configure Git

12 quiz questions

Chapter 4- Push Your Code with Git

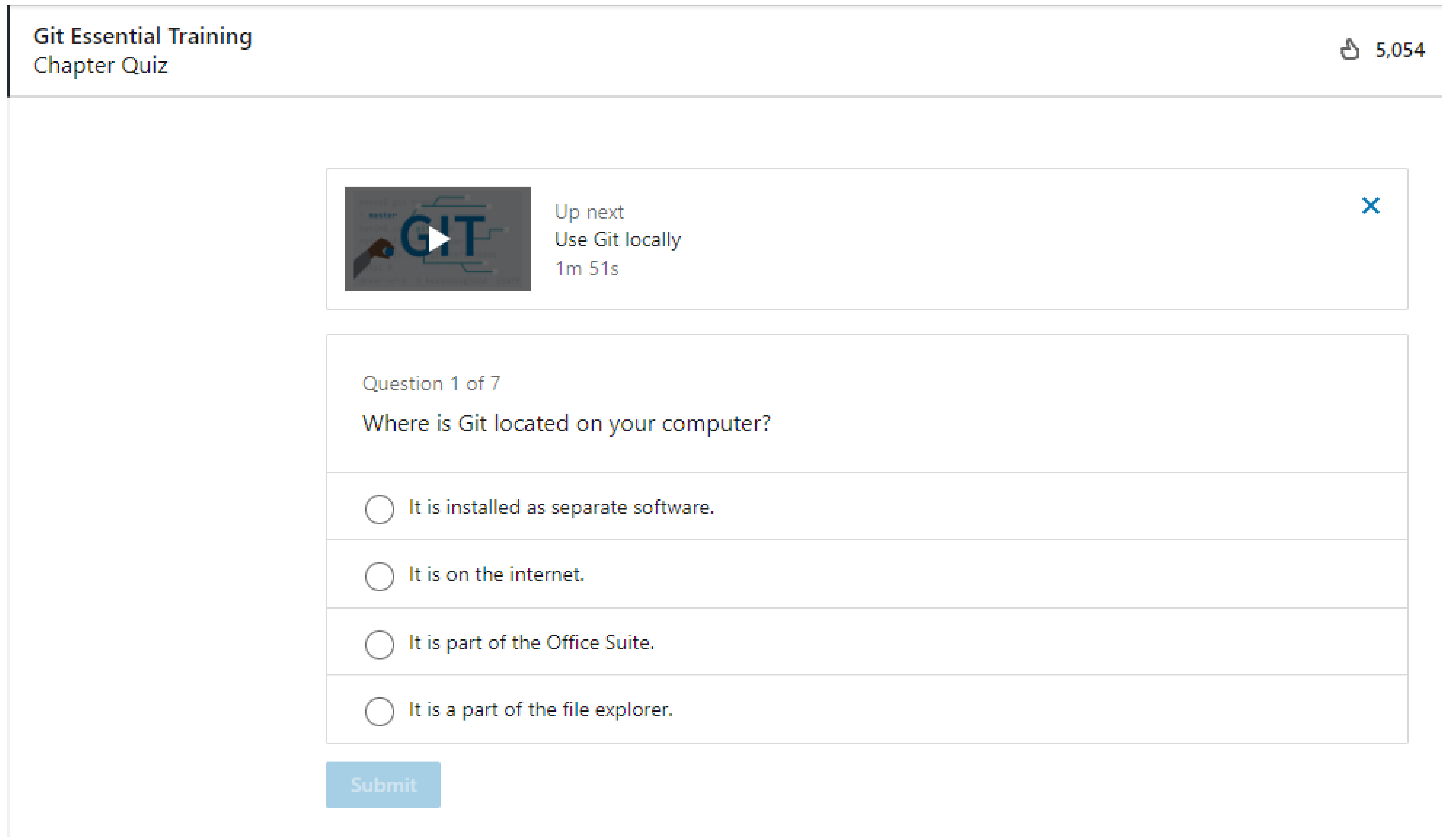
13 quiz questions

Chapter 5- Make Changes to File

22 quiz questions

Chapter 6- Important Concepts in Git

7 quiz questions


Git Essential Training
Chapter Quiz

5,054

Up next
Use Git locally
1m 51s

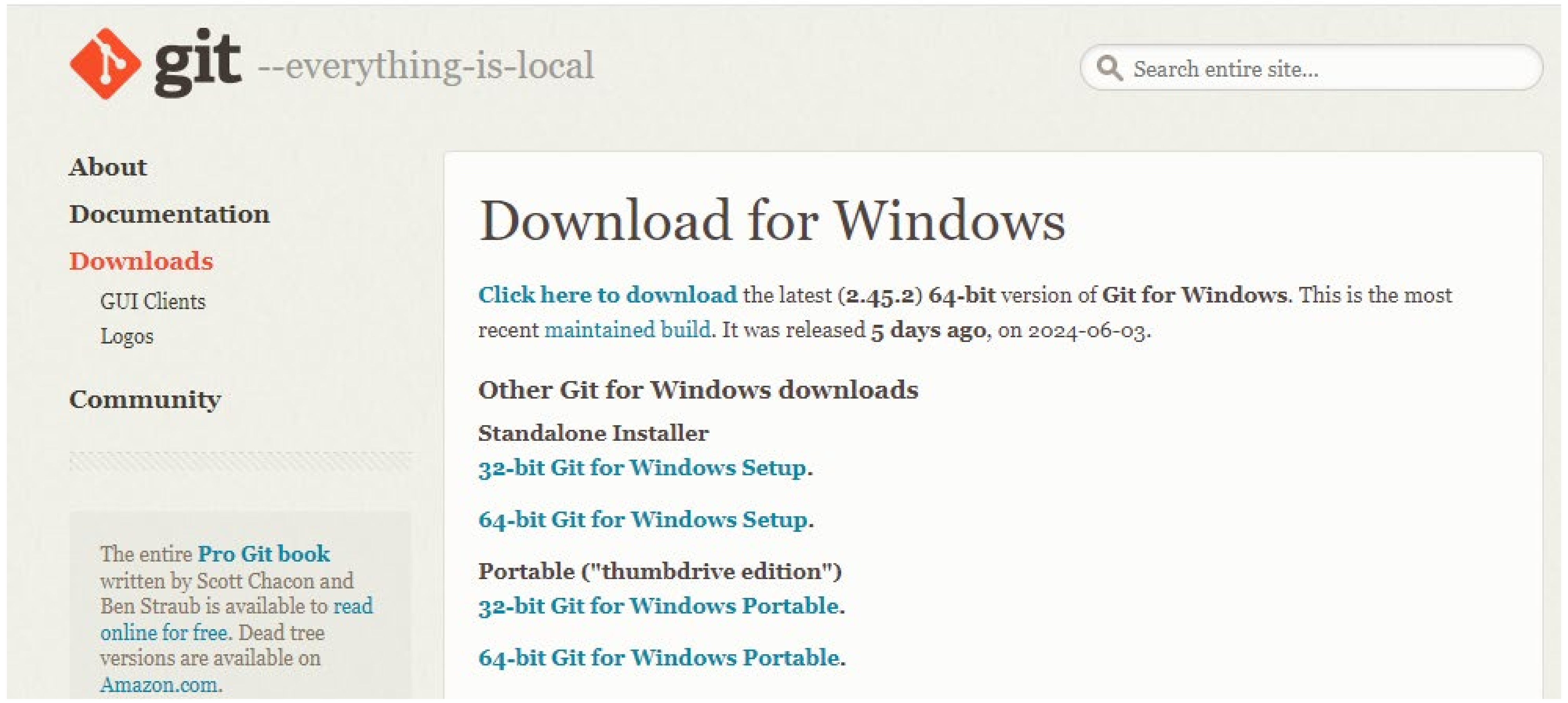
Question 1 of 7
Where is Git located on your computer?

- It is installed as separate software.
- It is on the internet.
- It is part of the Office Suite.
- It is a part of the file explorer.

WEB AGE SOLUTIONS

1.5 Git Download

https://www.git-scm.com/downloads
https://www.git-scm.com/download/mac
https://www.git-scm.com/download/win
https://www.git-scm.com/download/linux


git --everything-is-local

Search entire site...

About
Documentation
Downloads
GUI Clients
Logos
Community

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.

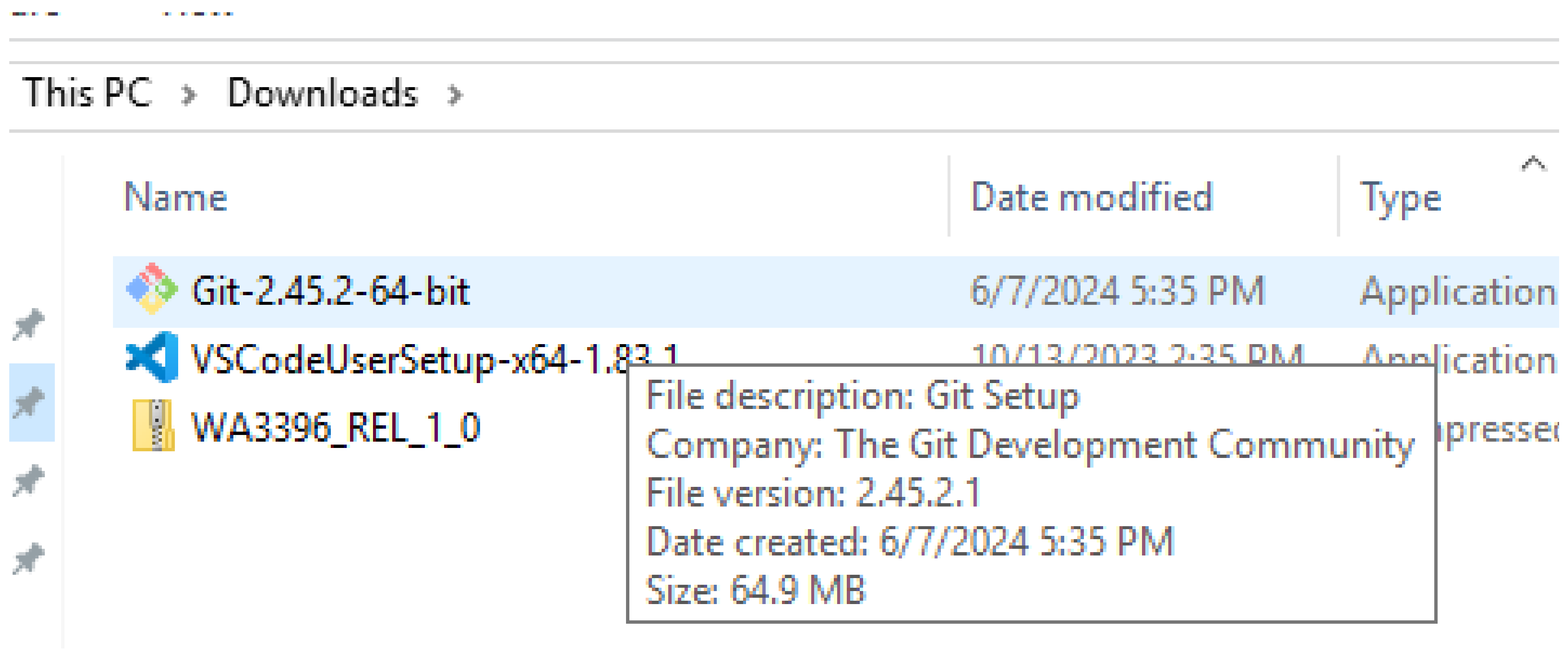
Download for Windows

Click here to download the latest (2.45.2) 64-bit version of Git for Windows. This is the most recent maintained build. It was released 5 days ago, on 2024-06-03.

Other Git for Windows downloads

Standalone Installer
32-bit Git for Windows Setup.
64-bit Git for Windows Setup.

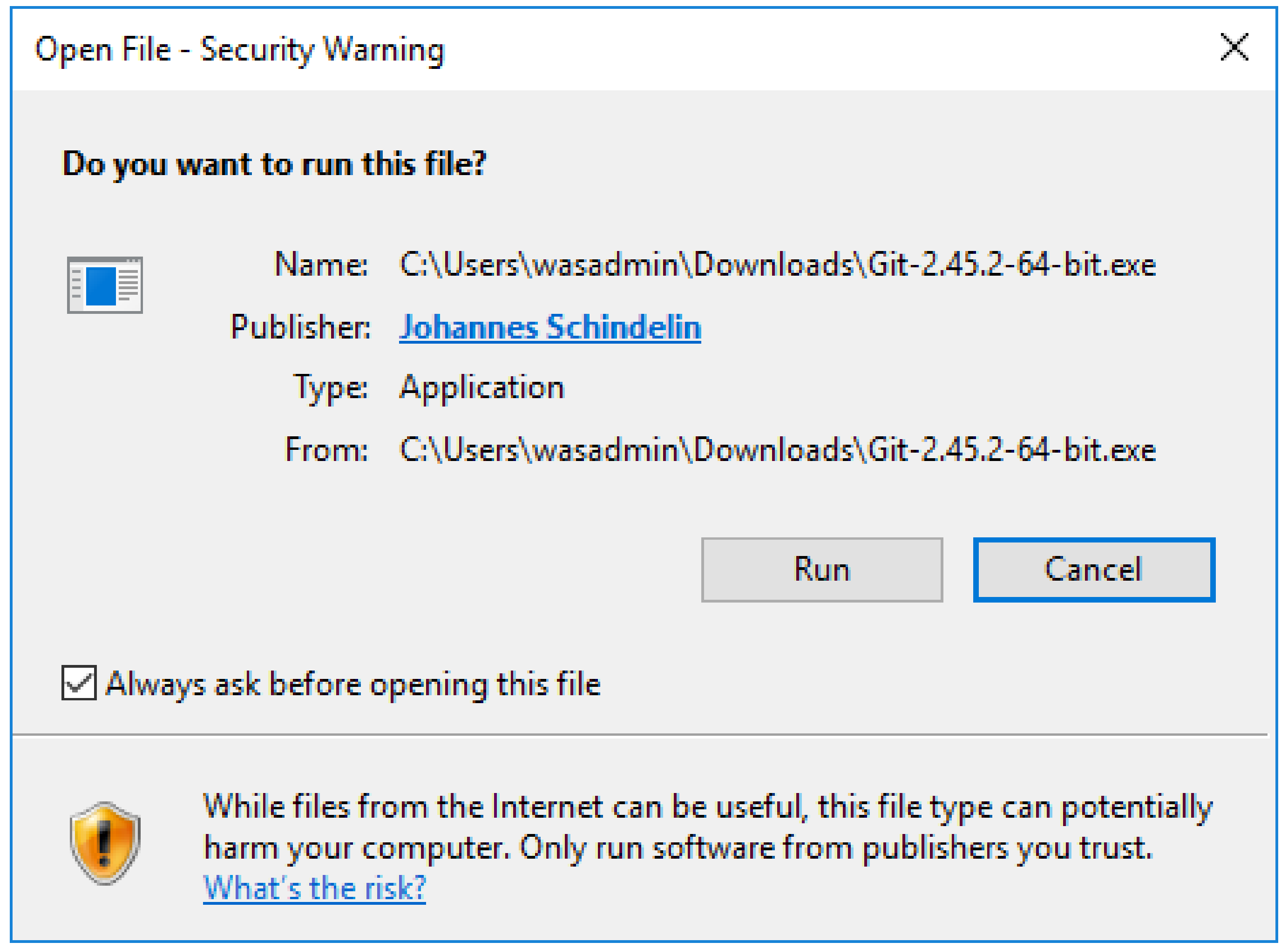
Portable ("thumbdrive edition")
32-bit Git for Windows Portable.
64-bit Git for Windows Portable.


This PC › Downloads ›

Name	Date modified	Type
Git-2.45.2-64-bit	6/7/2024 5:35 PM	Application
VSCodeUserSetup-x64-1.83.1	10/13/2023 2:35 PM	Application
WA3396_REL_1_0		pressed

File description: Git Setup
Company: The Git Development Community
File version: 2.45.2.1
Date created: 6/7/2024 5:35 PM
Size: 64.9 MB

1.6 Install Git (Steps 1-6)

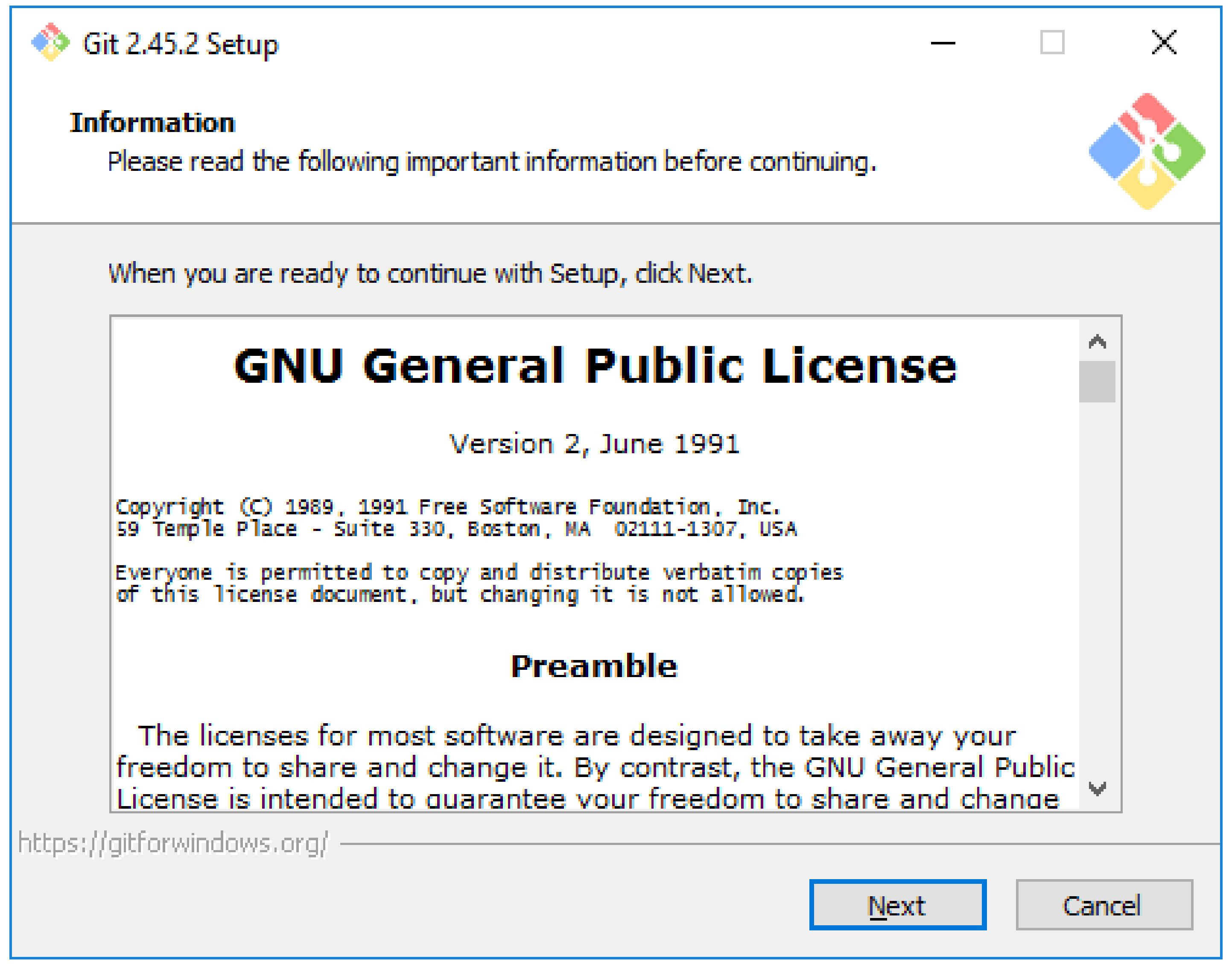

Open File - Security Warning

Do you want to run this file?

Name: C:\Users\wasadmin\Downloads\Git-2.45.2-64-bit.exe
Publisher: Johannes Schindelin
Type: Application
From: C:\Users\wasadmin\Downloads\Git-2.45.2-64-bit.exe

☑ Always ask before opening this file

While files from the Internet can be useful, this file type can potentially harm your computer. Only run software from publishers you trust. What's the risk?


Git 2.45.2 Setup

Information
Please read the following important information before continuing.

When you are ready to continue with Setup, click Next.

GNU General Public License

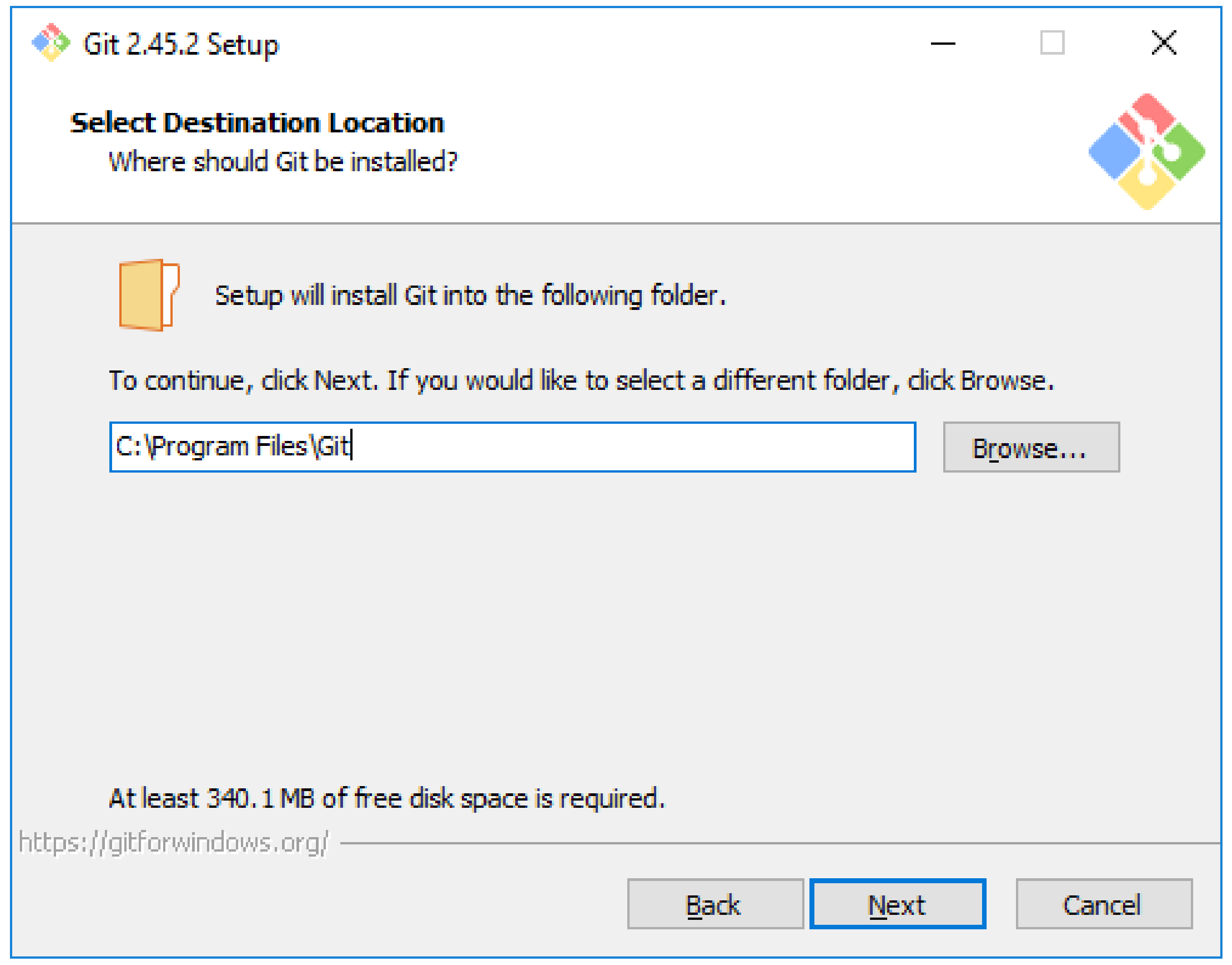
Version 2, June 1991

```
Copyright (C) 1989, 1991 Free Software Foundation, Inc.
59 Temple Place - Suite 330, Boston, MA  02111-1307, USA

Everyone is permitted to copy and distribute verbatim copies
of this license document, but changing it is not allowed.
```

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change

https://gitforwindows.org/


Git 2.45.2 Setup

Select Destination Location
Where should Git be installed?

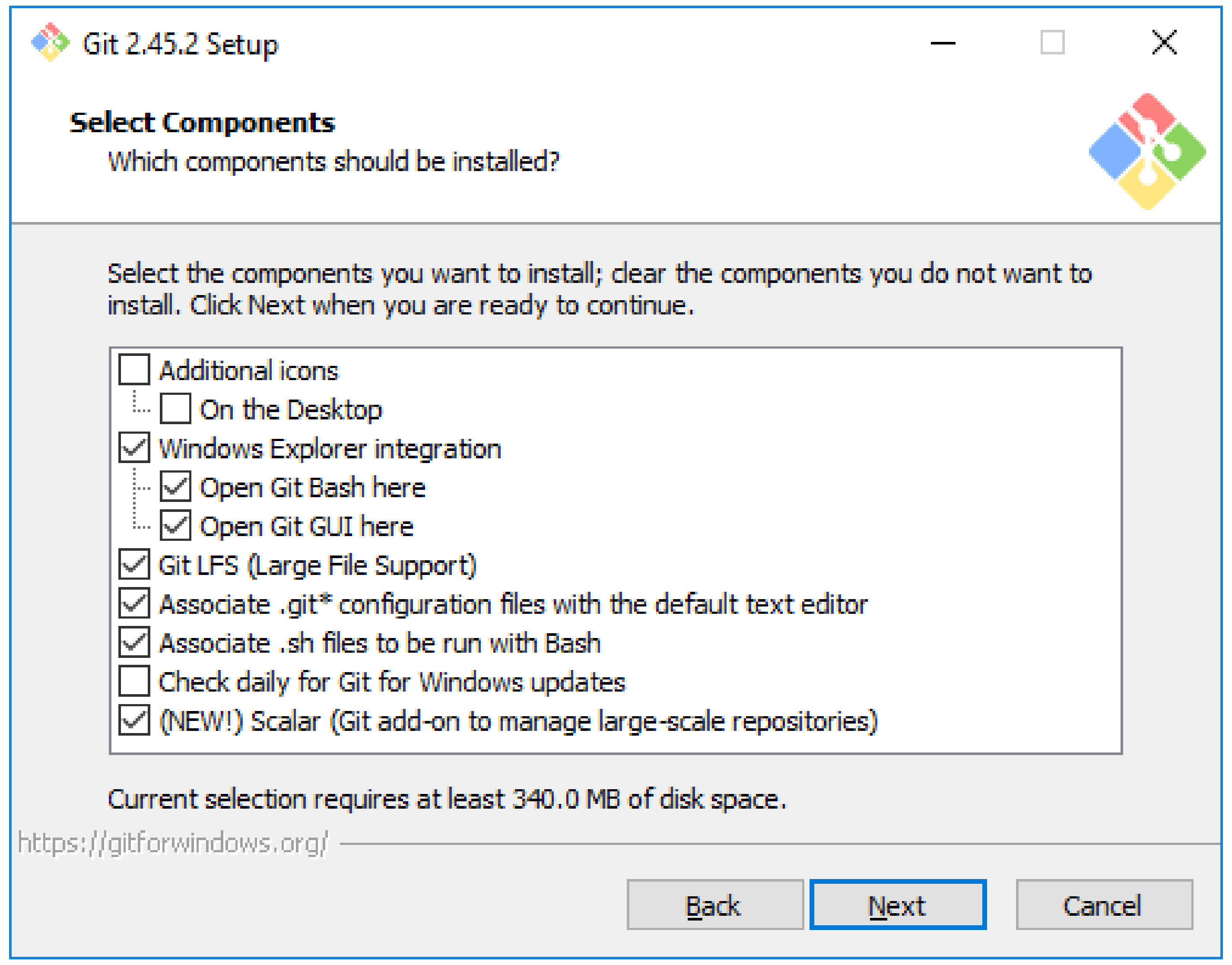
Setup will install Git into the following folder.

To continue, click Next. If you would like to select a different folder, click Browse.

C:\Program Files\Git

At least 340.1 MB of free disk space is required.

https://gitforwindows.org/


Git 2.45.2 Setup

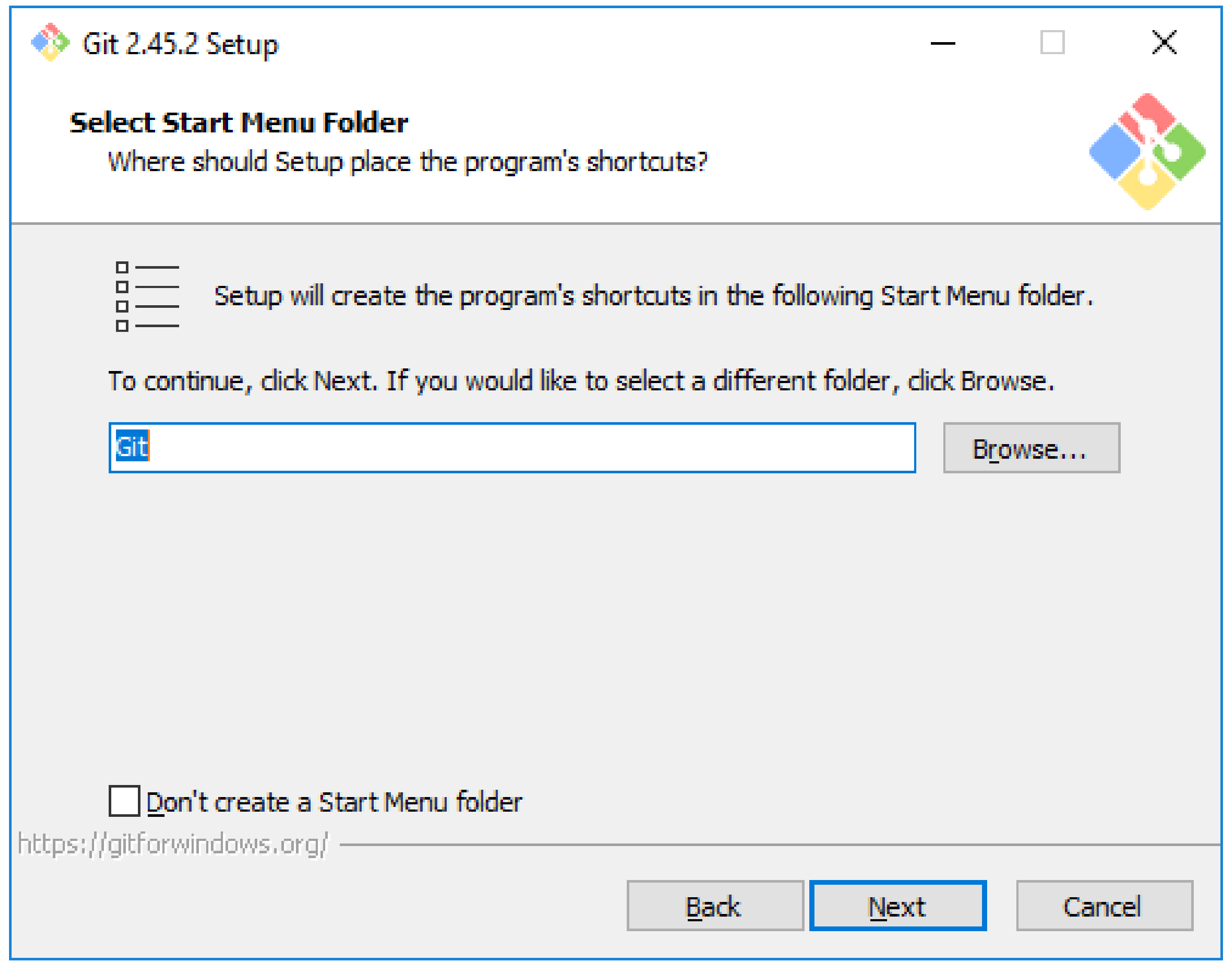
Select Components
Which components should be installed?

Select the components you want to install; clear the components you do not want to install. Click Next when you are ready to continue.

- ☐ Additional icons
  - ☐ On the Desktop
- ☑ Windows Explorer integration
  - ☑ Open Git Bash here
  - ☑ Open Git GUI here
- ☑ Git LFS (Large File Support)
- ☑ Associate .git* configuration files with the default text editor
- ☑ Associate .sh files to be run with Bash
- ☐ Check daily for Git for Windows updates
- ☑ (NEW!) Scalar (Git add-on to manage large-scale repositories)

Current selection requires at least 340.0 MB of disk space.

https://gitforwindows.org/


Git 2.45.2 Setup

Select Start Menu Folder
Where should Setup place the program's shortcuts?

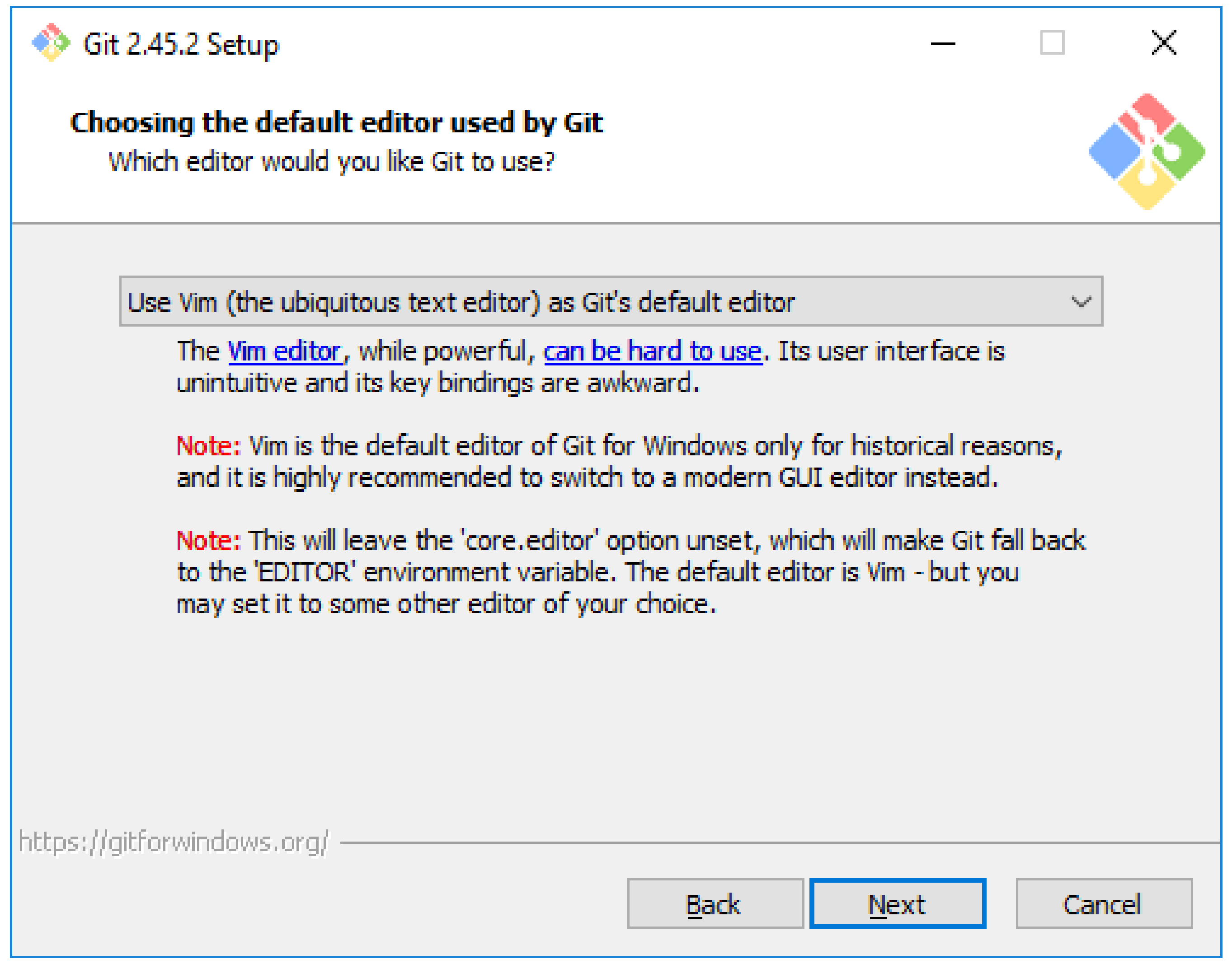
Setup will create the program's shortcuts in the following Start Menu folder.

To continue, click Next. If you would like to select a different folder, click Browse.

Git

☐ Don't create a Start Menu folder

https://gitforwindows.org/


Git 2.45.2 Setup

Choosing the default editor used by Git
Which editor would you like Git to use?

Use Vim (the ubiquitous text editor) as Git's default editor

The Vim editor, while powerful, can be hard to use. Its user interface is unintuitive and its key bindings are awkward.

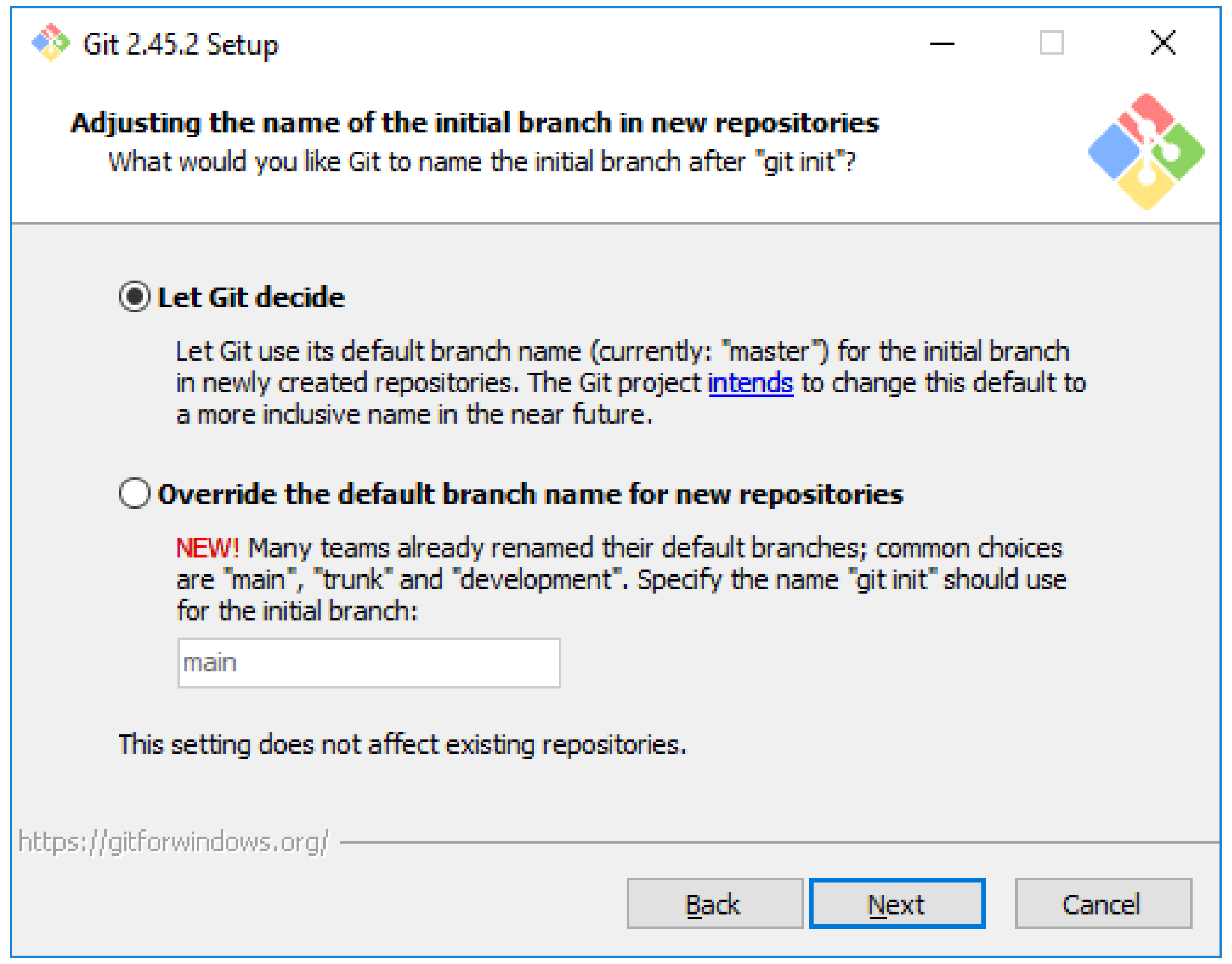
Note: Vim is the default editor of Git for Windows only for historical reasons, and it is highly recommended to switch to a modern GUI editor instead.

Note: This will leave the 'core.editor' option unset, which will make Git fall back to the 'EDITOR' environment variable. The default editor is Vim - but you may set it to some other editor of your choice.

https://gitforwindows.org/

WEB AGE SOLUTIONS

1.6 Install Git (Steps 7-12)


Git 2.45.2 Setup

Adjusting the name of the initial branch in new repositories
What would you like Git to name the initial branch after "git init"?

[✓ **Let Git decide**]

Let Git use its default branch name (currently: "master") for the initial branch in newly created repositories. The Git project intends to change this default to a more inclusive name in the near future.

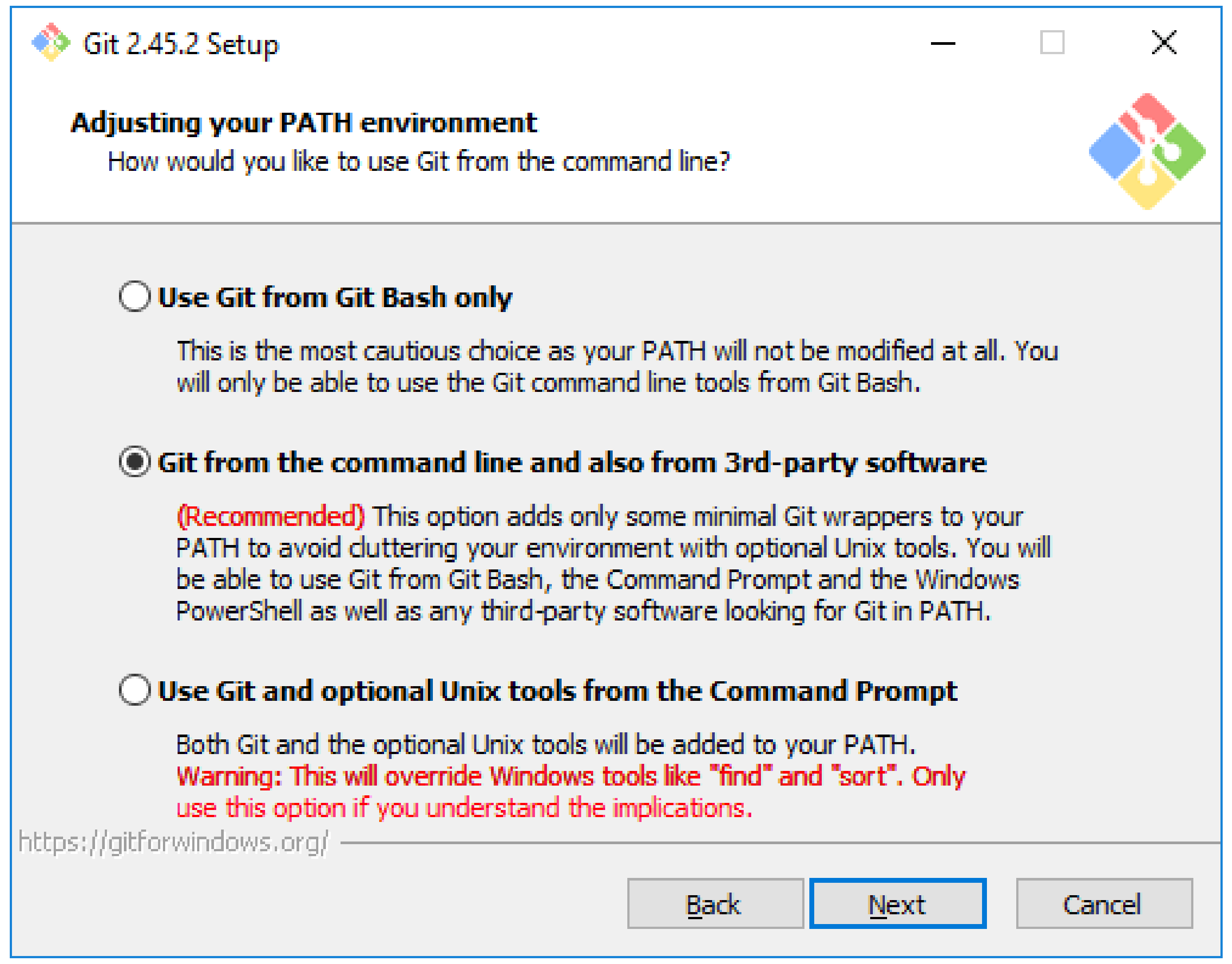
Override the default branch name for new repositories

NEW! Many teams already renamed their default branches; common choices are "main", "trunk" and "development". Specify the name "git init" should use for the initial branch:

main

This setting does not affect existing repositories.

https://gitforwindows.org/


Git 2.45.2 Setup

Adjusting your PATH environment
How would you like to use Git from the command line?

Use Git from Git Bash only

This is the most cautious choice as your PATH will not be modified at all. You will only be able to use the Git command line tools from Git Bash.

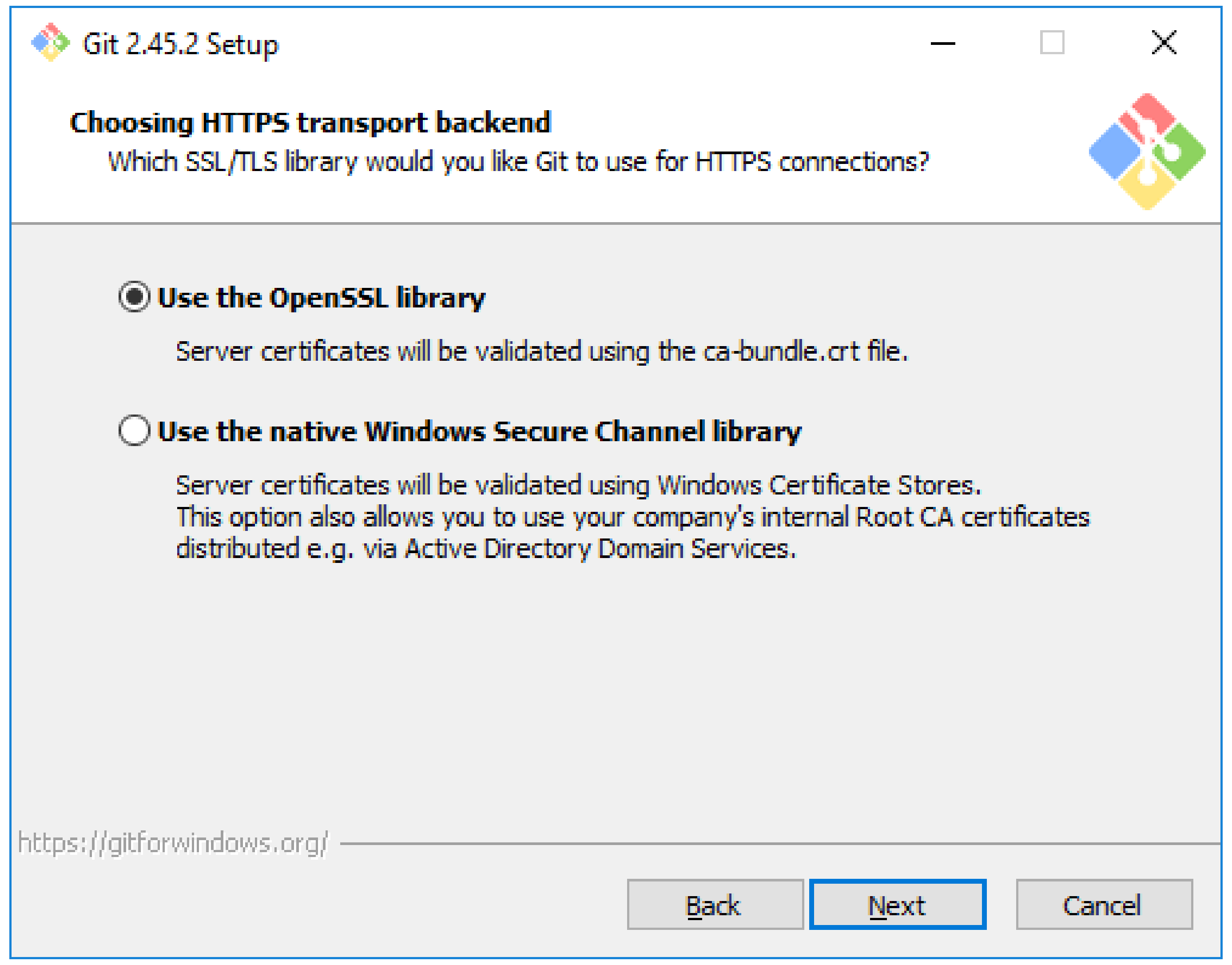
[✓ **Git from the command line and also from 3rd-party software**]

(Recommended) This option adds only some minimal Git wrappers to your PATH to avoid cluttering your environment with optional Unix tools. You will be able to use Git from Git Bash, the Command Prompt and the Windows PowerShell as well as any third-party software looking for Git in PATH.

Use Git and optional Unix tools from the Command Prompt

Both Git and the optional Unix tools will be added to your PATH.
Warning: This will override Windows tools like "find" and "sort". Only use this option if you understand the implications.

https://gitforwindows.org/


Git 2.45.2 Setup

Choosing HTTPS transport backend
Which SSL/TLS library would you like Git to use for HTTPS connections?

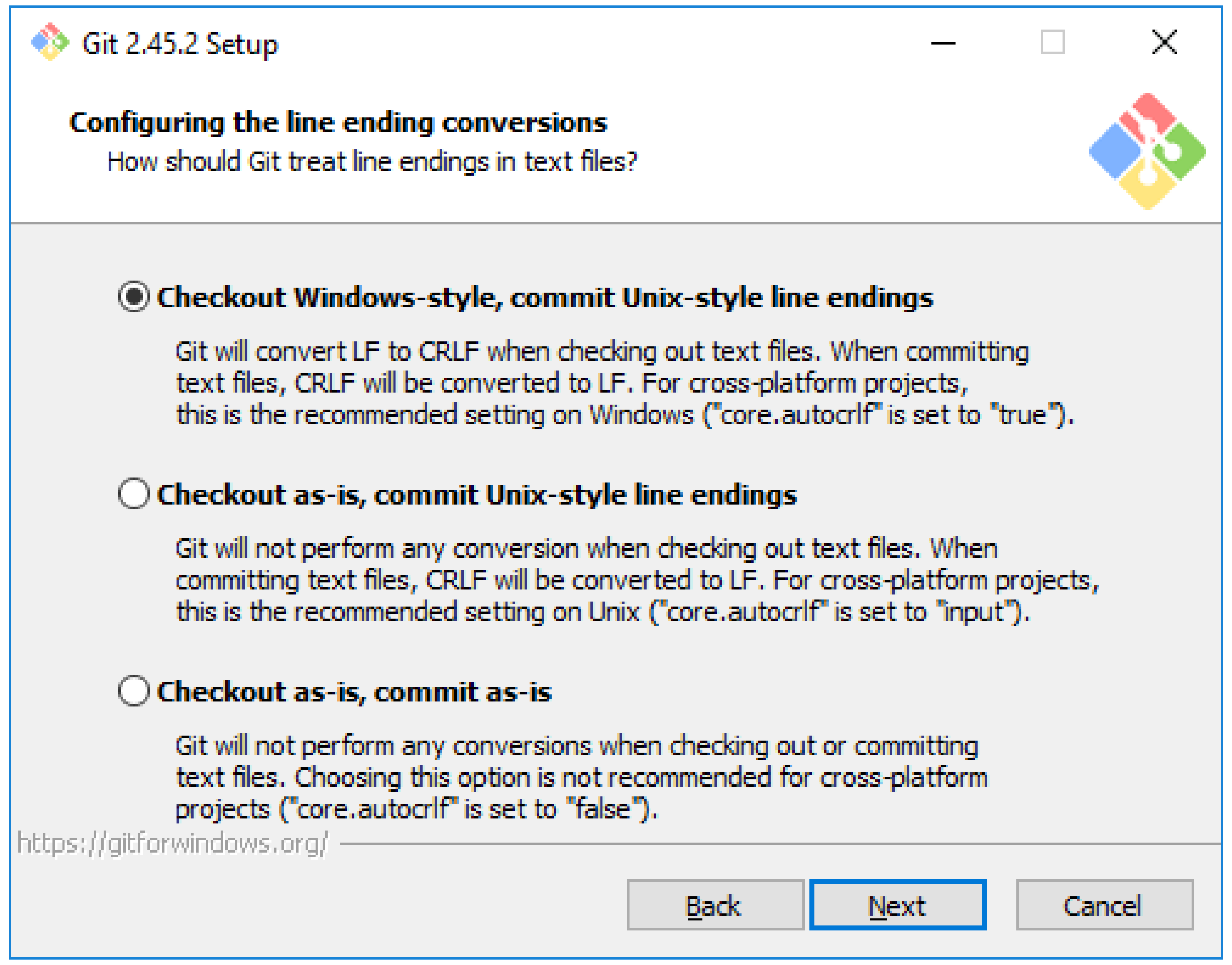
[✓ **Use the OpenSSL library**]

Server certificates will be validated using the ca-bundle.crt file.

Use the native Windows Secure Channel library

Server certificates will be validated using Windows Certificate Stores.
This option also allows you to use your company's internal Root CA certificates distributed e.g. via Active Directory Domain Services.

https://gitforwindows.org/


Git 2.45.2 Setup

Configuring the line ending conversions
How should Git treat line endings in text files?

[✓ **Checkout Windows-style, commit Unix-style line endings**]

Git will convert LF to CRLF when checking out text files. When committing text files, CRLF will be converted to LF. For cross-platform projects, this is the recommended setting on Windows ("core.autocrlf" is set to "true").

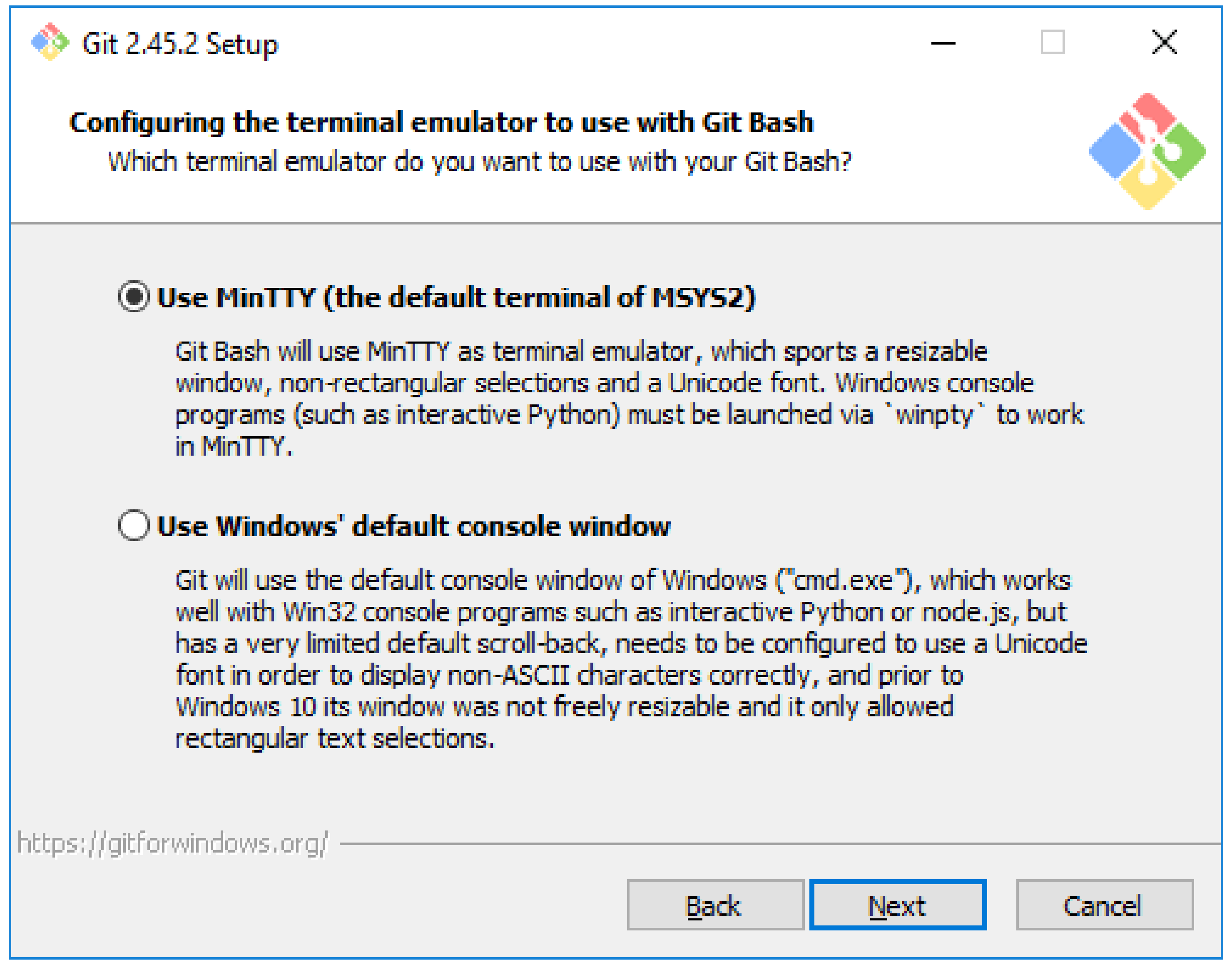
Checkout as-is, commit Unix-style line endings

Git will not perform any conversion when checking out text files. When committing text files, CRLF will be converted to LF. For cross-platform projects, this is the recommended setting on Unix ("core.autocrlf" is set to "input").

Checkout as-is, commit as-is

Git will not perform any conversions when checking out or committing text files. Choosing this option is not recommended for cross-platform projects ("core.autocrlf" is set to "false").

https://gitforwindows.org/


Git 2.45.2 Setup

Configuring the terminal emulator to use with Git Bash
Which terminal emulator do you want to use with your Git Bash?

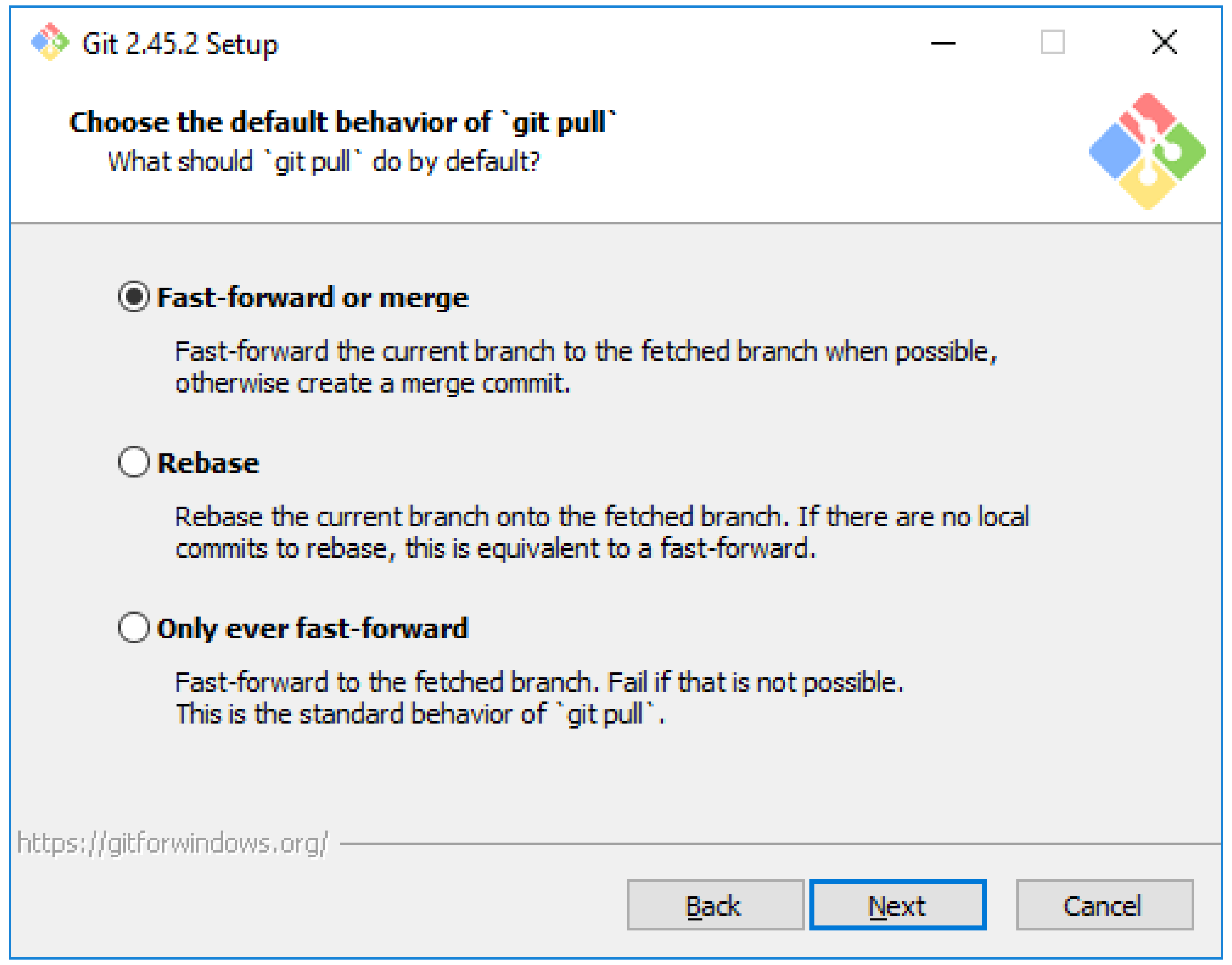
[✓ **Use MinTTY (the default terminal of MSYS2)**]

Git Bash will use MinTTY as terminal emulator, which sports a resizable window, non-rectangular selections and a Unicode font. Windows console programs (such as interactive Python) must be launched via `winpty` to work in MinTTY.

Use Windows' default console window

Git will use the default console window of Windows ("cmd.exe"), which works well with Win32 console programs such as interactive Python or node.js, but has a very limited default scroll-back, needs to be configured to use a Unicode font in order to display non-ASCII characters correctly, and prior to Windows 10 its window was not freely resizable and it only allowed rectangular text selections.

https://gitforwindows.org/


Git 2.45.2 Setup

Choose the default behavior of `git pull`
What should `git pull` do by default?

[✓ **Fast-forward or merge**]

Fast-forward the current branch to the fetched branch when possible, otherwise create a merge commit.

Rebase

Rebase the current branch onto the fetched branch. If there are no local commits to rebase, this is equivalent to a fast-forward.

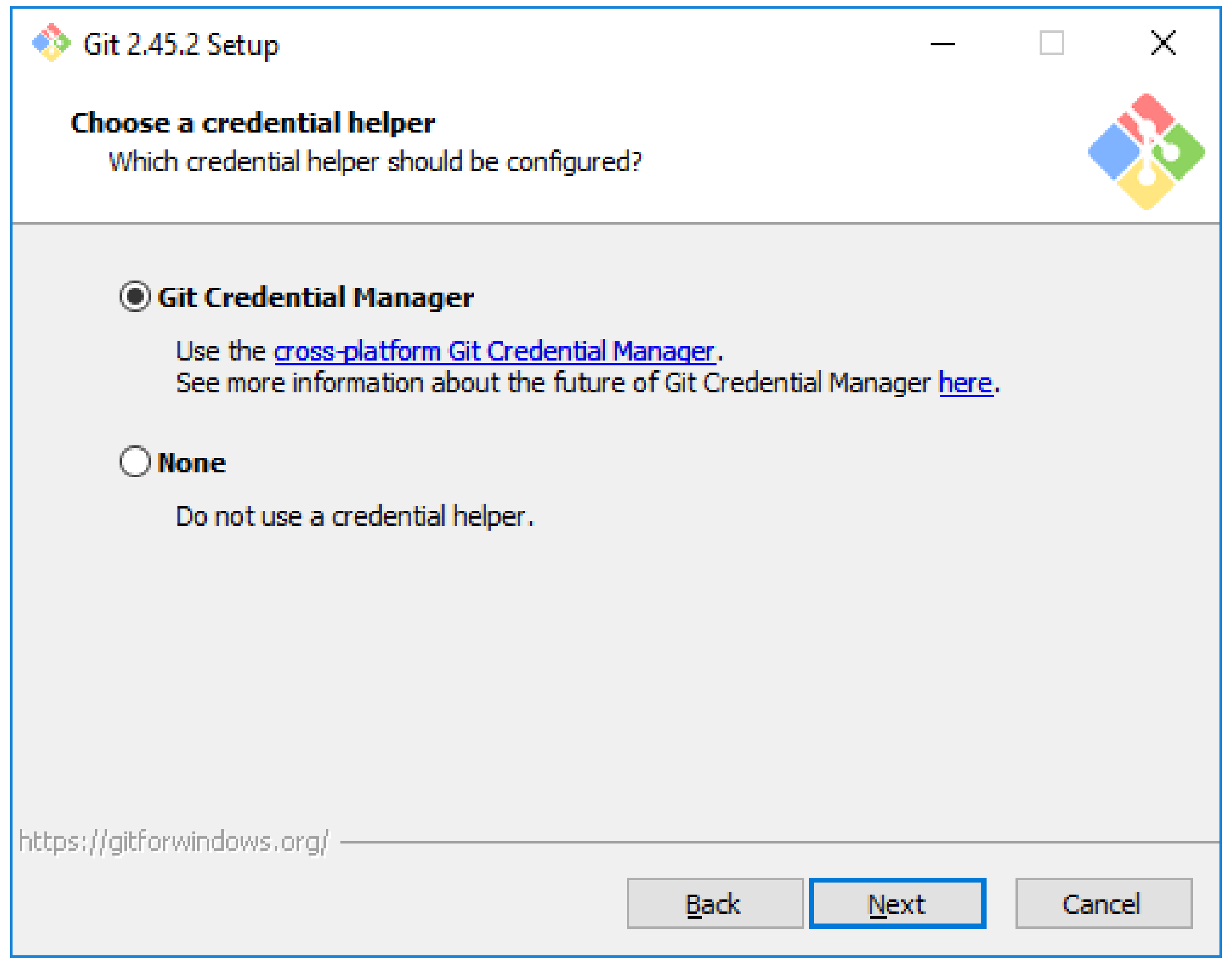
Only ever fast-forward

Fast-forward to the fetched branch. Fail if that is not possible.
This is the standard behavior of `git pull`.

https://gitforwindows.org/

WEB AGE SOLUTIONS

1.6 Install Git (Steps 13-17)



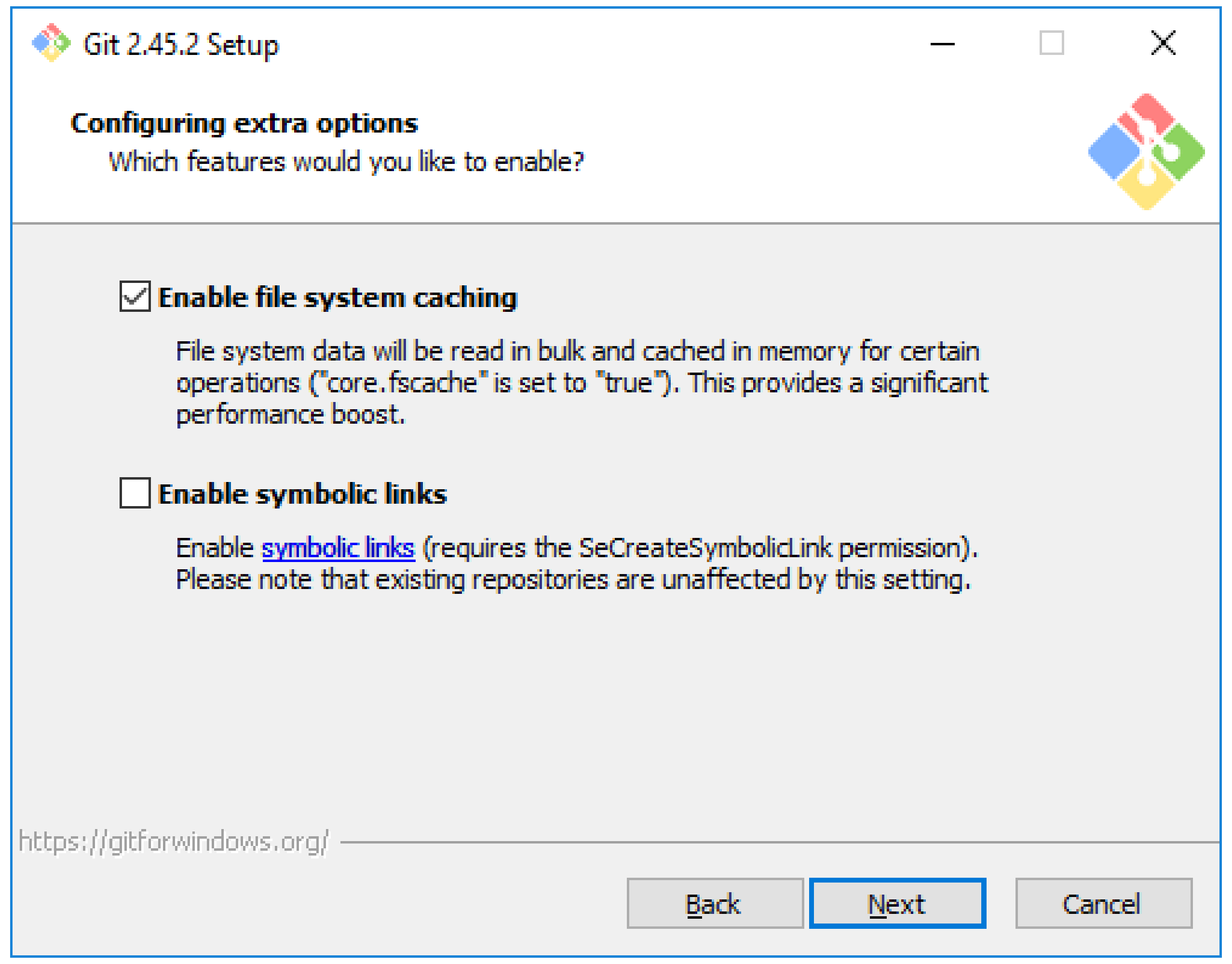
Git 2.45.2 Setup

Choose a credential helper
Which credential helper should be configured?

● **Git Credential Manager**
Use the cross-platform Git Credential Manager.
See more information about the future of Git Credential Manager here.

○ **None**
Do not use a credential helper.

https://gitforwindows.org/



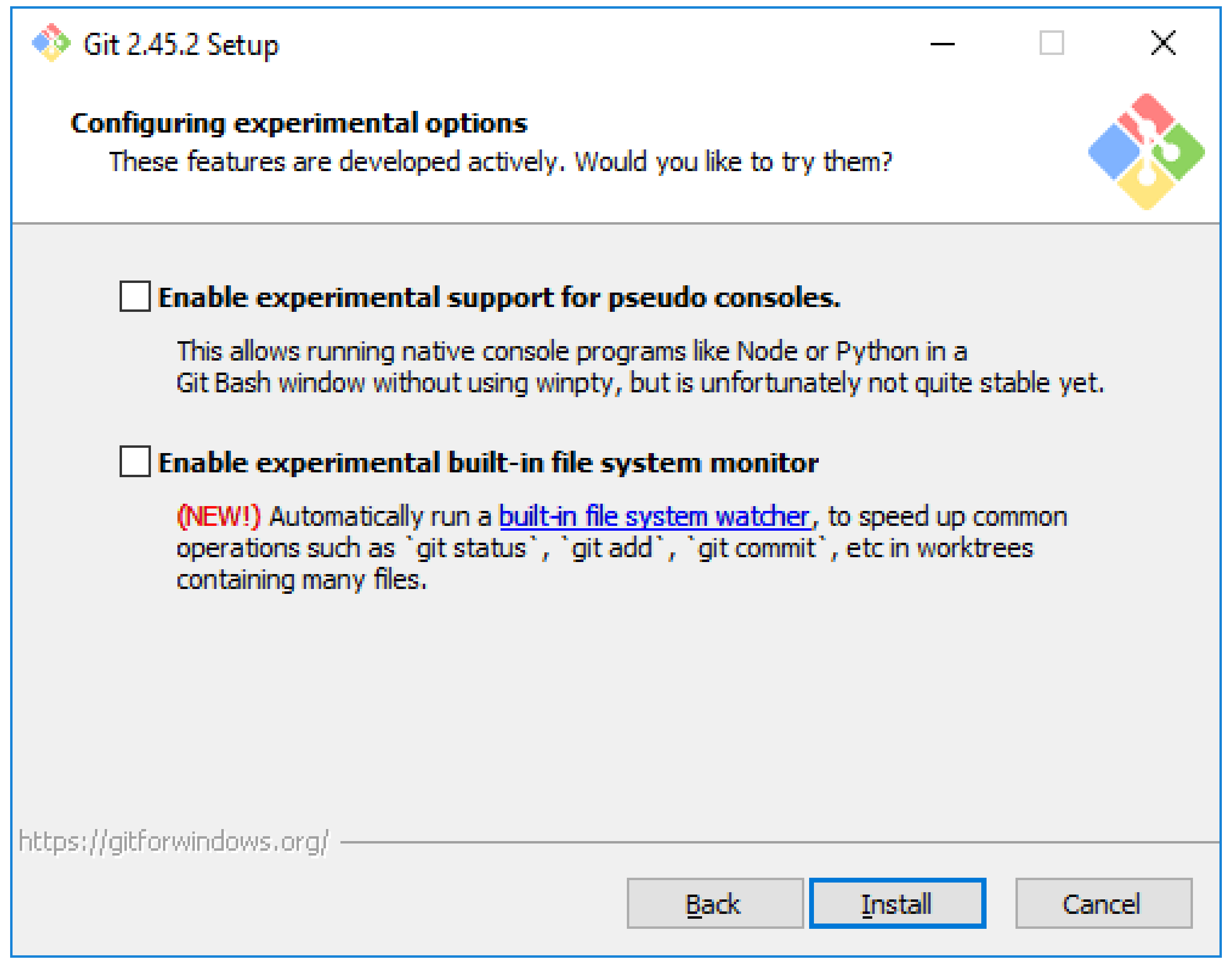
Git 2.45.2 Setup

Configuring extra options
Which features would you like to enable?

☑ **Enable file system caching**
File system data will be read in bulk and cached in memory for certain operations ("core.fscache" is set to "true"). This provides a significant performance boost.

☐ **Enable symbolic links**
Enable symbolic links (requires the SeCreateSymbolicLink permission).
Please note that existing repositories are unaffected by this setting.

https://gitforwindows.org/



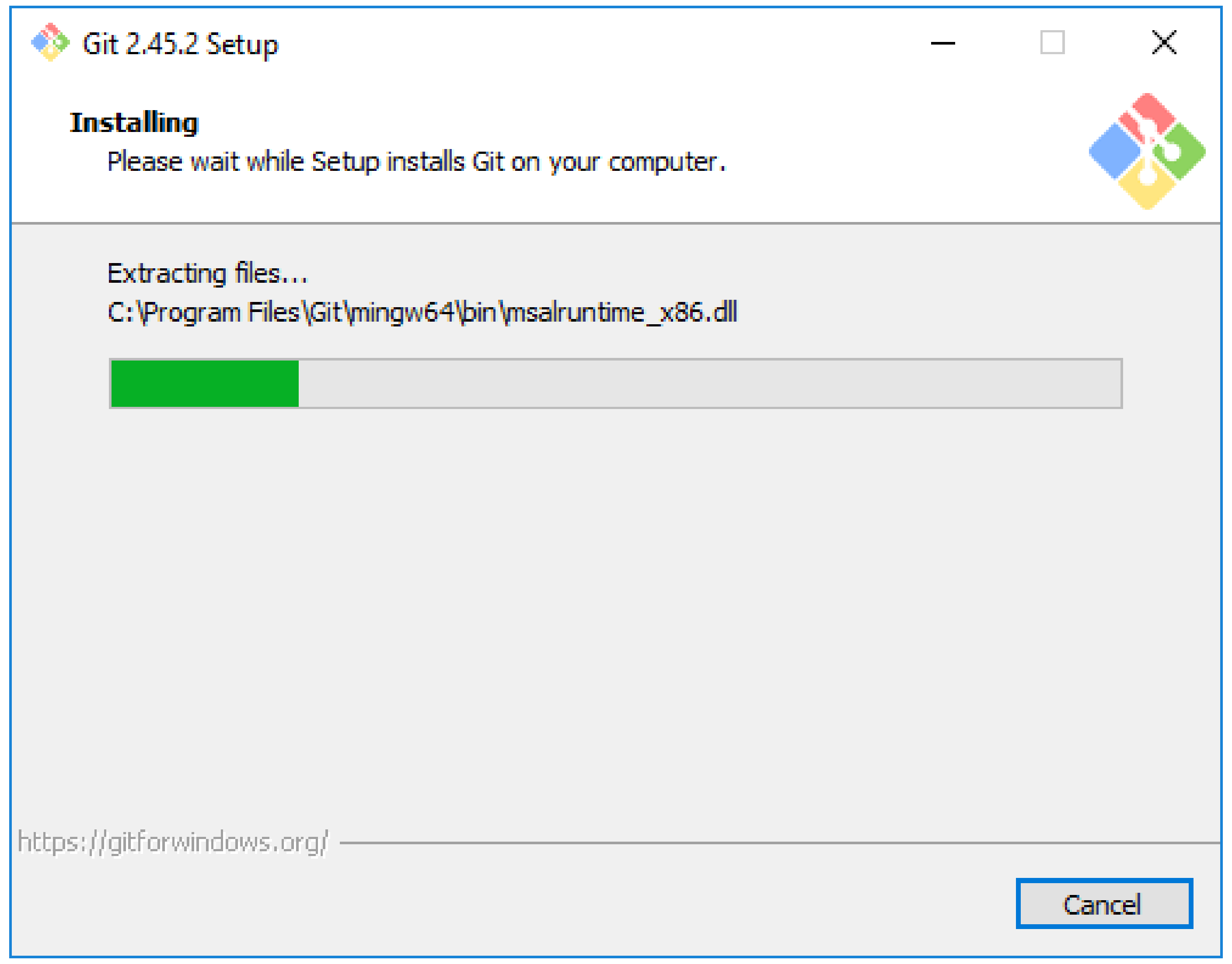
Git 2.45.2 Setup

Configuring experimental options
These features are developed actively. Would you like to try them?

☐ **Enable experimental support for pseudo consoles.**
This allows running native console programs like Node or Python in a Git Bash window without using winpty, but is unfortunately not quite stable yet.

☐ **Enable experimental built-in file system monitor**
(NEW!) Automatically run a built-in file system watcher, to speed up common operations such as `git status`, `git add`, `git commit`, etc in worktrees containing many files.

https://gitforwindows.org/

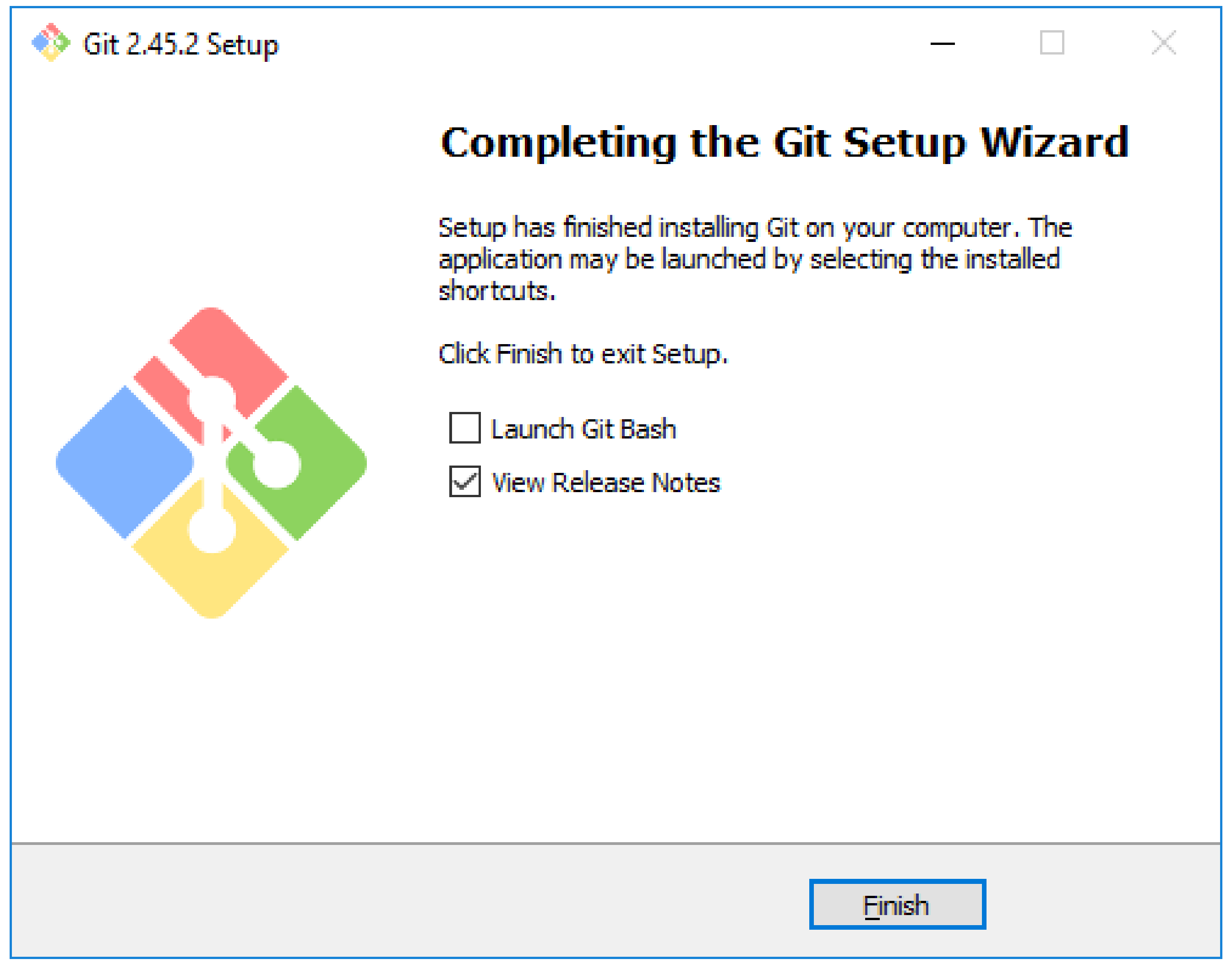


Git 2.45.2 Setup

Installing
Please wait while Setup installs Git on your computer.

Extracting files...
C:\Program Files\Git\mingw64\bin\msalruntime_x86.dll

https://gitforwindows.org/



Git 2.45.2 Setup

Completing the Git Setup Wizard

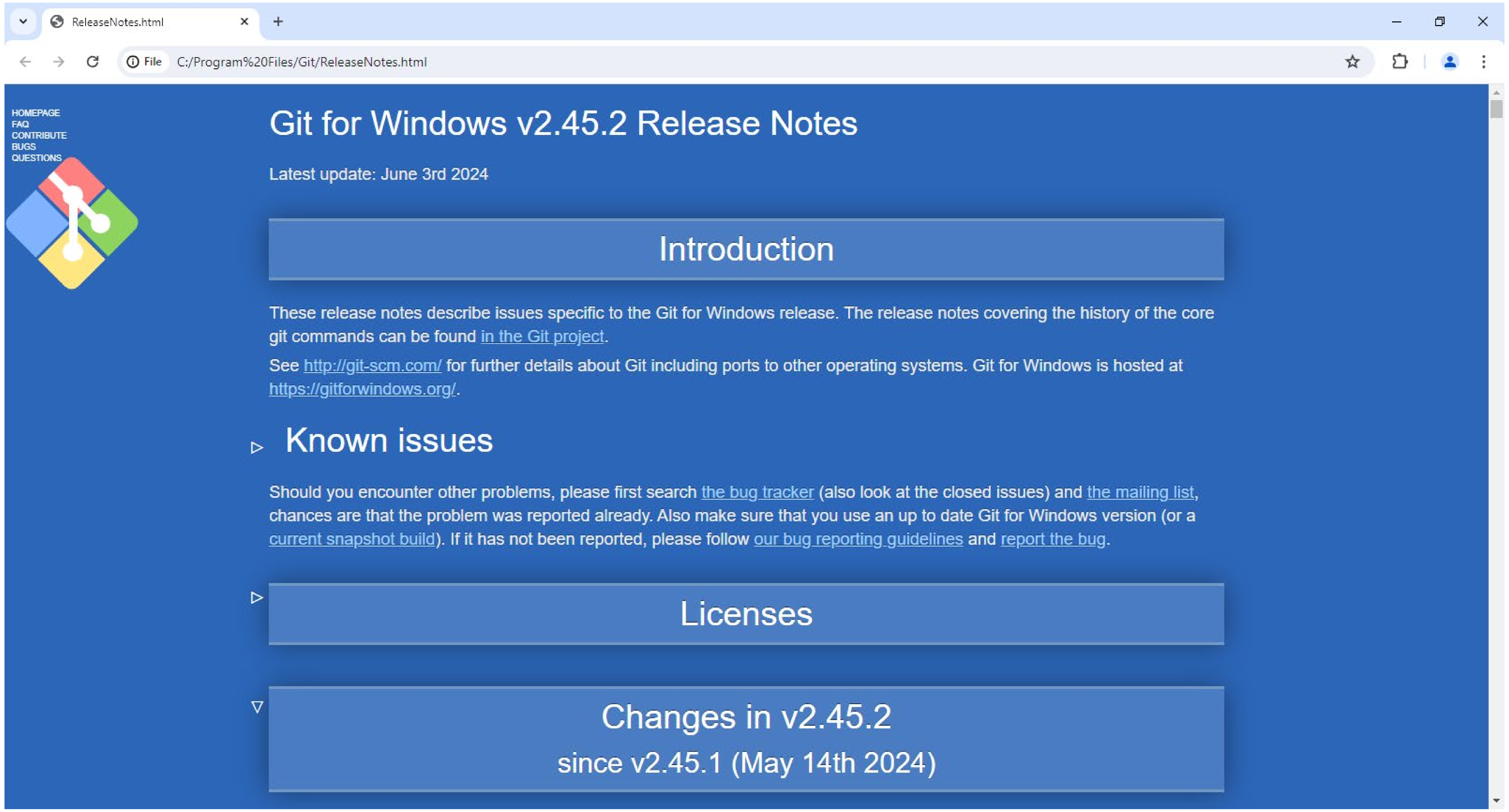
Setup has finished installing Git on your computer. The application may be launched by selecting the installed shortcuts.

Click Finish to exit Setup.

☐ Launch Git Bash
☑ View Release Notes

WEB AGE SOLUTIONS

1.7 Git for Windows Release Notes



Git for Windows v2.45.2 Release Notes

Latest update: June 3rd 2024

Introduction

These release notes describe issues specific to the Git for Windows release. The release notes covering the history of the core git commands can be found in the Git project.

See http://git-scm.com/ for further details about Git including ports to other operating systems. Git for Windows is hosted at https://gitforwindows.org/.

▷ Known issues

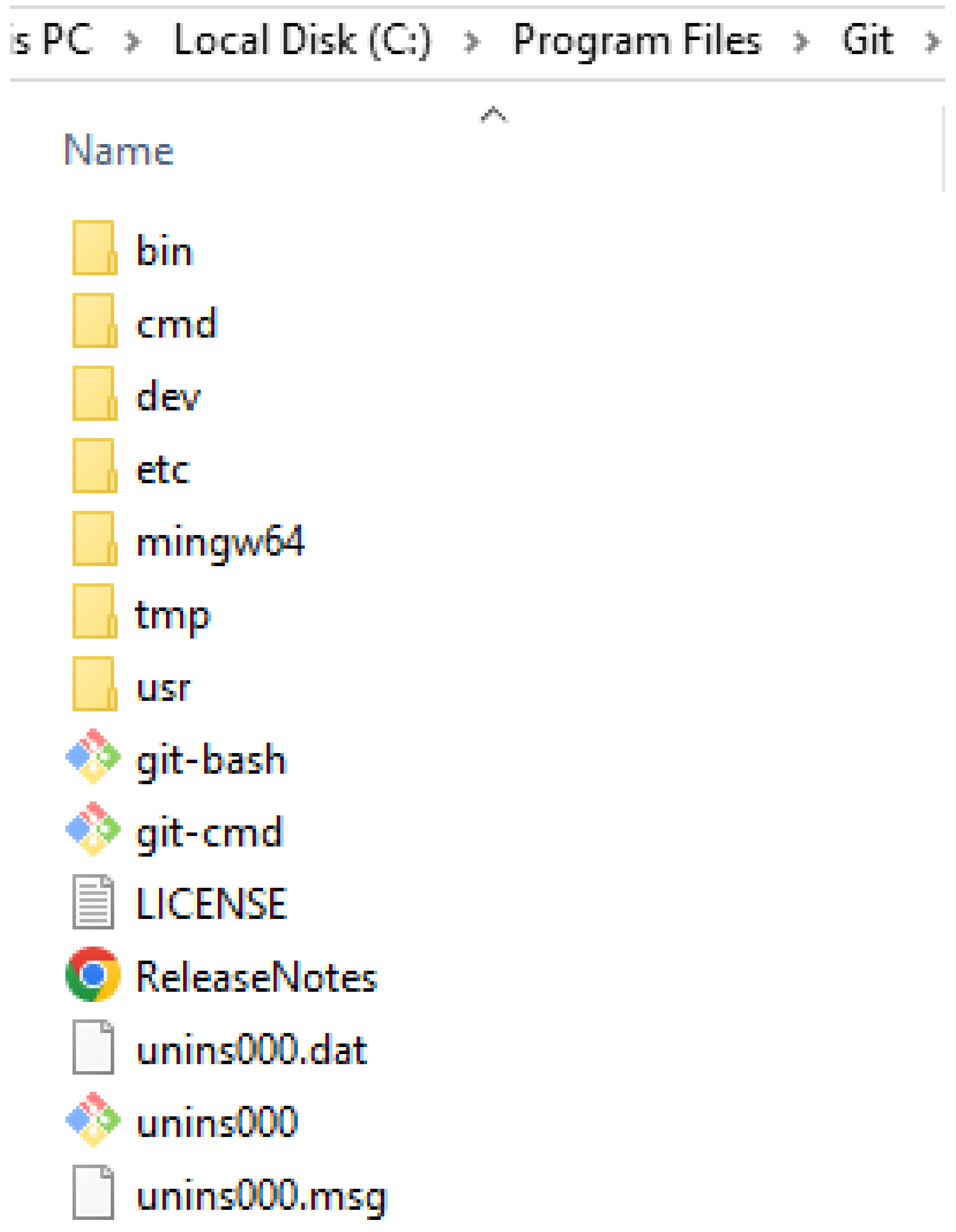
Should you encounter other problems, please first search the bug tracker (also look at the closed issues) and the mailing list, chances are that the problem was reported already. Also make sure that you use an up to date Git for Windows version (or a current snapshot build). If it has not been reported, please follow our bug reporting guidelines and report the bug.

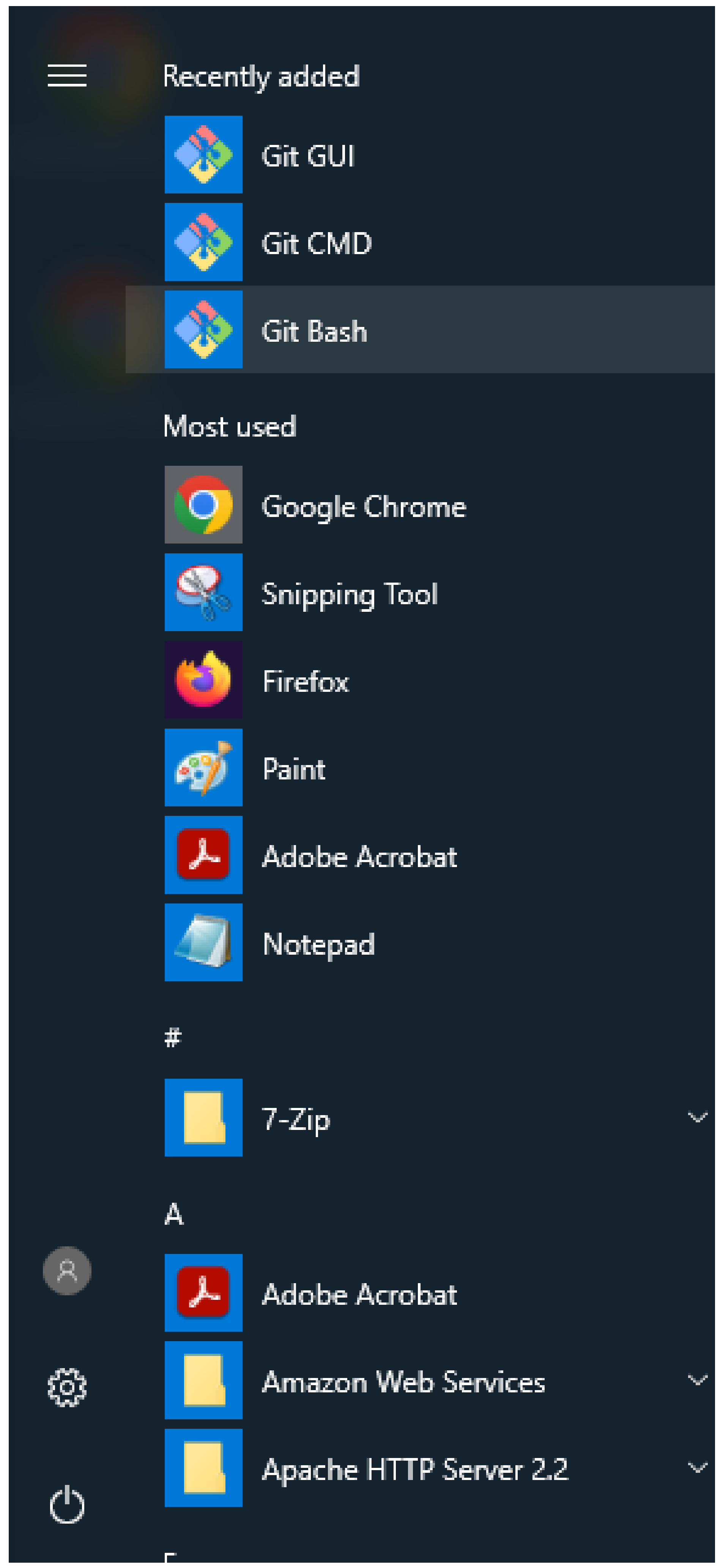
▷ Licenses

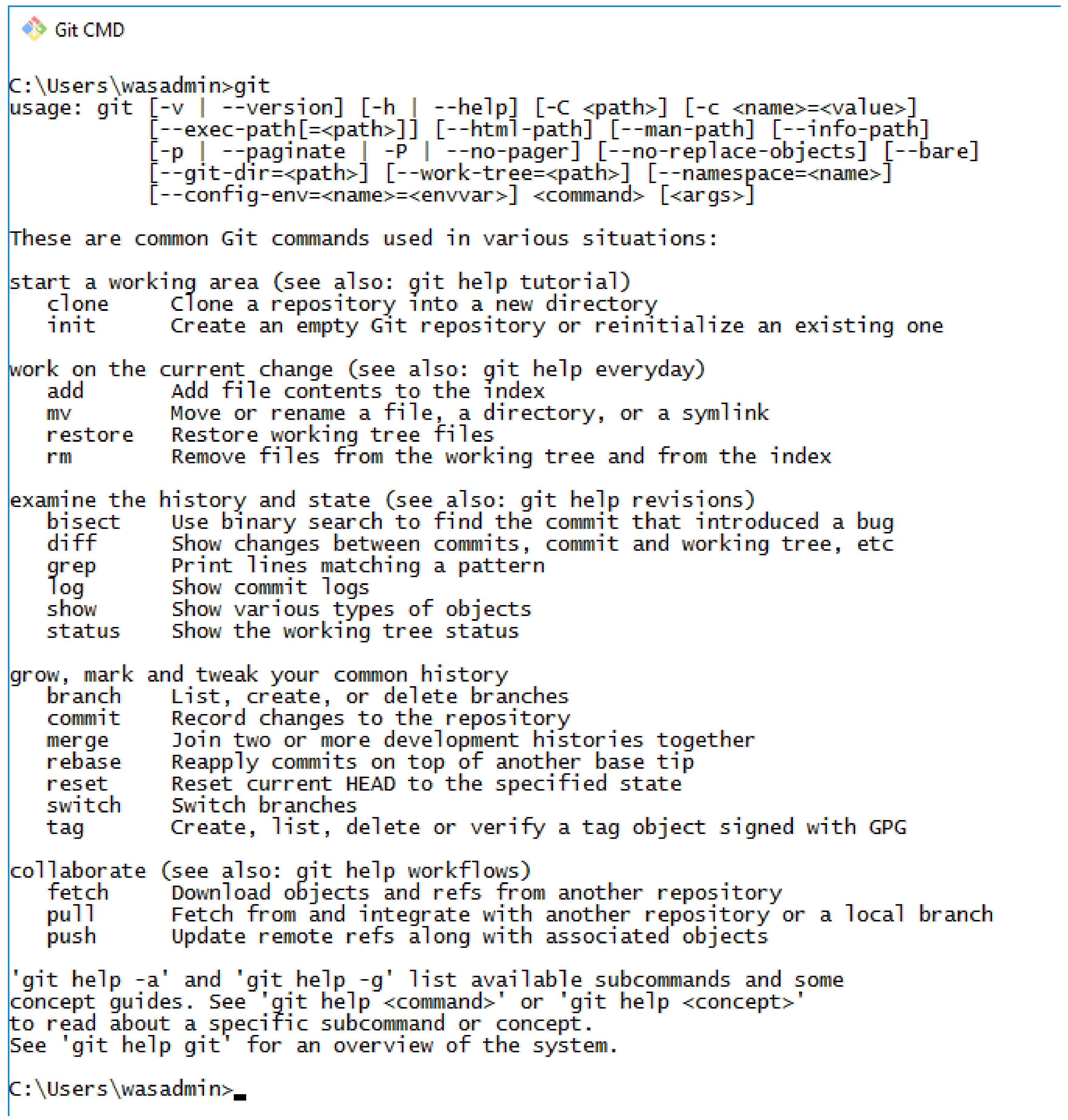
▽ Changes in v2.45.2
since v2.45.1 (May 14th 2024)

WEB AGE SOLUTIONS

1.7 Git Installation Details



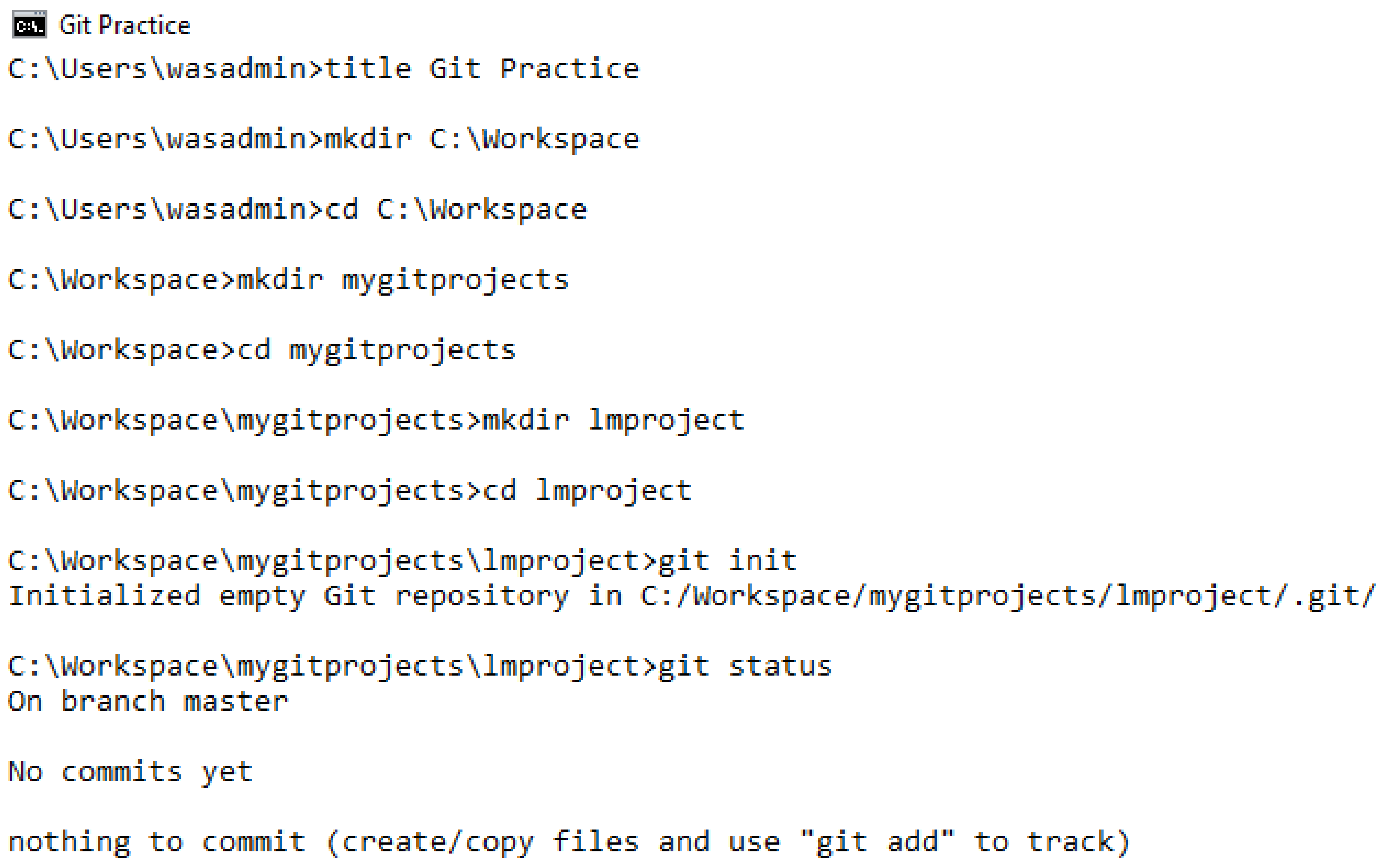




```
C:\Users\wasadmin>git
usage: git [-v | --version] [-h | --help] [-C <path>] [-c <name>=<value>]
           [--exec-path[=<path>]] [--html-path] [--man-path] [--info-path]
           [-p | --paginate | -P | --no-pager] [--no-replace-objects] [--bare]
           [--git-dir=<path>] [--work-tree=<path>] [--namespace=<name>]
           [--config-env=<name>=<envvar>] <command> [<args>]

These are common Git commands used in various situations:

start a working area (see also: git help tutorial)
   clone     Clone a repository into a new directory
   init      Create an empty Git repository or reinitialize an existing one

work on the current change (see also: git help everyday)
   add       Add file contents to the index
   mv        Move or rename a file, a directory, or a symlink
   restore   Restore working tree files
   rm        Remove files from the working tree and from the index

examine the history and state (see also: git help revisions)
   bisect    Use binary search to find the commit that introduced a bug
   diff      Show changes between commits, commit and working tree, etc
   grep      Print lines matching a pattern
   log       Show commit logs
   show      Show various types of objects
   status    Show the working tree status

grow, mark and tweak your common history
   branch    List, create, or delete branches
   commit    Record changes to the repository
   merge     Join two or more development histories together
   rebase    Reapply commits on top of another base tip
   reset     Reset current HEAD to the specified state
   switch    Switch branches
   tag       Create, list, delete or verify a tag object signed with GPG

collaborate (see also: git help workflows)
   fetch     Download objects and refs from another repository
   pull      Fetch from and integrate with another repository or a local branch
   push      Update remote refs along with associated objects

'git help -a' and 'git help -g' list available subcommands and some
concept guides. See 'git help <command>' or 'git help <concept>'
to read about a specific subcommand or concept.
See 'git help git' for an overview of the system.

C:\Users\wasadmin>
```

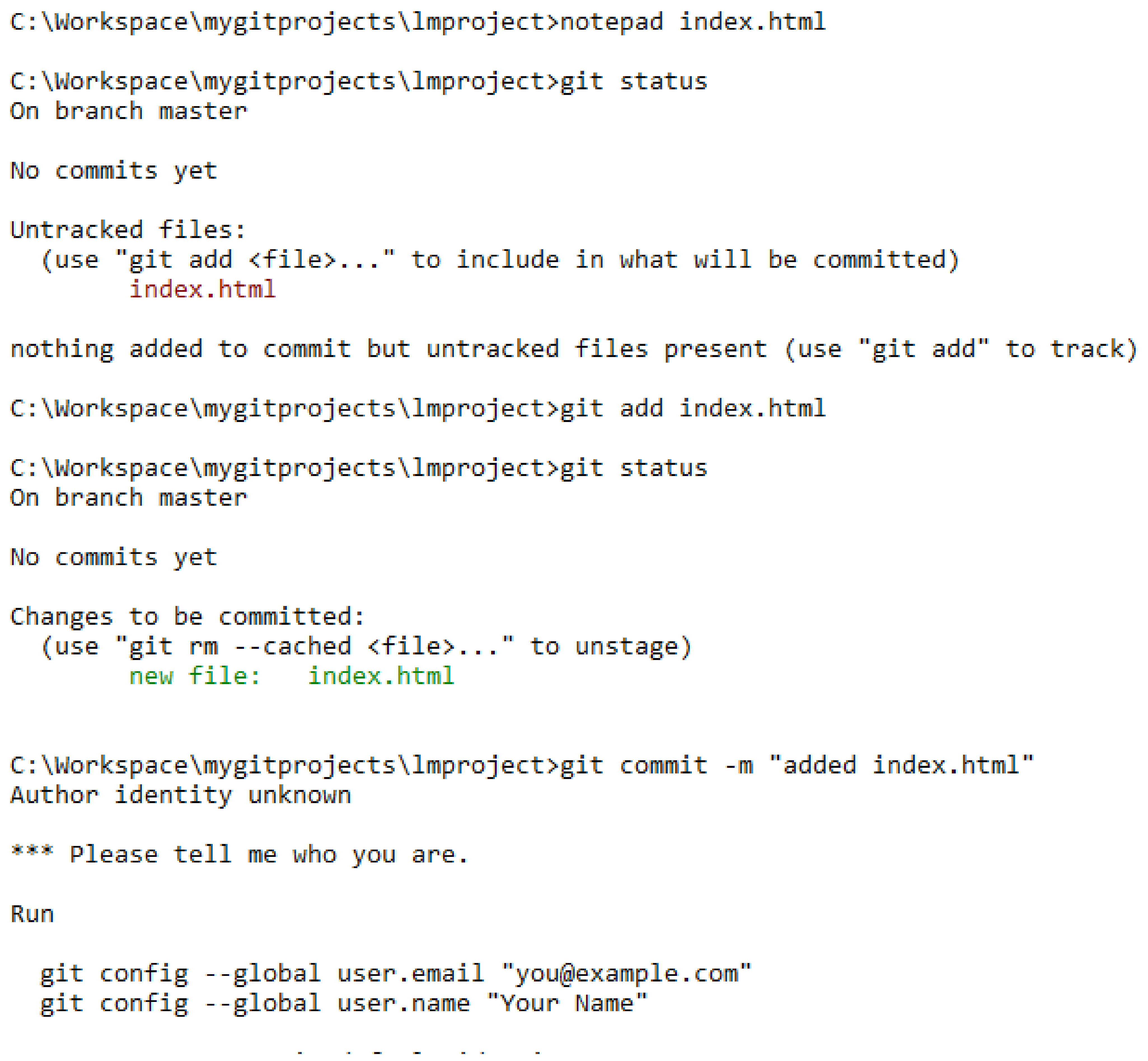
WEB AGE SOLUTIONS

1.8 Class Exercise 1



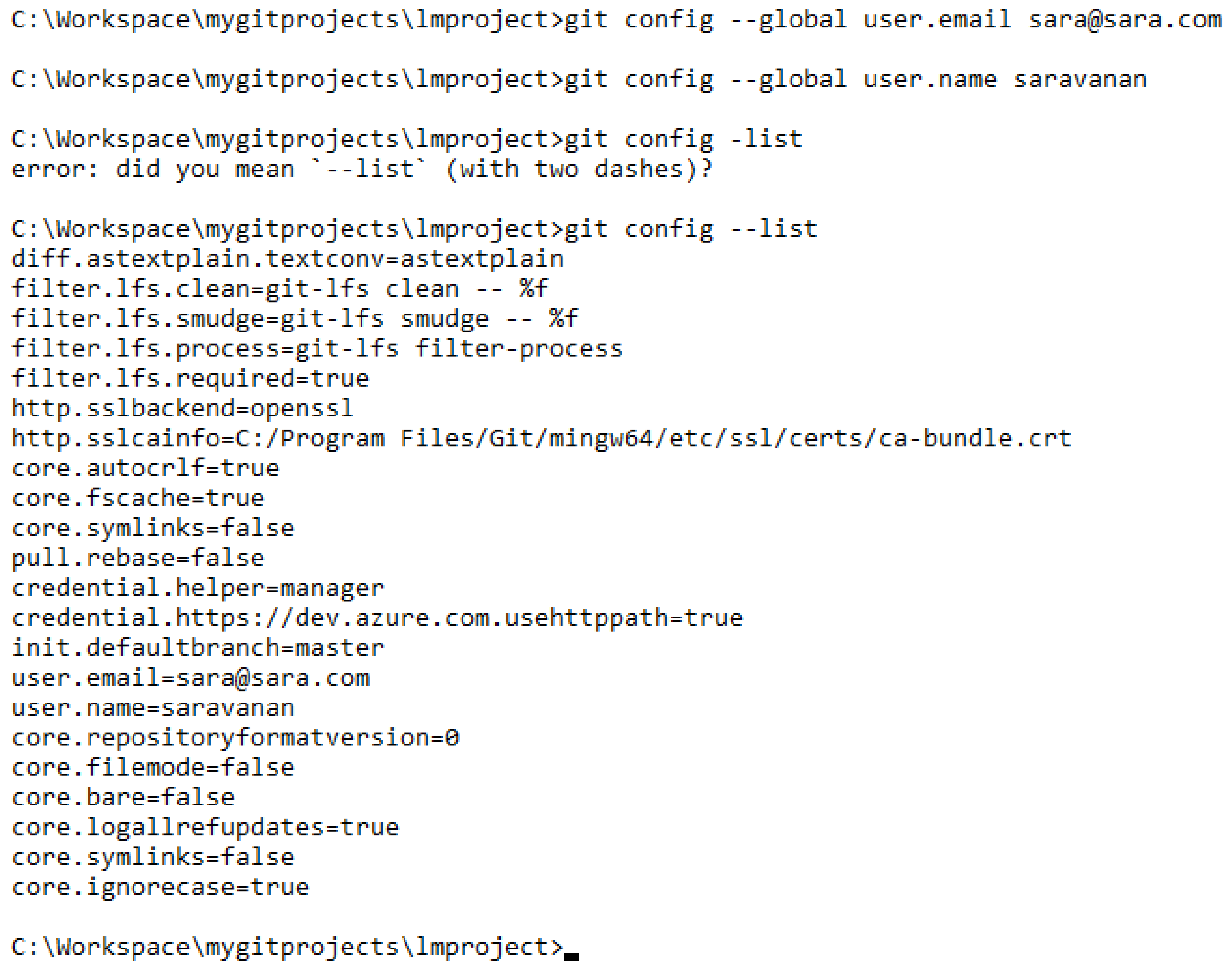
Git Practice

```
C:\Users\wasadmin>title Git Practice

C:\Users\wasadmin>mkdir C:\Workspace

C:\Users\wasadmin>cd C:\Workspace

C:\Workspace>mkdir mygitprojects

C:\Workspace>cd mygitprojects

C:\Workspace\mygitprojects>mkdir lmproject

C:\Workspace\mygitprojects>cd lmproject

C:\Workspace\mygitprojects\lmproject>git init
Initialized empty Git repository in C:/Workspace/mygitprojects/lmproject/.git/

C:\Workspace\mygitprojects\lmproject>git status
On branch master

No commits yet

nothing to commit (create/copy files and use "git add" to track)
```

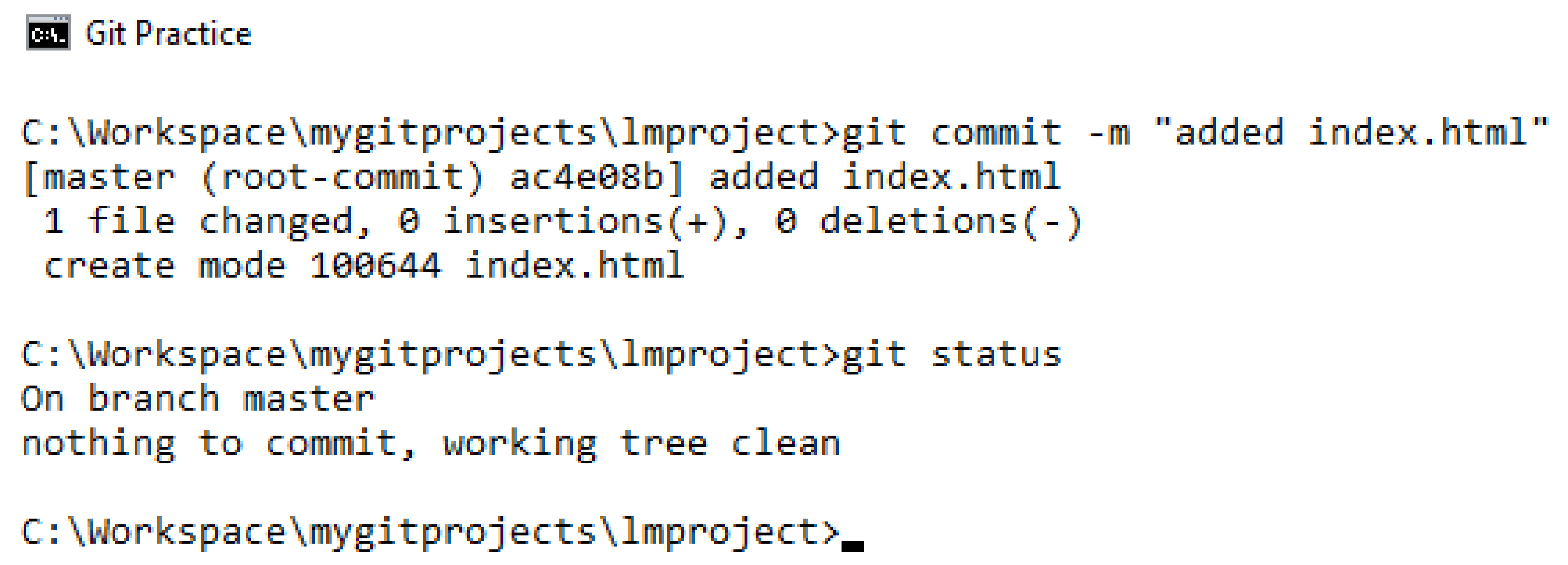


```
C:\Workspace\mygitprojects\lmproject>notepad index.html

C:\Workspace\mygitprojects\lmproject>git status
On branch master

No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        index.html

nothing added to commit but untracked files present (use "git add" to track)

C:\Workspace\mygitprojects\lmproject>git add index.html

C:\Workspace\mygitprojects\lmproject>git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
        new file:   index.html


C:\Workspace\mygitprojects\lmproject>git commit -m "added index.html"
Author identity unknown

*** Please tell me who you are.

Run

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

1.8 Class Exercise 2

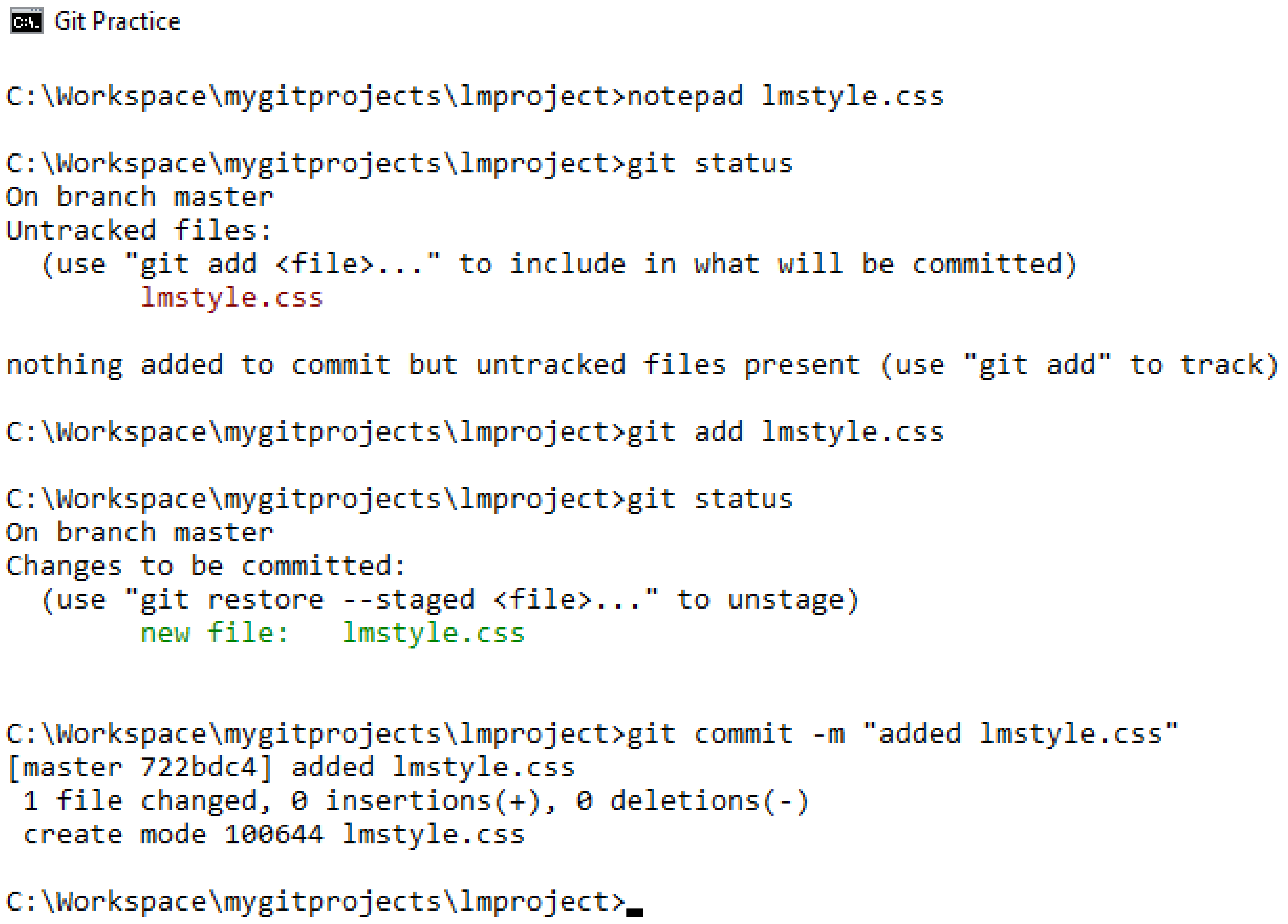


```
C:\Workspace\mygitprojects\lmproject>git config --global user.email sara@sara.com

C:\Workspace\mygitprojects\lmproject>git config --global user.name saravanan

C:\Workspace\mygitprojects\lmproject>git config -list
error: did you mean `--list` (with two dashes)?

C:\Workspace\mygitprojects\lmproject>git config --list
diff.astextplain.textconv=astextplain
filter.lfs.clean=git-lfs clean -- %f
filter.lfs.smudge=git-lfs smudge -- %f
filter.lfs.process=git-lfs filter-process
filter.lfs.required=true
http.sslbackend=openssl
http.sslcainfo=C:/Program Files/Git/mingw64/etc/ssl/certs/ca-bundle.crt
core.autocrlf=true
core.fscache=true
core.symlinks=false
pull.rebase=false
credential.helper=manager
credential.https://dev.azure.com.usehttppath=true
init.defaultbranch=master
user.email=sara@sara.com
user.name=saravanan
core.repositoryformatversion=0
core.filemode=false
core.bare=false
core.logallrefupdates=true
core.symlinks=false
core.ignorecase=true

C:\Workspace\mygitprojects\lmproject>
```



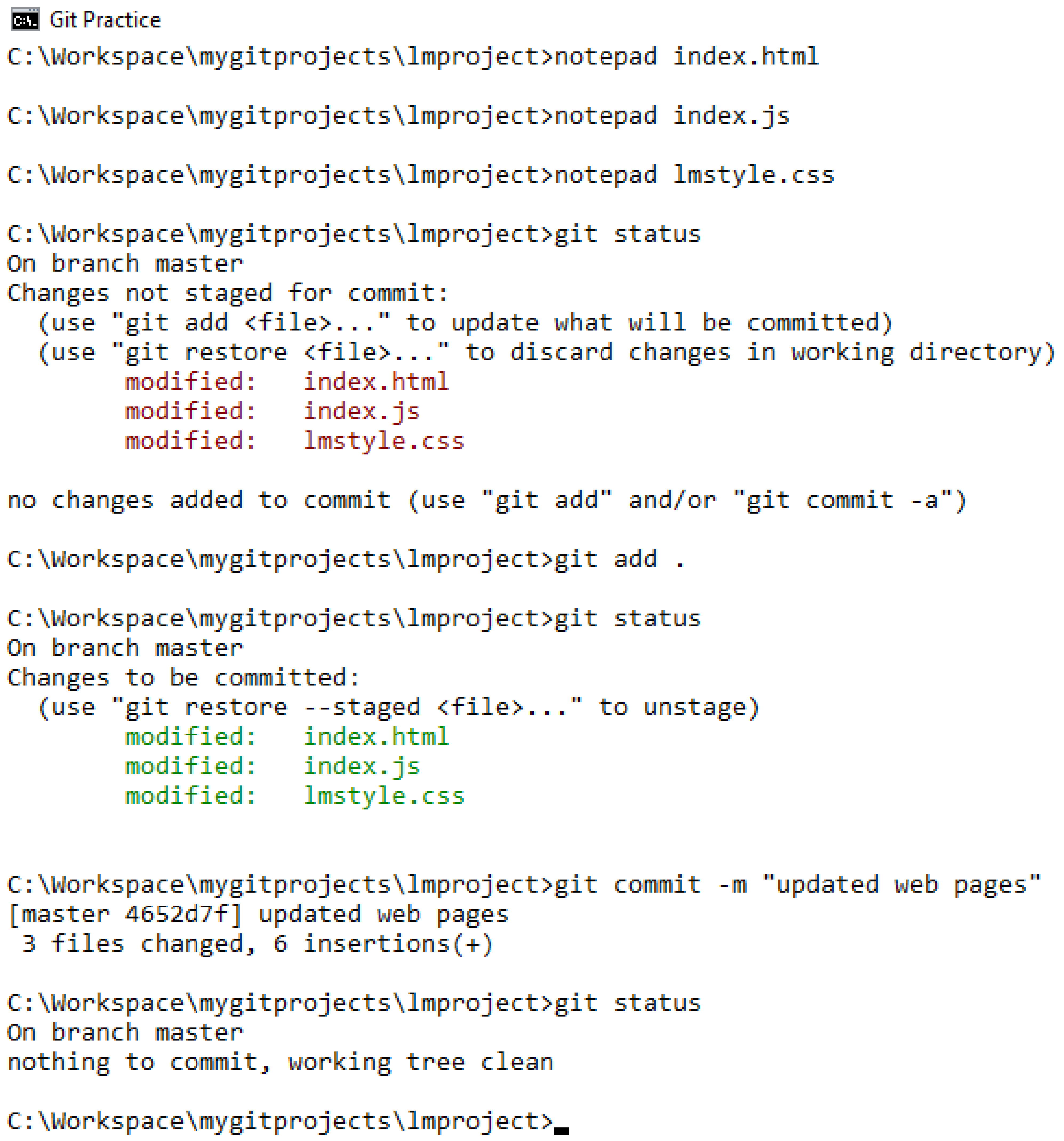
```
C:\Workspace\mygitprojects\lmproject>git commit -m "added index.html"
[master (root-commit) ac4e08b] added index.html
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 index.html

C:\Workspace\mygitprojects\lmproject>git status
On branch master
nothing to commit, working tree clean

C:\Workspace\mygitprojects\lmproject>
```

1.8 Class Exercise 3



Git Practice

```
C:\Workspace\mygitprojects\lmproject>notepad lmstyle.css

C:\Workspace\mygitprojects\lmproject>git status
On branch master
Untracked files:
  (use "git add <file>..." to include in what will be committed)
        lmstyle.css

nothing added to commit but untracked files present (use "git add" to track)

C:\Workspace\mygitprojects\lmproject>git add lmstyle.css

C:\Workspace\mygitprojects\lmproject>git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        new file:   lmstyle.css


C:\Workspace\mygitprojects\lmproject>git commit -m "added lmstyle.css"
[master 722bdc4] added lmstyle.css
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 lmstyle.css

C:\Workspace\mygitprojects\lmproject>
```

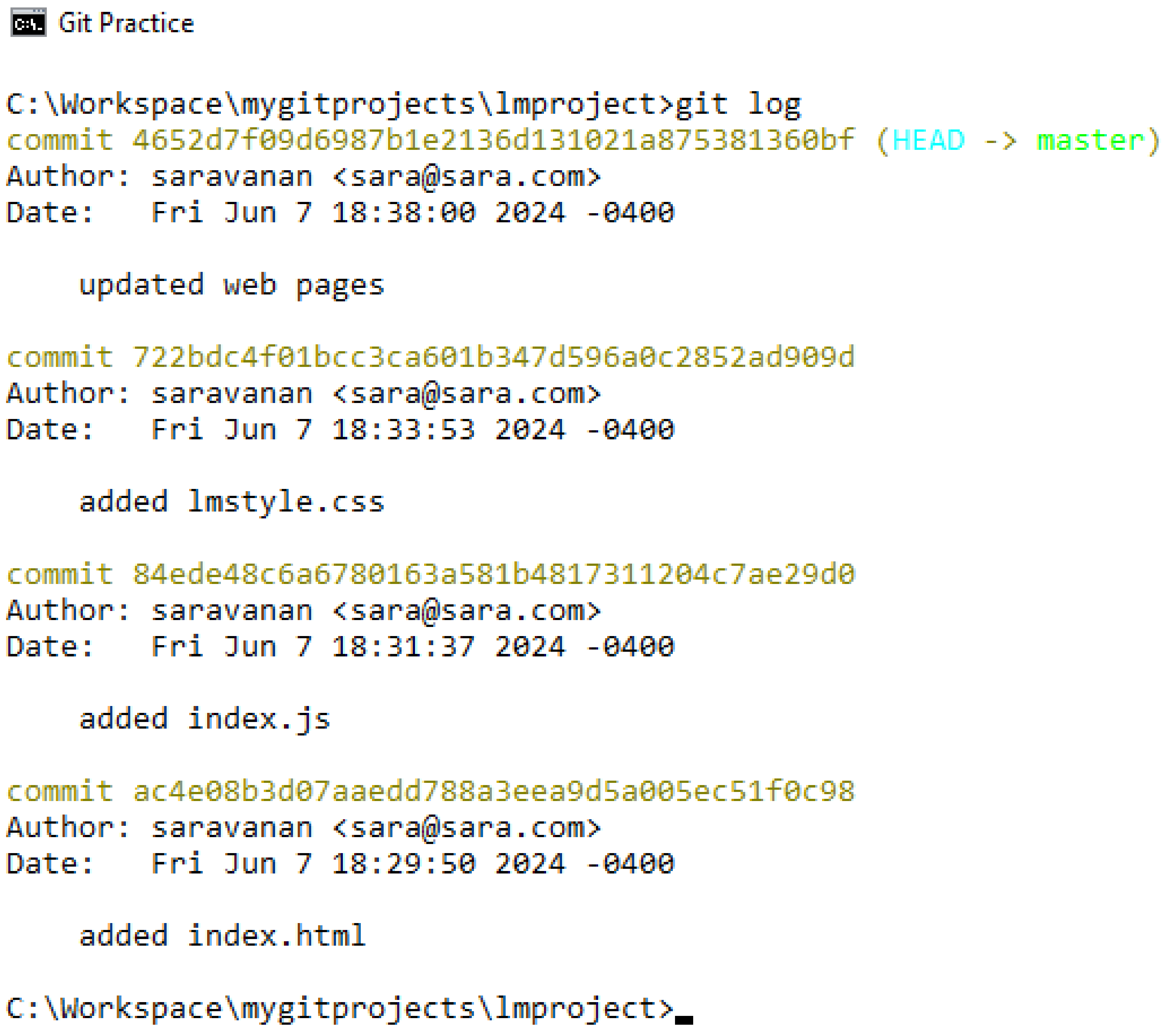


Git Practice

```
C:\Workspace\mygitprojects\lmproject>notepad index.html

C:\Workspace\mygitprojects\lmproject>notepad index.js

C:\Workspace\mygitprojects\lmproject>notepad lmstyle.css

C:\Workspace\mygitprojects\lmproject>git status
On branch master
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   index.html
        modified:   index.js
        modified:   lmstyle.css

no changes added to commit (use "git add" and/or "git commit -a")

C:\Workspace\mygitprojects\lmproject>git add .

C:\Workspace\mygitprojects\lmproject>git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        modified:   index.html
        modified:   index.js
        modified:   lmstyle.css


C:\Workspace\mygitprojects\lmproject>git commit -m "updated web pages"
[master 4652d7f] updated web pages
 3 files changed, 6 insertions(+)

C:\Workspace\mygitprojects\lmproject>git status
On branch master
nothing to commit, working tree clean

C:\Workspace\mygitprojects\lmproject>
```

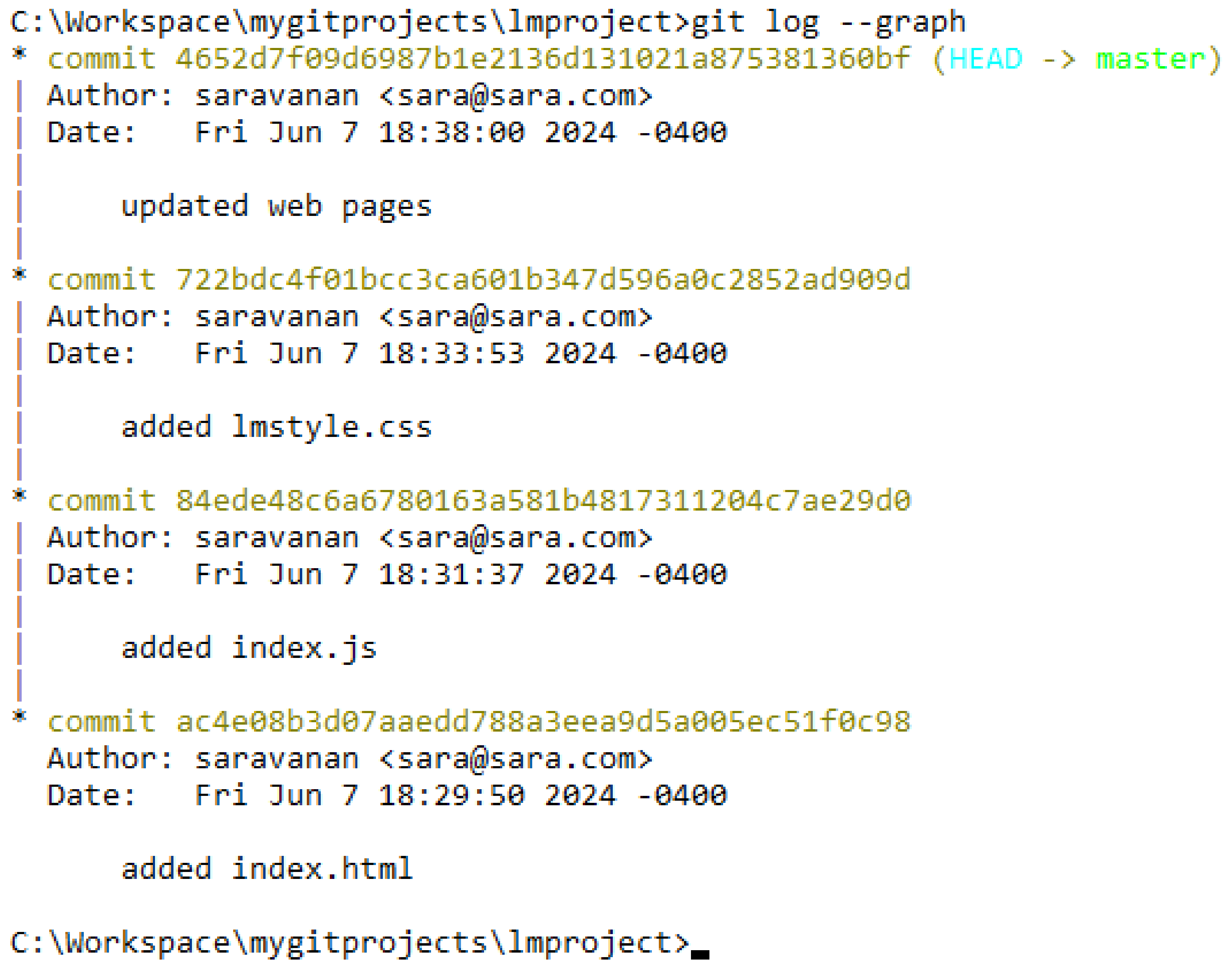
WEB AGE SOLUTIONS

1.8 Class Exercise 4



Git Practice

```
C:\Workspace\mygitprojects\lmproject>git log
commit 4652d7f09d6987b1e2136d131021a875381360bf (HEAD -> master)
Author: saravanan <sara@sara.com>
Date:   Fri Jun 7 18:38:00 2024 -0400

    updated web pages

commit 722bdc4f01bcc3ca601b347d596a0c2852ad909d
Author: saravanan <sara@sara.com>
Date:   Fri Jun 7 18:33:53 2024 -0400

    added lmstyle.css

commit 84ede48c6a6780163a581b4817311204c7ae29d0
Author: saravanan <sara@sara.com>
Date:   Fri Jun 7 18:31:37 2024 -0400

    added index.js

commit ac4e08b3d07aaedd788a3eea9d5a005ec51f0c98
Author: saravanan <sara@sara.com>
Date:   Fri Jun 7 18:29:50 2024 -0400

    added index.html

C:\Workspace\mygitprojects\lmproject>
```



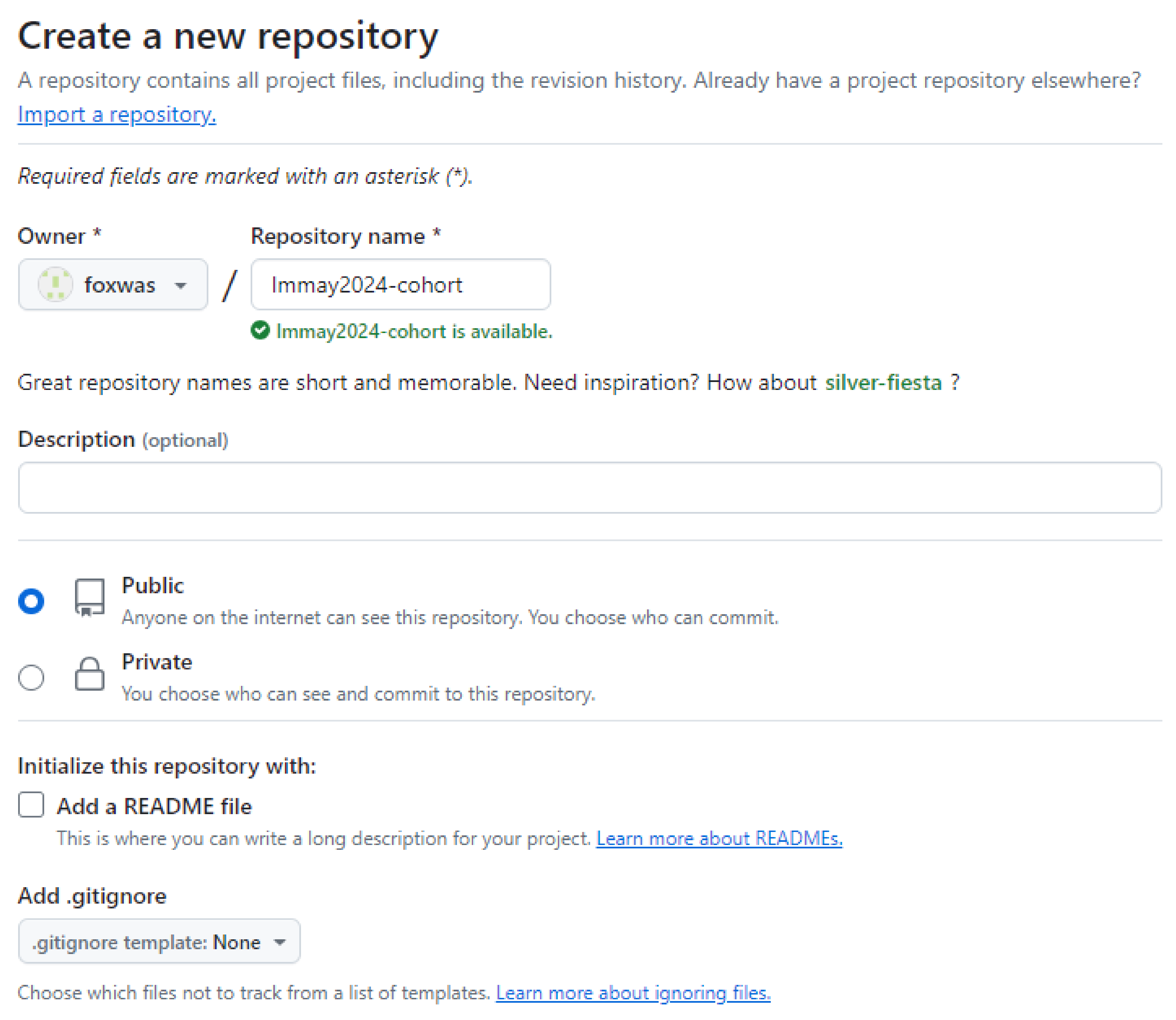
```
C:\Workspace\mygitprojects\lmproject>git log --graph
* commit 4652d7f09d6987b1e2136d131021a875381360bf (HEAD -> master)
| Author: saravanan <sara@sara.com>
| Date:   Fri Jun 7 18:38:00 2024 -0400
|
|     updated web pages
|
* commit 722bdc4f01bcc3ca601b347d596a0c2852ad909d
| Author: saravanan <sara@sara.com>
| Date:   Fri Jun 7 18:33:53 2024 -0400
|
|     added lmstyle.css
|
* commit 84ede48c6a6780163a581b4817311204c7ae29d0
| Author: saravanan <sara@sara.com>
| Date:   Fri Jun 7 18:31:37 2024 -0400
|
|     added index.js
|
* commit ac4e08b3d07aaedd788a3eea9d5a005ec51f0c98
  Author: saravanan <sara@sara.com>
  Date:   Fri Jun 7 18:29:50 2024 -0400

      added index.html

C:\Workspace\mygitprojects\lmproject>
```

1.8 Class Exercise 5 Remote Repository – Git Hub

1. Create a personal GitHub Account
   https://github.com/signup

2. Login to your personal GitHub Account

3. Create a GitHub Repository
   - Owner ____
   - Repository name
     - lmmay2024-cohort
   - Description
   - Public
   - Add a README file
   - Add .gitignore



Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? Import a repository.

Required fields are marked with an asterisk (*).

Owner *: foxwas / Repository name *: lmmay2024-cohort

lmmay2024-cohort is available.

Great repository names are short and memorable. Need inspiration? How about silver-fiesta ?

Description (optional)

[✓ Public] Anyone on the internet can see this repository. You choose who can commit.

Private — You choose who can see and commit to this repository.

Initialize this repository with:

☐ Add a README file — This is where you can write a long description for your project. Learn more about READMEs.

Add .gitignore

.gitignore template: None

Choose which files not to track from a list of templates. Learn more about ignoring files.

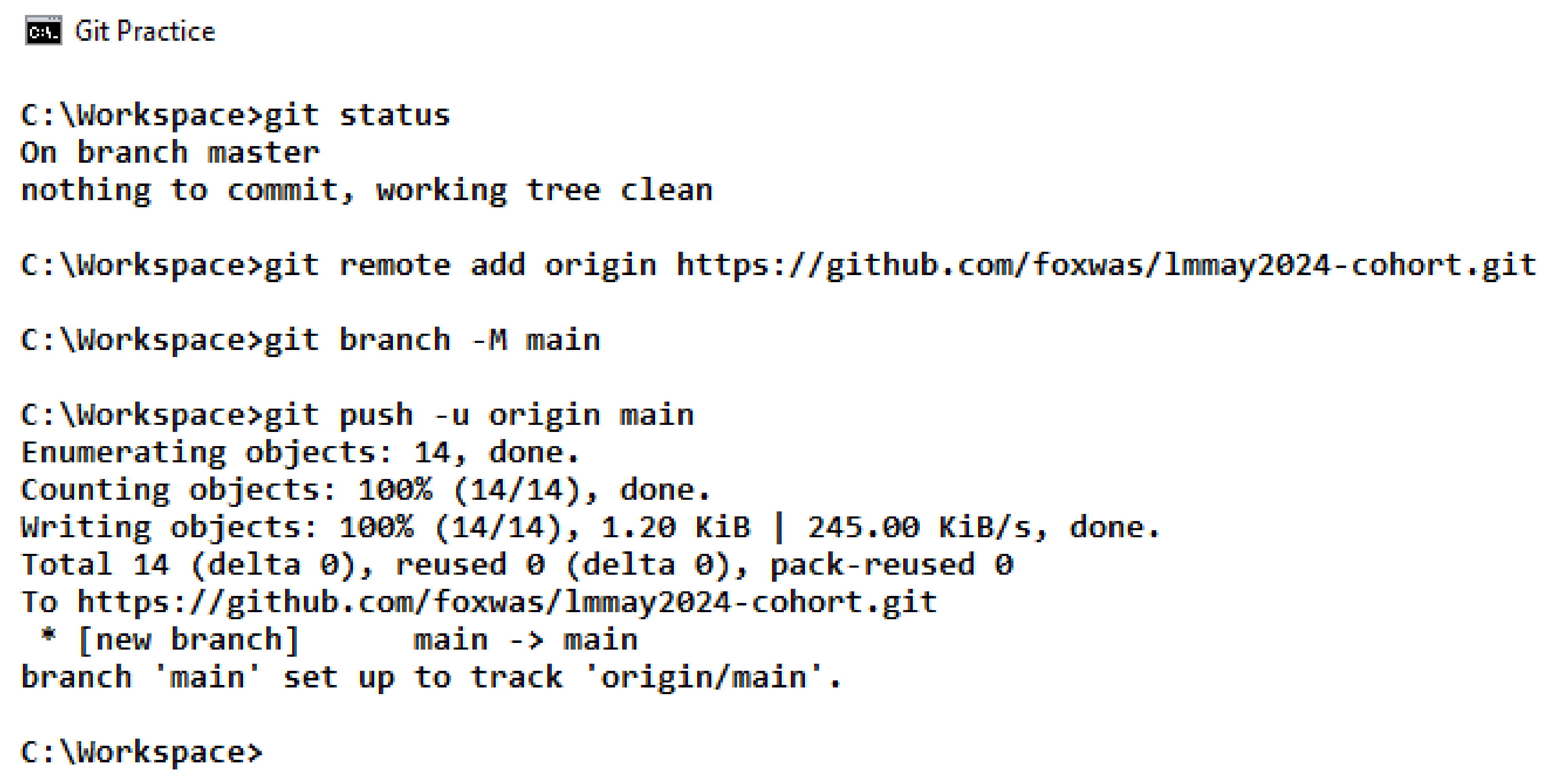
1.9 Add Remote Repository to Local Repo Configuration

- To create a new repository on the command line in your computer and add your repo contents to GitHub remote repository

```
echo "# lmmay2024-cohort" >> README.md
git init
git add README.md
git commit -m "first commit"
git branch -M main
git remote add origin https://github.com/foxwas/lmmay2024-cohort.git
git push -u origin main
```

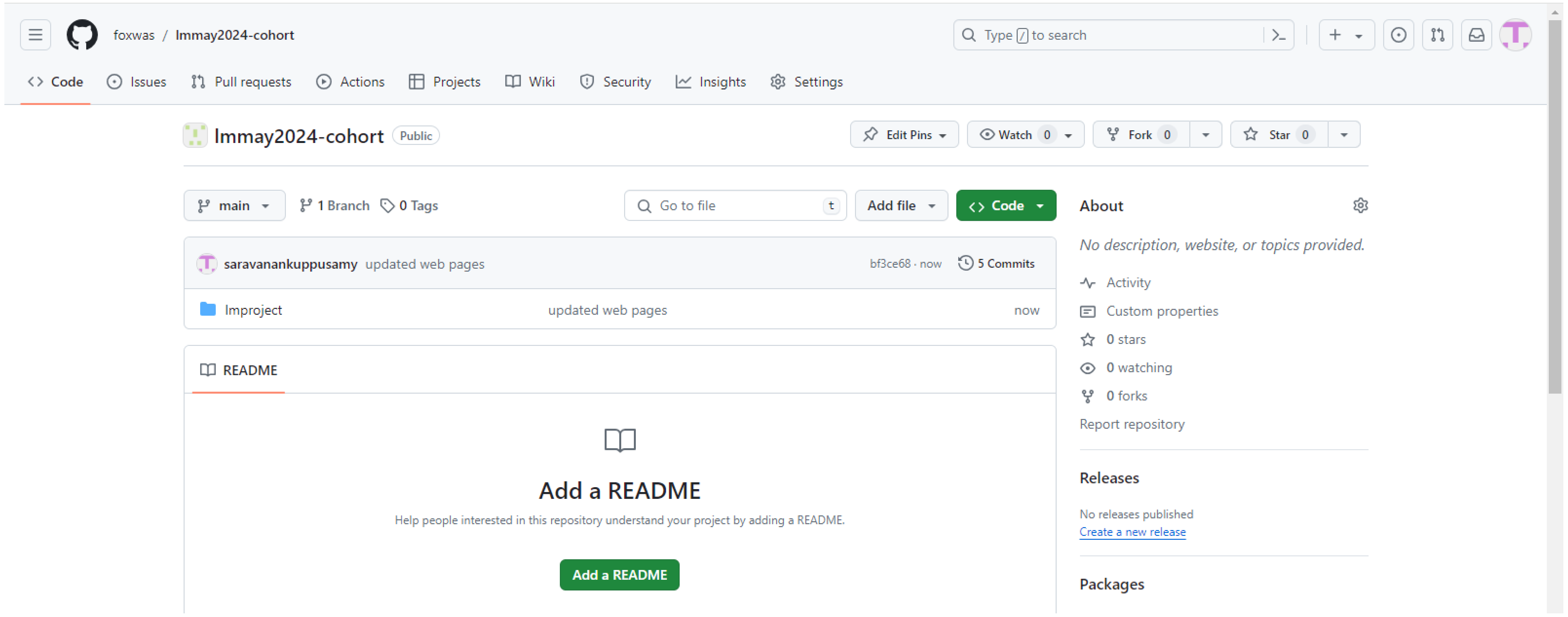
- To push an existing repository in your computer from the command line, add repo contents to GitHub remote repository

```
git remote add origin https://github.com/foxwas/lmmay2024-cohort.git
git branch -M main
git push -u origin main
```


Git Practice

```
C:\Workspace>git status
On branch master
nothing to commit, working tree clean

C:\Workspace>git remote add origin https://github.com/foxwas/lmmay2024-cohort.git

C:\Workspace>git branch -M main

C:\Workspace>git push -u origin main
Enumerating objects: 14, done.
Counting objects: 100% (14/14), done.
Writing objects: 100% (14/14), 1.20 KiB | 245.00 KiB/s, done.
Total 14 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/foxwas/lmmay2024-cohort.git
 * [new branch]      main -> main
branch 'main' set up to track 'origin/main'.

C:\Workspace>
```

1.10 Local Repo - Remote Repository Synchronized



foxwas / lmmay2024-cohort

lmmay2024-cohort Public

main · 1 Branch · 0 Tags

saravanankuppusamy updated web pages · bf3ce68 · now · 5 Commits

lmproject	updated web pages	now

README

Add a README

Help people interested in this repository understand your project by adding a README.

About

No description, website, or topics provided.

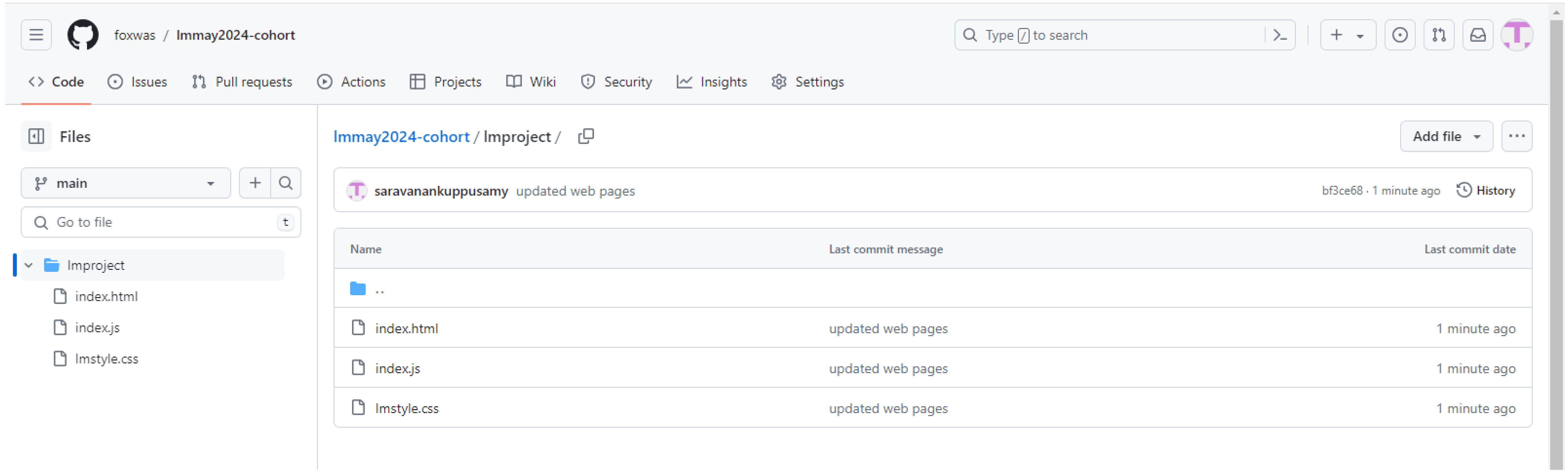
0 stars · 0 watching · 0 forks

Releases

No releases published

Create a new release

Packages

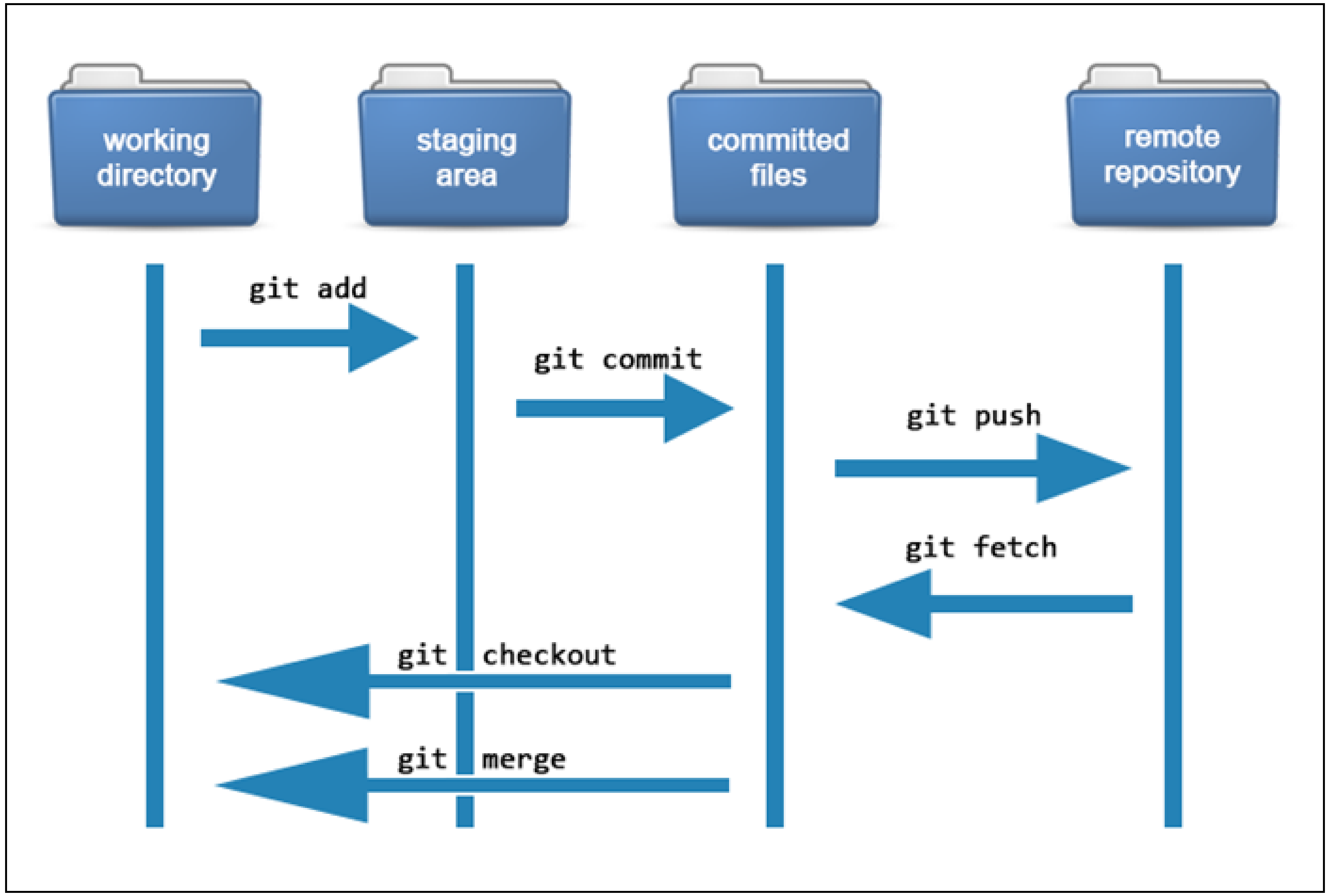


lmmay2024-cohort / lmproject /

saravanankuppusamy updated web pages · bf3ce68 · 1 minute ago · History

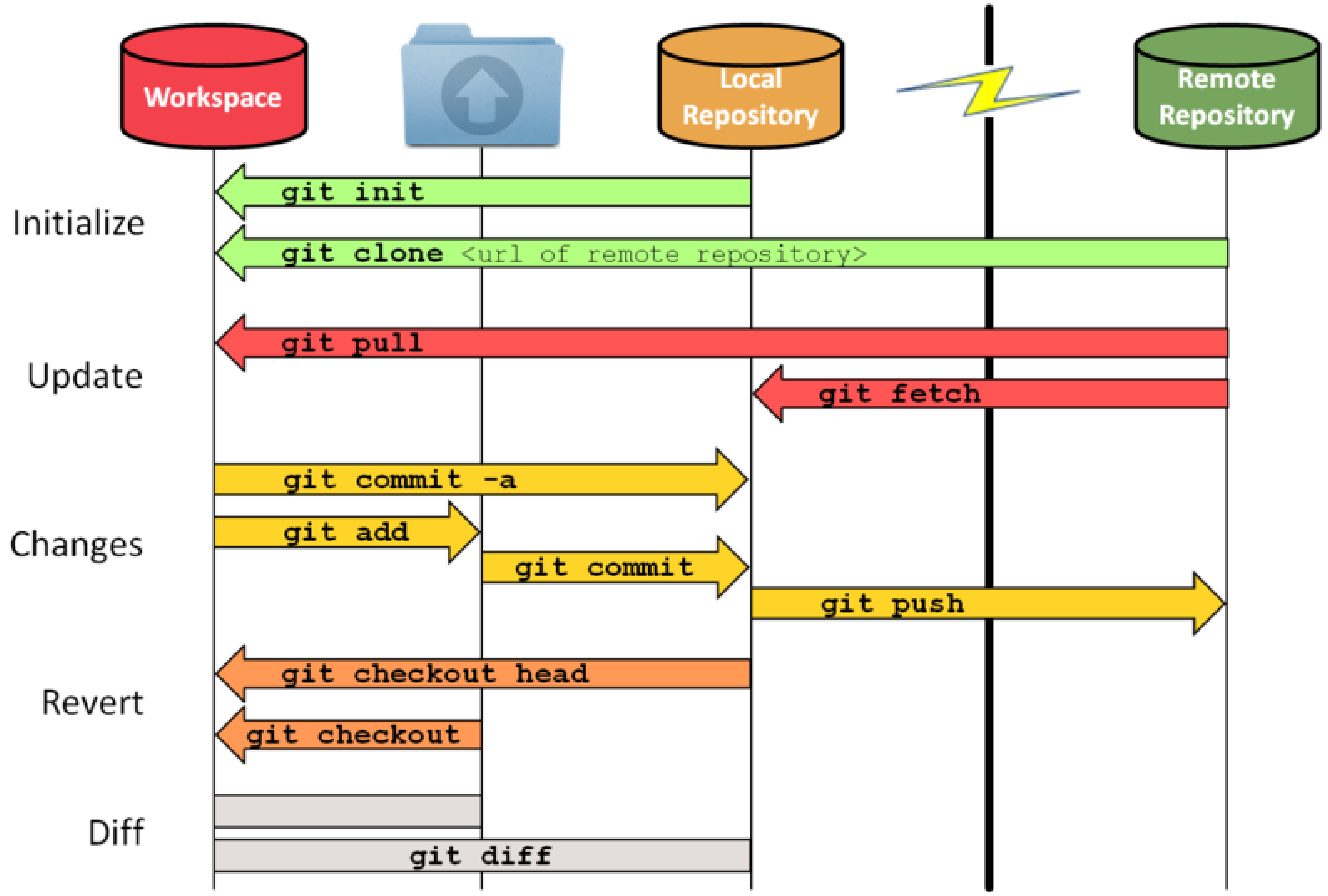
Name	Last commit message	Last commit date
..		
index.html	updated web pages	1 minute ago
index.js	updated web pages	1 minute ago
lmstyle.css	updated web pages	1 minute ago

1.11 Git Overview – Illustration 1



https://phoenixnap.com/kb/how-git-works

1.11 Git Overview – Illustration 2



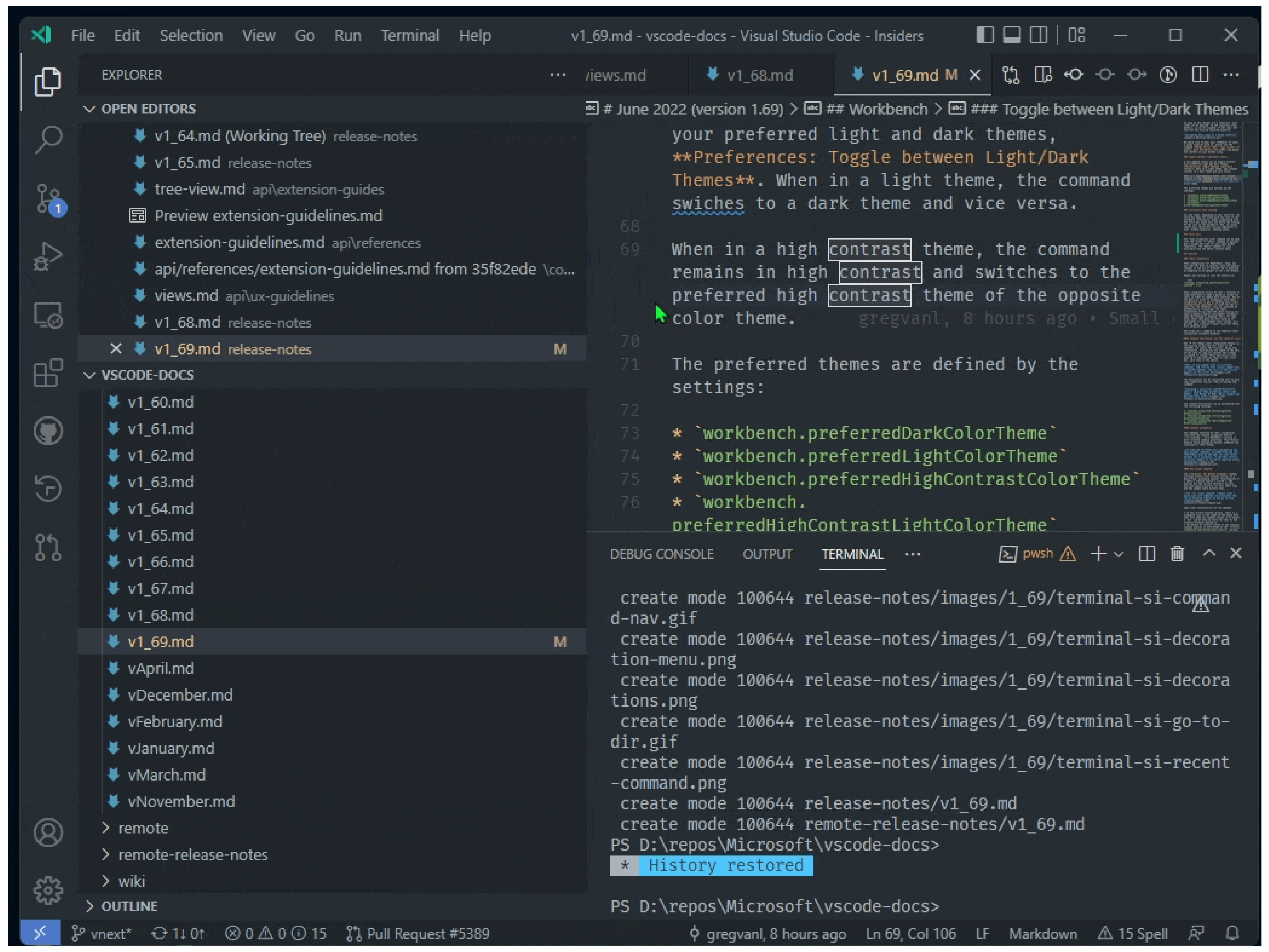
https://medium.com/analytics-vidhya/git-most-frequently-used-commands-9df9f200c235

WEB AGE SOLUTIONS

IDEs Visual Studio Code and IntelliJ

1.1 Visual Studio Code

- To be a strong developer you need to be comfortable with your integrated development environment (IDE).
- An integrated development environment (IDE) is a software application that helps programmers develop applications.
- Visual Studio Code is a popular choice.
- Visual Studio is currently installed on your VM's.


File Edit Selection View Go Run Terminal Help
v1_69.md - vscode-docs - Visual Studio Code - Insiders
EXPLORER
OPEN EDITORS
v1_64.md (Working Tree) release-notes
v1_65.md release-notes
tree-view.md api\extension-guides
Preview extension-guidelines.md
extension-guidelines.md api\references
api/references/extension-guidelines.md from 35f82ede \co...
views.md api\ux-guidelines
v1_68.md release-notes
v1_69.md release-notes
VSCODE-DOCS
v1_60.md
v1_61.md
v1_62.md
v1_63.md
v1_64.md
v1_65.md
v1_66.md
v1_67.md
v1_68.md
v1_69.md
vApril.md
vDecember.md
vFebruary.md
vJanuary.md
vMarch.md
vNovember.md
remote
remote-release-notes
wiki
OUTLINE
# June 2022 (version 1.69) > ## Workbench > ### Toggle between Light/Dark Themes
DEBUG CONSOLE
OUTPUT
TERMINAL
pwsh
vnext*
Pull Request #5389
gregvanl, 8 hours ago
Ln 69, Col 106
LF
Markdown
15 Spell

```
your preferred light and dark themes,
**Preferences: Toggle between Light/Dark
Themes**. When in a light theme, the command
swiches to a dark theme and vice versa.

When in a high contrast theme, the command
remains in high contrast and switches to the
preferred high contrast theme of the opposite
color theme.      gregvanl, 8 hours ago • Small

The preferred themes are defined by the
settings:

* `workbench.preferredDarkColorTheme`
* `workbench.preferredLightColorTheme`
* `workbench.preferredHighContrastColorTheme`
* `workbench.
preferredHighContrastLightColorTheme`
```

```
 create mode 100644 release-notes/images/1_69/terminal-si-comman
d-nav.gif
 create mode 100644 release-notes/images/1_69/terminal-si-decora
tion-menu.png
 create mode 100644 release-notes/images/1_69/terminal-si-decora
tions.png
 create mode 100644 release-notes/images/1_69/terminal-si-go-to-
dir.gif
 create mode 100644 release-notes/images/1_69/terminal-si-recent
-command.png
 create mode 100644 release-notes/v1_69.md
 create mode 100644 remote-release-notes/v1_69.md
PS D:\repos\Microsoft\vscode-docs>
 * History restored

PS D:\repos\Microsoft\vscode-docs>
```

1.2 VS Code Basic Layout

The user interface is divided into **five main areas:**

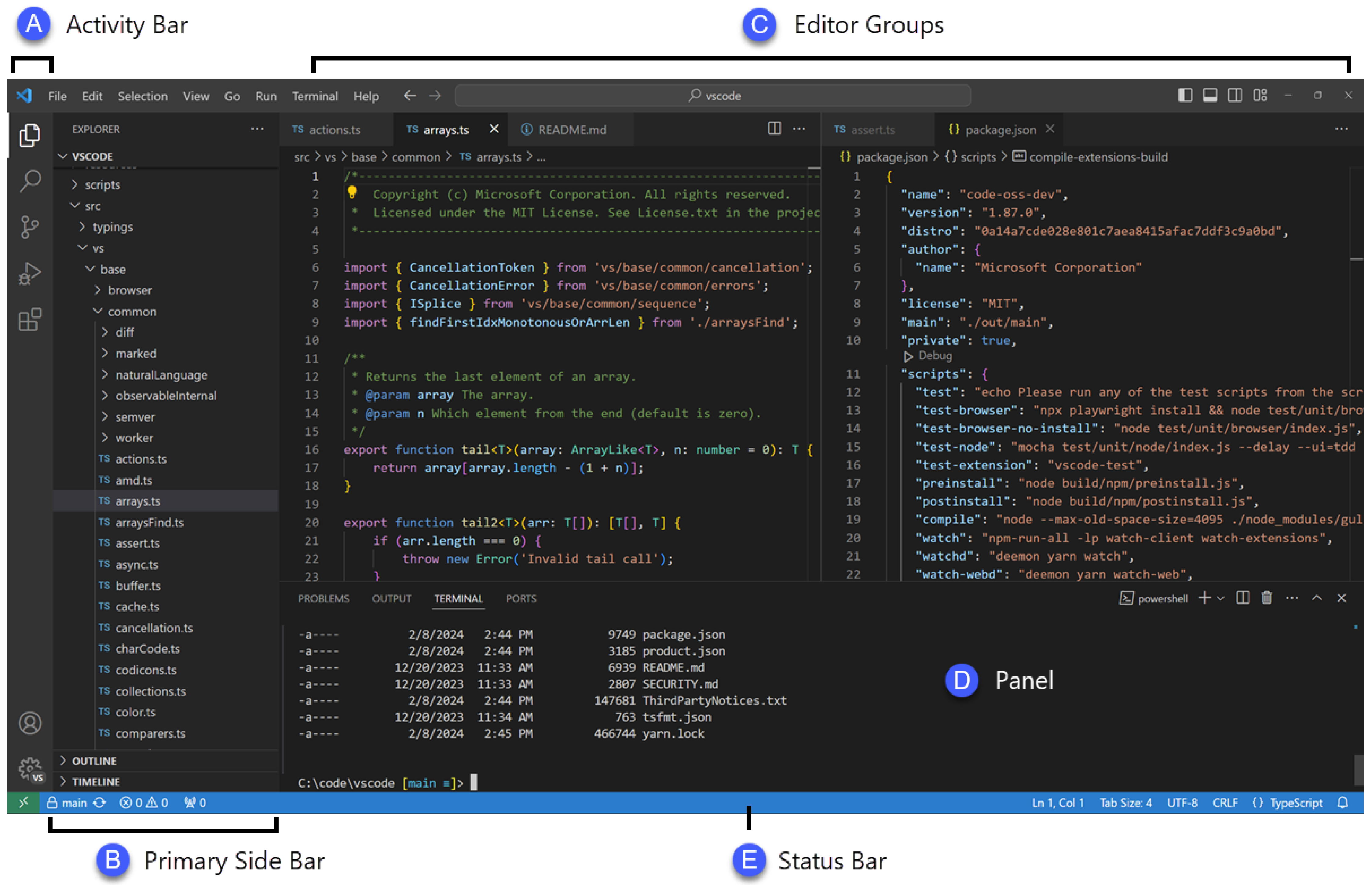
1. **Editor** - The main area to edit your files. You can open as many editors as you like side by side vertically and horizontally.

2. **Primary Side Bar** - Contains different views like the Explorer to assist you while working on your project.

3. **Status Bar** - Information about the opened project and the files you edit.

4. **Activity Bar** - Located on the far left-hand side. Let's you switch between views and gives you additional context-specific indicators, like the number of outgoing changes when Git is enabled. You can change the position of the Activity Bar.

5. **Panel** - An additional space for views below the editor region. By default, it contains output, debug information, errors and warnings, and an integrated terminal. The Panel can also be moved to the left or right for more vertical space.



```
/*---------------------------------------------------------------------
 *  Copyright (c) Microsoft Corporation. All rights reserved.
 *  Licensed under the MIT License. See License.txt in the projec
 *--------------------------------------------------------------------

import { CancellationToken } from 'vs/base/common/cancellation';
import { CancellationError } from 'vs/base/common/errors';
import { ISplice } from 'vs/base/common/sequence';
import { findFirstIdxMonotonousOrArrLen } from './arraysFind';

/**
 * Returns the last element of an array.
 * @param array The array.
 * @param n Which element from the end (default is zero).
 */
export function tail<T>(array: ArrayLike<T>, n: number = 0): T {
	return array[array.length - (1 + n)];
}

export function tail2<T>(arr: T[]): [T[], T] {
	if (arr.length === 0) {
		throw new Error('Invalid tail call');
	}
```

```
{
  "name": "code-oss-dev",
  "version": "1.87.0",
  "distro": "0a14a7cde028e801c7aea8415afac7ddf3c9a0bd",
  "author": {
    "name": "Microsoft Corporation"
  },
  "license": "MIT",
  "main": "./out/main",
  "private": true,
  "scripts": {
    "test": "echo Please run any of the test scripts from the scr
    "test-browser": "npx playwright install && node test/unit/bro
    "test-browser-no-install": "node test/unit/browser/index.js",
    "test-node": "mocha test/unit/node/index.js --delay --ui=tdd
    "test-extension": "vscode-test",
    "preinstall": "node build/npm/preinstall.js",
    "postinstall": "node build/npm/postinstall.js",
    "compile": "node --max-old-space-size=4095 ./node_modules/gul
    "watch": "npm-run-all -lp watch-client watch-extensions",
    "watchd": "deemon yarn watch",
    "watch-webd": "deemon yarn watch-web",
```

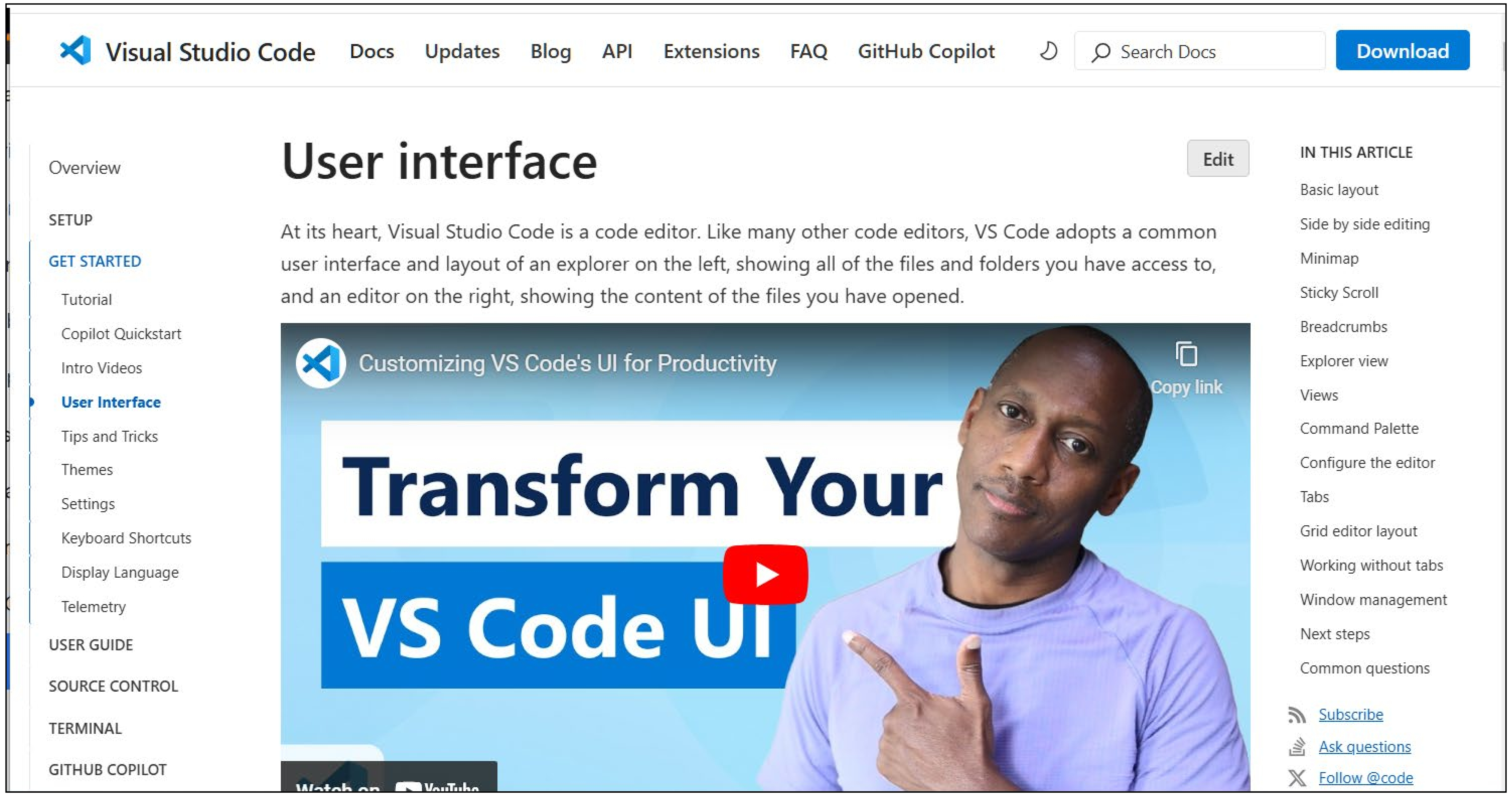
```
-a----          2/8/2024   2:44 PM           9749 package.json
-a----          2/8/2024   2:44 PM           3185 product.json
-a----        12/20/2023  11:33 AM           6939 README.md
-a----        12/20/2023  11:33 AM           2807 SECURITY.md
-a----          2/8/2024   2:44 PM         147681 ThirdPartyNotices.txt
-a----        12/20/2023  11:34 AM            763 tsfmt.json
-a----          2/8/2024   2:45 PM         466744 yarn.lock

C:\code\vscode [main ≡]>
```

WEB AGE SOLUTIONS

1.3 User Interface

Article

https://code.visualstudio.com/docs/getstarted/userinterface



Visual Studio Code Docs Updates Blog API Extensions FAQ GitHub Copilot Search Docs Download

Overview
SETUP
GET STARTED
- Tutorial
- Copilot Quickstart
- Intro Videos
- User Interface
- Tips and Tricks
- Themes
- Settings
- Keyboard Shortcuts
- Display Language
- Telemetry

USER GUIDE
SOURCE CONTROL
TERMINAL
GITHUB COPILOT

User interface

Edit

At its heart, Visual Studio Code is a code editor. Like many other code editors, VS Code adopts a common user interface and layout of an explorer on the left, showing all of the files and folders you have access to, and an editor on the right, showing the content of the files you have opened.

IN THIS ARTICLE
- Basic layout
- Side by side editing
- Minimap
- Sticky Scroll
- Breadcrumbs
- Explorer view
- Views
- Command Palette
- Configure the editor
- Tabs
- Grid editor layout
- Working without tabs
- Window management
- Next steps
- Common questions

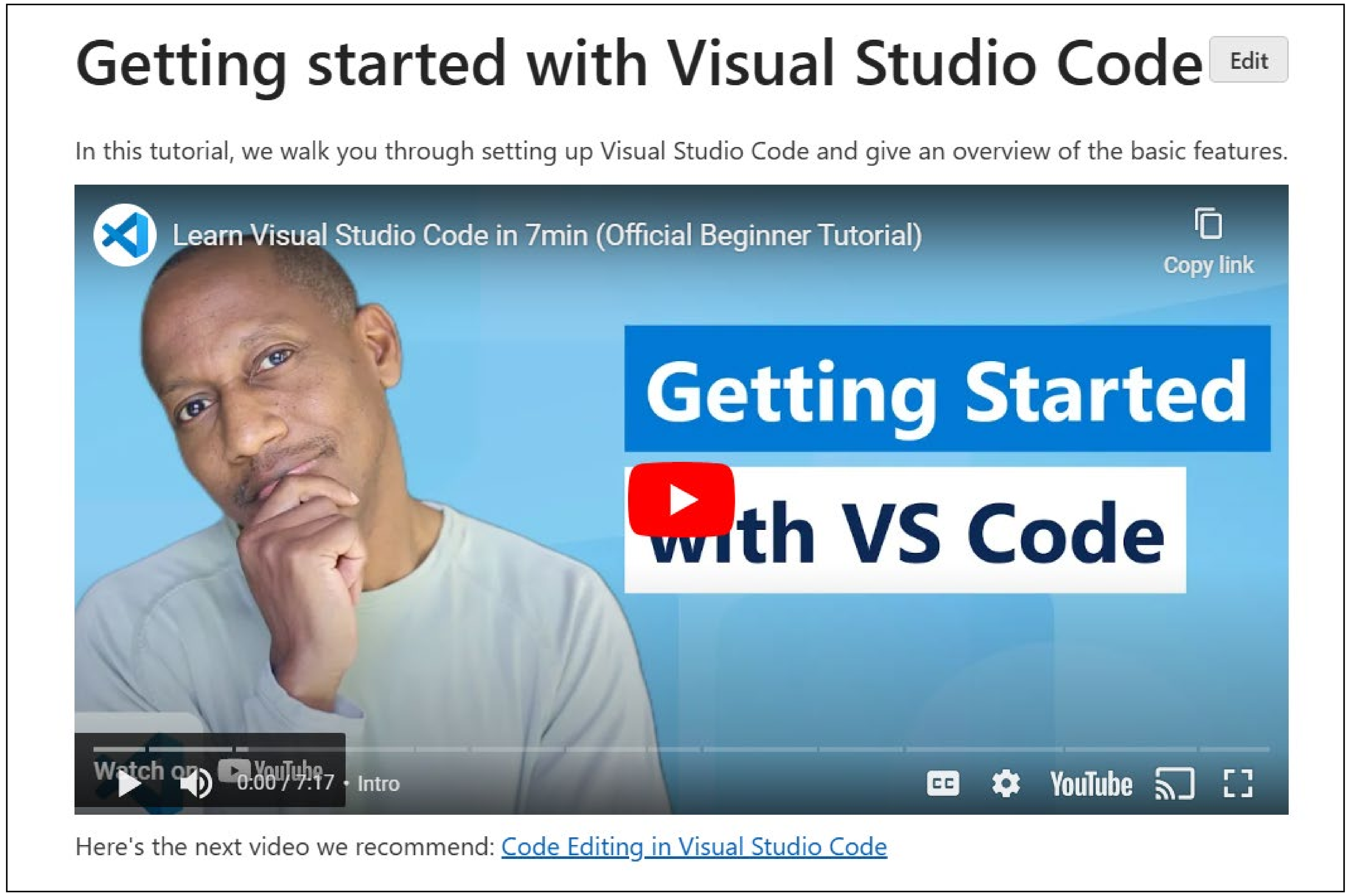
Subscribe
Ask questions
Follow @code

WEB AGE SOLUTIONS

1.4 Getting started with Visual Studio Code

Video

https://code.visualstudio.com/docs/introvideos/basics



Getting started with Visual Studio Code

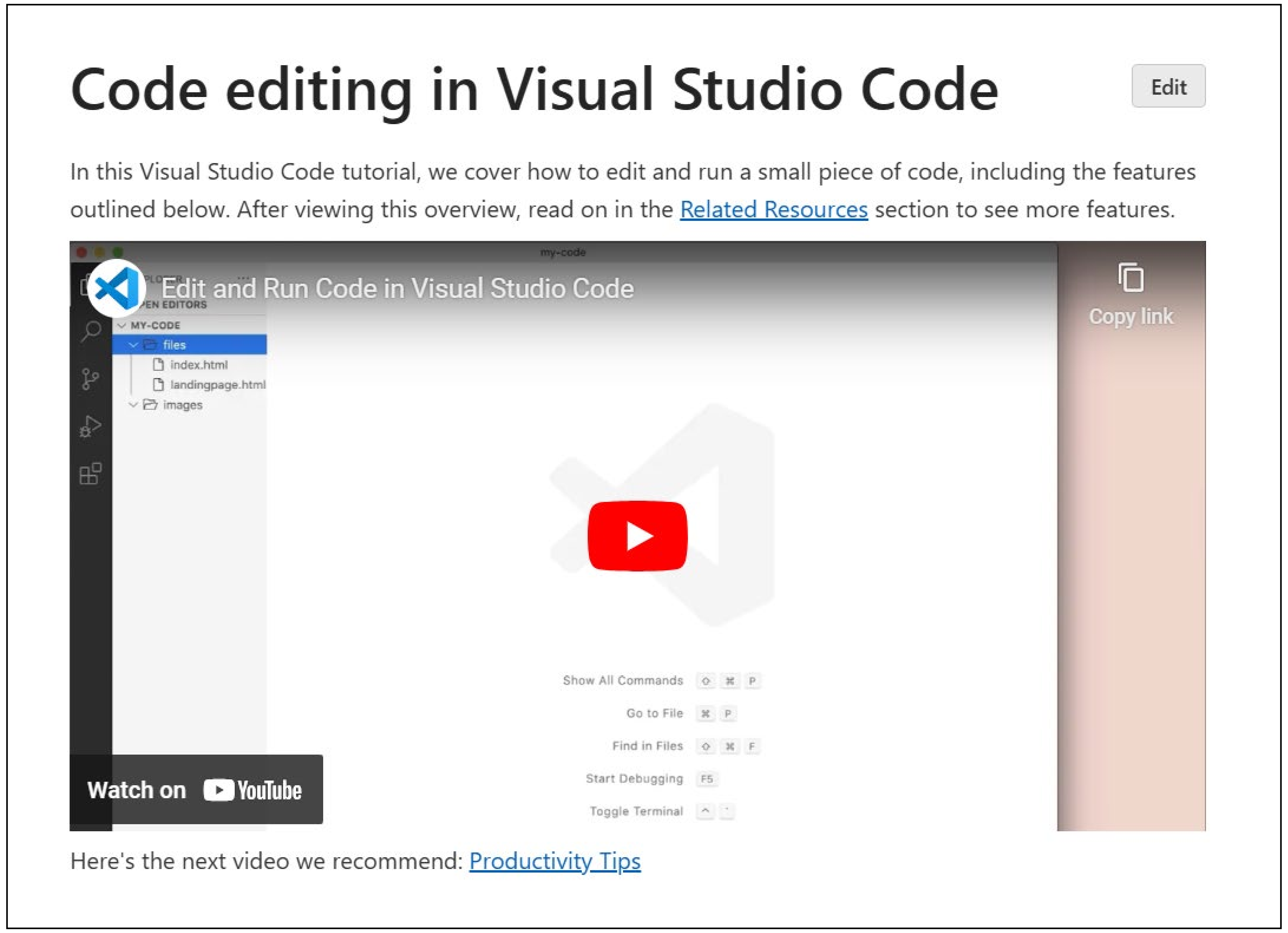
In this tutorial, we walk you through setting up Visual Studio Code and give an overview of the basic features.

Here's the next video we recommend: Code Editing in Visual Studio Code

1.5 Code editing in Visual Studio Code

Video

https://code.visualstudio.com/docs/introvideos/codeediting



Code editing in Visual Studio Code

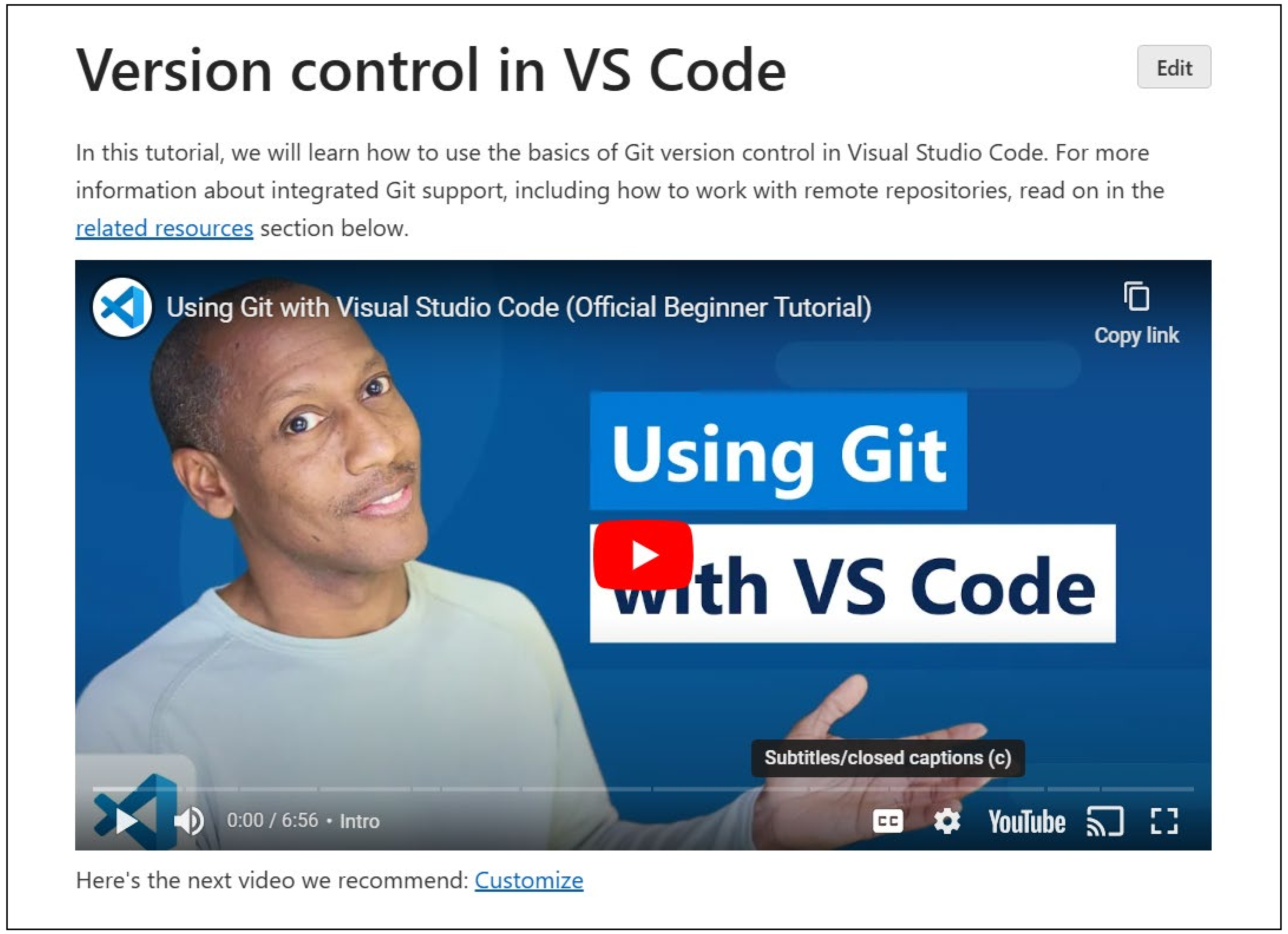
In this Visual Studio Code tutorial, we cover how to edit and run a small piece of code, including the features outlined below. After viewing this overview, read on in the Related Resources section to see more features.

Here's the next video we recommend: Productivity Tips

1.6 Version control in VS Code

Video

https://code.visualstudio.com/docs/introvideos/versioncontrol



Version control in VS Code

In this tutorial, we will learn how to use the basics of Git version control in Visual Studio Code. For more information about integrated Git support, including how to work with remote repositories, read on in the related resources section below.

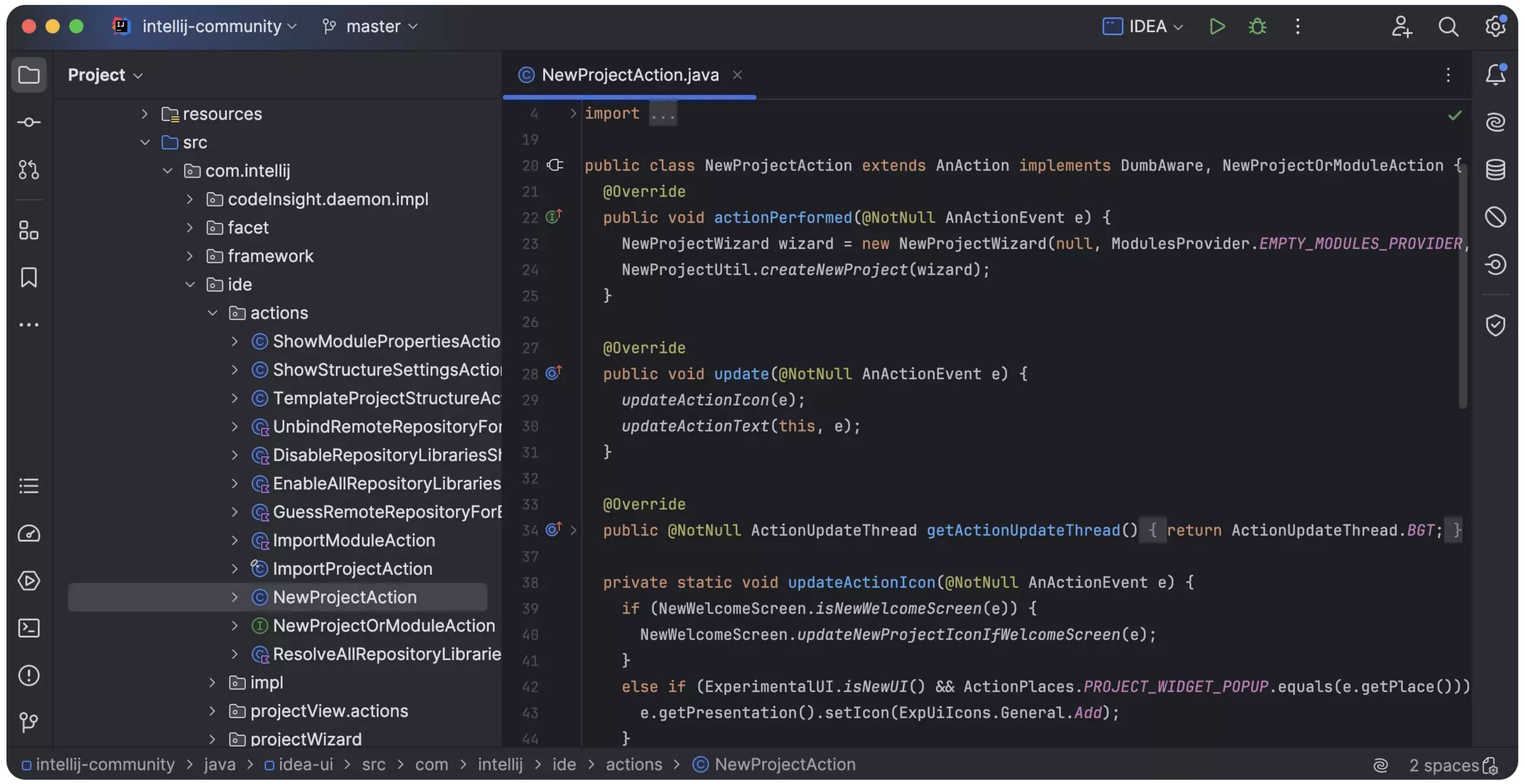
Edit
Using Git with Visual Studio Code (Official Beginner Tutorial)
Copy link
Using Git
with VS Code
Subtitles/closed captions (c)
0:00 / 6:56 • Intro
YouTube

Here's the next video we recommend: Customize

1.7 IntelliJ IDEA

IntelliJ IDEA:

- Is an Integrated Development Environment (IDE) for Java and other programming languages.
- Was Developed by JetBrains
- First released in 2001,
- Widely used, especially for Java development
- Supports Multiple Languages – Java, Kotlin, Python, JavaScript, SQL, …
- Includes built-in version control, debugger, database support, and plugins.
- Available as Community (free) and Ultimate (paid with advanced features) editions.

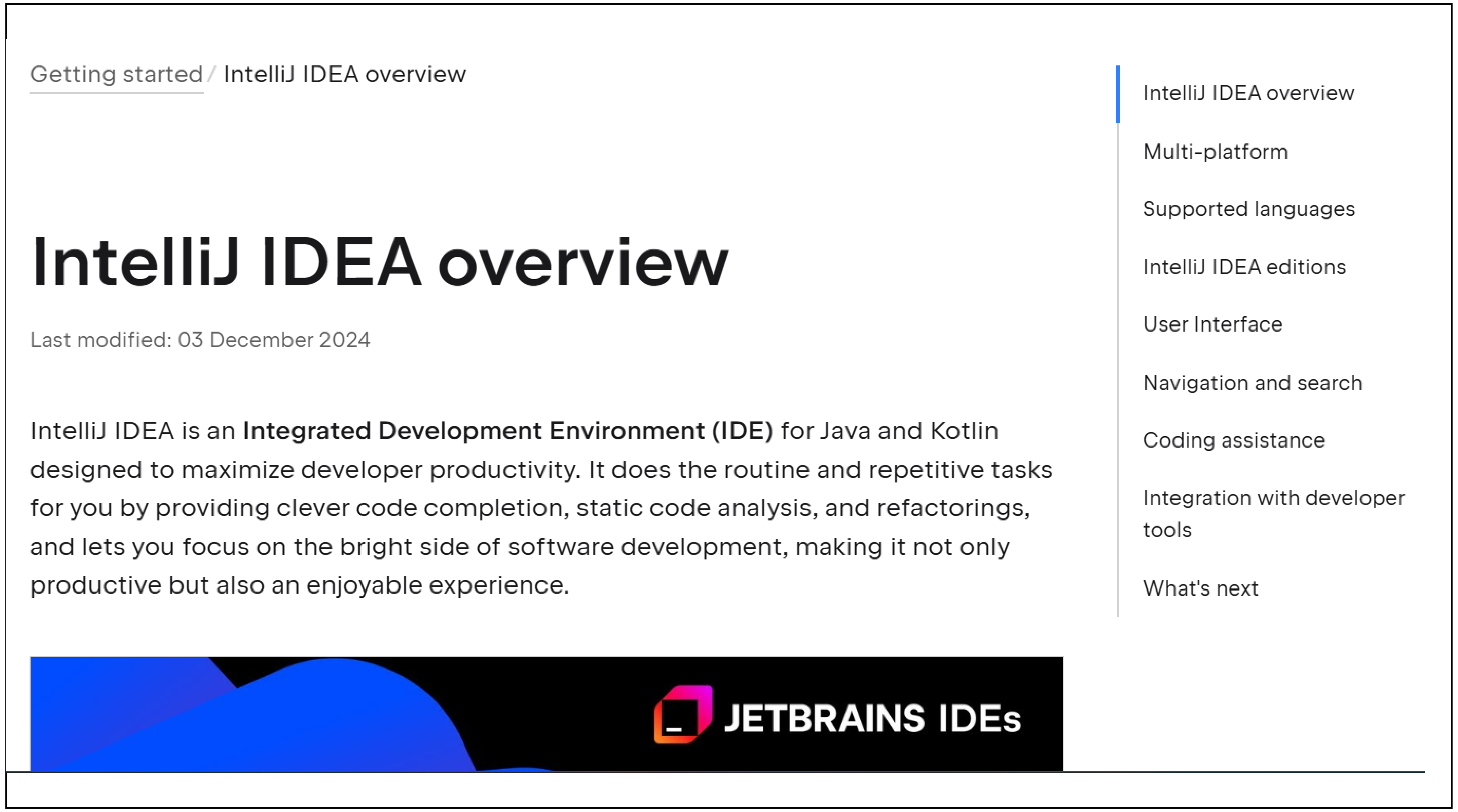

intellij-community
master
IDEA
Project
resources
src
com.intellij
codeInsight.daemon.impl
facet
framework
ide
actions
ShowModulePropertiesActio
ShowStructureSettingsActio
TemplateProjectStructureAc
UnbindRemoteRepositoryFor
DisableRepositoryLibrariesSł
EnableAllRepositoryLibraries
GuessRemoteRepositoryForE
ImportModuleAction
ImportProjectAction
NewProjectAction
NewProjectOrModuleAction
ResolveAllRepositoryLibrarie
impl
projectView.actions
projectWizard
NewProjectAction.java
intellij-community › java › idea-ui › src › com › intellij › ide › actions › NewProjectAction
2 spaces

```java
import ...

public class NewProjectAction extends AnAction implements DumbAware, NewProjectOrModuleAction {
  @Override
  public void actionPerformed(@NotNull AnActionEvent e) {
    NewProjectWizard wizard = new NewProjectWizard(null, ModulesProvider.EMPTY_MODULES_PROVIDER,
    NewProjectUtil.createNewProject(wizard);
  }

  @Override
  public void update(@NotNull AnActionEvent e) {
    updateActionIcon(e);
    updateActionText(this, e);
  }

  @Override
  public @NotNull ActionUpdateThread getActionUpdateThread() { return ActionUpdateThread.BGT; }

  private static void updateActionIcon(@NotNull AnActionEvent e) {
    if (NewWelcomeScreen.isNewWelcomeScreen(e)) {
      NewWelcomeScreen.updateNewProjectIconIfWelcomeScreen(e);
    }
    else if (ExperimentalUI.isNewUI() && ActionPlaces.PROJECT_WIDGET_POPUP.equals(e.getPlace()))
      e.getPresentation().setIcon(ExpUiIcons.General.Add);
    }
```

1.8 IntelliJ IDEA Overview – Article

Article

https://www.jetbrains.com/help/idea/discover-intellij-idea.html



Getting started / IntelliJ IDEA overview

IntelliJ IDEA overview

Last modified: 03 December 2024

IntelliJ IDEA is an **Integrated Development Environment (IDE)** for Java and Kotlin designed to maximize developer productivity. It does the routine and repetitive tasks for you by providing clever code completion, static code analysis, and refactorings, and lets you focus on the bright side of software development, making it not only productive but also an enjoyable experience.

WEB AGE SOLUTIONS

1.9 IntelliJ IDEA Overview – Video

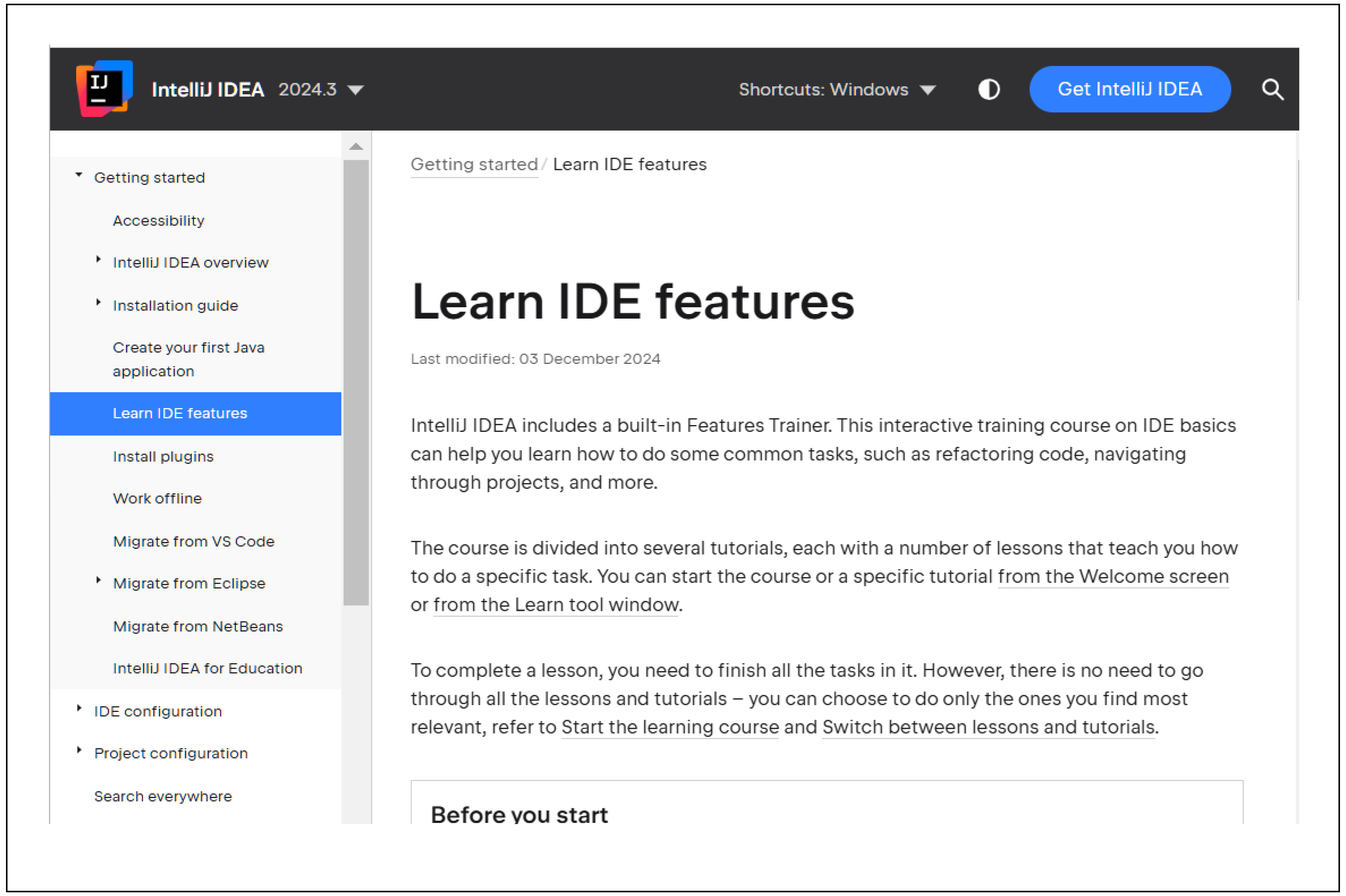
Video

https://www.youtube.com/watch?v=70_B2DyM8mU&t=17s

1.10 Learn IDE features

Interactive Training

https://www.jetbrains.com/help/idea/feature-trainer.html



Learn IDE features

Last modified: 03 December 2024

IntelliJ IDEA includes a built-in Features Trainer. This interactive training course on IDE basics can help you learn how to do some common tasks, such as refactoring code, navigating through projects, and more.

The course is divided into several tutorials, each with a number of lessons that teach you how to do a specific task. You can start the course or a specific tutorial from the Welcome screen or from the Learn tool window.

To complete a lesson, you need to finish all the tasks in it. However, there is no need to go through all the lessons and tutorials – you can choose to do only the ones you find most relevant, refer to Start the learning course and Switch between lessons and tutorials.

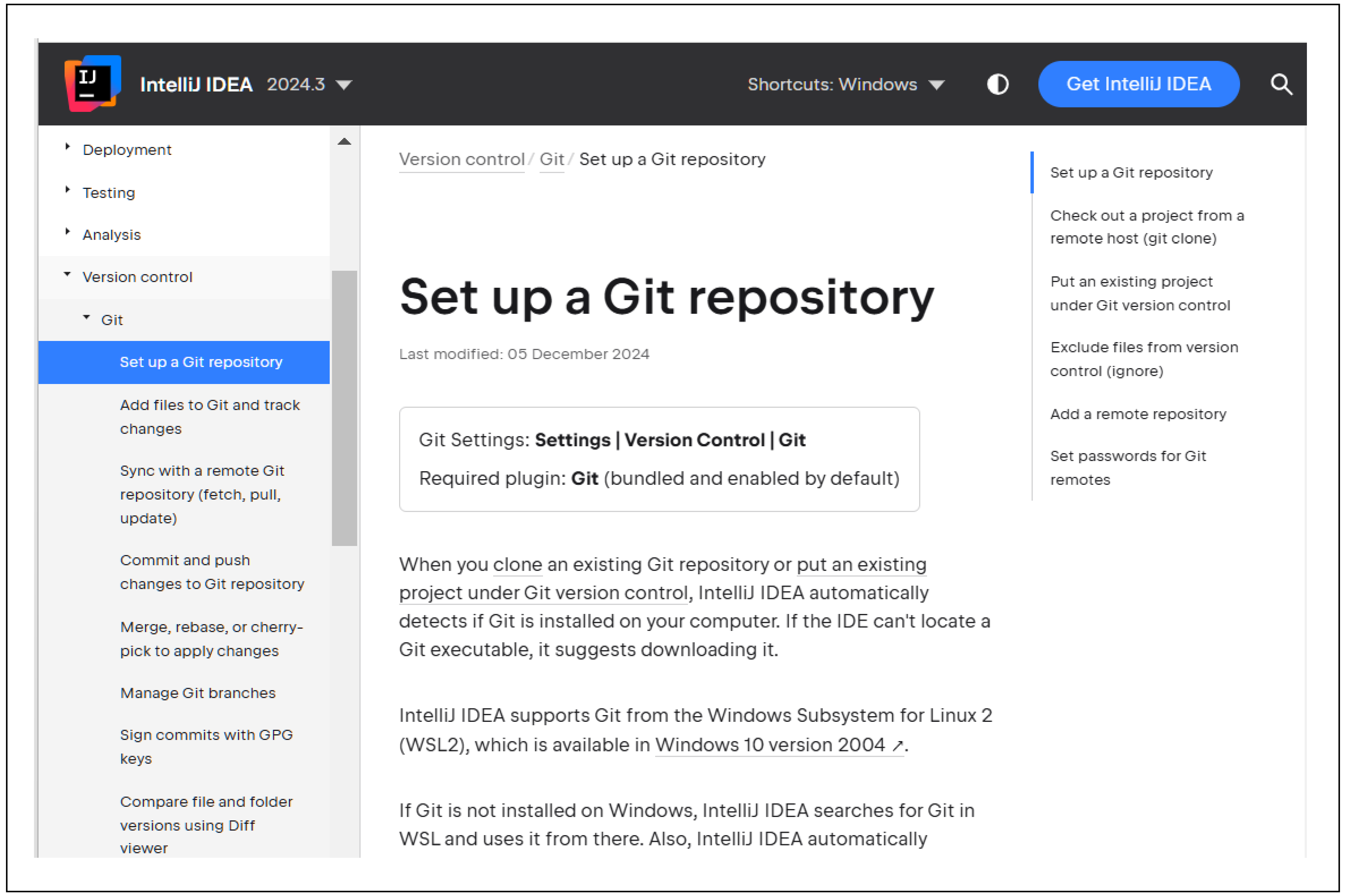
Before you start

WEB AGE SOLUTIONS

1.11 Set up a Git repository

Article

https://www.jetbrains.com/help/idea/set-up-a-git-repository.html



IntelliJ IDEA 2024.3
Shortcuts: Windows
Get IntelliJ IDEA
Deployment
Testing
Analysis
Version control
Git
Set up a Git repository
Add files to Git and track changes
Sync with a remote Git repository (fetch, pull, update)
Commit and push changes to Git repository
Merge, rebase, or cherry-pick to apply changes
Manage Git branches
Sign commits with GPG keys
Compare file and folder versions using Diff viewer
Version control / Git / Set up a Git repository
Set up a Git repository
Check out a project from a remote host (git clone)
Put an existing project under Git version control
Exclude files from version control (ignore)
Add a remote repository
Set passwords for Git remotes

Set up a Git repository

Last modified: 05 December 2024

Git Settings: **Settings | Version Control | Git**

Required plugin: **Git** (bundled and enabled by default)

When you clone an existing Git repository or put an existing project under Git version control, IntelliJ IDEA automatically detects if Git is installed on your computer. If the IDE can't locate a Git executable, it suggests downloading it.

IntelliJ IDEA supports Git from the Windows Subsystem for Linux 2 (WSL2), which is available in Windows 10 version 2004 ↗.

If Git is not installed on Windows, IntelliJ IDEA searches for Git in WSL and uses it from there. Also, IntelliJ IDEA automatically

Summary

- ✓ Have fun with your tasks this week!
- ✓ Remember you can always ask questions in the Basecamp message board. If you run into trouble, or need help, reach out and I can help.
- ✓ Be sure to remember which parts of the course were the most difficult.
- ✓ We can review those at our next Monday session.