Ethans
Learn from experts

WELCOME TO WORLD OF Git

About the Trainer

Siddhant Khanna

Mentor and Trainer - Ethans Tech

Who is this program for?

This program is designed for people who want to:

- Upgrade their skills from legacy technologies.
- Find better job/career opportunities post training.
- Git is a version control system intended for managing and tracking changes to code or text files.
- It caters to the needs of software developers, teams, and organizations, as well as non-technical users who require organized and efficient change management.

Table of Contents

Overview of Version Control Systems

- **What is a Version Control System?**

  Version control systems (VCS) are software tools that enable developers to track changes made to source code and other files over time. They provide a structured approach to managing changes, allowing multiple developers to work on a project simultaneously without conflicts.
- There are two main types of version control systems:
  - Centralized
  - Distributed.
- **Centralized version control systems** (CVCS) store all the changes made to a project in a central repository. This means that all developers have to connect to the central server to access the code and make changes. Examples of centralized version control systems include Subversion (SVN) and Concurrent Versions System (CVS).
- **Distributed version control systems** (DVCS) allow each developer to have a local copy of the entire codebase and history on their own computer. This makes it easier to work offline and collaborate with

Overview of Version Control Systems – Continue..

other developers without needing a central server. Examples of distributed version control systems include Git, Mercurial, and Bazaar.

- Version control systems provide several benefits, including:
  - **History tracking:** VCS allows developers to keep track of every change made to the codebase, including who made the change, when it was made, and why.
  - **Collaboration:** Multiple developers can work on the same project without conflicts or overriding each other's work.
  - **Branching:** VCS enables developers to create different branches of the codebase, allowing them to work on new features or bug fixes without affecting the main codebase.
  - **Rollbacks:** In case of errors or bugs, developers can easily roll back to a previous version of the codebase.
  - **Continuous integration and deployment:** VCS enables automated testing and deployment, ensuring that new changes are integrated and tested before they are released.

What is Git?

- Git is a free and open-source distributed version control system (DVCS) designed to handle everything from small to very large projects with speed and efficiency. It was created by Linus Torvalds in 2005 and has since become one of the most widely used version control systems in the world.
- Git is a distributed version control system, which means that every developer has a local copy of the entire codebase and history on their own computer. This makes it easy to work offline and collaborate with other developers without needing a central server.
- Git is used by millions of developers and organizations worldwide, including large companies like Microsoft, Google, and Facebook. It has become the de facto standard for version control in the software development industry, and many other tools and services have been built around Git to extend its functionality.
- Git provides a wide range of features, including:
  - **Branching and merging**: Git allows developers to create different branches of the codebase, allowing them to work on new features or bug fixes without affecting the main codebase. Git also

What is Git? – Continue..

makes it easy to merge changes from one branch to another.

- **History tracking**: Git allows developers to keep track of every change made to the codebase, including who made the change, when it was made, and why.
- **Rollbacks**: In case of errors or bugs, developers can easily roll back to a previous version of the codebase.
- **Collaboration**: Multiple developers can work on the same project without conflicts or overriding each other's work.
- **Speed and performance**: Git is designed to be fast and efficient, even with very large codebases.

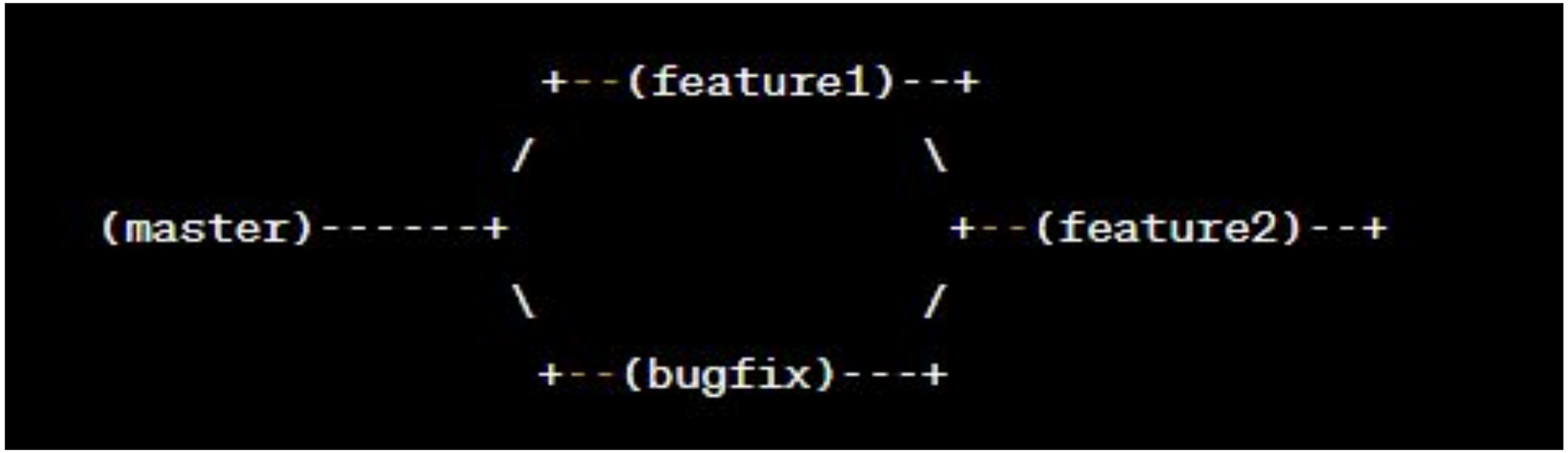
Git Workflow

- The workflow of Git can vary depending on the specific needs of a project or team, but the following is a common workflow that is often used:
  - **Create a Git repository:** The first step is to create a new Git repository. This can be done by running the "git init" command in the root directory of the project.
  - **Add files:** Once the repository is created, add files to the repository using the "git add" command. This stages the changes to be committed.
  - **Commit changes:** Once the changes are staged, commit them using the "git commit" command. This creates a new commit that includes the changes made to the files.
  - **Create branches:** To work on a new feature or bug fix, create a new branch using the "git branch" command. This creates a new copy of the codebase that can be worked on separately from the main codebase.
  - **Switch branches:** Use the "git checkout" command to switch between branches. This allows developers to work on different features or bug fixes without affecting the main codebase.

Git Workflow – Continue..

- ○ **Merge branches:** Once a feature or bug fix is complete, merge the changes back into the main codebase using the "git merge" command.
- ○ **Resolve conflicts:** Sometimes, conflicts can arise when merging changes from different branches. In this case, use the "git diff" command to compare the changes and resolve any conflicts manually.
- ○ **Push changes:** Once the changes are merged and conflicts are resolved, push the changes.

● The following image shows the minimal implementation for the version control for a software system using Git:



```
                +--(feature1)--+
               /                \
(master)------+                  +--(feature2)--+
               \                /
                +--(bugfix)---+
```

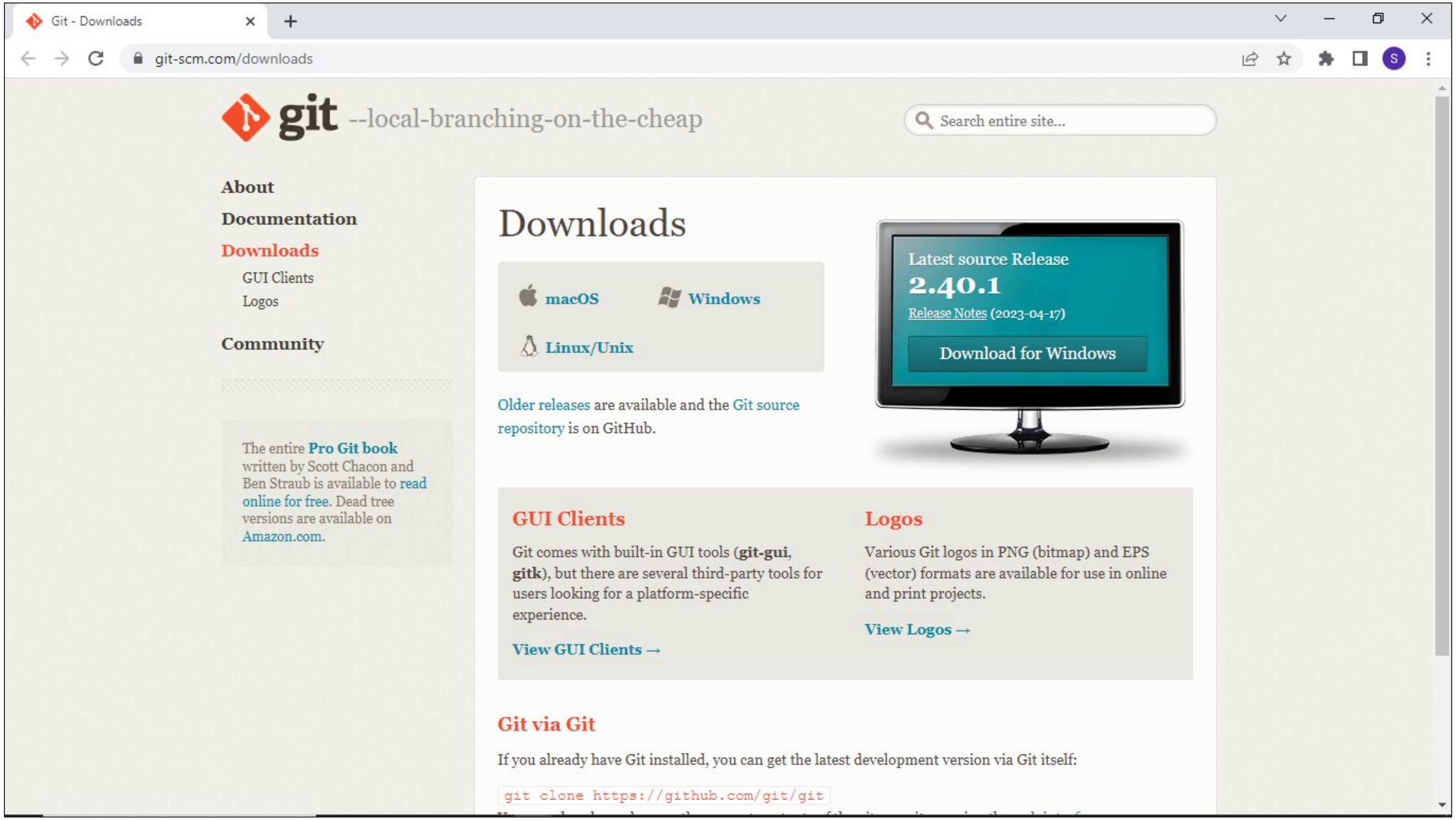
Git Workflow – Continue..

- In this diagram, the main branch of the codebase is called "master". Developers can create new branches, such as "feature1", "feature2", and "bugfix", to work on new features or bug fixes. Each branch is essentially a copy of the codebase that can be worked on independently.
- Developers can switch between branches using the "git checkout" command. When a feature or bug fix is complete, changes from the branch can be merged back into the main codebase using the "git merge" command.
- If conflicts arise during the merge process, developers can use the "git diff" command to compare the changes and manually resolve any conflicts.
- Once the changes are merged and conflicts are resolved, developers can push the changes to the remote repository using the "git push" command. Other developers can then pull the changes from the remote repository using the "git pull" command.

Git Installation

- To install Git on your computer, follow these steps:
  - Go to the Git website at https://git-scm.com/downloads.
  - Select the appropriate version of Git for your operating system. For example, if you're running Windows, select the "Windows" option.
  - Download the installer for Git.
  - Run the installer and follow the prompts to install Git on your computer. Be sure to select any additional components you may want to install, such as Git Bash (a Unix-style command-line interface for Windows).
  - Once the installation is complete, open a terminal or command prompt and type "git --version" to verify that Git has been installed correctly.
  - Configure Git with your name and email address using the following commands:
    - git config --global user.name "Your Name"
    - git config --global user.email "youremail@example.com"

Git Installation Steps



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.

Downloads

Older releases are available and the Git source repository is on GitHub.

Latest source Release
2.40.1
Release Notes (2023-04-17)

GUI Clients

Git comes with built-in GUI tools (**git-gui**, **gitk**), but there are several third-party tools for users looking for a platform-specific experience.

Logos

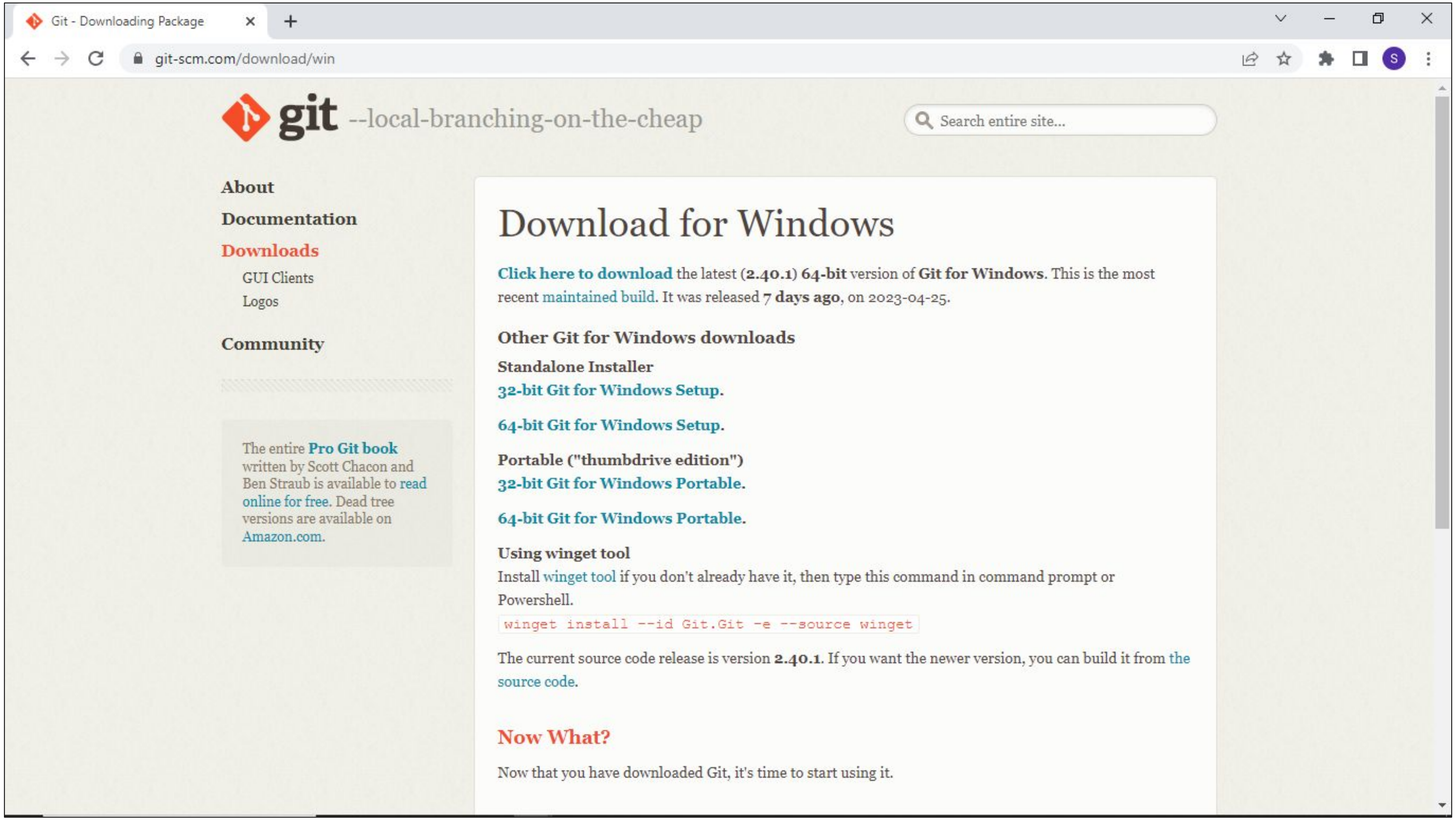
Various Git logos in PNG (bitmap) and EPS (vector) formats are available for use in online and print projects.

Git via Git

If you already have Git installed, you can get the latest development version via Git itself:

```
git clone https://github.com/git/git
```

Git Installation Steps – Continue..



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 (**2.40.1**) **64-bit** version of **Git for Windows**. This is the most recent maintained build. It was released **7 days ago**, on 2023-04-25.

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.

Using winget tool

Install winget tool if you don't already have it, then type this command in command prompt or Powershell.

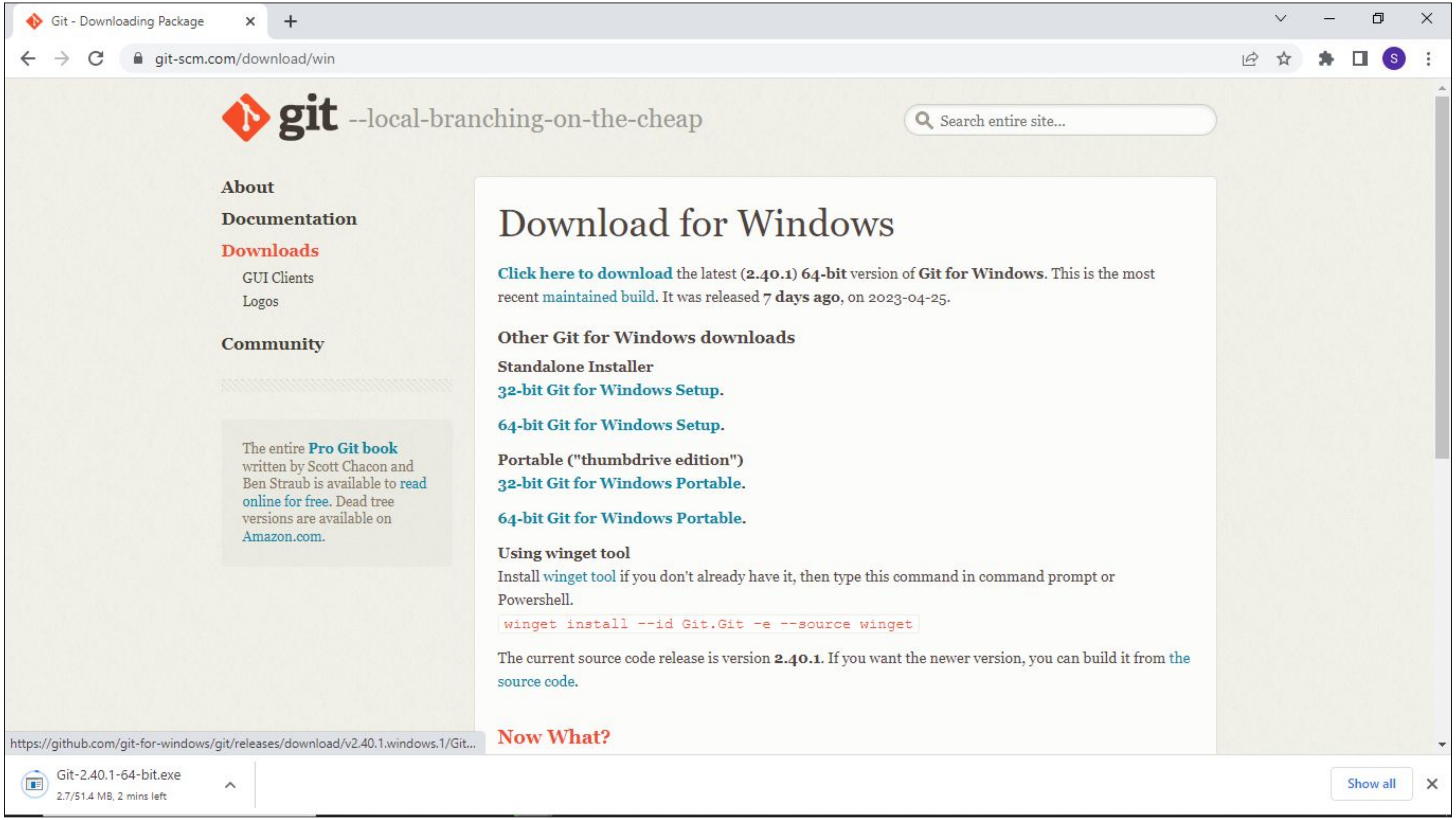
```
winget install --id Git.Git -e --source winget
```

The current source code release is version **2.40.1**. If you want the newer version, you can build it from the source code.

Now What?

Now that you have downloaded Git, it's time to start using it.

Git Installation Steps – Continue..



Download for Windows

Click here to download the latest (2.40.1) 64-bit version of Git for Windows. This is the most recent maintained build. It was released 7 days ago, on 2023-04-25.

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.

Using winget tool

Install winget tool if you don't already have it, then type this command in command prompt or Powershell.

```
winget install --id Git.Git -e --source winget
```

The current source code release is version 2.40.1. If you want the newer version, you can build it from the source code.

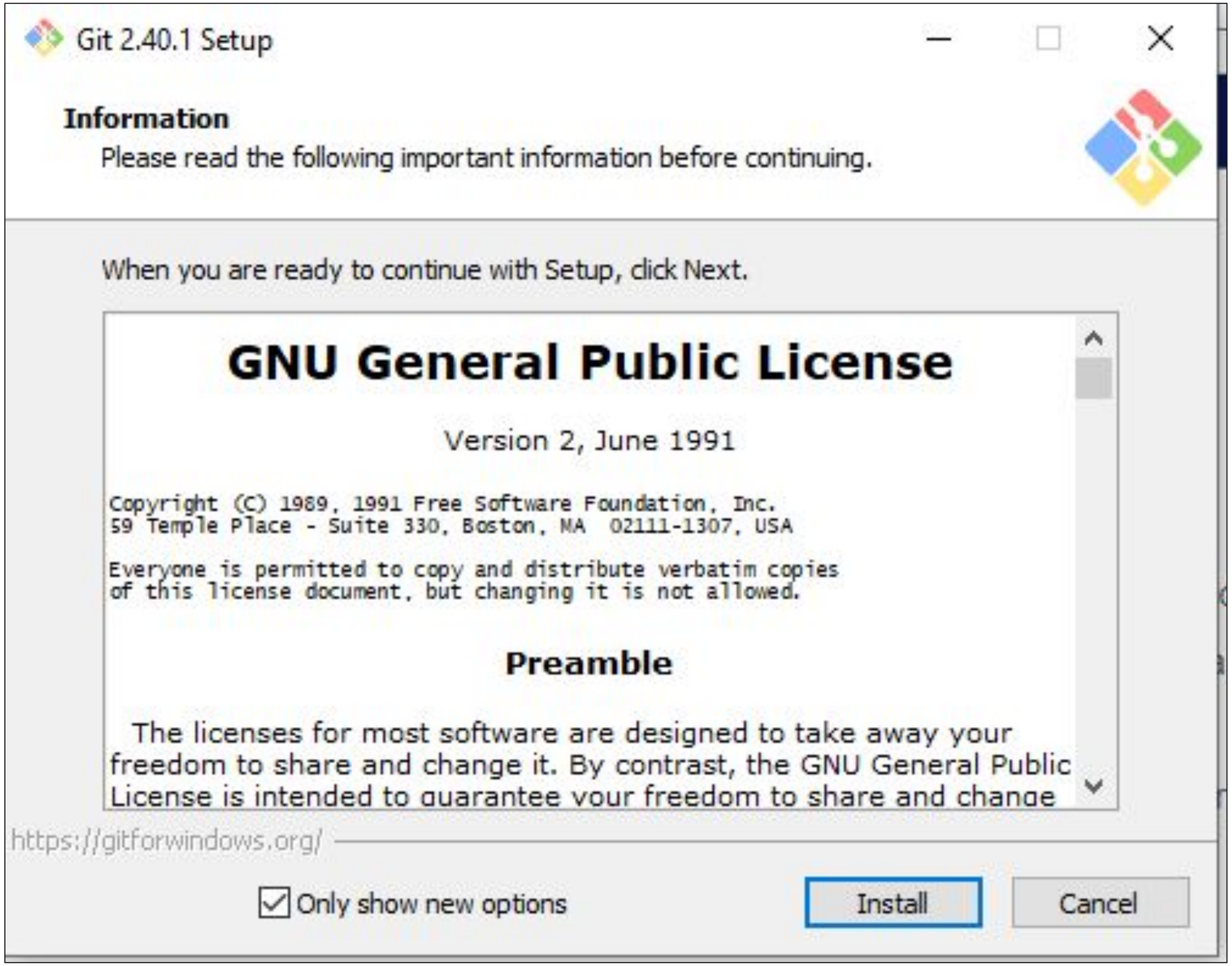
Now What?

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.

https://github.com/git-for-windows/git/releases/download/v2.40.1.windows.1/Git...

Git-2.40.1-64-bit.exe
2.7/51.4 MB, 2 mins left

Git Installation Steps – Continue..



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

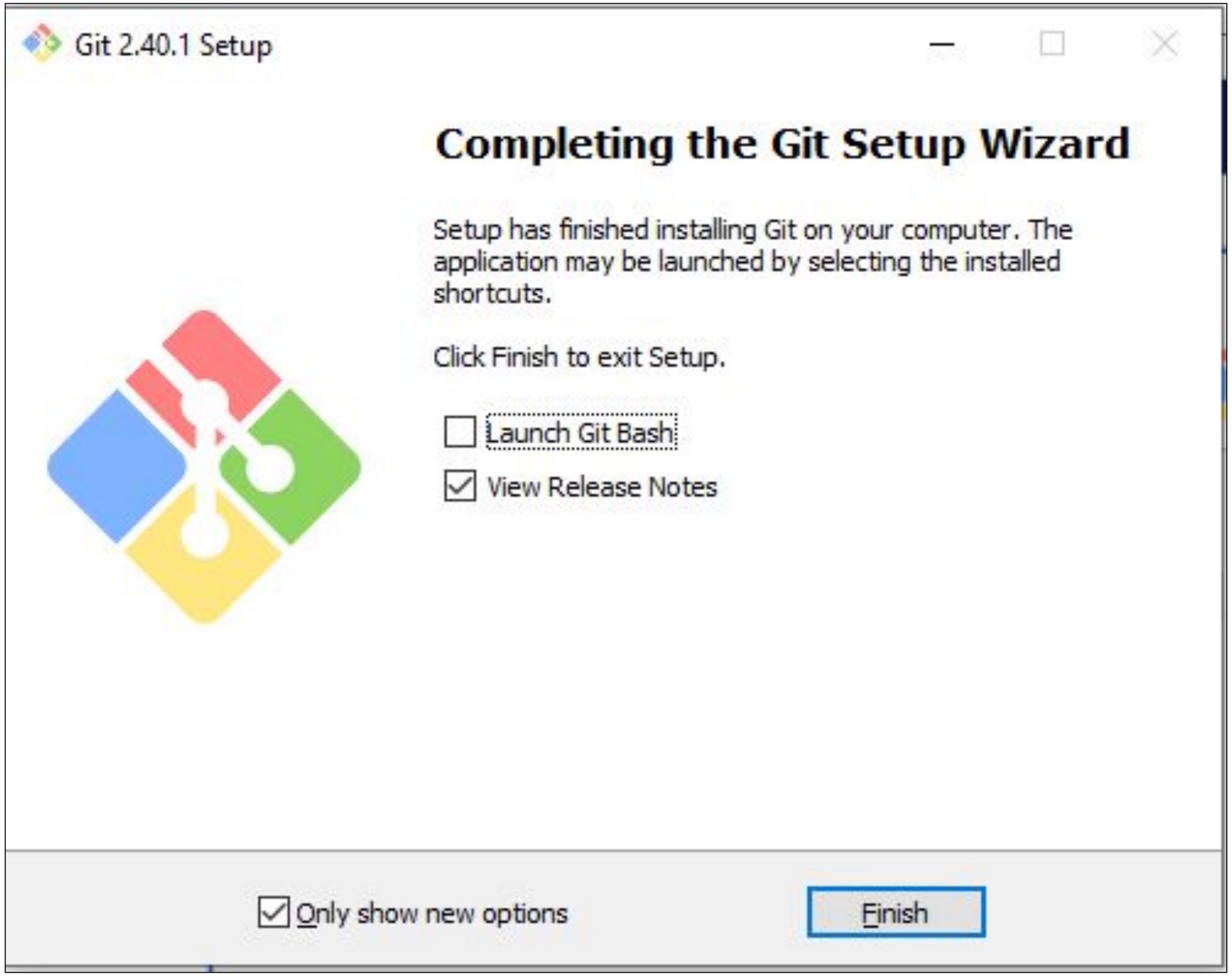
```
Copyright (C) 1989, 1991 Free Software Foundation, Inc.
59 Temple Place - Suite 330, Boston, MA  02111-1307, USA

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 Installation Steps – Continue..



Git 2.40.1 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.

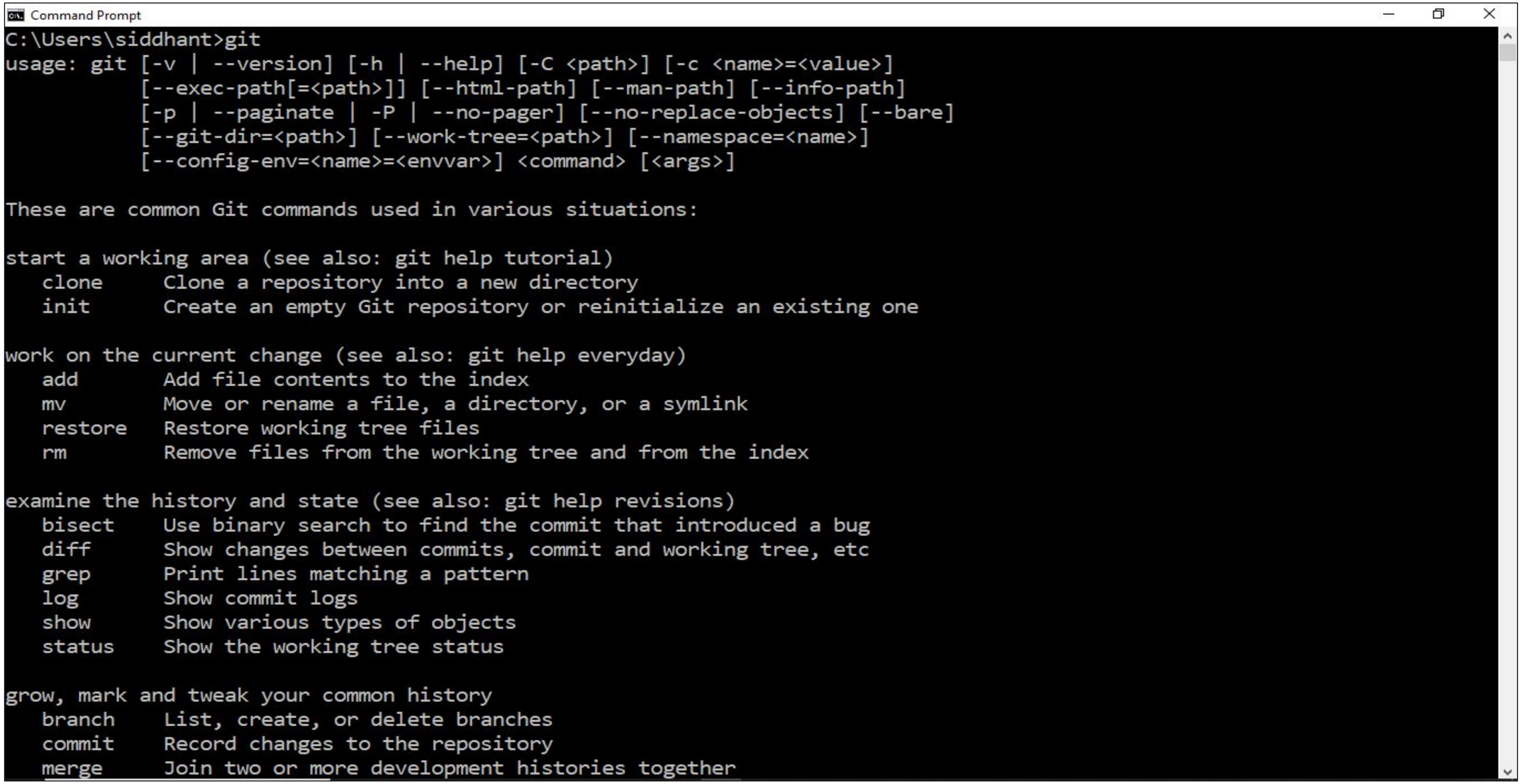
☐ Launch Git Bash

☑ View Release Notes

☑ Only show new options

Git Installation Steps – Continue..

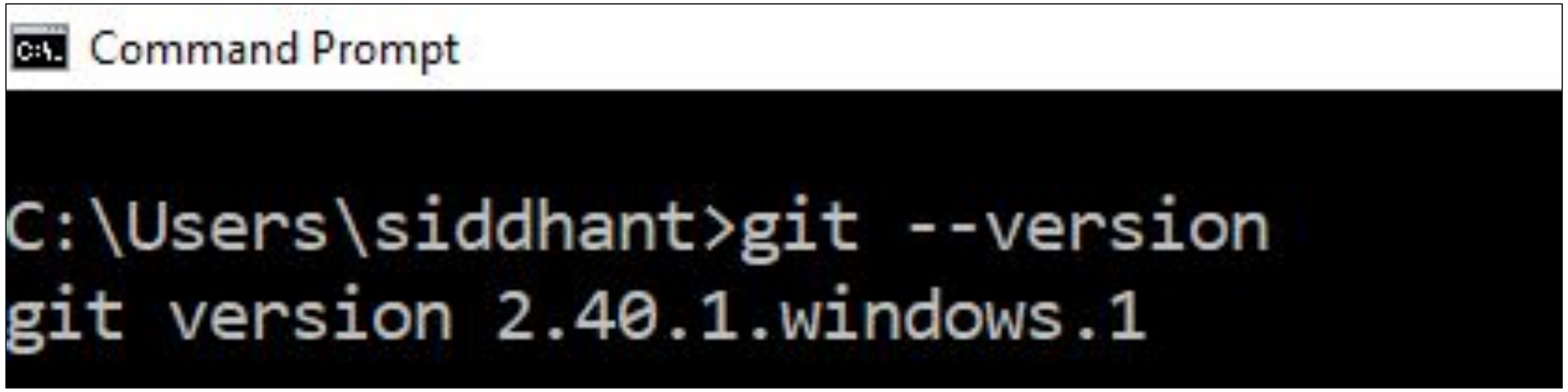
- You can check the installation of git by running the git command in the CMD/terminal:



```
C:\Users\siddhant>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
```

Git Installation Steps – Continue..

- You can check the version of git by running the git –version command in the CMD/terminal:



```
C:\Users\siddhant>git --version
git version 2.40.1.windows.1
```

Some Common FAQs

Q1: What is a version control system?

A1: A version control system (VCS) is a software tool that helps developers manage changes to their code over time. It allows developers to track changes to their codebase, collaborate with other developers, and revert to previous versions of their code if needed.

Q2: What are some common version control systems?

A2: Some common VCS include Git, Subversion (SVN), Mercurial, and Perforce.

Q3: What makes Git different from other version control systems?

A3: Git is a distributed version control system, which means that each developer has a complete copy of the codebase on their local machine. This makes it easier to work offline and collaborate with other developers.

Some Common FAQs

Q4: What are some popular Git hosting platforms?

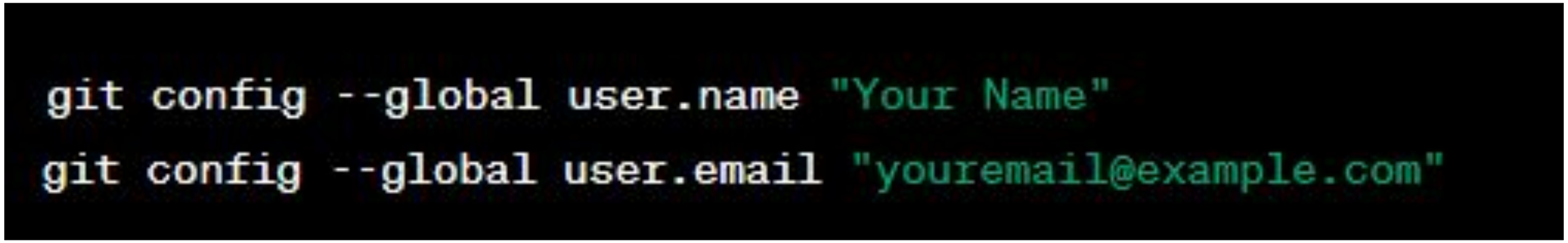
A4: Some popular Git hosting platforms include GitHub, GitLab, and Bitbucket.

Q5: How do I check which version of Git is installed on my computer?

A5: You can open a terminal or command prompt and type "git --version" to check the version of Git installed on your computer.

Q6: Do I need to configure Git after installing it?

A6: Yes, you should configure Git with your name and email address using the following commands:



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

Trivia Quiz

Q1. What is a version control system?

a. A tool used for debugging software

b. **A tool used for managing changes to code over time**

c. A tool used for compiling software

d. A tool used for testing software

Q2. What is Git?

a. A tool used for debugging software

b. **A tool used for managing changes to code over time**

c. A tool used for compiling software

d. A tool used for testing software

Q3. What is a distributed version control system?

a. A version control system that storeQs all code on a central server

b. **A version control system that stores all code on developers' local machines**

c. A version control system that stores code on multiple servers

d. A version control system that doesn't store code at all

Trivia Quiz

Q4. What is a benefit of using Git?

a. Faster performance
b. Better collaboration
c. Easier branching and merging
d. All of the above

Q5.Which of the following is a popular Git hosting platform?

a. GitHub
b. Subversion
c. Mercurial
d. Perforce

Q6. Which of the following is a common version control system?

a. Git
b. Jenkins
c. Docker
d. Kubernetes