4. Installation & Setting

1. Installation

▪ Download Git

- https://git-scm.com/
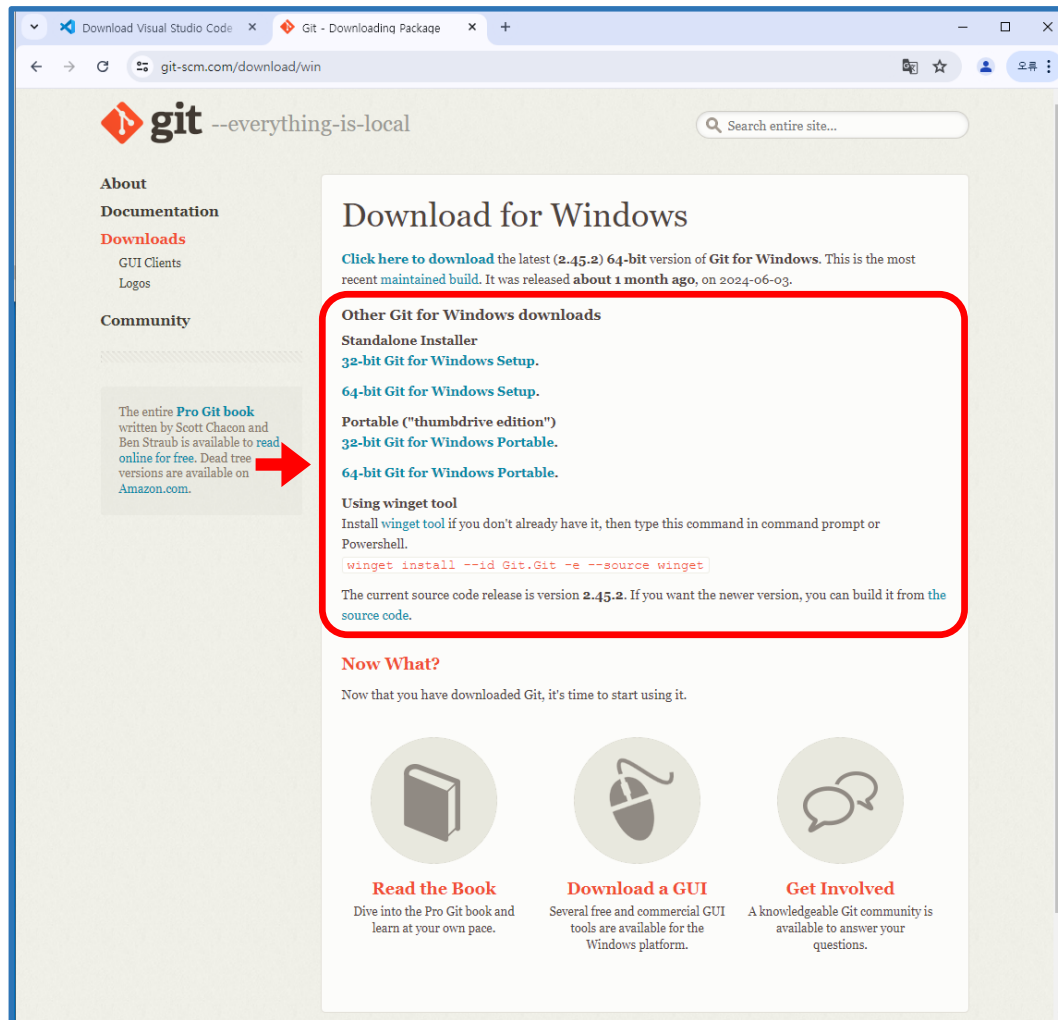
1. Installation

▪ Download Git

- https://git-scm.com/download/win

1. Installation

■ **Execute the downloaded .exe file**

Git-2.45.2-64-bit
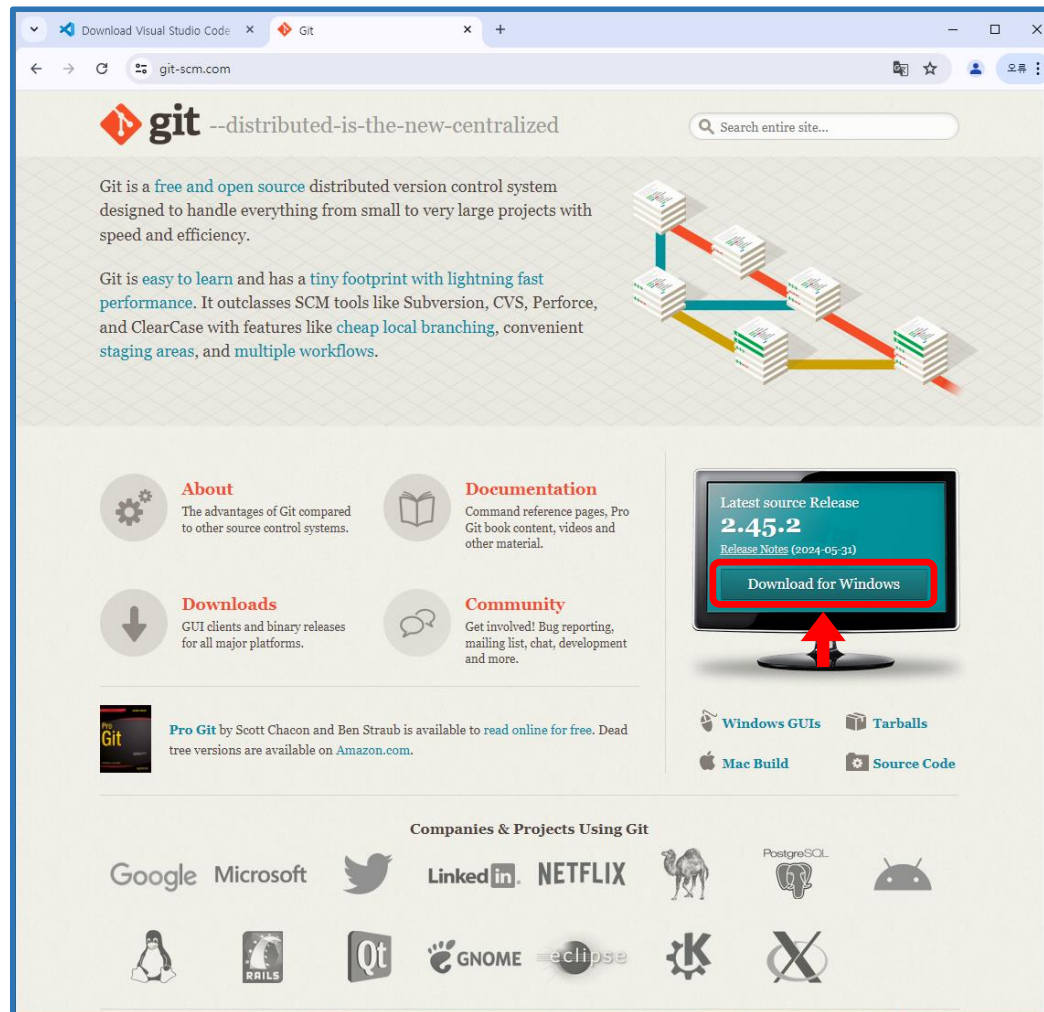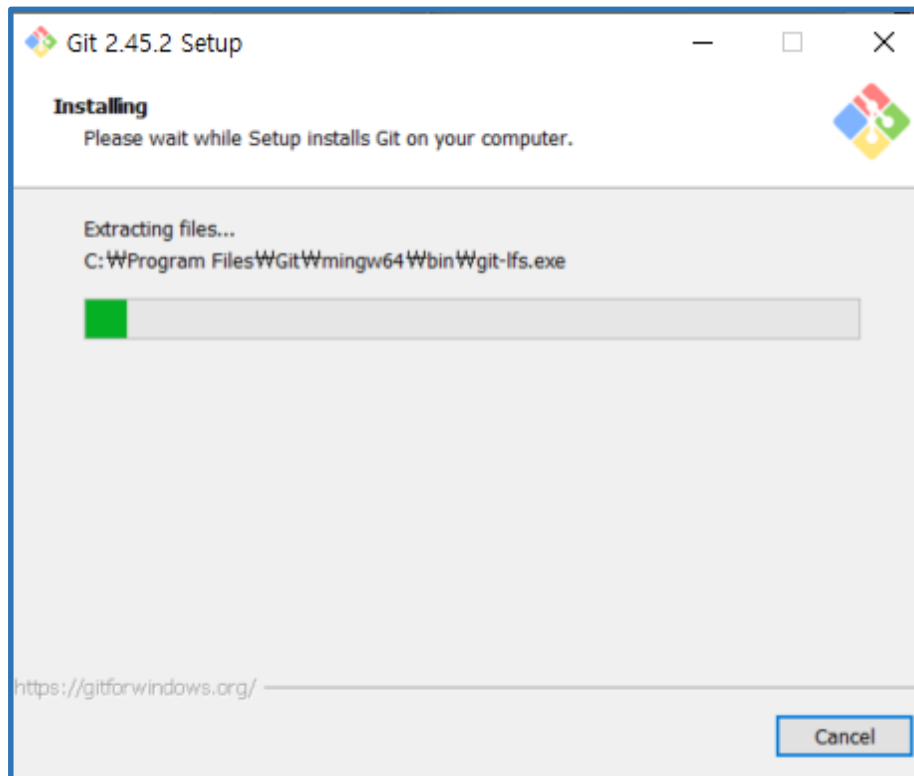
■ **Set it as default and move on to the "Next"**
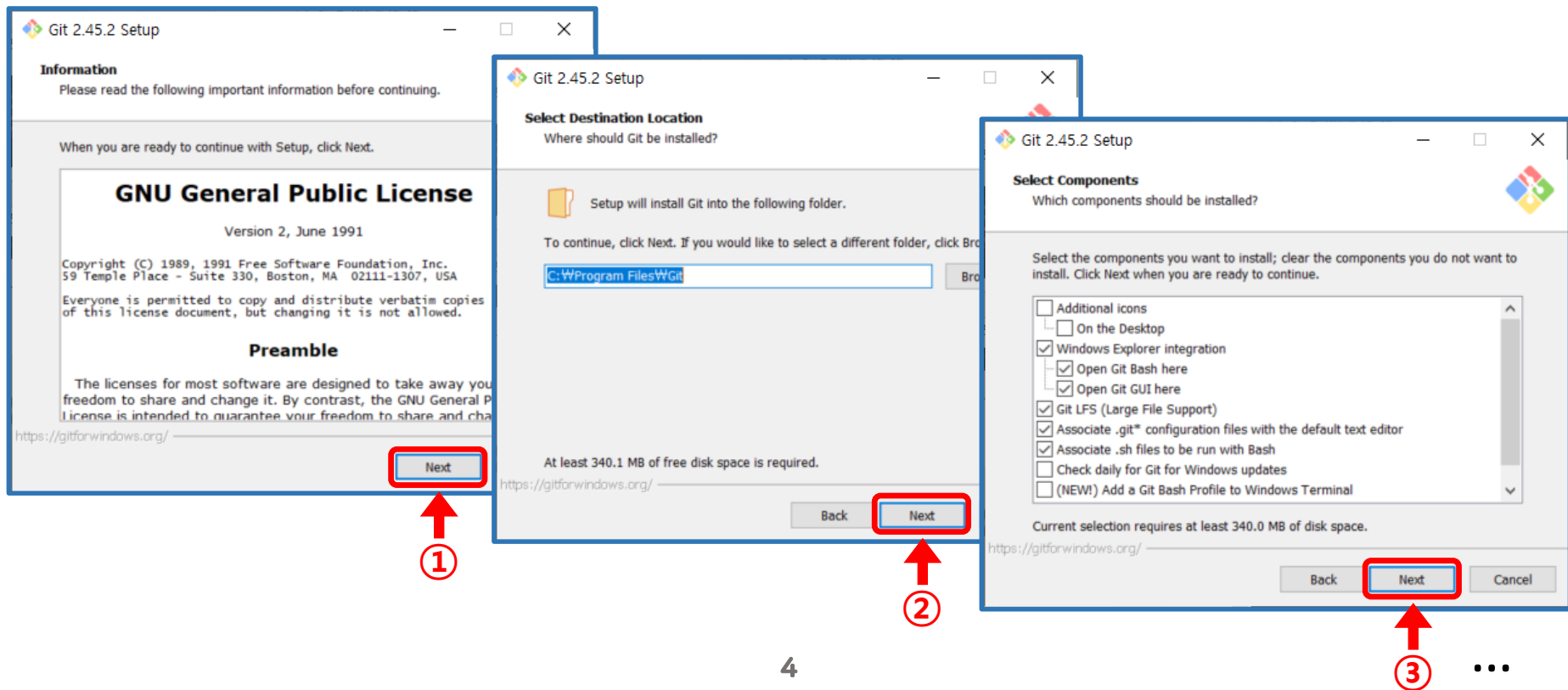
1. Installation

- **Execute the downloaded .exe file**
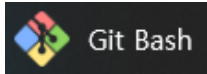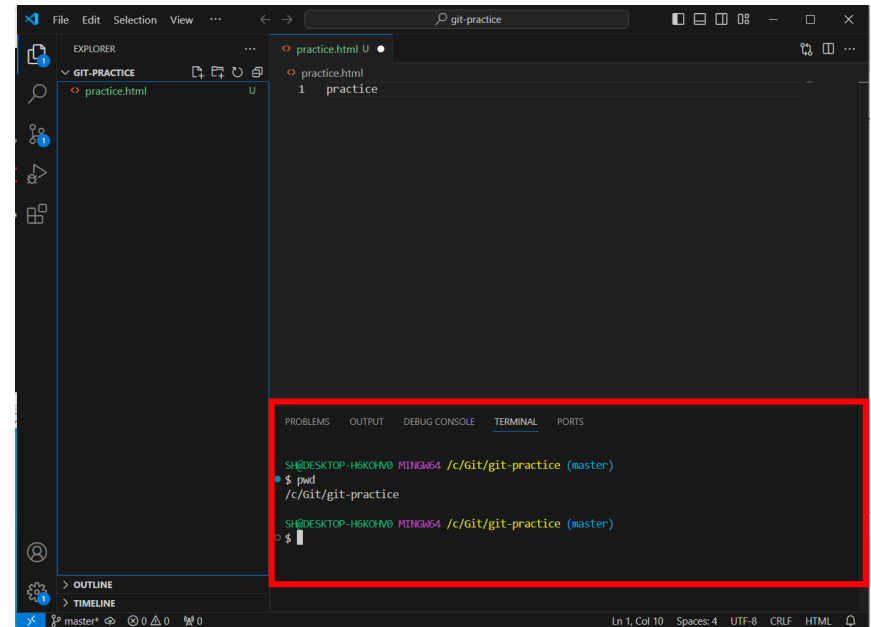

- **Set it as default and install**

1. Installation

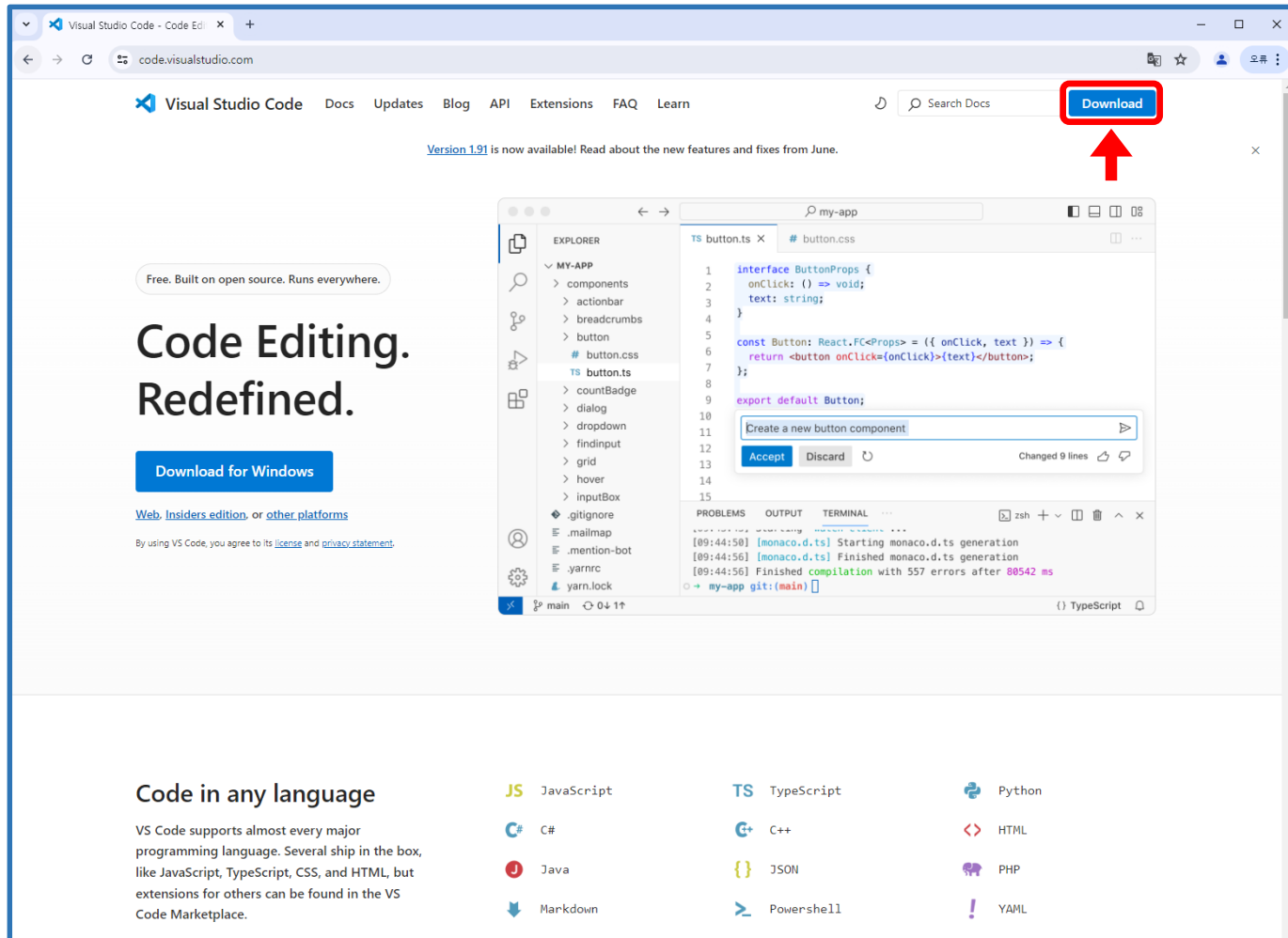- **Now can control Git in "Git Bash" or "VScode"**

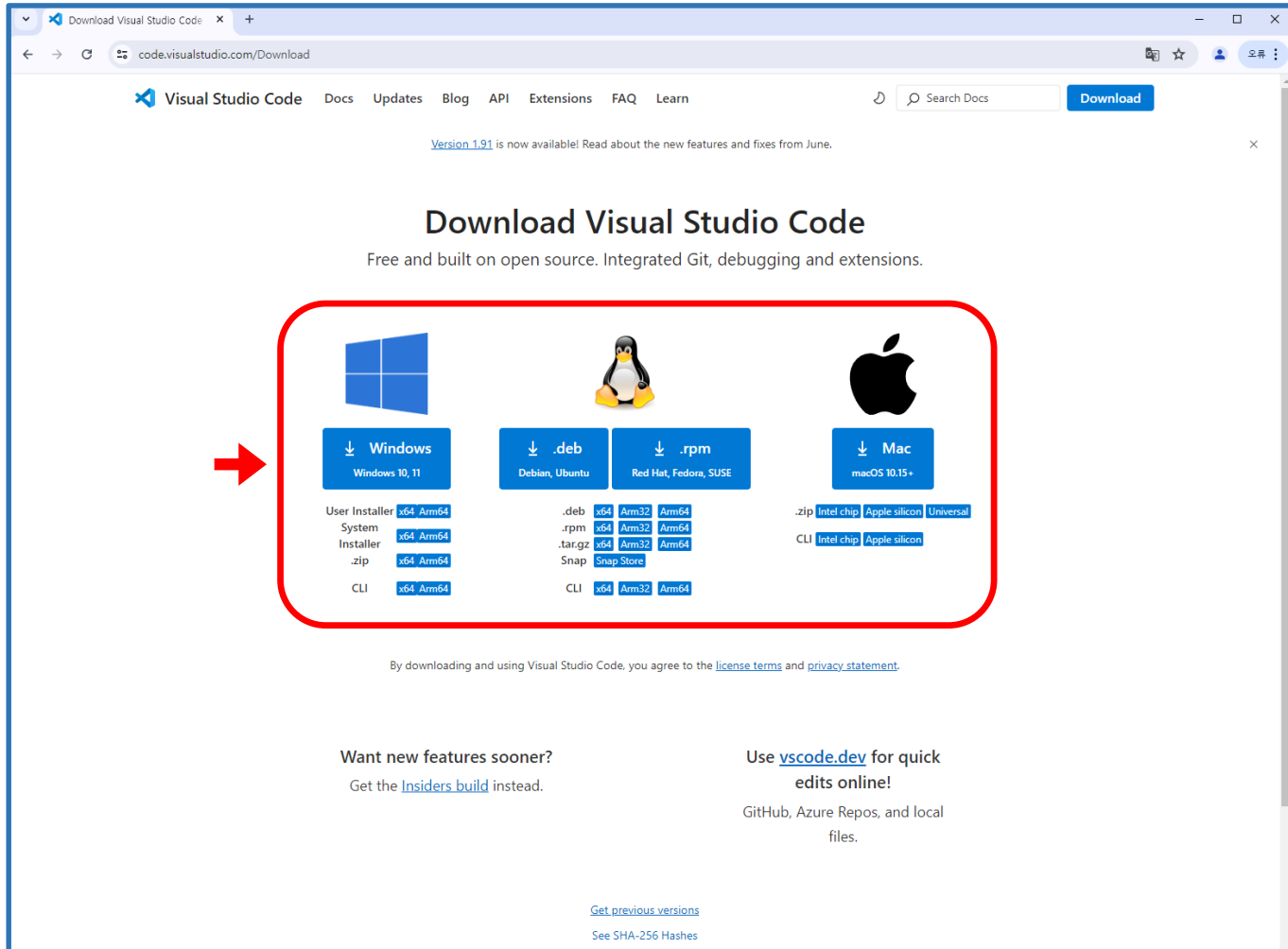# 1. Installation

■ Download Visual Studio Code (VScode)

- https://code.visualstudio.com/

1. Installation

- **Download Visual Studio Code (VScode)**

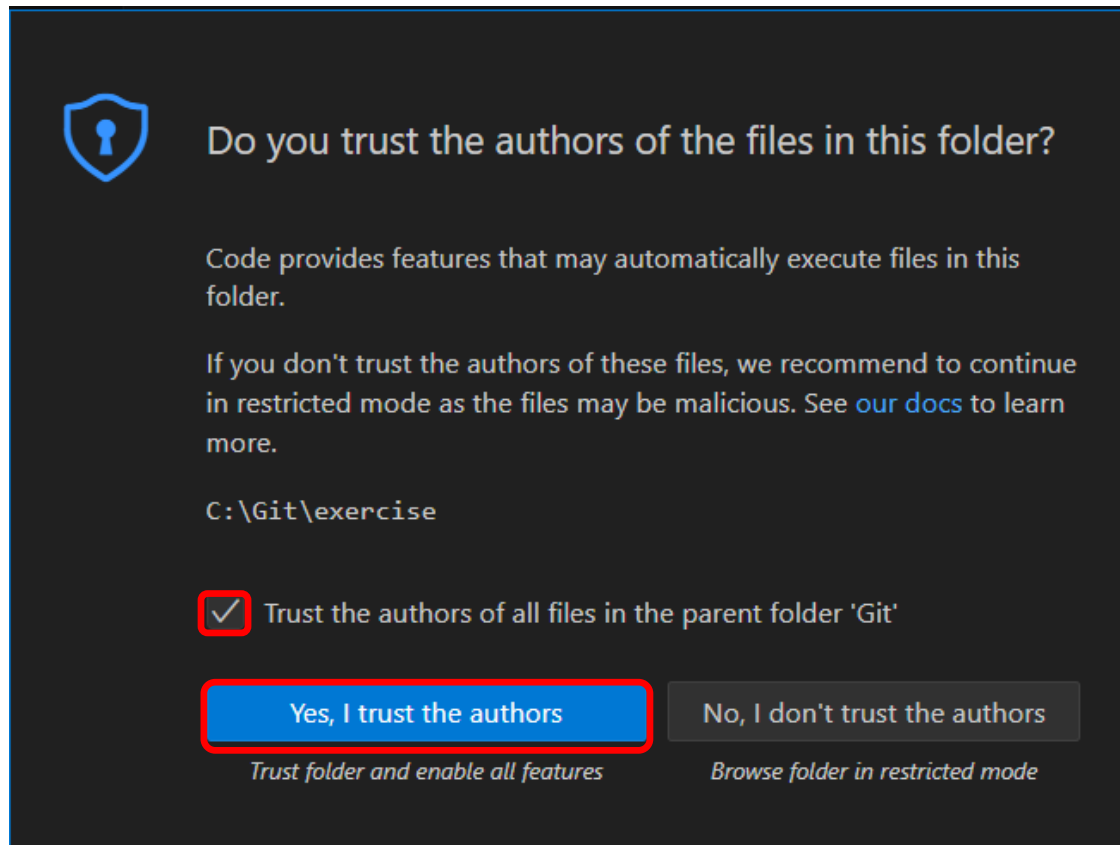 - https://code.visualstudio.com/Download

2. Setting

- **Make folder "exercise" in your directory**

- **Open this folder in VScode**
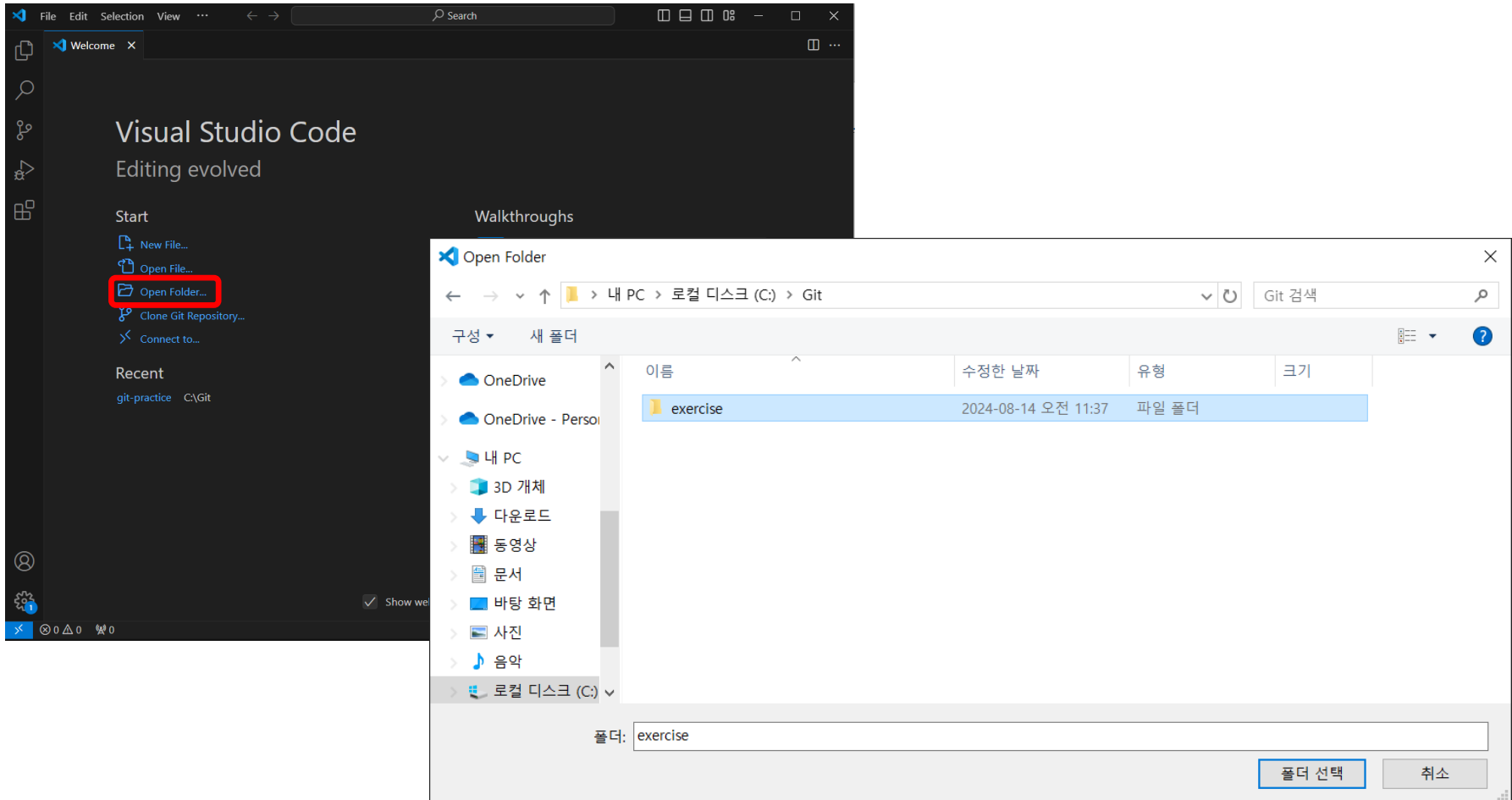
2. Setting

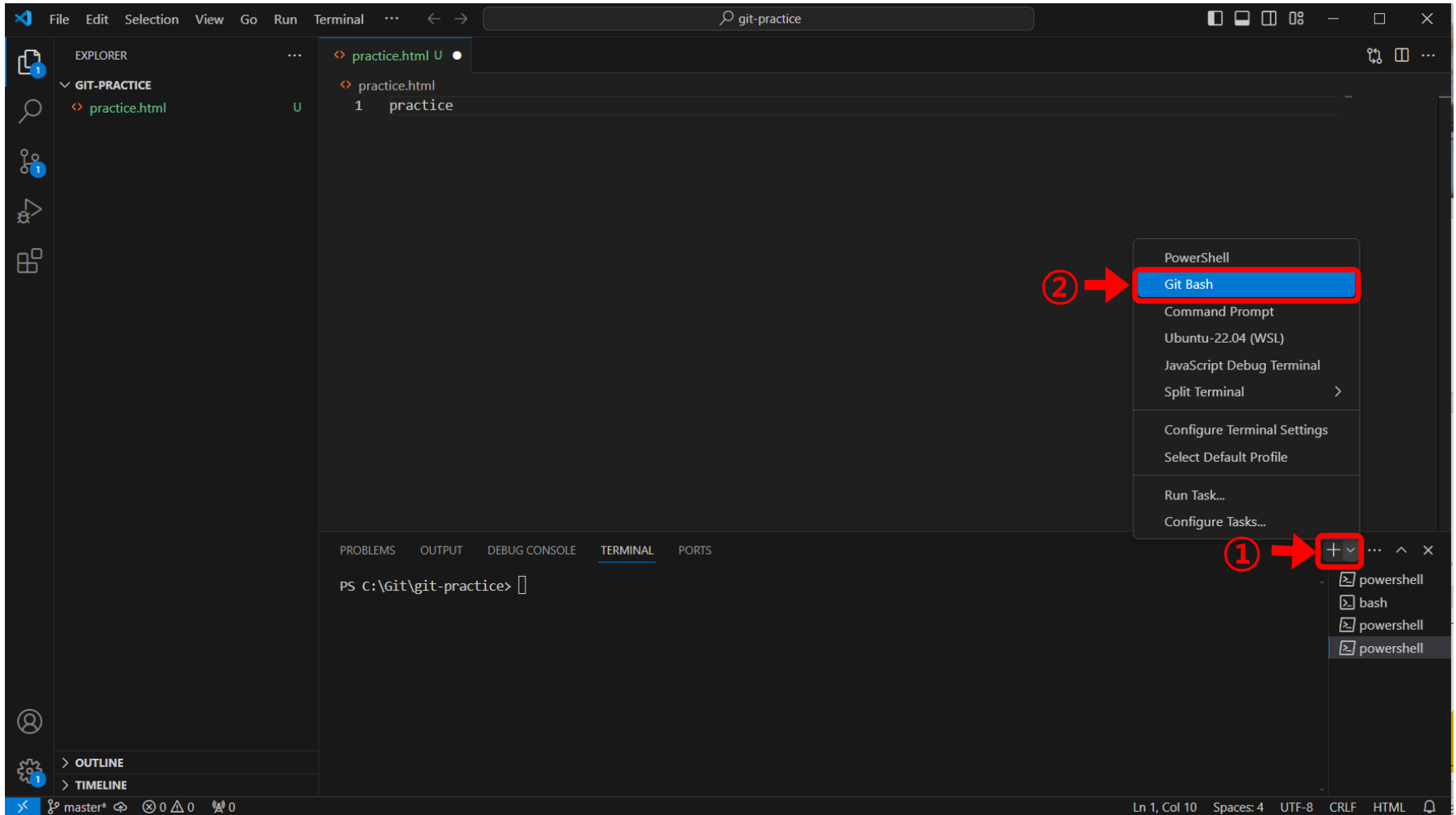- **Check "Trust the authors …"**

- **Yes, I trust the authors**

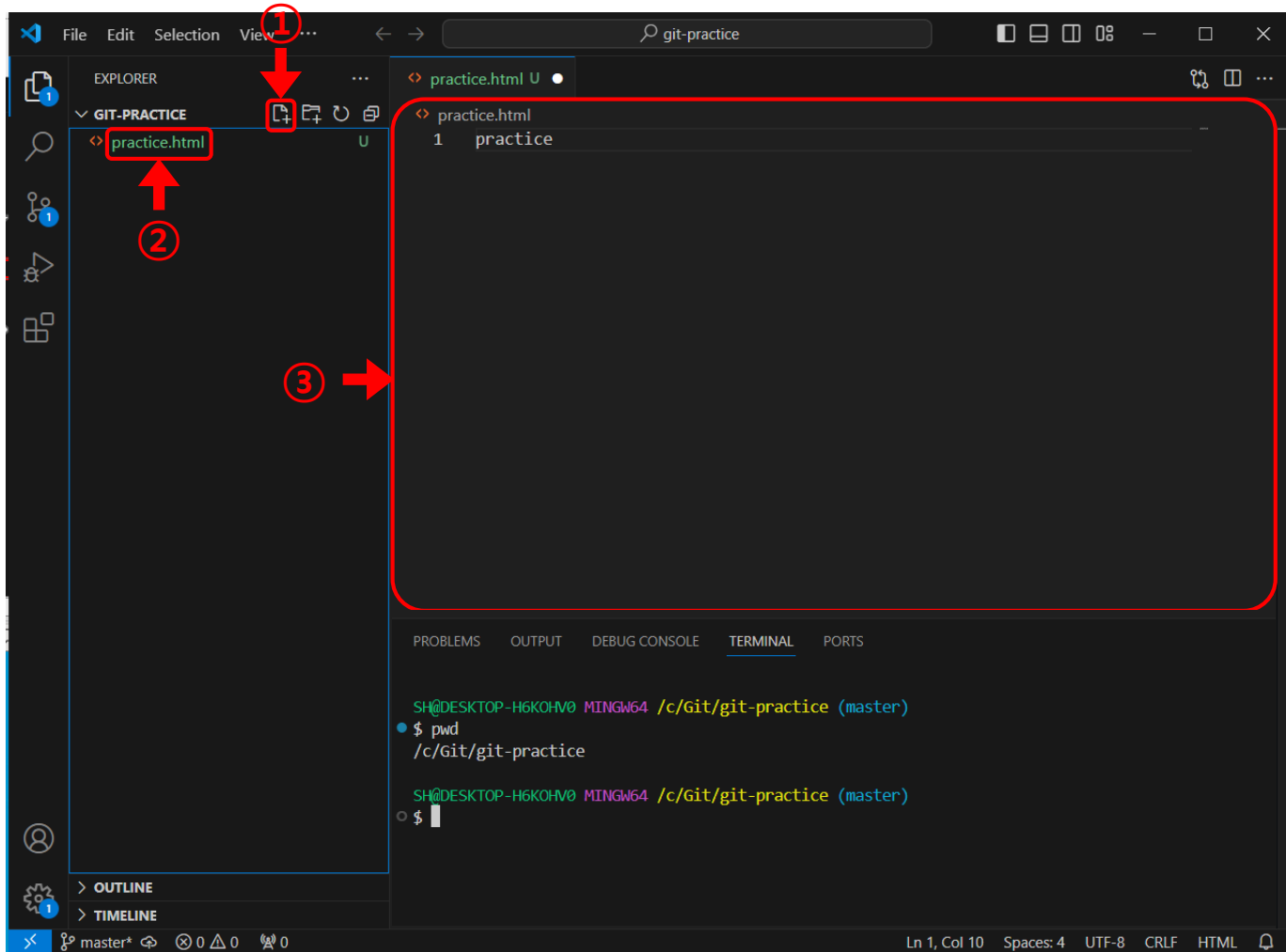# 2. Setting

- **Open "New Terminal" in VScode**
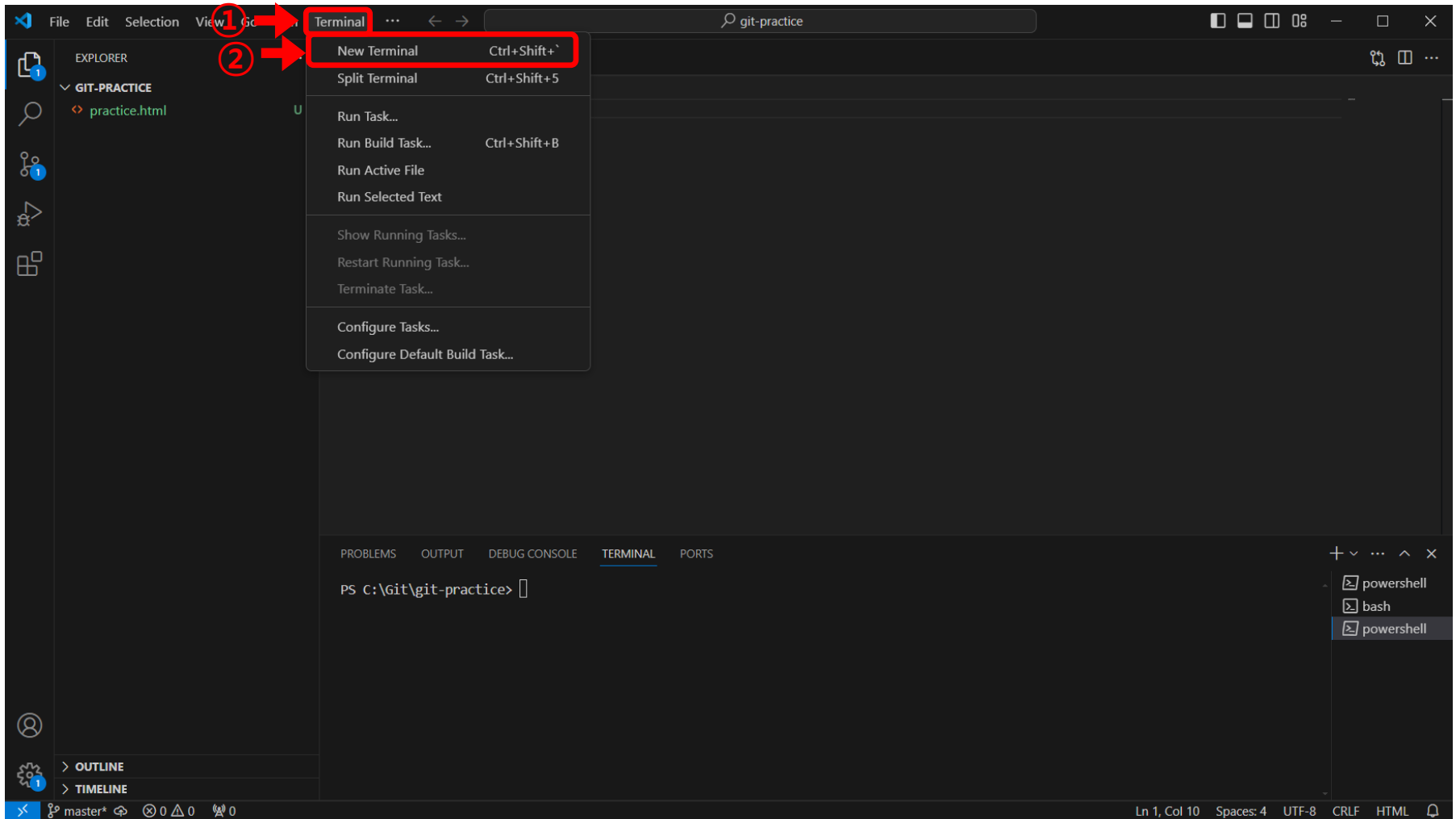
2. Setting

- **Change terminal to "Git Bash" or "Bash"**

2. Setting

- **Now you can add file (①, ②)**

- **Write code (③)**

4. Practical Exercise

1. Practical Exercise

- **Check the path of your directory**

- **Make your directory to a Git repository**

  - Set Git configuration

- **Initialize your Git repository**

- **Check the ".git" folder to make sure it has been created**

  - Check the list of files in your working directory

- **Check the name of your branch in the repository**

1. Practical Exercise

2. Practical Exercise

- **Create a new file "ex1.txt" in your directory**

  - Can create it using VScode or a terminal command

- **Check the status of your working directory**

- **Stage your new file in the index**

- **Commit your new file to your repository with the message "add ex1.txt"**

- **Check the history of your repository**

2. Practical Exercise

3. Practical Exercise

- **Write a code "Hello" in the file "ex1.txt"**

- **Check the status and difference of your working directory**
  - git status & git diff

- **Stage your new file in the index**

- **Check the status and differences of your working directory**
  - git status & git diff

3. Practical Exercise

3. Practical Exercise

4. Practical Exercise

- **Check the differences between the working directory and the repository**

- **Check the differences between the index and the repository**

- **Commit the previous change**
  - commit message: add first code in ex1.txt

- **Check the history of your repository**

4. Practical Exercise

4. Practical Exercise

5. Practical Exercise

- **Add a code "Welcome" in the file "ex1.txt"**

- **Expect the differences between the working directory, the index, and the repository and check the results**

- **Commit the changes**

5. Practical Exercise

6. Practical Exercise

- **Check the history and the file "ex1.txt"**

- **Hard reset the second commit in branch_2**

  - Second commit message: add first code in ex1.txt

- **Check the history of branch_2 and the file "ex1.txt"**

6. Practical Exercise

7. Practical Exercise

- **Check the branches in your repository**

- **Change your current branch name to "main"**
  - master -> main

- **Create a new branch named "branch_1"**

7. Practical Exercise
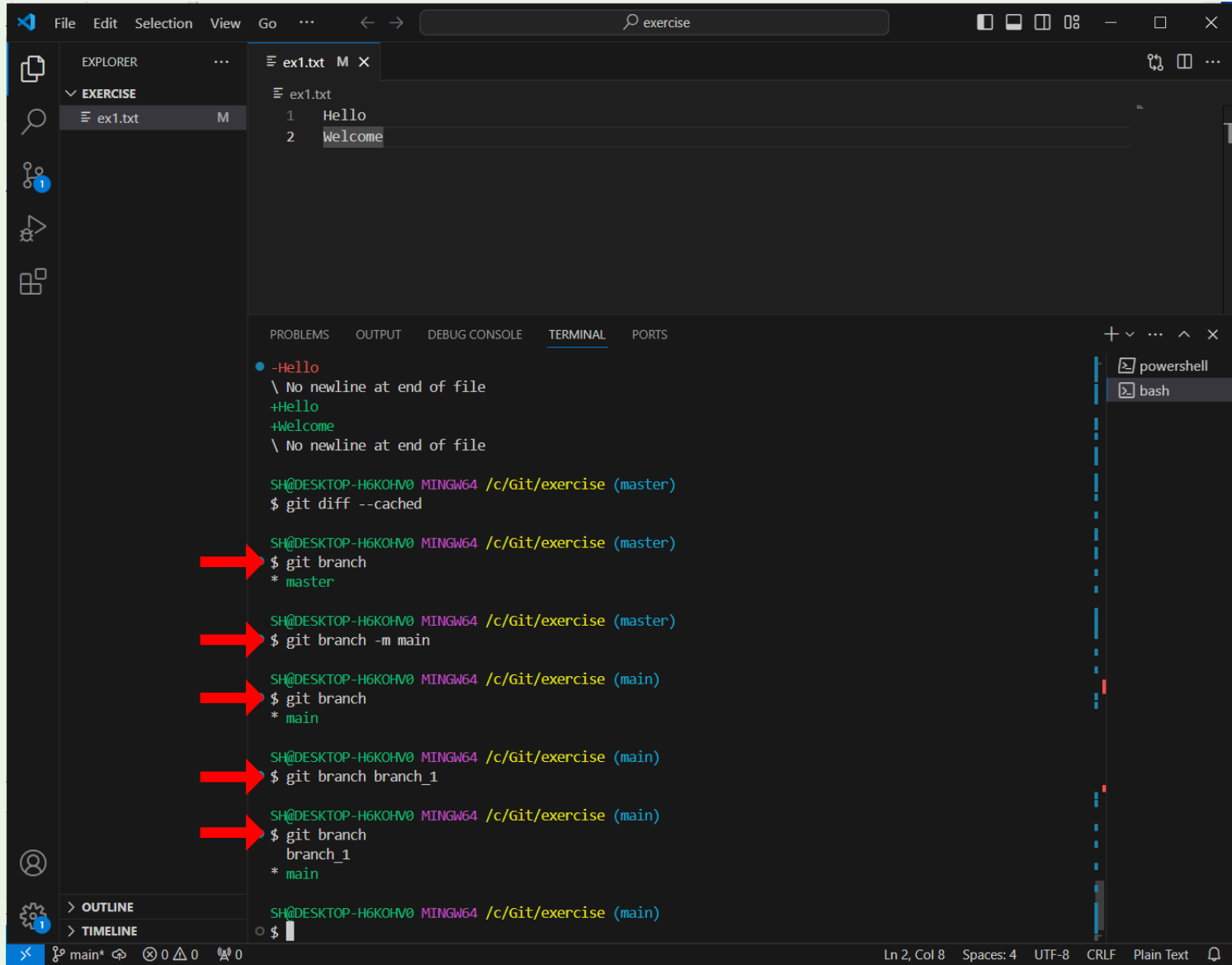
8. Practical Exercise

- **Switch the current branch to "branch_1"**

  - Use switch command

- **Check the history of the branch_1**

  - Check if it has the same history as its ancestor branch

- **Create and switch to a new branch named "branch_2"**

  - Use checkout command

8. Practical Exercise

9. Practical Exercise

- **Create a new file "ex2.txt" in branch "branch_1"**

- **Add a code "Hello" in the file "ex2.txt"**

- **Stage and commit the changes**
  - Commit message: "add ex2.txt"

- **Merge branch_1 into main**
  - Merge message: "merge branch_1 into main"

- **check the history of the main branch**

9. Practical Exercise

10.1 Practical Exercise

- **Create a new file "main1.txt" in branch "main"**

- **Stage and commit the changes**

  - Commit message: "add main1.txt in main"

- **Create new files "br1.txt" in branch "branch_1"**

- **Stage and commit the changes**

  - Commit message: "add br1.txt in branch_1"

- **Create new files "br2.txt" in branch "branch_1"**

- **Stage and commit the changes**

  - Commit message: "add br2.txt in branch_1"

10.2 Practical Exercise

- **Expect and check the differences in the history of the main branch and the branch_1**

- **Rebase branch_1 into main, and merge it**

  - Merge message: "merge branch_1 into main"

- **Check the history of the main branch**

10.1 Practical Exercise

10.2 Practical Exercise

10.2 Practical Exercise

Github

1. Setting

■ Login your ID in GitHub

- https://github.com/<ID>
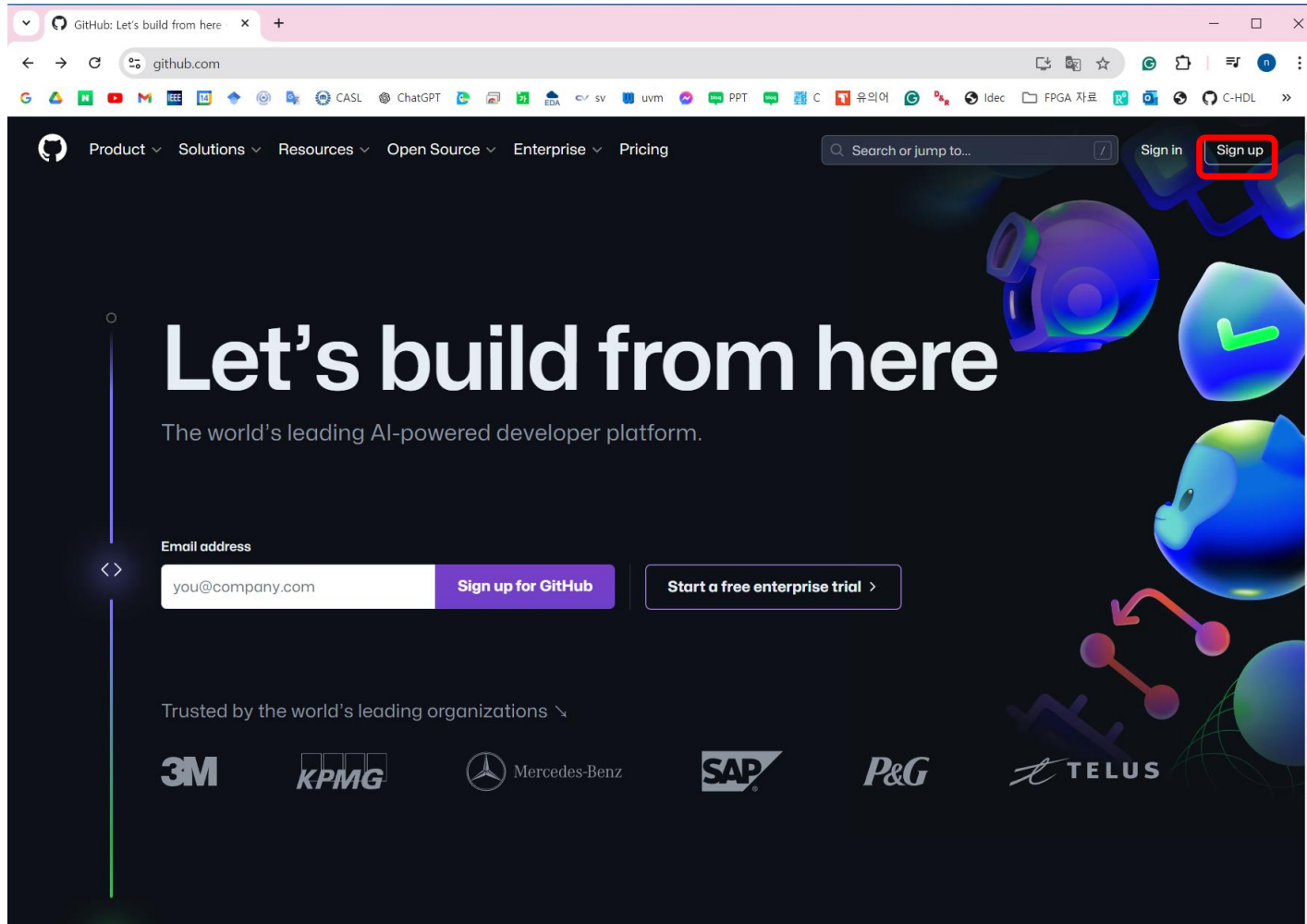
Setting

- **Create a new repository**

Setting

- **Set the repository name to "practice"**

Setting

- **Check your repository**
  - https://github.com/<ID>/<practice>.git
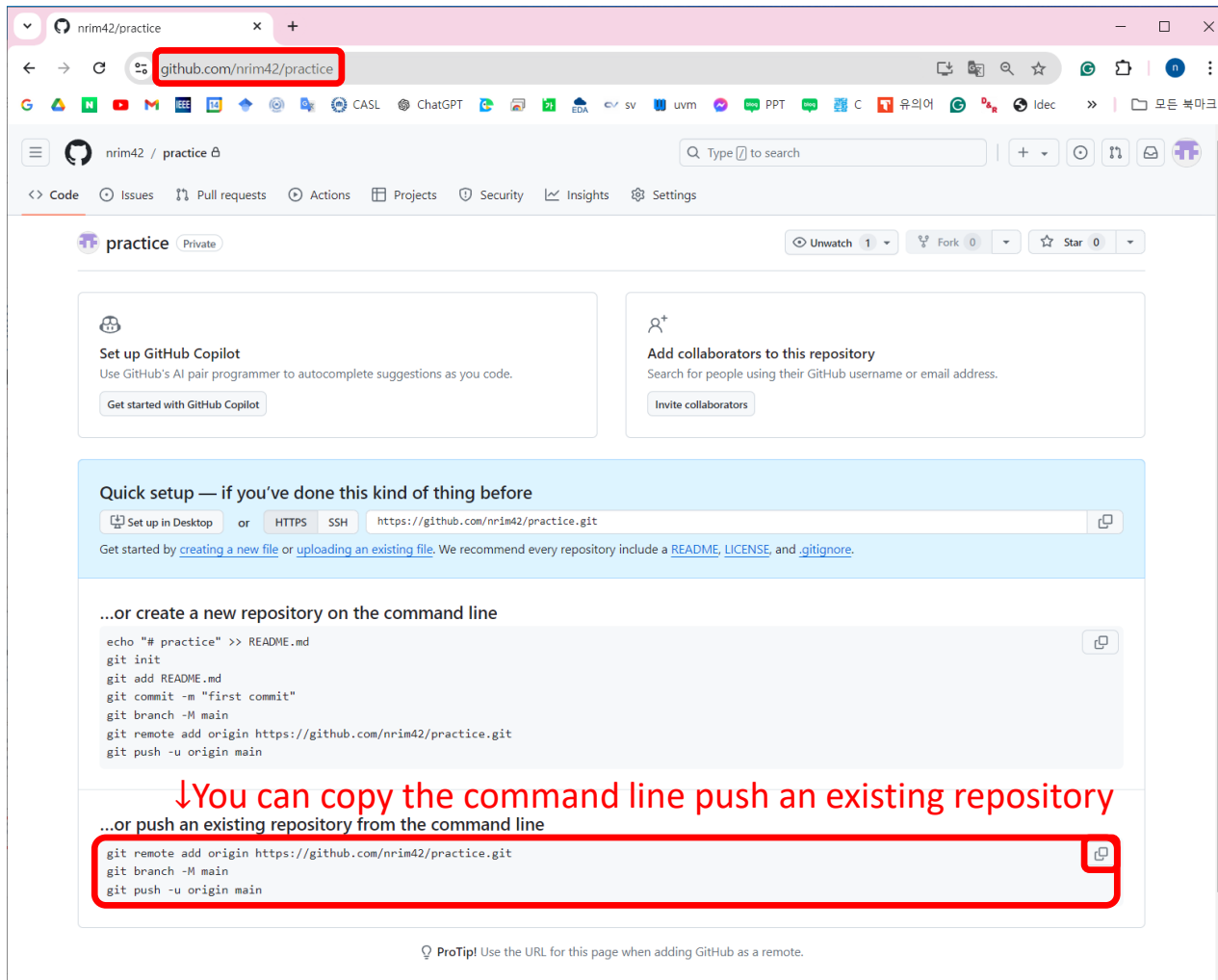
11. Practical Exercise

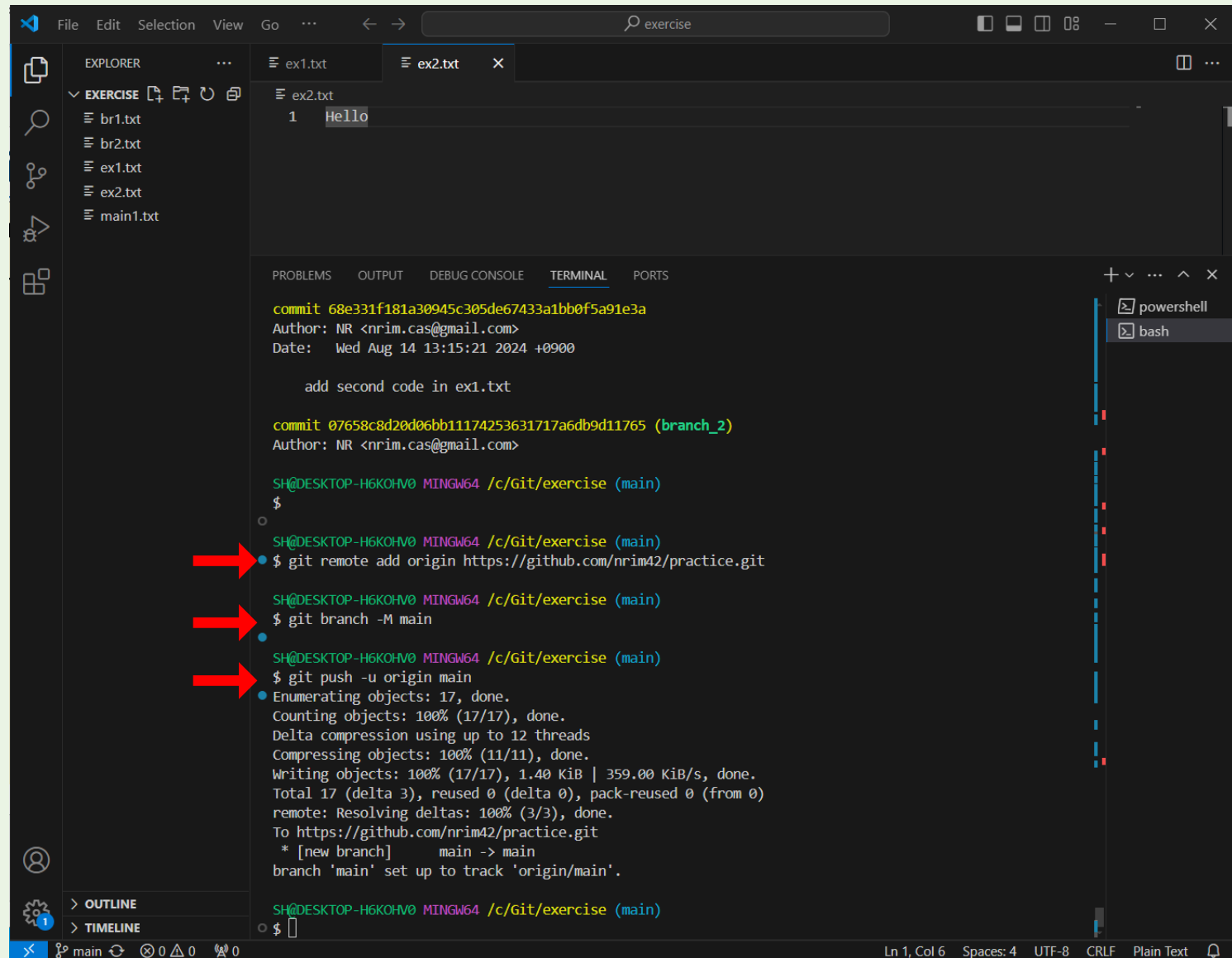- **What do the following commands do?**

  - git remote add origin https://github.com/nrim42/practice.git

  - git branch -M main

  - git push -u origin main

11. Practical Exercise

- **git remote add origin https://github.com/nrim42/practice.git**

  = Add a new remote repository (GitHub) to your local Git repository

- **git branch -M main**

  = Force to change the current branch name to "main"

- **git push -u origin main**

  = Push your local changes to a remote repository (GitHub) and set up tracking for the branch

  * You can use "git push/pull" in the future without having to specify the branch name

11. Practical Exercise

12. Practical Exercise

- **Edit code in your main branch**

- **Upload the changes to the GitHub**

12. Practical Exercise

Thank You