

# 비주얼 스튜디오 코드(vscode)로 깃 활용

Git & Github  
for pull request

강환수 교수



- 병합의 다른 방법인 rebase를 이해하고 수행할 수 있다.
  - Base를 재배치하는 rebase 병합을 수행할 수 있다.
  - 3-way 병합과 rebase의 차이를 이해할 수 있다.
  - 커밋이력 수정을 위해 대화방식인 옵션 -i를 사용해 rebase -i로 수행할 수 있다.

AI Experts  
Who Lead  
The Future

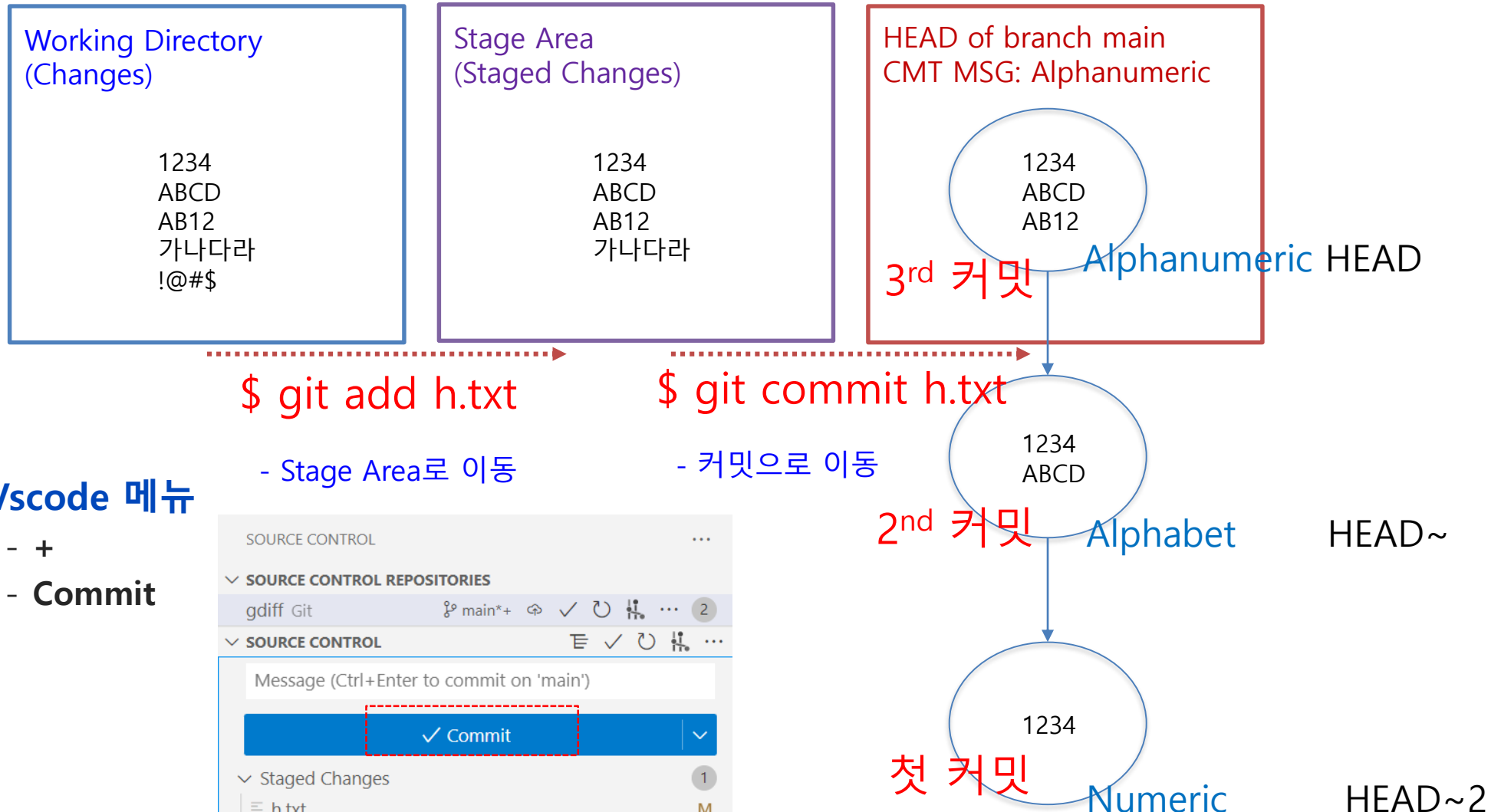
# 01

## Vscode로 깃 기초

- 폴더 생성
  - 저장소 초기화
  - 파일 생성
    - h.txt

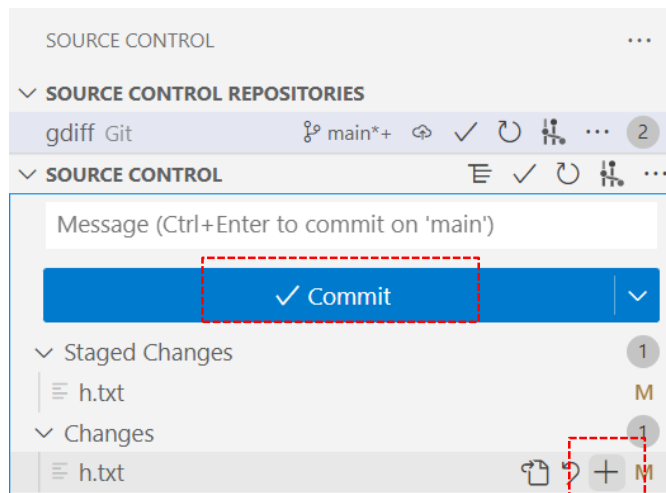
# 다음까지 add, commit

- 항상 add와 commit을 함께 가능
  - 구분해서도 가능

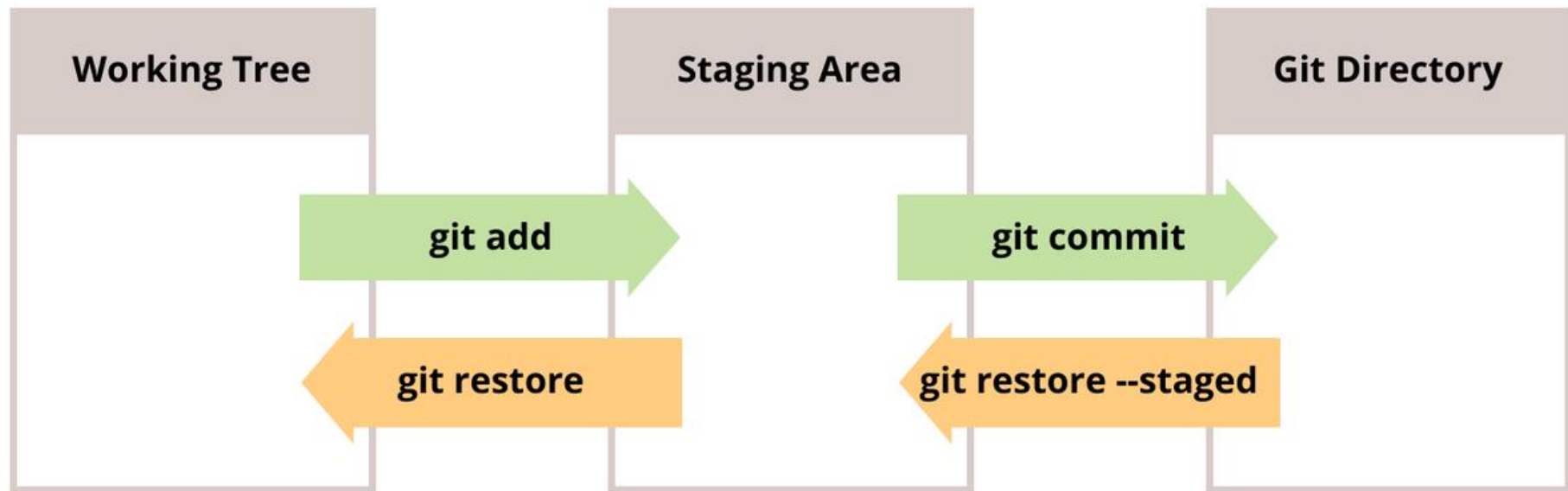


- Vscode 메뉴

- +
- Commit

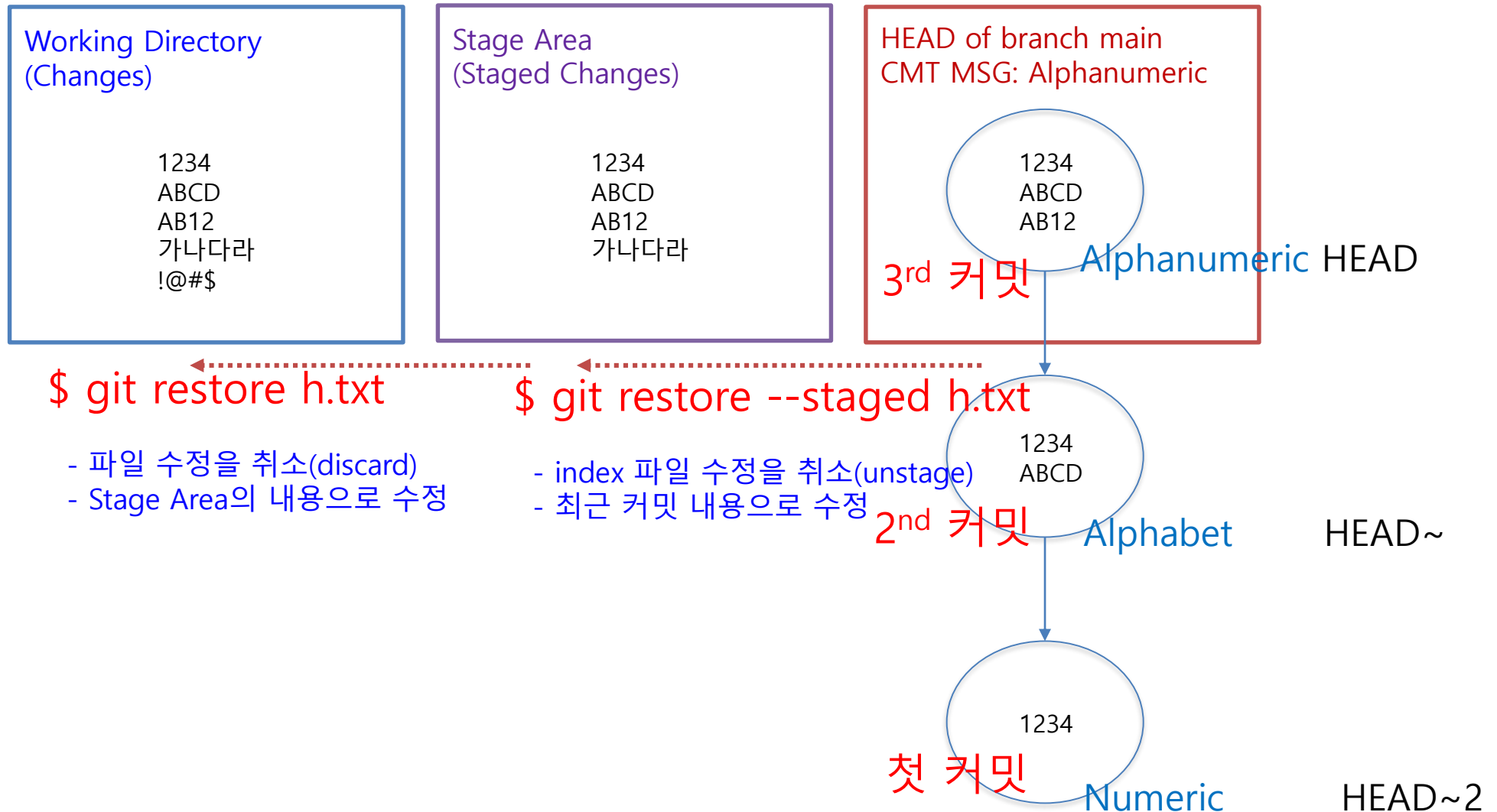


- `$ git restore`
- `$ git restore --staged`



- 명령어 **git restore**

- 옵션 **--staged** : index 파일 unstage(index를 최근 커밋으로 수정[복사])



- Vscod로 파악
  - Staged Changes
    - 작업 영역
  - Changes
    - 스테이지 영역

The screenshot shows the VS Code interface with the following components:

- Source Control Panel (Left):** Shows 'SOURCE CONTROL REPOSITORIES' with 'gdiff Git' and 'main\*+'. Below it, 'SOURCE CONTROL' shows 'Staged Changes' (1 file) and 'Changes' (1 file). A commit message box is visible with a 'Commit' button.
- Diff View (Center):** Shows a diff for 'h.txt'. The 'HEAD' (main) version has lines 1-3 (1234, ABCD, AB12). The 'index' version has lines 1-4 (1234, ABCD, AB12, 가나다라). The diff shows a green bar for the new line in the index.
- Git Graph (Right):** Shows 'Uncommitted Changes (1)' for the 'main' branch. The commit history table is as follows:

Graph	Description	Date	Author	Commit
Uncommitted Changes (1)		19 Nov 2022 1...	*	*
main	Alphanumeric	19 Nov 2022 1...	ai7dnn	92920efa
	Alphabhet	19 Nov 2022 1...	ai7dnn	f7962d57
	digit	19 Nov 2022 1...	ai7dnn	aefe6edd

Below the diff view, a terminal window shows the following commands and output:

```
PC@DESKTOP-482NOAB MINGW64 /c/OSS/git/vscode/gdiff (main)
$ git status -s
MM h.txt

PC@DESKTOP-482NOAB MINGW64 /c/OSS/git/vscode/gdiff (main)
$
```



- 현재의 index 파일로 수정

- \$ git restore h.txt

PC@DESKTOP-482NOAB MINGW64 /c/OSS/git/vscode/gdiff (main)

\$ git status

On branch main

Changes to be committed:

(use "git restore --staged <file>..." to unstage)

modified: h.txt

Changes not staged for commit:

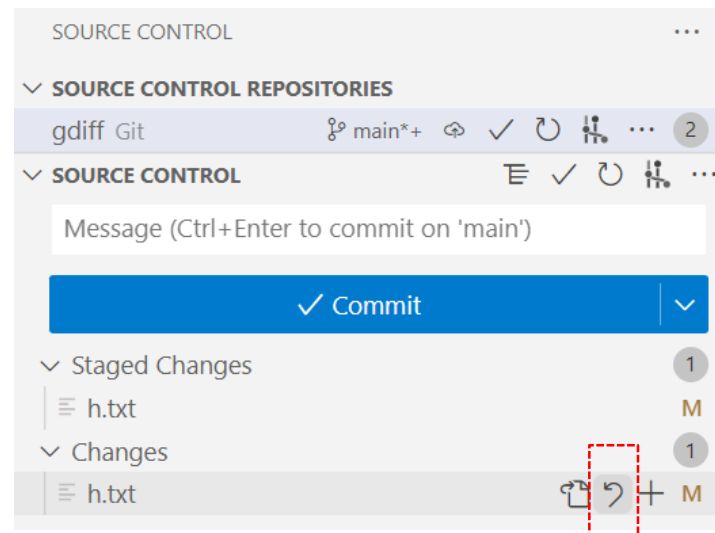
(use "git add <file>..." to update what will be committed)

(use "git restore <file>..." to discard changes in working directory)

modified: h.txt

- 메뉴

- Discard changes



- 현재의 index 파일로 수정

- \$ git restore h.txt

PC@DESKTOP-482NOAB MINGW64 /c/OSS/git/vscode/gdiff (main)

\$ git status

On branch main

Changes to be committed:

(use "git restore --staged <file>..." to unstage)

modified: h.txt

Changes not staged for commit:

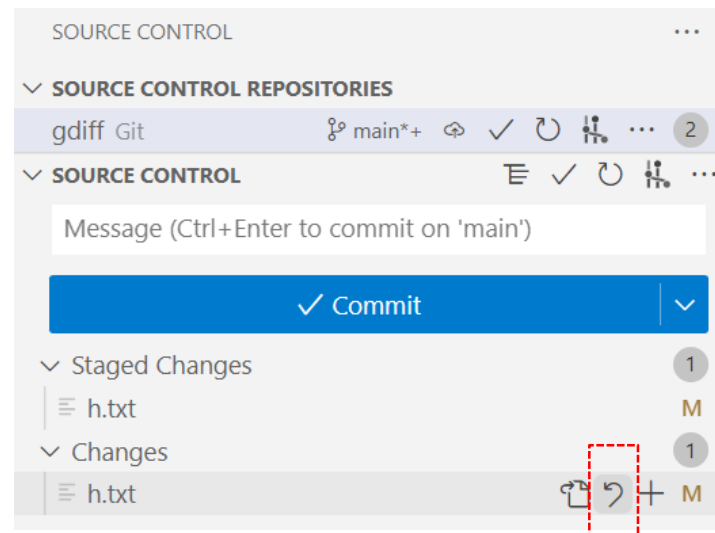
(use "git add <file>..." to update what will be committed)

(use "git restore <file>..." to discard changes in working directory)

modified: h.txt

- 메뉴

- Discard Changes



- 현재의 최근 커밋 파일로 수정, unstage

- \$ git restore --staged h.txt

PC@DESKTOP-482NOAB MINGW64 /c/OSS/git/vscode/gdiff (main)

\$ git status

On branch main

Changes to be committed:

(use "git restore --staged <file>..." to unstage)

modified: h.txt

Changes not staged for commit:

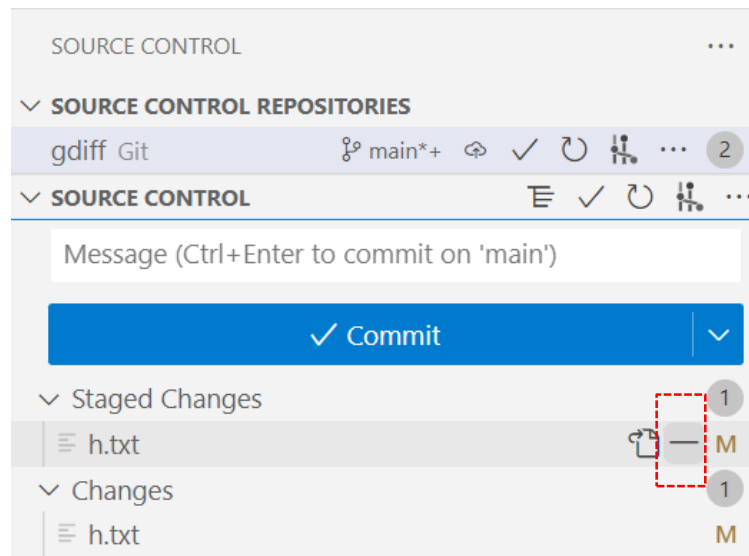
(use "git add <file>..." to update what will be committed)

(use "git restore <file>..." to discard changes in working directory)

modified: h.txt

- 메뉴

- Unstage Changes

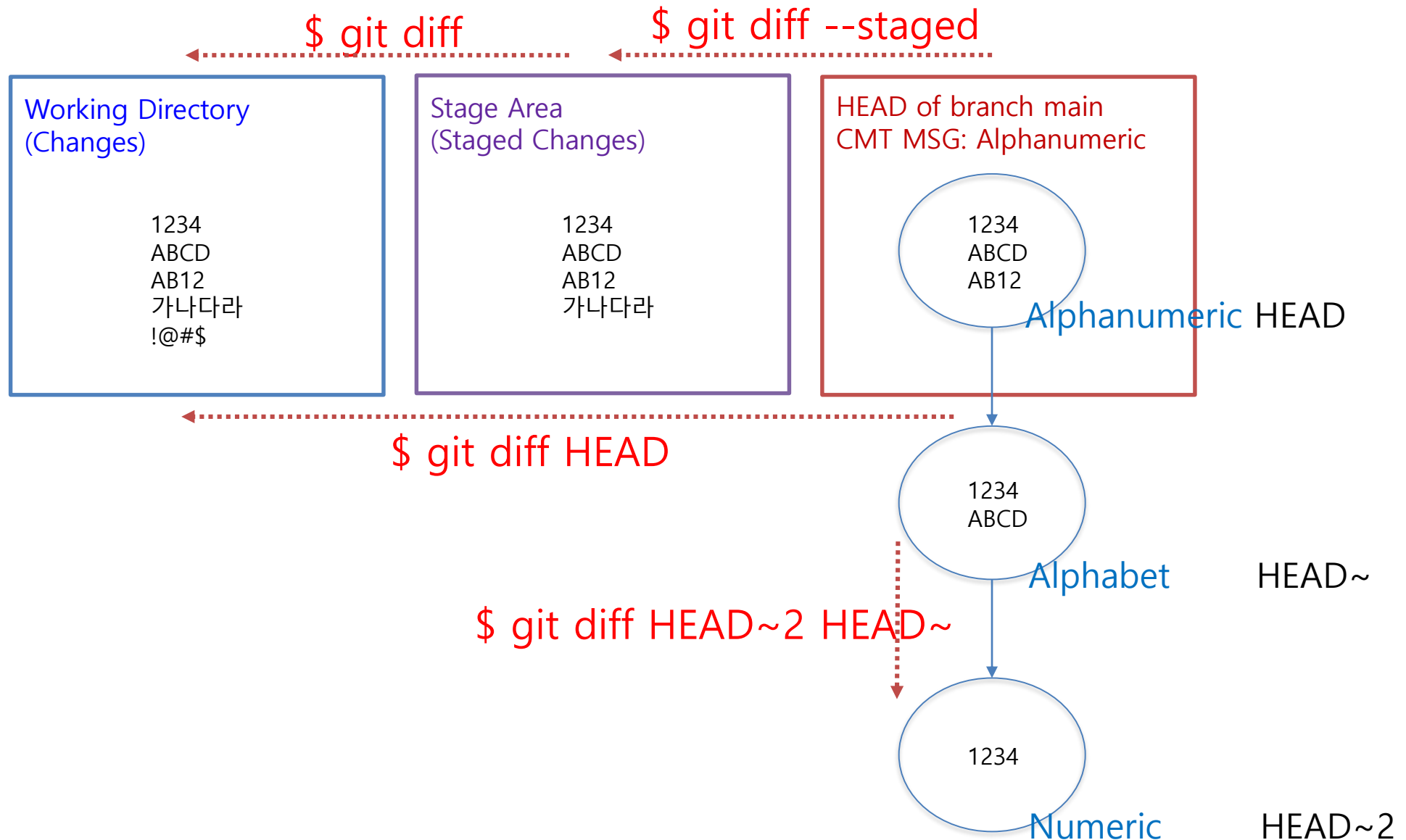


AI Experts  
Who Lead  
The Future

## 02

파일 비교 git diff

- 현재의 작업 영역과 스테이지 영역, 그리고 커밋 로그



- 다음 상태까지 수행 해 보세요

The screenshot shows the VS Code interface with the following components:

- SOURCE CONTROL:** Shows 'gdiff' as the current repository. The 'Commit' button is visible. The 'Staged Changes' section shows 'h.txt' with a message 'Message (Ctrl+Enter to commi...)'. The 'Changes' section also shows 'h.txt'.
- h.txt:** A file with the following content:

```
1 1234
2 ABCD
3 AB12
4 가나다라
5 !@#$
6
```
- Git Graph:** Shows the 'main' branch with a commit history table:

Graph	Description	Date	Author	Commit
Uncommitted Changes (1)		19 Nov ...	*	*
main	Alphanumeric	19 Nov ...	ai7dnn	92920efa
	Alpahbet	19 Nov ...	ai7dnn	f7962d57
	digit	19 Nov ...	ai7dnn	aefe6edd
- TERMINAL:** Shows the following commands and output:

```
PC@DESKTOP-482NOAB MINGW64 /c/OSS/git/vscode/gdiff (main)
$ git log --oneline --graph --all
* 92920ef (HEAD -> main) Alphanumeric
* f7962d5 Alpahbet
* aefe6ed digit

PC@DESKTOP-482NOAB MINGW64 /c/OSS/git/vscode/gdiff (main)
$ git status -s
MM h.txt

PC@DESKTOP-482NOAB MINGW64 /c/OSS/git/vscode/gdiff (main)
$
```

# \$ git diff

오픈소스 소프트웨어를 위한 깃과 깃허브 Python language

- Changes 하부 파일 h.txt 클릭

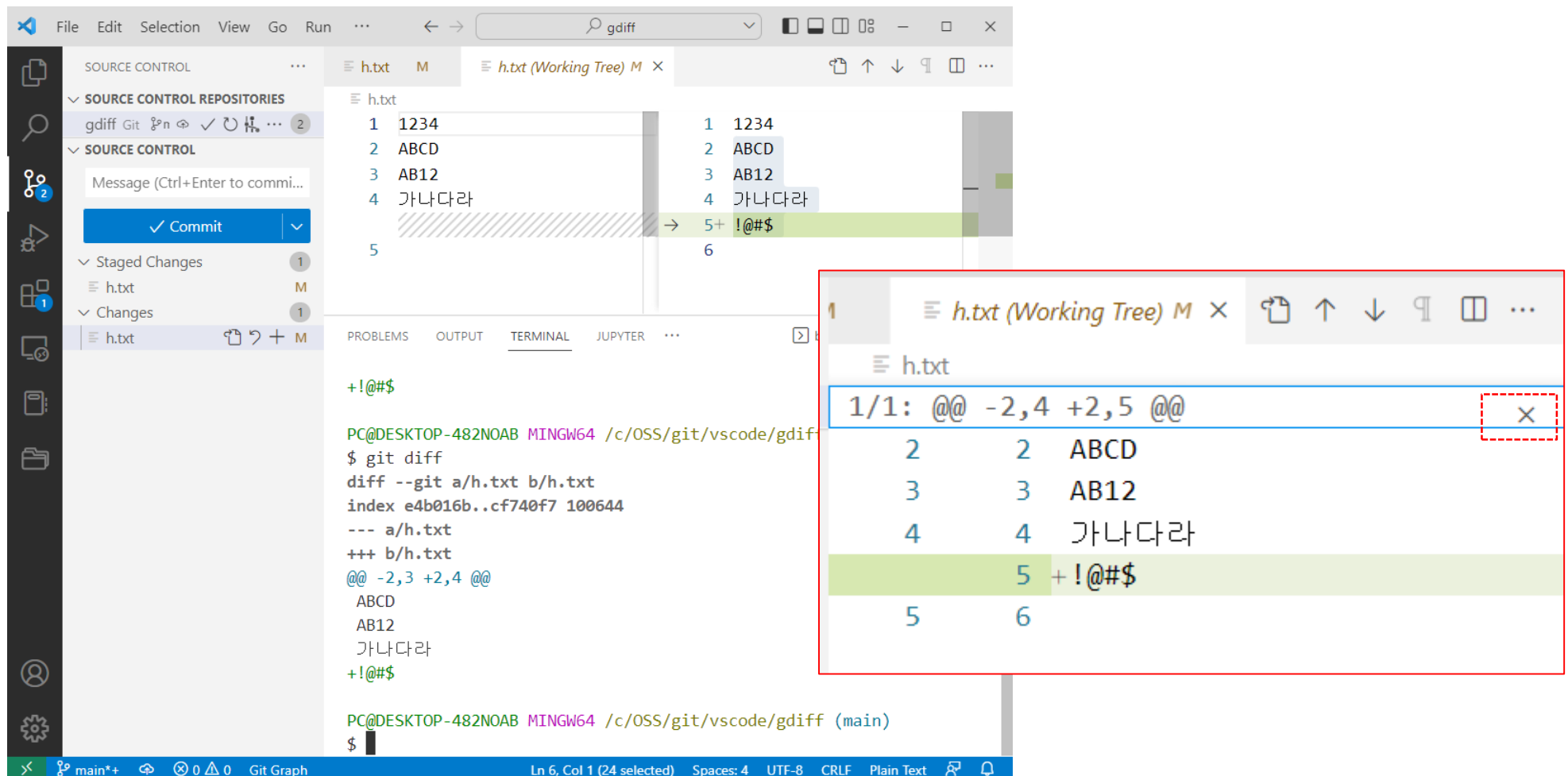
The screenshot shows the VS Code interface with the following components:

- Source Control Sidebar:** Shows 'SOURCE CONTROL REPOSITORIES' with 'gdiff' selected. Under 'SOURCE CONTROL', 'h.txt' is listed under 'Changes'.
- Diff Viewer:** Displays a comparison between two versions of 'h.txt'. The left version (index) has lines 1-5, and the right version (working tree) has lines 1-6. Line 5 in the left version is highlighted with a red dashed box.
- Git Graph Sidebar:** Shows a graph of commits. The 'main' branch is selected, and the commit 'Alphanu...' is highlighted.
- Terminal:** Shows the output of the command `$ git diff`. The output is as follows:

```
$ git diff
diff --git a/h.txt b/h.txt
index e4b016b..cf740f7 100644
--- a/h.txt
+++ b/h.txt
@@ -2,3 +2,4 @@
ABCD
AB12
가나다라
+!@#$

PC@DESKTOP-482NOAB MINGW64 /c/OSS/git/vscode/gdiff (main)
$
```

- 파일 비교가 보이는 창에서 F7 클릭
  - Git diff 명령 결과와 비슷한 결과를 보이며 파일 비교를 보여줌
    - X를 누르면 제거





# \$ git diff -staged

오픈소스 소프트웨어를 위한 깃과 깃허브 Python language

- Staged Changes 하부 파일 h.txt 클릭
  - F7으로 보기와 x로 삭제

The screenshot shows the VS Code interface with the Source Control view on the left. The 'Staged Changes' section shows 'h.txt' with a diff icon. The main editor displays the diff of 'h.txt', showing line 4 highlighted in green. A red box highlights the diff output in the terminal, showing the diff command and its output.

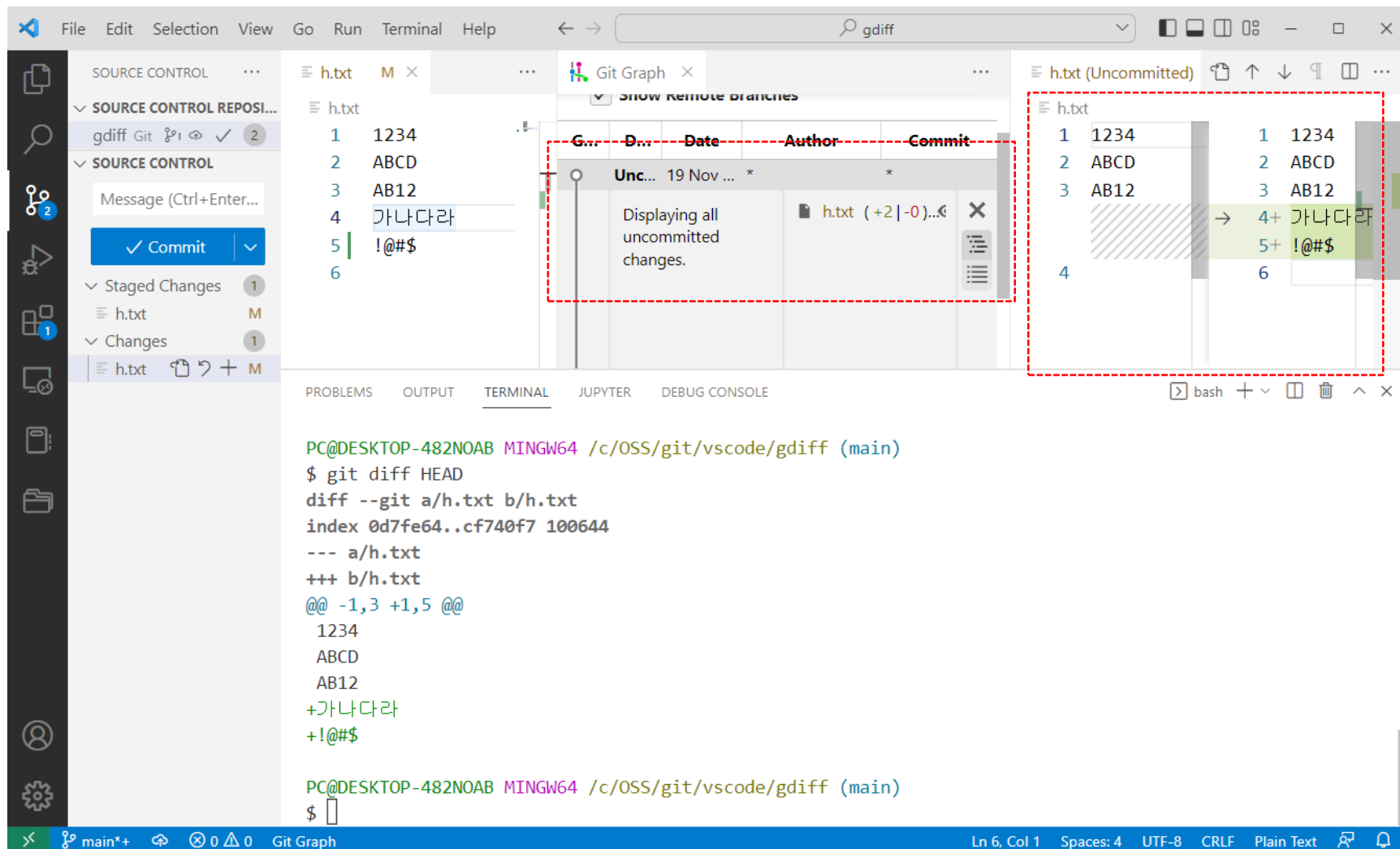
```
PC@DESKTOP-482NOAB MINGW64 /c/OSS/git/vscode $ git diff --staged
diff --git a/h.txt b/h.txt
index 0d7fe64..e4b016b 100644
--- a/h.txt
+++ b/h.txt
@@ -1,3 +1,4 @@
1234
ABCD
AB12
+가나다라

PC@DESKTOP-482NOAB MINGW64 /c/OSS/git/vscode/gdiff (main) $
```

# \$ git diff HEAD

오픈소스 소프트웨어를 위한 깃과 깃허브 Python language

- Git Graph의 최상단 Uncommitted Changes를 눌러 파일 h.txt를 클릭
  - HEAD와 changes 비교
    - F7으로 보기와 x로 삭제



# \$ git diff HEAD~2 HEAD^

오픈소스 소프트웨어를 위한 깃과 깃허브 Python language

- Git Graph의 Alphabet 커밋을 눌러 파일 h.txt를 클릭
  - 클릭한 커밋의 이전(HEAD~2)과 클릭한 커밋(HEAD^)을 비교
    - F7으로 보기와 x로 삭제

The screenshot shows the VS Code Git Graph interface. The left sidebar shows the 'SOURCE CONTROL' view with 'h.txt' selected. The main area shows the 'Graph' view with a commit history. The 'Alphabet' commit is selected, and the diff between 'HEAD~2' and 'HEAD^' is shown. The diff shows a change from '1234' to 'ABCD' in file 'h.txt'. The terminal window at the bottom shows the command 'git diff HEAD~2 HEAD^' and its output:

```
PC@DESKTOP-482NOAB MINGW64 /c/OSS/git/vscode/gdiff (main)
$ git diff HEAD~2 HEAD~
diff --git a/h.txt b/h.txt
index 81c545e..51c03ff 100644
--- a/h.txt
+++ b/h.txt
@@ -1,2 @@
 1234
+ABCD
```

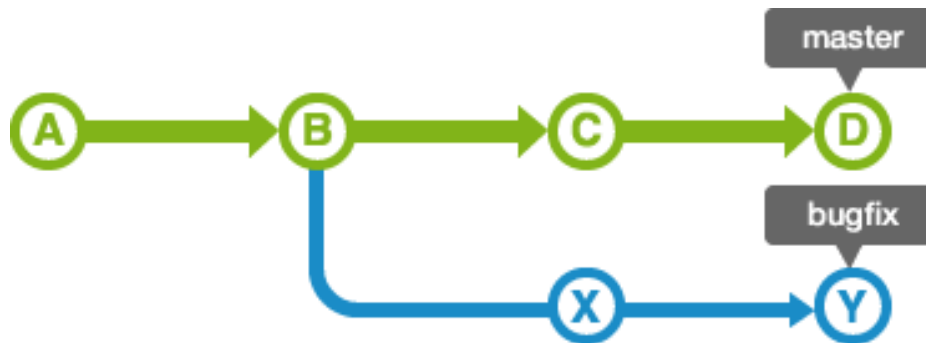
AI Experts  
Who Lead  
The Future

# 03

---

## 병합 rebase

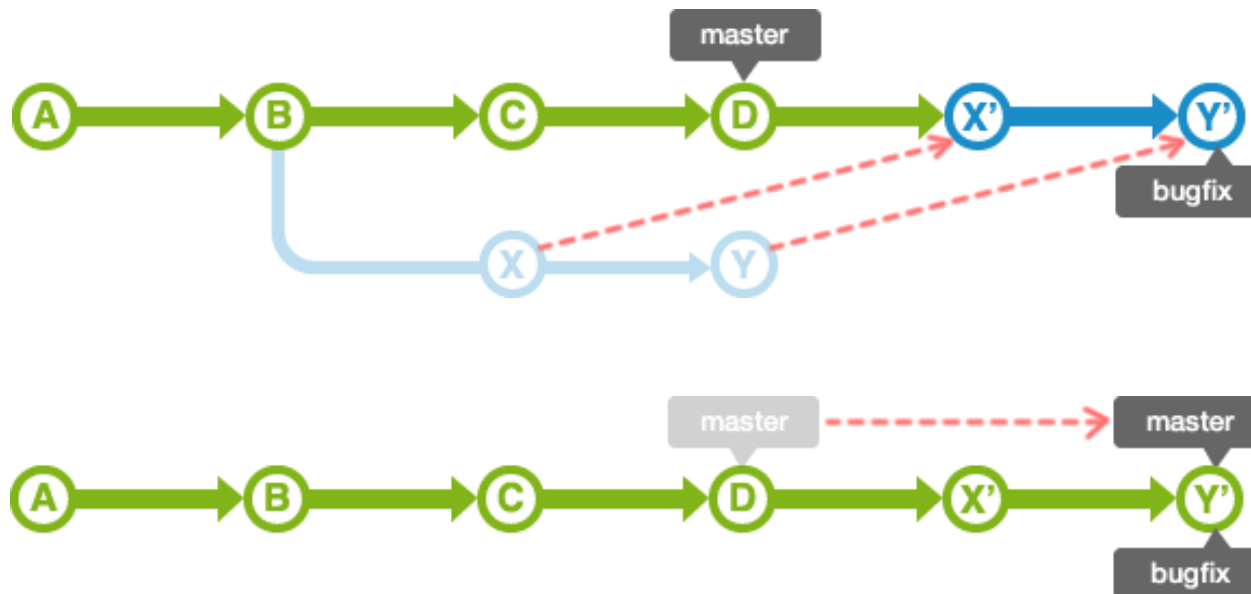
- 'master' 브랜치에서 분기하는 'bugfix' 브랜치가 있다고 가정



- Rebase 병합
  - 명령 rebase로 base를 수정하고
    - B에서 D로 수정, D 이후에 bugfix를 배치
  - 다시 'fast-forward 병합' 수행: 이 병합을 직접 다시 해야함
    - 'bugfix' 브랜치의 이력이 'master' 브랜치 뒤로 이동
      - 이력이 하나의 줄기로 이어짐



- 충돌 발생 가능
  - 이동되는 X와 Y의 내용이 'master'의 C, D 커밋들과 충돌하는 부분이 생길 수 있음
  - 각각의 커밋에서 발생한 충돌 내용을 수정 후, 추가, 계속 수행 필요
- 충돌 발생 후 해결 절차
  - 1. 파일 수정
  - 2. 파일 추가
    - `$ git add <수정파일>`
  - 3. rebase 계속 수행, 마지막 메시지 메시지 수정
    - `$ git rebase --continue`



# 다음에서 rebase

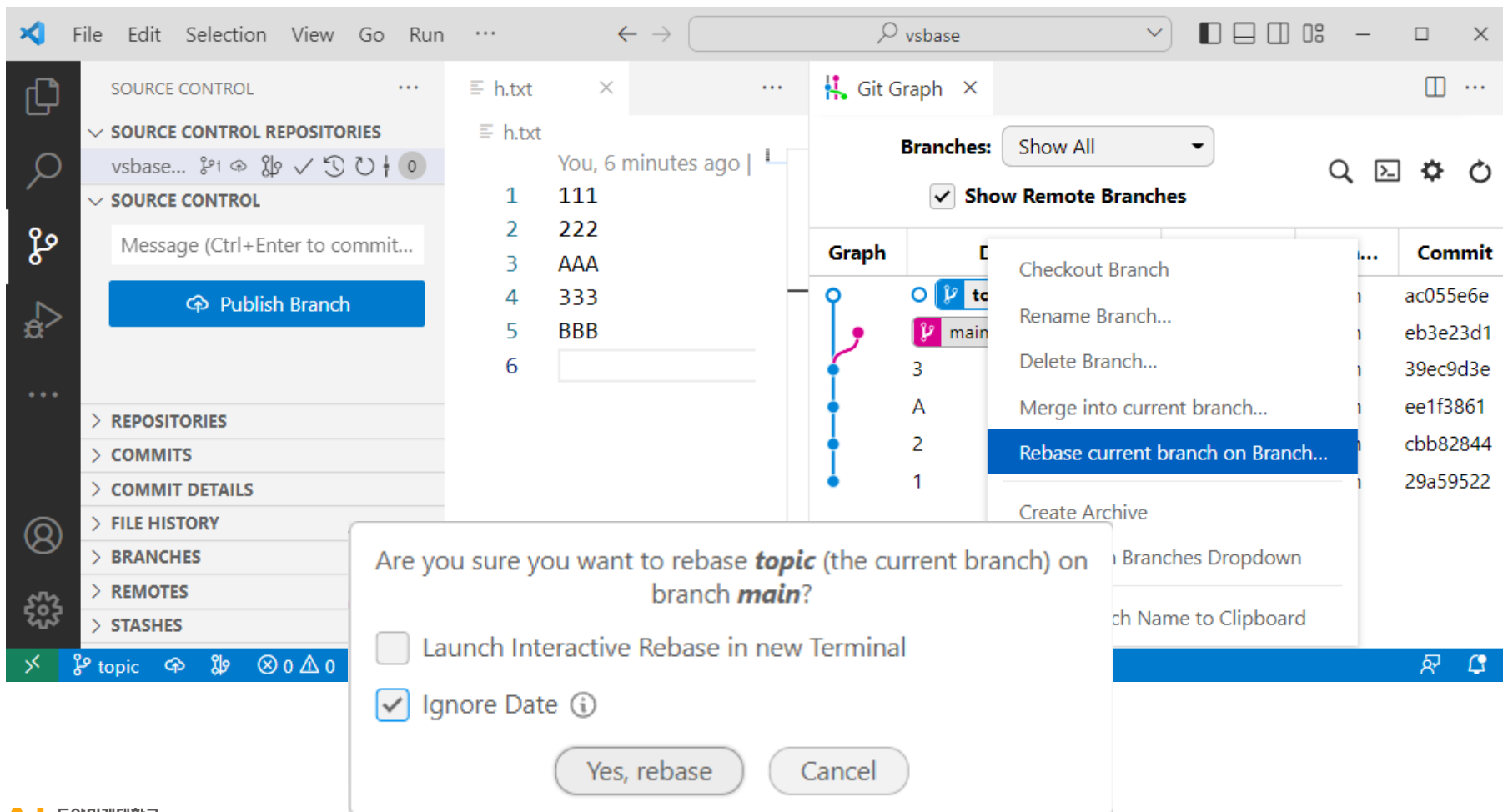
오픈소스 소프트웨어를 위한 깃과 깃허브 Python language

- 현재 브랜치 topic
  - \$ git rebase main

VS Code interface showing the Git Graph view. The left sidebar shows the Source Control panel with the 'topic' branch selected. The main editor shows the 'h.txt' file with content: 1 111, 2 222, 3 AAA, 4 333, 5 BBB, 6. The Git Graph view on the right shows a commit history with a 'topic' branch (blue) and a 'main' branch (pink). The 'topic' branch has a commit 'B' (ac055e6e) and the 'main' branch has a commit '4' (eb3e23d1).

Graph	Description	Date	Auth...	Commit
topic B		18 Nov 202...	ai7dnn	ac055e6e
main 4		18 Nov 202...	ai7dnn	eb3e23d1
3		18 Nov 202...	ai7dnn	39ec9d3e
A		18 Nov 202...	ai7dnn	ee1f3861
2		18 Nov 202...	ai7dnn	cbb82844
1		18 Nov 202...	ai7dnn	29a59522

- 브랜치 main을 누르고 팝업 메뉴에서
  - Rebase current branch on Branch... 선택
    - Launch interactive Rebase in new Terminal
      - 선택하지 않으면 커밋 메시지 수정 못함





Visual Studio Code interface showing a merge conflict in the file `h.txt`. The conflict is between the current HEAD (Current Change) and an incoming change from branch B (Incoming Change). The conflict is resolved in the Merge Editor.

The file `h.txt` content is as follows:

```

1 111
2 222
3 AAA
4 333
5 <<<<<<< HEAD (Current Change)
6 444
7 =====
8 BBB
9 >>>>>> ac055e6 (B) (Incoming Change)
10

```

The terminal output shows the following error message:

```

CONFLICT (content): Merge conflict in h.txt
error: could not apply ac055e6... B
hint: Resolve all conflicts manually, mark them as resolved with
hint: "git add/rm <conflicted_files>", then run "git rebase --continue".
hint: You can instead skip this commit: run "git rebase --skip".
hint: To abort and get back to the state before "git rebase", run "git re
--abort".
Could not apply ac055e6... B

```

The Git Graph panel on the right shows the commit history, including the commit `ac055e6` (B) (Incoming Change) which is the source of the conflict.

The screenshot shows the VS Code interface during a Git merge. The **SOURCE CONTROL** sidebar on the left shows the repository 'vsbase...' with a conflict in 'h.txt'. The **Git Graph** sidebar on the right shows the commit history. The main editor shows the file 'h.txt' with a conflict between the local branch 'B' and the remote branch 'main'. The conflict is highlighted with a red dashed box around the '+' icon in the Source Control sidebar. The terminal at the bottom displays the following messages:

```
CONFLICT (content): Merge conflict in h.txt
error: could not apply ac055e6... B
hint: Resolve all conflicts manually, mark them as resolved with
hint: "git add/rm <conflicted_files>", then run "git rebase --continue".
hint: You can instead skip this commit: run "git rebase --skip".
hint: To abort and get back to the state before "git rebase", run "git re
--abort".
Could not apply ac055e6... B
```

A blue button labeled "Resolve in Merge Editor" is visible in the bottom right of the editor area.

The screenshot shows the Visual Studio Code interface with a Git merge conflict in the file `h.txt`. The Source Control panel on the left shows the repository `vsbase...` with a branch `B` that is ahead of `main` by 4 commits. The editor window shows the content of `h.txt` with lines 1-7: `111`, `222`, `AAA`, `333`, `444`, `BBB`, and an empty line 7. The Git Graph panel on the right shows the commit history, with the current commit being `ac055e6` on branch `B`. The terminal at the bottom displays the following error message:

```
CONFLICT (content): Merge conflict in h.txt
error: could not apply ac055e6... B
hint: Resolve all conflicts manually, mark them as resolved with
hint: "git add/rm <conflicted_files>", then run "git rebase --continue".
hint: You can instead skip this commit: run "git rebase --skip".
hint: To abort and get back to the state before "git rebase", run "git re
--abort".
Could not apply ac055e6... B
```

# Rebase 성공

오픈소스 소프트웨어를 위한 깃과 깃허브 Python language

**SOURCE CONTROL**

**SOURCE CONTROL REPOSITORIES**

vsbase... topic

Message (Ctrl+Enter to commit on 'topic')

**Publish Branch**

**Git Graph**

**Branches:** Show All

☒ Show Remote Branches

Graph	Description	Date	Auth...	Commit
topic	B	18 Nov 202...	ai7dnn	16569246
main	4	18 Nov 202...	ai7dnn	eb3e23d1
	3	18 Nov 202...	ai7dnn	39ec9d3e
	A	18 Nov 202...	ai7dnn	ee1f3861
	2	18 Nov 202...	ai7dnn	cbb82844
	1	18 Nov 202...	ai7dnn	29a59522

**PROBLEMS OUTPUT TERMINAL GITLENS JUPYTER DEBUG CONSOLE**

CONFLICT (content): Merge conflict in h.txt  
error: could not apply ac055e6... B  
hint: Resolve all conflicts manually, mark them as resolved with  
hint: "git add/rm <conflicted\_files>", then run "git rebase --continue".  
hint: You can instead skip this commit: run "git rebase --skip".  
hint: To abort and get back to the state before "git rebase", run "git rebase --abort".  
Could not apply ac055e6... B

Ln 7, Col 1 Spaces: 4 UTF-8 CRLF Plain Text

# 다시 main에서 merge

오픈소스 소프트웨어를 위한 깃과 깃허브 Python language

- 브랜치 main을 더블 클릭
  - Fast-forward merge
    - No commit

VS Code interface showing a Git merge operation. The 'Git Graph' panel is open, displaying a commit history. A context menu is visible over the 'topic' branch, with 'Merge into current branch...' selected. A dialog box is shown in the foreground asking for confirmation to merge branch 'topic' into 'main' (the current branch). The dialog includes options: 'Create a new commit even if fast-forward is possible' (unchecked), 'Squash Commits' (unchecked), and 'No Commit' (checked). The 'Yes, merge' button is highlighted.

Are you sure you want to merge branch **topic** into **main** (the current branch)?

- ☐ Create a new commit even if fast-forward is possible
- ☐ Squash Commits ⓘ
- ☒ No Commit ⓘ

Yes, merge Cancel

# Rebase 성공

오픈소스 소프트웨어를 위한 깃과 깃허브 Python language

The screenshot displays the Visual Studio Code interface during a Git rebase operation. The **SOURCE CONTROL** panel on the left shows the repository 'vsbase...' with a 'main' branch. The **Git Graph** panel on the right shows a commit history with a 'main' branch and a 'topic' branch. The file editor in the center shows a file 'h.txt' with content: 1 111, 2 222, 3 AAA, 4 333, 5 444, 6 BBB, 7. The terminal at the bottom shows a conflict resolution message: 'CONFLICT (content): Merge conflict in h.txt' and instructions to resolve the conflict manually.

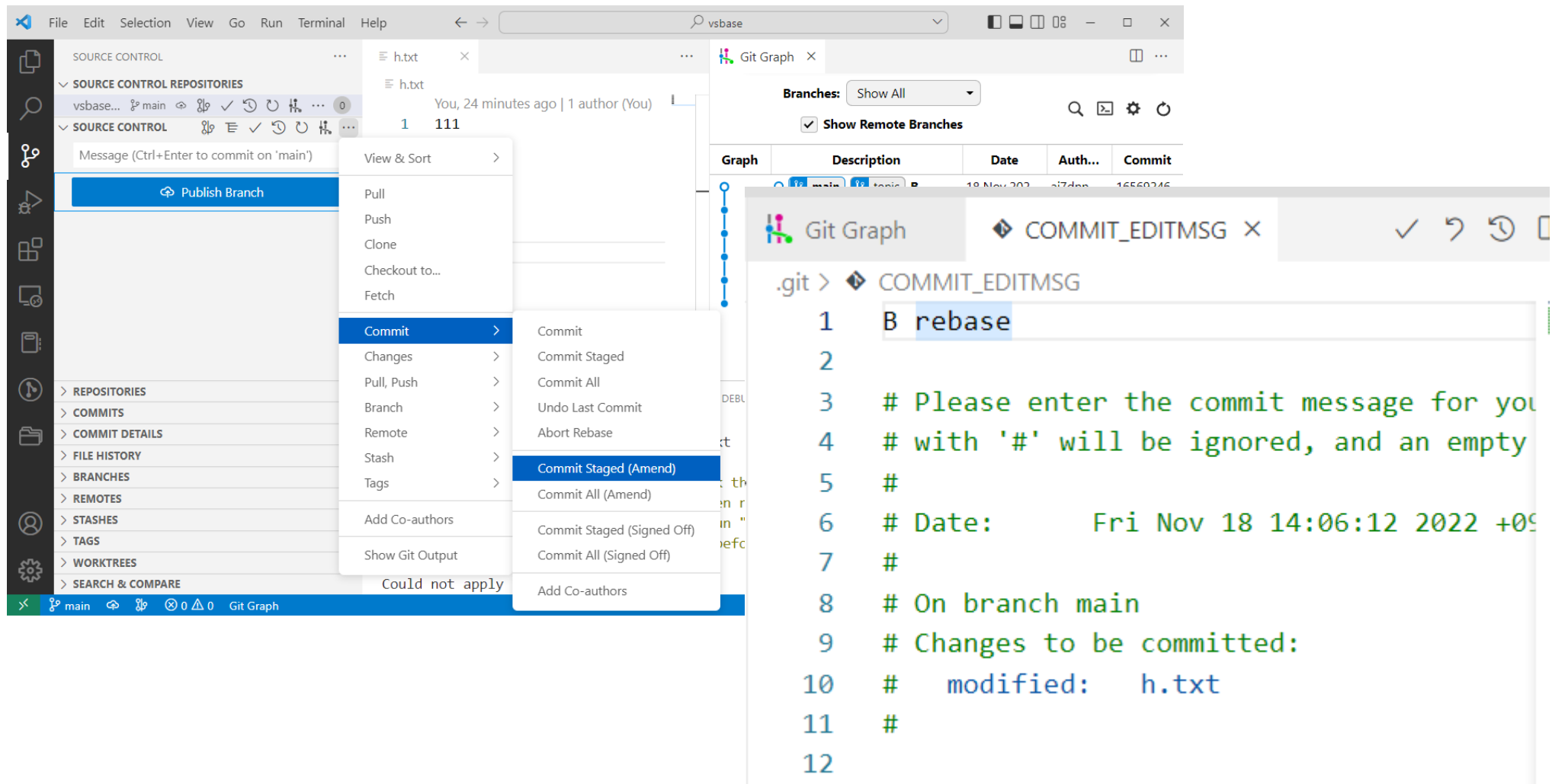
**Git Graph Table:**

Graph	Description	Date	Auth...	Commit
○	main topic B	18 Nov 202...	ai7dnn	16569246
●	4	18 Nov 202...	ai7dnn	eb3e23d1
●	3	18 Nov 202...	ai7dnn	39ec9d3e
●	A	18 Nov 202...	ai7dnn	ee1f3861
●	2	18 Nov 202...	ai7dnn	cbb82844
●	1	18 Nov 202...	ai7dnn	29a59522

**Terminal Output:**

```
CONFLICT (content): Merge conflict in h.txt
error: could not apply ac055e6... B
hint: Resolve all conflicts manually, mark them as resolved with
hint: "git add/rm <conflicted_files>", then run "git rebase --continue".
hint: You can instead skip this commit: run "git rebase --skip".
hint: To abort and get back to the state before "git rebase", run "git re
--abort".
Could not apply ac055e6... B
```

- 소스 제어에서 메뉴 Commit > Commit Staged(Amend) 선택
  - 메시지 수정 후 저장, 닫기



The screenshot shows the VS Code interface with the Git menu open. The path to 'Commit Staged (Amend)' is highlighted. The resulting commit message editor is also shown, with the message 'B rebase' entered.

VS Code Interface:

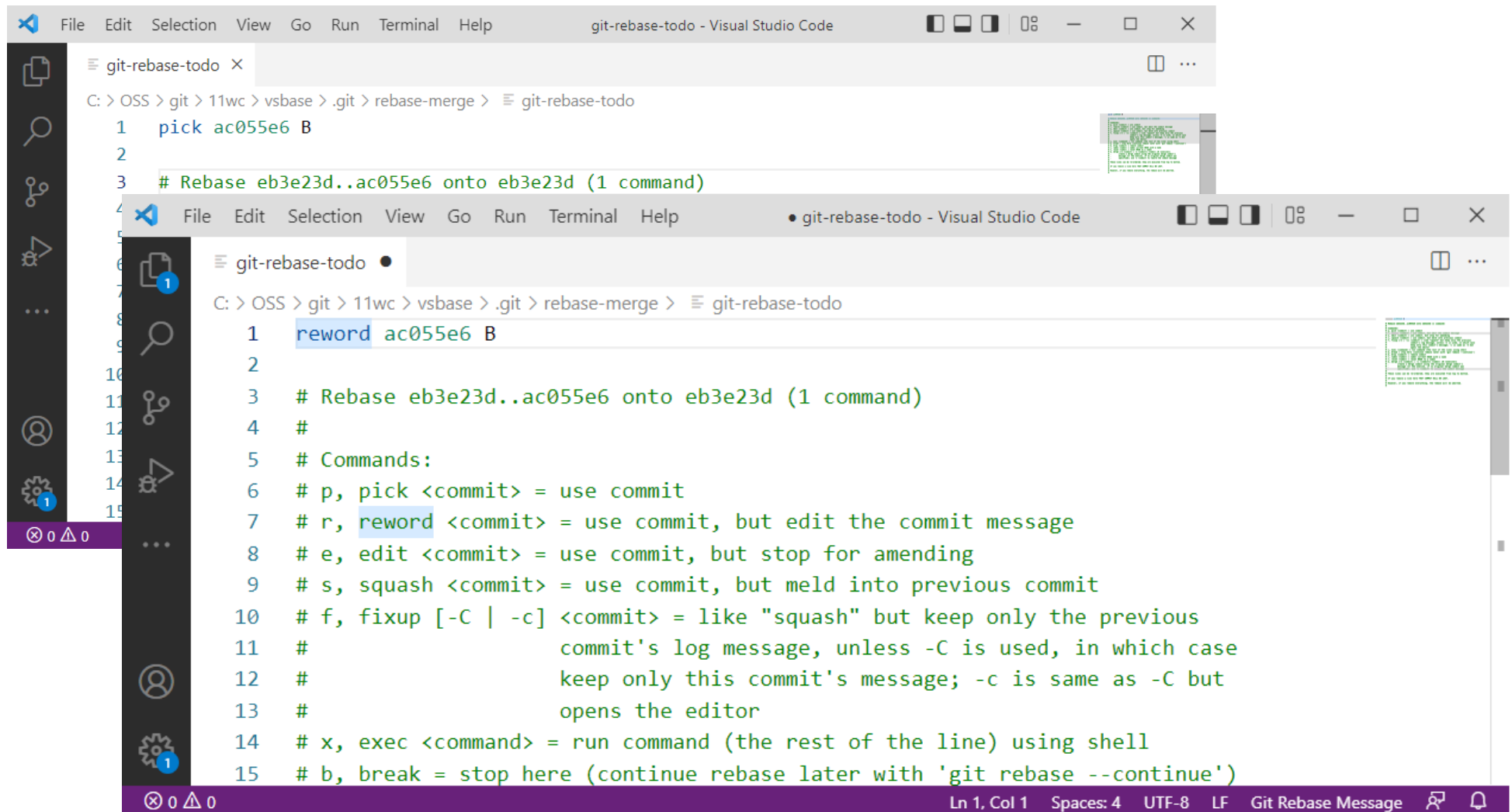
- File Edit Selection View Go Run Terminal Help
- vsbase
- SOURCE CONTROL
- SOURCE CONTROL REPOSITORIES
- vsbase... main
- Message (Ctrl+Enter to commit on 'main')
- Commit Staged (Amend)

Git Graph:

- Branches: Show All
- Show Remote Branches
- Graph
- Description
- Date
- Auth...
- Commit

COMMIT\_EDITMSG

```
.git > COMMIT_EDITMSG
1 B rebase
2
3 # Please enter the commit message for your commit
4 # with '#' will be ignored, and an empty
5 #
6 # Date: Fri Nov 18 14:06:12 2022 +09
7 #
8 # On branch main
9 # Changes to be committed:
10 #   modified:   h.txt
11 #
12
```



```

git-rebase-todo
C: > OSS > git > 11wc > vsbase > .git > rebase-merge > git-rebase-todo
1 pick ac055e6 B
2
3 # Rebase eb3e23d..ac055e6 onto eb3e23d (1 command)
4
5 # Commands:
6 # p, pick <commit> = use commit
7 # r, reword <commit> = use commit, but edit the commit message
8 # e, edit <commit> = use commit, but stop for amending
9 # s, squash <commit> = use commit, but meld into previous commit
10 # f, fixup [-C | -c] <commit> = like "squash" but keep only the previous
11 #      commit's log message, unless -C is used, in which case
12 #      keep only this commit's message; -c is same as -C but
13 #      opens the editor
14 # x, exec <command> = run command (the rest of the line) using shell
15 # b, break = stop here (continue rebase later with 'git rebase --continue')

```