

# Git – 버전관리시스템

Samkeun Kim <skim@hknu.ac.kr>

<http://cyber.hankyong.ac.kr>

**GIT = Version Control System**

**We are  
ALREADY  
managing  
versions**



**report.xls**



**report\_final.xls**



**report\_final\_real\_final.xls**



**report\_final\_real\_final\_the\_real\_final.xls**

# Version Control System?



**Version Management,  
not by changing the file name**

# **Backup**

# **Recovery**

# **Collaboration**



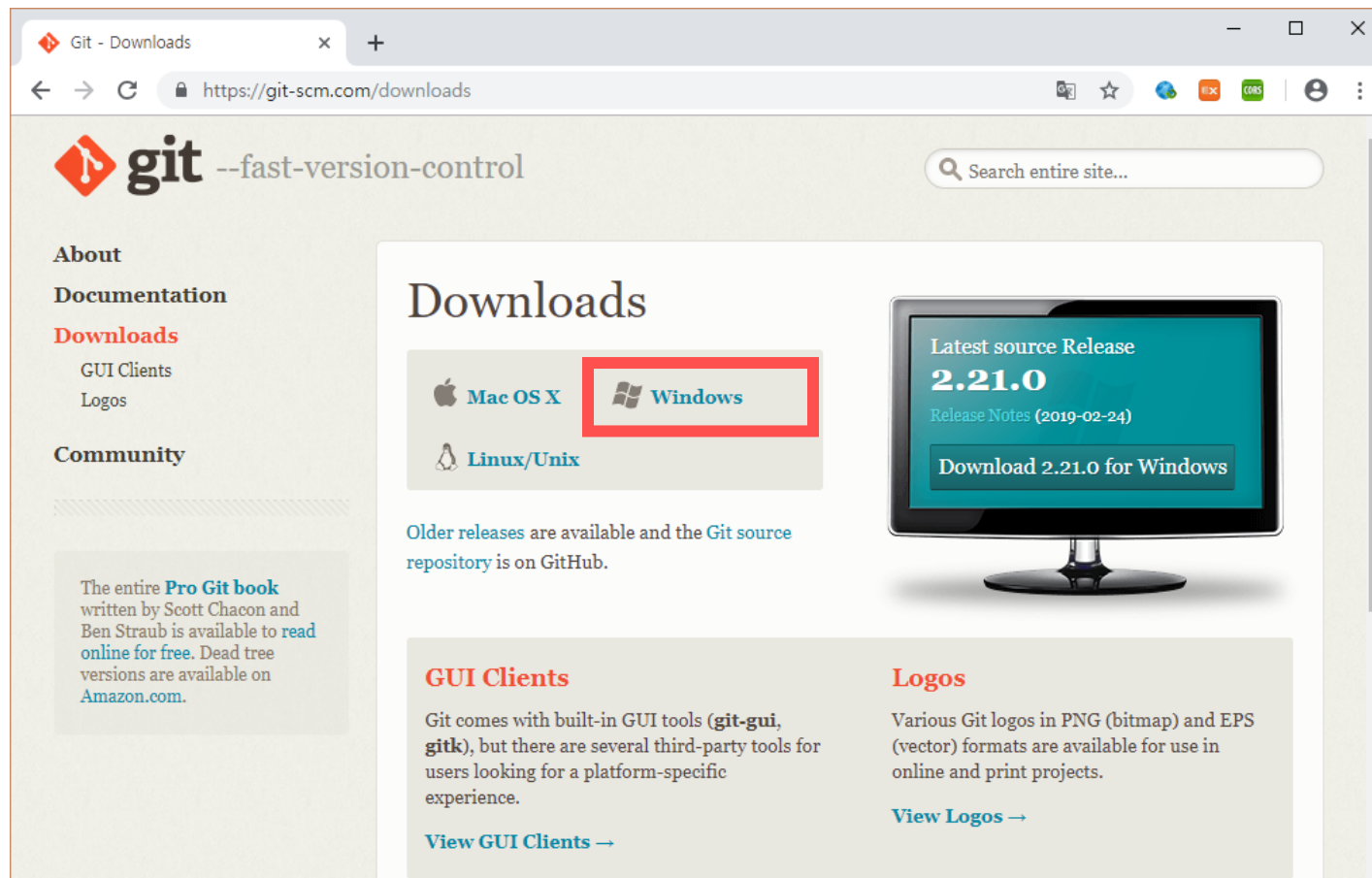
CVS SVN GIT

**Version Control Systems**

# 설치 및 실습방법

Git 공식 홈페이지: <https://git-scm.com/downloads>

자신의 환경에 맞는 버전 설치





🔍 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](#).

## Downloading Git



### Your download is starting...

You are downloading the latest (**2.21.0**) **64-bit** version of **Git for Windows**. This is the most recent [maintained build](#). It was released **10 days ago**, on 2019-02-26.

If your download hasn't started, [click here to download manually](#).

### Other Git for Windows downloads

Git for Windows Setup

[32-bit Git for Windows Setup](#).

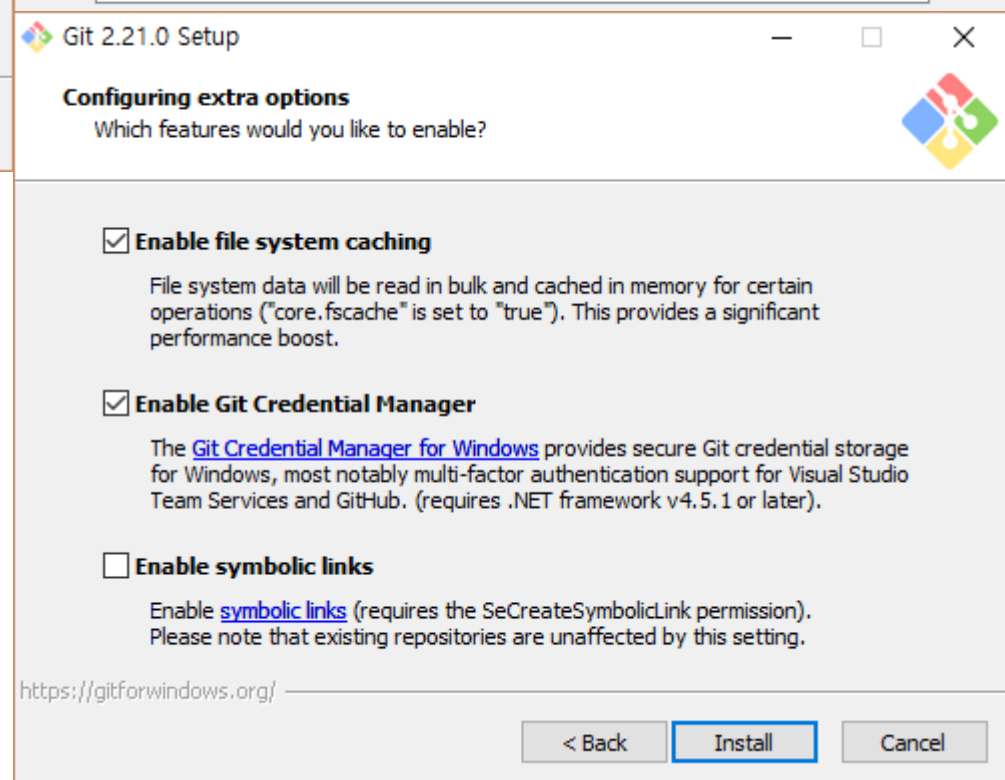
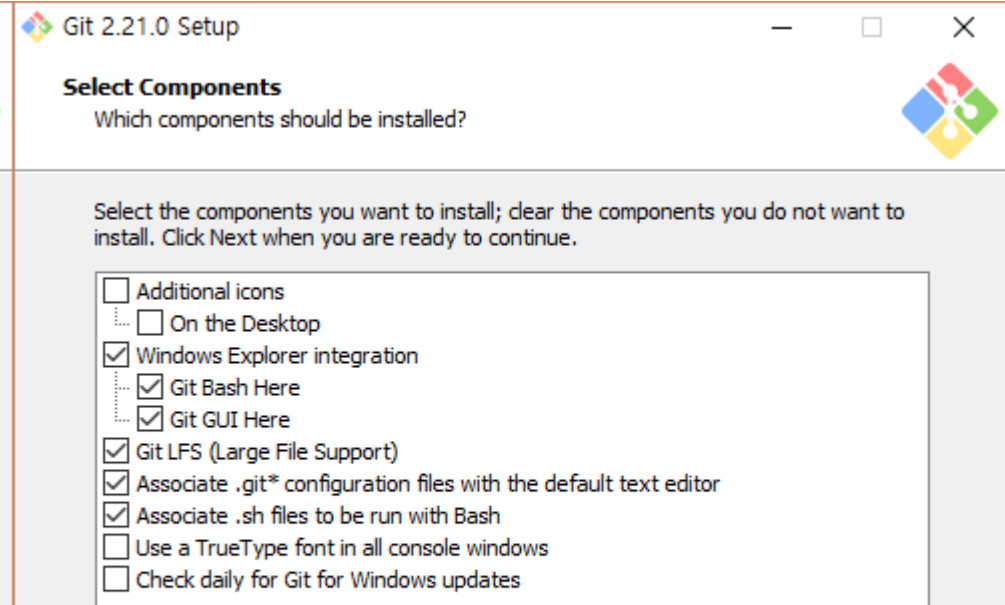
[64-bit Git for Windows Setup](#).

Git for Windows Portable ("thumbdrive edition")

[32-bit Git for Windows Portable](#).

📄 Git-2.21.0-64-bit.exe ^

전체 보기



오른쪽 install 화면이 나올 때까지 계속  
next 클릭!

윈도 검색 > git 입력 > **Git Bash** 클릭!

또는 작업 디렉토리에서 우클릭!



```
MINGW64:/c/Users/sam

sam@DESKTOP-QRCKCRR MINGW64 ~
$ |

MINGW64:/c/Users/sam
$ git
usage: git [--version] [--help] [-C <path>] [-c name=value]
       [--exec-path[=<path>]] [--html-path] [--man-path] [--info-path]
       [-p | --paginate | --no-pager] [--no-replace-objects] [--bare]
       [--git-dir=<path>] [--work-tree=<path>] [--namespace=<name>]
       <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
    reset      Reset current HEAD to the specified state
    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
    grep       Print lines matching a pattern
```

# 저장소 만들기

```
MINGW64:/e/practice/wsc/gitsam
sam@DESKTOP-QRCKCRR MINGW64 ~
$ pwd
/c/Users/sam

sam@DESKTOP-QRCKCRR MINGW64 ~
$ cd /e/practice/wsc/

sam@DESKTOP-QRCKCRR MINGW64 /e/practice/wsc
$ mkdir gitsam

sam@DESKTOP-QRCKCRR MINGW64 /e/practice/wsc
$ cd gitsam/

sam@DESKTOP-QRCKCRR MINGW64 /e/practice/wsc/gitsam
$ git
usage: git [--version] [--help] [-C <path>] [-c name=value]
           [--exec-path[=<path>]] [--html-path] [--man-path] [--info-path]
           [-p | --paginate | --no-pager] [--no-replace-objects] [--bare]
           [--git-dir=<path>] [--work-tree=<path>] [--namespace=<name>]
           <command> [<args>]

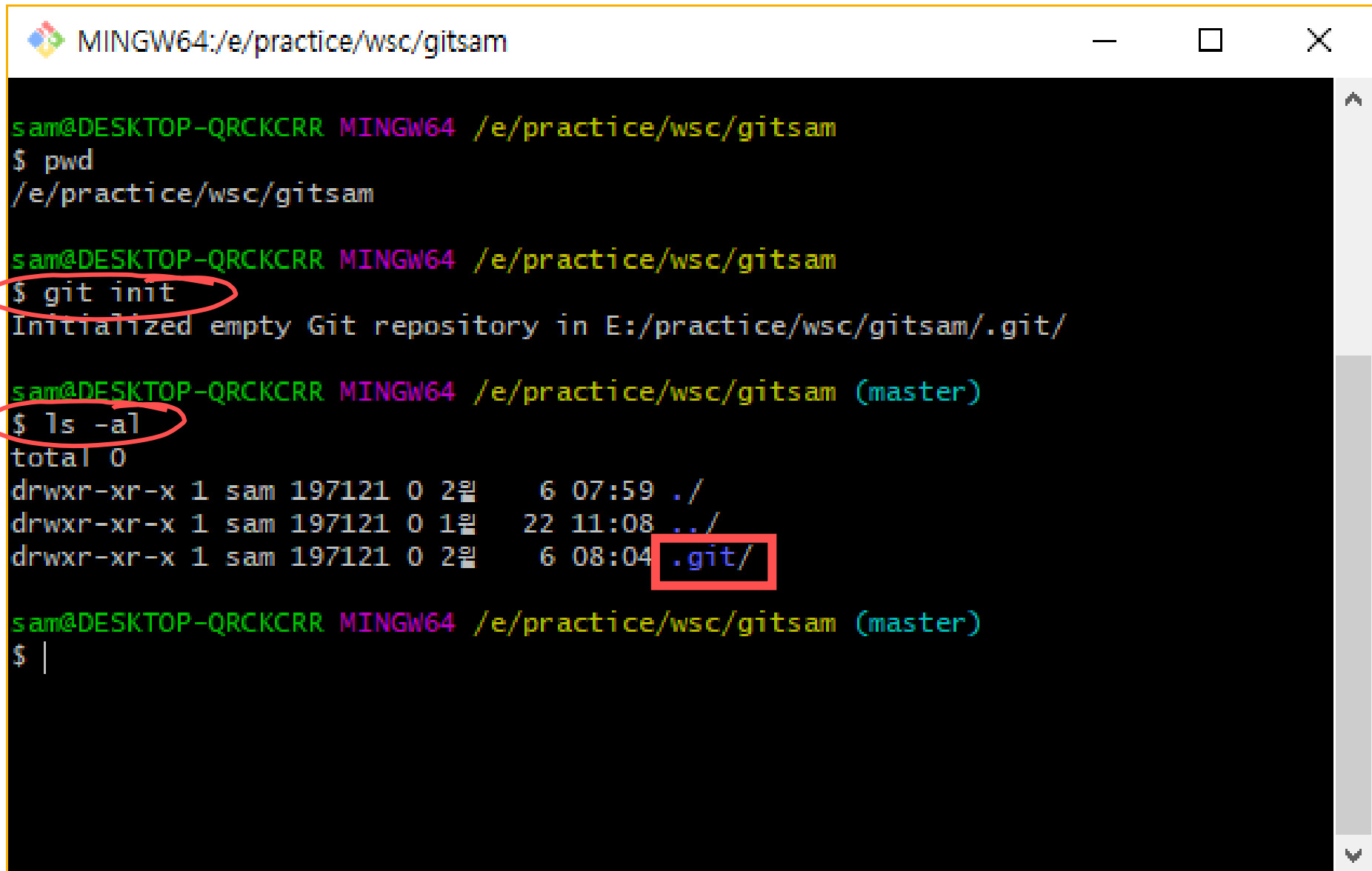
These are common Git commands used in various situations:

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
```



\$ git init



```
MINGW64:/e/practice/wsc/gitsam

sam@DESKTOP-QRCKCRR MINGW64 /e/practice/wsc/gitsam
$ pwd
/e/practice/wsc/gitsam

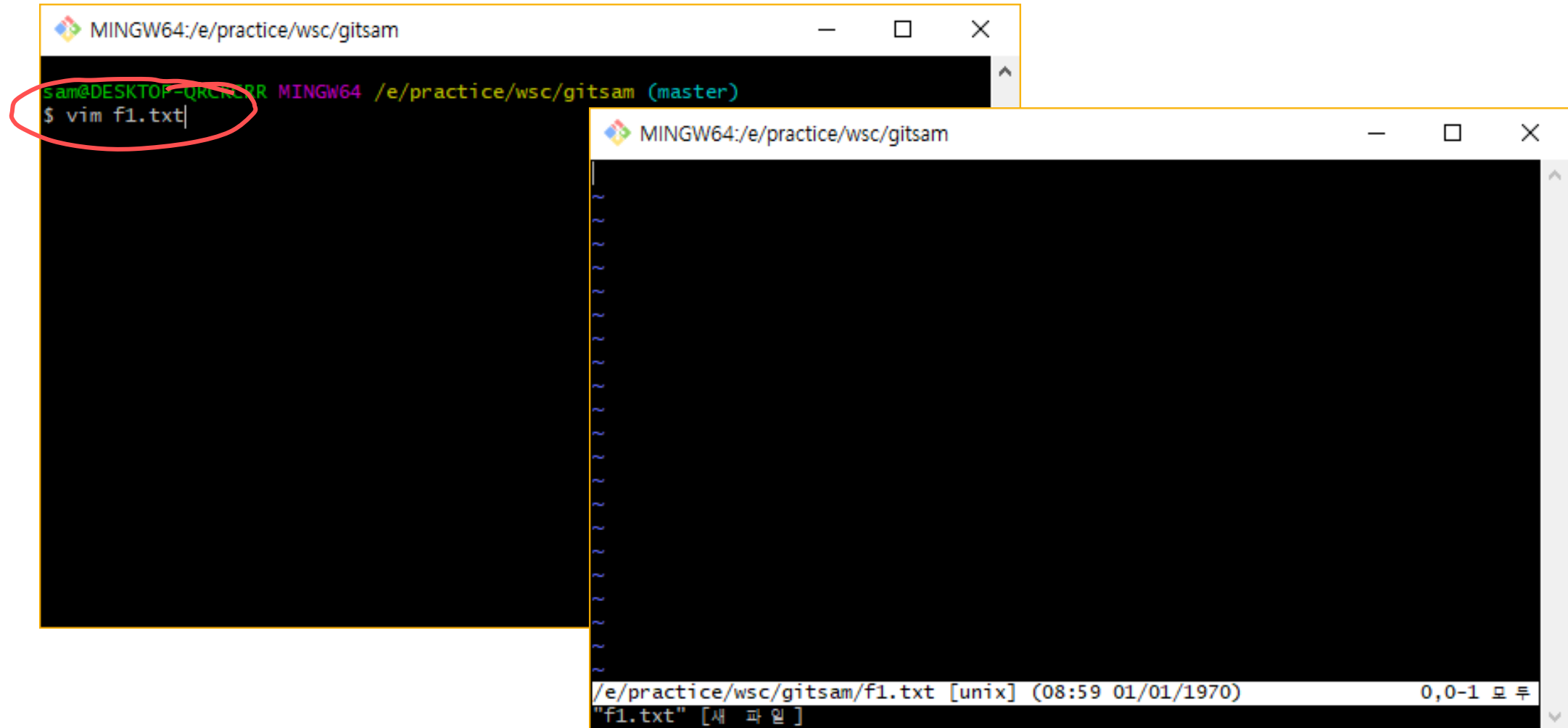
sam@DESKTOP-QRCKCRR MINGW64 /e/practice/wsc/gitsam
$ git init
Initialized empty Git repository in E:/practice/wsc/gitsam/.git/

sam@DESKTOP-QRCKCRR MINGW64 /e/practice/wsc/gitsam (master)
$ ls -al
total 0
drwxr-xr-x 1 sam 197121 0 2월  6 07:59 ./
drwxr-xr-x 1 sam 197121 0 1월  22 11:08 ../
drwxr-xr-x 1 sam 197121 0 2월  6 08:04 .git/

sam@DESKTOP-QRCKCRR MINGW64 /e/practice/wsc/gitsam (master)
$ |
```

# Git이 관리할 대상으로 파일 등록

\$ vim f1.txt



The image shows two overlapping terminal windows. The top window has a title bar 'MINGW64:/e/practice/wsc/gitsam' and a command prompt 'sam@DESKTOP-QKCN2RR MINGW64 /e/practice/wsc/gitsam (master)'. The command '\$ vim f1.txt' is entered and circled in red. The bottom window also has a title bar 'MINGW64:/e/practice/wsc/gitsam' and shows the vim editor interface. The status bar at the bottom of the vim window displays '/e/practice/wsc/gitsam/f1.txt [unix] (08:59 01/01/1970)' and '0,0-1 모 투', with a message '"f1.txt" [새 파일]' below it.



\$ git status

```
MINGW64:/e/practice/wsc/gitsam
drwxr-xr-x 1 sam 197121 0 2월  6 07:59 ./
drwxr-xr-x 1 sam 197121 0 1월 22 11:08 ../
drwxr-xr-x 1 sam 197121 0 2월  6 08:04 .git/
-rw-r--r-- 1 sam 197121 11 2월  6 08:13 f1.txt

sam@DESKTOP-QRCKCRR MINGW64 /e/practice/wsc/gitsam (master)
$ cat f1.txt
source : 1

sam@DESKTOP-QRCKCRR MINGW64 /e/practice/wsc/gitsam (master)
$ git status
On branch master

No commits yet

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

f1.txt

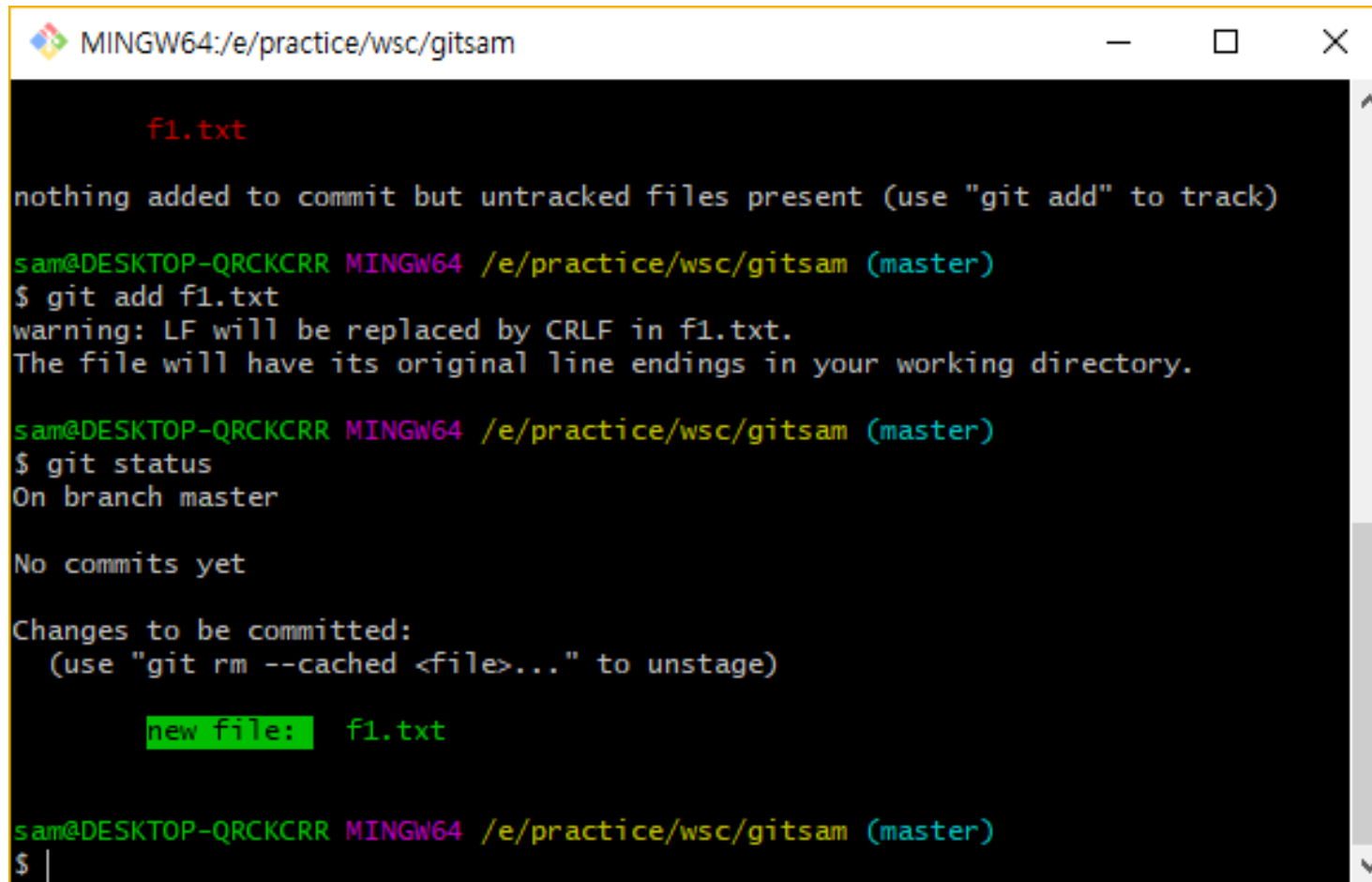
nothing added to commit but untracked files present (use "git add" to track)

sam@DESKTOP-QRCKCRR MINGW64 /e/practice/wsc/gitsam (master)
$ |
```

**Untracked files** - git에게 “버전관리를 시작해”라고 하기 전까지는 git이 f1.txt 파일을 무시한다!

Git에게 "이거 관리해!"라고 얘기해 줘야 한다.

\$ **git add f1.txt**



```
MINGW64:/e/practice/wsc/gitsam

f1.txt

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

sam@DESKTOP-QRCKCRR MINGW64 /e/practice/wsc/gitsam (master)
$ git add f1.txt
warning: LF will be replaced by CRLF in f1.txt.
The file will have its original line endings in your working directory.

sam@DESKTOP-QRCKCRR MINGW64 /e/practice/wsc/gitsam (master)
$ git status
On branch master

No commits yet

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

    new file:   f1.txt

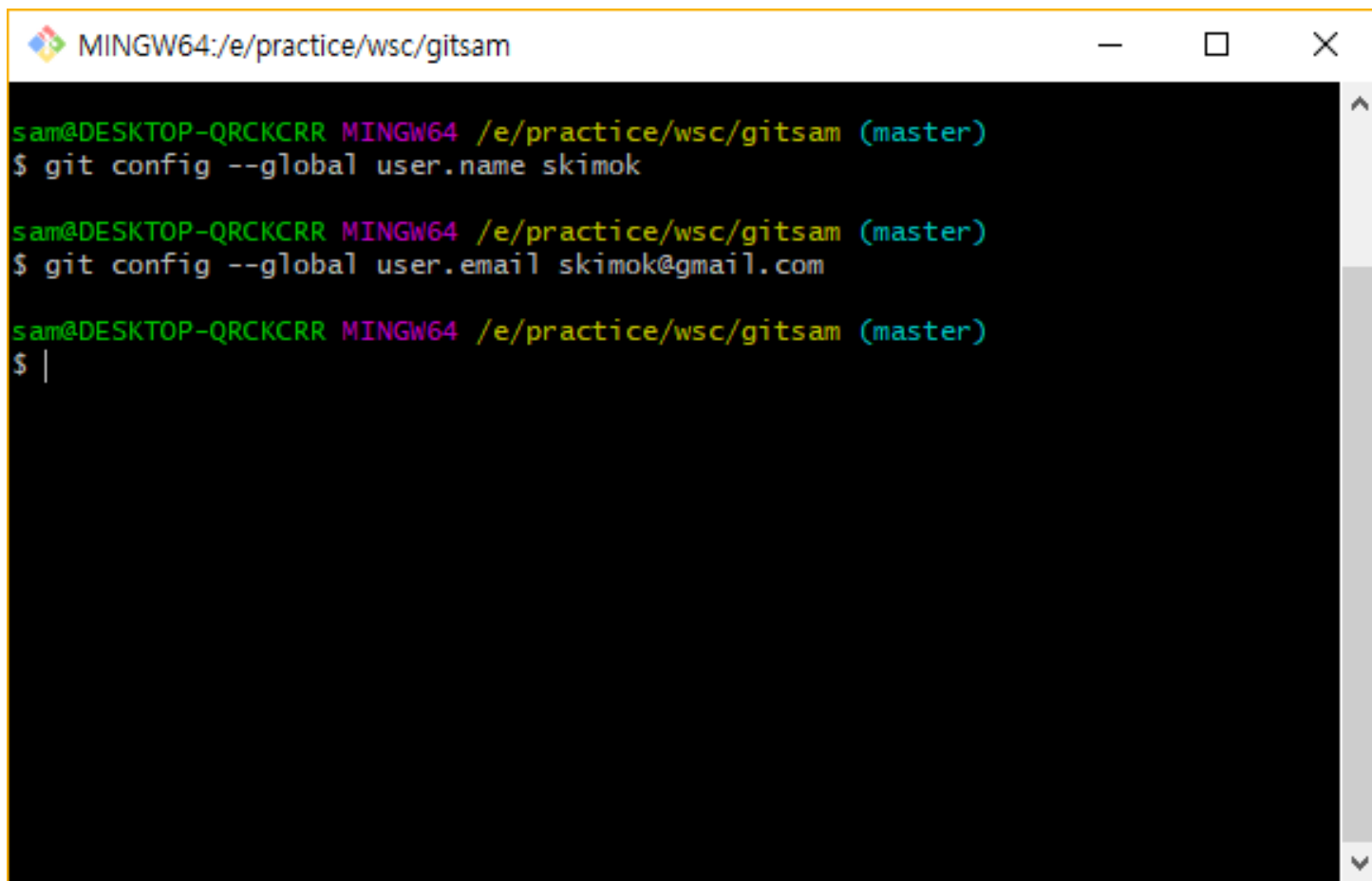
sam@DESKTOP-QRCKCRR MINGW64 /e/practice/wsc/gitsam (master)
$ |
```

관리해야 하는 파일이 무엇인가를 명확하게 Git에게 알려줘야 한다!

=> 왜냐하면 버전관리에서 배제시켜야 하는 파일이 있을 수 있기 때문에 이런 방식을 취한다!

# 버전 만들기(commit)

지금부터 작성할 버전들이 자신이 만든 것이라라는 사실을 다른 사람에게 알려주기 위해서 이름을 세팅해줘야 한다:


A screenshot of a Windows terminal window titled 'MINGW64:/e/practice/wsc/gitsam'. The window has standard Windows window controls (minimize, maximize, close) in the top right. The terminal shows three lines of command execution. The first line shows the prompt 'sam@DESKTOP-QRCKCRR MINGW64 /e/practice/wsc/gitsam (master)' followed by '\$ git config --global user.name skimok'. The second line shows the same prompt followed by '\$ git config --global user.email skimok@gmail.com'. The third line shows the same prompt followed by '\$ |'.

```
MINGW64:/e/practice/wsc/gitsam
sam@DESKTOP-QRCKCRR MINGW64 /e/practice/wsc/gitsam (master)
$ git config --global user.name skimok

sam@DESKTOP-QRCKCRR MINGW64 /e/practice/wsc/gitsam (master)
$ git config --global user.email skimok@gmail.com

sam@DESKTOP-QRCKCRR MINGW64 /e/practice/wsc/gitsam (master)
$ |
```

```
$ git commit
```



```
MINGW64:/e/practice/wsc/gitsam
sam@DESKTOP-QRCKCRR MINGW64 /e/practice/wsc/gitsam (master)
$ git status
On branch master

No commits yet

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

        new file:   f1.txt

sam@DESKTOP-QRCKCRR MINGW64 /e/practice/wsc/gitsam (master)
$ git commit
Please enter the commit message for
# (and the message will be ignored, and an empty
# commit will be created)
#
# On branch master
#
# Initial commit
#
Changes to be committed:
  new file:   f1.txt
```

vim이 실행된다!

vim이 실행된다!

A screenshot of a Windows command prompt window titled "MINGW64:/e/practice/wsc/gitsam". The terminal displays standard Git commit instructions in green monospace font:  

```
# Please enter the commit message for your changes. Lines starting  
# with '#' will be ignored, and an empty message aborts the commit.  
#  
# On branch master  
#  
g # Initial commit  
#  
Changes to be committed:  
   new file:   f1.txt
```

  
Below the main instruction area are several tilde (~) symbols. At the bottom, there is a status bar with two entries: "`/e/practice/wsc/gitsam/.git/COMMIT_EDITMSG [unix] (08:46 06/02/2018) 1,0-1 모 투`" and "`E:/practice/wsc/gitsam/.git/COMMIT_EDITMSG" [유틸리티] 11L, 228C`". A small white sticky note with an exclamation mark (!) is partially visible on the left edge of the terminal window.

## 현재 버전의 메시지를 적는다:

메시지에 이 변화가 어떤 변화를 담고 있는지, 이 파일이 왜 변경되었는지 그 이유를 적는다!

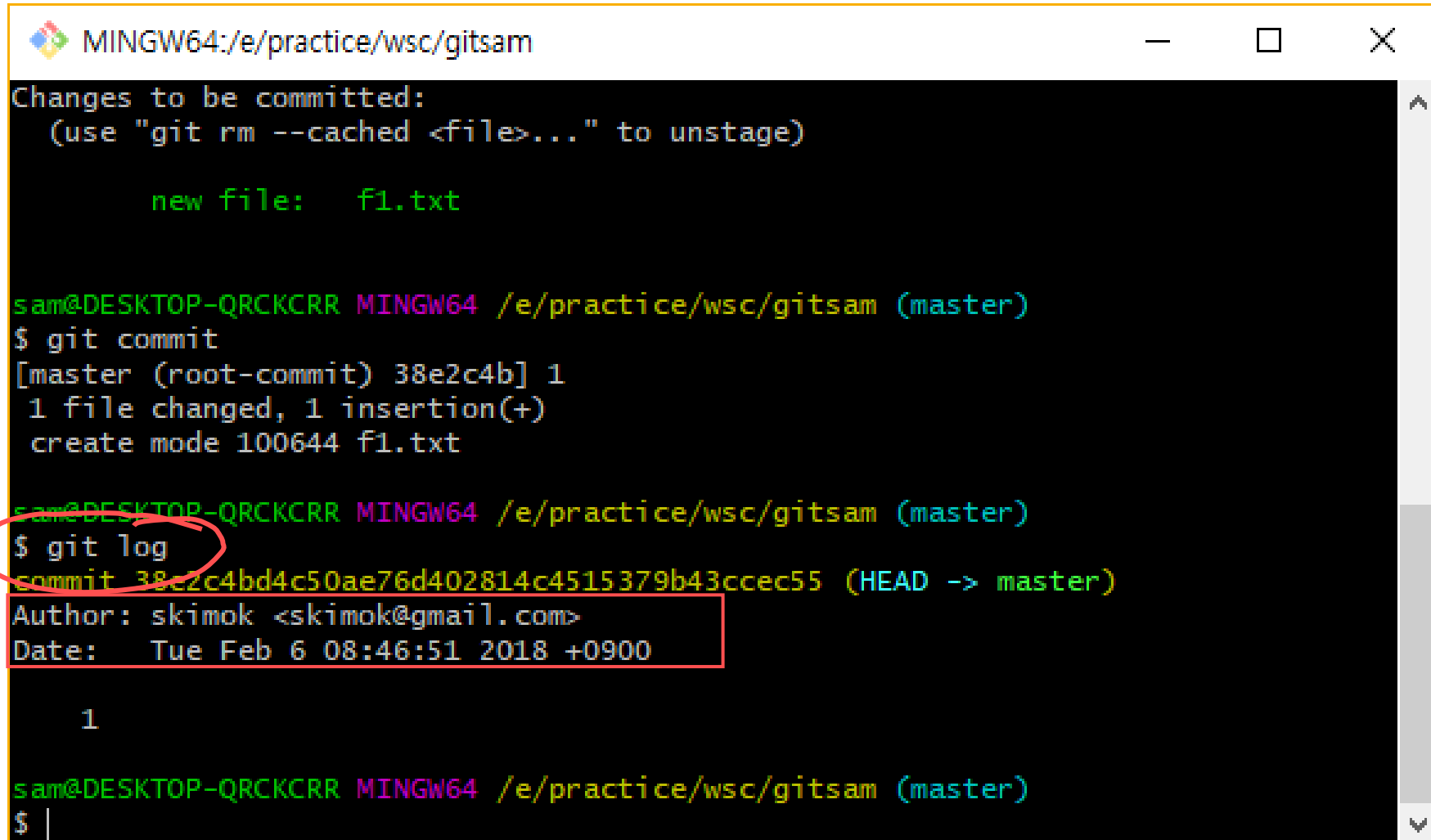
The image contains two screenshots of a Windows terminal window titled "MINGW64:/e/practice/wsc/gitsam". The top screenshot shows the user entering a commit message in the editor. The bottom screenshot shows the execution of several Git commands: \$ git status, \$ git commit, and \$ |. The output of \$ git status shows "On branch master" and "No commits yet". The output of \$ git commit shows "[master (root-commit) 38e2c4b] 1 file changed, 1 insertion(+), create mode 100644 f1.txt". A red circle highlights the output of the \$ git commit command. The terminal also shows the prompt "\$ |" at the end.

```
MINGW64:/e/practice/wsc/gitsam  
# Please enter the commit message for your changes. Lines starting  
# with '#' will be ignored, and an empty message aborts the commit.  
#  
# On branch master  
#  
# Initial commit  
# Changes to be committed:  
#   new file:   f1.txt  
~  
~  
~  
~  
~  
~  
~  
~  
~  
~  
/e/practice/wsc/gitsam/.git/COMMIT_EDITMSG[+] [unix]  
:wq
```

```
MINGW64:/e/practice/wsc/gitsam  
sam@DESKTOP-QRCKCRR MINGW64 /e/practice/wsc/gitsam (master)  
$ git status  
On branch master  
  
No commits yet  
  
Changes to be committed:  
  (use "git rm --cached <file>..." to unstage)  
  
       new file:   f1.txt  
  
sam@DESKTOP-QRCKCRR MINGW64 /e/practice/wsc/gitsam (master)  
$ git commit  
[master (root-commit) 38e2c4b] 1  
1 file changed, 1 insertion(+)  
create mode 100644 f1.txt  
  
sam@DESKTOP-QRCKCRR MINGW64 /e/practice/wsc/gitsam (master)  
$ |
```



\$ git log



The screenshot shows a Windows command prompt window titled "MINGW64:/e/practice/wsc/gitsam". The window contains the following text:

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

       new file:   f1.txt

sam@DESKTOP-QRCKCRR MINGW64 /e/practice/wsc/gitsam (master)
$ git commit
[master (root-commit) 38e2c4b] 1
 1 file changed, 1 insertion(+)
 create mode 100644 f1.txt

sam@DESKTOP-QRCKCRR MINGW64 /e/practice/wsc/gitsam (master)
$ git log
commit 38e2c4bd4c50ae76d402814c4515379b43ccec55 (HEAD -> master)
Author: skimok <skimok@gmail.com>
Date:   Tue Feb 6 08:46:51 2018 +0900

    1

sam@DESKTOP-QRCKCRR MINGW64 /e/practice/wsc/gitsam (master)
$ |
```

The output of the `git log` command is highlighted with a red box. The commit message "1" is also highlighted with a red box.

```
MINGW64:/e/practice/wsc/gitsam
[master (root-commit) 38e2c4b] 1
1 file changed, 1 insertion(+)
create mode 100644 f1.txt
```

```
sam@DESKTOP-QRCKCRR MINGW64
$ git log
commit 38e2c4bd4c50ae76d40
Author: skimok <skimok@gmail.com>
Date: Tue Feb 6 08:46:51 2024

    1

sam@DESKTOP-QRCKCRR MINGW64
$ ls -al
total 16
drwxr-xr-x 1 sam 197121 0
drwxr-xr-x 1 sam 197121 0
drwxr-xr-x 1 sam 197121 0
-rw-r--r-- 1 sam 197121 11

sam@DESKTOP-QRCKCRR MINGW64
$ vim f1.txt |
```

\$ vim f1.txt

```
MINGW64:/e/practice/wsc/gitsam
source : 2
```

```
MINGW64:/e/practice/wsc/gitsam
total 16
drwxr-xr-x 1 sam 197121 0 2월 6 07:59 ./
drwxr-xr-x 1 sam 197121 0 1월 22 11:08 ../
drwxr-xr-x 1 sam 197121 0 2월 6 08:04 .git/
-rw-r--r-- 1 sam 197121 11 2월 6 08:13 f1.txt

sam@DESKTOP-QRCKCRR MINGW64 /e/practice/wsc/gitsam (master)
$ vim f1.txt

sam@DESKTOP-QRCKCRR MINGW64 /e/practice/wsc/gitsam (master)
$ git status
On branch master
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git checkout -- <file>..." to discard changes in working directory)

    modified:   f1.txt

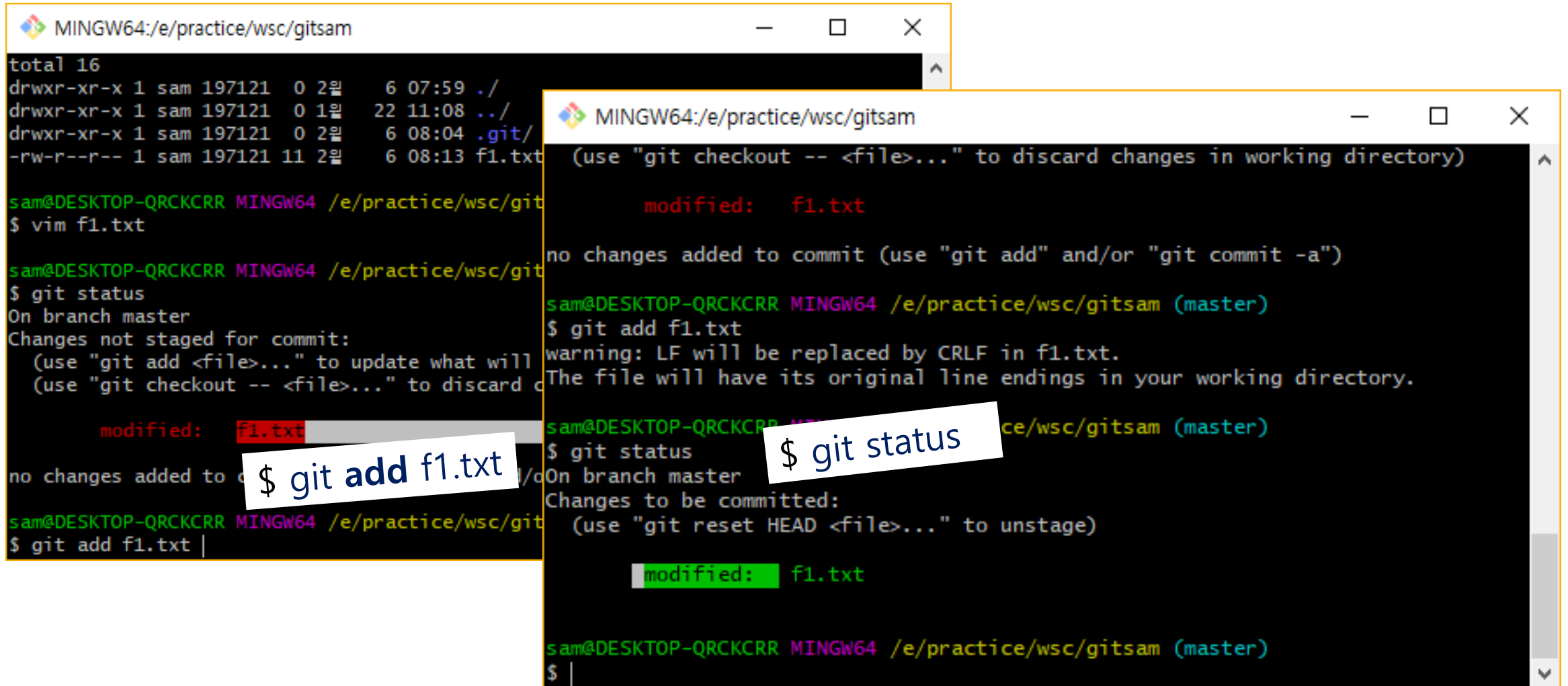
no changes added to commit (use "git add" and/or "git commit -a")

sam@DESKTOP-QRCKCRR MINGW64 /e/practice/wsc/gitsam (master)
$ |
```

\$ git status

/e/practice/wsc/gitsa  
:wq|

최초로 추적할 때도 '**git add**'를 실행해야 하지만, 파일이 수정되어서 버전을 만들기 전에도 '**add**'를 먼저 수행해야 한다:



```
MINGW64:/e/practice/wsc/gitsam
total 16
drwxr-xr-x 1 sam 197121 0 2월 6 07:59 ./
drwxr-xr-x 1 sam 197121 0 1월 22 11:08 ../
drwxr-xr-x 1 sam 197121 0 2월 6 08:04 .git/
-rw-r--r-- 1 sam 197121 11 2월 6 08:13 f1.txt

sam@DESKTOP-QRCKCRR MINGW64 /e/practice/wsc/gitsam
$ vim f1.txt

sam@DESKTOP-QRCKCRR MINGW64 /e/practice/wsc/gitsam
$ git status
On branch master
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git checkout -- <file>..." to discard changes in working directory)

        modified:   f1.txt

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

sam@DESKTOP-QRCKCRR MINGW64 /e/practice/wsc/gitsam (master)
$ git add f1.txt
warning: LF will be replaced by CRLF in f1.txt.
The file will have its original line endings in your working directory.

sam@DESKTOP-QRCKCRR MINGW64 /e/practice/wsc/gitsam (master)
$ git status
On branch master
Changes to be committed:
  (use "git reset HEAD <file>..." to unstage)

        modified:   f1.txt

sam@DESKTOP-QRCKCRR MINGW64 /e/practice/wsc/gitsam (master)
$
```

```
MINGW64:/e/practice/wsc/gitsam
no changes added to commit (use "git add" and/or "git commit -a")

sam@DESKTOP-QRCKCRR MINGW64 /e/practice/wsc/gitsam (master)
$ git add f1.txt
warning: LF will be replaced by CRLF in f1.txt.
The file will have its original line endings.

sam@DESKTOP-QRCKCRR MINGW64 /e/practice/wsc/gitsam (master)
$ git commit
$ git status
On branch master
Changes to be committed:
  (use "git reset HEAD <file>" to unstage tracked files)
        modified:   f1.txt

sam@DESKTOP-QRCKCRR MINGW64 /e/practice/wsc/gitsam (master)
$ git commit
Aborting commit due to empty commit.
```

```
MINGW64:/e/practice/wsc/gitsam
sam@DESKTOP-QRCKCRR MINGW64 /e/practice/wsc/gitsam (master)
$ git commit
[master 3cf9373] 2
1 file changed, 1 insertion(+), 1 deletion(-)

sam@DESKTOP-QRCKCRR MINGW64 /e/practice/wsc/gitsam (master)
$ git log
commit 3cf9373684b3a3d835126d563eaae4ea93cdaa29 (HEAD -> master)
Author: skimok <skimok@gmail.com>
Date:   Tue Feb 6 09:15:48 2018 +0900

    2

commit 38e2c4bd4c50ae76d402814c4515379b43ccec55
Author: skimok <skimok@gmail.com>
Date:   Tue Feb 6 08:46:51 2018 +0900

    1

sam@DESKTOP-QRCKCRR MINGW64 /e/practice/wsc/gitsam (master)
$ |
```

# 실습

f2.txt 파일을 commit 해보시오.

```
MINGW64:/e/practice/wsc/gitsam

sam@DESKTOP-QRCKCRR MINGW64 /e/practice/wsc/gitsam (master)
$ cp f1.txt f2.txt

sam@DESKTOP-QRCKCRR MINGW64 /e/practice/wsc/gitsam (master)
$ ls -al
total 32
drwxr-xr-x 1 sam 197121 0 2월 6 07:59
drwxr-xr-x 1 sam 197121 0 1월 22 11:08
drwxr-xr-x 1 sam 197121 0 2월 6 08:04
-rw-r--r-- 1 sam 197121 11 2월 6 09:07
-rw-r--r-- 1 sam 197121 11 2월 6 09:24

sam@DESKTOP-QRCKCRR MINGW64 /e/practice/wsc/gitsam (master)
$ |

MINGW64:/e/practice/wsc/gitsam
(use "git add <file>..." to include in what will be committed)

f2.txt

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

sam@DESKTOP-QRCKCRR MINGW64 /e/practice/wsc/gitsam (master)
$ git add f2.txt
warning: LF will be replaced by CRLF in f2.txt.
The file will have its original line endings in your working directory.

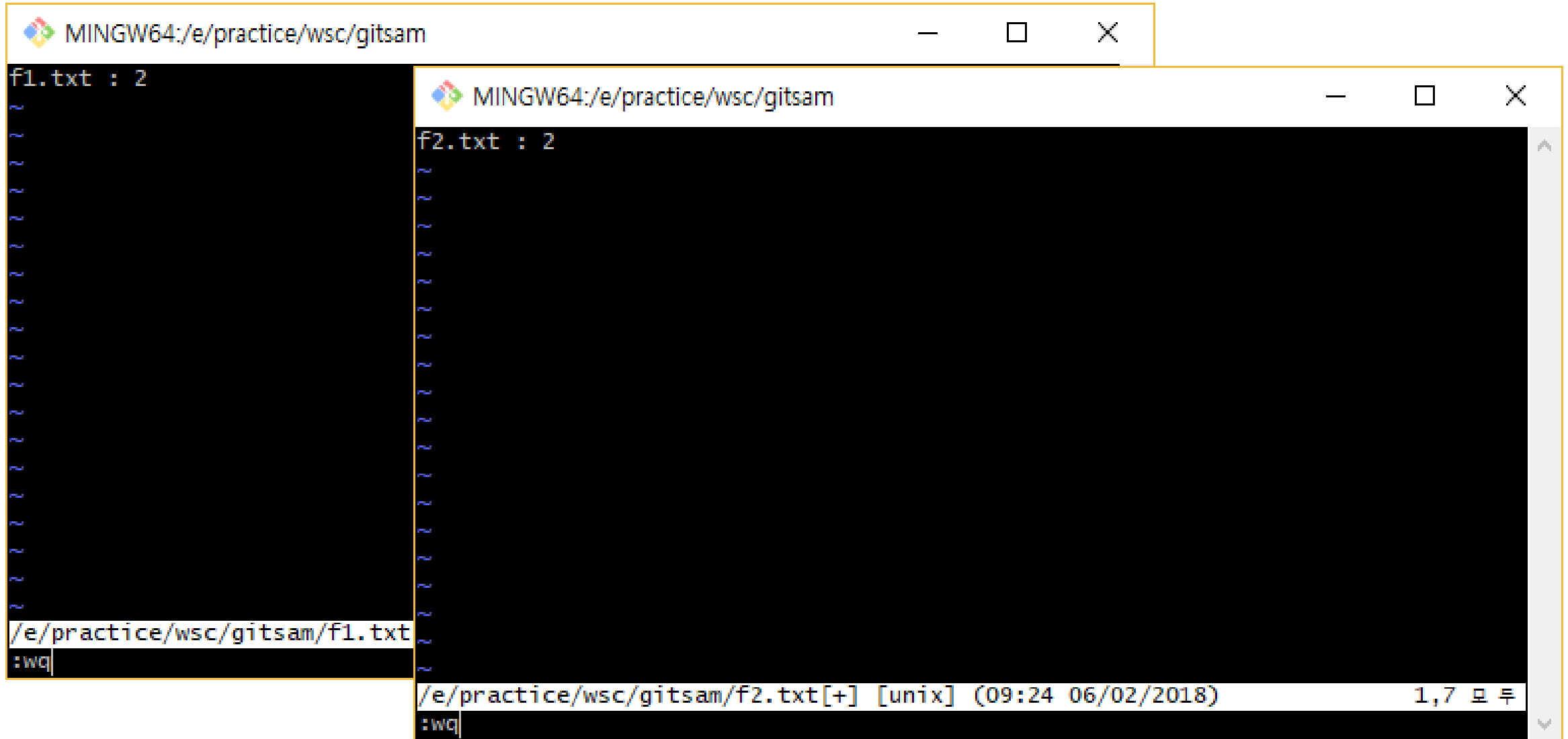
sam@DESKTOP-QRCKCRR MINGW64 /e/practice/wsc/gitsam (master)
$ git status
On branch master
Changes to be committed:
  (use "git reset HEAD <file>..." to unstage)

    new file:   f2.txt

sam@DESKTOP-QRCKCRR MINGW64 /e/practice/wsc/gitsam (master)
$ git commit |
```



f1.txt와 f2.txt 모두 수정:



The image shows two overlapping terminal windows from the MINGW64 environment, both located at the path `/e/practice/wsc/gitsam`. The left window is editing `f1.txt` and shows a vertical list of tilde (~) characters on the left margin, indicating line numbers. The bottom status bar of this window shows the file path `/e/practice/wsc/gitsam/f1.txt` and the command `:wq` at the prompt. The right window is editing `f2.txt` and also shows a vertical list of tilde (~) characters on the left margin. The bottom status bar of this window shows the file path `/e/practice/wsc/gitsam/f2.txt`, the command `:wq` at the prompt, and additional information: `[+] [unix] (09:24 06/02/2018) 1,7 모 투`.

\$ **git add f1.txt f2.txt** ?? 왜 Git이 **add**라는 과정을 포함하고 있는가?

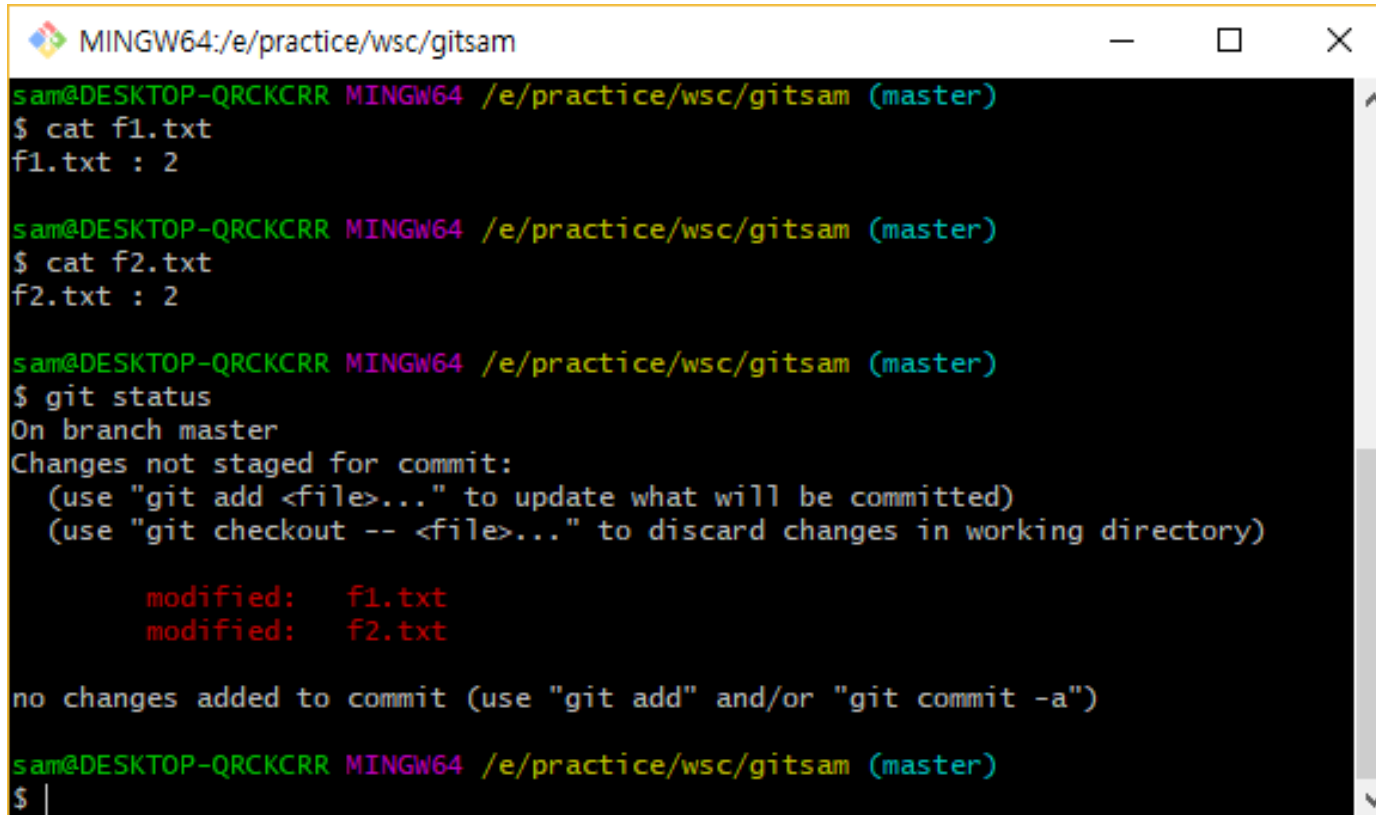
그냥 **Commit**하면 되지, 왜 **Commit**하기 전에 구태여 **Add**를 해야 되는가?

커밋할 시기를 놓쳤을 때 ... 거의 매일 그런 일이 일어난다.

커밋 1개는 하나의 작업을 담고 있는 것이 가장 이상적이다.

그러나, 많은 작업들을 포함하는 거대한 버전을 만들어야 경우가 종종 생긴다.

⇒ **Add**라는 과정을 통해서 커밋하고 싶은 파일만 커밋할 수 있다



```
MINGW64:/e/practice/wsc/gitsam
sam@DESKTOP-QRCKCRR MINGW64 /e/practice/wsc/gitsam (master)
$ cat f1.txt
f1.txt : 2

sam@DESKTOP-QRCKCRR MINGW64 /e/practice/wsc/gitsam (master)
$ cat f2.txt
f2.txt : 2

sam@DESKTOP-QRCKCRR MINGW64 /e/practice/wsc/gitsam (master)
$ git status
On branch master
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git checkout -- <file>..." to discard changes in working directory)

        modified:   f1.txt
        modified:   f2.txt

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

sam@DESKTOP-QRCKCRR MINGW64 /e/practice/wsc/gitsam (master)
$ |
```





```
MINGW64:/e/practice/wsc/gitsam
sam@DESKTOP-QRCKCRR MINGW64 /e/practice/wsc/gitsam (master)
$ git status
On branch master
Changes to be committed:
  (use "git reset HEAD <file>..." to unstage)

        modified:   f1.txt

Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git checkout -- <file>..." to discard changes in working directory)

        modified:   f2.txt

sam@DESKTOP-QRCKCRR MINGW64 /e/practice/wsc/gitsam (master)
$ git commit
[master 86ebfbe] 4
1 file changed, 1 insertion(+), 1 deletion(-)

sam@DESKTOP-QRCKCRR MINGW64 /e/practice/wsc/gitsam (master)
$ git log
```

```
MINGW64:/e/practice/wsc/gitsam
commit 86ebfbaef45fab4c671f3f277fc777f537bb1d1 (HEAD -> master)
Author: skimok <skimok@gmail.com>
Date:   Tue Feb 6 11:40:18 2018 +0900

    4

commit eb66866869c4403096c3f11f505463c490bdc9c5
Author: skimok <skimok@gmail.com>
Date:   Tue Feb 6 10:10:22 2018 +0900

    3

commit 3cf9373684b3a3d835126d563eaae4ea93cdaa29
Author: skimok <skimok@gmail.com>
Date:   Tue Feb 6 09:15:48 2018 +0900

    2

commit 38e2c4bd4c50ae76d402814c4515379b43ccec55
Author: skimok <skimok@gmail.com>
Date:   Tue Feb 6 08:46:51 2018 +0900

    1
```

f1.txt만 커밋 됨!

```
MINGW64:/e/practice/wsc/gitsam

commit 38e2c4bd4c50ae76d402814c4515379b43ccec55
Author: skimok <skimok@gmail.com>
Date: Tue Feb 6 08:46:51 2018 +0900

sam@DESKTOP-QRCKCRR MINGW64 /e/practice/wsc/gitsam (master)
$

sam@DESKTOP-QRCKCRR MINGW64 /e/practice/wsc/gitsam (master)
$ git status
On branch master
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git checkout -- <file>..." to discard changes in working directory)

    modified:   f2.txt

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

sam@DESKTOP-QRCKCRR MINGW64 /e/practice/wsc/gitsam (master)
$ |
```

예전 버전 관리 시스템에서는 항상 한꺼번에 커밋을 수행해야 했었다.

그러나, **Git**에서는 선택적으로 커밋할 수 있게 해준다.

\$ **git add f1.txt**                      => commit 대기 상태

\$ **git commit**

- ✓ 커밋 대기상태에 있는 파일들만을 현재 버전에 포함시킨다

Git에서 커밋 대기상태를 뭐라고 부르나? '**Stage area**'라고 부른다.

\$ **git add f1.txt**

f1.txt가 **stage area**에 올라간다.

\$ **git commit**

**Stage**에 있는 파일들만 커밋한다. 즉, **stage area**에 있는 파일들만 현재 버전이 된다.

# 변경사항 확인하기

\$ git log -p

```
MINGW64:/e/practice/wsc/gitsam
commit 86ebfbaef45fab4c671f3f277fc777f537bb1d1 (HEAD -> master)
Author: skimok <skimok@gmail.com>
Date: Tue Feb 6 11:40:18 2018 +0900

    4

diff --git a/f1.txt b/f1.txt
index 2456b16..9462317 100644
--- a/f1.txt
+++ b/f1.txt
@@ -1,1 @@
-source : 2
+f1.txt : 2

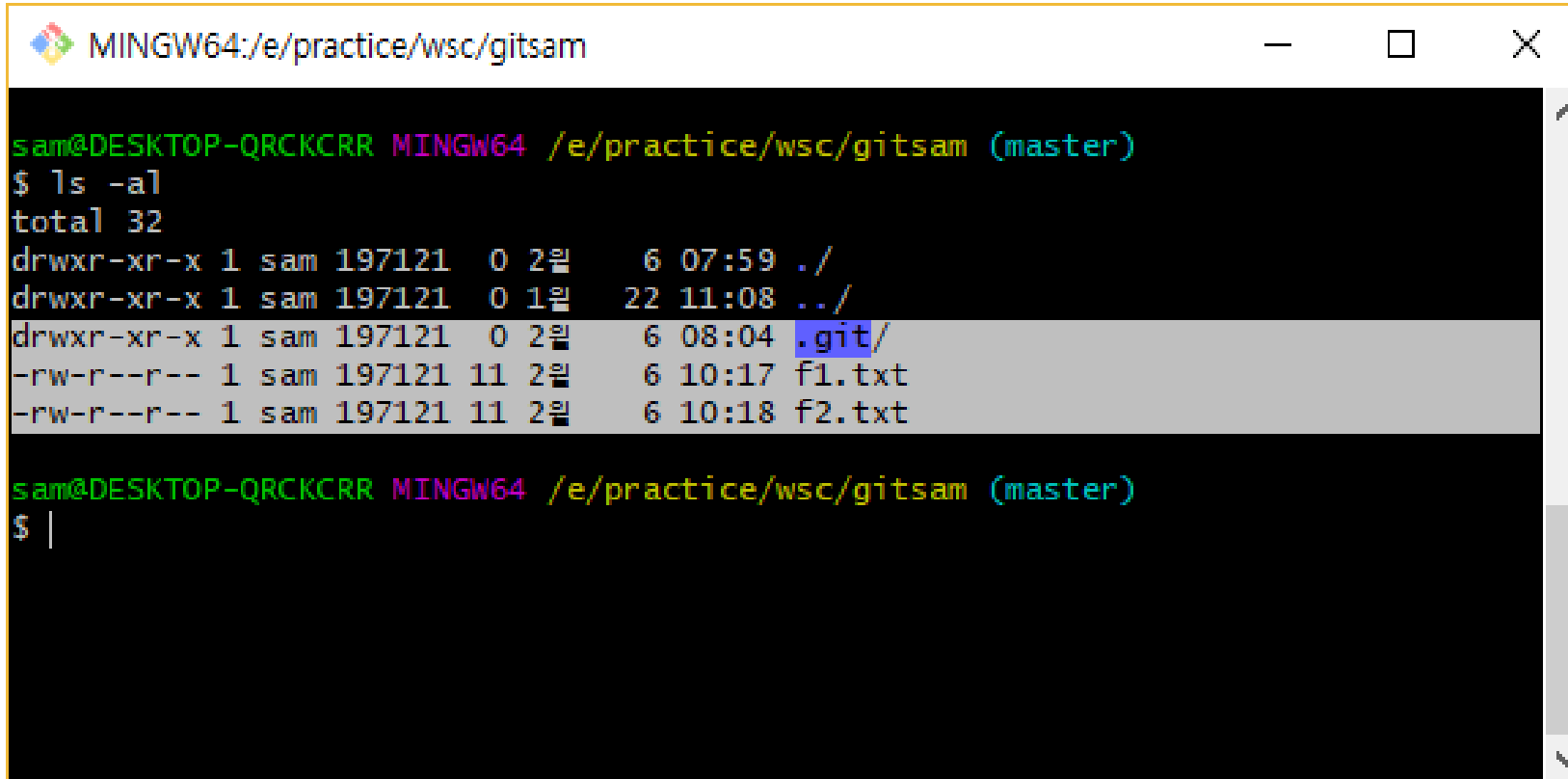
commit eb66866869c4403096c3f11f505463c490bdc9c5
Author: skimok <skimok@gmail.com>
Date: Tue Feb 6 10:10:22 2018 +0900

    3

diff --git a/f2.txt b/f2.txt
new file mode 100644
index 0000000..2456b16
--- /dev/null
+++ b/f2.txt
@@ -0,0 +1 @@
+source : 2

commit 3cf9373684b3a3d835126d563eaae4ea93cdaa29
:
```

# 과거의 버전으로 돌아가기

A screenshot of a Windows terminal window titled 'MINGW64:/e/practice/wsc/gitsam'. The prompt is 'sam@DESKTOP-QRCKCRR MINGW64 /e/practice/wsc/gitsam (master)'. The user has entered the command '\$ ls -al', and the output shows a directory listing. The '.git/' directory is highlighted. The user has then entered '\$ |' and the prompt is waiting for more input.

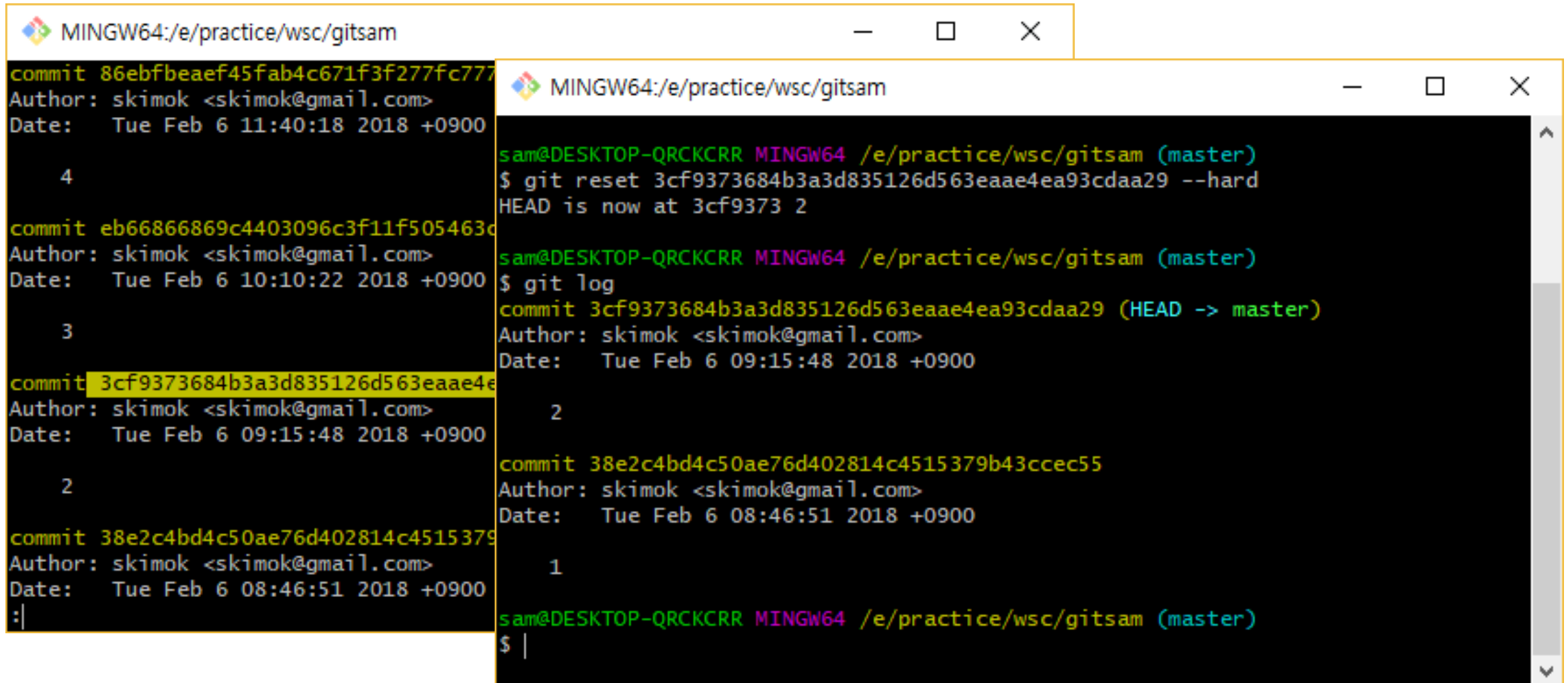
```
sam@DESKTOP-QRCKCRR MINGW64 /e/practice/wsc/gitsam (master)
$ ls -al
total 32
drwxr-xr-x 1 sam 197121  0 2월  6 07:59 ./
drwxr-xr-x 1 sam 197121  0 1월 22 11:08 ../
drwxr-xr-x 1 sam 197121  0 2월  6 08:04 .git/
-rw-r--r-- 1 sam 197121 11 2월  6 10:17 f1.txt
-rw-r--r-- 1 sam 197121 11 2월  6 10:18 f2.txt

sam@DESKTOP-QRCKCRR MINGW64 /e/practice/wsc/gitsam (master)
$ |
```

가장 원시적인 방법:

**.git/** 디렉토리를 포함하는 작업 디렉토리 전체를 복사해 놓으면 추후 과거 버전으로 돌아갈 수 있다.

\$ git reset 3cf9373684b3a3d835126d563eaae4ea93cdaa29 --hard



The image shows two overlapping terminal windows from a MINGW64 environment. The background window displays a list of Git commits, with the commit hash 3cf9373684b3a3d835126d563eaae4ea93cdaa29 highlighted in yellow. The foreground window shows the execution of the command 'git reset 3cf9373684b3a3d835126d563eaae4ea93cdaa29 --hard', which successfully resets the HEAD to the specified commit. Subsequent 'git log' and 'git reset' commands are also shown.

```
MINGW64:/e/practice/wsc/gitsam
commit 86ebfbaef45fab4c671f3f277fc777
Author: skimok <skimok@gmail.com>
Date: Tue Feb 6 11:40:18 2018 +0900
    4
commit eb66866869c4403096c3f11f505463d
Author: skimok <skimok@gmail.com>
Date: Tue Feb 6 10:10:22 2018 +0900
    3
commit 3cf9373684b3a3d835126d563eaae4ea93cdaa29
Author: skimok <skimok@gmail.com>
Date: Tue Feb 6 09:15:48 2018 +0900
    2
commit 38e2c4bd4c50ae76d402814c4515379b43ccec55
Author: skimok <skimok@gmail.com>
Date: Tue Feb 6 08:46:51 2018 +0900
    1
:|

MINGW64:/e/practice/wsc/gitsam
sam@DESKTOP-QRCKCRR MINGW64 /e/practice/wsc/gitsam (master)
$ git reset 3cf9373684b3a3d835126d563eaae4ea93cdaa29 --hard
HEAD is now at 3cf9373 2
sam@DESKTOP-QRCKCRR MINGW64 /e/practice/wsc/gitsam (master)
$ git log
commit 3cf9373684b3a3d835126d563eaae4ea93cdaa29 (HEAD -> master)
Author: skimok <skimok@gmail.com>
Date: Tue Feb 6 09:15:48 2018 +0900
    2
commit 38e2c4bd4c50ae76d402814c4515379b43ccec55
Author: skimok <skimok@gmail.com>
Date: Tue Feb 6 08:46:51 2018 +0900
    1
sam@DESKTOP-QRCKCRR MINGW64 /e/practice/wsc/gitsam (master)
$ |
```

