

Introduction to Git & Github

조형기

<https://johg15.github.io>

전북대학교 전자공학부

2022-1

What is Git & Github?

Version

흔한 대학생의 삶..

```
HW_조형기_v1.cpp  
HW_조형기_v2.cpp  
HW_조형기_v2(최종).cpp  
HW_조형기_v2(진짜최종).cpp  
...  
HW_조형기_final.cpp  
HW_조형기_final_final.cpp
```



Version



- 내가 원하는 시점, 원하는 버전에 자유롭게 이동 가능
- 디자인, 웹 개발 등등 어떤 프로젝트든지 가능

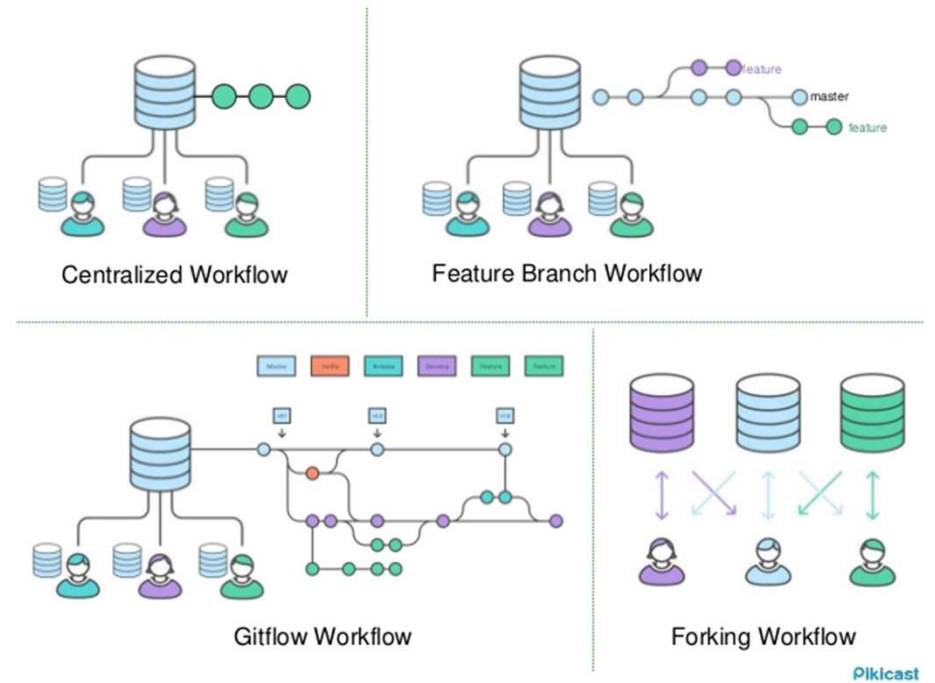
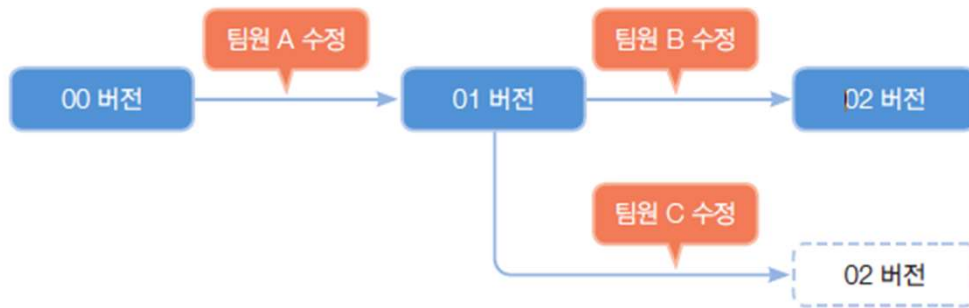
Version

여러 명이 같이 하는 Project라면..?

```
Project_조형기_v1.cpp  
Project_조형기_v2.cpp  
Project_조형기_v2(최종).cpp  
Project_조형기_v2(진짜최종).cpp  
Project_김북대_v2(최종).cpp  
Project_팀1_합본.cpp  
...  
Project_조형기_final.cpp  
Project_조형기_final_final.cpp  
Project_팀1_합본_최종.cpp
```



Version



- 여럿이 함께 작업하는 협업 프로젝트에서 더욱 강력함
- 따로 조금씩 작업하다가 원할 때 합치기 + 백업도 가능
- 기존 버전과 현재 버전을 비교하여 바뀌어 있는 부분 확인 가능

Powerful Git

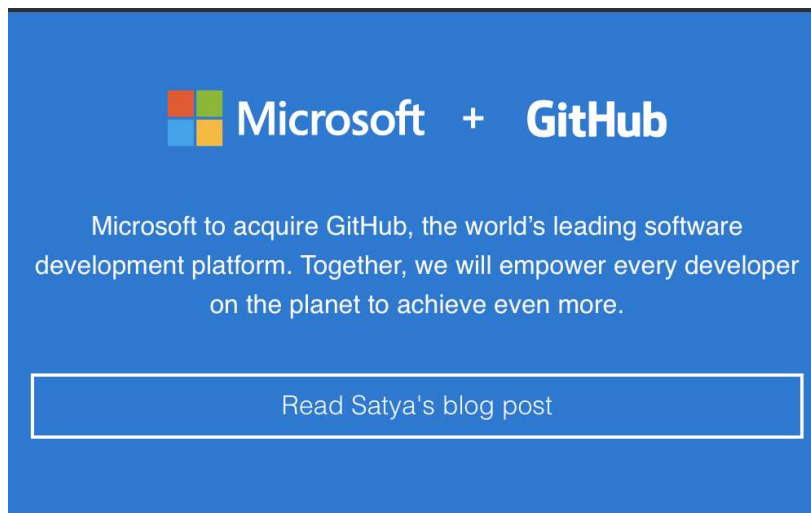
1.
코딩할 때 (코딩에만 한정되어 있지 않음)
단순히 ctrl+z를 눌러 이전상태로 되돌리는 것이 아니라,
원하는 시점마다 깃발을 꽂고(버전을 만들고)
이들 간에 **자유롭게 돌아다닐 수 있다.**

2.
내가 만든 버전 뿐만 아니라
동료가 만든 버전으로 이동할 수 있고,
동료와 내 버전을 **비교**해서
최신본으로 코드를 업데이트 할 수 있다.

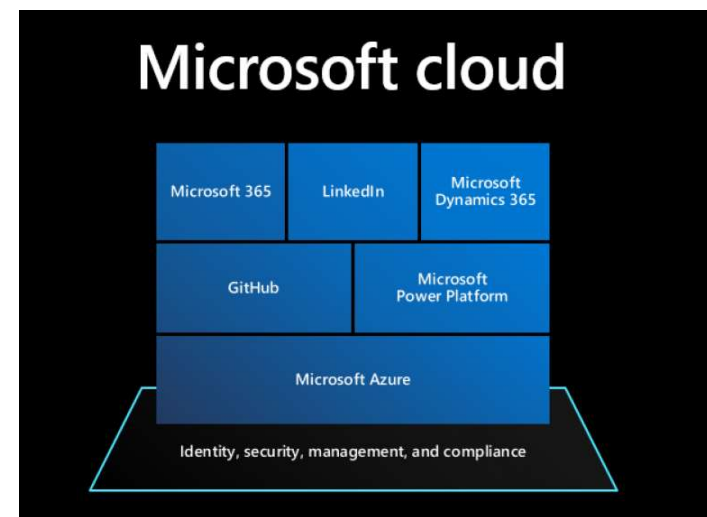
Git

➤ Git(버전이 관리되는 작업공간)은 어디에나 저장 가능함

- USB
- PC
- 특정 서버
- 클라우드
- 웹 (**GitHub**, GitLab, BitButcket)

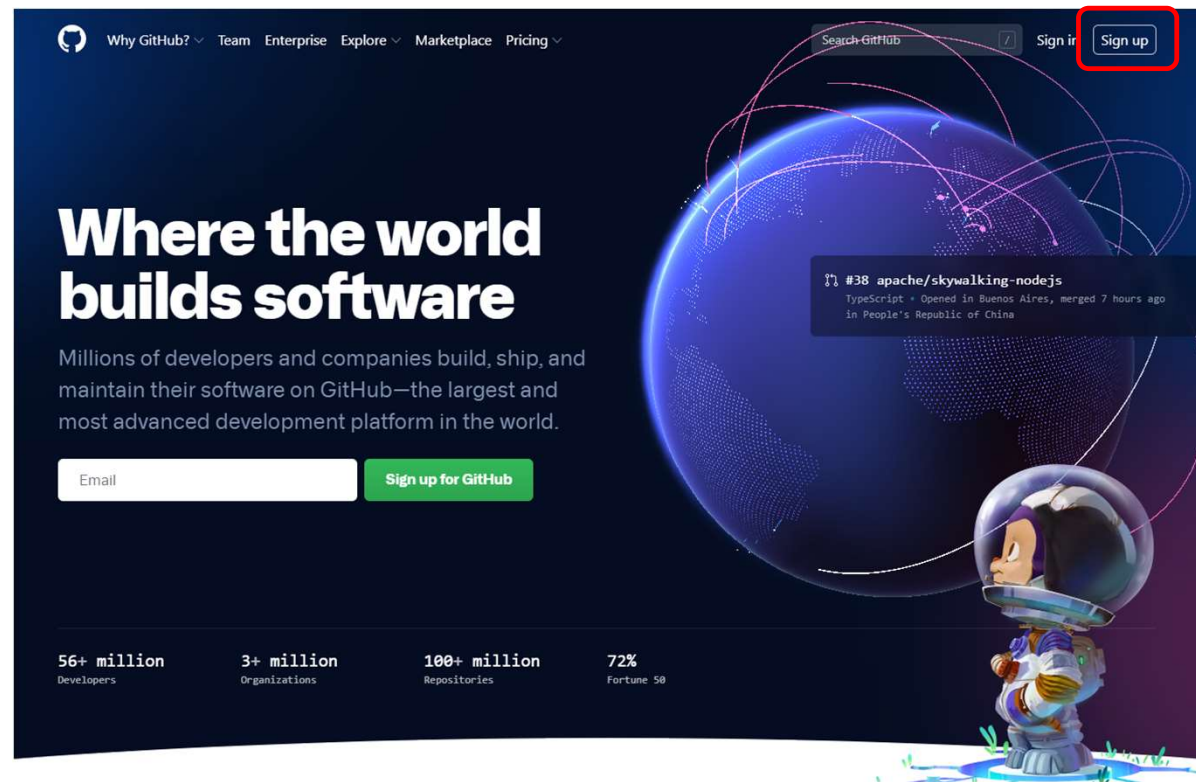


The slide features the Microsoft logo (four colored squares) followed by the text "Microsoft + GitHub". Below this, a paragraph states: "Microsoft to acquire GitHub, the world's leading software development platform. Together, we will empower every developer on the planet to achieve even more." At the bottom, there is a button with the text "Read Satya's blog post".



Sign up (GitHub)

1. GitHub 사이트 가입하기 <https://github.com>



Sign up (GitHub)

2. Account 생성

- **Email address, Username** – 본명(영문)으로
- Email Verification 완료하기

Join GitHub

Create your account

Username *

johg-jbnu ✓

Email address *

hygjo@jbnu.ac.kr ✓

Password *

***** ✓

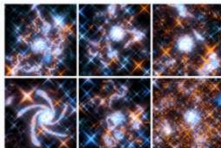
Make sure it's at least 15 characters OR at least 8 characters including a number and a lowercase letter. [Learn more.](#)

Email preferences

☐ Send me occasional product updates, announcements, and offers.

Verify your account

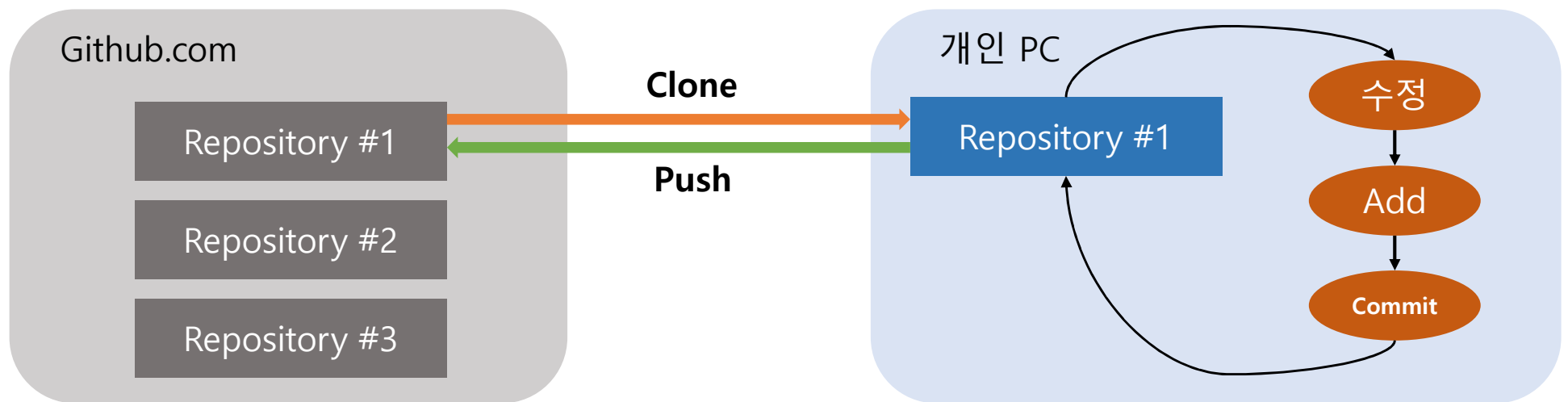
나선운하를 고르십시오



Create account

By creating an account, you agree to the [Terms of Service](#). For more information about GitHub's privacy practices, see the [GitHub Privacy Statement](#). We'll occasionally send you account-related emails.

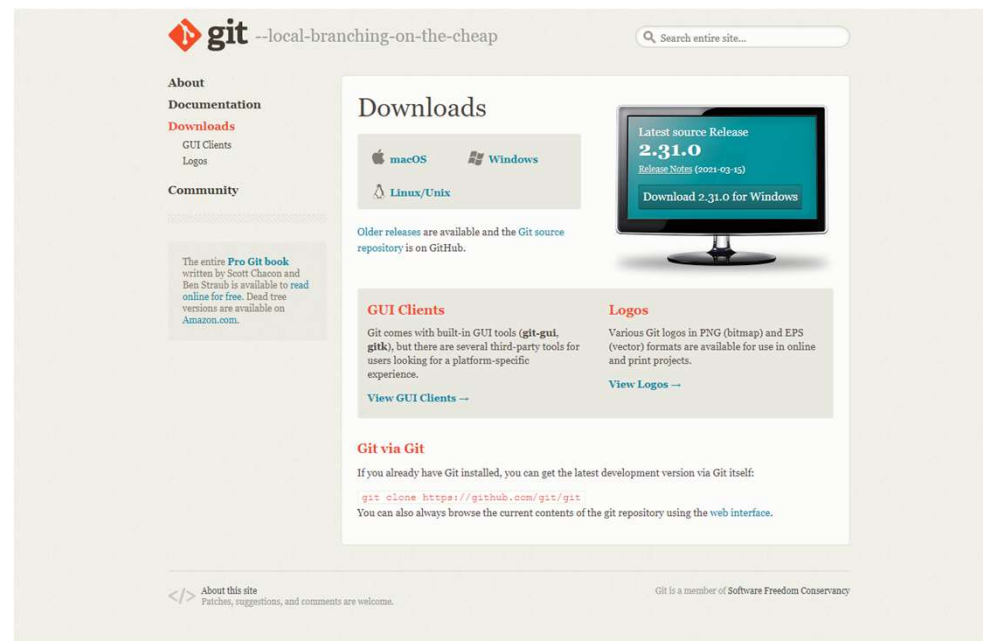
Git, GitHub



로컬에서 Git으로 관리하기

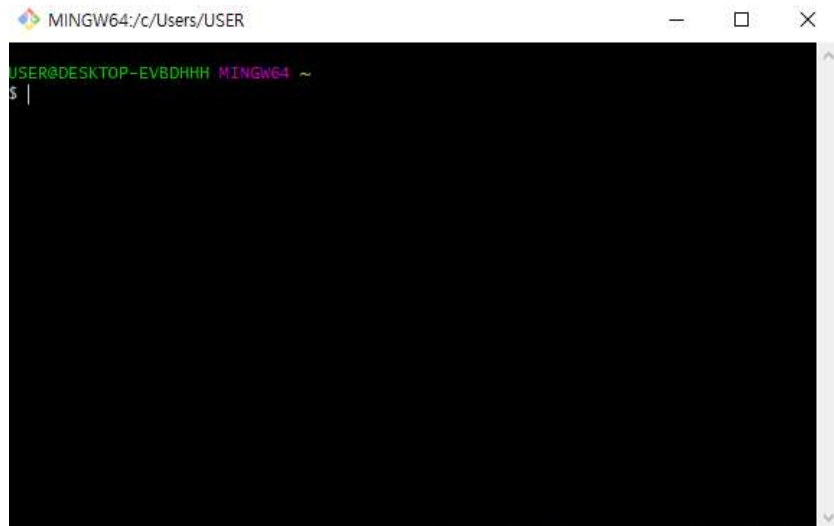
Install

1. Git 다운로드 (윈도우 버전) <https://git-scm.com/downloads>
2. Git 설치하기
 - 설정 바꾸지 않고 Next, Install 진행

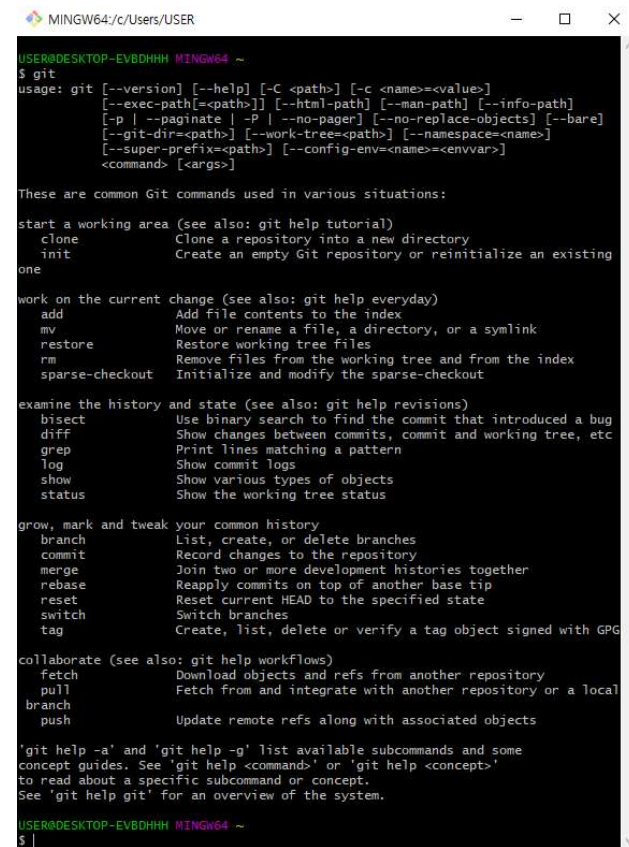


Install

1. 윈도우 시작 버튼 옆 돋보기 버튼
2. "Git Bash" 입력
3. "\$git" 입력



A screenshot of a Windows Start menu search. The search bar at the top contains the text "Git Bash". Below the search bar, a list of search results is displayed, including "Git Bash" and "Git Bash (x86)". The "Git Bash" result is highlighted.



A screenshot of a terminal window titled "MINGW64/c/Users/USER". The terminal shows the output of the command `$ git`. The output displays the usage of the `git` command, followed by a list of common Git commands and their descriptions. The commands are grouped into sections: "start a working area", "work on the current change", "examine the history and state", "grow, mark and tweak your common history", and "collaborate".

```
USER@DESKTOP-EVBDHHH MINGW64 ~
$ git
usage: git [--version] [--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>]
        [--super-prefix=<path>] [--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
  sparse-checkout  Initialize and modify the sparse-checkout

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

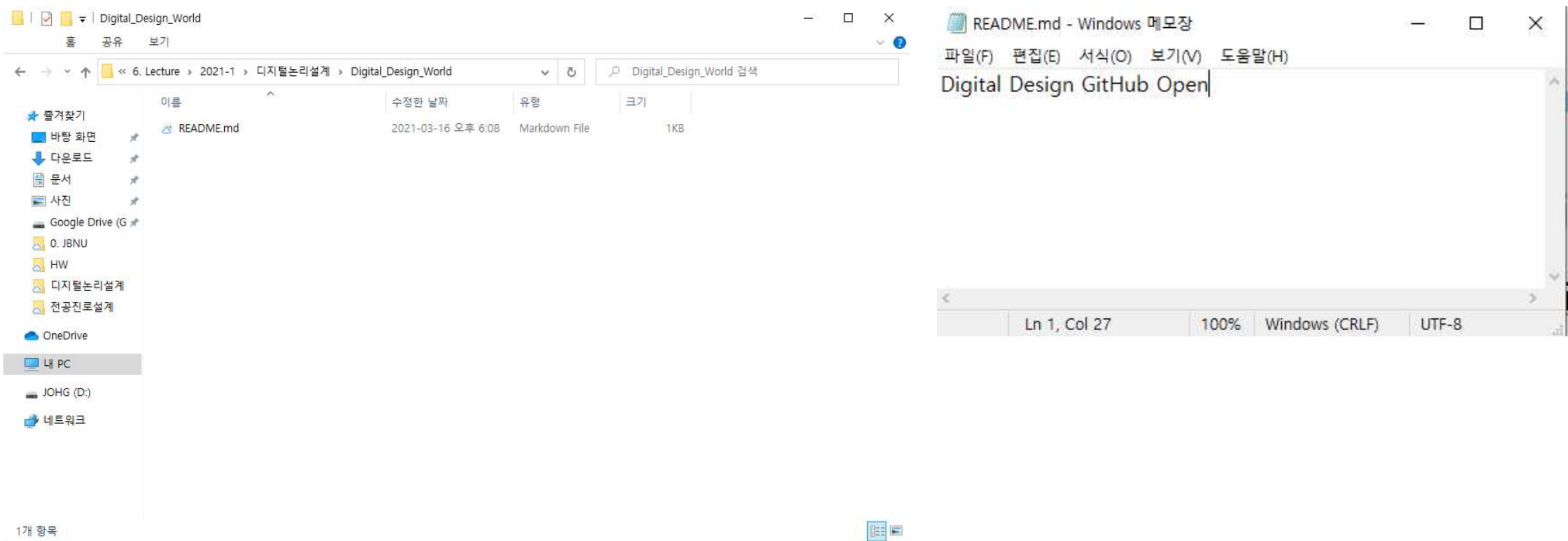
collaborate (see also: git help workflows)
  fetch      Download objects and refs from another repository
  pull       Fetch from and integrate with another repository or a local
  branch
  push       Update remote refs along with associated objects

'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.

USER@DESKTOP-EVBDHHH MINGW64 ~
$
```

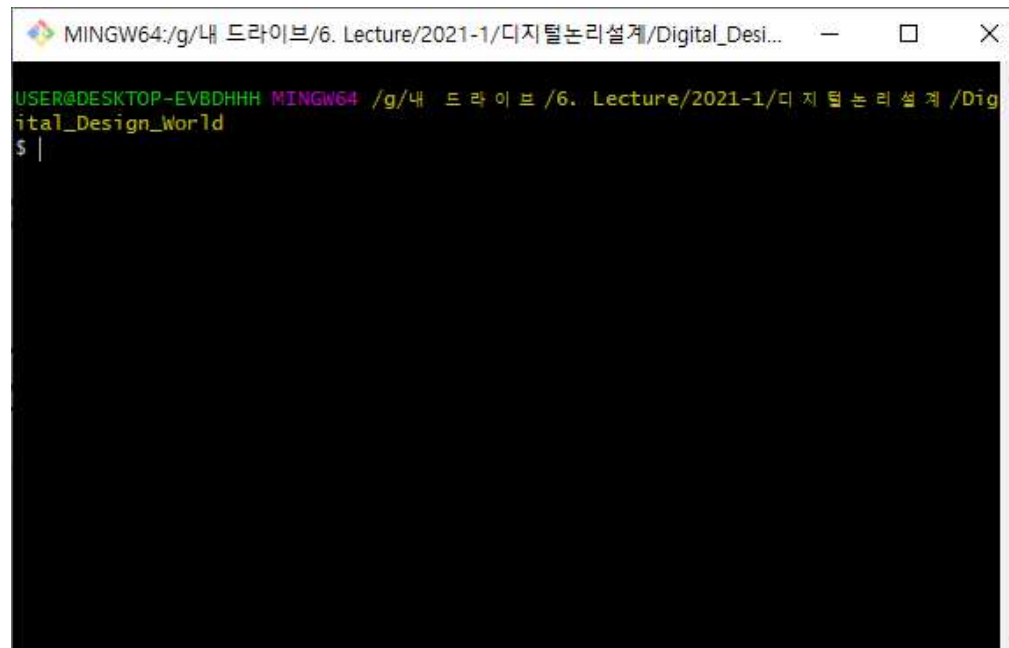
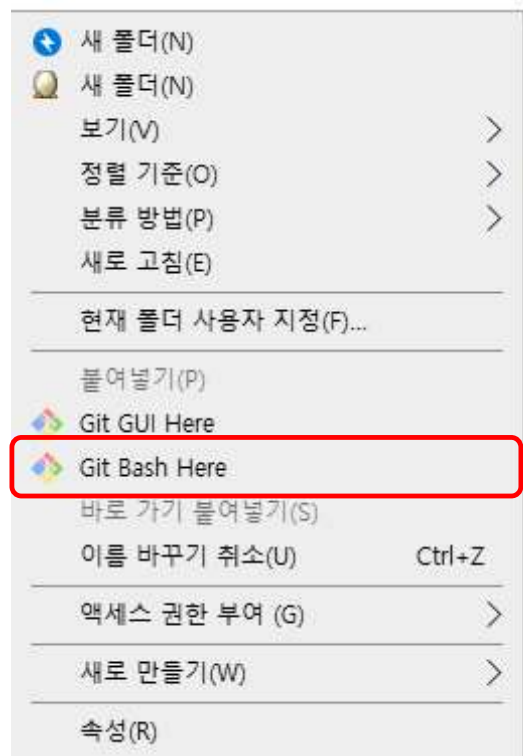
Sample Project - Init

1. 작업 폴더 생성
2. 메모장으로 README.md 파일 생성 (아무내용이나)



Sample Project - Init

3. 작업 폴더 안에서 오른쪽 클릭
4. 해당 폴더 경로의 터미널이 열림



Sample Project - Init

5. Git 초기화

\$ git init

Git Repository 생성됨 (.git 폴더가 생성)

```
MINGW64:/g/내 드라이브/6. Lecture/2021-1/디지털논리설계/Digital_Design_World
$ git init
Initialized empty Git repository in G:/내 드라이브/6. Lecture/2021-1/디지털논리설계/Digital_Design_World/.git/
$ git init
Initialized empty Git repository in G:/내 드라이브/6. Lecture/2021-1/디지털논리설계/Digital_Design_World/.git/
$
```



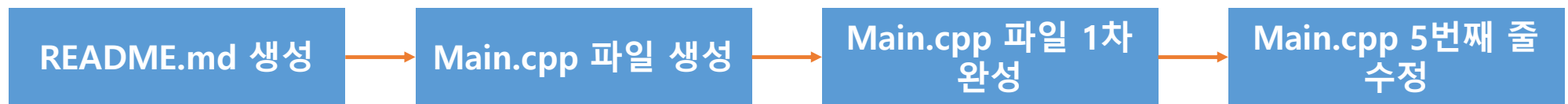
Commit 커밋

➤ 커밋이란?

- 하나의 변경 사항 단위
- Ex) xxx_yyy.cpp 5번째 줄 수정
- Ex) README.md 파일 생성

➤ 꾸준히 쌓이는 커밋

- 어떤 의미있는 변동사항이 있을 때마다 커밋을 하면 된다.
- `$ git commit -m "어떤 변동사항이 있었는지 메시지 간단하게"`



Sample Project

6. 커밋 Commit

- README.md 파일을 하나의 버전으로 저장
- RPG 게임에서 중간 저장이라고 생각

6-1. 계정 등록

- Email address, Username

```
$ git config --global user.email "your_email@gmail.com"
```

```
$ git config --global user.name "username"
```

```
USER@DESKTOP-EVBDHHH MINGW64 /g/내 드라이브/6. Lecture/2021-1/디지털논리설계/Digital_Design_World (master)
$ git config --global user.email "hygijo@gmail.com"

USER@DESKTOP-EVBDHHH MINGW64 /g/내 드라이브/6. Lecture/2021-1/디지털논리설계/Digital_Design_World (master)
$ git config --global user.name "johg15"
```

Sample Project

6-2. 커밋할 파일 선택

- 특정 파일 선택 또는 전체를 지정할 수 있음

```
$ git add README.txt
```

```
$ git add .
```

```
USER@DESKTOP-EVBDHHH MINGW64 /g/내 드라이브/6. Lecture/2021-1/디지털논리설계/Digital_Design_World (master)
$ git add .
```

6-3. 커밋하기

- 어떤 커밋인지 설명 메시지 간략하게 넣어서 커밋하기

```
$ git commit -m "readme 파일 생성"
```

```
USER@DESKTOP-EVBDHHH MINGW64 /g/내 드라이브/6. Lecture/2021-1/디지털논리설계/Digital_Design_World (master)
$ git commit -m "readme 파일 생성"
[master (root-commit) 003de72] readme 파일 생성
1 file changed, 1 insertion(+)
create mode 100644 README.md
```

Sample Project

6-4. 두 번째 커밋하기

- 파일 추가하고 커밋해보기

\$ git add .

\$ git commit -m "second file added"

```
USER@DESKTOP-EVBDHHH MINGW64 /g/내 드라이브/6. Lecture/2021-1/디지털논리설계/Digital_Design_World (master)
$ git add .

USER@DESKTOP-EVBDHHH MINGW64 /g/내 드라이브/6. Lecture/2021-1/디지털논리설계/Digital_Design_World (master)
$ git commit -m "second file add"
[master 8e67e13] second file add
1 file changed, 76 insertions(+)
create mode 100644 AND_gate.circ
```

Sample Project

6-5. 시간 여행

- 지금까지 커밋 확인
- 첫 번째 커밋 ID 앞 7자리 확인

\$ git log

```
USER@DESKTOP-EVBDHHH MINGW64 /g/내 드라이브/6. Lecture/2021-1/디지털논리설계/Digital_Design_World (master)
$ git log
commit 8e67e137cb53b7133060be4a43d91fe127df99ba (HEAD -> master)
Author: johg15 <hygijo@gmail.com>
Date: Tue Mar 16 18:46:41 2021 +0900

    second file add

commit 003de7203452c108cf764b7e57aa710943923ce6
Author: johg15 <hygijo@gmail.com>
Date: Tue Mar 16 18:41:04 2021 +0900

    readme 파일 생성
```

Sample Project

6-6. 시간 여행

- 첫 번째 커밋으로 돌아가기

\$ git checkout 003de720

```
USER@DESKTOP-EVBDHHH MINGW64 /g/내 드라이브/6. Lecture/2021-1/디지털논리설계/Digital_Design_World (master)
$ git checkout 003de720
Note: switching to '003de720'.

You are in 'detached HEAD' state. You can look around, make experimental
changes and commit them, and you can discard any commits you make in this
state without impacting any branches by switching back to a branch.

If you want to create a new branch to retain commits you create, you may
do so (now or later) by using -c with the switch command. Example:

    git switch -c <new-branch-name>

Or undo this operation with:

    git switch -

Turn off this advice by setting config variable advice.detachedHead to false

HEAD is now at 003de72 readme 파일 생성
```

GitHub 원격저장소

GitHub 원격저장소

➤Repository 생성

The screenshot shows the GitHub profile of user 'johg15'. The 'Repositories' tab is selected and highlighted with a red box and the number '1.'. In the top right corner, the 'New' button is highlighted with a red box and the number '2.'. The profile section on the left includes the user's avatar, name 'johg15', an 'Edit profile' button, and statistics: 0 followers, 0 following, and 2 stars. Below this is the 'Organizations' section with three icons. The main content area displays a list of repositories:

- johg15.github.io**: HTML, ISC License, Updated on 4 Mar. Star button.
- Shift-GCN-HCR**: Private, Updated on 20 Nov 2020. Star button.
- Azure-Kinect-Samples**: Forked from microsoft/Azure-Kinect-Samples, Samples for Azure Kinect, C++, 186 forks, MIT License, Updated on 17 Nov 2020. Star button.
- Shift-GCN**: Forked from kchengiva/Shift-GCN, The implementation for "Skeleton-Based Action Recognition with Shift Graph Convolutional Network" (CVPR2020 oral), Python, 32 forks, Other, Updated on 25 Sep 2020. Star button.
- test**: Python, Updated on 21 Sep 2020. Star button.

GitHub 원격저장소

➤Repository 이름

Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository.](#)

Repository template

Start your repository with a template repository's contents.

No template ▾

Owner *

johg15 ▾

Repository name *

Digital-Design ✓

Great repository names are short and memorable. Need inspiration? How about [super-engine?](#)

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:

Skip this step if you're importing an existing repository.

☐ Add a README file

This is where you can write a long description for your project. [Learn more.](#)

☐ Add .gitignore

Choose which files not to track from a list of templates. [Learn more.](#)

☐ Choose a license





A license tells others what they can and can't do with your code. [Learn more.](#)

Create repository

GitHub 원격저장소

➤ Quick Setup

Quick setup — if you've done this kind of thing before

 Set up in Desktop or  HTTPS  SSH 

Get started by [creating a new file](#) or [uploading an existing file](#). We recommend every repository include a [README](#), [LICENSE](#), and [.gitignore](#).

1. ...or create a new repository on the command line

```
echo "# Digital-Design" >> README.md
git init
git add README.md
git commit -m "first commit"
git branch -M main
git remote add origin https://github.com/johg15/Digital-Design.git
git push -u origin main
```

2. ...or push an existing repository from the command line

```
git remote add origin https://github.com/johg15/Digital-Design.git
git branch -M main
git push -u origin main
```

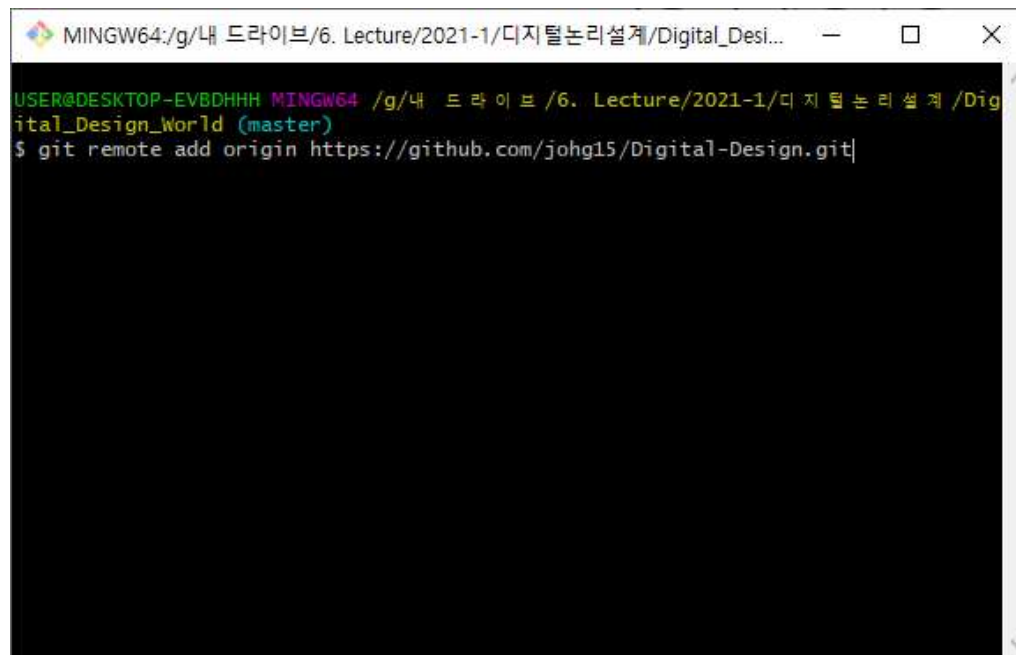
...or import code from another repository

You can initialize this repository with code from a Subversion, Mercurial, or TFS project.

GitHub 원격저장소

➤ Quick Setup

- 로컬과 원격저장소 연결
- `$git remote add origin https://github.com/(your_username)/(Repository_name).git`

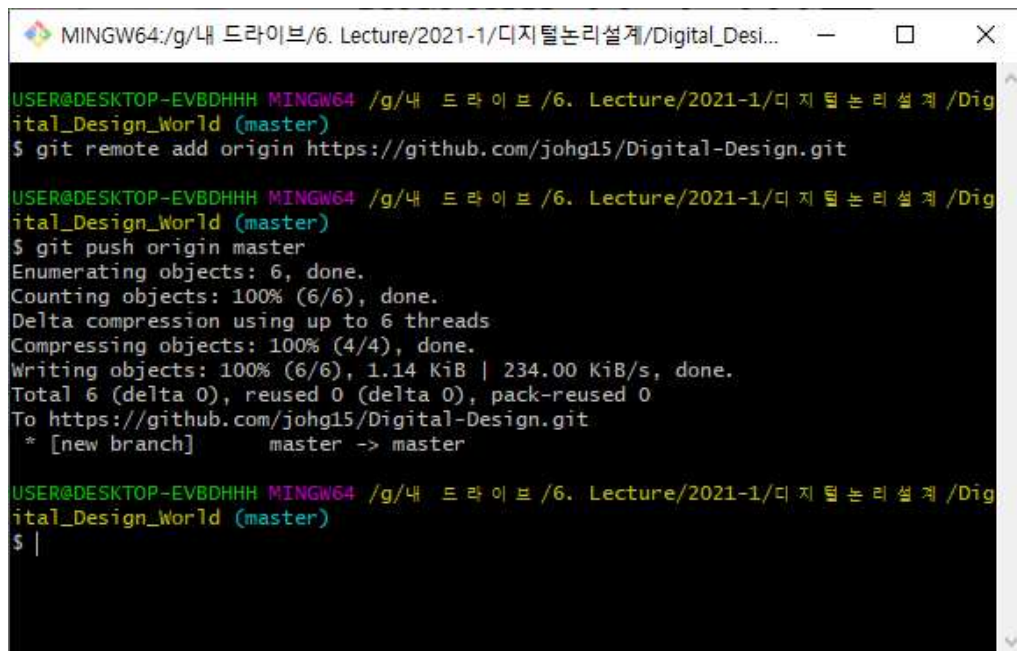


```
MINGW64:/g/내 드라이브/6. Lecture/2021-1/디지털논리설계/Digital_Desi...
USER@DESKTOP-EVBDHHH MINGW64 /g/내 드라이브/6. Lecture/2021-1/디지털논리설계/Dig
ital_Design_World (master)
$ git remote add origin https://github.com/johg15/Digital-Design.git
```

GitHub 원격저장소

➤ Quick Setup

- 로컬 저장소의 커밋을 push 명령어로 원격저장소에 올리기
- `$git push origin master`



```
MINGW64:/g/내 드라이브/6. Lecture/2021-1/디지털논리설계/Digital_Desi...
USER@DESKTOP-EVBDHHH MINGW64 /g/내 드라이브/6. Lecture/2021-1/디지털논리설계/Dig
ital_Design_World (master)
$ git remote add origin https://github.com/johg15/Digital-Design.git

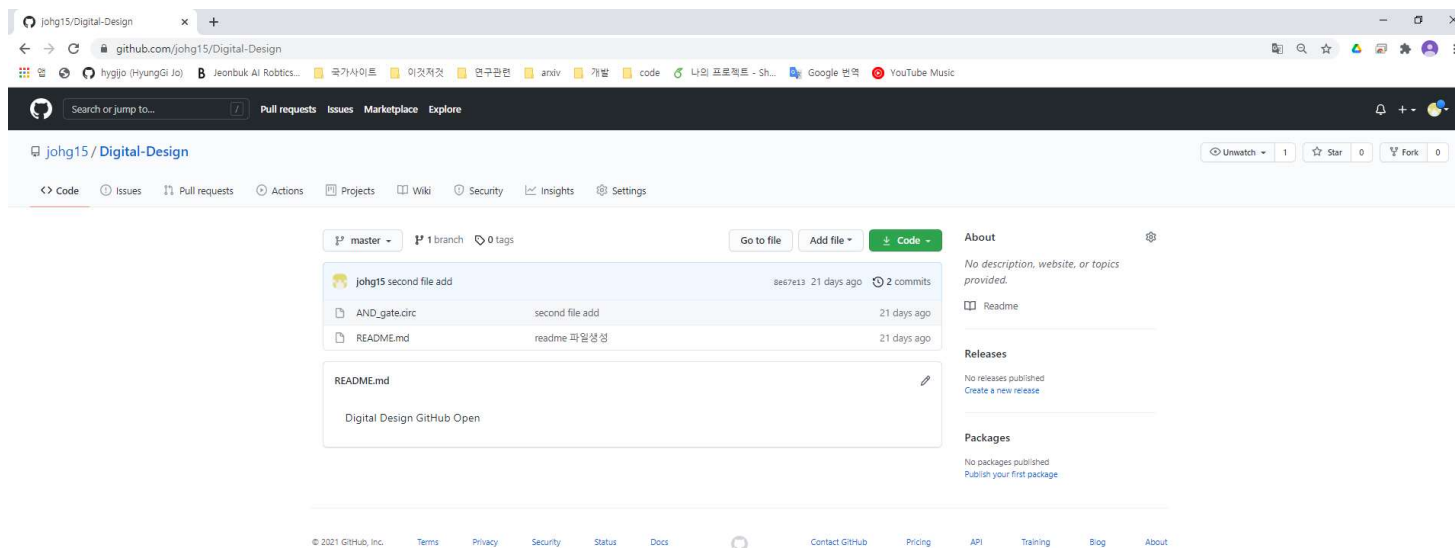
USER@DESKTOP-EVBDHHH MINGW64 /g/내 드라이브/6. Lecture/2021-1/디지털논리설계/Dig
ital_Design_World (master)
$ git push origin master
Enumerating objects: 6, done.
Counting objects: 100% (6/6), done.
Delta compression using up to 6 threads
Compressing objects: 100% (4/4), done.
Writing objects: 100% (6/6), 1.14 KiB | 234.00 KiB/s, done.
Total 6 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/johg15/Digital-Design.git
 * [new branch]      master -> master

USER@DESKTOP-EVBDHHH MINGW64 /g/내 드라이브/6. Lecture/2021-1/디지털논리설계/Dig
ital_Design_World (master)
$ |
```

GitHub 원격저장소

➤ Quick Setup

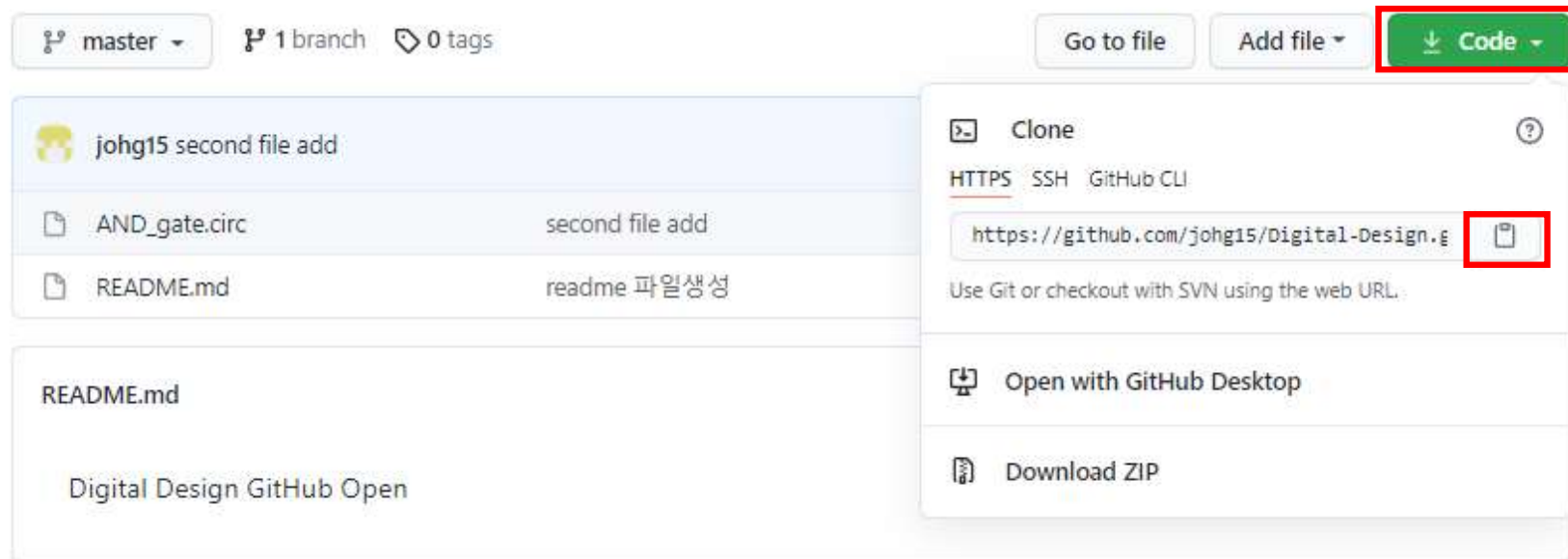
- GitHub에서 잘 올라왔는지 확인



GitHub 내려받기

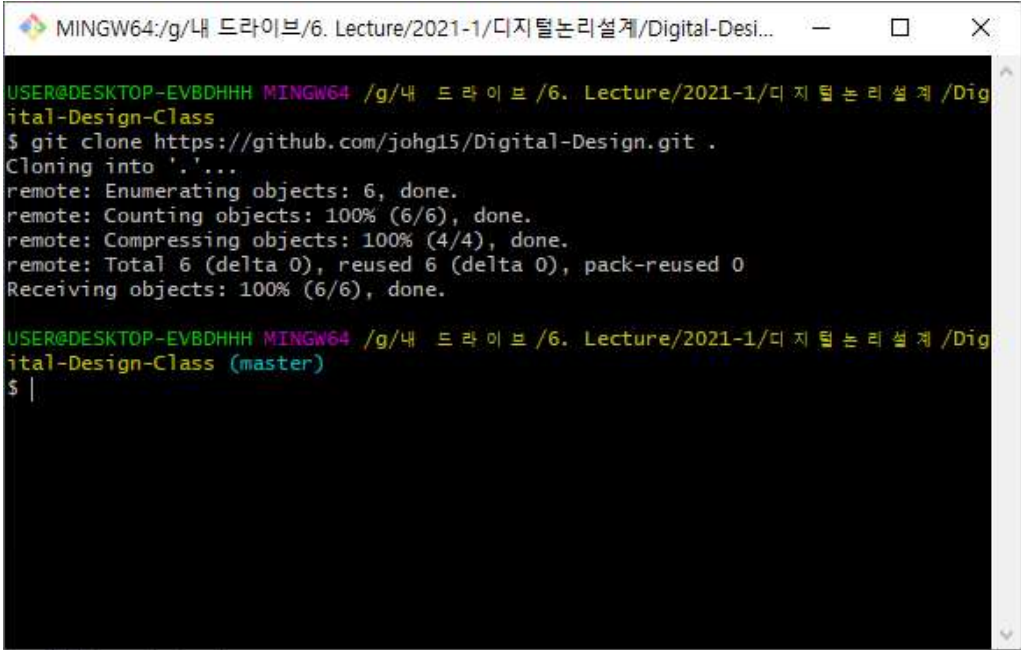
➤ 원격저장소에서 내려받기 (Clone)

- 다른 새로운 폴더 생성 (Creative-Eng-Class) → Git Bash Here



GitHub 내려받기

- 원격저장소에서 내려받기 (Clone)
 - 다른 새로운 폴더 생성 (Creative-Eng-Class) → Git Bash Here



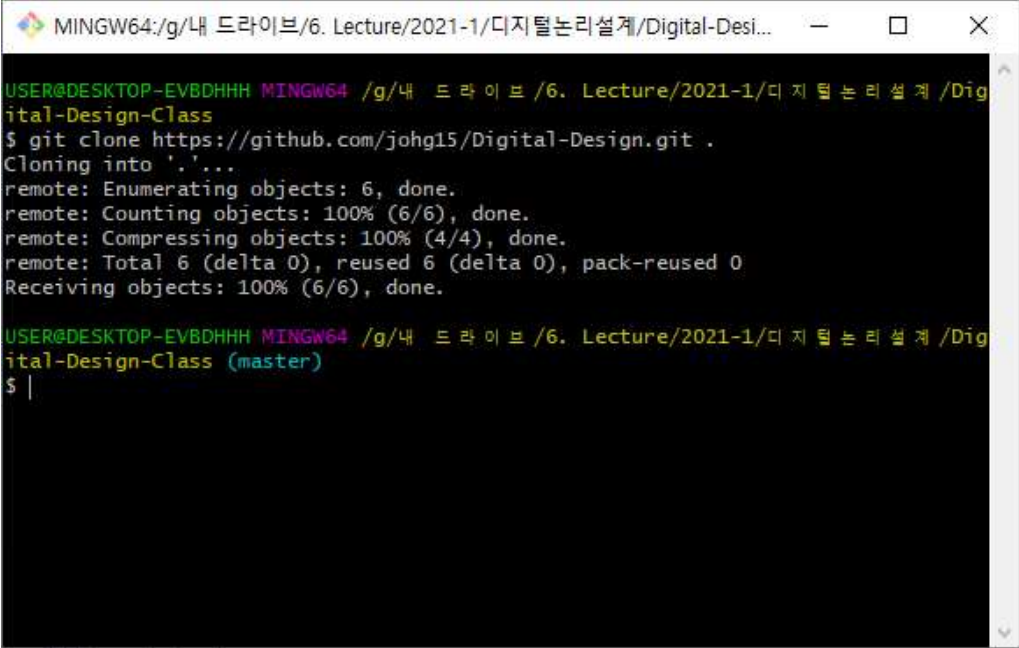
```
MINGW64:/g/내 드라이브/6. Lecture/2021-1/디지털논리설계/Digital-Desi...
USER@DESKTOP-EVBDHHH MINGW64 /g/내 드라이브/6. Lecture/2021-1/디지털논리설계/Dig
ital-Design-Class
$ git clone https://github.com/johg15/Digital-Design.git .
Cloning into '.'...
remote: Enumerating objects: 6, done.
remote: Counting objects: 100% (6/6), done.
remote: Compressing objects: 100% (4/4), done.
remote: Total 6 (delta 0), reused 6 (delta 0), pack-reused 0
Receiving objects: 100% (6/6), done.

USER@DESKTOP-EVBDHHH MINGW64 /g/내 드라이브/6. Lecture/2021-1/디지털논리설계/Dig
ital-Design-Class (master)
$ |
```


GitHub 내려받기

➤ 원격저장소에서 내려받기 (Clone)

- \$git clone (복사한 주소) .
- 내려받기 확인



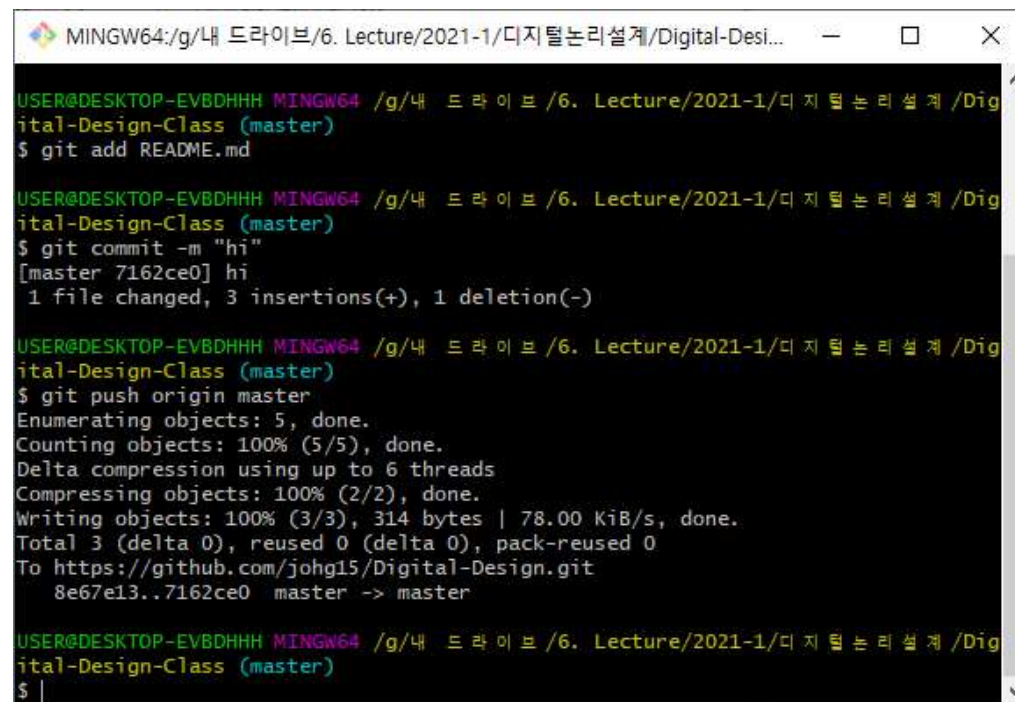
```
MINGW64:/g/내 드라이브/6. Lecture/2021-1/디지털논리설계/Digital-Desi...
USER@DESKTOP-EVBDHHH MINGW64 /g/내 드라이브/6. Lecture/2021-1/디지털논리설계/Dig
ital-Design-Class
$ git clone https://github.com/johg15/Digital-Design.git .
Cloning into '.'...
remote: Enumerating objects: 6, done.
remote: Counting objects: 100% (6/6), done.
remote: Compressing objects: 100% (4/4), done.
remote: Total 6 (delta 0), reused 6 (delta 0), pack-reused 0
Receiving objects: 100% (6/6), done.

USER@DESKTOP-EVBDHHH MINGW64 /g/내 드라이브/6. Lecture/2021-1/디지털논리설계/Dig
ital-Design-Class (master)
$ |
```

GitHub 올리기

➤로컬저장소에서 수정 후 올리기

- \$git add README.md
- \$git commit -m "hi"
- \$git push origin master



```
MINGW64:/g/내 드라이브/6. Lecture/2021-1/디지털논리설계/Digital-Desi...
USER@DESKTOP-EVBDHHH MINGW64 /g/내 드라이브/6. Lecture/2021-1/디지털논리설계/Dig
ital-Design-Class (master)
$ git add README.md

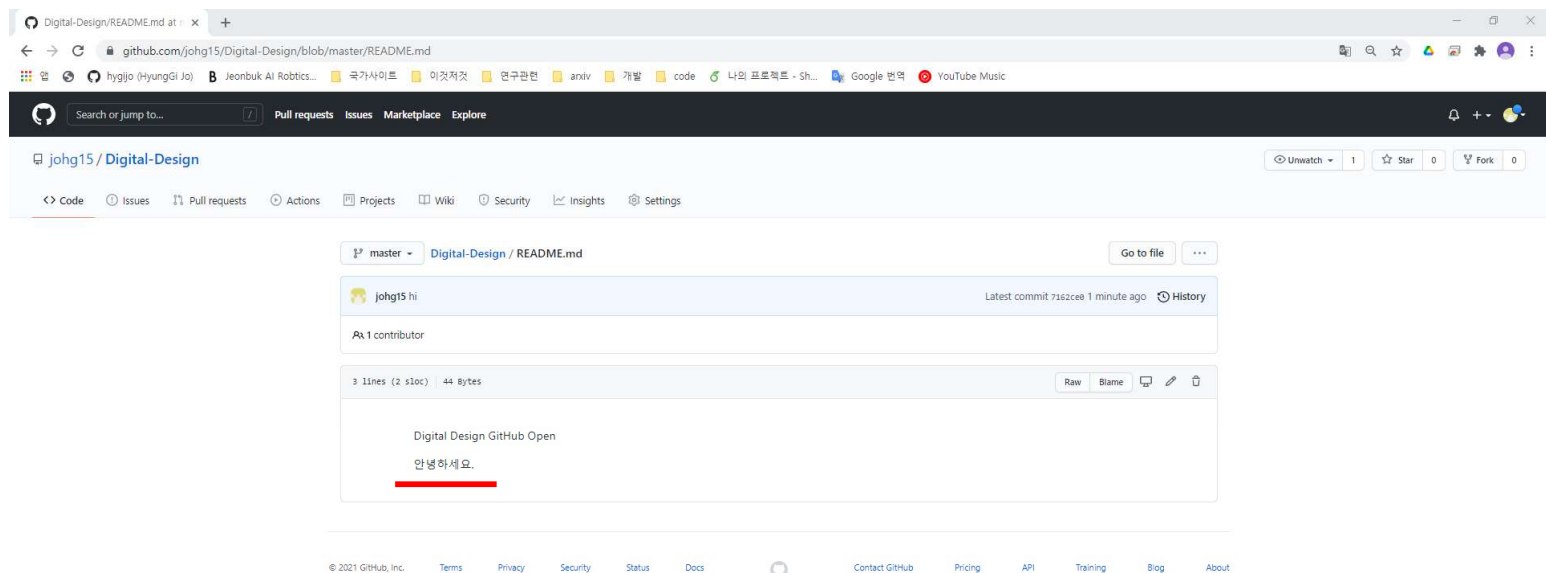
USER@DESKTOP-EVBDHHH MINGW64 /g/내 드라이브/6. Lecture/2021-1/디지털논리설계/Dig
ital-Design-Class (master)
$ git commit -m "hi"
[master 7162ce0] hi
1 file changed, 3 insertions(+), 1 deletion(-)

USER@DESKTOP-EVBDHHH MINGW64 /g/내 드라이브/6. Lecture/2021-1/디지털논리설계/Dig
ital-Design-Class (master)
$ git push origin master
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 6 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 314 bytes | 78.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/johg15/Digital-Design.git
8e67e13..7162ce0 master -> master

USER@DESKTOP-EVBDHHH MINGW64 /g/내 드라이브/6. Lecture/2021-1/디지털논리설계/Dig
ital-Design-Class (master)
$ |
```

GitHub 올리기

➤로컬저장소에서 수정 후 올리기



GitHub 다시 내려받기

- 수정된 사항 다시 내려받기
 - 작업 폴더에서 다시 내려받기
 - `$git pull origin master`

A terminal window titled 'MINGW64:/g/내 드라이브/6. Lecture/2021-1/디지털논리설계/Digital_Desi...' displays the output of a 'git pull' command. The output shows the process of fetching and merging updates from the remote repository 'origin/master'. It includes details about object counts, sizes, and a diff summary for 'README.md' showing 4 additions and 1 deletion. The terminal text is as follows:

```
Writing objects: 100% (6/6), 1.14 KiB | 234.00 KiB/s, done.
Total 6 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/johg15/Digital-Design.git
* [new branch]      master -> master

USER@DESKTOP-EVBDHHH MINGW64 /g/내 드라이브/6. Lecture/2021-1/디지털논리설계/Digital_Design_World (master)
$ git pull origin master
remote: Enumerating objects: 5, done.
remote: Counting objects: 100% (5/5), done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 3 (delta 0), reused 3 (delta 0), pack-reused 0
Unpacking objects: 100% (3/3), 294 bytes | 6.00 KiB/s, done.
From https://github.com/johg15/Digital-Design
* branch            master       -> FETCH_HEAD
   8e67e13..7162ce0  master       -> origin/master
Updating 8e67e13..7162ce0
Fast-forward
 README.md | 4 +++-
 1 file changed, 3 insertions(+), 1 deletion(-)
```