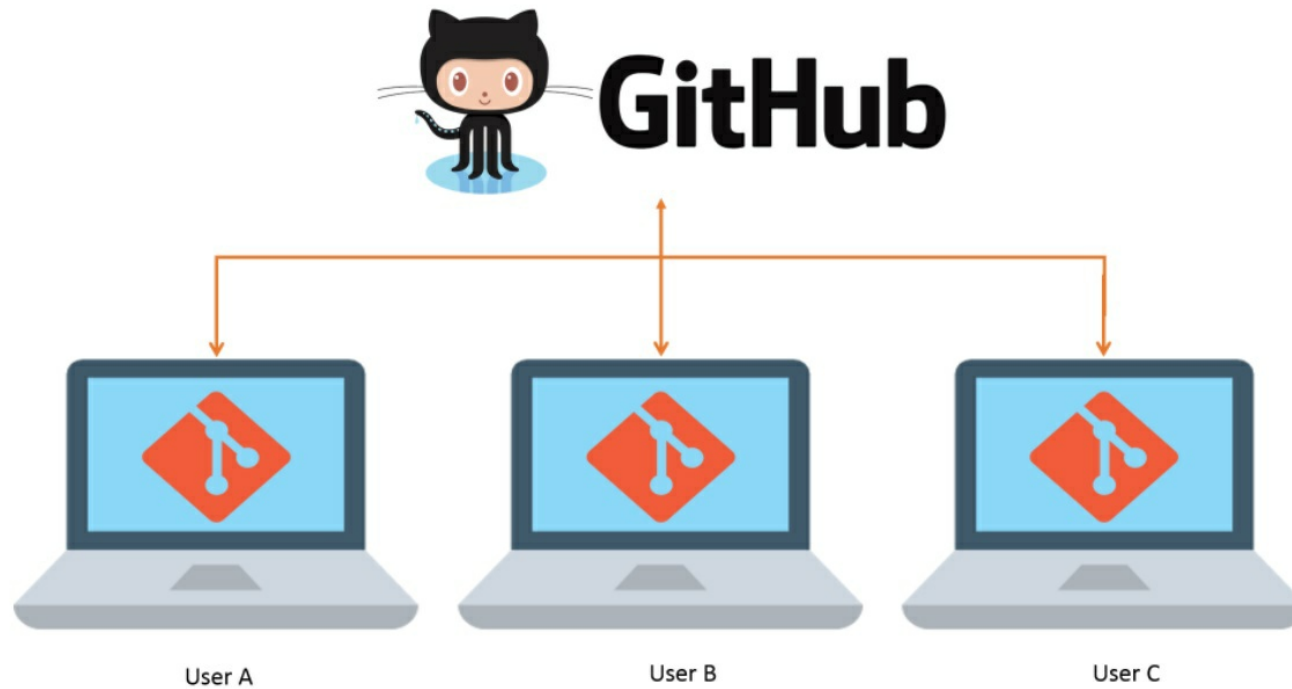




Git / GitHub

Part 0. GitHub 기초



1 Git, 그리고 GitHub

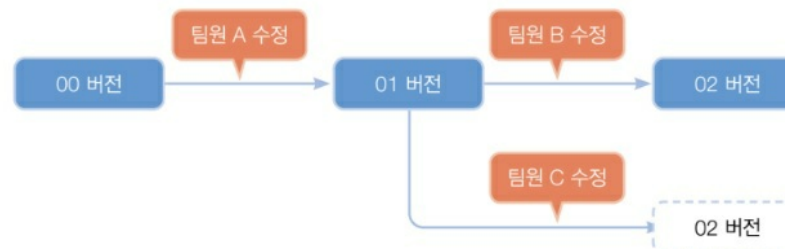
★ 버전 관리란?

여러 사람이 함께 만드는 프로그램이라면 팀 프로젝트에서 사용할 공용 폴더를 만들고 그곳에 소스를 올려진 최초의 소스코드를 00버전이라고 할 것이다. 팀원 A가 소스를 수정하여 01버전이라고 하고, 다시 팀원 B가 그 소스를 수정하여 02버전으로 저장할 수 있을 것이다. 이런 일련의 과정을 버전 관리라 할 수 있다. 그런데 팀원 B와 D가 동시에 02버전을 수정해서 저장한다면 어떤 일이 벌어질까?

내가 원하는 시점마다 깃발을 꽂고, 깃발이 꽂힌 시점으로 자유롭게 이동할 수 있다면 편안하게 새로운 소스코드를 추가하거나 삭제할 수 있을 것이다. 소스코드 오류가 난다면 바로 전에 꽂은 깃발 시점으로 돌아가면 되니까. 이를 가능하게 해주는 소스코드 버전 관리 시스템이 바로 **Git**이다.

깃은 데이터를 저장할 공간만 있다면 어디서나 사용할 수 있다. 자신의 PC나 USB나 드롭박스든 클라우드 서버인 구글 드라이브든... 만약 구글 드라이브에 올려놓는다면 인터넷을 통해 팀원들과 함께 소스를 공유하며 관리할 수 있을 것이다. 이렇게 깃으로 관리하는 프로젝트를 올려놓을 수 있는 깃 호스팅 중 하나가 **GitHub**이다. 깃으로 관리하는 사이트는 깃허브 외에도 GitLab, BitButcket 등 다양하다.

누구든지 협업할 수 있는 공개저장소 프로젝트를 **오픈소스**라고 한다. 구글이 만드는 기계학습 라이브러리인 텐서플로, 애플이 만든 언어 스위프트, Vue.js 등 수천 만개의 공개저장소가 있다.



1 Git, 그리고 GitHub

✧ GitHub 가입하기

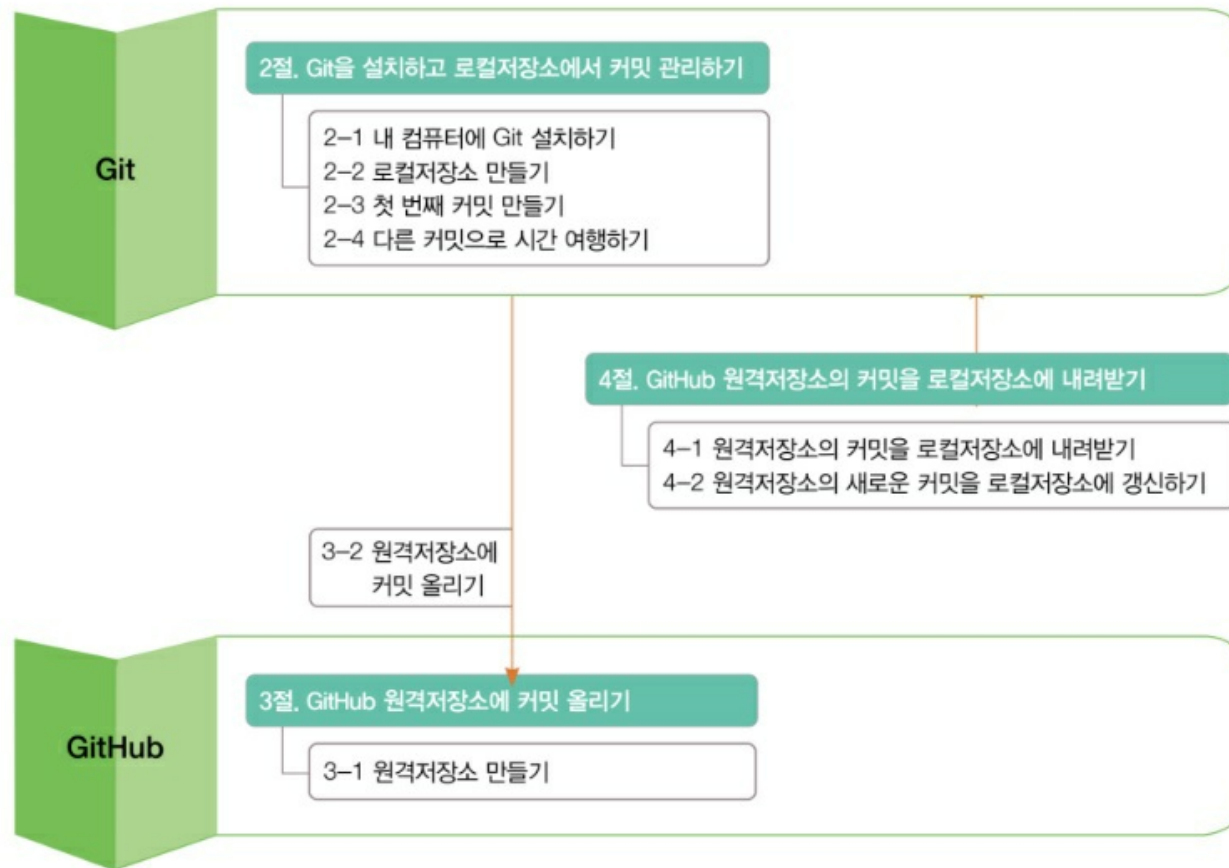
깃허브에 가입해보자. 네이버나 구글에 가입하는 과정과 유사하다.

1. <https://github.com>에 접속하여 [Sign Up] 클릭한다.
2. [Enter your email]에 이메일 입력하고 [Continue]버튼을 클릭한다.
3. [Create a password]에 패스워드 입력하고 [Continue]버튼을 클릭한다.
4. [Enter a username]에 이름 입력하고 [Continue]버튼을 클릭한다.
5. 공지 이메일을 받을지 물어보면 'y' 입력하고 [Continue]버튼을 클릭한다.
6. 사람임을 인증하는 단계에서 나선 은하 두번 정도 선택하면 v로 체크되고 인증이 완료된다. 인증 후 [Create account] 버튼을 클릭.
7. 입력했던 이메일을 열어보면 GitHub에서 보낸 코드를 확인할 수 있다. [Enter code]에 해당 코드를 입력한다.
8. 추가로 설문을 위한 화면에서는 하단의 [Skip personalization]을 클릭한다.

<https://goddaehee.tistory.com/218>

2 Git을 설치하고 로컬저장소에서 커밋 관리하기

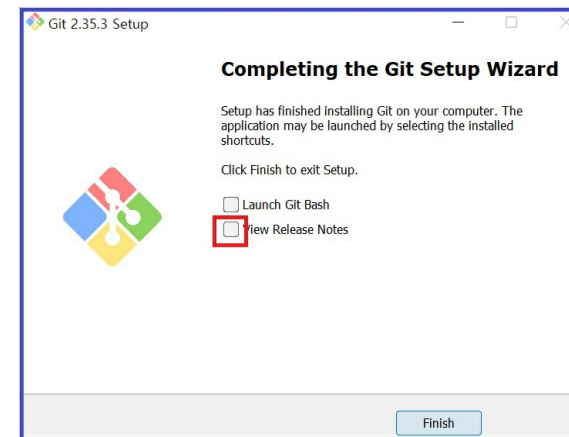
※ 전체 실습 이해하기



2 Git을 설치하고 로컬저장소에서 커밋 관리하기

✱ 내 컴퓨터에 Git 설치하기

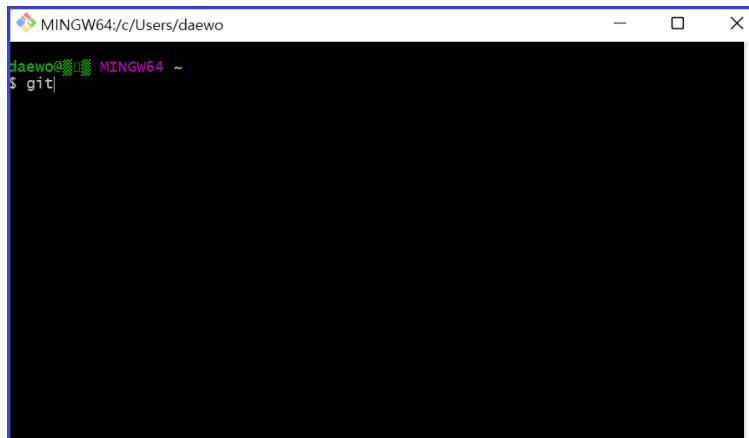
1. 구글 검색창에서 'git 다운로드'를 검색하고 <https://git-scm.com/downloads> 링크로 접속한다.
2. 다운로드 페이지에 접속해서 Downloads 항목에서 본인의 운영체제에 맞는 최신버전 링크를 클릭한다.
3. 다운받은 Git 설치 파일을 실행하면 설치창이 뜬다. 기본 설정을 유지한 채 계속 [Next]를 누르고 마지막 [Install] 클릭한다.
4. 마지막 화면에서 [View Release Notes]를 체크하지 않고 [Finish]버튼을 클릭한다.



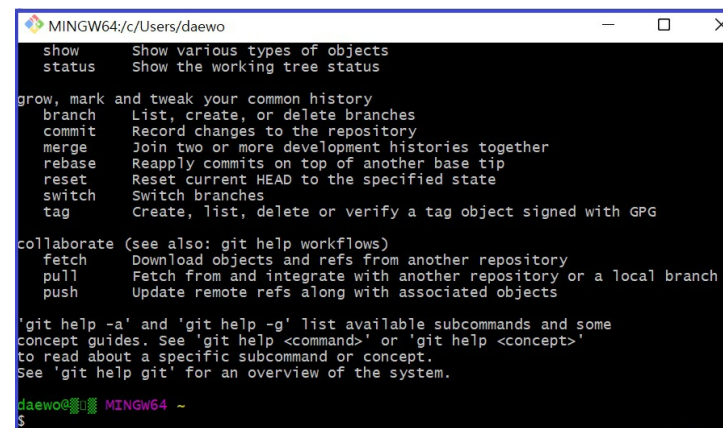
2 Git을 설치하고 로컬저장소에서 커밋 관리하기

✳ 내 컴퓨터에 Git 설치하기

5. 잘 설치되었나 확인하려면 윈도우 시작 버튼 옆에 있는 돋보기 아이콘을 클릭해서 'git bash'를 입력하고, 찾아서 실행한다.
6. 검점화면이 뜨면 \$ 표시 옆에 'git'이라고 입력 후 [Enter]를 누른다. 깃 기본 명령어 대한 안내가 나오면 제대로 설치된 것이다.



```
MINGW64:/c/Users/daewo
daewo@MINGW64 ~
$ git
```



```
MINGW64:/c/Users/daewo
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

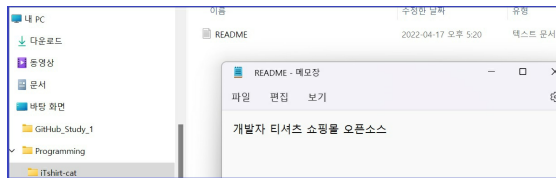
'git help -a' and 'git help -g' list available subcommands and some
concept guides. See 'git help <command>' or 'git help <concept>'
to read about a specific subcommand or concept.
See 'git help git' for an overview of the system.

daewo@MINGW64 ~
$
```

② Git을 설치하고 로컬저장소에서 커밋 관리하기

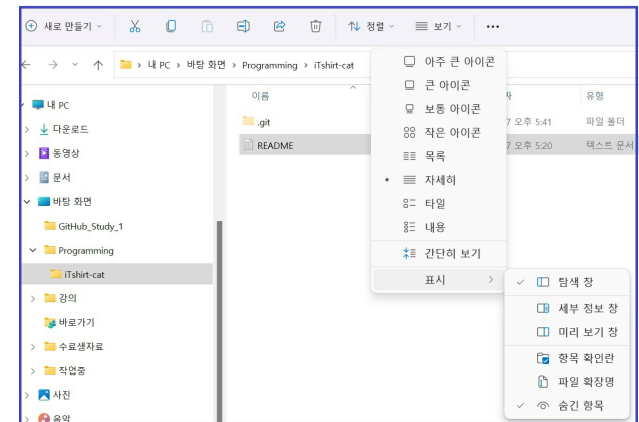
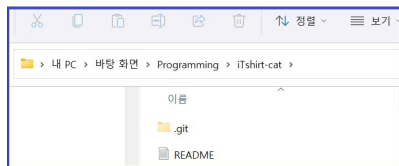
✱ 로컬저장소 만들기

1. 내 컴퓨터/ 바탕화면/ Programming/ iTshirt-cat 폴더를 만든다.
2. 폴더 [iTshirt-cat] 안에 프로젝트에 대한 설명이 담긴 텍스트 문서 README.txt를 하나 만든다.
3. 텍스트 파일을 열고 '개발자 티셔츠 쇼핑몰 오픈소스'라고 기록하고 저장한다.



4. 폴더 [iTshirt-cat]에서 마우스 오른쪽 버튼을 클릭하고 [Git Bash Here]를 클릭한다.
5. Git Bash 창이 열리면 `$ git init` 명령어를 입력한다.
6. 'Initialized empty Git repository'라는 문구가 나오면 성공이다. 폴더 [iTshirt-cat]에는 로컬저장소 [.git]폴더가 자동 생성된다.

만약 [.git]폴더가 보이지 않으면 폴더 [iTshirt-cat] README 파일 클릭 후 상단메뉴에서 [보기]를 선택하고 [숨긴항목]을 체크해 주면 된다.



Git을 설치하고 로컬저장소에서 커밋 관리하기

✧ 첫 번째 커밋 만들기

1. 방금 생성했던 README.txt 파일을 하나의 버전으로 만들어 보자. Git에서는 이렇게 생성된 각 버전을 커밋(Commit)이라고 한다.

2. 먼저 버전 관리를 위해 내 정보를 등록해야 한다. 각 버전을 누가 만들었는지 알아야 협업하기 편하다.

폴더 [iTshirt-cat]에서 마우스 오른쪽 버튼을 클릭하고 [Git Bash Here]를 클릭해서 Bash 창을 연다.

```
$ git config --global user.email "자신의 것"
```

```
$ git config --global user.name "자신의 것"
```

3. 다음으로 커밋에 추가할 파일을 선택한다.

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

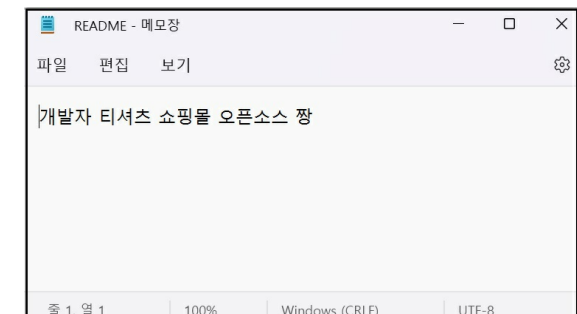
4. 커밋에는 상세 설명을 적을 수 있다. 잘 적어 넣은 설명은 파일 생성 이유와 수정 이유를 알 수 있고, 해당 버전을 찾아 시간여행을 하기도 수월하다. -m은 message의 약자이다.

```
$ git commit -m "사이트 설명 추가"
```

5. 두 번째 커밋을 해보자. 먼저 README.txt 파일을 열어 "짱" 문자를 추가하고 저장한다.

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

```
$ git commit -m "설명 업데이트"
```



② Git을 설치하고 로컬저장소에서 커밋 관리하기

★ 다른 으로 시간 여행하기

1. 시간여행, 개발하다가 요구사항이 변경되어 이전 커밋부터 다시 개발하고 싶다면 Git을 사용해 그 커밋으로 돌아가면 된다.
2. 먼저 log 명령어로 지금까지 만든 커밋을 확인한다.

\$ git log

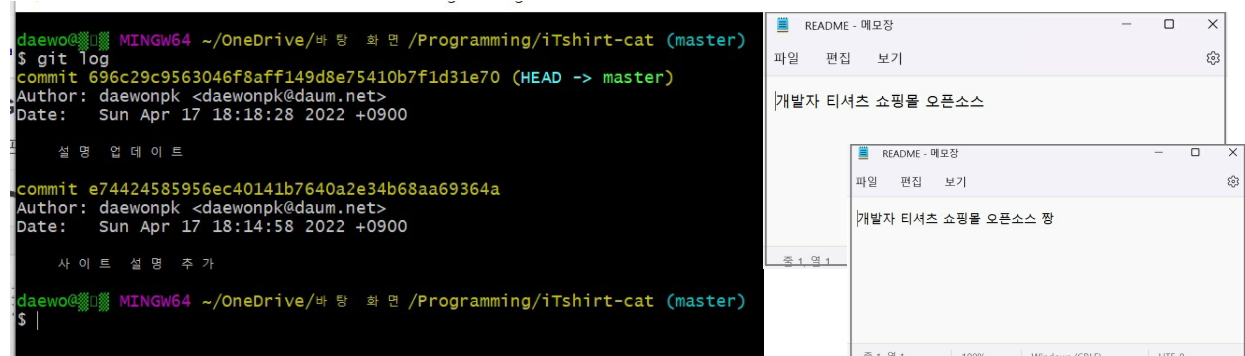
3. 두 개의 커밋을 확인하고 아래 첫 번째 커밋의 앞 7자리(전체를 복사해도 된다) 커밋 아이디를 복사하고 checkout 명령어로 해당 커밋으로 되돌린다.

\$ git checkout e744245(자신의 첫 번째 커밋의 아이디)

4. README.txt 파일을 열어보자. "짱"이 사라진 첫 번째 커밋으로 돌아간 것을 확인할 수 있다.
5. 다시 체크아웃을 해서 최신 커밋인 두 번째 커밋으로 돌아가 보자.

\$ git checkout - (혹은 두 번째 커밋 아이디)

\$ git commit -m "설명 업데이트"



3 GitHub 원격저장소에 커밋 올리기

* 원격저장소 만들기

1. 깃허브에서는 원격저장소를 레포지토리라고 부른다.
2. GitHub.com에 접속하고 로그인한다. 상단 네비게이션 바에서 오른쪽에 있는 [+] 아이콘을 클릭 후 [New repository] 메뉴 선택.
3. 어떤 repository를 만들지 세부 항목을 작성하는 페이지가 나오면 이름과 설명에 아래처럼 정보를 입력하고 [Create repository]

[Repository name] iTshirt

[Description] IT인을 위한 티셔츠 쇼핑몰 오픈소스

4. 다른 개발자와 함께 작업할 때는 원격저장소 주소를 알려주면 된다.
5. 주소를 복사해서 브라우저에서 `https://github.com/daewonpk/iTshirt.git`

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](#).

Create a new repository

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

Owner *

Repository name *

daewonpk

iTshirt

Great repository names are short and memorable. Need inspiration? How about [furry-octo-bassoon?](#)

Description (optional)

IT인을 위한 티셔츠 쇼핑몰 오픈소스

☒ 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.](#)

.gitignore template: None

Choose a license

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

License: None

You are creating a public repository in your personal account.

Create repository

3 GitHub 원격저장소에 커밋 올리기

* 원격저장소에 커밋 올리기

1. 깃허브에서 만든 원격저장소 주소를 내 컴퓨터의 [iTshirt-cat] 로컬저장소에 알려주고, 로컬저장소에서 만들었던 커밋들을 원격저장소에 올려 보자.

2. [iTshirt-cat] 폴더의 Git Bash로 들어온다. `remote add origin` 명령어는 로컬 저장소에 원격저장소 주소를 알려준다.

```
$ git remote add origin https://github.com/daewonpk/iTshirt.git
```

3. 로컬저장소에 있는 커밋들을 `push` 명령어로 원격저장소에 올려보자.

```
$ git push origin master
```

4. 원격저장소에서 제대로 `push` 됐나 확인해 보자.

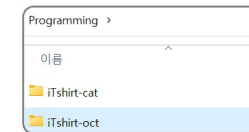


4 GitHub 원격저장소의 커밋을 로컬저장소에 내려받기

★ 원격저장소의 커밋을 로컬저장소에 내려받기

1. 원격저장소의 코드와 버전 전체를 내 컴퓨터로 내려받는 것을 클론이라고 한다. 클론하면 최신 버전뿐만 아니라 이전 버전들과 원격저장소 주소 등이 내 컴퓨터의 로컬저장소에 저장된다.

2. 먼저 아래 경로로 내 컴퓨터에 [iTshirt-oct] 폴더를 만들어 준다. 이 폴더를 문어 로컬저장소로 지정한다. 그리고 고양이와 원격저장소로 올렸던 커밋을 이곳으로 내려받을 것이다.



3. [iTshirt-oct] 폴더에서 마우스 오른쪽 버튼 누르고 [Git Bash Here]를 클릭.

4. 원격저장소 주소를 복사해서 아래 명령어에 붙인다. 원격저장소 주소 맨뒤에서 한 칸 띄우고 마침표(.)를 찍는다.

```
$ git clone https://github.com/daewonpk/iTshirt.git .
```

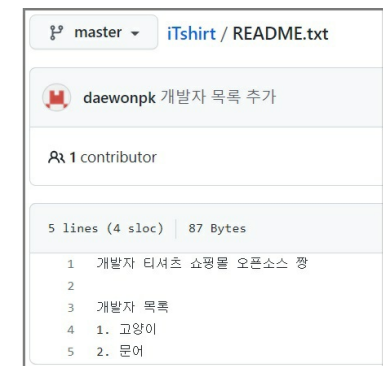
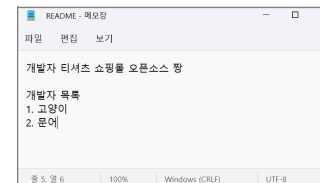
4. [iTshirt-oct] 폴더에 있는 README.txt 파일을 열어보면 원격저장소의 동일한 내용을 확인할 수 있다. 내용을 수정하고 저장한다.

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

```
$ git commit -m "개발자 목록 추가"
```

```
$ git push origin master
```

5. 원격저장소에서 확인해 보자.



GitHub 원격저장소의 커밋을 로컬저장소에 내려받기

✱ 원격저장소의 새로운 커밋을 로컬저장소에 갱신하기

1. [iTshirt-cat] 폴더 README.txt 파일을 열어보면 세 번째 커밋이 반영되지 않은 상태 그대로 인 것을 알 수 있다.
2. [iTshirt-oct] 폴더에서 원격저장소에 올렸던 새로운 커밋을 [iTshirt-cat] 폴더 로컬저장소에 내려받아서 현재 상태를 갱신해
보자. [iTshirt-cat] 폴더에서 마우스 오른쪽 버튼을 클릭한 다음 [Git Bash Here]를 선택한다.
3. 아래 명령을 입력하자. **pull**은 원격저장소에 **새로운 커밋이 있다면** 그걸 내 로컬저장소에 받아 오라는 명령이다.

```
$ git pull origin master
```

4. [iTshirt-cat] 폴더 README.txt 파일을 다시 열어보면 갱신된 것을 알 수 있다.

. 깃은 **CLI**(Command Line Interface) 즉 명령어를 하나씩 입력하는 방식과 **GUI**(Graphic User Interface) 즉 마우스로 클릭해서 사용하는 방식 2가지가 있다. 서버에 접속하거나 특정 상황에서 Git은 CLI만 지원한다.