

Git & Github Python Tutorial

Practice 1

Git?

- 리누스 토르발스가 리눅스 커널의 버전 관리를 위해 BitKeeper를 쓰다가 유료로 전환되는 바람에 직접 개발한 분산형 버전 관리 시스템(VCS).
- 매우 빠른 속도와 분산형 저장소 지원이 특징이다. Git 자체는 오픈 소스이며 저장소는 <https://github.com/git/git>이다.

✓ PICK ①

마이크로소프트, 깃허브 인수한다

기사입력 2018-06-04 08:56 최종수정 2018-06-04 14:59 [기사원문](#) [스크랩](#)  본문듣기 · 설정

 5  2

[요약본](#) [가](#) [인](#) [공](#)

(지디넷코리아=이정현 기자)마이크로소프트(MS)가 소스코드 공유사이트 깃허브(Github)를 인수하는 데 합의했다고 블룸버그통신이 3일(현지 시간) 보도했다.

이번 인수가 어느 정도 가격에 성사됐는지는 알려지지 않았다. 깃허브는 2015년 20억 달러에 기업가치를 평가 받은 적 있다. 블룸버그는 이르면 4일 인수 협상이 발표될 수 있다고 보도했다.

2008년에 설립된 깃허브는 현재 7800만 명의 소프트웨어 개발자가 8000만개의 소스코드를 보유하고 있는 세계 최대의 오픈소스 소프트웨어 플랫폼이다.

하지만 깃허브는 아직 수익은 내지 못하고 있다. 또 최근 9개월 동안 새로운 최고경영자(CEO) 영입 작업을 추진해 왔지만 뜻을 이루지 못했다.

기본 용어

■ 커맨드 라인(Command Line)

- 깃 명령어를 입력할 때 사용하는 컴퓨터 프로그램. 맥에선 터미널이라고 한다. PC에선 기본적인 프로그램이 아니어서 처음엔 깃을 다운로드해야 한다. 두 경우 모두 마우스를 사용하는 것이 아닌 프롬프트로 알려진 텍스트 기반 명령어를 입력한다.

■ 저장소(Repository)

- 프로젝트가 거주(live)할 수 있는 디렉토리나 저장 공간. 깃허브 사용자는 종종 “repo”로 줄여서 사용한다. 당신의 컴퓨터 안의 로컬 폴더가 될 수도 있고, 깃허브나 다른 온라인 호스트의 저장 공간이 될 수도 있다. 저장소 안에 코드 화일, 텍스트 화일, 이미지 화일을 저장하고, 이름붙일 수 있다.

■ 버전관리(Version Control)

- 기본적으로, 깃이 서비스되도록 고안된 목적. MS 워드 작업할 때, 저장하면 이전 화일 위에 겹쳐쓰거나 여러 버전으로 나누어 저장한다. 깃을 사용하면 그럴 필요가 없다. 프로젝트 히스토리의 모든 시점의 “스냅샷”을 유지하므로, 결코 잃어버리거나 겹쳐쓰지 않을 수 있다.

■ 커밋(Commit)

- 깃에게 파워를 주는 명령이다. 커밋하면, 그 시점의 당신의 저장소의 “스냅샷”을 찍어, 프로젝트를 이전의 어떠한 상태로든 재평가하거나 복원할 수 있는 체크포인트를 가질 수 있다.

■ 브랜치(Branch)

- 여러 명이 하나의 프로젝트에서 깃 없이 작업하는 것이 얼마나 혼란스러울 것인가? 일반적으로, 작업자들은 메인 프로젝트의 브랜치를 따와서(branch off), 자신이 변경하고 싶은 자신만의 버전을 만든다. 작업을 끝낸 후, 프로젝트의 메인 디렉토리인 “master”에 브랜치를 다시 “Merge”한다.

주요 명령어

■ git init

- 깃 저장소를 초기화한다. 저장소나 디렉토리 안에서 이 명령을 실행하기 전까지는 그냥 일반 폴더이다. 이것을 입력한 후에야 추가적인 깃 명령어들을 줄 수 있다.

■ git add

- 이 명령이 저장소에 새 파일들을 추가하진 않는다. 대신, 깃이 새 파일들을 지켜보게 한다. 파일을 추가하면, 깃의 저장소 “스냅샷”에 포함된다.

■ git commit

- 깃의 가장 중요한 명령어. 어떤 변경사항이라도 만든 후, 저장소의 “스냅샷”을 찍기 위해 이것을 입력한다. 보통 “git commit -m “Message hear.” 형식으로 사용한다. -m은 명령어의 그 다음 부분을 메시지로 읽어야 한다는 것을 말한다.

■ git branch

- 여러 협업자와 작업하고 자신만의 변경을 원한다? 이 명령어는 새로운 브랜치를 만들고, 자신만의 변경사항과 파일 추가 등의 커밋 타임라인을 만든다. 당신의 제목이 명령어 다음에 온다. 새 브랜치를 “cats”로 부르고 싶으면, git branch cats를 타이핑한다.

■ git push

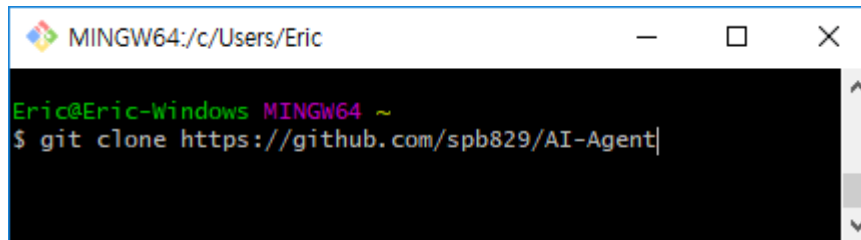
- 로컬 컴퓨터에서 작업하고 당신의 커밋을 깃허브에서 온라인으로도 볼 수 있기를 원한다면, 이 명령어로 깃허브에 변경사항을 “push”한다.

■ git pull

- 로컬 컴퓨터에서 작업할 때, 작업하고 있는 저장소의 최신 버전을 원하면, 이 명령어로 깃허브로부터 변경사항을 다운로드한다(“pull”).

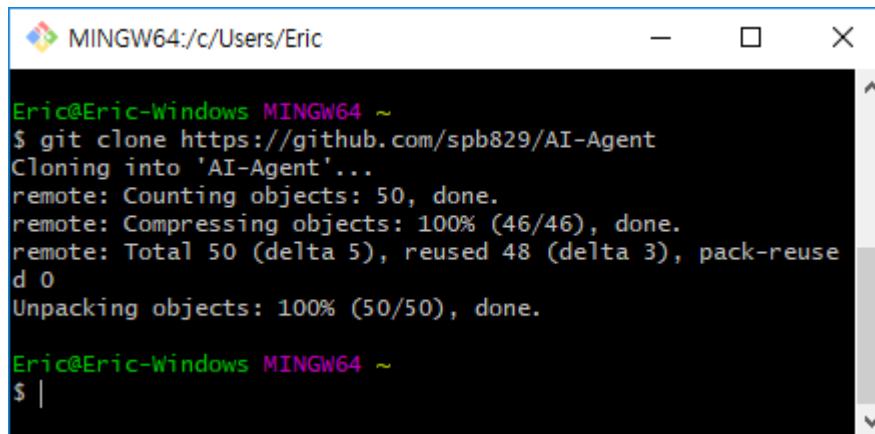
git clone

- Git 저장소를 복사하고 싶을 때 사용



```
MINGW64:/c/Users/Eric
Eric@Eric-Windows MINGW64 ~
$ git clone https://github.com/spb829/AI-Agent|
```

- git clone https://github.com/spb829/AI-Agent



```
MINGW64:/c/Users/Eric
Eric@Eric-Windows MINGW64 ~
$ git clone https://github.com/spb829/AI-Agent
Cloning into 'AI-Agent'...
remote: Counting objects: 50, done.
remote: Compressing objects: 100% (46/46), done.
remote: Total 50 (delta 5), reused 48 (delta 3), pack-reuse
d 0
Unpacking objects: 100% (50/50), done.

Eric@Eric-Windows MINGW64 ~
$ |
```

- cd AI-Agent
- ls

git clone

```
MINGW64:/c/Users/Eric/AI-Agent
Eric@Eric-Windows MINGW64 ~
$ cd AI-Agent/

Eric@Eric-Windows MINGW64 ~/AI-Agent (master)
$ ls
README.md

Eric@Eric-Windows MINGW64 ~/AI-Agent (master)
$
```

- git checkout practice1
- ls

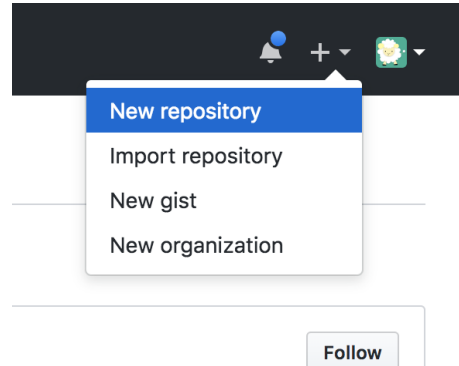
```
MINGW64:/c/Users/Eric/AI-Agent
Eric@Eric-Windows MINGW64 ~/AI-Agent (master)
$ git checkout practice1
Switched to a new branch 'practice1'
Branch 'practice1' set up to track remote branch 'practice1'
from 'origin'.

Eric@Eric-Windows MINGW64 ~/AI-Agent (practice1)
$ ls
Git/  Jupyter/  Python/  README.md

Eric@Eric-Windows MINGW64 ~/AI-Agent (practice1)
$
```

첫 Repository 만들기

- Github 가입 & 로그인
- New Repository 클릭




- 새로운 Repository의 제목을 설정 (ex. AI-Agent)

Create a new repository

A repository contains all the files for your project, including the revision history.

Owner

 spb829 ▾

Repository name

AI-Agent



Great repository names are short and memorable. Need inspiration? How about **automatic-eureka**.

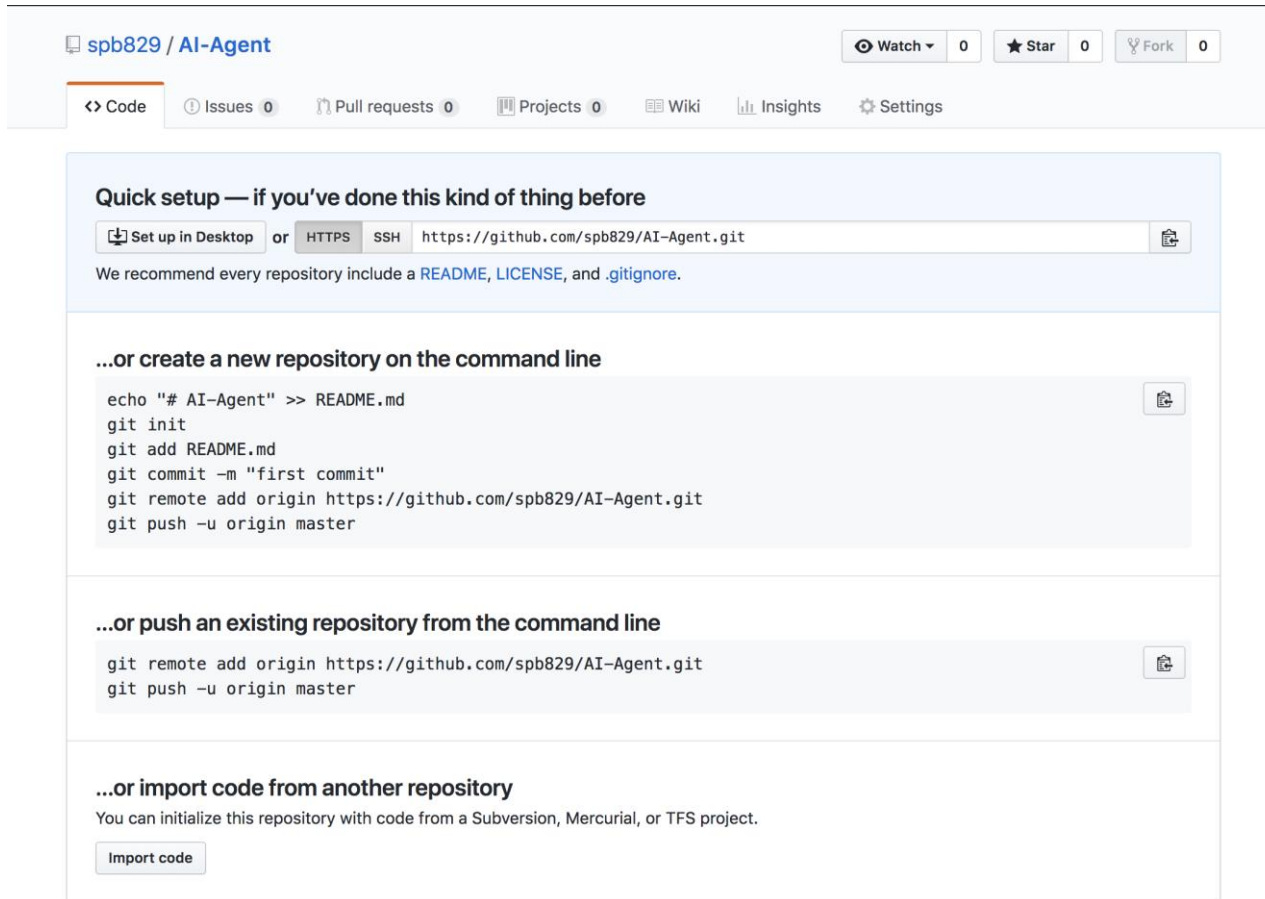
Description (optional)

☒  **Public**

Anyone can see this repository. You choose who can commit.

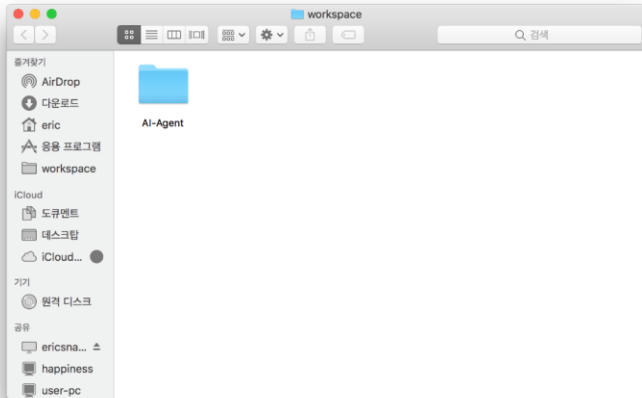
첫 Repository 만들기

- 다음과 같은 화면이 나온다면 Repository 생성완료



첫 Repository 만들기

- 이 Repository를 위한 폴더를 생성



- 터미널 실행 후, 해당 경로로 진입

```
Eric's-MacBook-Pro:workspace eric$ cd workspace/  
Eric's-MacBook-Pro:workspace eric$ ls  
AI-Agent  
Eric's-MacBook-Pro:workspace eric$ cd AI-Agent/  
Eric's-MacBook-Pro:AI-Agent eric$
```

첫 Repository 만들기

- 명령어 입력
- `git init`
- `git remote add origin https://github.com/spb829/AI-Agent.git`

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

☒ Set up in Desktop or ☐ HTTPS ☐ SSH

We recommend every repository include a [README](#), [LICENSE](#), and [.gitignore](#).

- 본인의 Repository 주소를 입력
- `echo "# AI-Agent" >> README.md`
- `git add README.md`
- `git commit -m "First commit"`
- `git push -u origin master`
- 아이디와 비밀번호를 물어볼 경우, 입력

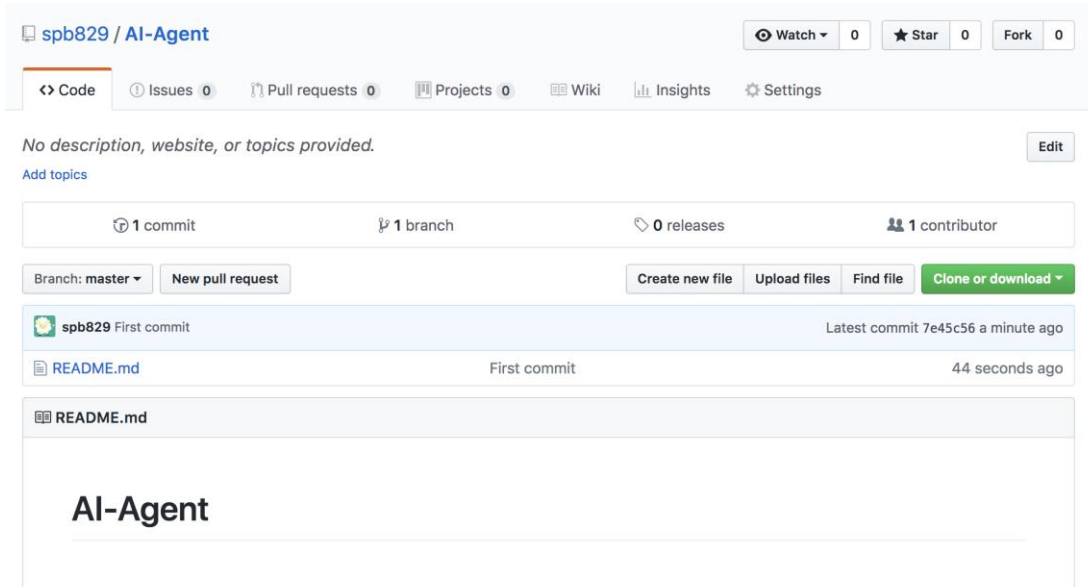
첫 Repository 만들기

■ 결과화면

```
Eric-MacBook-Pro:AI-Agent eric$ git init
Initialized empty Git repository in /Users/eric/workspace/workspace/AI-Agent/.git/
Eric-MacBook-Pro:AI-Agent eric$ git remote add origin https://github.com/spb829/AI-Agent.git
Eric-MacBook-Pro:AI-Agent eric$ echo "# AI-Agent" >> README.md
Eric-MacBook-Pro:AI-Agent eric$ git add README.md
Eric-MacBook-Pro:AI-Agent eric$ git commit -m "First commit"
[master (root-commit) 7e45c56] First commit
1 file changed, 1 insertion(+)
create mode 100644 README.md
Eric-MacBook-Pro:AI-Agent eric$ git push -u origin master
Username for 'https://github.com': spb829
Password for 'https://spb829@github.com':
Counting objects: 3, done.
Writing objects: 100% (3/3), 222 bytes | 222.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0)
To https://github.com/spb829/AI-Agent.git
* [new branch]      master -> master
Branch 'master' set up to track remote branch 'master' from 'origin'.
Eric-MacBook-Pro:AI-Agent eric$
```

■ Push가 완료되었다면 해당 Repository 페이지를 확인

첫 Repository 만들기



- 위와 같은 화면을 확인했다면 성공
- 명령어 요약
 - `git init`
 - `git remote add origin [본인의 Github Repository 주소]`
 - `echo "# AI-Agent" >> README.md`
 - `git commit -m "First commit"`
 - `git push -u origin master`

Python Tutorial

- Python 개발환경 설정

- Anaconda
- Jupyter notebook

- Basic Python

- Load package
- For, if/else
- Function
- List
- Dictionary
- Class