

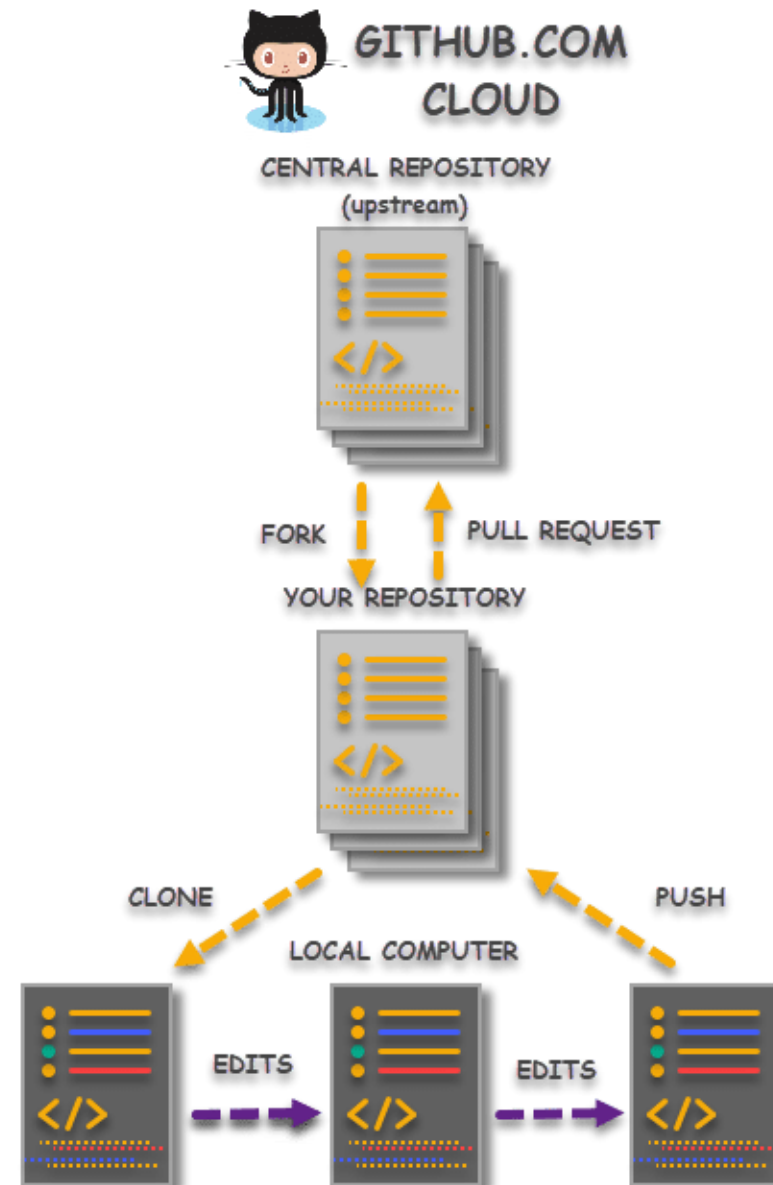
지역과 원격저장소 연동

Version Control and
Git & Github basics

강환수 교수



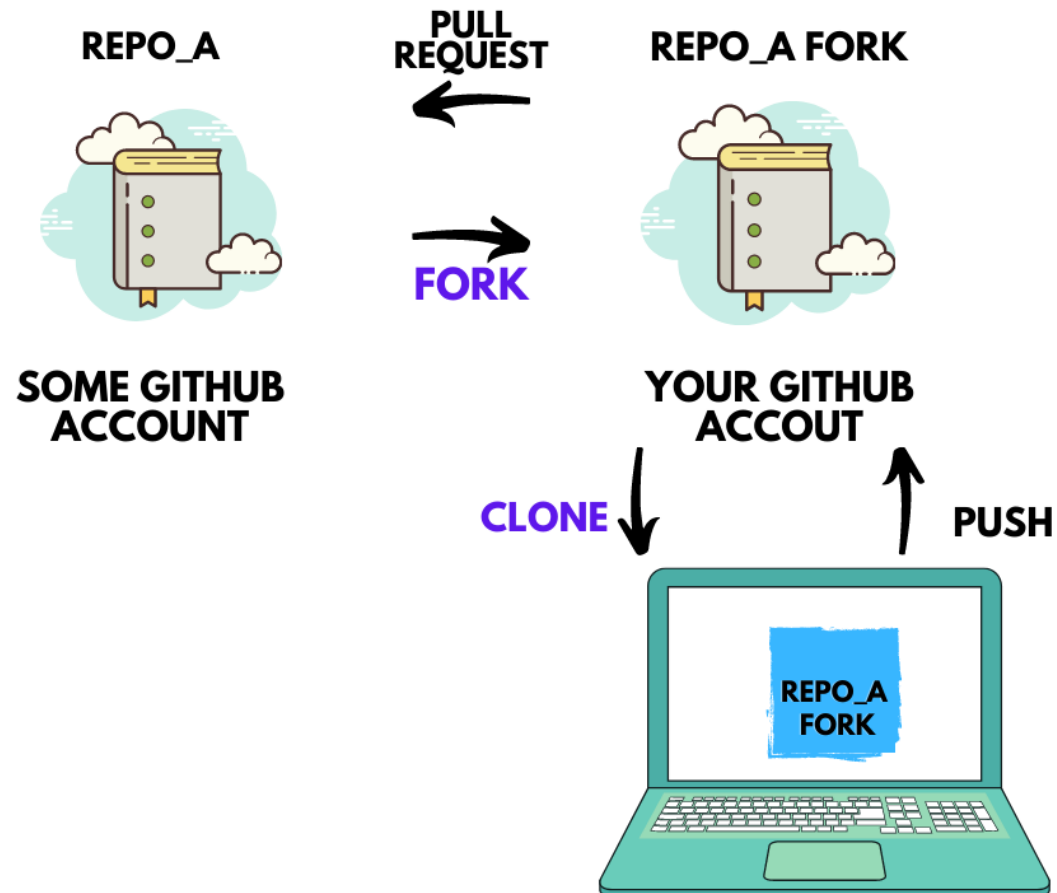
- **Fork**
 - 타인의 깃허브 원격저장소를 자신의 깃허브에 복제하는 포크(fork)
- **Pull request**
 - (포크된) 타인의 깃허브의 내용 수정(파일 추가나 수정)을 요청하는 끌기요청[PR(pull request)]의 이해
- **Clone**
 - 깃허브 원격저장소를 지역저장소로 복제(clone)해 지역과 원격을 연결
- **Pull**
 - 이후 깃허브 원격저장소의 수정 내용을 다시 지역저장소로 끌어오기(pull)
- **Push**
 - 이후 지역저장소의 수정 내용을 다시 원격저장소로 올리기(push)



AI Experts
Who Lead
The Future

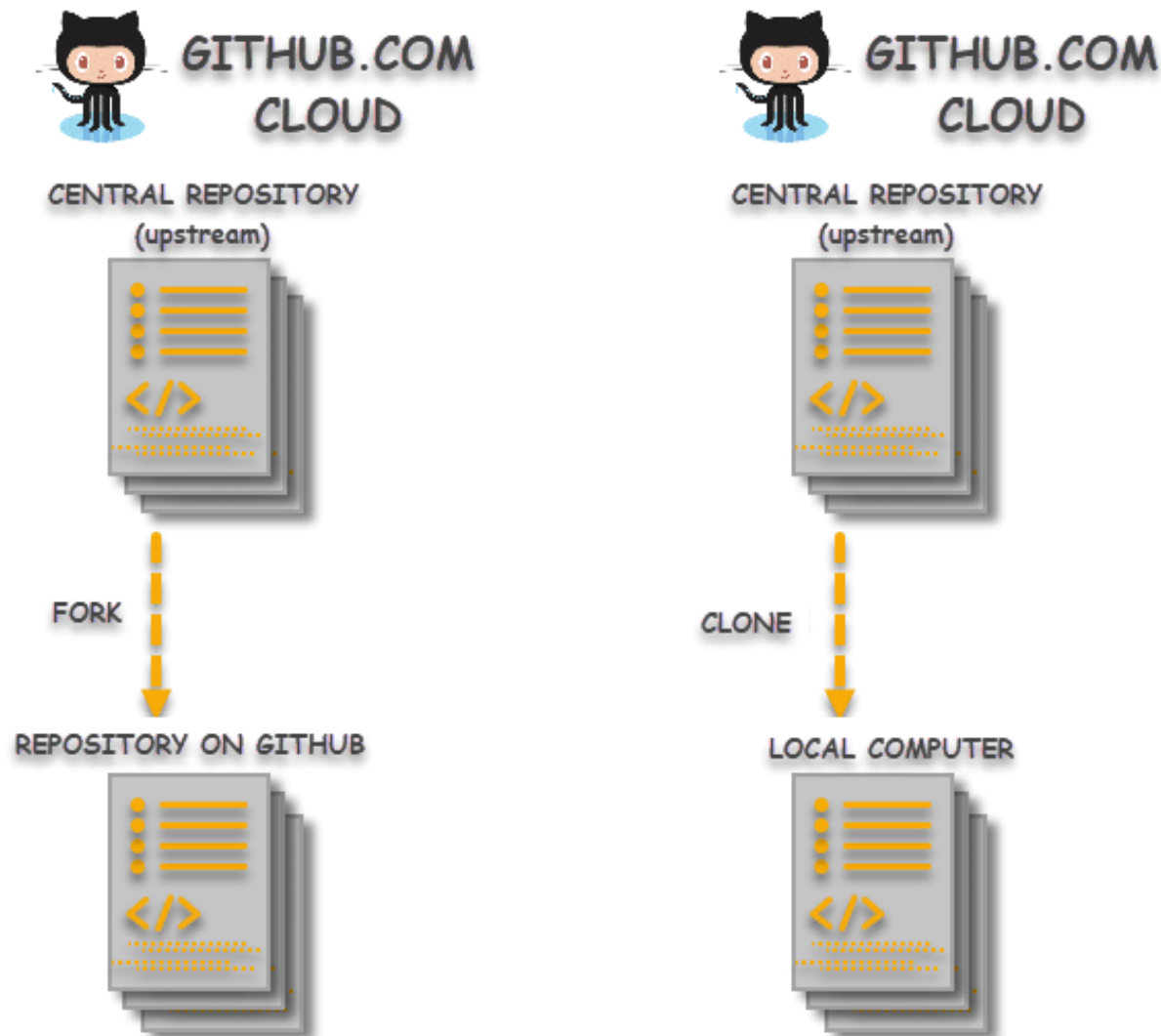
01

지역에서 깃허브 저장소 클론(clone)



 **CHEDY HAMMAMI**

- 원격 저장소를 지역 저장소에 복제
 - 공개된 저장소는 소유와 관계 없이 가능



- 1. 깃허브 원격저장소 생성
 - git-clone
 - https 주소 복사
- 2. PC 깃에서 원격저장소 복제(git clone)
 - \$ git clone [복사된-주소]
 - 원격저장소와 동일한 이름으로 복제
 - \$ git clone [복사된-주소] [새로운-폴더명]
 - 하부 폴더 [새로운-폴더명] 이름으로 복제
- 3. 원격 저장소 별칭 이름 점검
 - \$ git remote
 - 원격 저장소 목록
 - 기본 이름 origin 점검
 - 위에서 clone을 했기 때문에 별칭 origin이 위의 주소를 정해짐
 - \$ git remote -v
 - 자세한 원격 저장소 목록

- 이름 git-clone
- Add a README file 체크


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 *

Great repository names git-clone is available. orable. Need inspiration? How about [miniature-sniffle?](#)

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.](#)

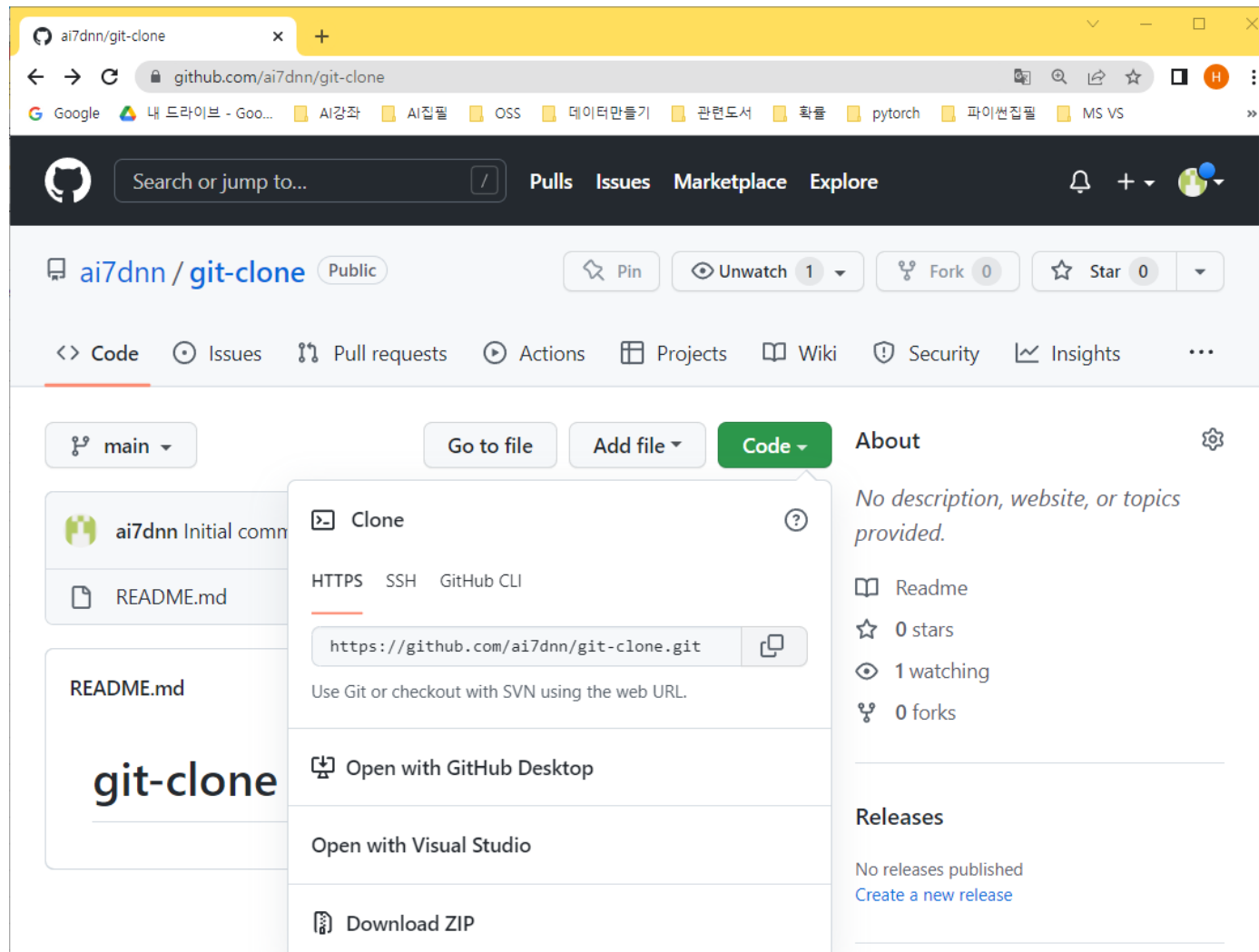
Add .gitignore
Choose which files not to track from a list of templates. [Learn more.](#)
.gitignore template:

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

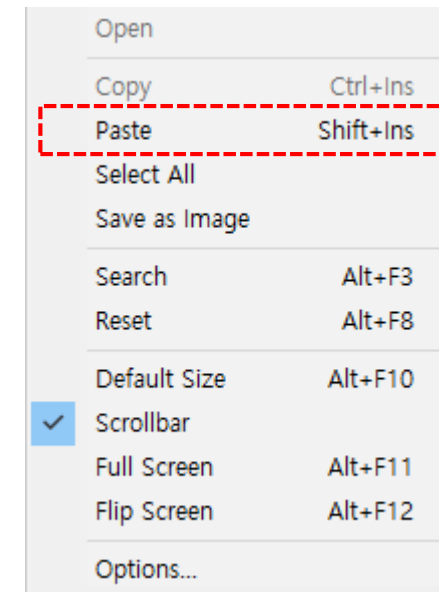
This will set  **main** as the default branch. Change the default name in your [settings](#).

 You are creating a public repository in your personal account.

Create repository



- 명령어
 - \$ git clone 저장소주소 [저장소폴더명]
- 자신의 지역 저장소에 복사 명령
 - \$ git clone https://github.com/ai7dnn/git-clone.git
 - 오른쪽 마우스 팝업 메뉴로 붙여 넣기
- 복사된 폴더 확인



```
PC@DESKTOP-482NOAB MINGW64 /c/[git tutorial]
```

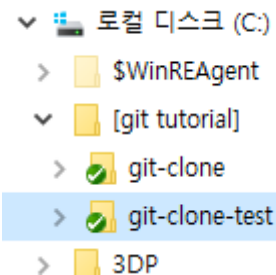
```
$ pwd  
/c/[git tutorial]
```

```
PC@DESKTOP-482NOAB MINGW64 /c/[git tutorial]
```

```
$ git clone https://github.com/ai7dnn/git-clone.git  
Cloning into 'git-clone'...  
remote: Enumerating objects: 3, done.  
remote: Counting objects: 100% (3/3), done.  
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0  
Receiving objects: 100% (3/3), done.
```

- `git clone https://github.com/ai7dnn/git-clone.git` 하부폴더명

```
PC@DESKTOP-482NOAB MINGW64 /c/[git tutorial]
$ git clone https://github.com/ai7dnn/git-clone.git git-clone-test
Cloning into 'git-clone-test'...
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (3/3), done.
```



- **\$ git remote**
 - 원격 저장서 이름 목록만
- **\$ git remote -v**
 - 원격 저장서 주소와 이름 목록

```
PC@DESKTOP-482NOAB MINGW64 /c/[git tutorial]
$ pwd
/c/[git tutorial]
```

```
PC@DESKTOP-482NOAB MINGW64 /c/[git tutorial]
$ cd git-clone
```

```
PC@DESKTOP-482NOAB MINGW64 /c/[git tutorial]/git-clone (main)
$ pwd
/c/[git tutorial]/git-clone
```

```
PC@DESKTOP-482NOAB MINGW64 /c/[git tutorial]/git-clone (main)
$ git remote -v
origin https://github.com/ai7dnn/git-clone.git (fetch)
origin https://github.com/ai7dnn/git-clone.git (push)
```

```
PC@DESKTOP-482NOAB MINGW64 /c/[git
tutorial]/git-clone (main)
$ git remote
origin
```

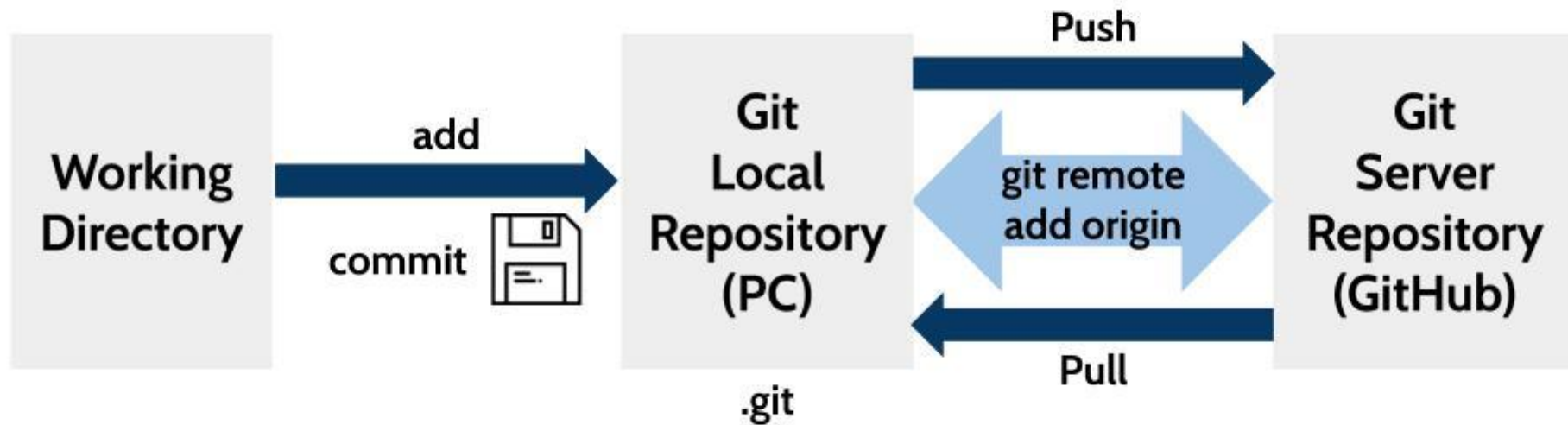
AI Experts
Who Lead
The Future

02

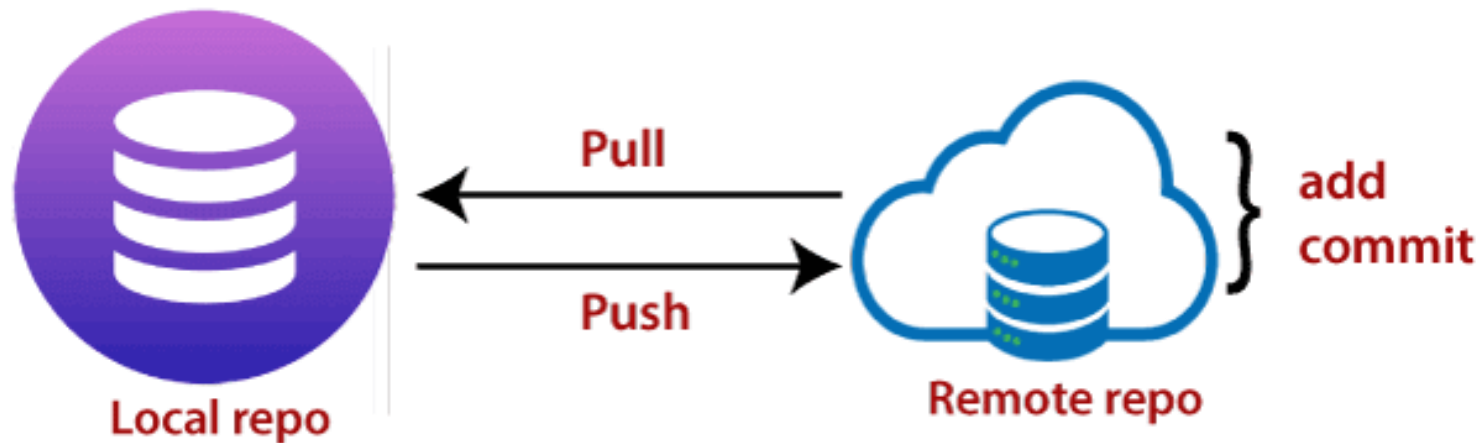
깃허브 저장소 수정 후 풀(pull)



git structure



- 원격저장소에서도 커밋 가능

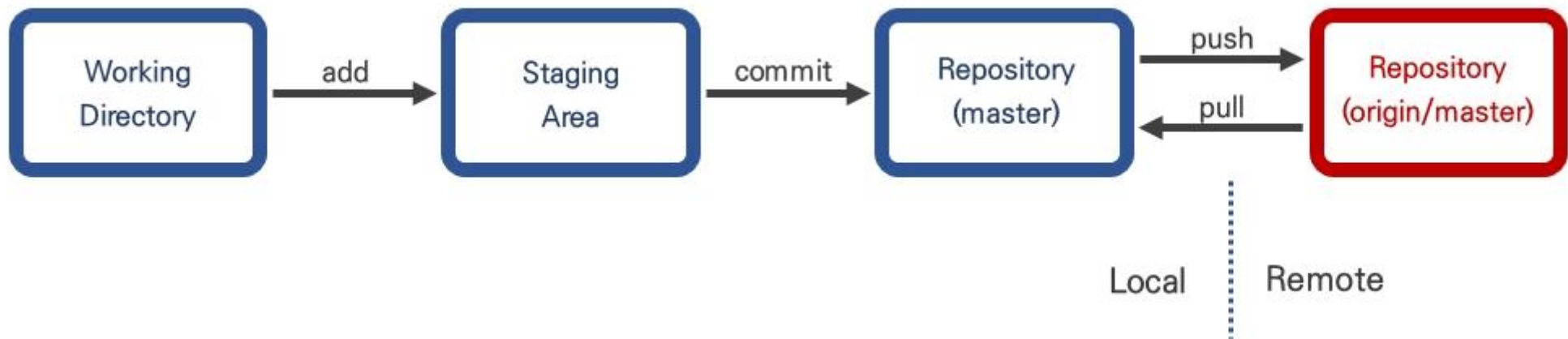


AI Experts
Who Lead
The Future

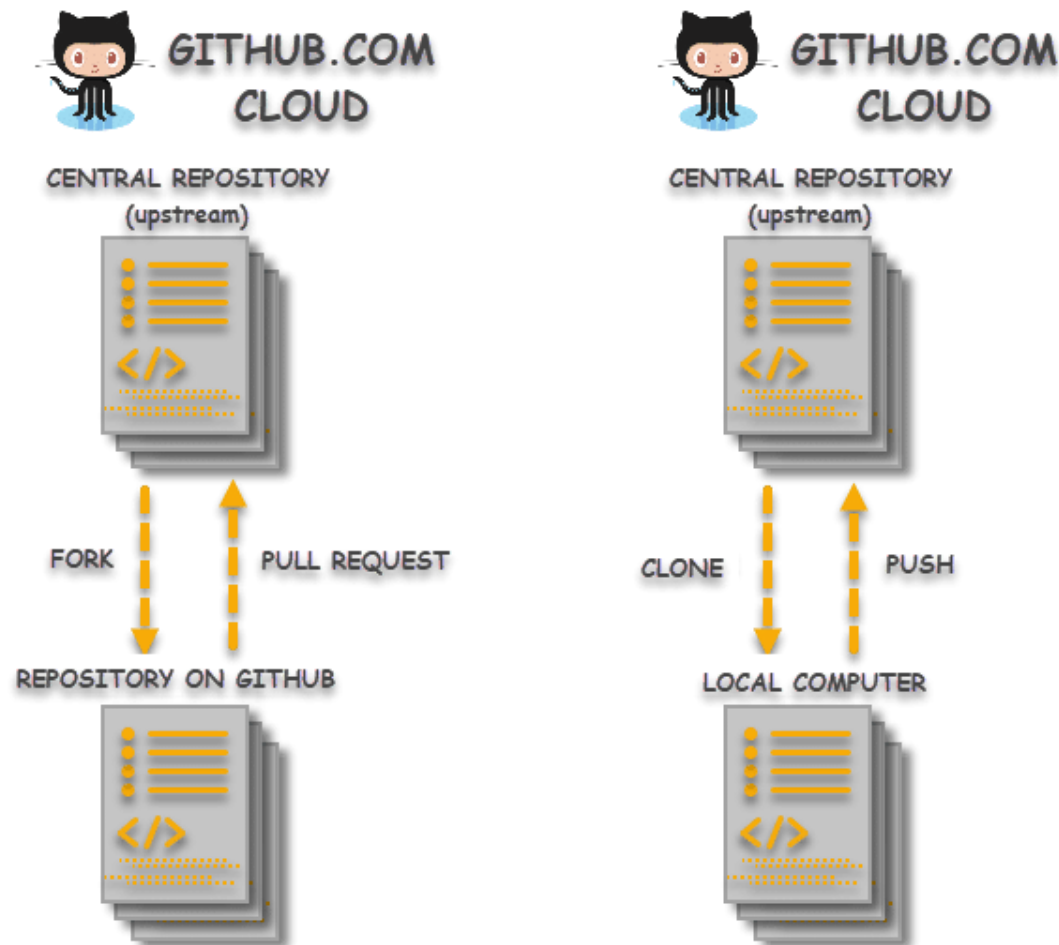
03

지역에서 깃허브 저장소로 푸시(push)

- 쓰기 권한이 있어야 가능
 - 자신의 저장소
 - 협업자로 등록



- PR(pull request), pull 요청, 끌기 요청
 - 포크된 리포지토리에 대한 변경 사항은 PR을 통해 원래 리포지토리와 병합 요청
 - 저장소 소유자를 노크하고 "내가 일부 변경을 수행했습니다. 원하는 경우 이러한 변경 사항을 저장소에 병합(반영)해 주세요."라고 요청하는 것
- 쓰기 권한이 없는 깃허브의 수정 요청을 하고 싶다면 끌기 요청



- PC에서 파일을 만들어 add, commit 후 깃허브 저장소 git-clone에 올리기 push
 - 파일 편집을 vscode로 설정
 - `$ git config core.editor code`
 - `$ code hello.py`
- 1. PC에서 hello.py 파일 생성
 - `$ git add hello.py`
 - `$ git commit -m 'add'`
- 2. PC 지역저장소에서 원격저장소로 보내기 push
 - `$ git push`
 - 오류 발생?
 - `$ git push -u https://{token}@github.com/{username}/{repo_name}.git`
 - `$ git push -u https://ghp_GeU8yVKNmKdIpZ787LHDb6HARqF8h@github.com/lee7py/Python-Programming.git`
- 3. 깃허브에서 확인

- 접근 오류
 - 403 오류

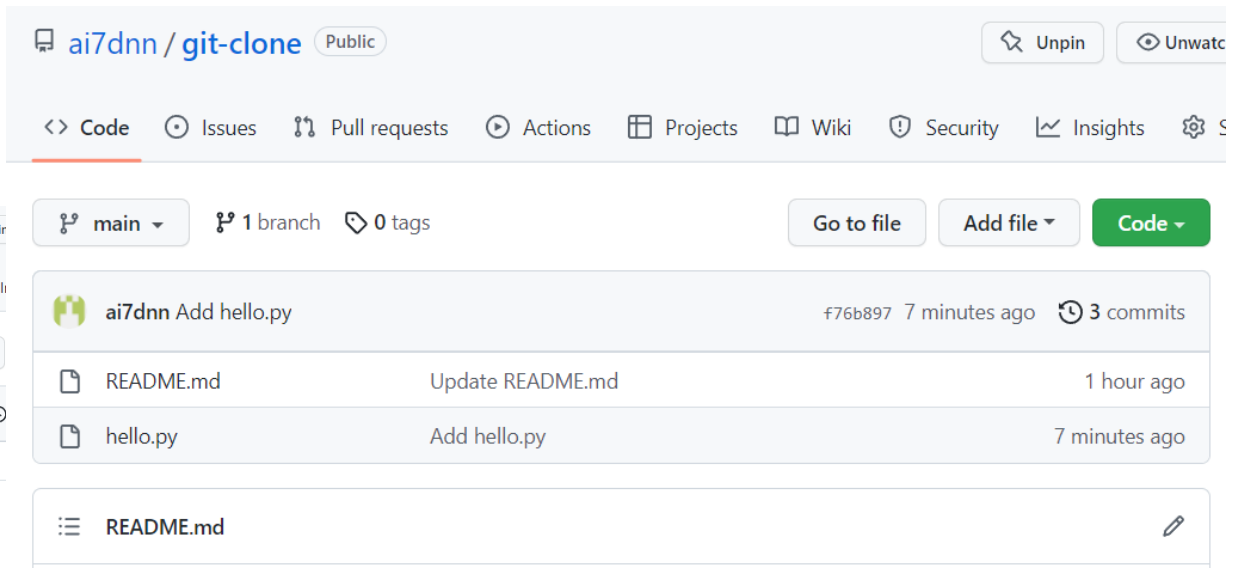
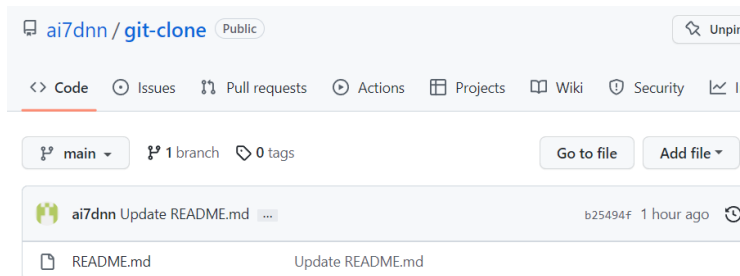
```
PC@DESKTOP-482NOAB MINGW64 /c/[git tutorial]/repo-edu4 (main)
$ git push
remote: Permission to edu4py/repo-edu4.git denied to ai7dnn.
fatal: unable to access 'https://github.com/edu4py/repo-edu4.git/': The requested URL
returned error: 403
```

- \$ git push https://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---@github.com/edu4py/repo-edu4.git
 - \$ git push https://{token}@github.com/{username}/{repo_name}.git

원격 서버로 푸시, 한번 push가 성공한 이후

깃과 깃허브 Python language

- \$ git push [origin] [main]



```
PC@DESKTOP-482NOAB MINGW64 /c/[git tutorial]/git-clone (main)
```

```
$ git push
```

```
Enumerating objects: 4, done.
```

```
Counting objects: 100% (4/4), done.
```

```
Delta compression using up to 8 threads
```

```
Compressing objects: 100% (2/2), done.
```

```
Writing objects: 100% (3/3), 303 bytes | 303.00 KiB/s, done.
```

```
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
```

```
To https://github.com/ai7dnn/git-clone.git
```

```
b25494f..f76b897 main -> main
```

```
PC@DESKTOP-482NOAB MINGW64 /c/[git tutorial]/git-clone (main)
```

```
$
```