

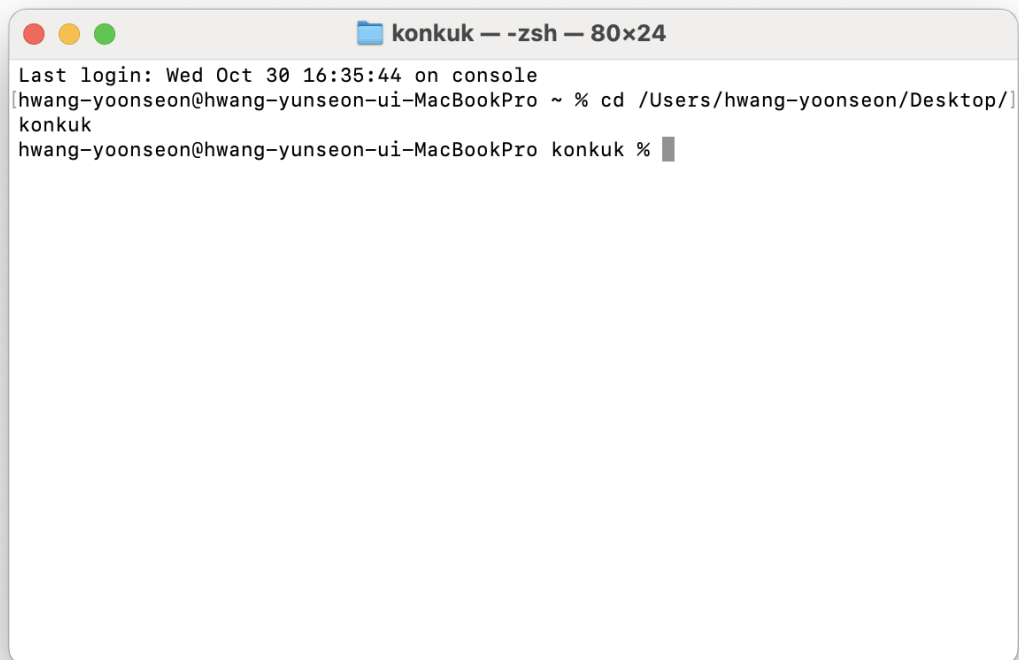
git 기초 사용법

☼ Status	시작 전
☰ Week	

1. 외부저장소에서 불러온 파일을 내 컴퓨터 어디에 저장할래?

: 저장할 폴더로 이동하기

`cd` 폴더경로



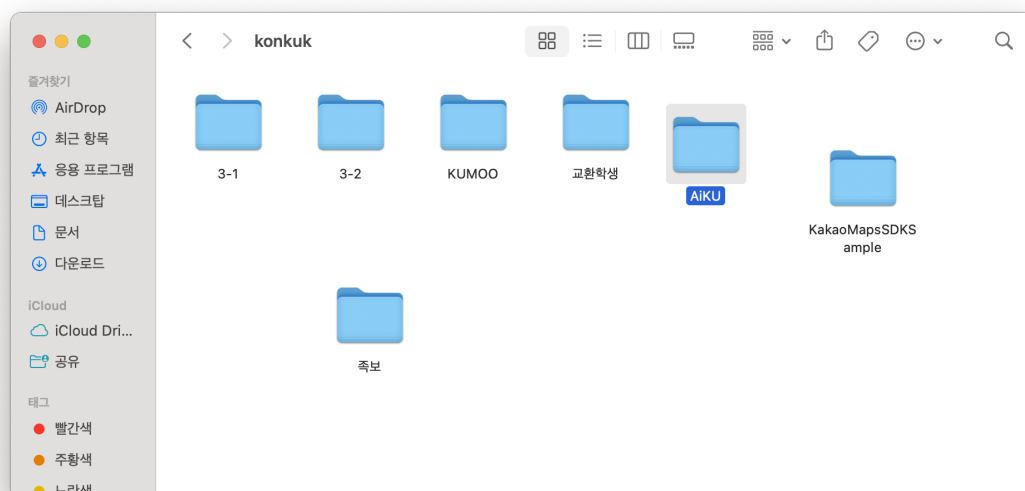
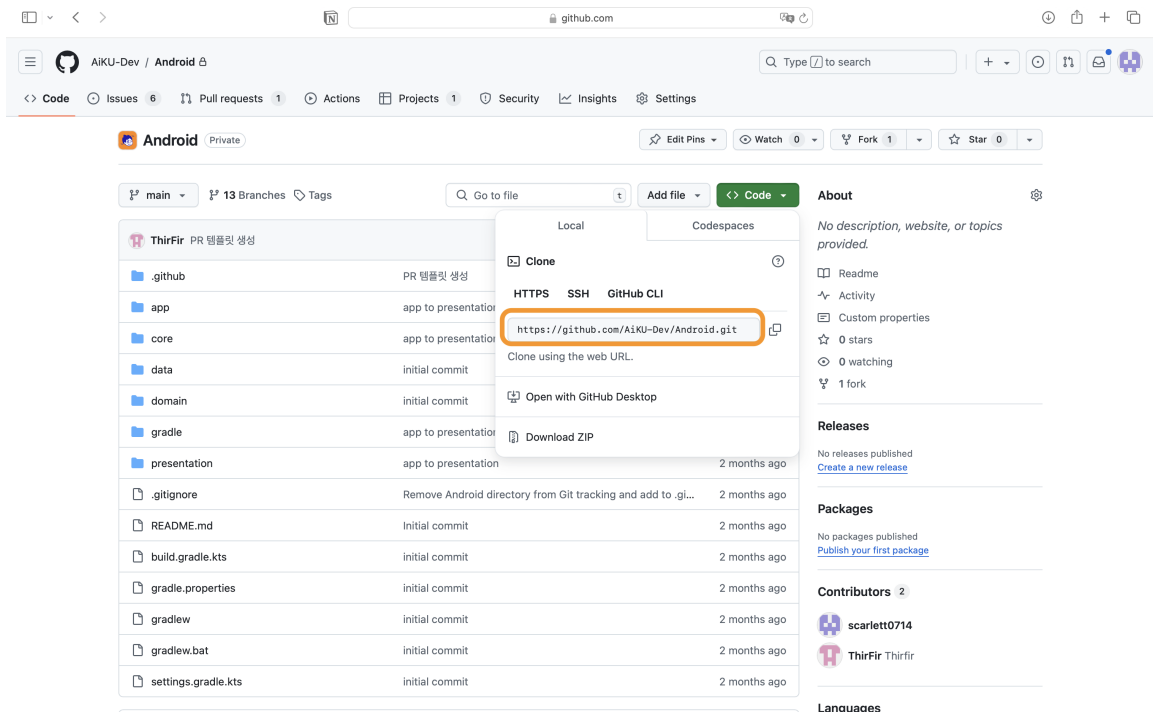
```
konkuk — -zsh — 80x24
Last login: Wed Oct 30 16:35:44 on console
[hwang-yoonseon@hwang-yunseon-ui-MacBookPro ~ % cd /Users/hwang-yoonseon/Desktop/]
konkuk
hwang-yoonseon@hwang-yunseon-ui-MacBookPro konkuk % █
```

그럼 사진처럼 konkuk이라는 폴더로 이동함

2. 외부저장소(Github)에 있는 레포지토리 로컬(내컴퓨터)로 불러오기

: 이걸 맨 처음 한번만 하면됨

`git clone` 원격저장소주소



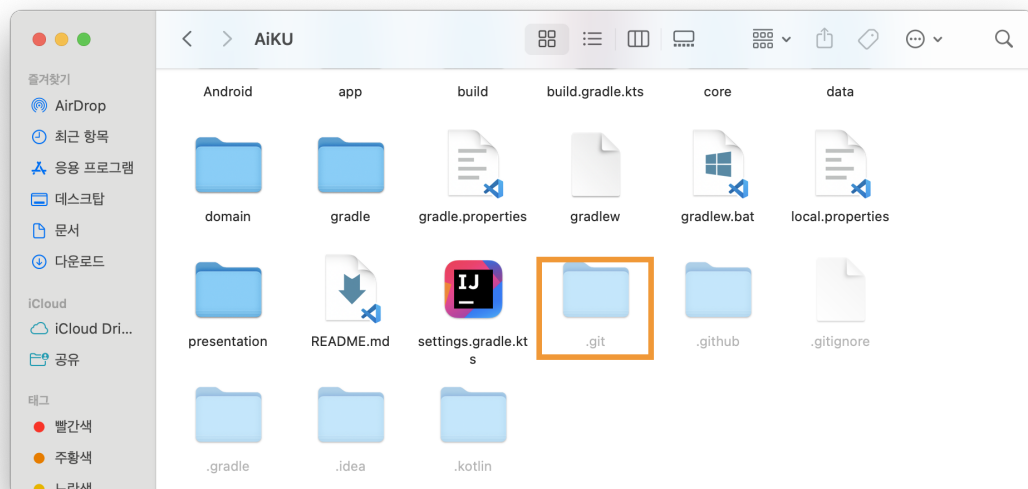
그럼 이렇게 konkuk이라는 폴더에 레포지토리(AiKU)가 불러와짐

3. `cd` 폴더경로 : AiKU로 이동

```
AiKU — -zsh — 80x24
Last login: Wed Oct 30 16:35:44 on console
[hwang-yoonseon@hwang-yunseon-ui-MacBookPro ~ % cd /Users/hwang-yoonseon/Desktop/]
konkuk
[hwang-yoonseon@hwang-yunseon-ui-MacBookPro konkuk % cd /Users/hwang-yoonseon/Desktop/konkuk/AiKU]
[hwang-yoonseon@hwang-yunseon-ui-MacBookPro AiKU % ]
```

4. `git init`

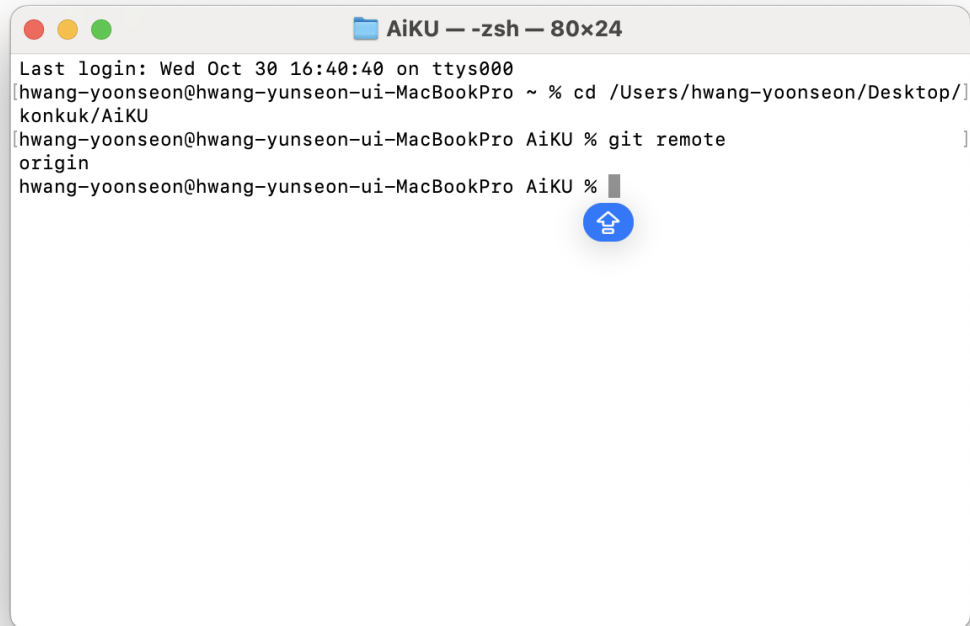
: 이것도 맨 처음에 한번만 하면 됨



숨겨진 폴더 열기 하면 이렇게 `.git`이라는 폴더가 생김

→ 이제부터 AiKU라는 폴더에서 git 기능을 이용할 수 있음.

5. `git remote`

A terminal window titled "AiKU — -zsh — 80x24" showing the execution of the `git remote` command. The output lists the 'origin' remote. A blue home button is visible below the terminal output.

```
Last login: Wed Oct 30 16:40:40 on ttys000
[hwang-yoonseon@hwang-yunseon-ui-MacBookPro ~ % cd /Users/hwang-yoonseon/Desktop/]
konkuk/AiKU
[hwang-yoonseon@hwang-yunseon-ui-MacBookPro AiKU % git remote
origin
hwang-yoonseon@hwang-yunseon-ui-MacBookPro AiKU % ]
```

원격저장소에 잘 연결되었다면, 저렇게 origin 이든 뭐든 뜰건데 만약에 아무것도 안뜬다면?

```
git remote add origin 원격저장소주소
```

6. 원격저장소와 로컬저장소 동기화

```
git fetch origin
```

```
AiKU — -zsh — 80x24
origin
[hwang-yoonseon@hwang-yunseon-ui-MacBookPro AiKU % git fetch origin
[hwang-yoonseon@hwang-yunseon-ui-MacBookPro AiKU % git branch -a
develop
* feature/#15
feature/#23
feature/login
main
remotes/origin/develop
remotes/origin/feature/#15
remotes/origin/feature/#17
remotes/origin/feature/#21
remotes/origin/feature/#22
remotes/origin/feature/#23
remotes/origin/feature/base
remotes/origin/feature/create-profile
remotes/origin/feature/example
remotes/origin/feature/group-api
remotes/origin/feature/group-members
remotes/origin/feature/login
remotes/origin/feature/schedule
remotes/origin/feature/temp
remotes/origin/main
hwang-yoonseon@hwang-yunseon-ui-MacBookPro AiKU %
```

검은글씨가 로컬브랜치 / remotes/origin으로 시작하는 애들이 원격저장소에 있는 브랜치

로컬브랜치에 main 만 뜰 거임

7. 다른 사람이 코드를 변경했다면 너도 그걸 로컬로 가져와야되겠지? 그래야 작업을 이어
서하지

`git pull origin 브랜치이름`

→ 이러면 원격저장소 브랜치의 모든 코드 내용을 로컬로 불러온거임

- 주의사항

만약 니가 같은 브랜치에서 작업중이었다면, pull을 할 경우 니가 작업한 내용과 불러온 내용을 자동으로 합쳐줌 → 코드가 충돌이 나면 수동으로 충돌을 해결하고 다시 git pull 하면 합쳐짐

8. 내가 지금 어느 브랜치에서 작업중인거지?

`git branch` : * 초록색으로 표시된 부분이 현재 니 위치

```

switched to branch main
[hwang-yoonseon@hwang-yunseon-ui-MacBookPro AiKU % git branch
  develop
  feature/#15
  feature/#23
  feature/login
* main

```

9. 새로운 브랜치를 생성해보자

```
git checkout -b 브랜치이름
```

내가 만약 git checkout -b feature/#15 를 입력했다면,

```

[hwang-yoonseon@hwang-yunseon-ui-MacBookPro AiKU % git branch
  develop
* feature/#15
  feature/#23
  feature/login
  main

```

feature/#15가 생성되면서 현재 내 위치가 feature/#15가 되는거임

8. 로컬에서 작업한 내용을 원격저장소(github)에 올리는 경우

```
git add .
```

```
git commit -m "이런 이런거 작업함"
```

```
git push origin 브랜치이름
```

니가 로컬에서 뭔가 변경한 내용이 있다면 반드시 add, commit 을 한 이후에 pull을 하든 push 를 하든 다른 작업을 할 수 있음

```
git push origin 브랜치이름
```

feature/#15에서 작업했다면 `git push origin feature/#15`

 main ▾

 13 Branches  Tags

Switch branches/tags



 Find or create a branch...

Branches

Tags

feature/base

feature/create-profile

feature/example

feature/group-api

feature/group-members

feature/login

feature/schedule

feature/#15

feature/#21

feature/#22

View all branches

그럼 원격저장소(github)에 이렇게 feature/#15 브랜치가 생기고 니 코드가 올라감

다른 사람 코드를 니꺼랑 합치는건 `git merge`인데 심화내용이니 다음시간에~~ㄱ~~—