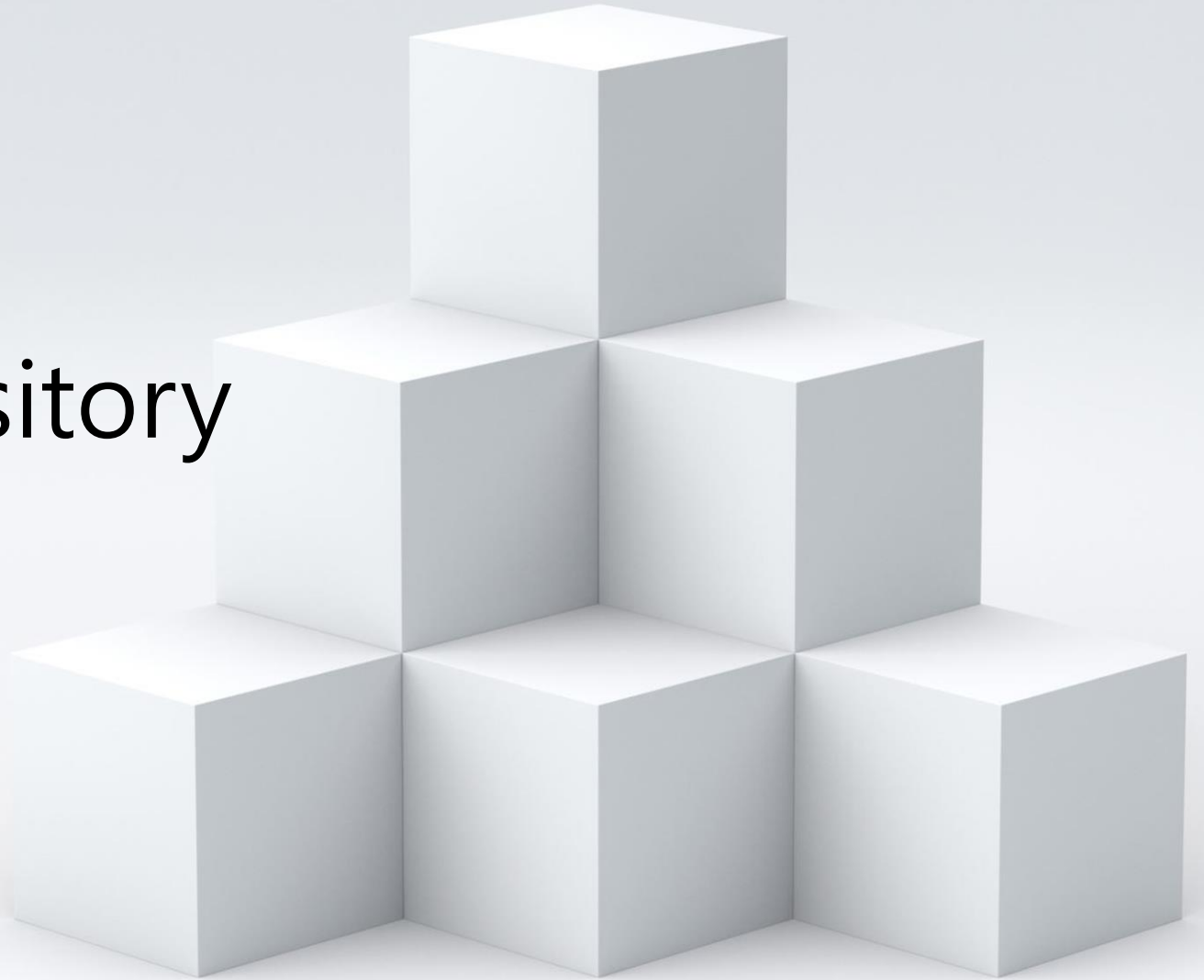


# Software Engineering

- github -
- a remote git repository web service-

Professor Han-gyoo Kim

2022

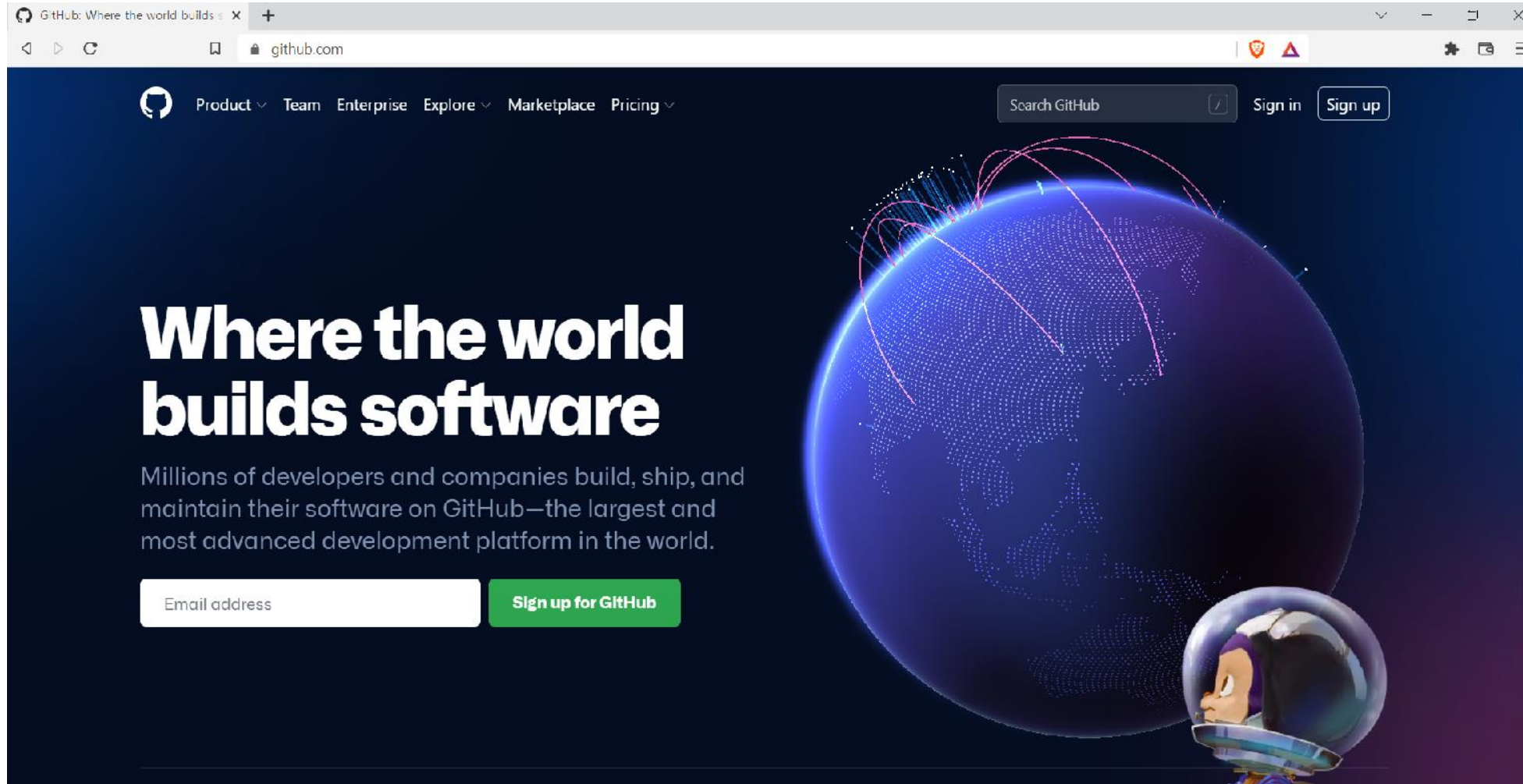


# Remote git server (repository)

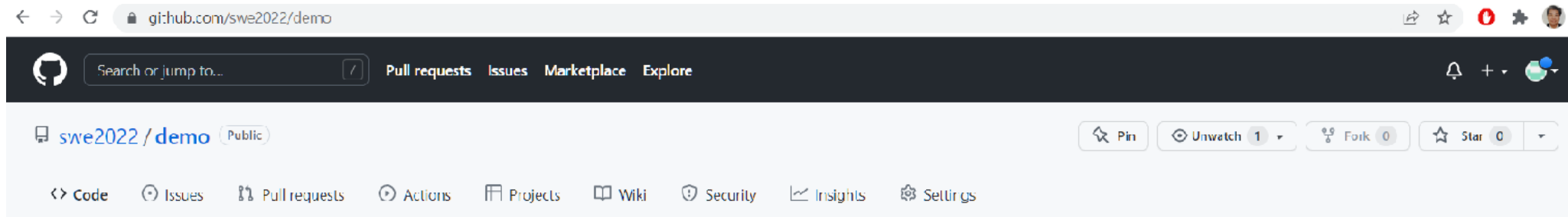
- Git은 open source project로서 remote git server도 설치하여 운영할 수 있음
- 2008년 github이 설립되어 서비스를 시작하던 초창기에는 회사들이 private remote git 를 설치해서 사용하는 것이 상당히 많았으나
- 관리의 어려움 등등으로 github 과 같은 remote repository 서비스를 이용하는 것이 관례가 됨
- Github은 remote repository 중 하나 – 여러 유사 서비스가 있음
- Github 은 MS가 2018년에 인수하여 repository 서비스 외에 CI/CD의 다양한 기능을 Github에 합쳐 넣은 유료 DevOps 도구로 발전시키고 있음

Github = one of Git remote repo services





# Github 계정 만들기 (github.com)



# Github 시작하기



## 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](#).


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

```
echo "# demo" >> README.md
git init
git add README.md
git commit -m "first commit"
git branch -M main
git remote add origin https://github.com/swe2022/demo.git
git push -u origin main
```



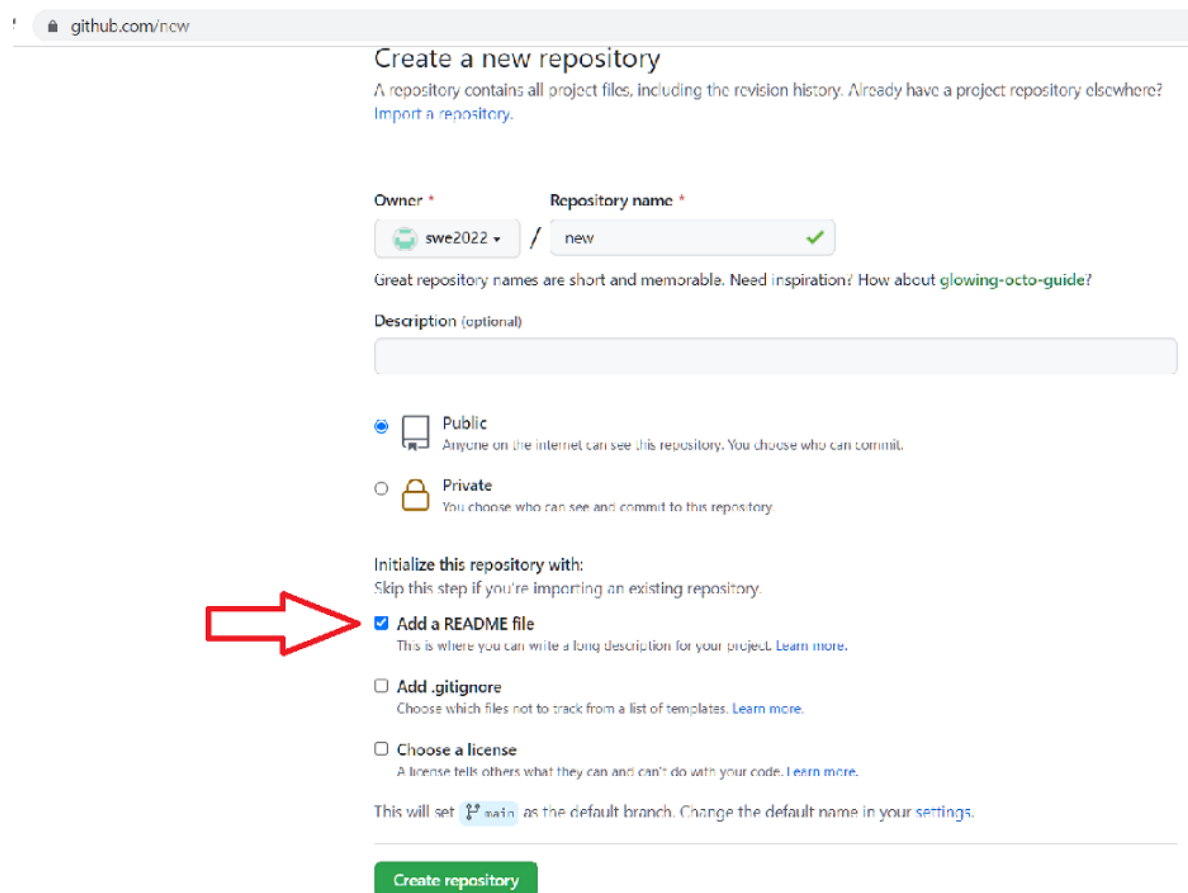
## ...or push an existing repository from the command line

```
git remote add origin https://github.com/swe2022/demo.git
git branch -M main
git push -u origin main
```



# Github GUI

- 새로이 repo를 만들 때 아래 option을 선택하면서 생성하면 조금 덜 성가심



github.com/new

## 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 \*

swc2022 / new ✓

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

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.](#)

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

Create repository

# No more password supported since Aug. 2021

- Have to use Personal Access Token instead of password authentication
- Settings -> developer settings -> personal access tokens -> generate token
- SSH Key 만들어 사용하는 방법 등도 있음
- 단 한번만 token key가 만들어지므로 메모장에 보관 (왜 만들어 놓은 key를 볼 수 없도록 했을까? 🤔)
- 매번 local 기계에서 github repo에 연결할 때마다 key 복붙이 불편하면 윈도우에 서는 control panel->credential manager->edit -> personal key 복붙하고 저장
- 잃어버리거나 문제가 생길 것 같으면 예전 것은 없애고 다시 생성
- 불편! – 유료 서비스 (private)는 이런 불편 없음! 역시 마이크로소프트!!!
- Github desktop에서는?

# Git remote commands

- `git remote add origin <remote repository URL>`
- `git clone`
- `git push`
- `git pull`
- Fork는 git 명령어가 아니라 github의 기능 – git의 명령어 조합으로 구현



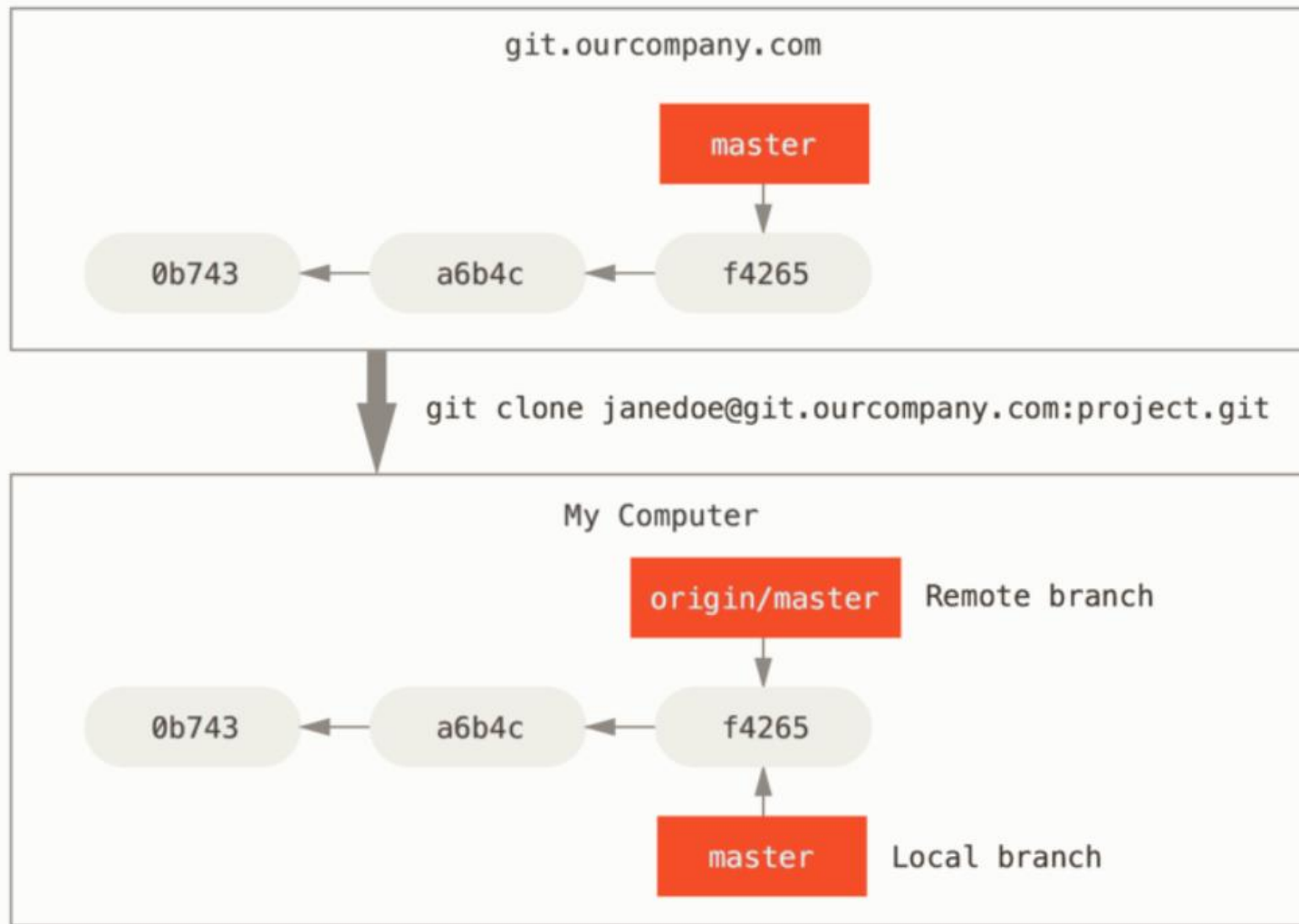
# Git remote repo demo

Git clone

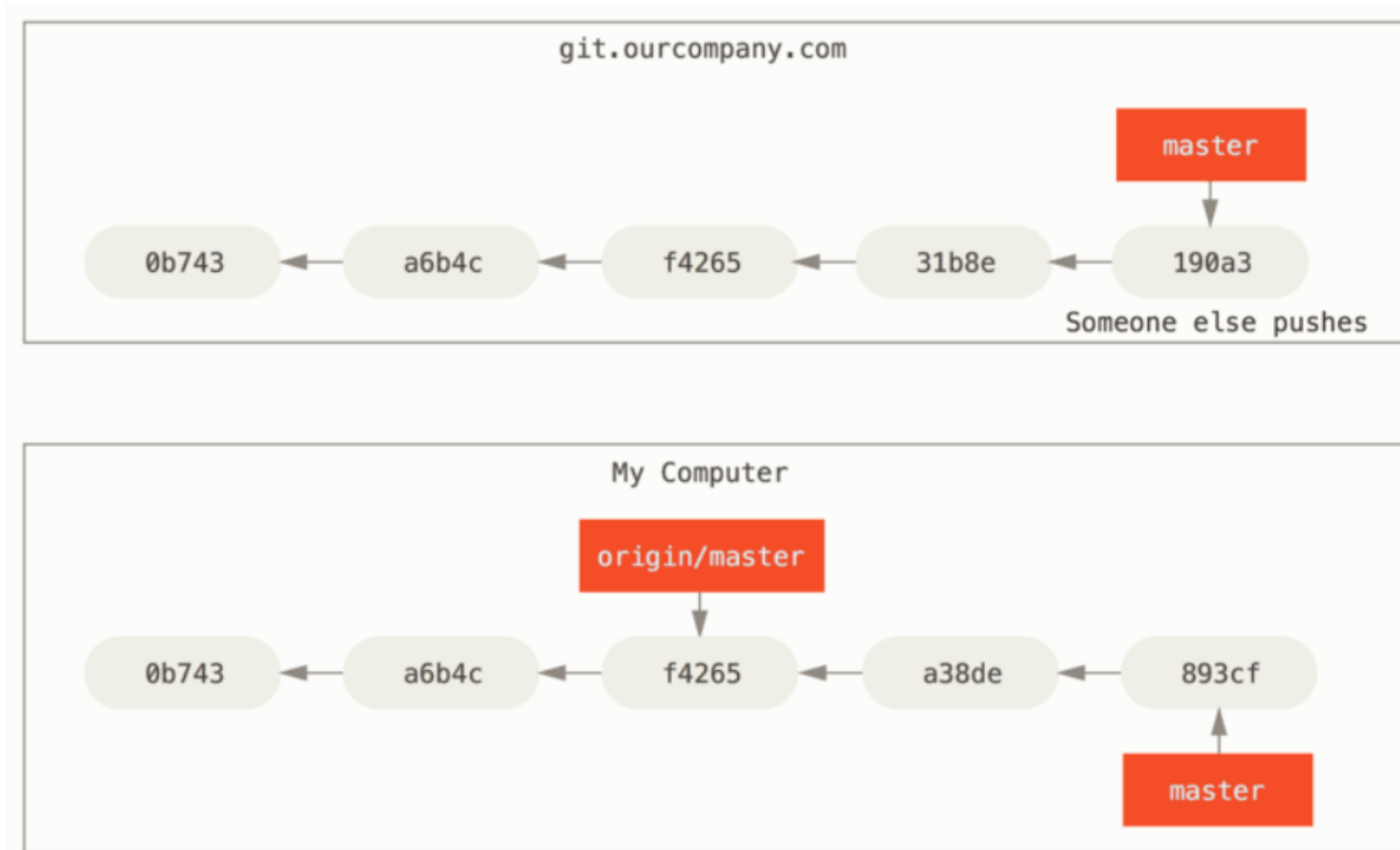
Git push

# git clone

- Remote repo에는 아무 변경 없이 local repo로 복제해 오는 것

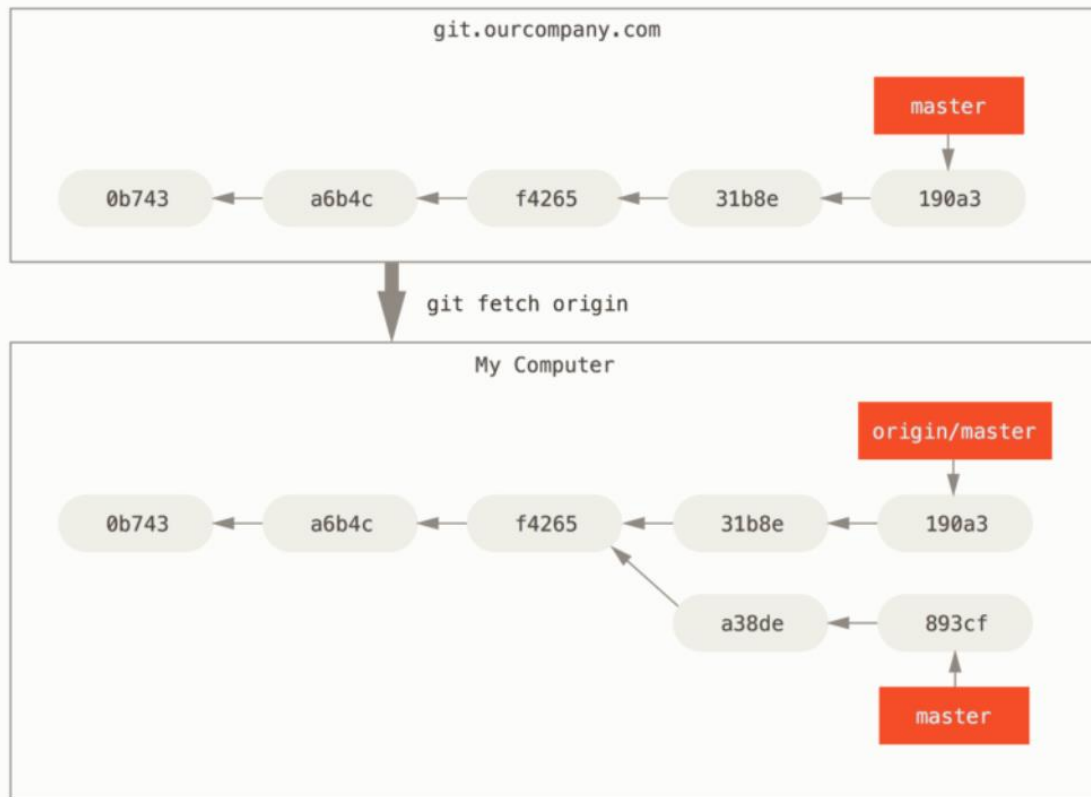


# 공동 작업에 의한 commit history



# git fetch origin – 동기화 (merge는 동기화의 요건이 아님)

- Fetch는 commit point 오브젝트와 실제 data를 local repo로 가져오는 오지만 아직 local working directory의 파일 시스템에는 반영시키지 않음
- 사용자가 필요에 따라 merge 명령을 사용해서 local working directory에 반영
- Pull 명령어를 사용하여 fetch+merge 를 동시에 할 수 있으나 fetch 한 다음에 확인 후 merge하는 것을 권장

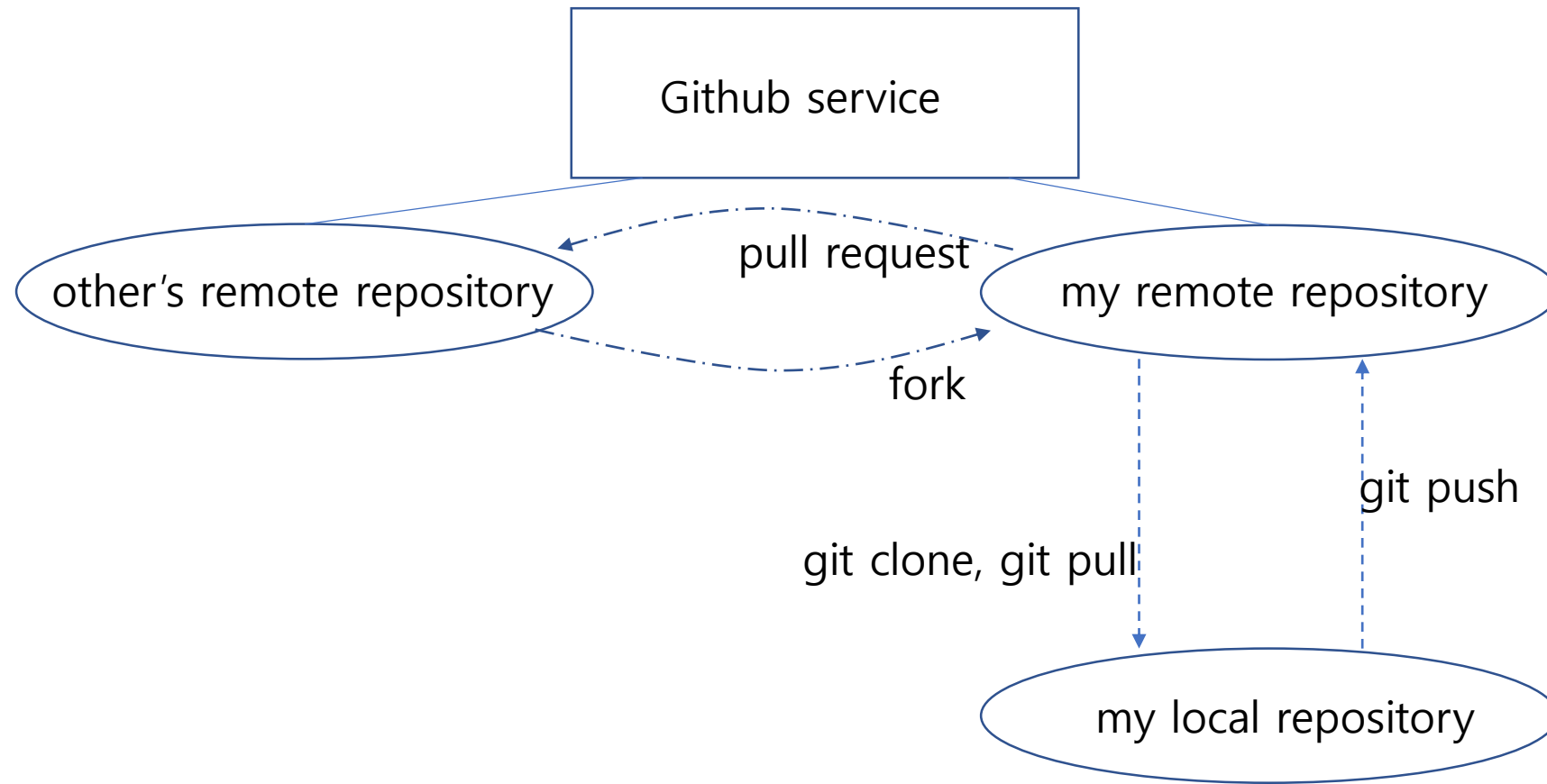


# Git branch and related commands 데모

# Github 데모

- GUI demo
- branch
- pull request
- merge
- fork (for outsiders' contribution)
  - Clone : remote repo에서 자신의 local repo로 복제
  - fork : 다른 github 계정의 repo를 자신의 github 계정으로 복사 (git 명령이 아님)

# Github fork and pull request



- git fetch vs pull (=fetch and merge)