Git For Everybody

by 코딩하러 오시영 ohah

1. Git For Me (0.5 hour)

2. Git For You & Me

3. Contribution For Everybody

Guide : Git scm

Git Cheat Sheet Eng | Ko / Visual Git Cheat Sheet

🗺️ Keyword Map - What is 'Git'?

Version Control

🗺️ Keyword Map - Git For Me

💡 **Concept** : Version Control

📖 **Terms** : Local Repository, Remote Repository, Untracked | Tracked, Staged, .gitignore

⌨️ **Command** : clone, add, commit, push, pull, log, diff, status

👋 **Action** : Basic workflow

🗺️ Keyword Map - Git For You & Me

💡 **Concept** : Branch, Issue, Pull Request(PR)

📖 **Terms** :

⌨️ **Command** :

👋 **Action** : Fix conflict, PR

🗺️ Keyword Map - Contribution For Everybody

- Find / Start Open source project
- Join The Community 🤝

  👋 **Action** : First Contribution

What is 'Git'?

The Birth of Git

- 👐 오픈소스 프로젝트인 Linux 개발 커뮤니티에서 개발
- 빠르고 빠르고 단순한 구조로 동시에 개발이 가능하고, 대형 프로젝트에서도 유용한 걸 만들어보자

Photo by Felipe Salgado on Unsplash

What is Version Control?

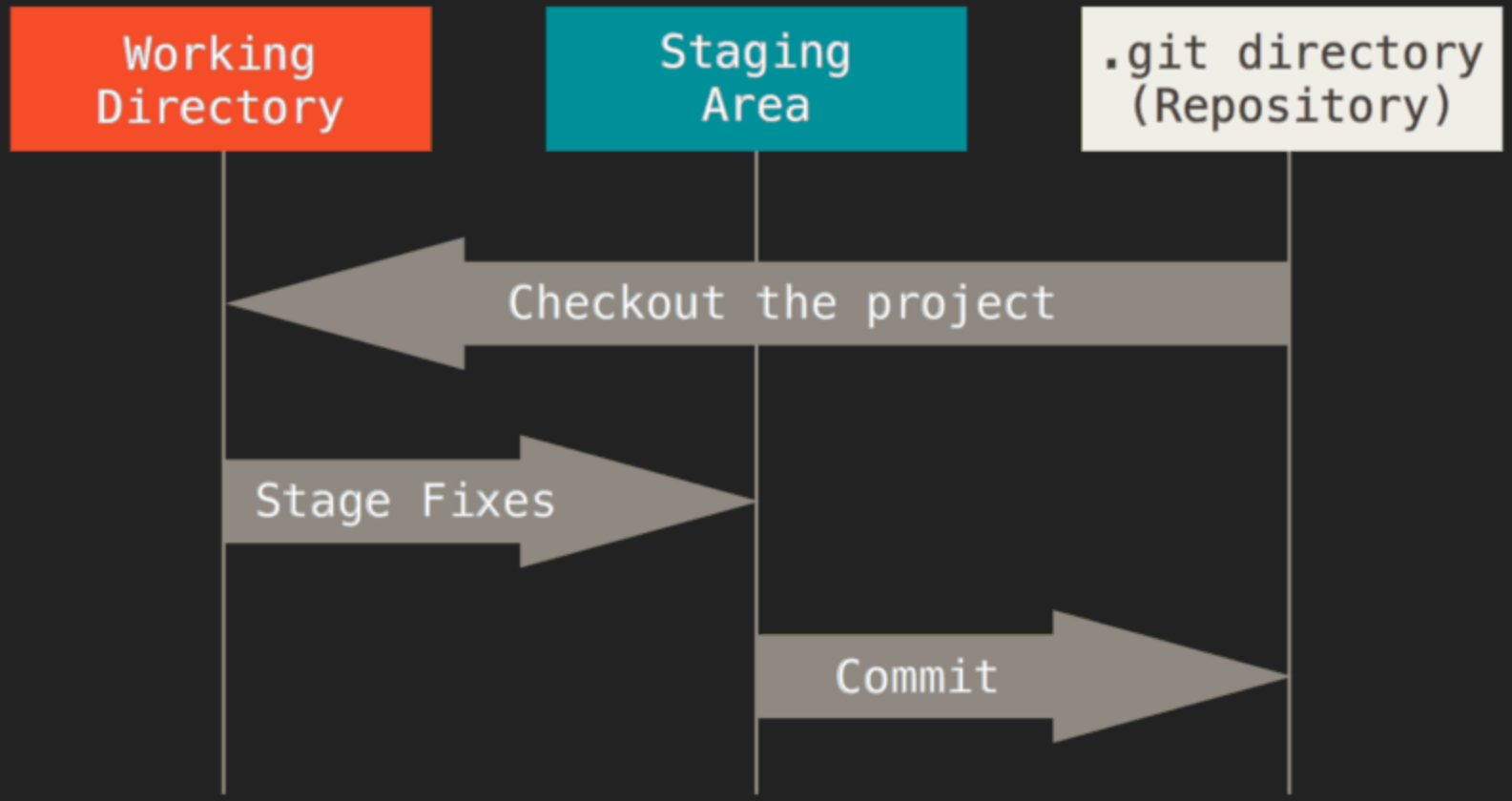
> Version control is a system that records changes to a file or set of files over time so that you can recall specific versions later.

-> 특정 버전으로 되돌릴 수 있다. 그러기 위해 **작업 내역**을 남기는 것이 중요!

Git For Me - Three states

- Git 은 Git DB에 '데이터를 저장'합니다.
- Git 은 파일을 세 가지 상태로 관리합니다.
  - 🥚 modified : 파일이 수정되었어요. in Working Directory
  - 🐣 staged : 수정된 파일 중에 여기 있는 걸 곧 commit 할 꺼에요. to the Staging Area
  - 🐥 committed : 데이터가 local db 에 저장되었어요. to .git directory(Repository)

Git For Me - Three states



Git scm - 1.3 Figure 6. Working tree, staging area, and Git directory

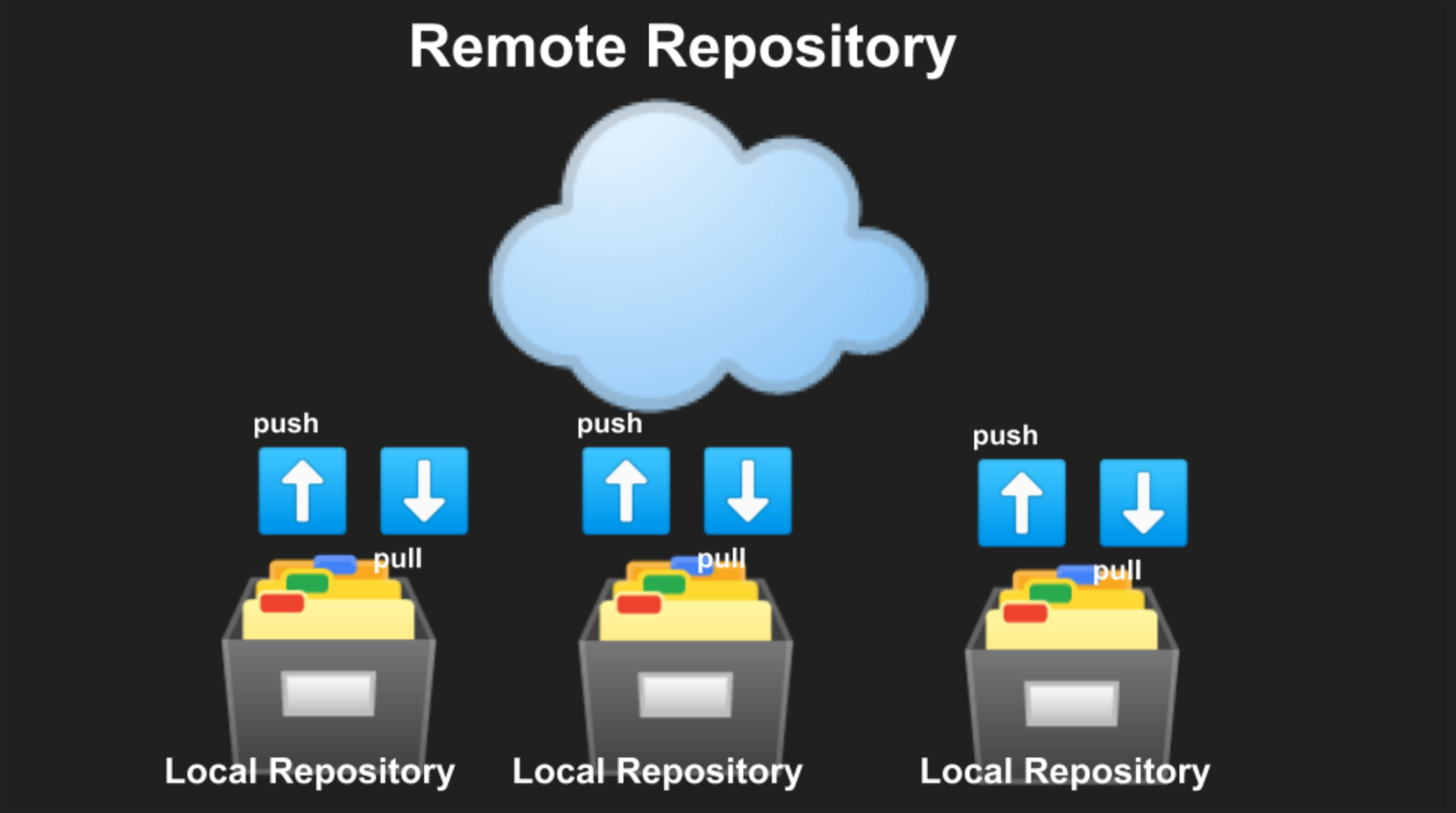
Git For Me

Version Control

👋 Action 01. Basic workflow

👋 Action 02. Conflict

Git For Me - Local / Remote Repository



Git For Me - 👋 Action 01. Basic workflow

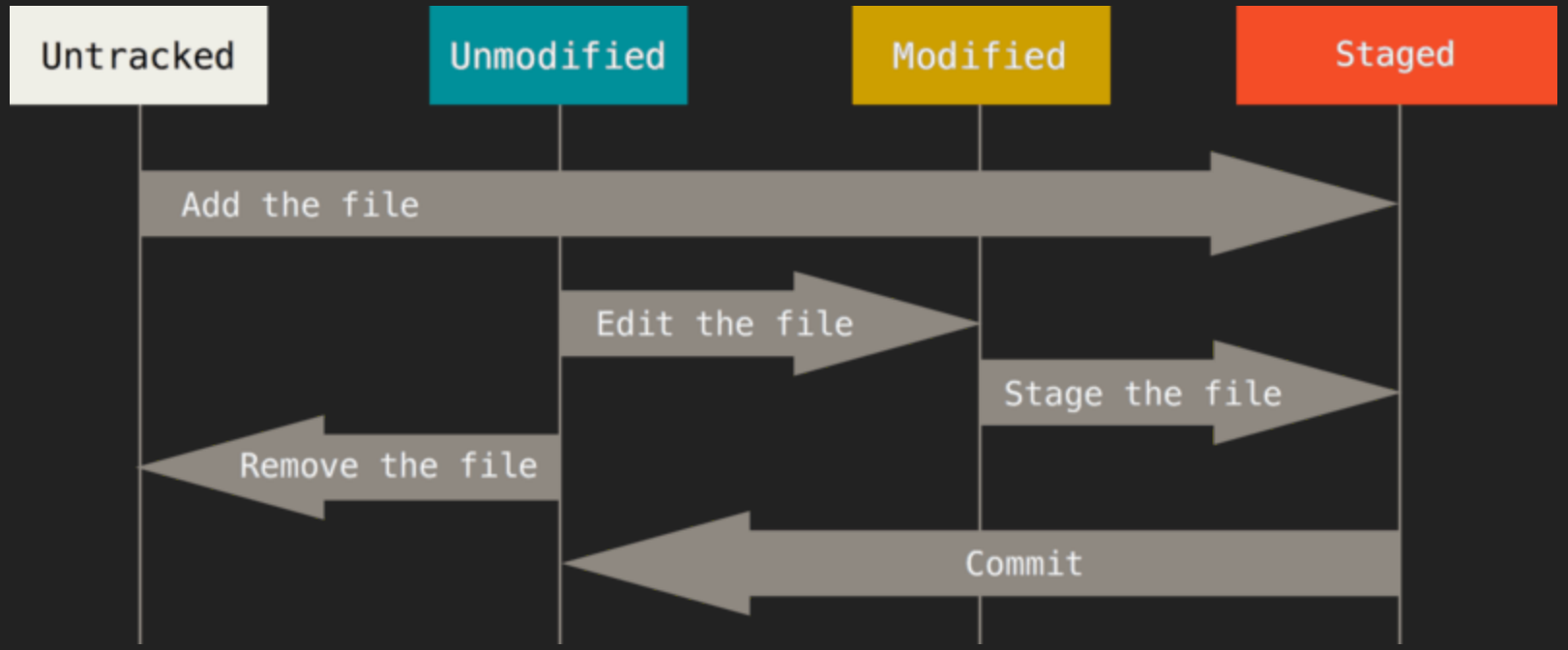
Learn

- `clone` : kind of download
- `add` : staging
- `commit` ☑️ : Save working log to DB
- `push` ➡️ : Load working log **from Local to Remote**
- `pull` ⬅️ : Load working log **from Remote to Local**

Git For Me - 👋 Action 01. Basic workflow

1. Create Repository in Github
2. `clone` to local (repo address)
3. Modify files
4. `add` files what you want to commit
5. `commit` : good commit message!
6. `push`
7. check your github repo
8. Edit file in github repo
9. `pull`

Git For Me - Commit what I want



- Untracked : `.gitignore`
- Show Files status : `status`
- Staged : `add`
- Unstaged : `git reset HEAD <file>`
- To know exactly what you changed : `diff`

One more time

👋 Action 01. Basic workflow - commit files only what you want

You don't have to remember these things. **Find** *it when you need it*.

Just Do it ❗ X 100

Git For Me - View History

- `log`
- In Github
  https://github.com/your_github_name/repo_name/commits/master
  like this
  https://github.com/pandas-dev/pandas/commits/master
- In github's file, click 'blame' button

Git For Me 2/3

- Issue : pandas Issue
- Release - versioning : Pycon KR CoC release note
- License : pandas License

Git For Me (more) 3/3

- branch : per feature, issue / git | github | gitlab workflow
- project board : kanbas style pandas Project board