

# Git の紹介

yiw0

2021年7月30日

# 発表内容

## 1. Git の概要

## 2. Git の簡単な使い方

## 3. GitHub で git リポジトリ管理



# git とは

- 分散型バージョン管理システム (VCS)
  - ファイルの変更履歴を管理できる
  - 複数人でバージョン管理可能
- Linus Torvalds が2005年に開発
- 日本人の濱野 純がメンテナ (2005-)

Git を利用しているプロジェクト例

- Linux カーネル  
(<https://github.com/torvalds/linux>)

## 非表示

ワタリナックスフォットデビル Tシャツを着てる  
Linus の写真

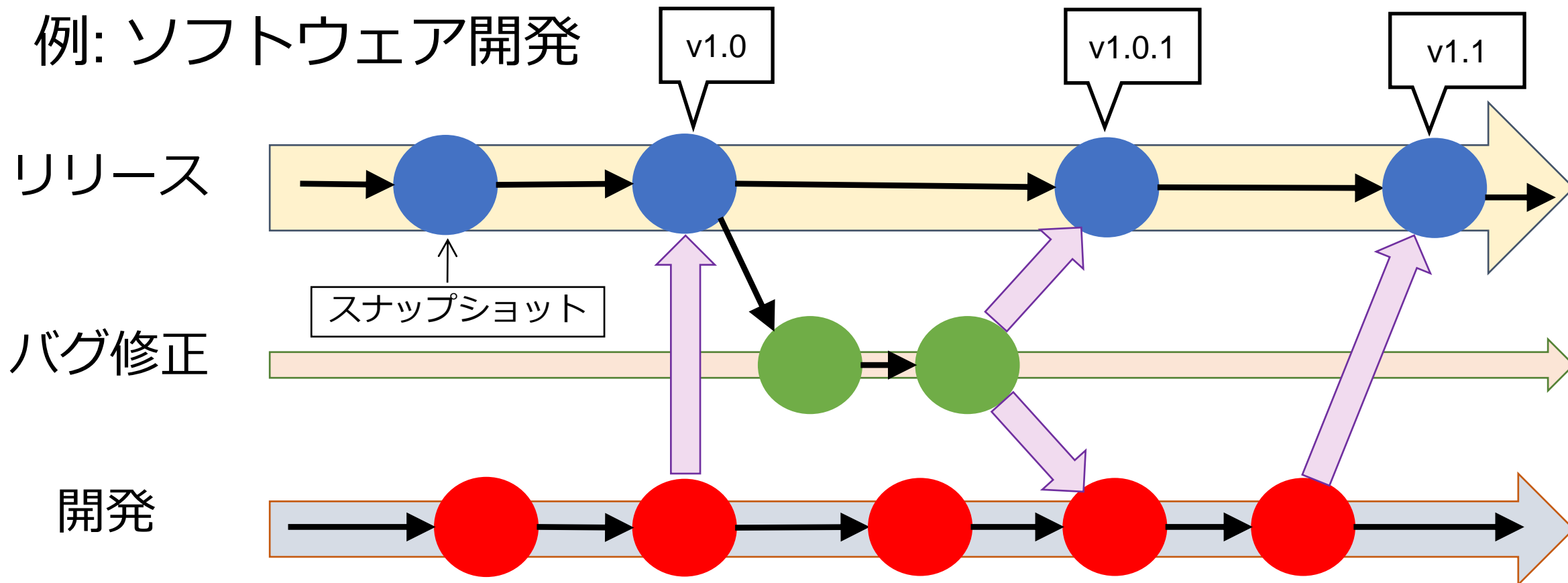
Git 開発者の Linus Torvalds  
(Linux カーネル開発者)

Git logo by Jason Long (CC BY 3.0)

# バージョン管理

- スナップショットを管理
- 以前のスナップショットに戻ることができる

例: ソフトウェア開発



# 用語

この図全体: repository

**commit**

(ID: SHA1 hash)

**tag**

**branch**

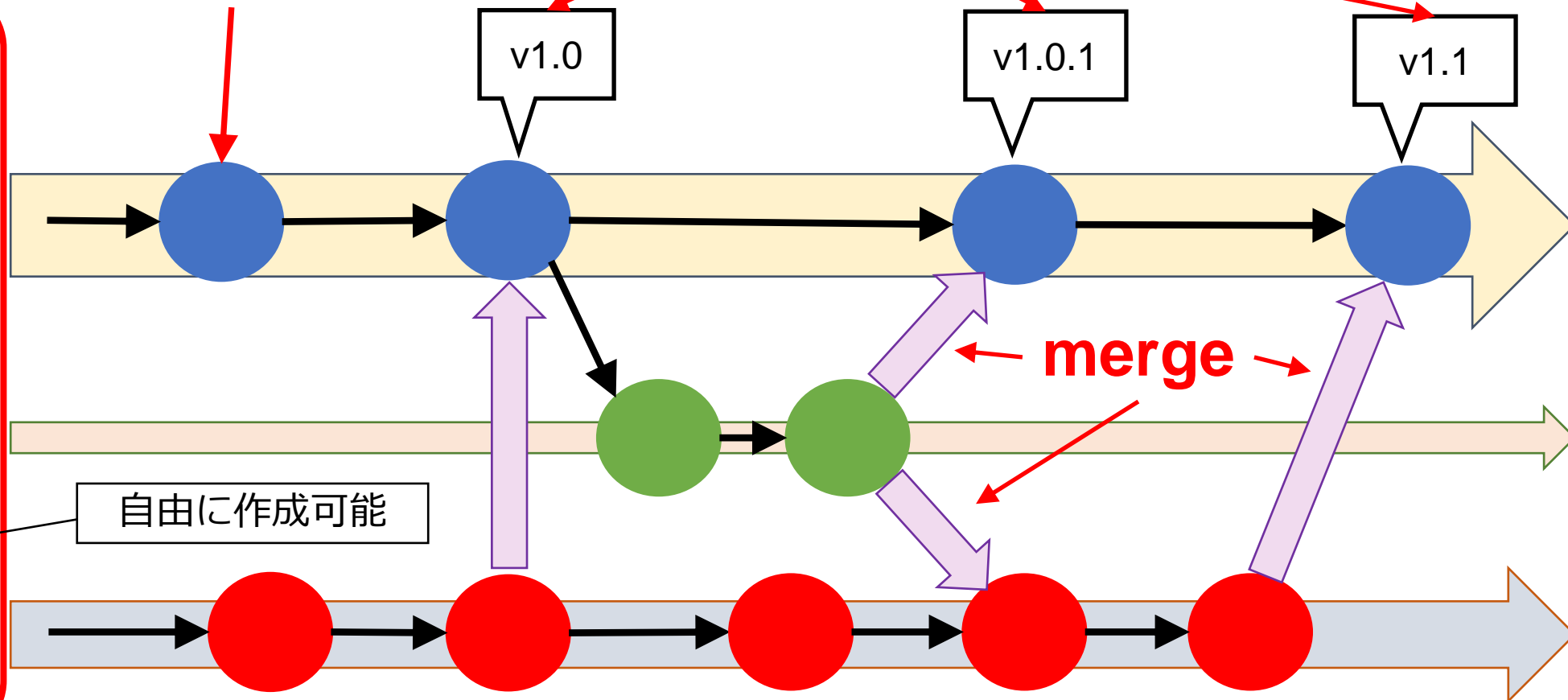
**master**  
(デフォルト)

hotfix

develop

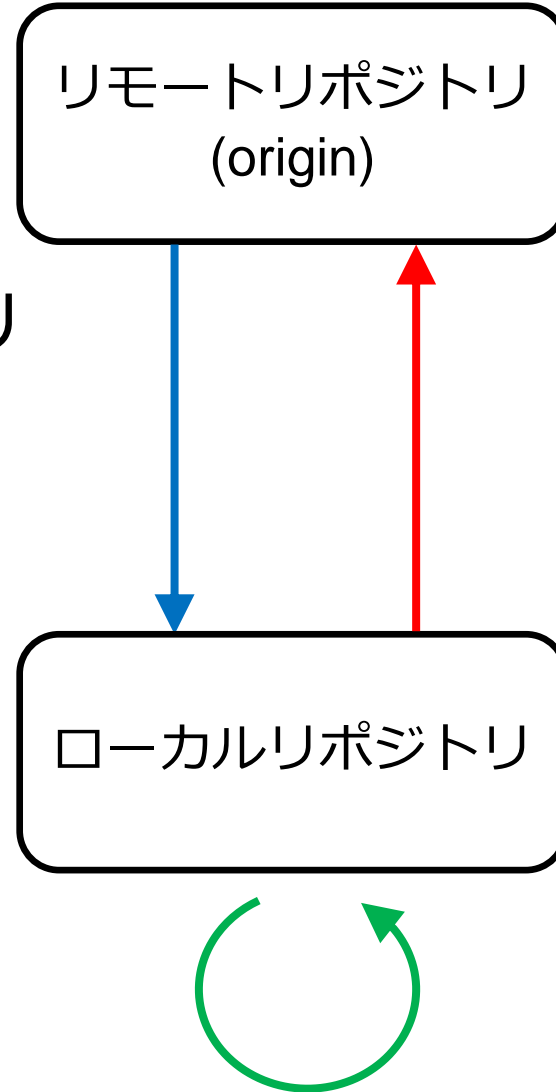
自由に作成可能

**merge**



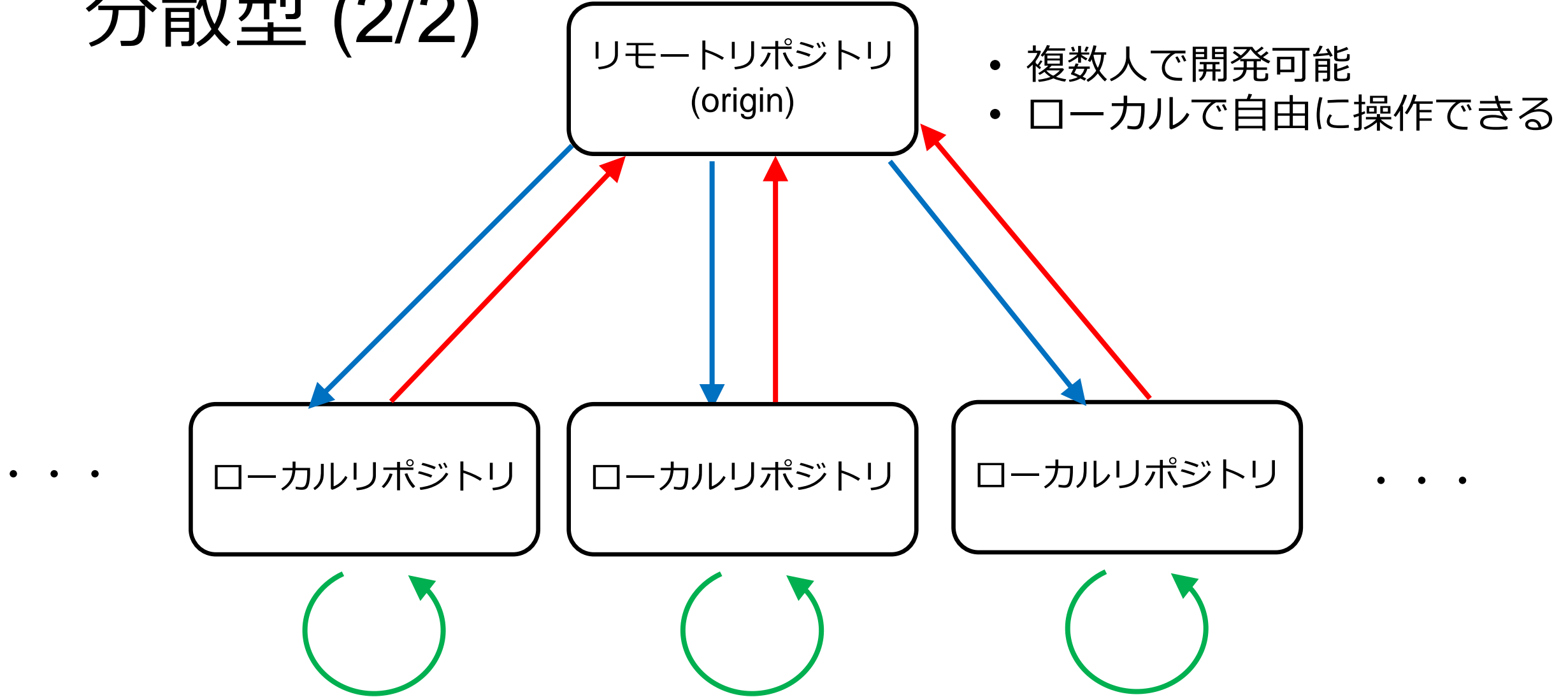
# 分散型 (1/2)

- **git clone** でリポジトリの**完全なコピー**をローカルに生成 (初回)
- **git pull** で origin の変更をローカルに反映



- **git push** でローカルの作業を origin に反映
- **オフラインで作業可能**
  - commit できる
  - branch 作れる
  - merge できる (簡単)

# 分散型 (2/2)



# 発表内容

1. Git の概要

**2. Git の簡単な使い方**

3. GitHub で git リポジトリ管理

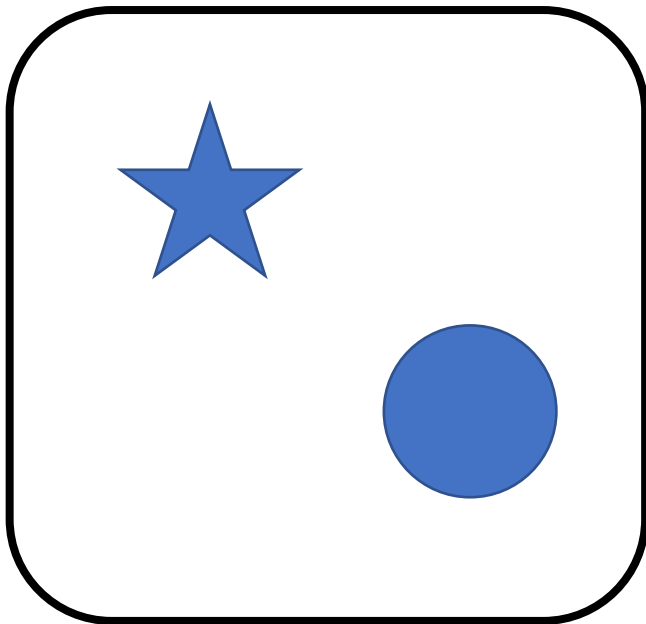


# Git の初期設定 ・ 最初のコミット

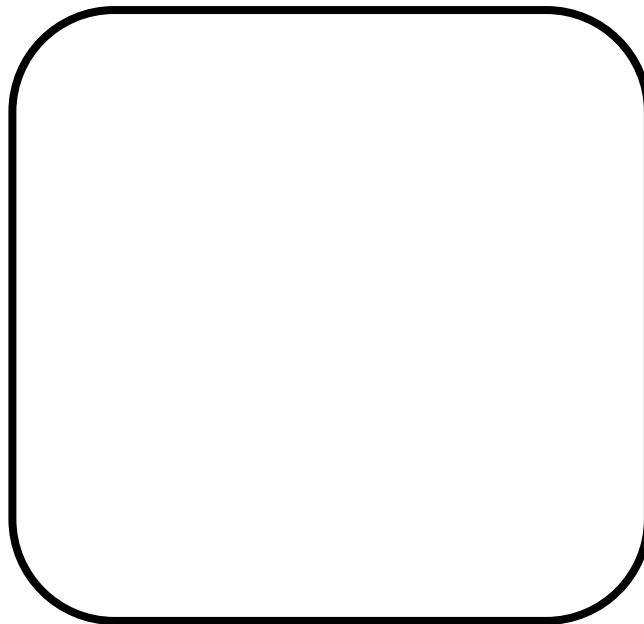
1. `git config --global user.name 'Taro Yamada'`
  - ユーザ名を設定
2. `git config --global user.email 'taro@example.com'`
  - Eメールアドレスを設定
3. `git init`
  - カレントディレクトリに空の git リポジトリを作成
4. `git add README.md`
  - ファイル README.md を **staging area/index** に追加
5. `git commit -m 'Initial commit'`
  - コミットメッセージ「Initial commit」を付けてコミット

# Staging area/index (1/4)

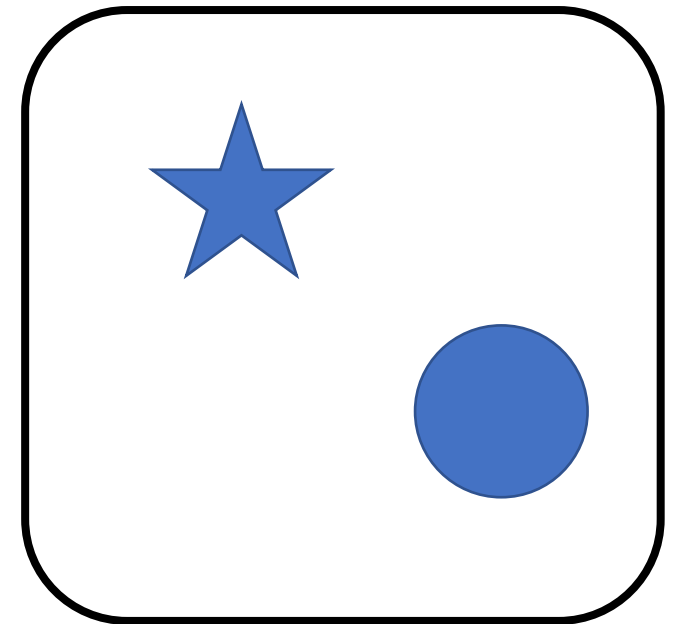
working directory



staging area/index

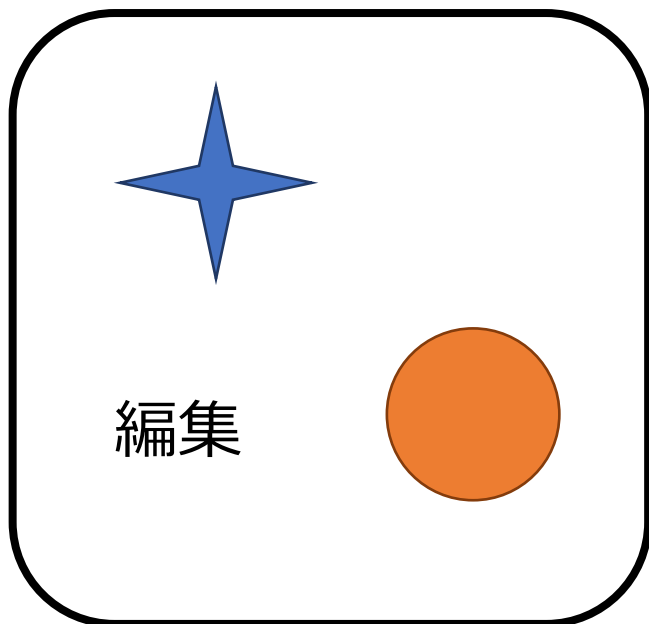


repository

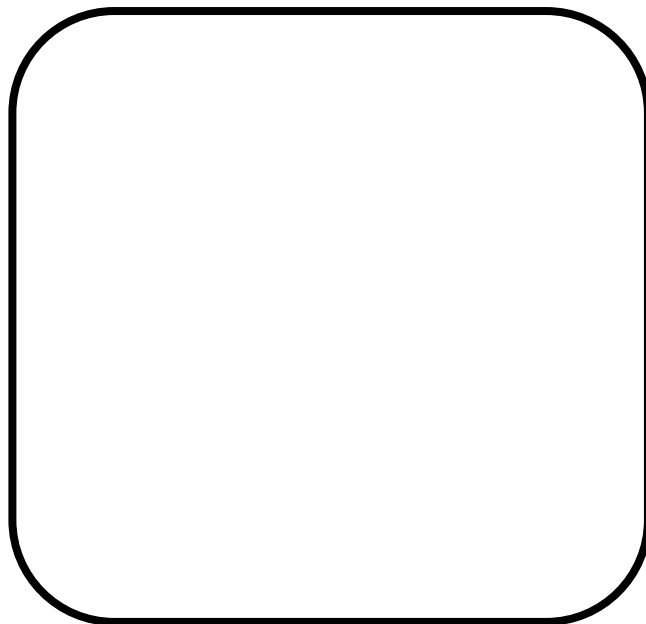


# Staging area/index (2/4)

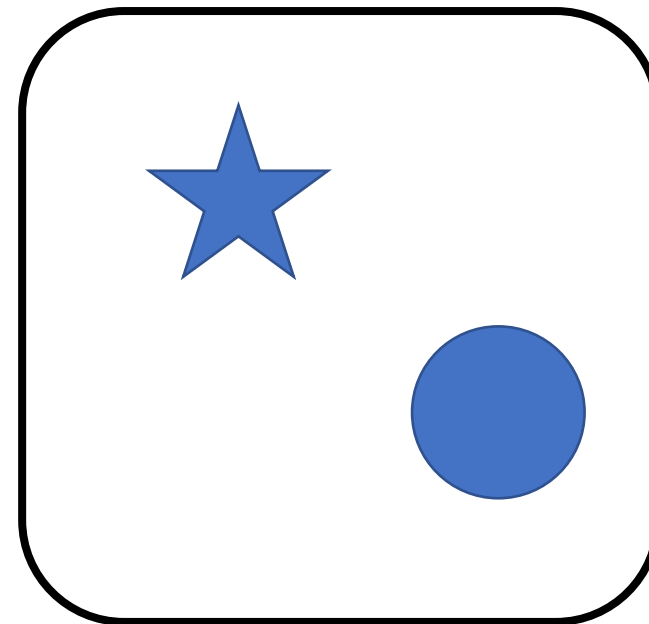
working directory



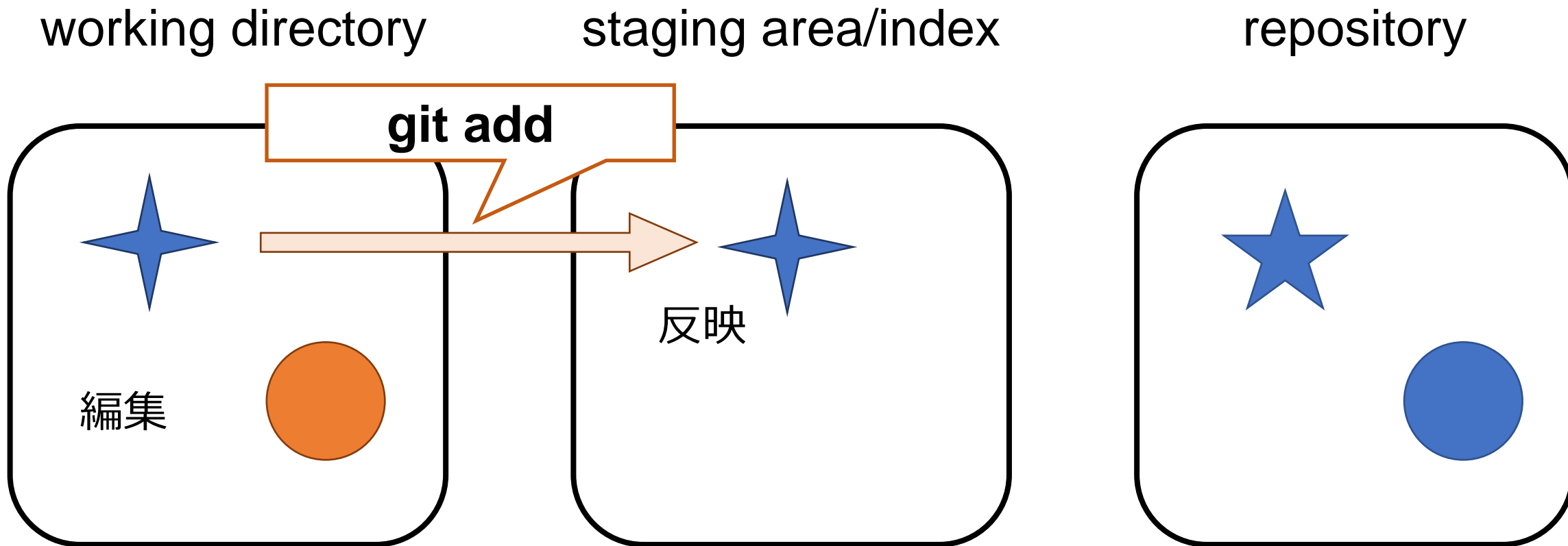
staging area/index



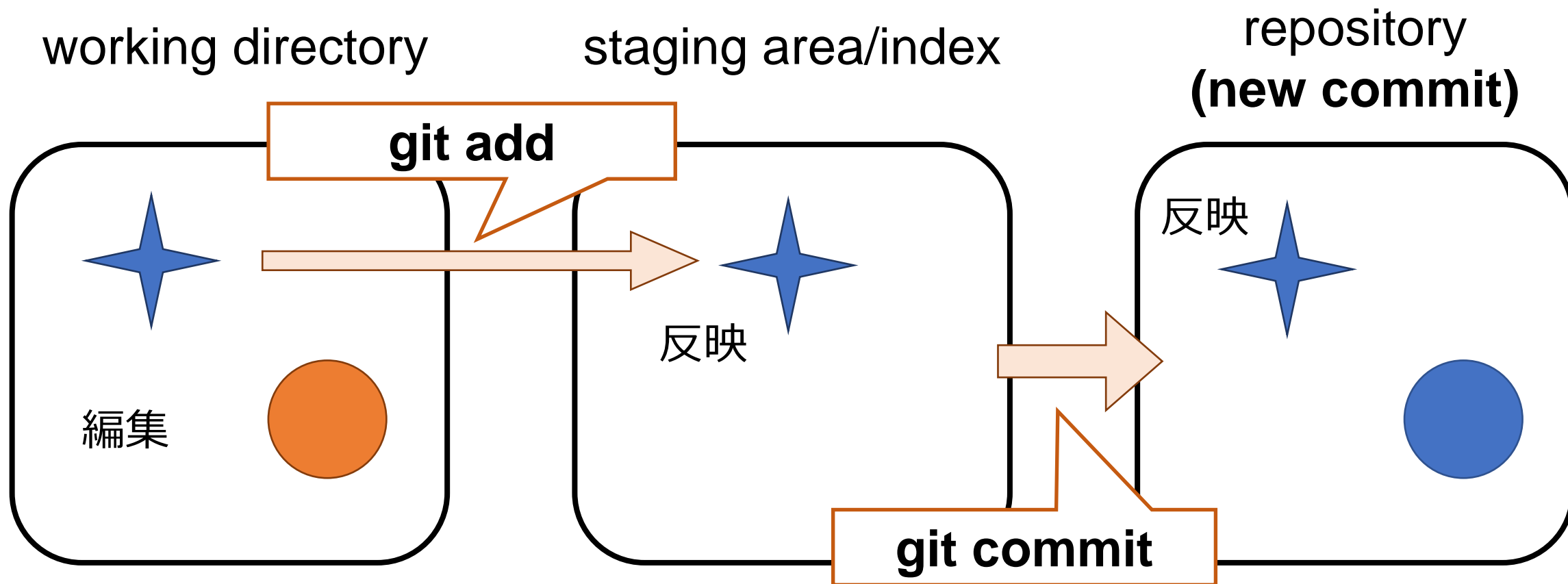
repository



# Staging area/index (3/4)



# Staging area/index (4/4)





# Git のコマンド例

- git log
  - コミットログを確認
- git branch/git tag
  - ブランチ/タグ一覧を表示
- git branch develop
  - develop ブランチを作成
- git checkout develop
  - develop ブランチにチェックアウト (切換え)
- git merge master
  - 現在のブランチに master ブランチの (最新) コミットをマージ

## Create a Repository

From scratch – Create a new local repository  
\$ git init [project name]

Download from an existing repository  
\$ git clone my\_url

## Observe your Repository

List new or modified files not yet committed  
\$ git status

Show the changes to files not yet staged  
\$ git diff

Show the changes to staged files  
\$ git diff --cached

Show all staged and unstaged file changes  
\$ git diff HEAD

Show the changes between two commit ids  
\$ git diff commit1 commit2

List the change dates and authors for a file  
\$ git blame [file]

Show the file changes for a commit id and/or file  
\$ git show [commit]:[file]

Show full change history  
\$ git log

Show change history for file/directory including diffs  
\$ git log -p [file/directory]

## Working with Branches

List all local branches  
\$ git branch

List all branches, local and remote  
\$ git branch -av

Switch to a branch, my\_branch, and update working directory  
\$ git checkout my\_branch

Create a new branch called new\_branch  
\$ git branch new\_branch

Delete the branch called my\_branch  
\$ git branch -d my\_branch

Merge branch\_a into branch\_b  
\$ git checkout branch\_b  
\$ git merge branch\_a

Tag the current commit  
\$ git tag my\_tag

## Make a change

Stages the file, ready for commit  
\$ git add [file]

Stage all changed files, ready for commit  
\$ git add .

Commit all staged files to versioned history  
\$ git commit -m "commit message"

Commit all your tracked files to versioned history  
\$ git commit -am "commit message"

Unstages file, keeping the file changes  
\$ git reset [file]

Revert everything to the last commit  
\$ git reset --hard

## Synchronize

Get the latest changes from origin (no merge)  
\$ git fetch

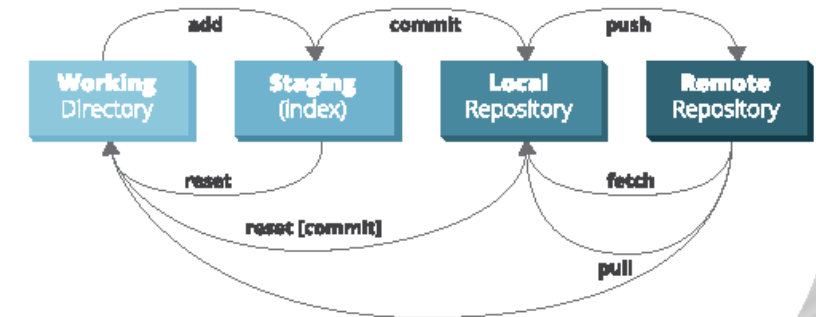
Fetch the latest changes from origin and merge  
\$ git pull

Fetch the latest changes from origin and rebase  
\$ git pull --rebase

Push local changes to the origin  
\$ git push

**Finally!**  
When in doubt, use git help  
\$ git command --help

Or visit <https://training.github.com/> for official GitHub training.



<https://www.jrebel.com/blog/git-cheat-sheet>

# 発表内容

1. Git の概要
2. Git の簡単な使い方
- 3. GitHub で git リポジトリ管理**

# GitHub で Git リポジトリ管理 (1/2)

- GitHub: Git のホスティングサービス (一番有名)
- 使い方: <https://docs.github.com/ja/github/getting-started-with-github>
  1. GitHub にユーザ登録 (ユーザ名: taro)
  2. **SSH 鍵ペア生成・設定** ←
  3. GitHub に SSH 公開鍵登録
    - <https://docs.github.com/ja/github/authenticating-to-github/connecting-to-github-with-ssh>
  4. GitHub に空のリポジトリ作成 (リポジトリ名: repo)
  5. `git remote add origin github:taro/repo.git`
    - origin に GitHub 上の repo を登録
  6. `git branch -m master main`
    - master ブランチを main ブランチにリネーム
  7. `git push -u origin main` (-u は初回のみ)

次のスライド参照

全世界に「公開」か「非公開」  
を選択可能



# GitHub で Git リポジトリ管理 (2/2)

- ~/.ssh で作業 (Linux 上の作業を想定)
- ssh-keygen -f github\_key -t ed25519 -C main\_laptop
  - github\_key (秘密鍵), github\_key.pub (公開鍵) を生成
- ~/.ssh/config を生成:

ssh -T **github**  
でログインできれば設定成功

```
IdentitiesOnly yes

Host github
  HostName github.com
  Port 22
  User git
  IdentityFile ~/.ssh/github_key
```

~/.ssh/config

# まとめ

- Git を使うと高度なバージョン管理が可能
- GitHub などを使うと他人とバージョン管理が可能
  - バックアップとしても使える
- プレーンテキストファイル (LaTeX, ソースコードなど) の管理におすすめ