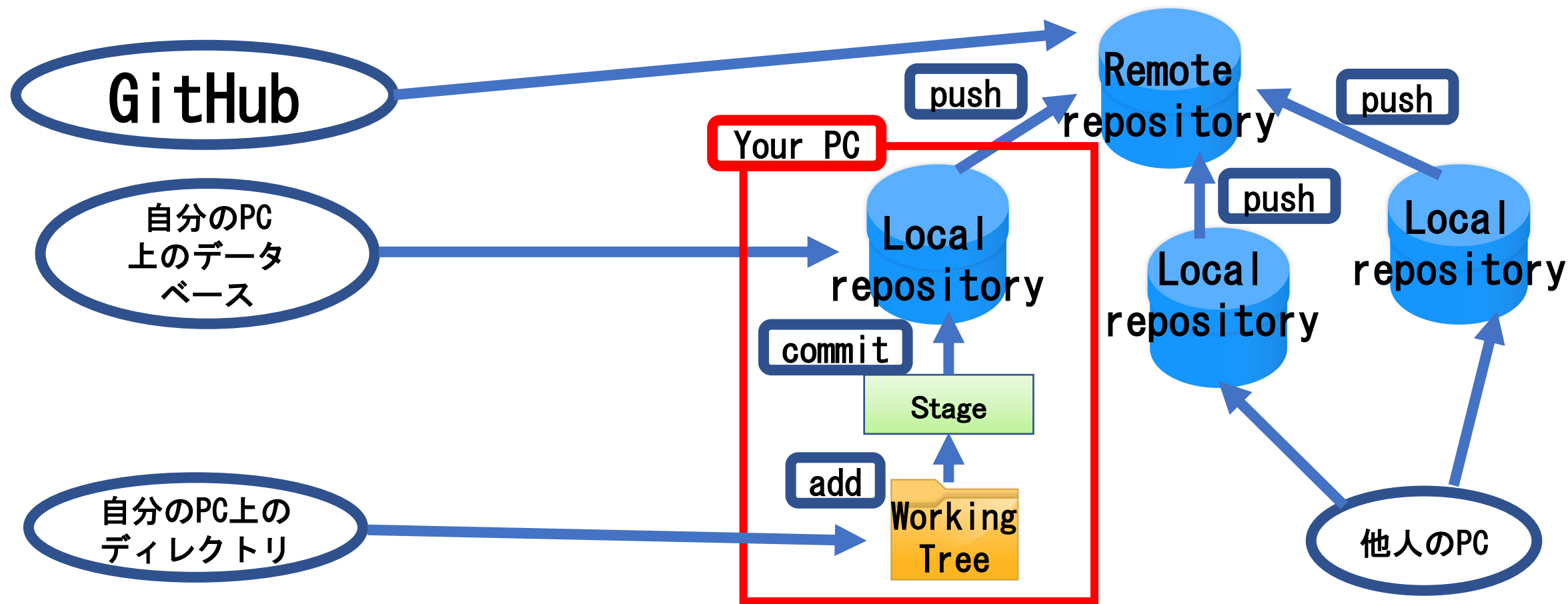


# gitの基礎



# gitとは何か

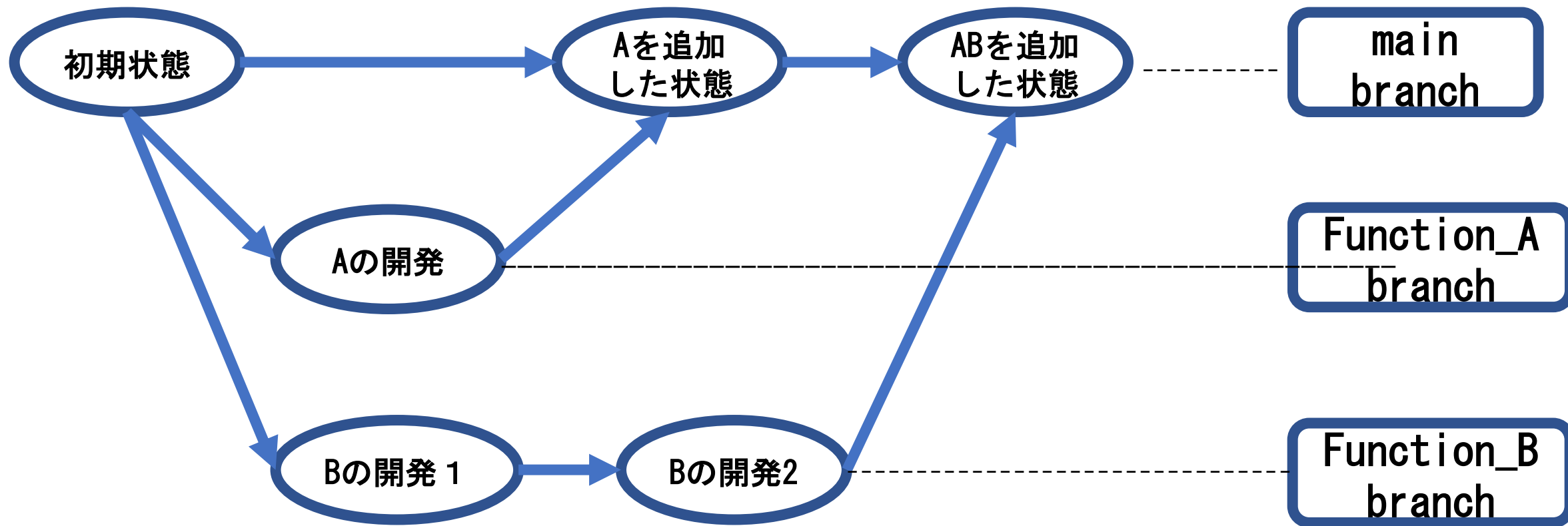
gitはCUI (ターミナル) で操作するオープンソースの分散バージョン管理システム  
リモートとローカルでファイルの変更履歴をそれぞれ管理する



# バージョン管理について

ファイルの追加、編集、削除等の編集履歴をbranchごとに記録する

- ・ デフォルトのブランチはmainのみ
- ・ mainを直接編集するのではなく、必ず新しいbranchを作成する



# gitのコマンド1 git init

```
$git init
```

実行した自分のPCの作業ディレクトリにgitで管理するローカルディレクトリを設定するコマンド(.gitというディレクトリを作る)

最初に実行するコマンド（後述するcloneの場合は実行しない）

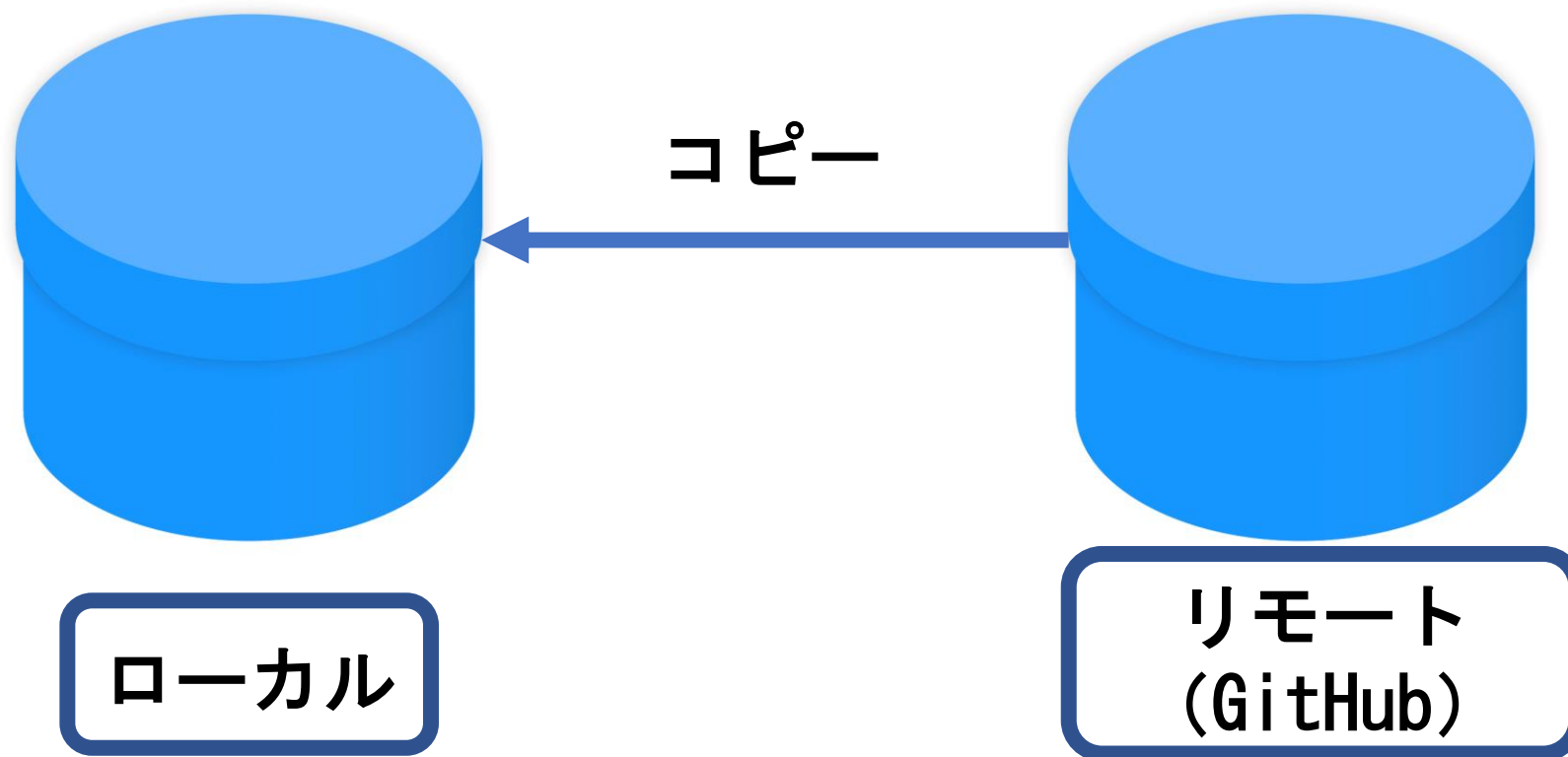


ローカル

# gitのコマンド2 git clone

`$git clone <リモートリポジトリのURL>`

リモートリポジトリの内容をローカルにコピーする（実行したディレクトリがローカルリポジトリになる）（1回だけ実行）



# gitのコマンド3 git branch, git checkout

\$git branch <新branch名>

branchの作成 (branchの切り替えは行わない)

\$git checkout <既存branch名>

branchの切り替え

\$git checkout -b <新branch名>

branchの作成と切り替えをする



# gitのコマンド4 git add , git commit

ファイルの変更をローカルリポジトリに登録するためのコマンド

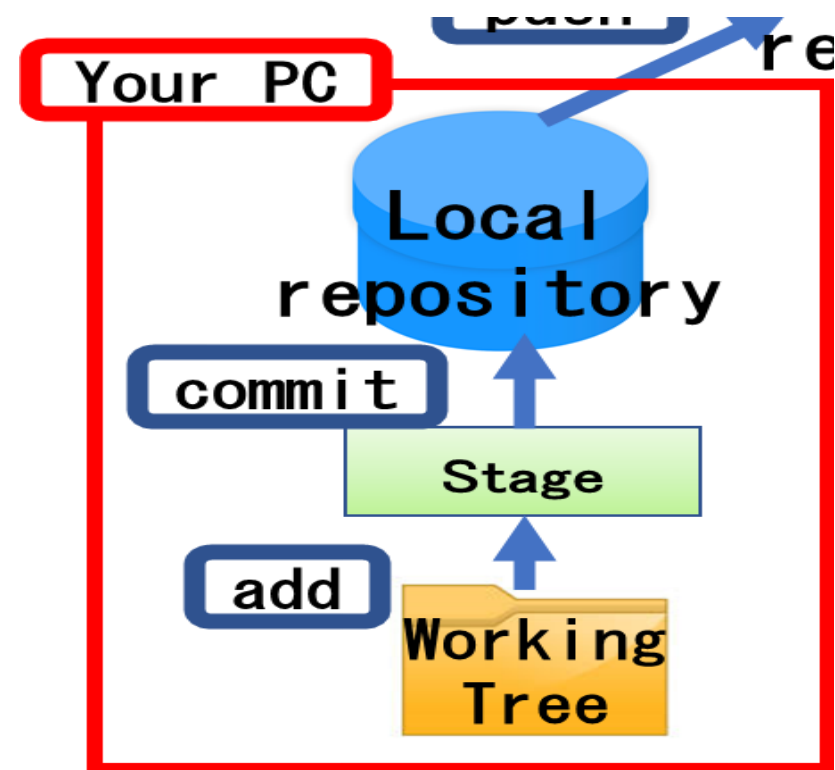
addでステージ（前段階）に上げて、commitでローカルリポジトリに登録される二段階での登録

```
$git add .
```

すべてのファイルをステージに上げる  
（.ではなくファイル名も指定できる）

```
$git commit -m “<メッセージ名>”
```

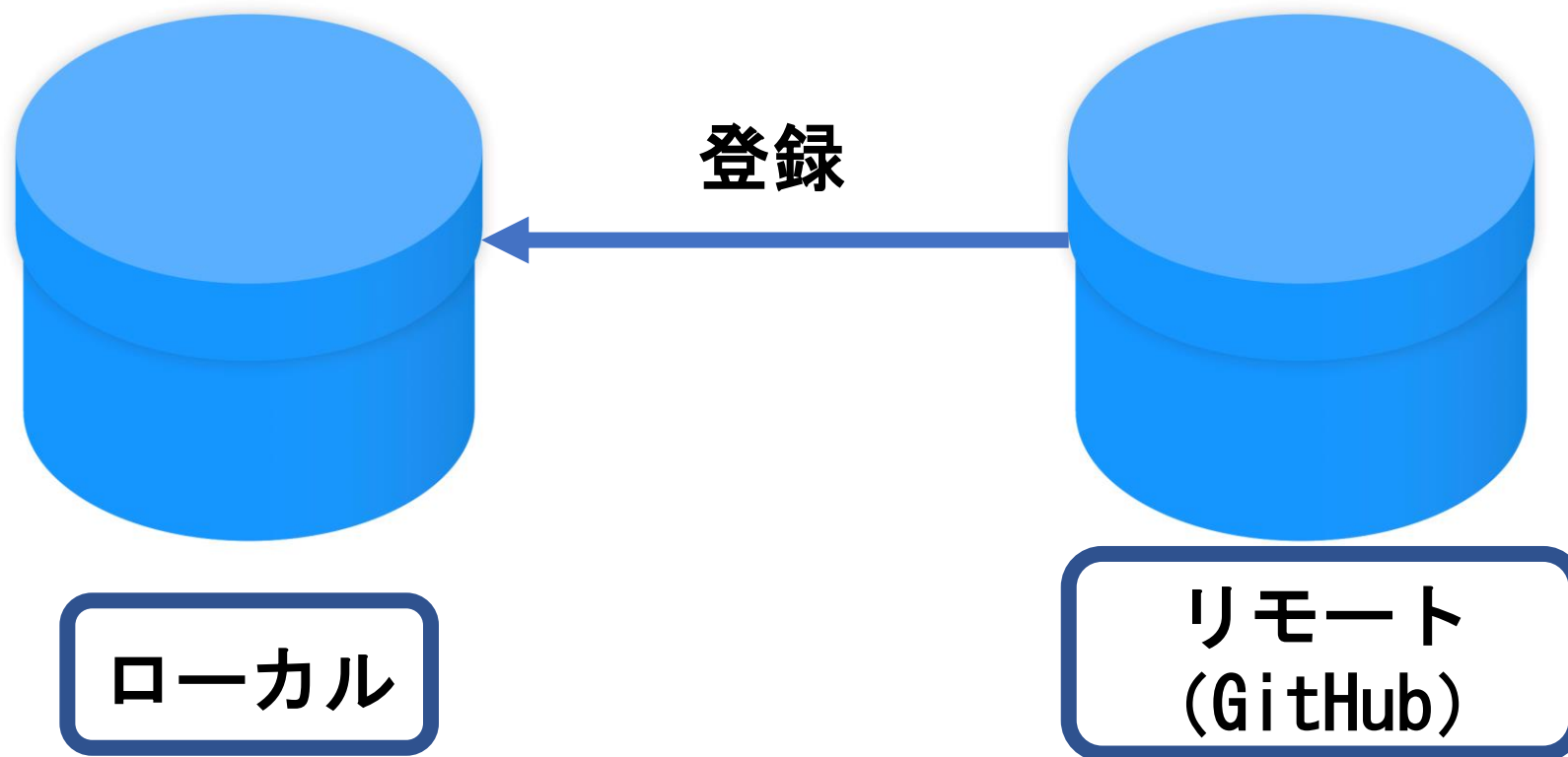
コミット時に何の変更をしたか書く



# gitのコマンド5 git remote add

`$git remote add <リモートリポジトリの名前> <GitHubのURL>`

作業したいGitHubのリモートリポジトリをローカルに登録する(1回だけ実行)(cloneした場合は不要)(リモートの名前はoriginにすることが多い)

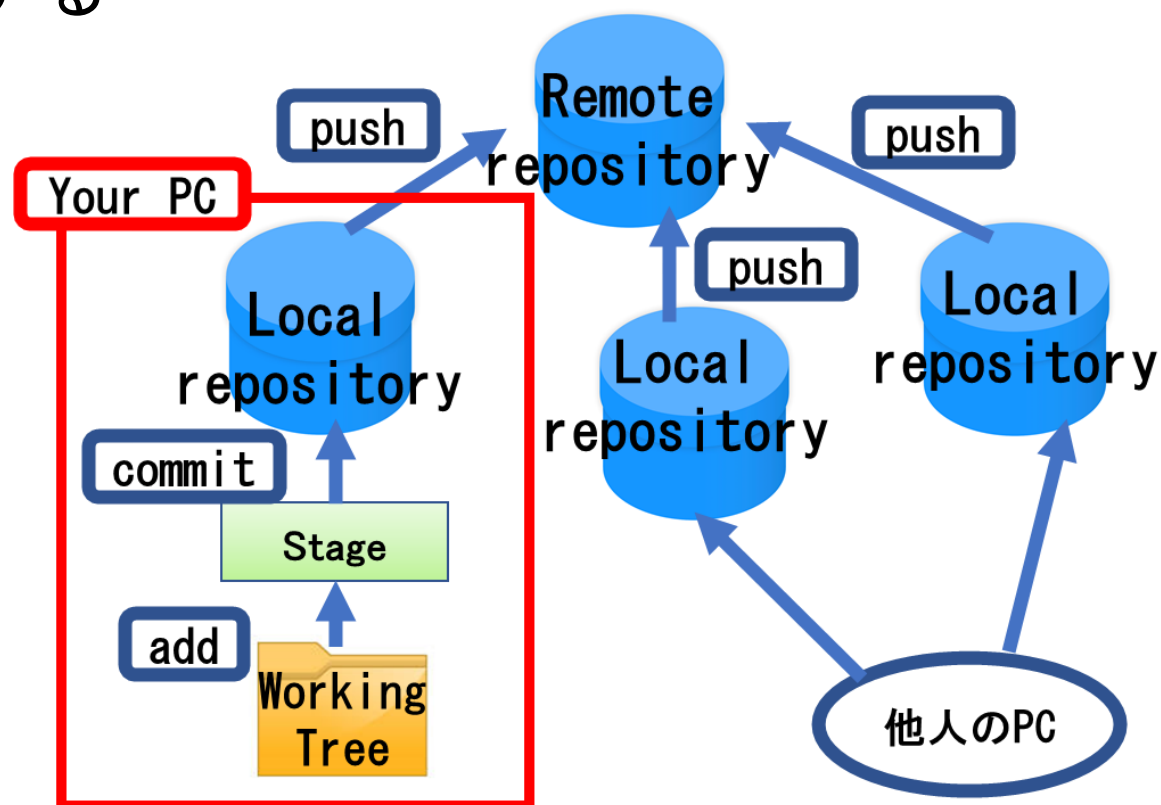




# gitのコマンド6 git push

\$git push <リモートリポジトリの名前> <ブランチの名前>

ローカルリポジトリの内容（1つのブランチ）をリモートリポジトリに登録する



# gitのコマンド7

## git fetch, git merge, git pull

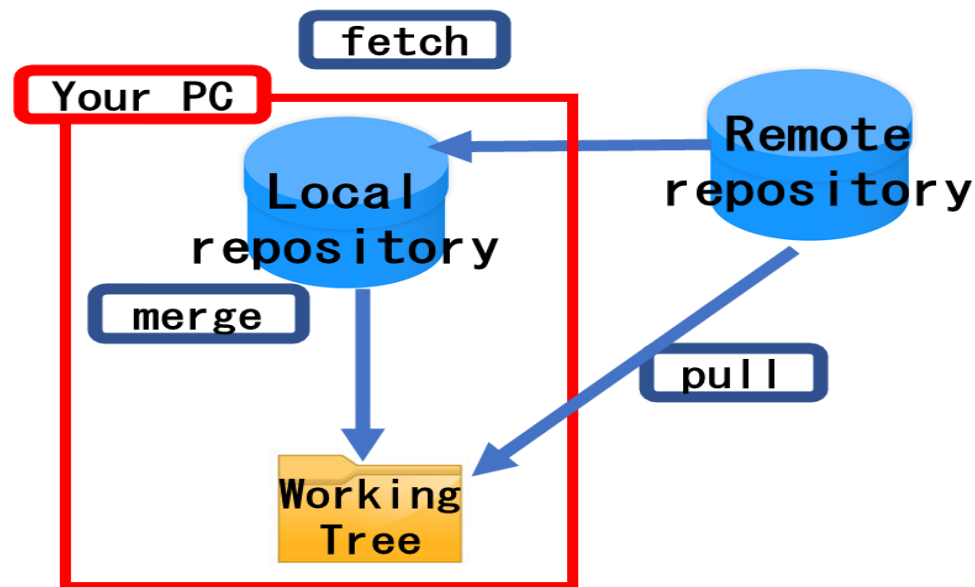
\$git fetch <リモート名>

リモートリポジトリの内容（1つのブランチ）  
をローカルリポジトリに登録する  
（pushの逆）

\$git merge <branch名>

ローカルリポジトリの内容を  
作業ディレクトリに反映する  
（add+commitの逆）

\$git pull <リモートリポジトリの名前> <ブランチの名前>  
fetchとmergeを同時に行う（add+commit+pushの逆）



# GitHubとは

gitを利用したソースコードのホスティングサービス（基本料金無料）

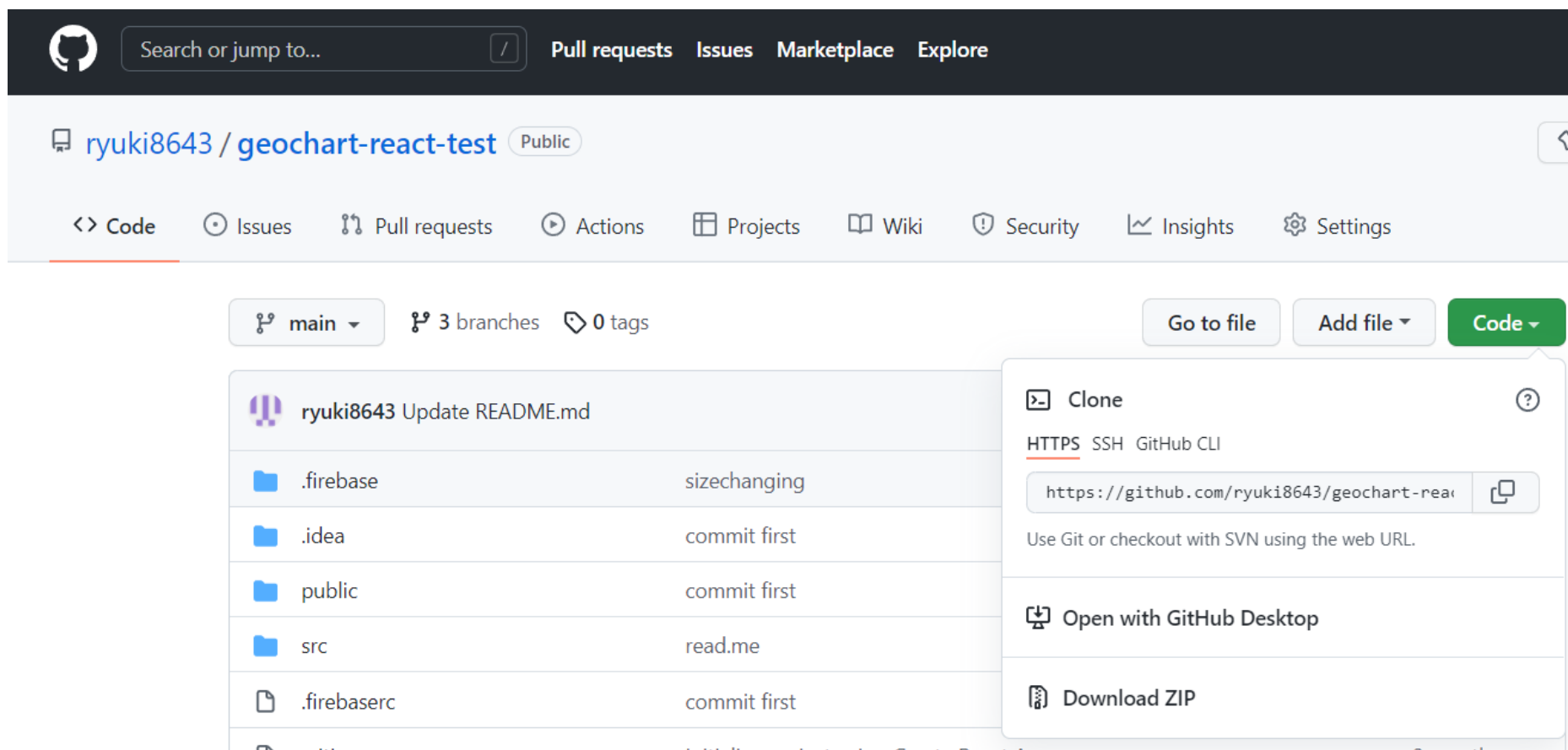
gitとは違いGUI（ブラウザ上）で操作する

gitにおけるリモートリポジトリの機能を果たし、GUI上でコードレビューやテスト、他人のソースコード検索など様々な追加機能を持ち使いやすい。

非公開のprivateリポジトリと公開のpublicリポジトリがあり、個人が複数のリポジトリを開設することが可能（1リポジトリ10GBまで）

# GitHub上での操作0 clone用のURL取得

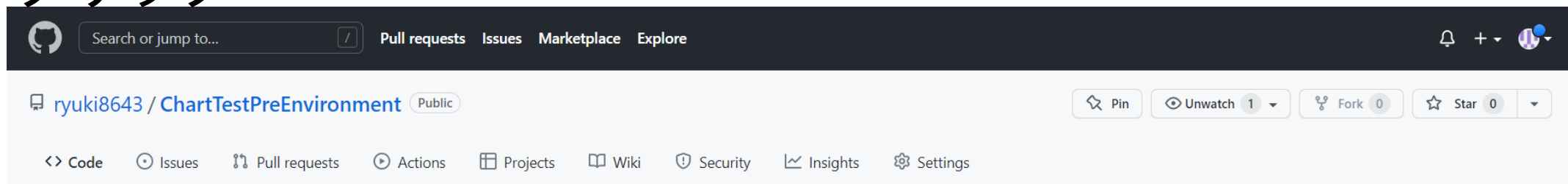
Codeボタンを押すとHTTPS、SSHでcloneするためのURLが出てくる



# GitHub上での操作 1 pull requests

pull requestsは変更を承認してもらうための作業

branchをpushするとcompare & pull requestのボタンが出るのでクリック



change had recent pushes less than a minute ago

Compare & pull request

change

2 branches

0 tags

Go to file

Add file

Code

This branch is 1 commit ahead, 1 commit behind main.

Contribute

ryuki8643 Update README.md

aa687f2 now 4 commits

.firebase

'changed'

last month

.idea

style-change

last month

public

Initialize project using Create React App

last month

src

'changed'

last month

firebase

style-change

last month

About



No description, website, or topics provided.

Readme

0 stars

1 watching

0 forks

Releases

No releases published

[Create a new release](#)

Packages

No packages published

# GitHub上での操作 1 pull requests

Create pull requestを押して特に問題なければ、able to mergeと出るので、他の人に変更を承認してもらう

The screenshot shows a GitHub pull request interface. At the top, the repository name is 'ryuki8643 / ChartTestPreEnvironment' (Public). The navigation bar includes 'Code', 'Issues', 'Pull requests' (1), 'Actions', 'Projects', 'Wiki', 'Security', 'Insights', and 'Settings'. The main title is 'Update README.md #2'. Below the title, it says 'ryuki8643 wants to merge 1 commit into main from change'. A comment from 'ryuki8643' states 'No description provided.' and is linked to a commit 'Update README.md' (Verified, aa687f2). On the right, there are sections for 'Reviewers' (No reviews), 'Assignees' (No one—assign yourself), 'Labels' (None yet), 'Projects' (None yet), and 'Milestone' (No milestone). At the bottom, a green box indicates 'Continuous integration has not been set up' and another green box states 'This branch has no conflicts with the base branch'. A 'Merge pull request' button is visible at the bottom left.

# GitHub上での操作 1 pull requests

Create pull requestを押して問題があるとconflictって出るのでResolve Conflictを押してなおしに行く

The screenshot shows a GitHub pull request for the repository `ryuki8643 / ChartTestPreEnvironment`. The pull request is titled "Update README.md #2" and is created by user `ryuki8643`. It shows 1 commit being merged from the `change` branch into the `main` branch. The pull request status is "Open".

Below the title, there are tabs for Conversation (0), Commits (1), Checks (0), and Files changed (1). The "Files changed" tab is selected, showing a diff for `README.md`. The diff shows a conflict in the `README.md` file, with a message: "This branch has conflicts that must be resolved". The message also states: "Use the [web editor](#) or the [command line](#) to resolve conflicts." and lists the conflicting files: `README.md`.

On the right side of the pull request, there are sections for Reviewers, Assignees, Labels, Projects, and Milestone. The "Reviewers" section shows "No reviews" and a button to "Convert to draft". The "Assignees" section shows "No one—assign yourself". The "Labels" section shows "None yet". The "Projects" section shows "None yet". The "Milestone" section shows "None yet".

# GitHub上での操作2 issue

IssueにはDevelopmentのボタンからcommitと紐づけられる機能がある。解決したらclose issueを押す

problem occur #3

Edit

New issue

Open

ryuki8643 opened this issue now · 0 comments



ryuki8643 commented now

Owner



conflict in branch



Write

Preview

H B I ≡ <> 🔗 ☰ ☷ ☑ @ ↻ ↶

Leave a comment

Attach files by dragging & dropping, selecting or pasting them.



Close issue

Comment

Remember, contributions to this repository should follow our [GitHub Community Guidelines](#).

Assignees

No one—assign yourself



Labels

None yet



Projects

None yet



Milestone

No milestone



Development



Link a pull request from this repository

Filter

✓ Update README.md  
ryuki8643/ChartTestPreEnvironment#2

'changed'  
ryuki8643/ChartTestPreEnvironment#1



# まとめ

1. gitはリモート (GitHub) とローカル (自分のPC) による分散バージョン管理システム
2. 最初にinitかcloneをする (init後にリモートを登録するのはremote)
3. 変更をリモートに反映するのはadd commit push
4. 変更をリモートから取得するのはpullまたはfetch+merge
5. branchの作成はbranch、切り替えはcheckout
6. GitHub上ではpull requests とissueを作る

# gitでの作業で気を付けること

1. mainのbranchでは絶対に作業せず、変更のたびbranchを作ること  
(人の作ったbranchで作業するときも同様)
2. branchは機能の追加ごとなど適切な頻度で作成すること
3. commitは一度に大量の変更に対してに行わず、作業の区切りごとに回数を分けて実行する
4. file名、commitメッセージ、branch名等は英語で、他人に分かりやすく、一意性のあるものにすること
5. Pull requestの承認は他人にしてもらうこと

# git、GitHubの最初にやる作業

```
$git config --global user.name "<GitHubのユーザー名>"
```

```
$git config --global user.email "<GitHubに登録したEmail>"
```

GitHubにssh鍵を登録（今日やんなくてもいいが、GitHubをPrivate（非公開）で開発するときとかに必要）

<https://www.sejuku.net/blog/74220>

<https://qiita.com/zakio0227/items/fcbad988c53f578ca3b2?msckid=bcd64a12c21e11ec8c2d4c043713db07>

# ハンズオン

1. 他人の作ったリポジトリにbranchを作り、変更 pull requestsを出してみる

<https://github.com/ryuki8643/git-handson>

2. 自分で新しく GitHub のリポジトリを作って最初のコミットを行う。