

Git / GitHub ガイダンス

久保拓巳

kubo@disc.lab.uec.ac.jp

2023 年 04 月 05 日

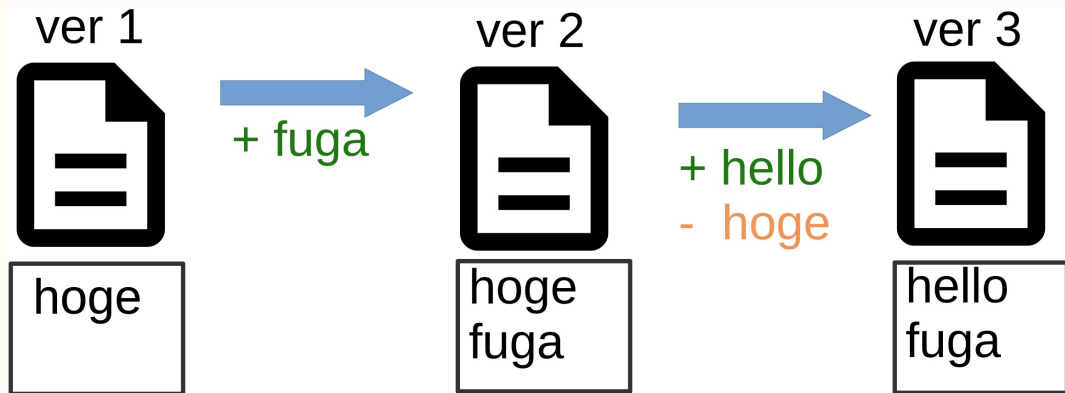
目次

- ◆ Git / GitHub とは
- ◆ 使う準備

Git / GitHub とは

Git とは

- ◆ **Git** とはバージョン管理システムの一つ
- ◆ ファイルの変更内容を追跡する
 - ▶ ファイルの差分確認・復元などが自由にできる

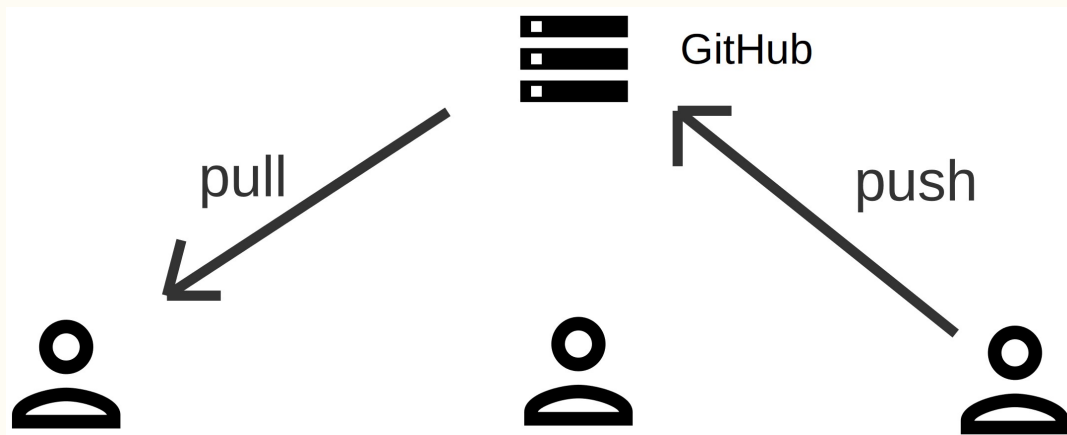


こんなときに役立つ

- ◆ うっかりバグが混入してしまった
 - ▶ → もとの状態に戻せる
- ◆ いつ / どこを変更したかわからない
 - ▶ → 差分を管理しやすい

GitHub とは

- ◆ **GitHub** とは Git のホスティングサービス
 - ▶ Git で管理している内容を同期させる
 - ▶ 複数人でファイルを共有できる
- ◆ ユーザがコードを変更, Git で追跡, GitHub にアップロードという手順を踏む



实践

覚えるべき用語

- ◆ **commit** : 変更を追跡する
 - ▶ version を 1 つ上げるみたいなもの
- ◆ **push** : 変更分を GitHub にアップロードする
- ◆ **pull** : 変更分をダウンロードする
- ◆ **リポジトリ** : Git で追跡する 1 つのフォルダ単位のこと
- ◆ **branch** : 並行作業できるファイル, ある branch で作業 / push をしても他の branch には影響を与えない

使う

以下のような方法がある

- ◆ コマンドラインから

- ▶ `git commit...`などを打ち込んで直接操作する

- ◆ GUI から

- ◆ GitHub 上で直接操作する

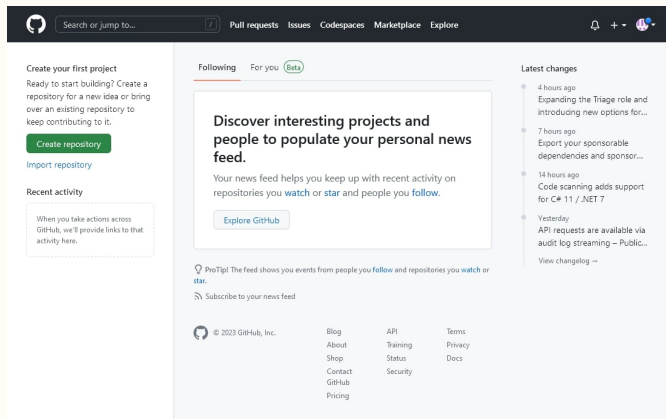
- ▶ これはあくまで Push などの操作のみで、ローカルに対して変更はできない

以下では GUI を用いて, GitHub アカウントの作成からファイルの push までを行う

アカウントを作る

- ◆ <https://github.com/> にアクセス
- ◆ Sign Up を選択して email / password / username の登録 , 認証を行う
- ◆ アンケートの回答 / プランの選択などをする
 - ▶ Free プランでよい（実は：学割が利用できる）

この画面になったらアカウントの作成に成功している



toda-lab への参加

- ◆ GitHub で toda-lab に参加する

GUI アプリケーション："Github Desktop"の用意

- ◆ <https://desktop.github.com/> にアクセス
- ◆ ダウンロードなどを行う
- ◆ Github のアカウントにログインする

リポジトリの clone

- ◆ progress というリポジトリが存在するので自分の PC にコピーする
 - ▶ この操作を **clone** とよび、普通はそのリポジトリに対して最初だけ行う
- ◆ GitHub Desktop で `Clone a repository from the Internet...` を選択
- ◆ 適当なフォルダを選び、そこに clone する

Branch の作成

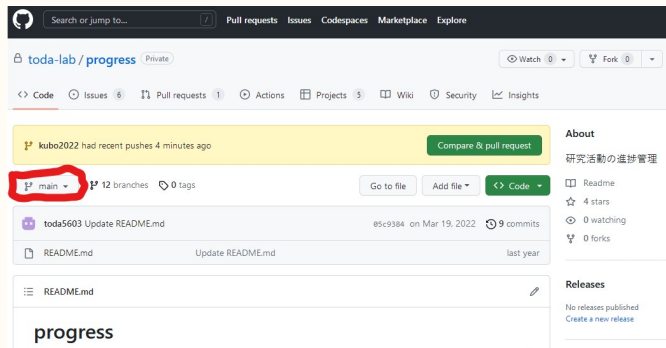
- ◆ メニューの `Branch > New branch` を選択
- ◆ Name を決める（ここでは `Name2023`：例えば `kubo2023` を推奨）

Branch の切り替え， ファイルの作成

- ◆ 上部の `Current Branch` を自分のものに切り替える
- ◆ ファイルを作成する
- ◆ 左下の狭い欄に `add test.txt` などと書く
- ◆ `Commit to {kubo2023}` を実行
 - ▶ これで `commit` ができた
- ◆ 上部の `Push Origin` を実行
 - ▶ これで `push` ができた

Github で確認

- ◆ <https://github.com/toda-lab/progress> にアクセス
- ◆ 左上の方の `main` を選択, 自分の branch に切り替える
 - ▶ 黄色い枠は無視してよい
 - ▶ これを実行すると, 他の branch に自分の変更を取り込むようにお願いする
- ◆ さっき作ったファイルがあれば OK



参考文献

- ◆ 『わかばちゃんと学ぶ Git 使い方入門』, 湊川 あい, シーアンドアール研究所
 - ▶ 漫画で分かりやすい
- ◆ 『GitHub 実践入門 ~Pull Request による開発の変革』, 大塚 弘記, 技術評論社
 - ▶ 解説が丁寧, ちょっと古いのでメニューが一部異なる
- ◆ サル先生の Git 入門~バージョン管理を使いこなそう ,<https://backlog.com/ja/git-tutorial/>
 - ▶ 解説が丁寧

GitHub はとても便利なので入門書などに目を通しておくとよいかも

最後に

- ◆ 戸田研究室では進捗状況の共有などに GitHub を利用している
 - ▶ 運用の仕方などは資料を確認するとよい
- ◆ 質問はいつでも受け付けています