

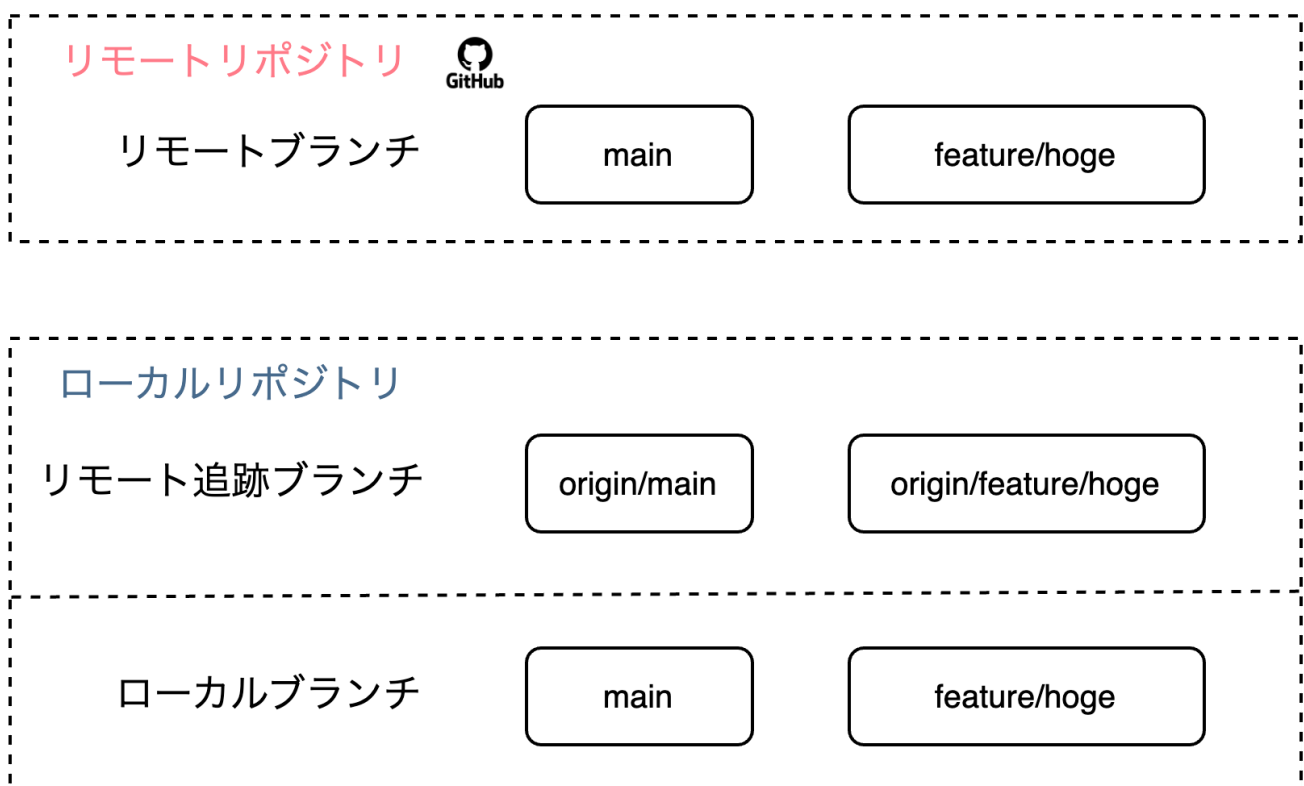
今までなんとなく、**fetchしてmergeするのを一気にやってくれるのがpull**という理解で、詳しく調べたことがありませんでした。改めて調べてみたのでまとめます。

## 前提知識

**fetch, merge, pull**を理解する上で欠かせないのが以下の言葉です。

- **リモートリポジトリ**: GitHubなどリモート上に存在するリポジトリ
- **ローカルリポジトリ**: PC上に存在するリポジトリ
  - **リモート追跡ブランチ**: **origin/main**など、リモートリポジトリを追跡するブランチ
  - **ローカルブランチ**: **main**など、ローカル上で実際に扱うブランチ

図にすると以下ようになります。

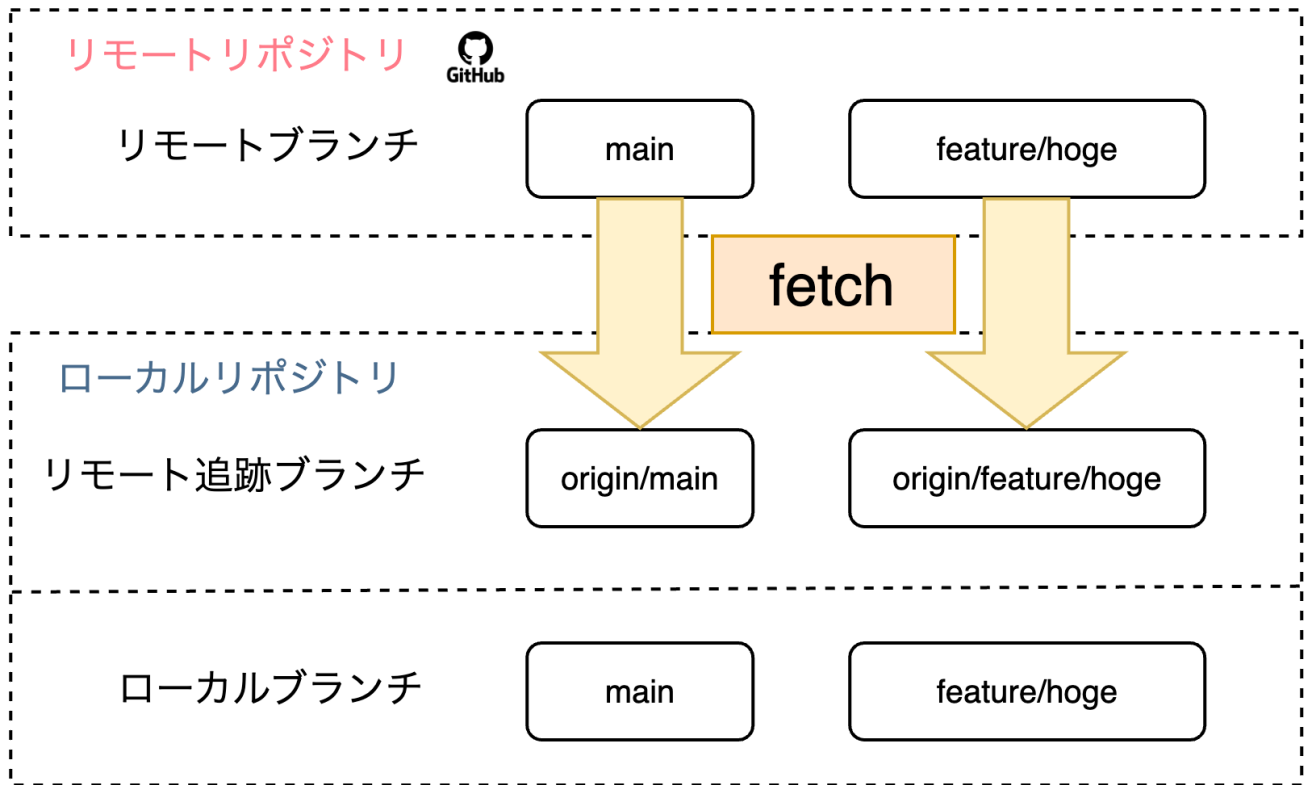


すこしわかりにくいのが**リモート追跡ブランチ**ですね。これは**リモートブランチの状態をとりあえずローカルに持ってきている場所**というような理解であっていると思います。ここで大事なのは、リモート追跡ブランチはあくまで**ローカルにあるブランチ**であることです。

## fetch

**fetch**とは、リモートリポジトリのブランチの状態を**リモート追跡ブランチ**に反映させる操作です。リモートの最新の状態をローカルでも見れるようにするイメージです。Gitはリモートリポジトリとリアルタイムで同期するわけではないので、最新の状態を見るためには**毎回git fetchをする必要があります**。

**fetch**をしただけではローカルブランチ(**main**など)の状態は何も変わりません。



## merge

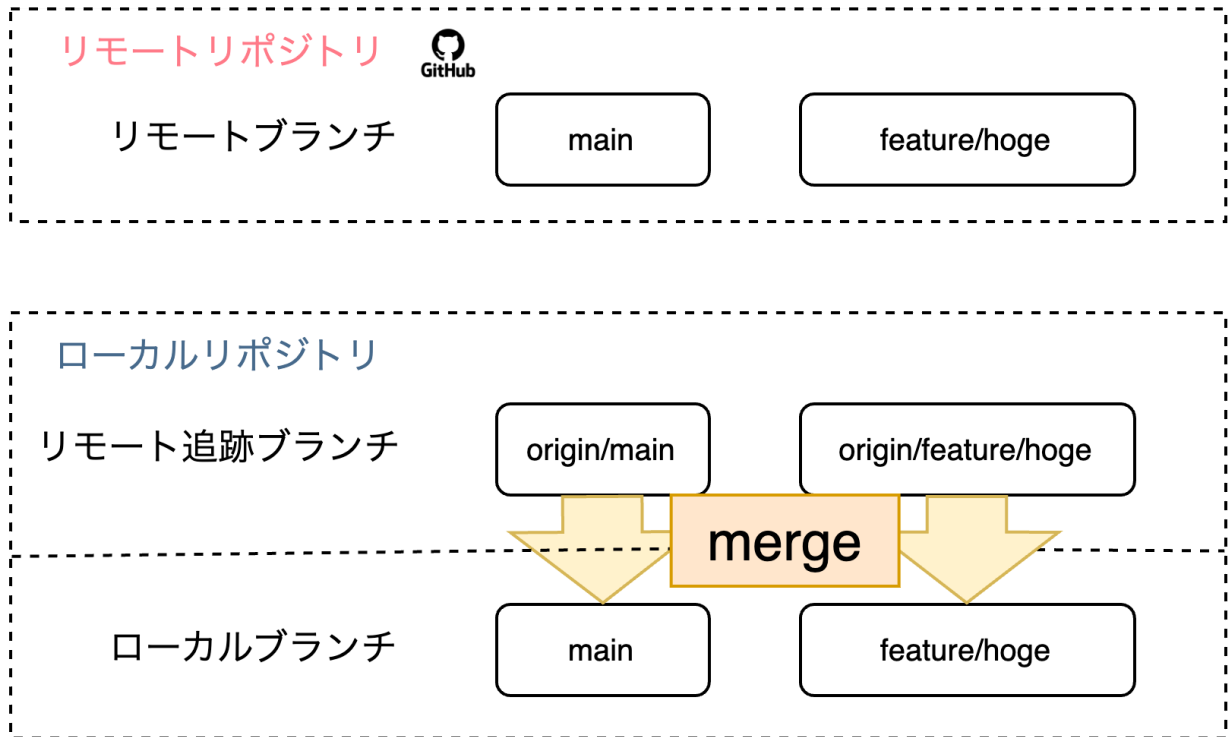
---

`merge`は異なるブランチの変更を自分のブランチに取り込む操作で、Pull requestを作ってマージする、あれと同じです。リモート追跡ブランチとの関係に限って説明すると、**リモート追跡ブランチをローカルブランチに取り込む**操作です。

取り込み方で以下の2つの方法があります。

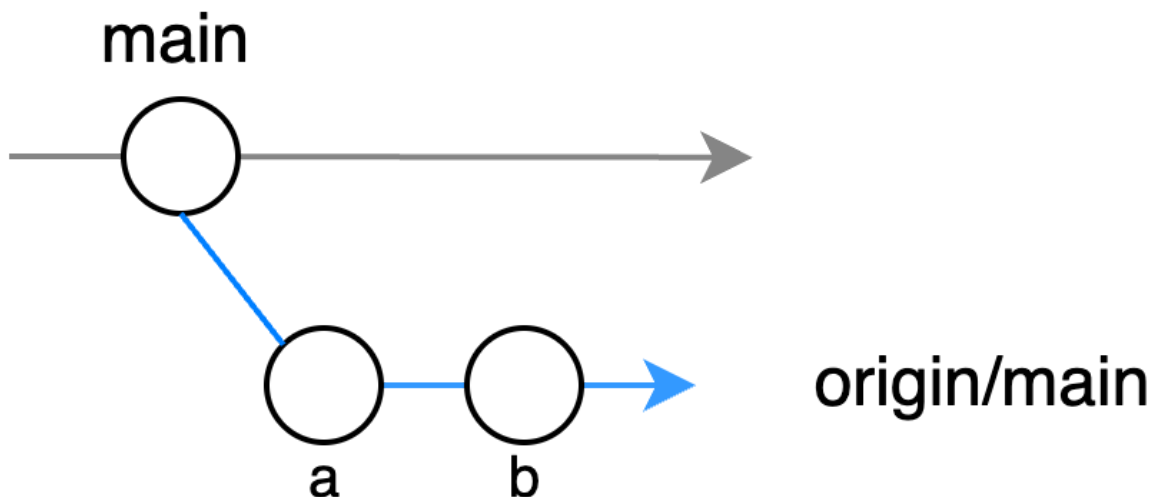
- `fast-forward`

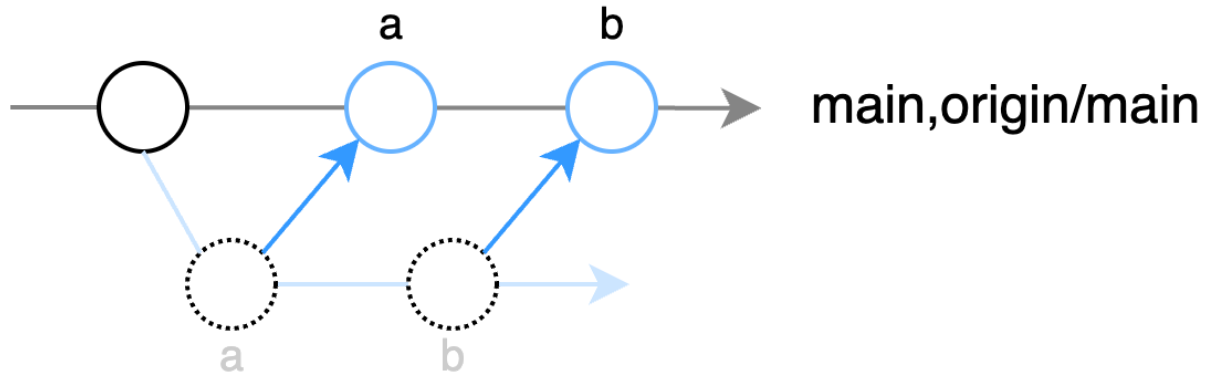
- non-fast-forward



## fast-forwardマージ

リモート追跡ブランチの状態をローカルブランチに取り込む場合、通常はfast-forward(マージコミットを作成しない)でマージします。図示するとこのような感じです。



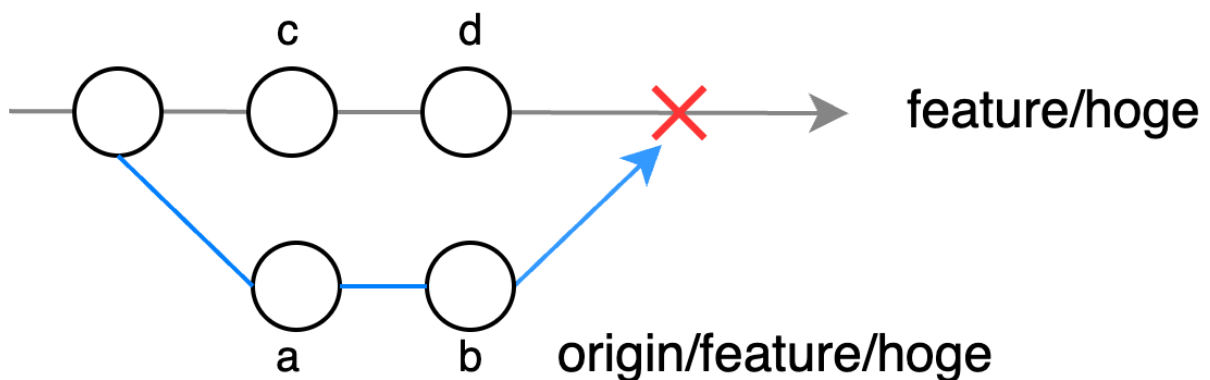


`origin/main`の方がaとbというコミット分、ローカルの`main`よりも進んでいるとき、`fast-forward`マージでは`main`にaとbのコミットを取り込みます。ここでマージコミットは作らないので、`main`ブランチは単にaとbのコミットを積み上げたようになります。

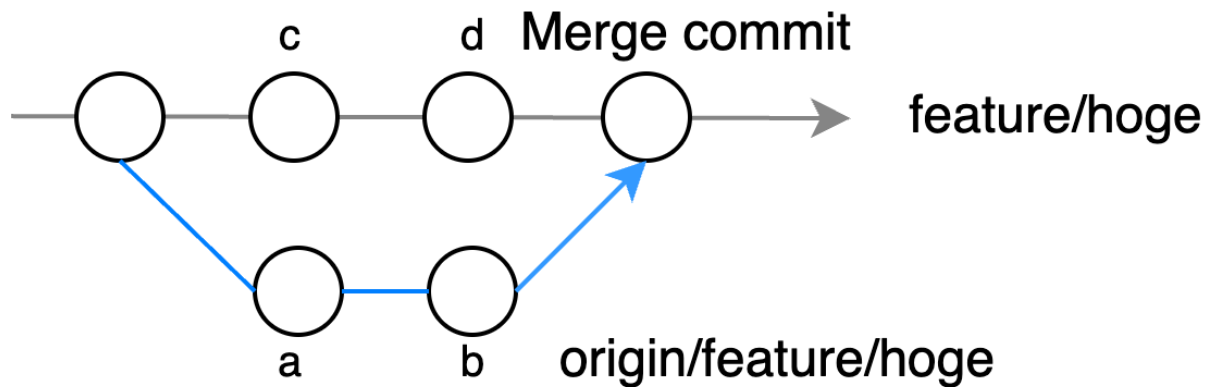
## non-fast-forwardマージ

`main`ブランチは基本的にローカルで変更を加えることがないと思うので、`fast-forward`でマージできます。

しかし`feature`ブランチなどの作業ブランチで、複数人で一つのブランチを共有して開発などしている場合、`origin/feature/hoge`と`feature/hoge`で異なる履歴(コミット)が積み上がってしまうことがあります。この場合、コンフリクトが発生しなくとも、`fast-forward`でマージすることはできません。



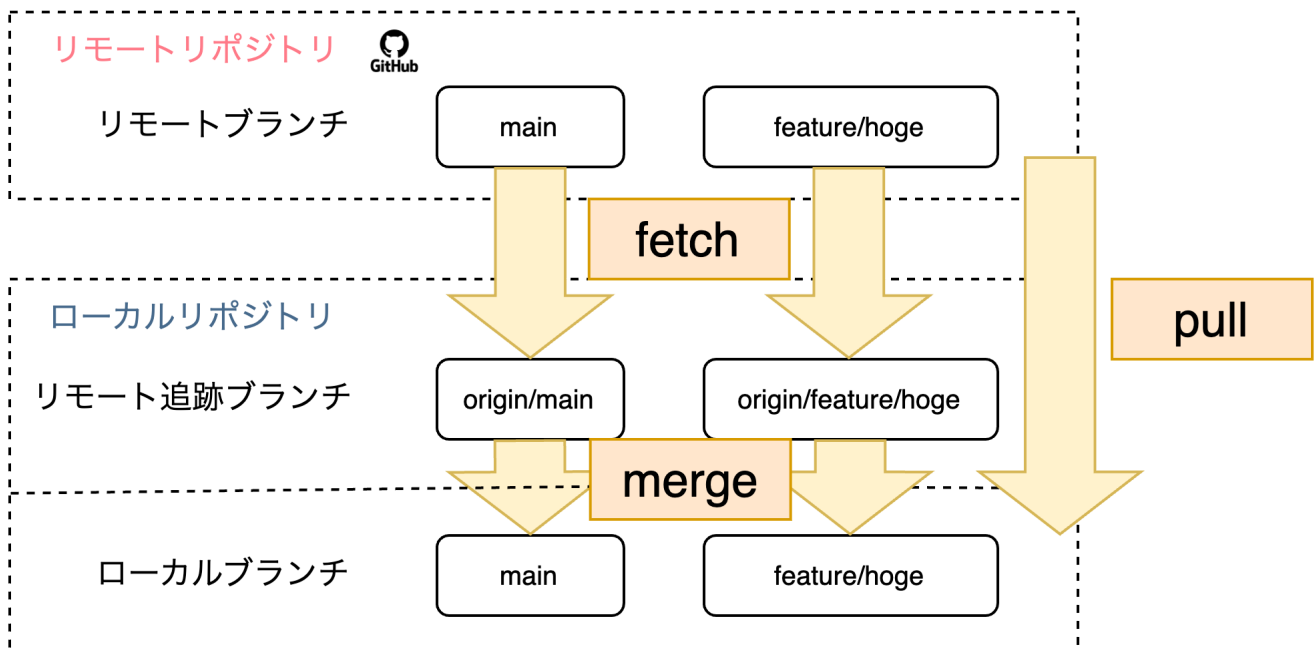
そこで**マージコミット**を作ってマージすることができます。



このように、**マージコミット**を作ってマージするのが**non-fast-forward**マージです。

## pull

**pull**は冒頭で言ったように**fetch + merge**です。



もう少し詳しく見ると、**merge**の方法でいくつか挙動が異なります。

- 通常(特に何も設定していない場合)
  - **fast-forward**マージを試す
  - できなければ**non-fast-forward**マージをする
- **--ff-only**オプションを付けた場合
  - **fast-forward**マージを試す
  - できなければマージしないでエラー
- **--no-ff**オプションを付けた場合
  - **non-fast-forward**マージでマージする

# おわりに

---

改めてGitでよく使う`fetch`, `merge`, `pull`についてまとめました。

私はGit初心者だった頃、`git pull`したときにマージコンフリクトが発生してマージしようとし始めたら`pull`するのをやめて`reset --hard`で強制移動するなどしていたのですが、今回調べてみて、`--ff-only`オプションを付けていれば良かったんだなと学びがありました。(今はそもそも`pull`をほとんど使わないのですが...)

特に`merge`部分の違いについて理解できていると、`pull`したときの挙動で焦らないで済むと思います。