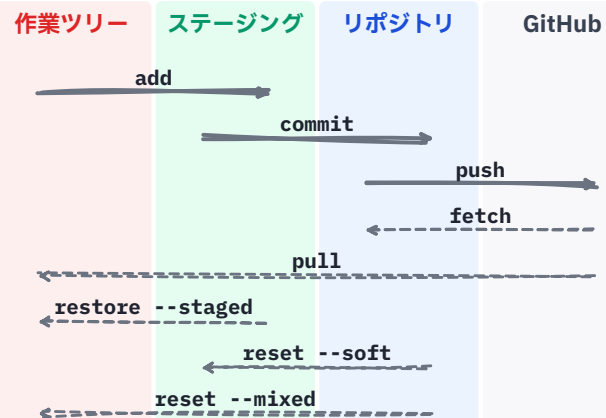




## &gt; コマンドの地図



## &gt; はじめる

```
git init このディレクトリで Git を始める
git clone <url> GitHub のリポジトリを手元に写す
git config user.name "<name>"
コミットに残す名前
```

## &gt; 管理しないファイル

**.gitignore に 1 行ずつ書く**

|               |            |
|---------------|------------|
| *.log         | ログファイル     |
| node_modules/ | ディレクトリごと   |
| .env          | パスワードなどの設定 |

## &gt; 記録する

```
git status いまの状態を見る
git add <file> ファイルをステージする
git add . 変更をまとめてステージする
git add -p 選びながらステージする
git commit -m "<message>" コミットする
```

## &gt; 変更と履歴を見る

```
git diff まだステージしていない変更
git diff --staged ステージした変更
git log --oneline コミットの履歴を 1 行ずつ
git show <commit> そのコミットの中身
```

## &gt; 変更を捨てる・ステージを外す

```
git restore <file> 編集を捨てる ⚠
git restore --staged <file>
ステージから外す (編集は残る)
```

## &gt; コミットを直す・取り消す

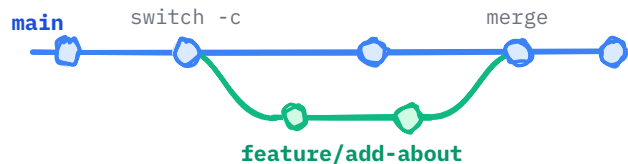
```
git commit --amend -m "<message>"
直前のコミットを直す
git reset --soft HEAD~1
直前のコミットを取り消す (変更は残る)
git reset --hard HEAD~1
取り消して、変更も捨てる ⚠
git revert <commit>
打ち消すコミットを足す (push 後はこれ)
git reflog HEAD の動き (消えたコミットに戻れる)
```

## &gt; GitHub と同期する

```
git remote add origin <url>
GitHub のリポジトリを登録する
git push -u origin main 初めて送る
git push 送る
git pull 取り込む
```



## &gt; ブランチ



**git branch** ブランチの一覧 (\* がいま)

**git switch <branch>** ブランチを移る

**git switch -c <branch>** ブランチを作って移る

**git log --oneline --graph --all**  
枝分かれを図で見る

**git branch -d <branch>**

マージ済みのブランチを消す

**git branch -D <branch>** マージしていなくても消す



## &gt; ブランチを取り込む

前: ブランチを作ったあと、main が進んだ



後: git rebase main で、main の先に付け直す



**git merge <branch>** いまのブランチにマージする

**git merge --squash <branch>**

1つのコミットにまとめてマージする

**git rebase <branch>** そのブランチの先に付け直す

## &gt; コンフリクトを直す

```
<<<<<< HEAD
color: blue;
=====
color: green;
>>>>>> feature/update-style
```

いまのブランチ (main)

取り込む側

片方に直して印を消す → git add → git commit

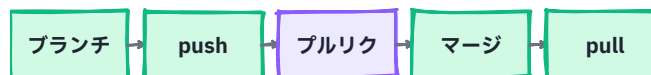
**git rebase --continue**

(リベース中) 直したら続ける

**git merge --abort** マージをやめて元に戻す

**git rebase --abort** リベースをやめて元に戻す

## &gt; チームで進める



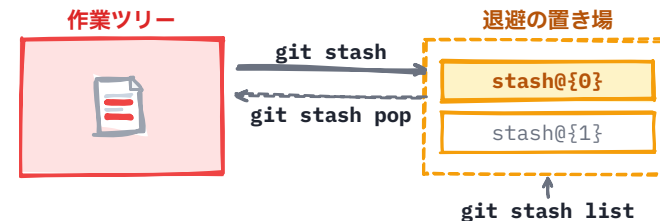
**git push -u origin <branch>**

ブランチを送る (プルリクエストの前)

**git fetch** 取ってくるだけ (取り込まない)

**git tag <name>** いまのコミットに名前を付ける

## &gt; 作業を一時退避する



新しく退避したものが stash@{0} (一番上)

**git stash** 変更を退避する

**git stash pop** 退避した変更を戻す

**git stash list** 退避したものの一覧

## &gt; 履歴を調べる・直す

**git log -- <file>** そのファイルを変えたコミット

**git log --oneline -S <text>**

その文字列を足した・消したコミット

**git blame <file>** 各行を最後に変えた人



そのコミットだけを写して、いまのブランチに足す (元は残る)

**git cherry-pick <commit>**

そのコミットだけ取り込む

**git rebase -i HEAD~<n>**

直近のコミットを並べ直す・まとめる

