基本の流れ（超基本）

1. リモートリポからクローンしてローカルリポを作成する（git clone）
↓
2. 作業用のブランチを切る（git branch）
↓
3. 作業内容をStaging Areaに上げる（git add）
↓
4. Staging Areaの内容をリポジトリにコミットする（git commit）
↓
5. ローカルリポの内容をリモートリポに反映させる（git push）
↓
6. pushしたブランチをmainにマージしてもらうよう依頼: pull request （GitHub上）
↓
7. ローカルリポのmainブランチをリモートリポからpullして更新（git pull）
↓
8. 不要になったブランチを削除する（git branch -d）

前提条件

- gitユーザ名、メールアドレス設定済み（~/.gitconfig）
- githubとのSSH接続設定済み
- リモートリポジトリ作成済み

ワークフロー手順

1. リモートリポからクローンしてローカルリポを作成する

①GitHubにアクセスし、対象リポジトリの **<>Codeボタン** から SSH のURLをコピー
②クローンを作成したいディレクトリに移動してクローン

```
$ cd /home/user/test
$ git clone git@github.com:Kensuke-c7/test-repo.git
Cloning into 'test-repo'...
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (3/3), done.
```

③ローカルにクローンしたリポジトリの中身を確認

```
$ ls -a test-repo/
.  ..  .git  sample.txt
```

.git/ はGitの設定ファイルフォルダ。このフォルダがあるためgitコマンドが有効になる。

③リモートリポジトリとの紐づけを確認する。

```
$ git remote -v
origin  git@github.com:hogehoge/tutorial.git (fetch)
origin  git@github.com:hogehoge/tutorial.git (push)
```

2. 業用のブランチを切る（git branch）

作業用のブランチ（update-readmne）を作成する。
・**更新は必ずブランチ側で行う。mainブランチでの作業はあり得ない**
・ブランチ作成はローカル側で行う（Master側で不要なブランチを発生させないため）

①現在のブランチを確認

```
# git branch
* main
```

②新しいブランチを作成

```
# git branch update-readme
# git branch
* main
  update-readme
```

③作業するブランチを切り替える

```
# git checkout update-readme
Switched to branch 'update-readme'

# git branch
  main
* update-readme    ← アクティブなブランチが切り替わった
```

3. 作業内容をStaging Areaに上げる（git add）

コミットの流れ

対象ファイルを更新 -> 更新分をStaging areaにadd -> リモートリポにcommit

①Git作業状態を確認する: git status

```
# git status
On branch update-readme
nothing to commit, working tree clean  ←更新なし
```

リモートリポジトリと紐付いている場合、リモートリポジトリとの時間的な差分（ahead, behind, up to date）が表示される。

②ファイルを更新

```
# vim sample.txt
Hello,world!!
Hello,Tech-Blue!!  ← 末尾に文言追加
```

③Stating Areaに上げる: git add <filename>

```
$ git add sample.txt
```

・カレントディレクトリ配下のすべてのファイル・フォルダを再帰的に一括でStaging Areaに追加する: git add .

(補足) 本当は追加したくない変更をgit addしてしまった場合に元に戻す

テージングエリア上にある変更を作業ディレクトリへ戻す、取り消す。

```
$ git restore --staged <file>
```

④作業状態を再度確認

```
$ git status
ブランチ update-readme
コミット予定の変更点:
  (use "git restore --staged <file>..." to unstage)
        modified:   sample.txt
```

4. Staging Areaの内容をリポジトリにコミットする（git commit）

コミット時には必ずコミットメッセージをつける: git commit -m “<commit message>”
・コミットのコメント必須
・基本的にコメントは１文

```
$ git commit -m "update sample.txt"
[update-readme 366c42b] update sample.txt
 1 file changed, 1 insertion(+)
```

```
$
$ git status
ブランチ update-readme
nothing to commit, working tree clean
```

今いるブランチにコミットポイントが作成される。

5. ローカルリポの内容をリモートリポに反映させる（git push）

①push前にリモートリポから一旦pullする
※他の人がリモートリポに変更を加えている可能性があるため

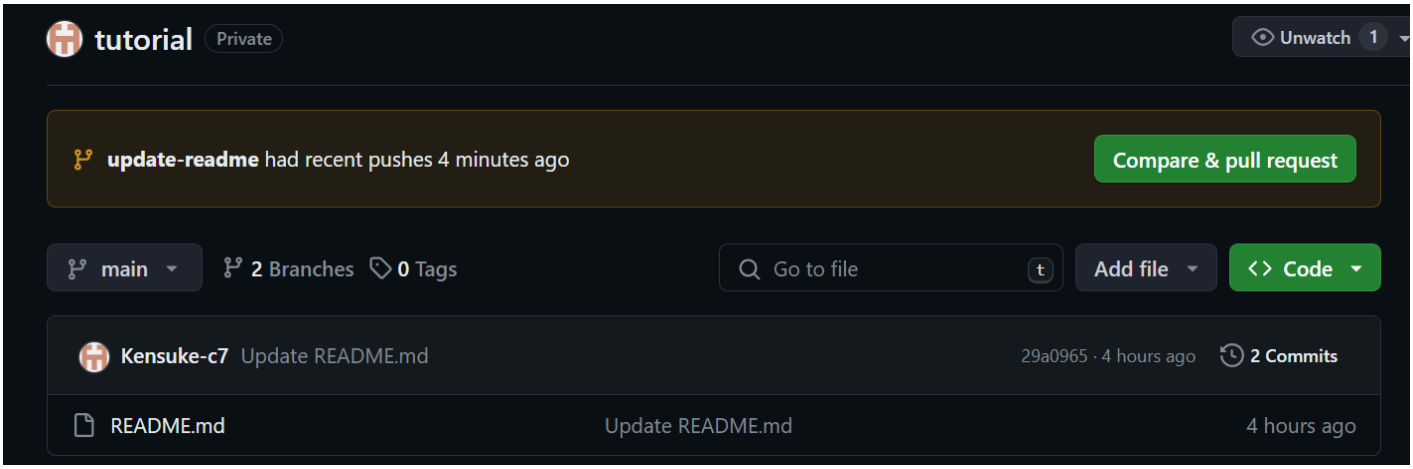
```
# pullを反映させるブランチ（ローカル）に現在いることを確認
$ git branch

main
* update-readme

# pullしてくるリモートリポジトリを指定する
# 指定のため、紐付いているリモートリポジトリを確認
$ git remote -v

origin  git@github.com:atsuyamaru/sample-repo.git (fetch) # pull先
origin  git@github.com:atsuyamaru/sample-repo.git (push) # push先

# git pull <remote_ref> <branchname>（メインブランチを指定）
$ git pull origin main

From github.com:atsuyamaru/sample-repo
 * branch            main       -> FETCH_HEAD
# FETch_HEADのmainブランチからbranch(Wordking Directory)にpull
Already up to date.

# git push
$ git push origin update-readme # update-readmeブランチを指定。 ※mainではない

Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Writing objects: 100% (3/3), 263 bytes | 263.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
remote:
remote: Create a pull request for 'update-readme' on GitHub by visiting:
remote:      https://github.com/atsuyamaru/sample-repo/pull/new/update-readme
remote:
To github.com:atsuyamaru/sample-repo.git
 * [new branch]      update-readme -> update-readme
```

・pushするときは、mainを指定するのではなく、リモートリポジトリのupdate-readmeブランチにpushする。
・update-readmeブランチにpushした後、mainにプルリクエストを依頼する。

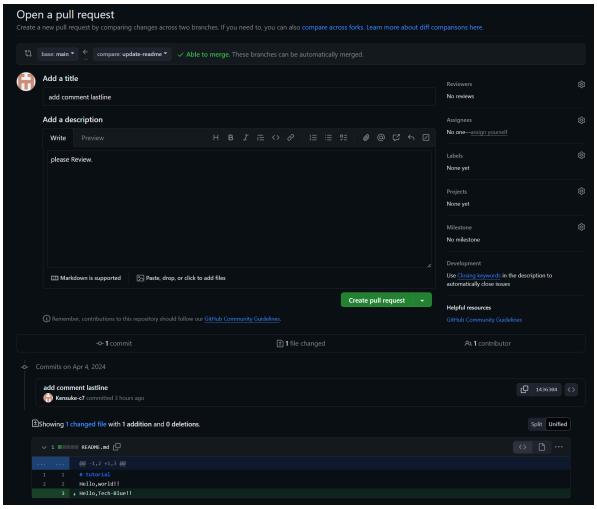
6. pushしたブランチをmainにマージしてもらうよう依頼: pull request（GitHub上）

基本的にGitでの開発は複数人のメンバー（チーム）で行うため自由に自分の更新分をマージしてはいけない（することはできない）。

①GitGubのmainブランチで「Compare & Pull requests」をクリック



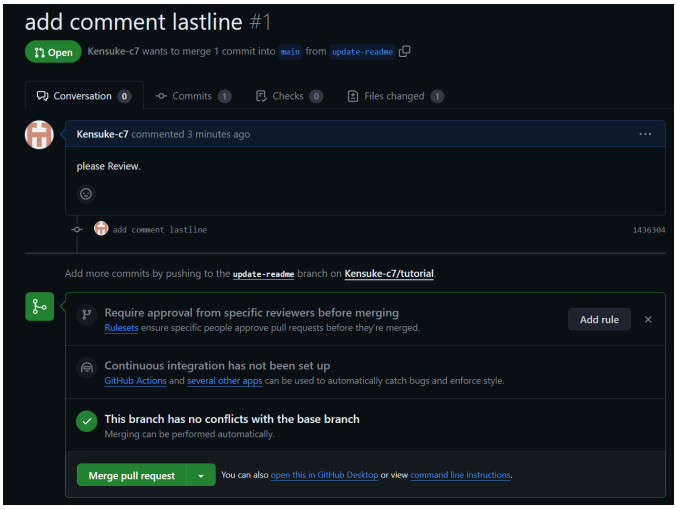
②画面遷移後、pull request作成画面が表示される。どのブランチからどのブランチにマージするかを、update-readme から main となるよう設定すると変更箇所の差分が表示される。更新内容(差分)に問題なければ画面右横の「Create pull request」をクリックする



さらに画面遷移後、リクエストのコメントを書くことができるので適当に「Please review.」を書いて「Create pull request」をクリック。

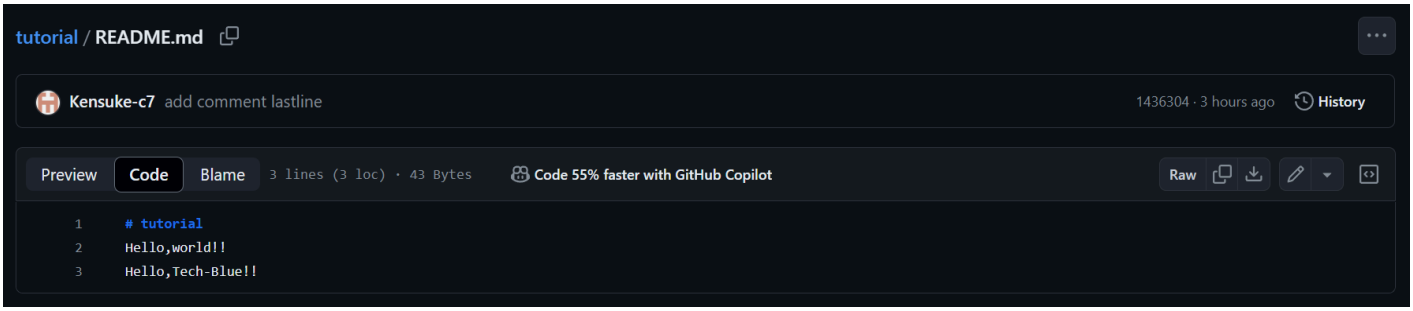
③pull requestをマージする

内容に問題なければ（チェック者が）プルリクエストをマージする



④mainブランチのマージを確認する

mainブランチのREADME.mdが更新されていることが確認する



・3行目が追加されている

7. ローカルリポのmainブランチをリモートリポからpullして更新

ローカルのmainブランチに移動してリモートリポジトリのmainブランチをpullする
※ ローカルでupdate-readmeをマージするのではなく、リモートリポジトリからpullすることでmainブランチを最新にすること

```
# ローカル：どこのブランチにいるか確認
$ git branch

main
* update-readme

# mainブランチに移動：checkout
$ git checkout main

# リモートリポジトリからpull
$ git pull origin main
```

8. 不要になったブランチを削除する（git branch -d）

特定のブランチを削除: git branch -d <branch-name>

```
$ git branch -d update-readme
Deleted branch update-readme (was 67866ec).
```

・mainにマージしていないブランチは削除できない
・強制的に削除する場合は -D （大文字）オプションで削除できる

リモートリポジトリはGitHub上から削除

※リモートリポジトリのブランチは削除を忘れやすいので、定期的に削除を実行する。

GitHubのSSH接続設定

①ローカルマシンでSSHキーペアを作成
②公開鍵をGitHubに登録
③SSH configファイルの修正（プライベート鍵とGitHubの紐づけ）
④接続確認

参考

★Gitの基本ワークフロー。流れとコマンドまとめ
Git_GitHub再入門
【入門】Github Flowとは？使い方の基本
GitHub Flowとは
とほほのGit入門