

Github講習



基本情報

Gitとは？



- ・バージョン管理システム
ファイルの変更履歴を管理するツール
- ・分散型システム
各ユーザが完全な履歴を持ち、オフラインでも作業ができる
- ・オープンソースソフトウェア

Gitとは？



最終レポート.txt



New_最終レポート.txt



完成_最終レポート.txt

どのファイルが最新でいつ誰が更新したのかわかりづらい



Gitとは？



ローカル

リモート

コミット

マージ

プルリク
エスト

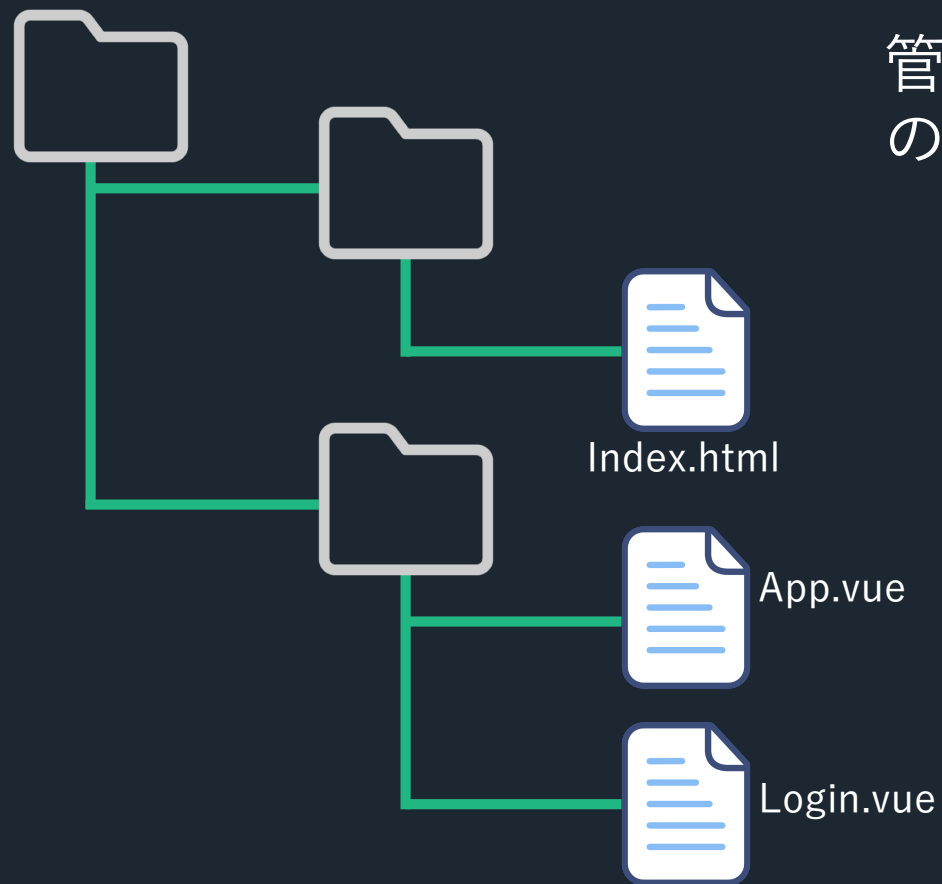
プル

プッシュ

ブランチ



リポジトリ(repository)



管理したいファイル(ソースコード)
のまとまりのこと

コミット(commit)



(例)

ステートマシン
の構築



物体認識
の実装



人物認識の
追加



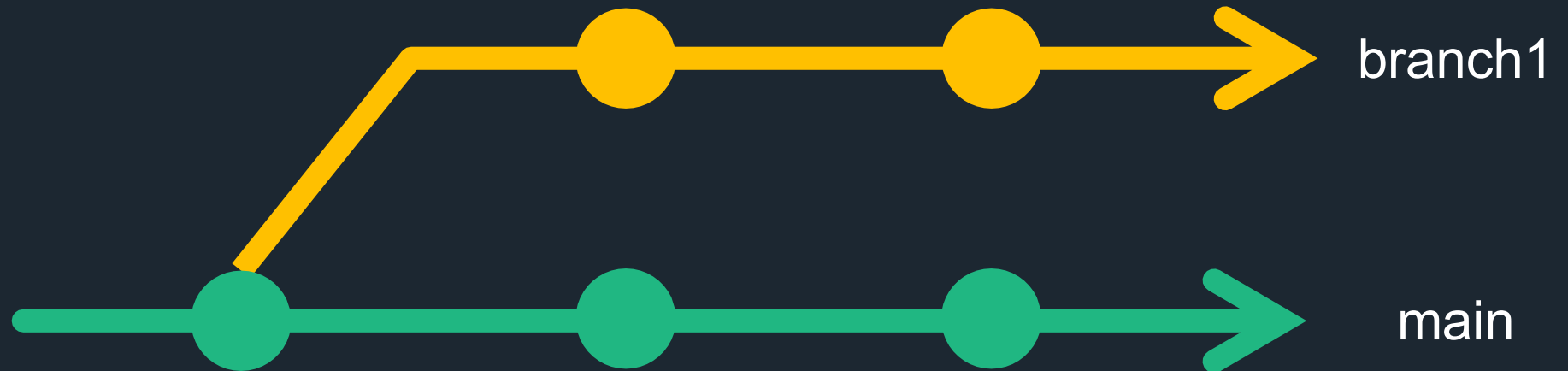
コミット1

コミット2

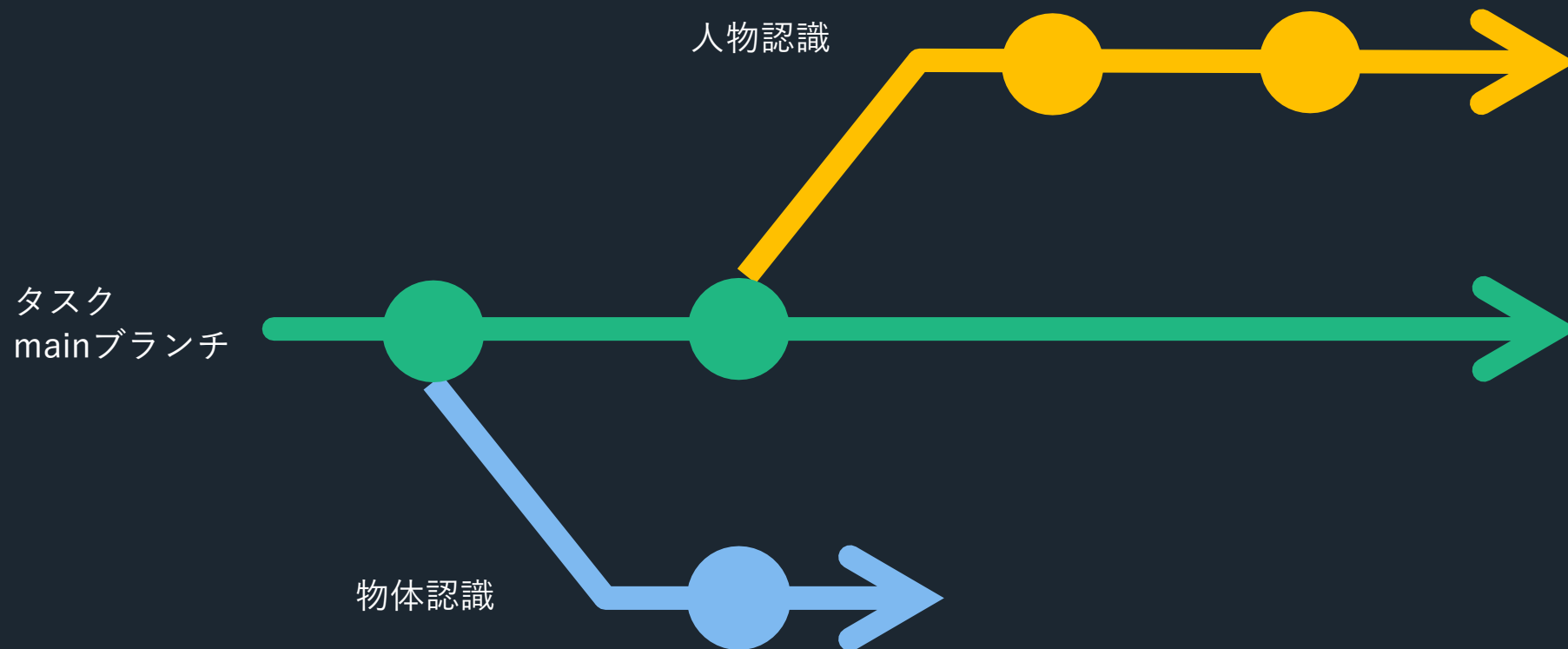
コミット3

ブランチ (branch)

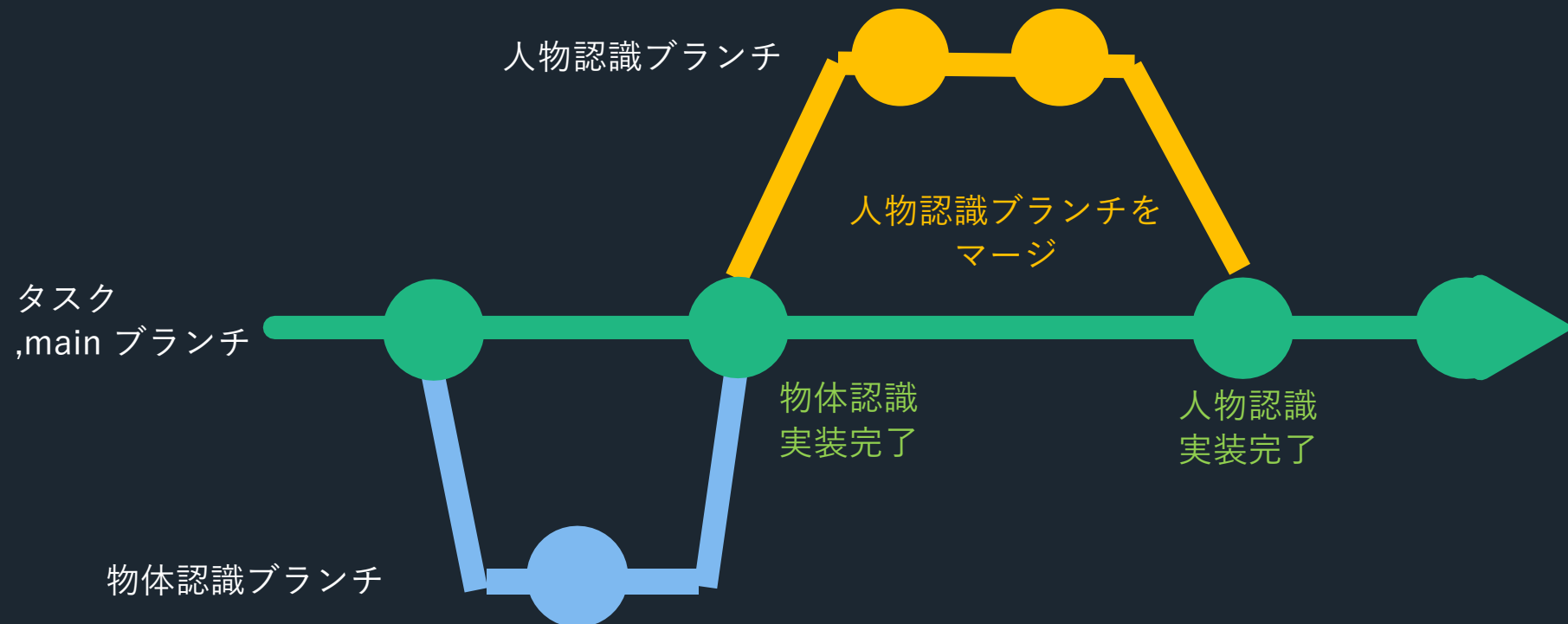
親のコミットから履歴を分岐する



ブランチ(branch)

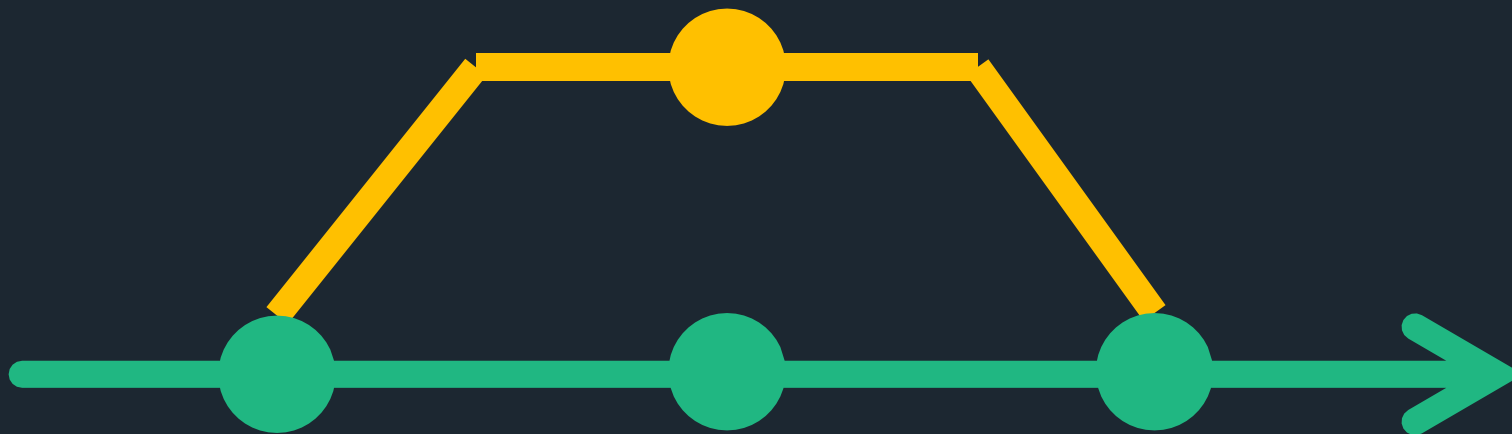


マージ(merge)



マージ(merge)

特定のブランチから別のブランチに変更を取り込む



Github

Gitで管理されたファイルを
インターネットにアップロードして
複数人での取り扱いを便利にする
プラットフォーム

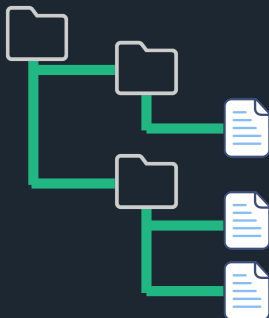


リモート・ローカル/リポジトリ

各ユーザのPC

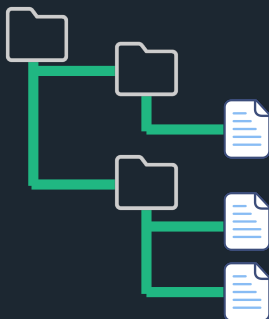

ユーザ A

ローカルリポジトリ




ユーザ B

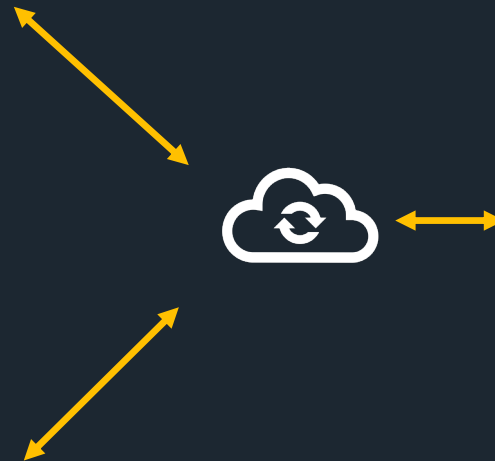
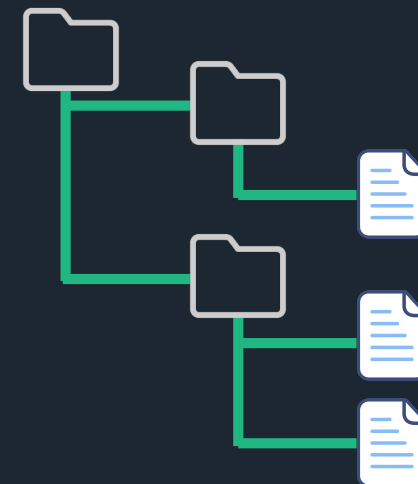
ローカルリポジトリ



Github(インターネット上)



リモートリポジトリ



リモート・ローカル/リポジトリの利点

- ・ チームでの共同作業が容易

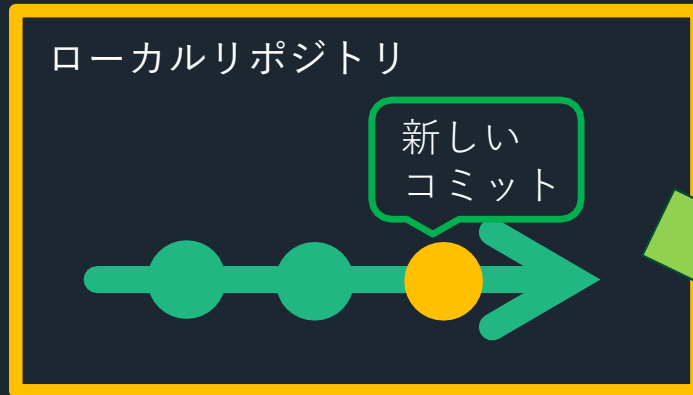
ネットにアップロードすれば, ほかの人と容易に
コードの共有が可能になる

- ・ バックアップの役割

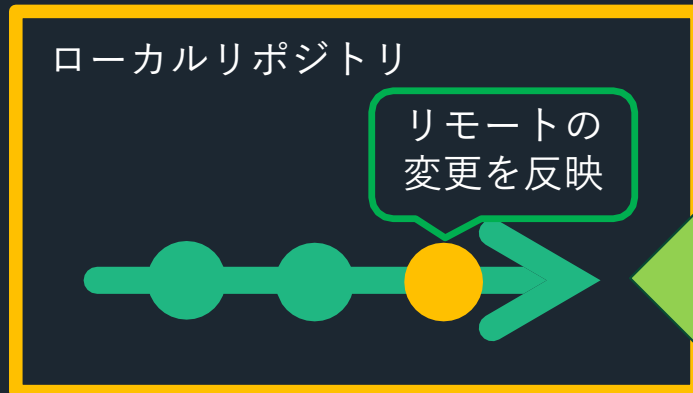
自分のPCが壊れてもネットから
もう一度引っ張ってくれば問題ない

プッシュ・プル

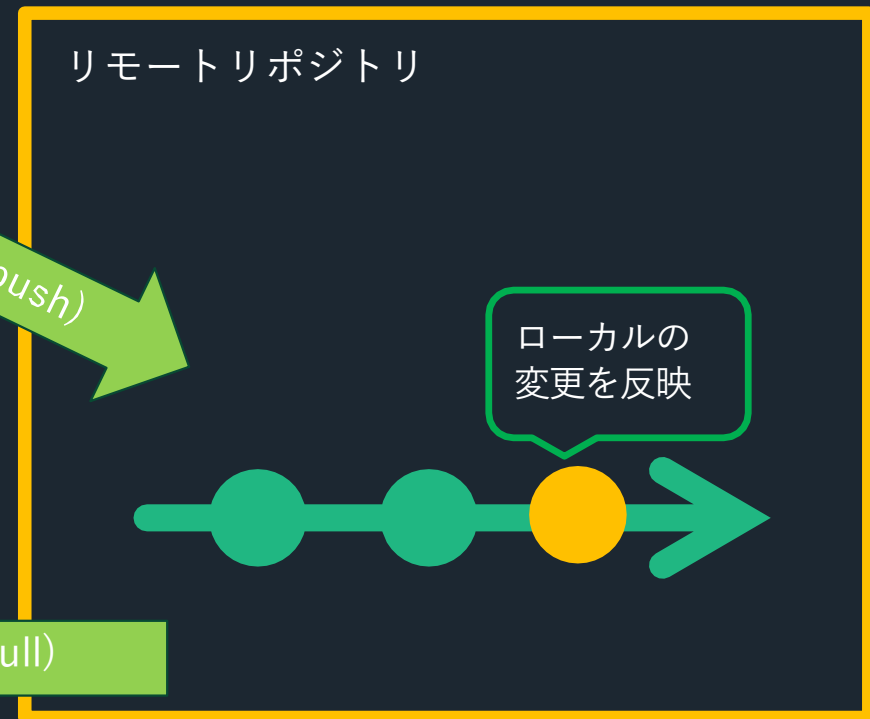
ユーザAのPC



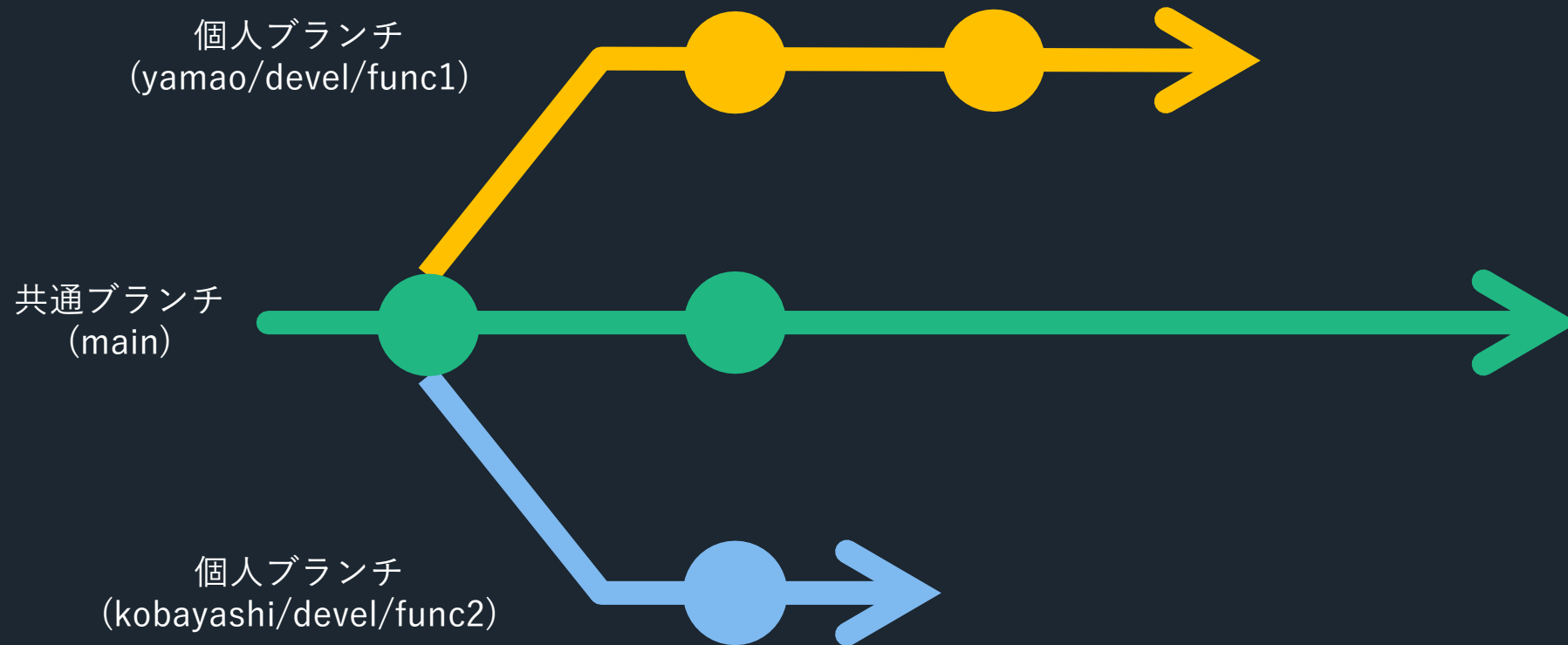
ユーザBのPC



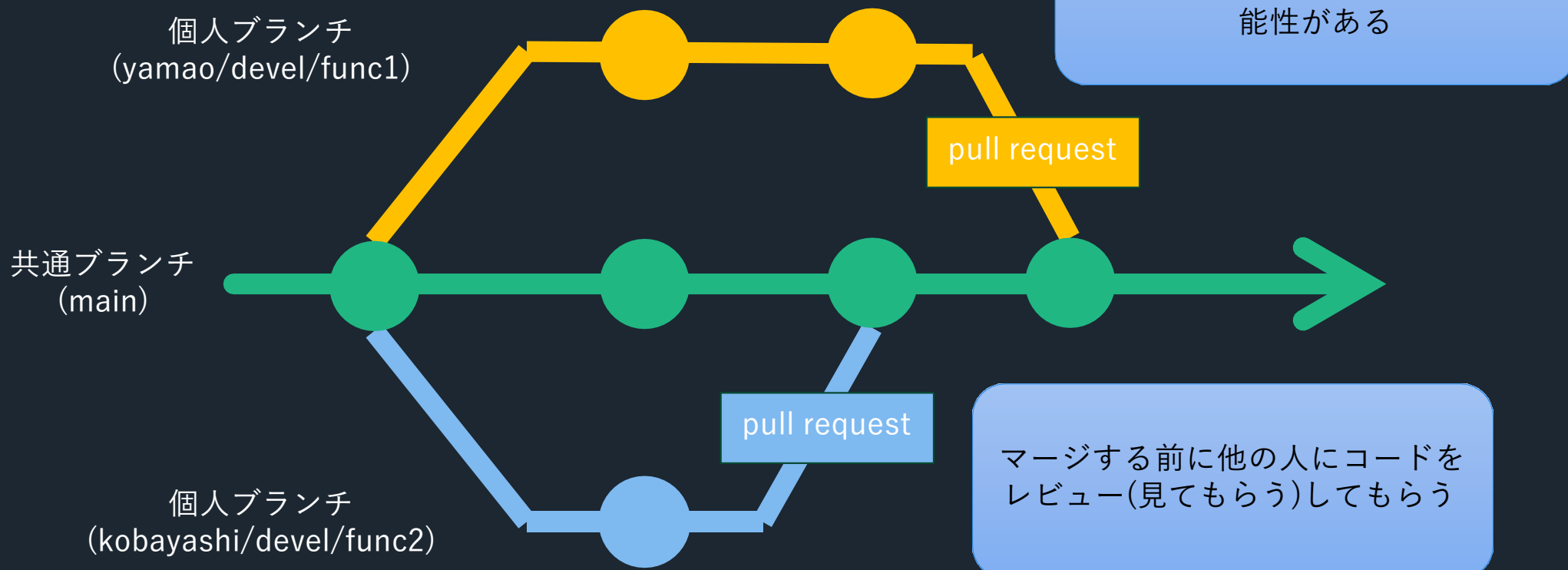
Github(インターネット上)



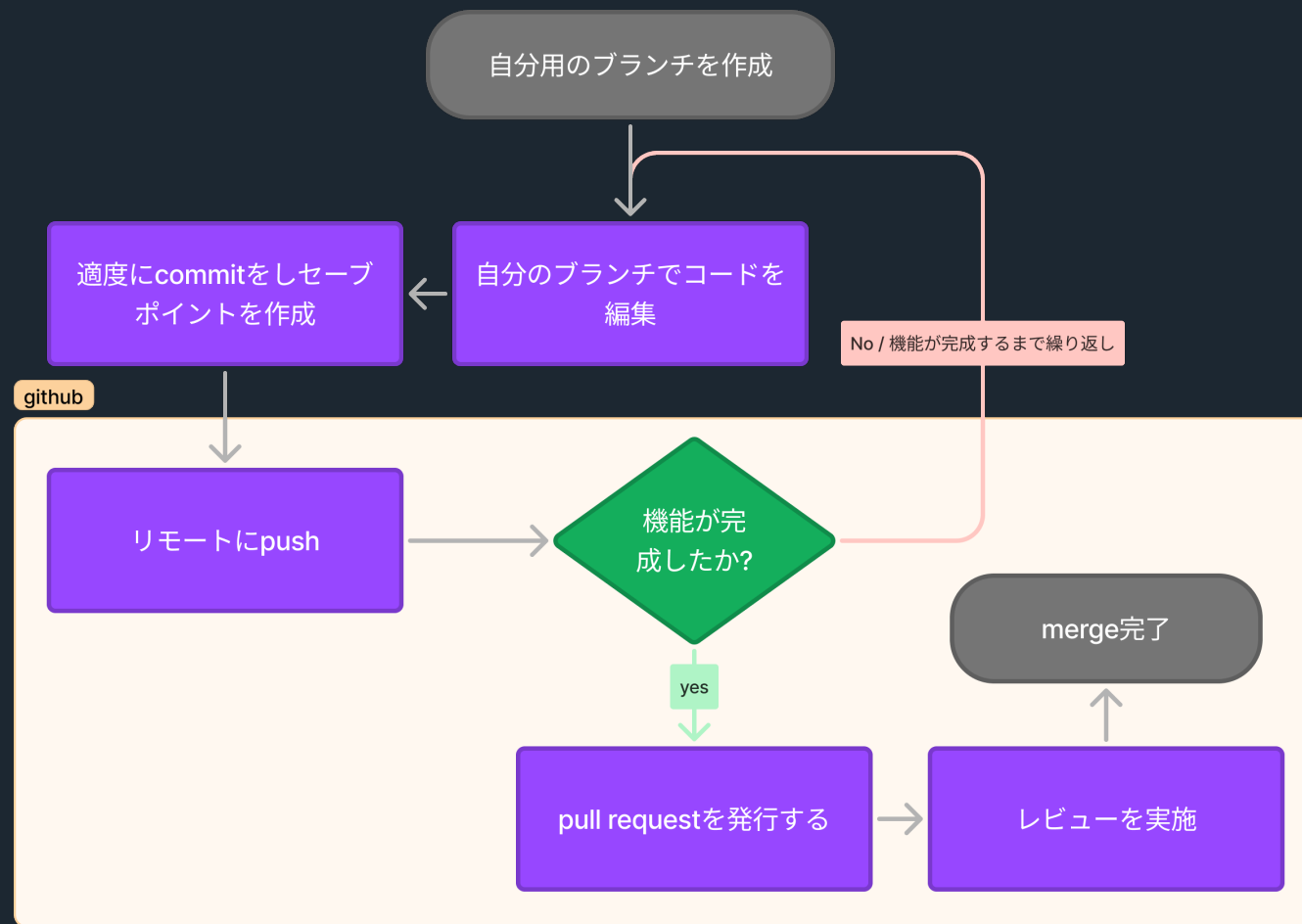
pull request



pull request



フローチャート ブランチ作成からマージまで



Gitを扱う際の注意点

命名規則について

※ 命名規則はチームによって大きく異なる

- ・ ブランチ名

自分の名前/開発カテゴリ/開発内容 としてください

(例) yamao/devel/add_human_detection , yamao/hotfix/bug_fix
yamao/check/mask_filter , yamao/fix/order_bug

- ・ コミットメッセージ

開発内容が分かりやすいメッセージにしてください

(例) 人物認識の機能を追加 , マスク処理の微修正

心がけてほしいこと

- ・ プログラムに何かしらの追加をする場合は必ずブランチを切る
競合やソースコードの損失を防ぎ、バージョン管理が正しく行う
大会やデモ等で使用するブランチに直書きすると本番で動かなくなり、事故につながる
- ・ こまめにコミット、プッシュする癖をつける
こまめなコミットを積み上げることで問題発生時の切り分けが
すぐにできたり、他のユーザとの競合を防げたりする

以下，適当である.

ハンズオン

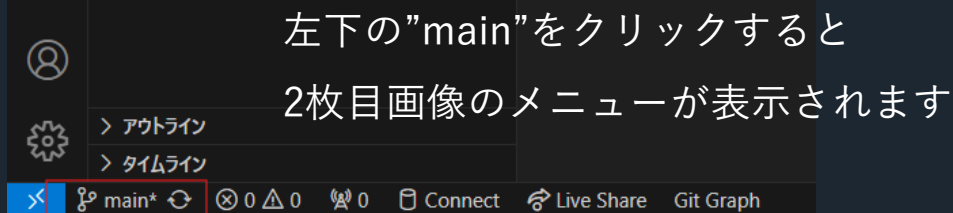
<https://github.com/TN-Project/github-lecture>

```
git clone { url }
```


自分のブランチを作成してみよう

VSCode

左下の”main”が現在参照している
ブランチを指しています



チェックアウトするブランチまたはタグを選択します

+ 新しいブランチの作成...

+ 新しいブランチを以下から作成...

🔗 チェックアウトがデタッチされました...

🔗 main df7a224e ブランチ

🔗 origin/main df7a224e でのリモート ブランチ リモート ブランチ

CLI

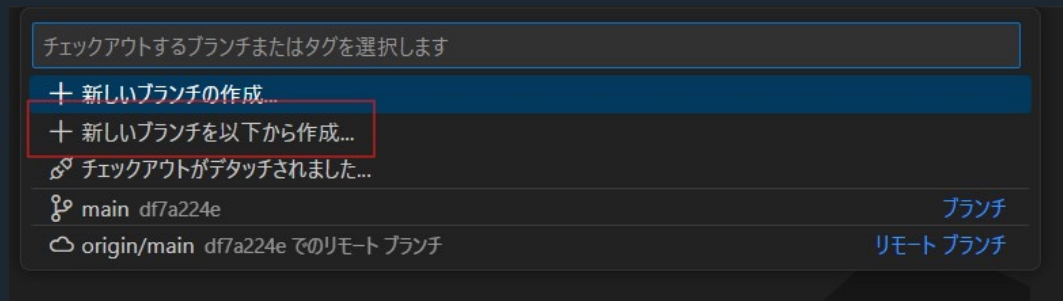
```
git checkout -b { 作成するブランチ名 }  
                { ブランチを作成する元のブランチ }
```

例 (mainからyamao/devel/demoを作成)

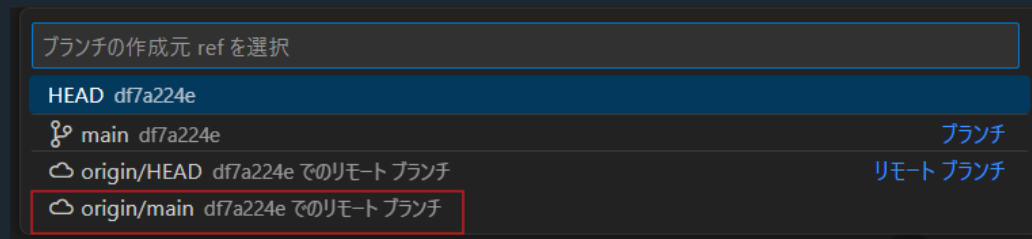
```
git checkout -b yamao/devel/demo main
```

自分のブランチを作成してみよう

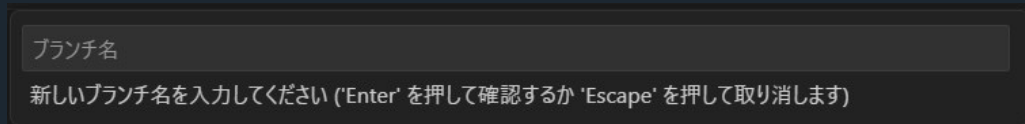
VSCode



新しいブランチを以下から作成をクリック。



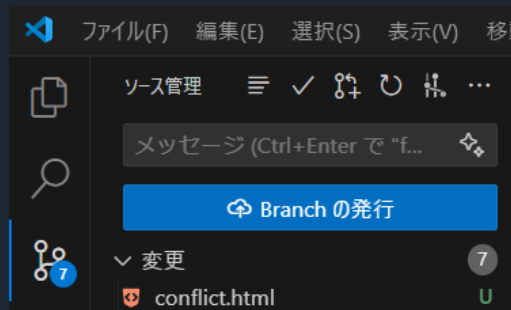
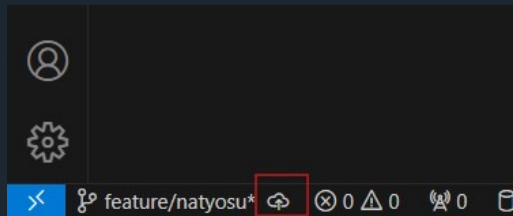
リモートブランチの"origin/main"を選択



ブランチ名を入力する

自分の名前/devel/機能名

自分のブランチを作成してみよう(リモートに登録)

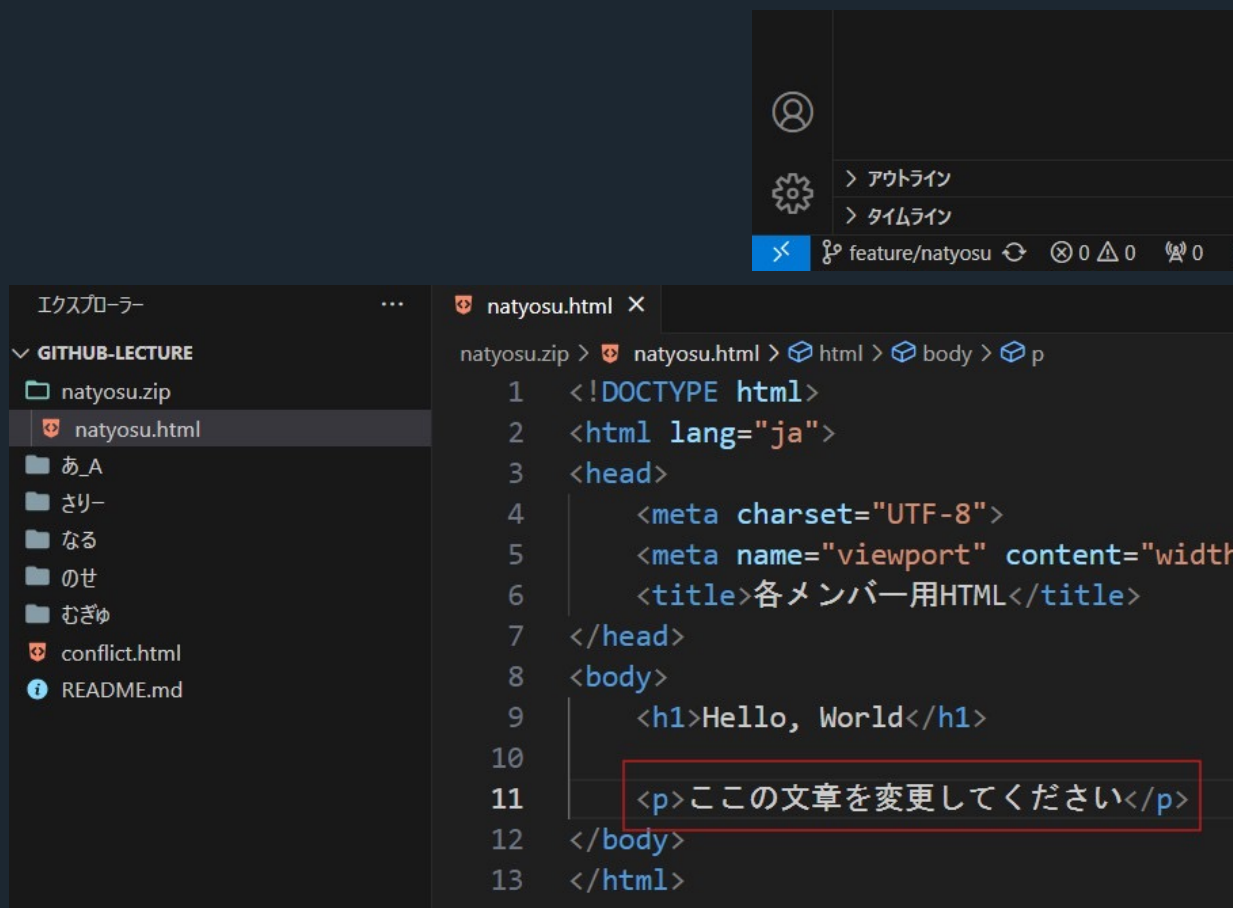


左下の雲に矢印が刺さっているマークをクリックするか、サイドバーのgit管理からbranchの発行をクリック

CLI

```
git push -u origin yamao/devel/demo
```

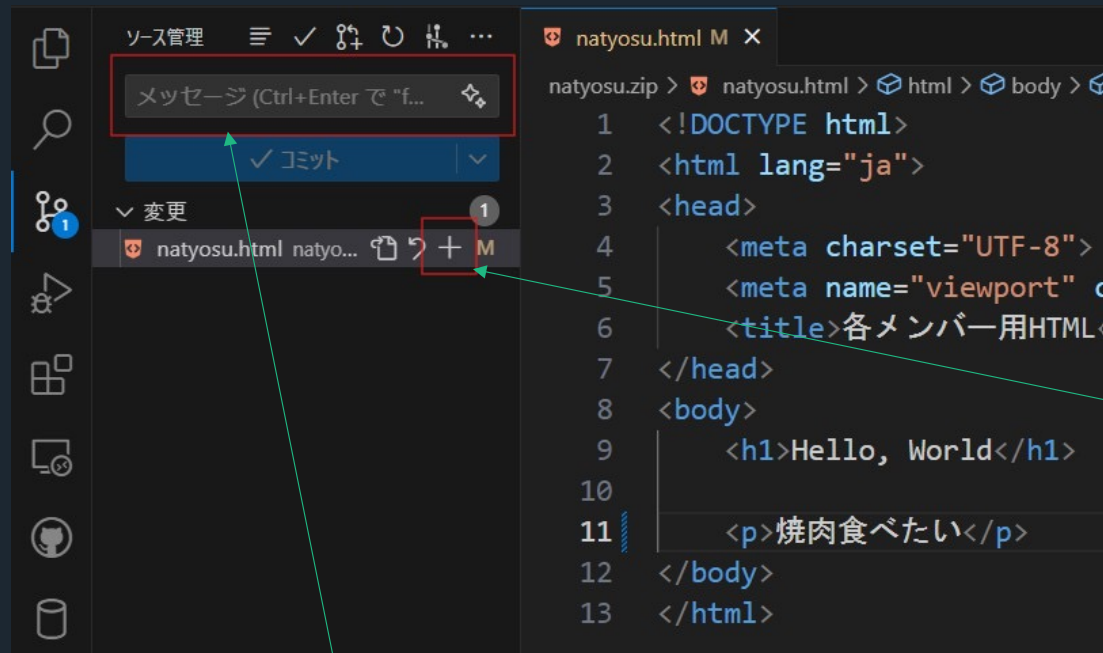
コミットを作成してみよう



左下で自分のブランチに移動
(参照)しているかを確認する.

自分のブランチに移動できて
いれば, 自分の名前のフォル
ダ内のhtmlファイルを編集し
て保存してください.

コミットを作成してみよう

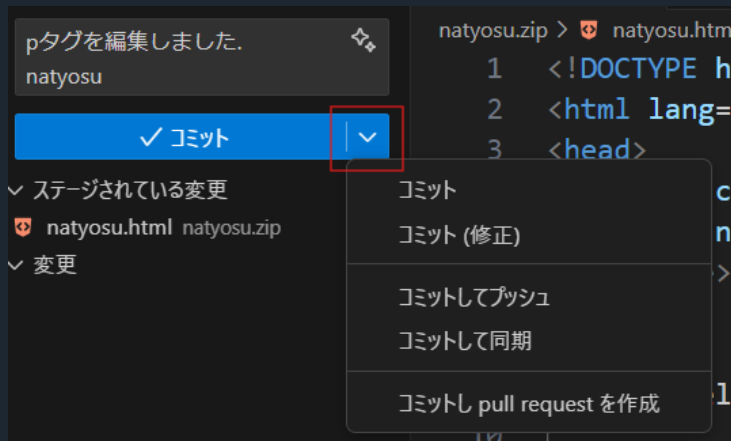


保存をしたら画像のようになっていたはず。

コミットを作成するときは、セーブするファイルを選択(ステージする)してコメントを書いてセーブ(コミット)をします。

セーブのコメントを書く, ex: pタグを編集しました。

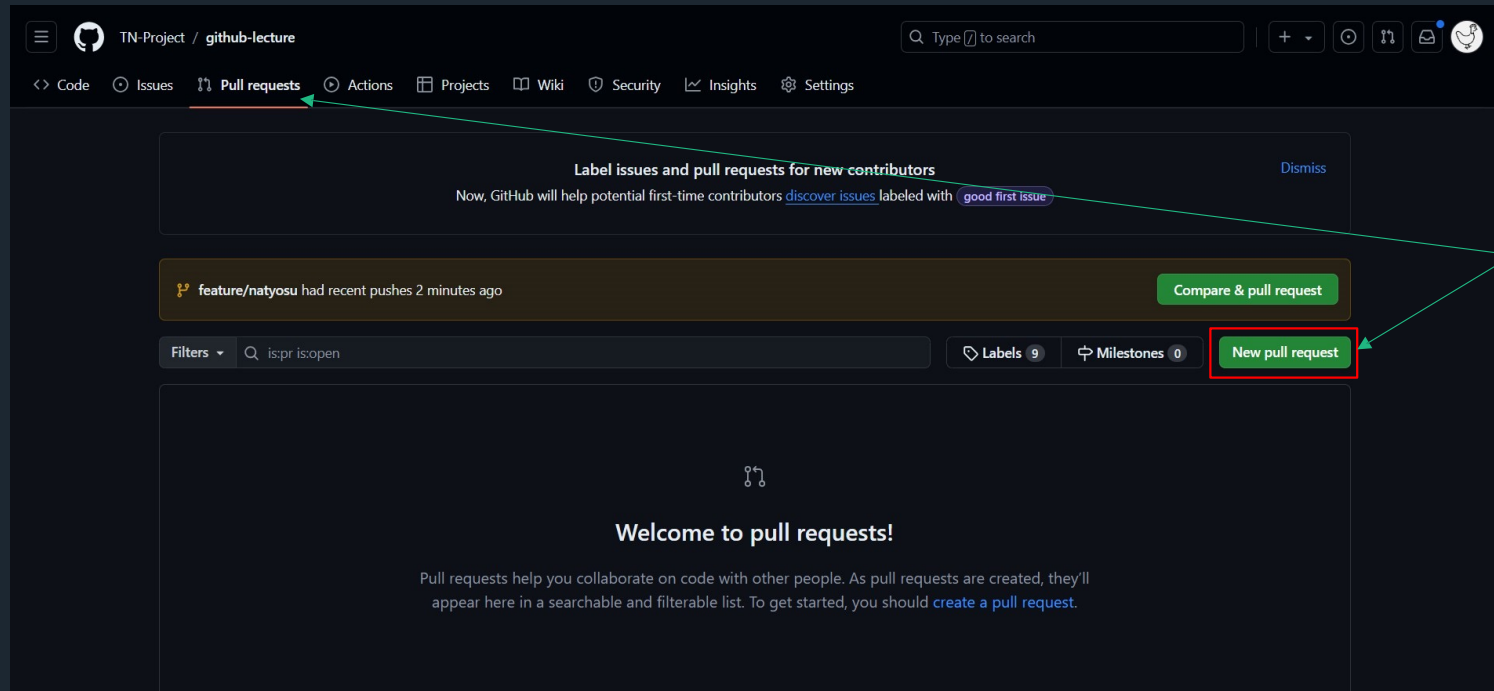
コミットを作成してみよう



逆さ山をクリックし, “コミットしてプッシュ”をクリックすることでコミットができます.

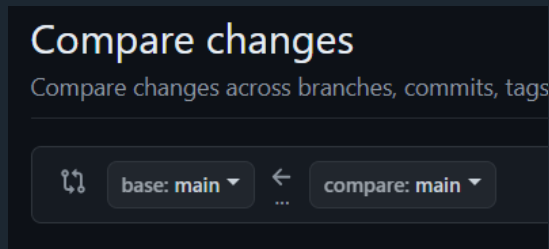
同様に2, 3個コミットを作成してみましょう

Pull requestを作成してみよう

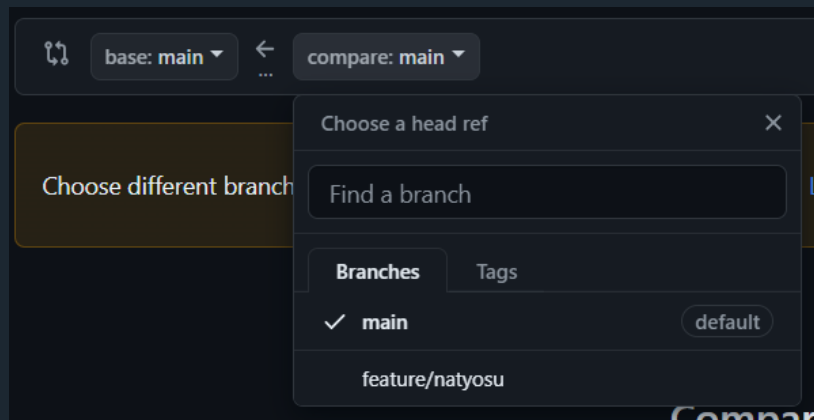


Githubのリポジトリページで pull request タブに移動して New pull request をクリックしてください。

Pull requestを作成してみよう

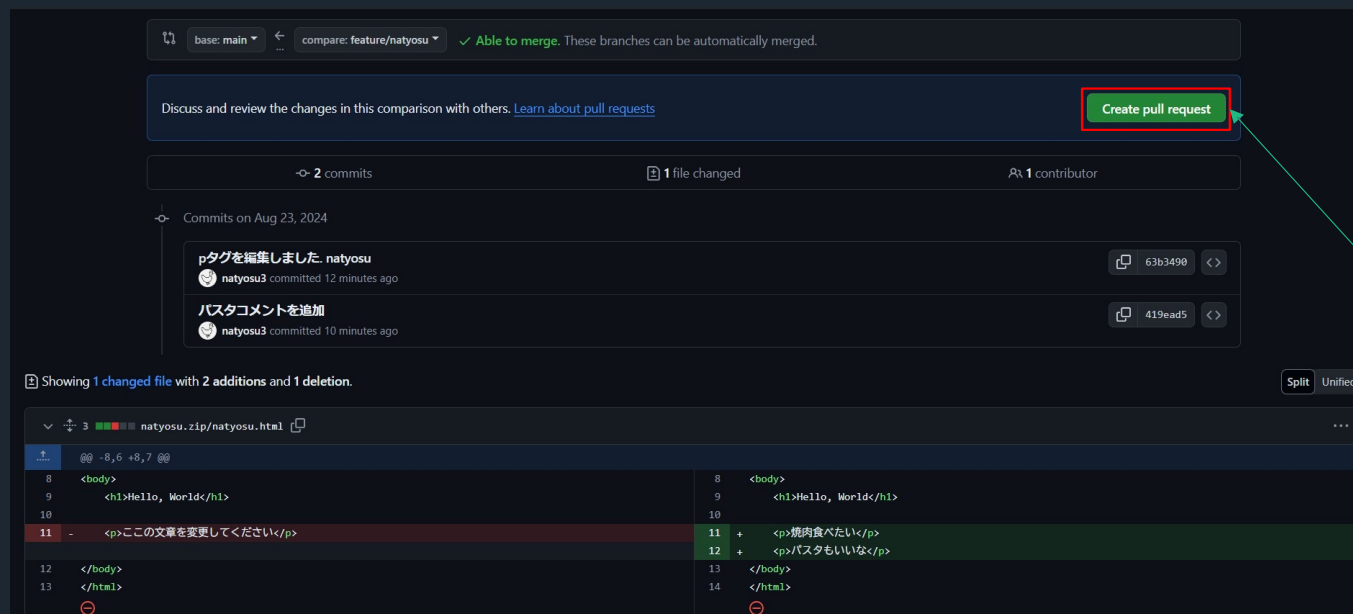


これはどのブランチをどこのブランチにmerge(結合)したいのかを選びます。



左側が結合させる先, 右側が結合させるブランチ

Pull requestを作成してみよう



選択すると、今までのコミット履歴と差分が表示されます。

問題無ければ, create pull requestをクリックします。

Pull requestを作成してみよう

Pull requestのタイトルを入力して、簡単なpull requestに関するコメントを書きます。

タイトル例: 機能Aを実装しました。

Reviewersにレビューをして欲しい人を選択します。

Add a title

Feature/natyosu

Add a description

Write Preview

H B I

Add your description here...

Markdown is supported Paste, drop, or click to add files

Reviewers

No reviews

Assignees

No one—[assign yourself](#)

Labels

None yet

Projects

None yet

Milestone

No milestone

Development

Use [Closing keywords](#) in the description to automatically close issues



Helpful resources

Create pull request

最後にCreate pull requestを押しpull requestを作成できます。

レビュー&マージをする

natyosuブランチの編集が完了しました. #1

 Open natyosu3 wants to merge 2 commits into `main` from `feature/natyosu` 

 Conversation 0  Commits 2  Checks 0  Files changed 1




natyosu3 commented 4 minutes ago

Member ...


aaa



 natyosu3 added 2 commits 28 minutes ago

 pタグを編集しました. natyosu

63b3490

 パスタコメントを追加

419ead5



Require approval from specific reviewers before merging

[Rulesets](#) ensure specific people approve pull requests before they're merged.

Add rule ×



This branch has no conflicts with the base branch


Merging can be performed automatically.

Merge pull request





You can also [open this in GitHub Desktop](#) or view [command line instructions](#).

Conversation 0 Commits 2 Checks 0 Files changed 1

Changes from all commits File filter Conversations Jump to 

Review

natyosu.zip/natyosu.html 

Viewed  ...

@@ -8,6 +8,7 @@

8	<body>	8	<body>
9	<h1>Hello, World</h1>	9	<h1>Hello, World</h1>
10		10	
11	- <p>この文章を変更してください</p>	11	+ <p>焼肉食べたい</p>
		12	+ <p>パスタもいいな</p>
12	</body>	13	</body>
13	</html>	14	</html>

レビューが問題無ければ, Merge pull requestをクリックしてmergeを実行できます.