

Git & GitHub

GitとGitHubの使い方

2024/02/20

Git

Gitとは

- ・ 現在、世界で最も使われている
バージョン管理ツール
- ・ バージョン管理は**ローカル**で行われる
- ・ コードは**ローカル**で管理される



GitHub

GitHubとは

- ・ 現在、世界で最も使われている
バージョン管理ツール
- ・ バージョン管理は**オンラインで**
行われる
- ・ コードは**GitHubサーバー上で**
管理される
- ・ OSSなどのコードを公開



Difference

GitとGitHubの違い



Git

ローカル



管理



GitHub

リモート

Repository

リポジトリ

- Gitにおける**ファイルや履歴をまとめて保存**できる
フォルダーのようなもの
- GitとGitHub、どちらにも必要



ローカルリポジトリ

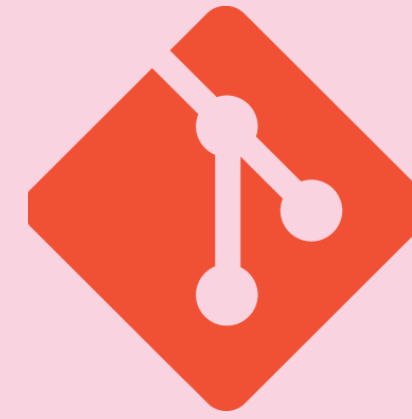


リモートリポジトリ

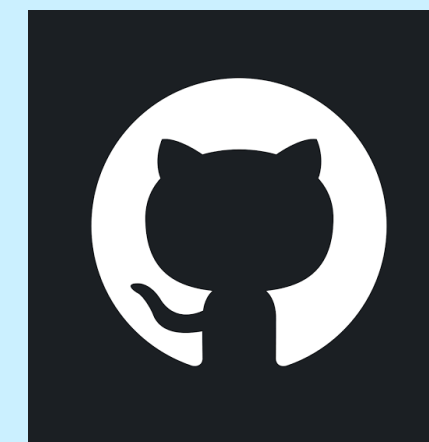
Initialize

リポジトリの初期化

- GitとGitHubを
使用できるようにする操作
- ローカルリポジトリは
作業ディレクトリ内に作られる
- GitHubはRepositoryから
新しく作成する
- GitHubとは紐付けられない



ローカルリポジトリ



GitHub上でGUIによる操作

リモートリポジトリ

Initialize

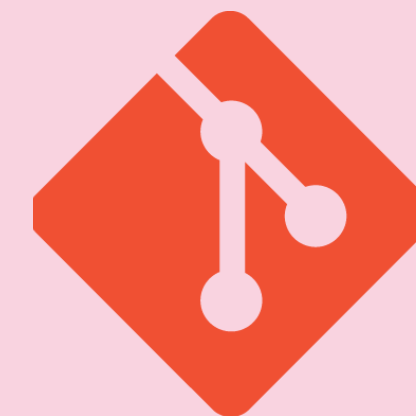
リポジトリの初期化

- GitとGitHubを
使用できるようにする操作
- ローカルリポジトリは
作業ディレクトリ内に作られる
- GitHubはRepositoryから
新しく作成する
- GitHubとは紐付けられない

作業ディレクトリ



アップロードする元データ



.git

コミットログなどのデータ

ローカル環境

Initialize

リポジトリの初期化

- GitとGitHubを
使用できるようにする操作
- ローカルリポジトリは
作業ディレクトリ内に作られる
- GitHubはRepositoryから
新しく作成する
- GitHubとは紐付けられない

The screenshot shows the GitHub 'New repository' page. The browser address bar is 'github.com'. The page title is 'New repository'. The main heading is 'Create a new repository'. Below it, a subtext says 'A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository.](#)'. A note states 'Required fields are marked with an asterisk (*)'. The form has two required fields: 'Owner *' (a dropdown menu showing 'somando') and 'Repository name *' (an empty text input). To the right of the 'Repository name' field, the text 'リポジトリ名' is written. Below these fields, a hint says 'Great repository names are short and memorable. Need inspiration? How about [studious-octo-invention](#) ?'. There is an optional 'Description' text input field. Below the description field, there are two radio button options for visibility: 'Public' (selected) and 'Private'. To the right of these options, the text '公開設定' is written. Below the visibility options, there is a section 'Initialize this repository with:'. It contains two checkboxes: 'Add a README file' (unchecked) and 'Add .gitignore' (unchecked). To the right of the 'Add a README file' checkbox, the text 'README' is written. To the right of the 'Add .gitignore' checkbox, the text '.gitignore' is written. Below the 'Add .gitignore' checkbox, there is a dropdown menu for '.gitignore template: None'. Below the 'Add .gitignore' section, there is a section 'Choose a license' with a dropdown menu for 'License: None'. Below the 'Choose a license' section, there is a note: 'A license tells others what they can and can't do with your code. [Learn more about licenses.](#)'. At the bottom of the page, there is a green button labeled 'Create repository'. To the left of this button, the text '作成' is written.

.gitignore

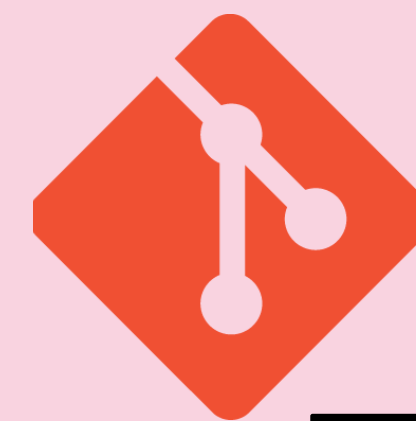
.gitignoreとは

- Gitで管理しないファイルを登録するファイル
- 公開しない情報を選択する
- 非公開情報が入っているものを.gitignoreに登録する事で情報漏洩のリスクを抑えられる

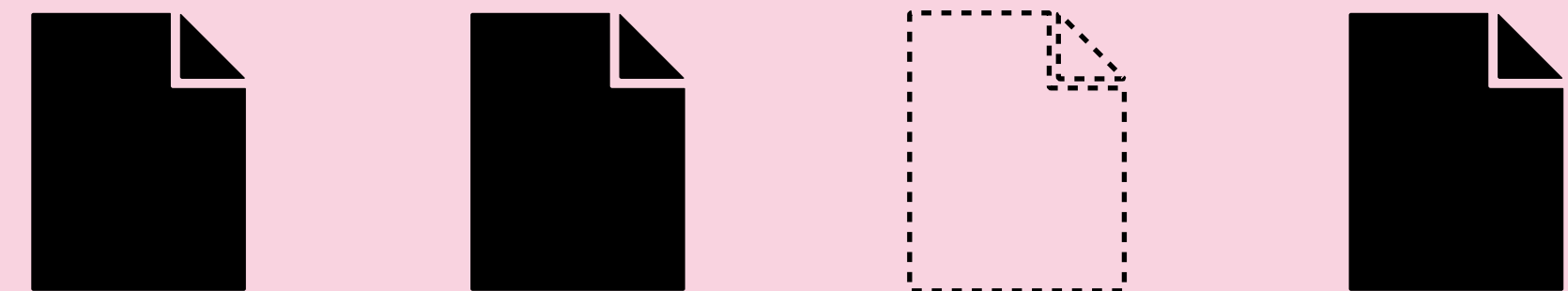
作業ディレクトリ



アップロードする元データ



.git



Gitの管理から外れる

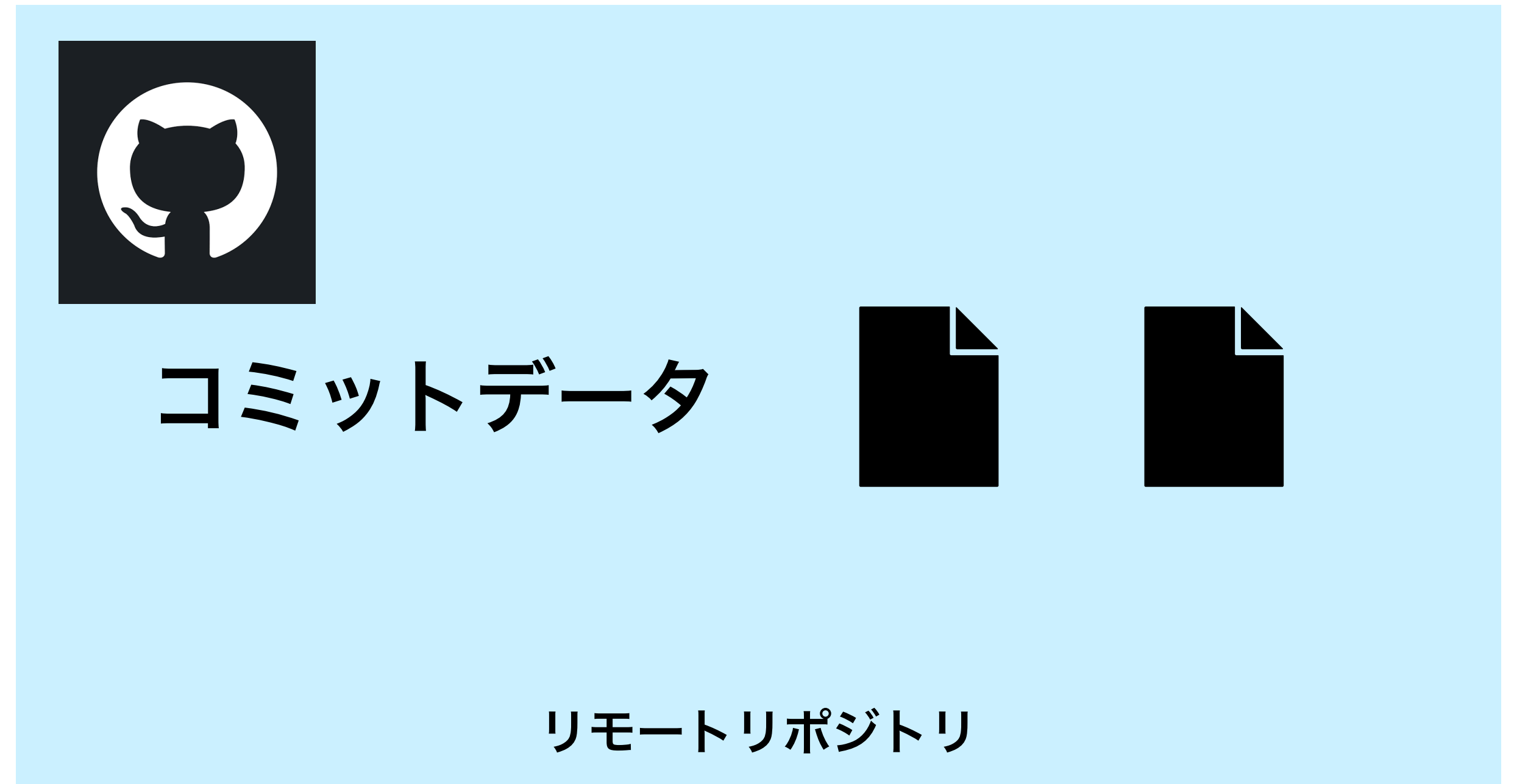
コミットログなどのデータ

ローカル環境

Clone

リポジトリの複製

- GitHubに上がっている
リポジトリを**複製する**操作
- **コミットログも複製**される
- **コマンドを実行した場所**に
新しくフォルダーが作成される
- GitHubとも**紐付け**られる

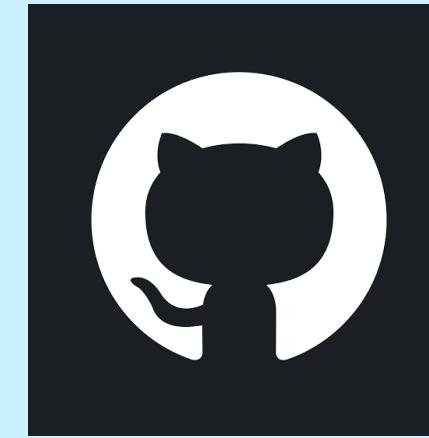


```
git clone [HTTPS]
```

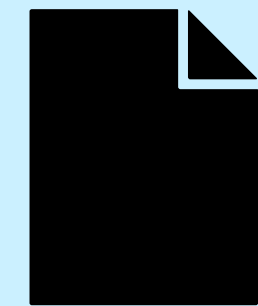
Clone

リポジトリの複製

- GitHubに上がっている
リポジトリを**複製する**操作
- コミットログも複製される
- コマンドを実行した場所に
新しくフォルダーが作成される
- GitHubとも**紐付け**られる

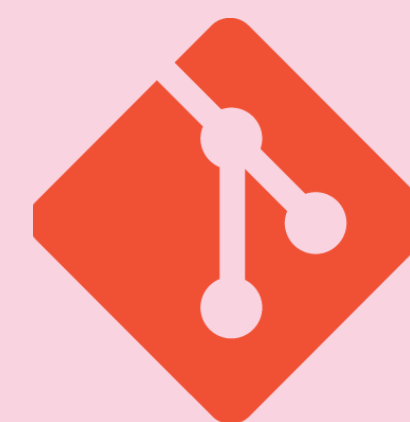


コミットデータ

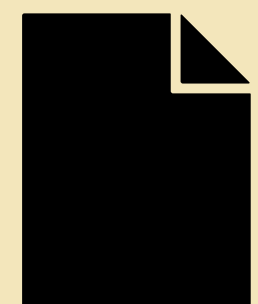


リモートリポジトリ

新規ディレクトリ



.git



ローカルリポジトリ

Remote add

リモートリポジトリの紐付け

- git init で定義した場合は
どのリモートリポジトリにも
紐づいていない
- 以下のコマンドでリポジトリ
紐付ける必要がある

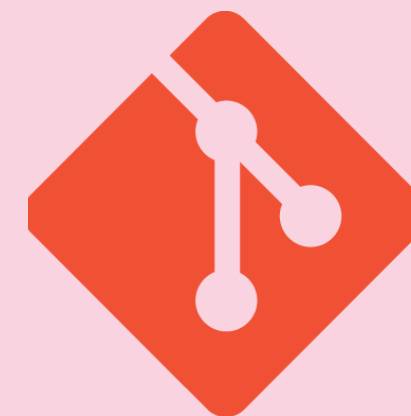


```
git remote add origin [HTTPS]
```

作業ディレクトリ



アップロードする元データ



.git

リモートとの接続情報

コミットログなどのデータ

ローカル環境

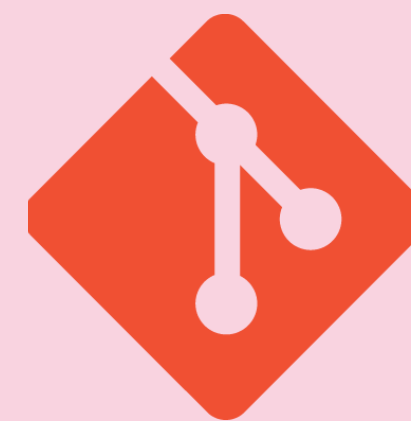
Add

ファイルのステージング

- Gitに登録するファイルを選択する
- フォルダー全体だけでなくファイルを個別に選んでステージングもできる

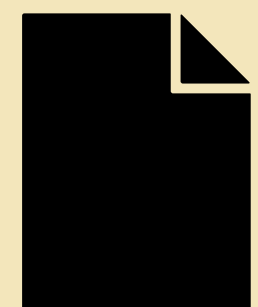
```
git add .  
  
or  
  
git add [ファイル名]
```

作業ディレクトリ



.git

ローカルリポジトリ



Add

ファイルのステージング

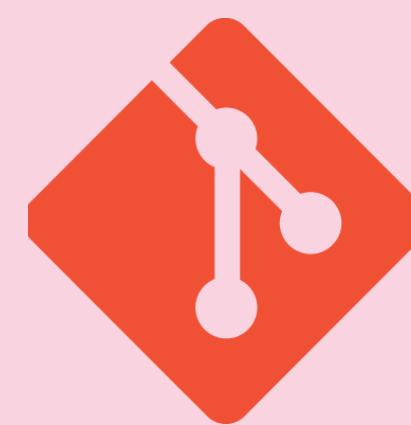
- Gitに登録するファイルを選択する
- フォルダー全体だけでなくファイルを個別に選んでステージングもできる

コミットデータ



ステージング

作業ディレクトリ



.git

ローカルリポジトリ



Commit

コミット

- Gitにステージングしたデータを登録する
- この際にコミット名を付けてどのような変更をしたのかを簡潔に書く



```
git commit -m "[コミット名]"
```

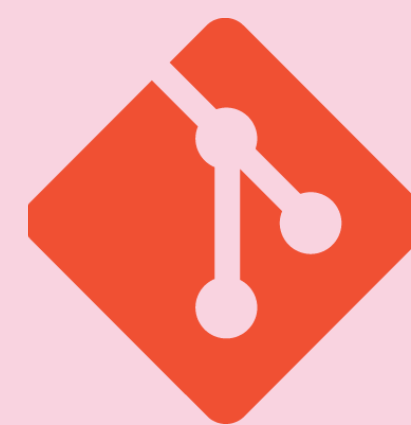
[main xxxxxx] First commit

コミットデータ



ステージング

作業ディレクトリ

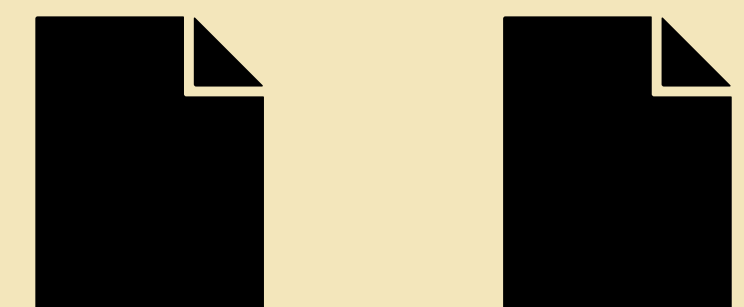


.git

[main xxxxxx]

First commit

ローカルリポジトリ



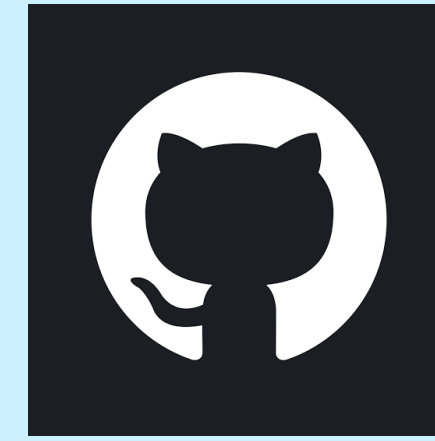
Push

GitHubへアップロード

- ・ コミットされたデータを
GitHubへアップロードする
- ・ 送信先の**ブランチ**を選択する

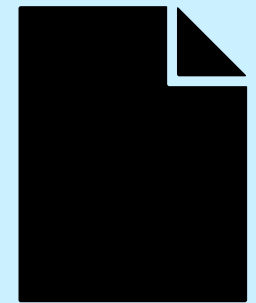


```
git push origin [ブランチ名]
```

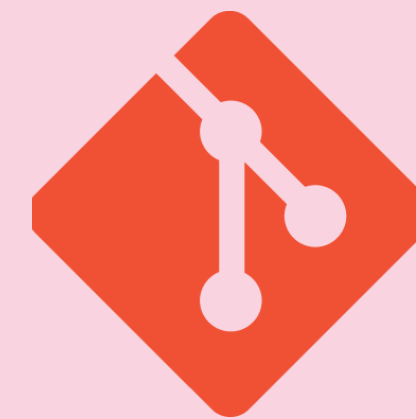


[main] ---

古いコミットデータ



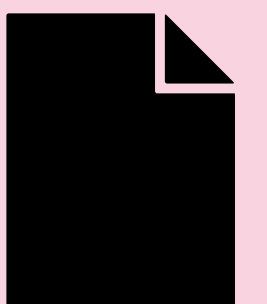
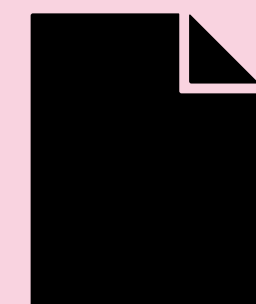
リモートリポジトリ



.git

[main xxxxxx] First commit

最新のコミットデータ



ローカルリポジトリ

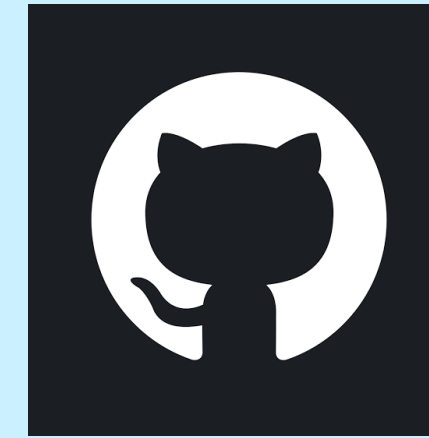
Push

GitHubへアップロード

- ・ コミットされたデータを
GitHubへアップロードする
- ・ 送信先の**ブランチ**を選択する

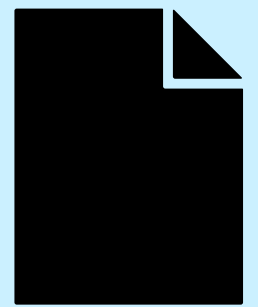
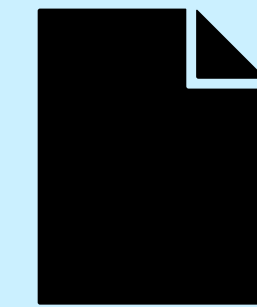


```
git push origin [ブランチ名]
```

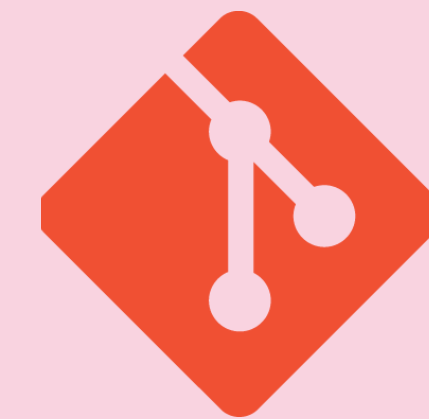


[main xxxxxx] First commit

最新のコミットデータ



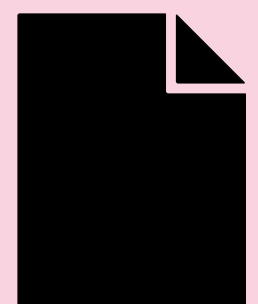
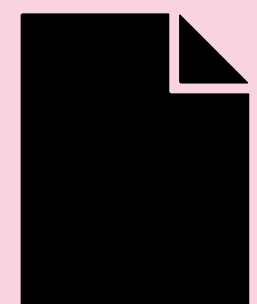
リモートリポジトリ



.git

[main xxxxxx] First commit

最新のコミットデータ



ローカルリポジトリ

Pull

GitHubからプル

- ・ GitHubからプルする
- ・ プル元の**ブランチを選択**する

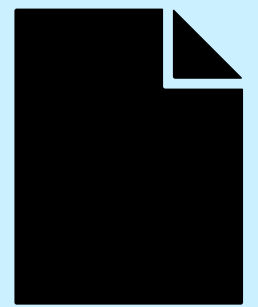


```
git pull origin [ブランチ名]
```

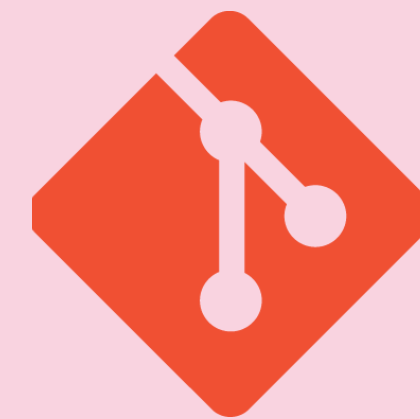


[main xxxxxx] First commit

最新のコミットデータ



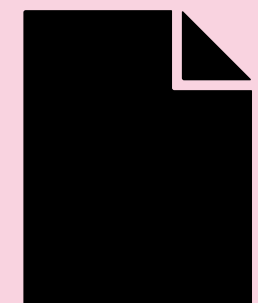
リモートリポジトリ



.git

[main] ---

古いコミットデータ



ローカルリポジトリ

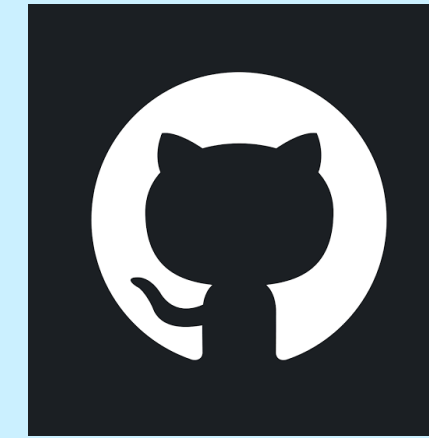
Pull

GitHubからプル

- ・ GitHubからプルする
- ・ プル元の**ブランチを選択**する

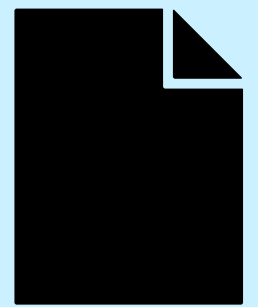
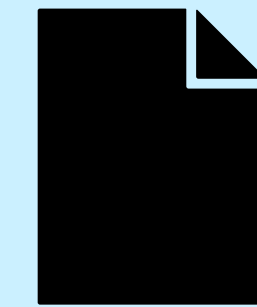


```
git pull origin [ブランチ名]
```

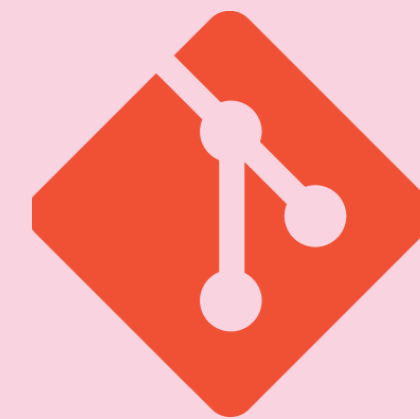


[main xxxxxx] First commit

最新のコミットデータ



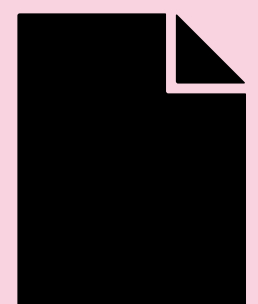
リモートリポジトリ



.git

[main xxxxxx] First commit

最新のコミットデータ

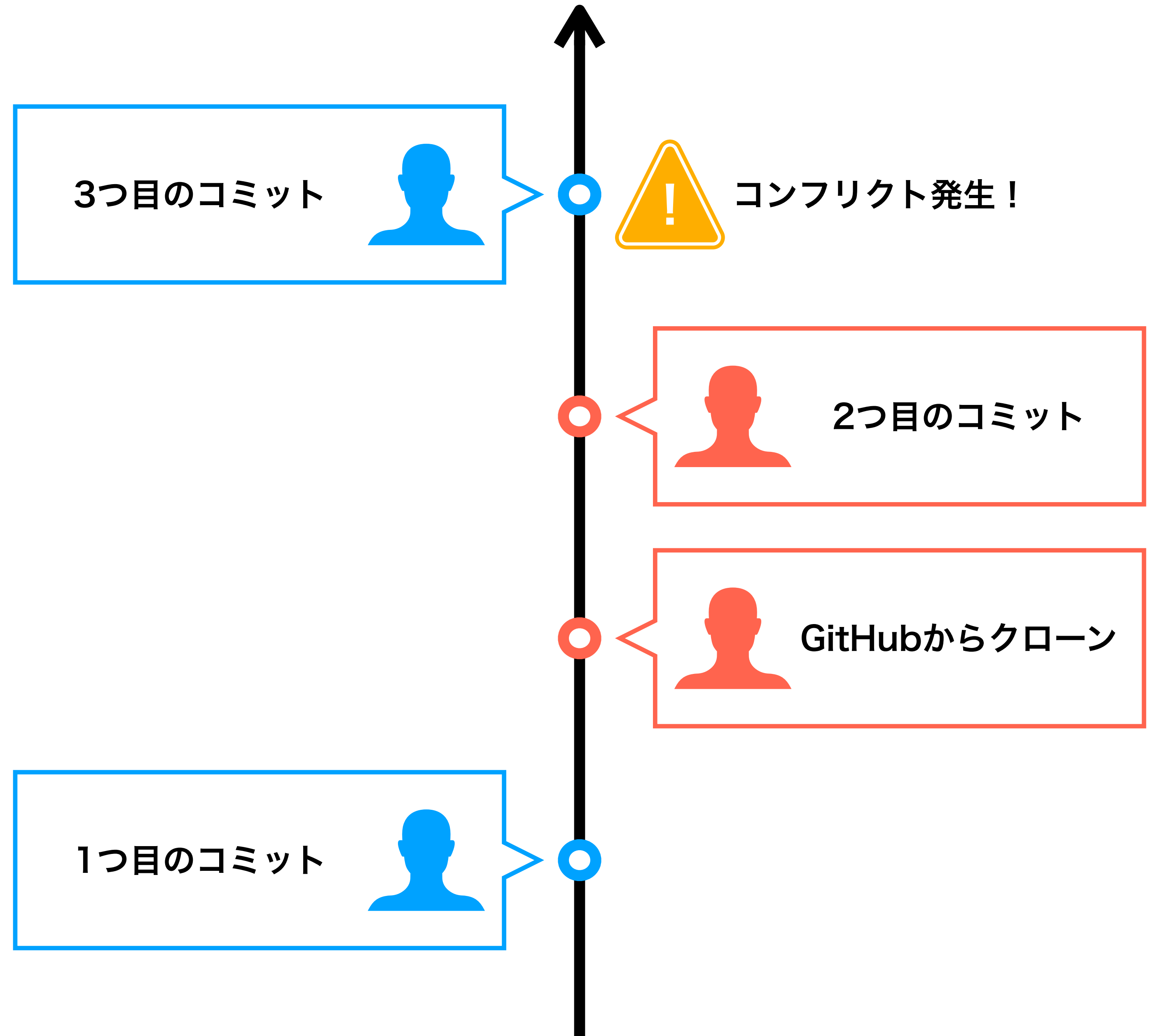


ローカルリポジトリ

Conflict

変更履歴の競合

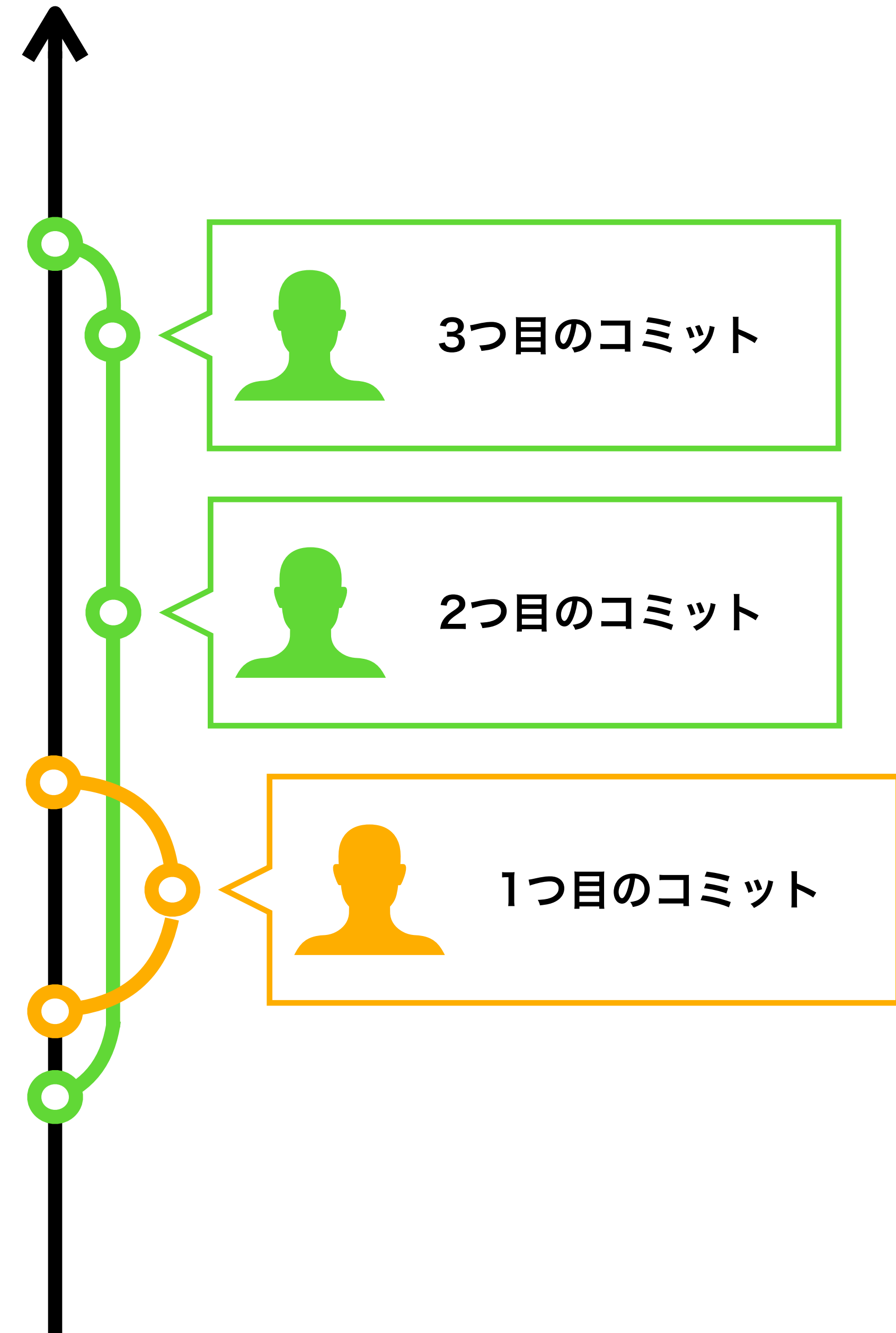
- ・ 複数人で開発している際に
Gitの変更履歴が競合してしまうこと
- ・ コンフリクトは
GitHubからプルし、
変更履歴を合わせることで解消



Branch

ブランチの使用

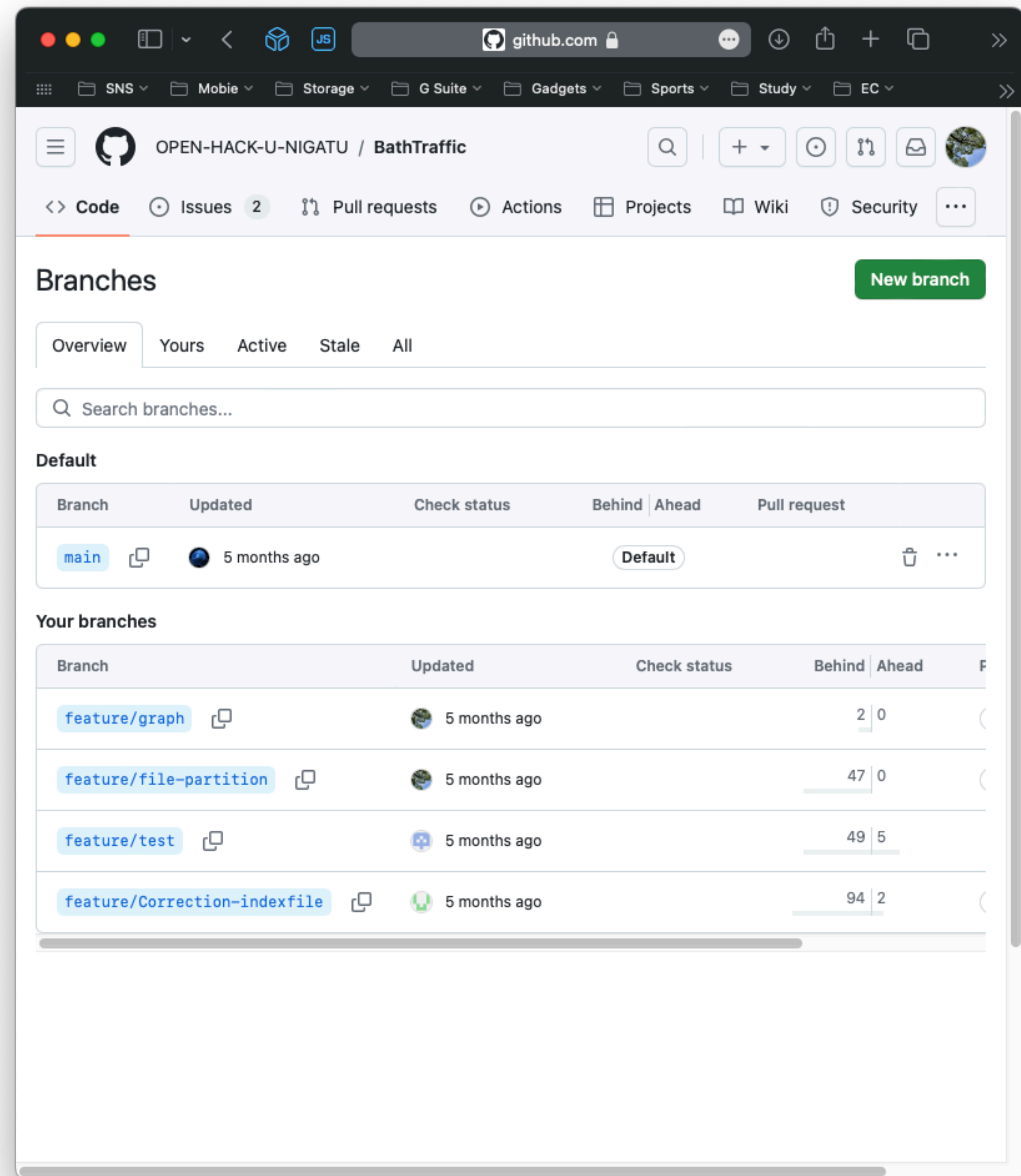
- mainブランチに直接ではなく、別のブランチを作成することでプロジェクト本体に影響を与えずに開発できる



Branch

ブランチの使用

- mainブランチに直接ではなく、別のブランチを作成することでプロジェクト本体に影響を与えずに開発できる



Branch

ブランチの確認

- 以下のコマンド実行で
現在ローカルで使用している
ブランチが確認できる

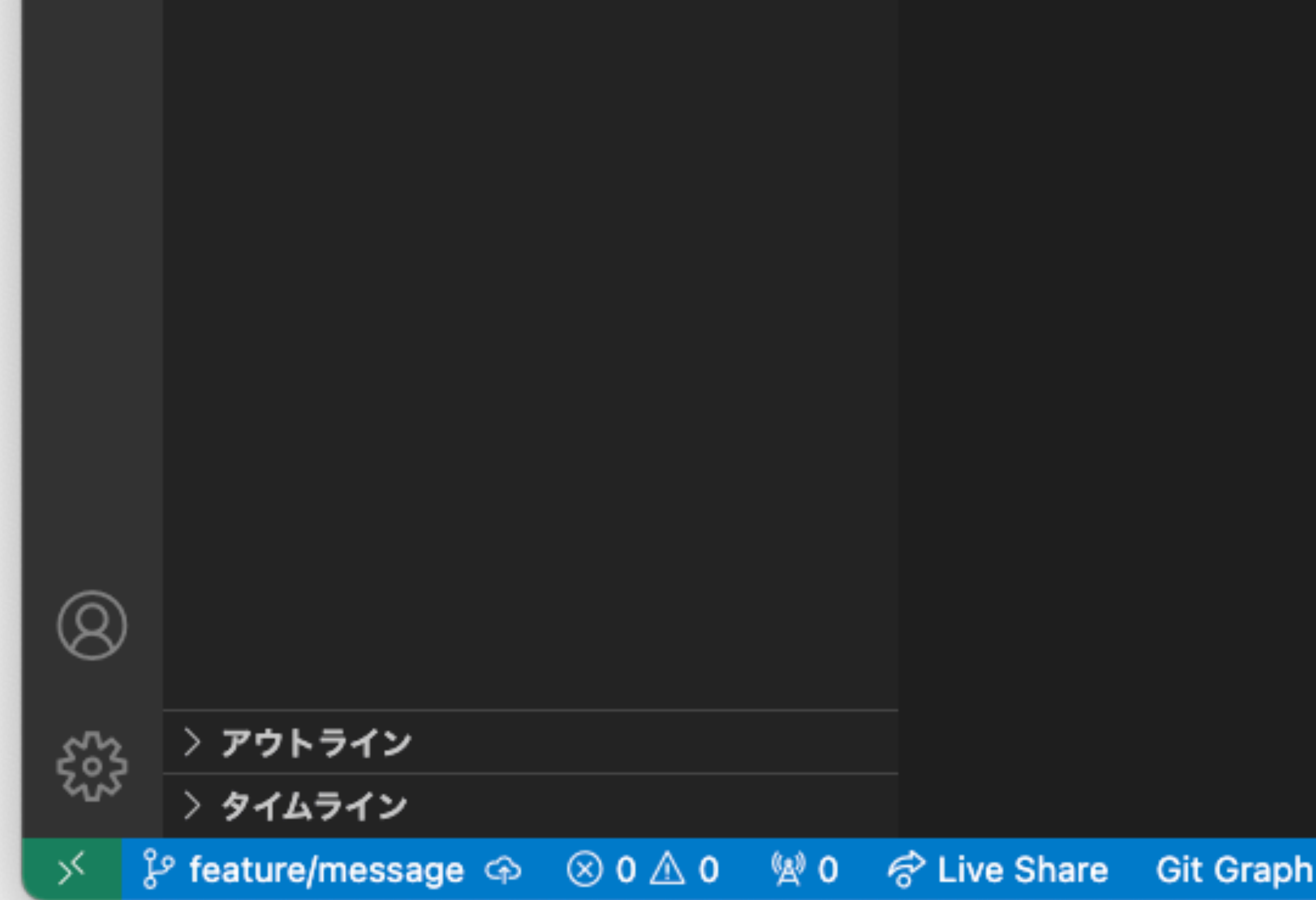
A screenshot of a macOS terminal window titled 'BathTraffic — -zsh — 50x10'. The terminal shows the command 'git branch' being executed, resulting in the output: '* feature/message' and 'main'. The prompt is 'soma@MacBook-Air-2 BathTraffic %'.

```
[soma@MacBook-Air-2 BathTraffic % git branch  
* feature/message  
main  
soma@MacBook-Air-2 BathTraffic % ]
```

Branch

ブランチの確認

- 以下のコマンド実行で
現在ローカルで使用している
ブランチが確認できる
- VS Codeの左下部分にも表示



Create Branch

リモートブランチの作成

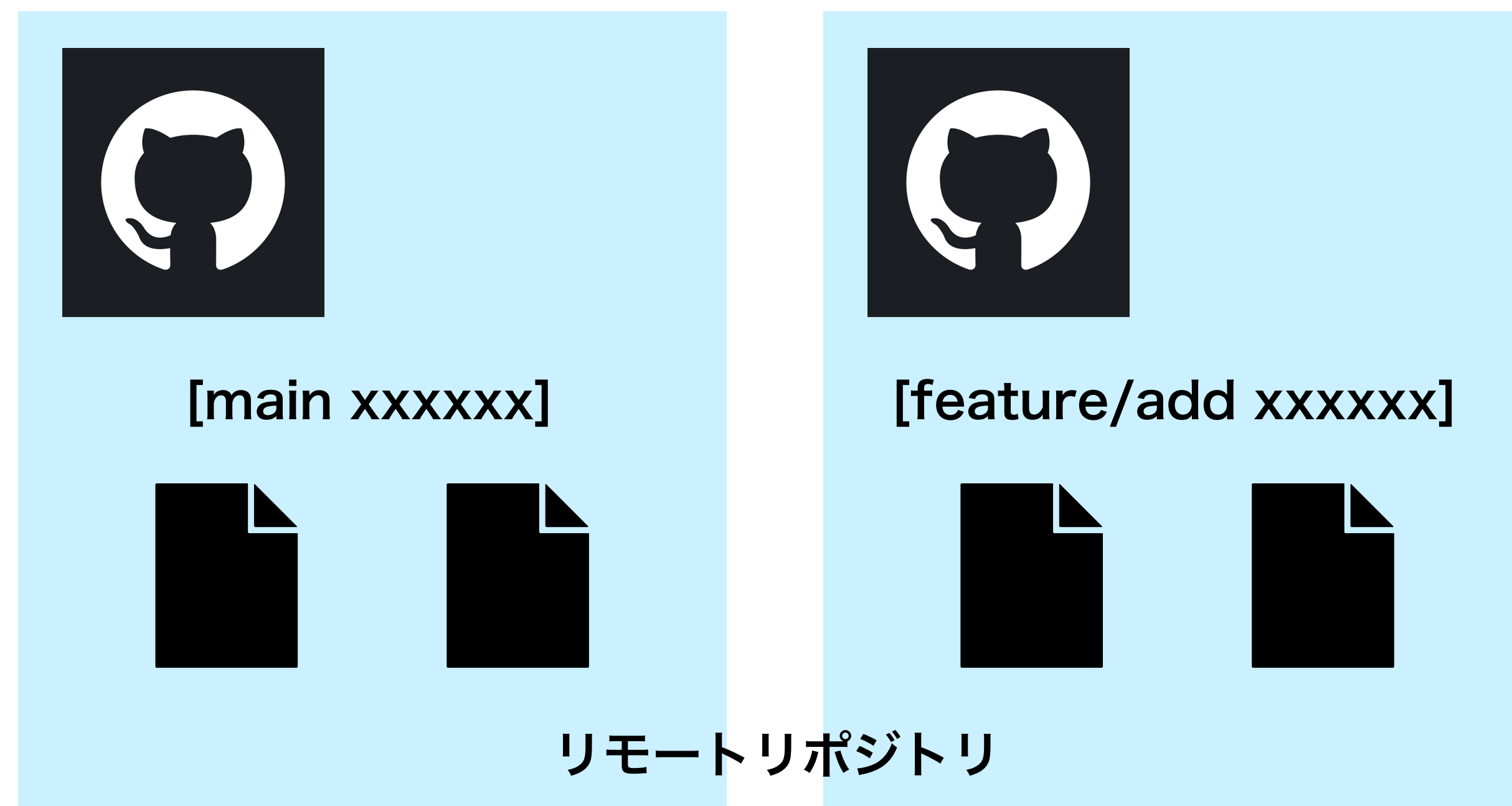
- ・ リモートブランチは**GitHub**など
リモート上で作成する
- ・ **分岐元のブランチ**を選択する
- ・ リモートブランチの作成前に
**現在のブランチに
全てPushしておく**



Create Branch

リモートブランチの作成

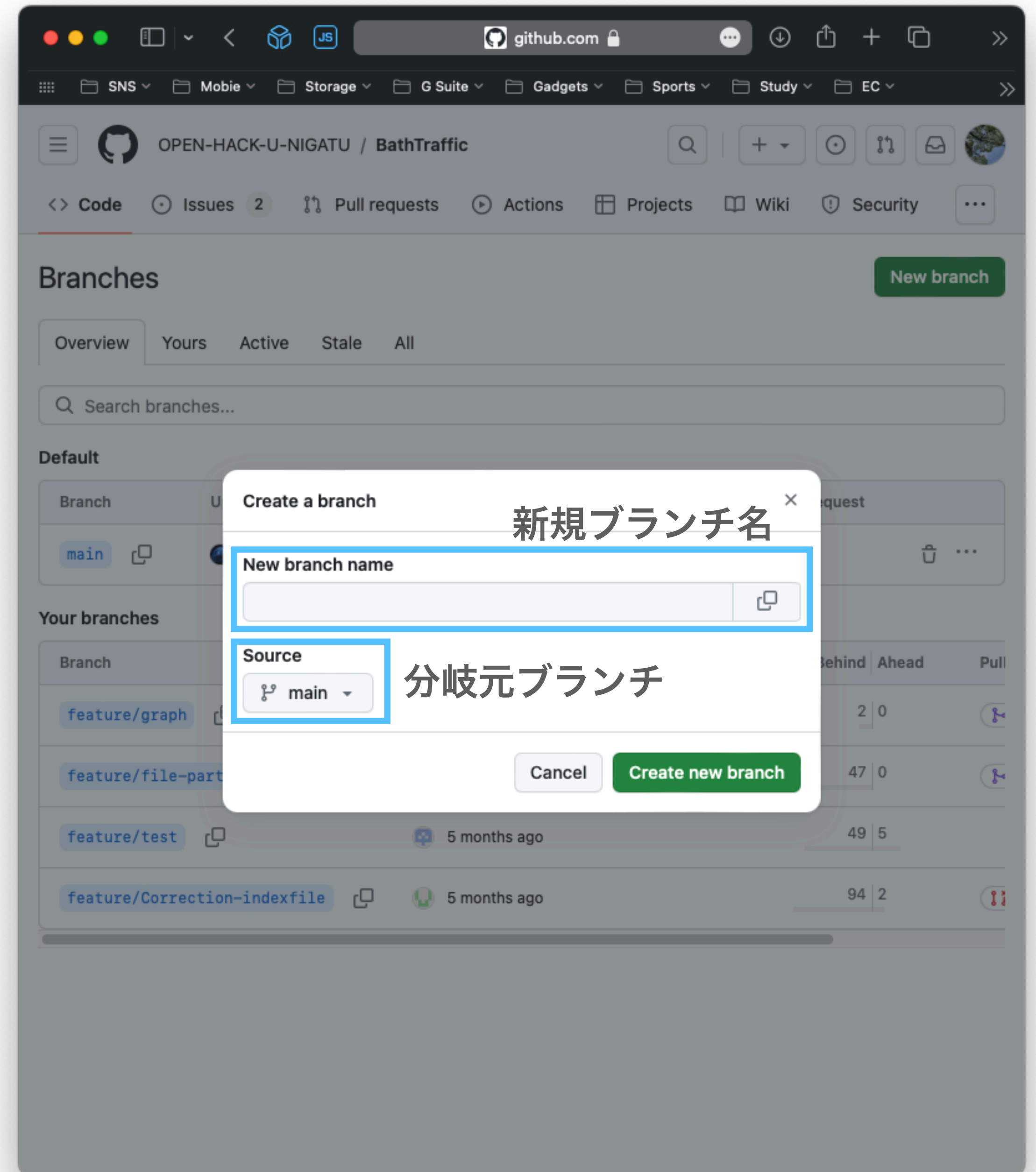
- ・ リモートブランチは**GitHub**など
リモート上で作成する
- ・ **分岐元のブランチ**を選択する
- ・ リモートブランチの作成前に
**現在のブランチに
全てPushしておく**



Create Branch

リモートブランチの作成

- ・ リモートブランチは**GitHub**など
リモート上で作成する
- ・ 分岐元のブランチを選択する
- ・ リモートブランチの作成前に
現在のブランチに
全て**Push**しておく

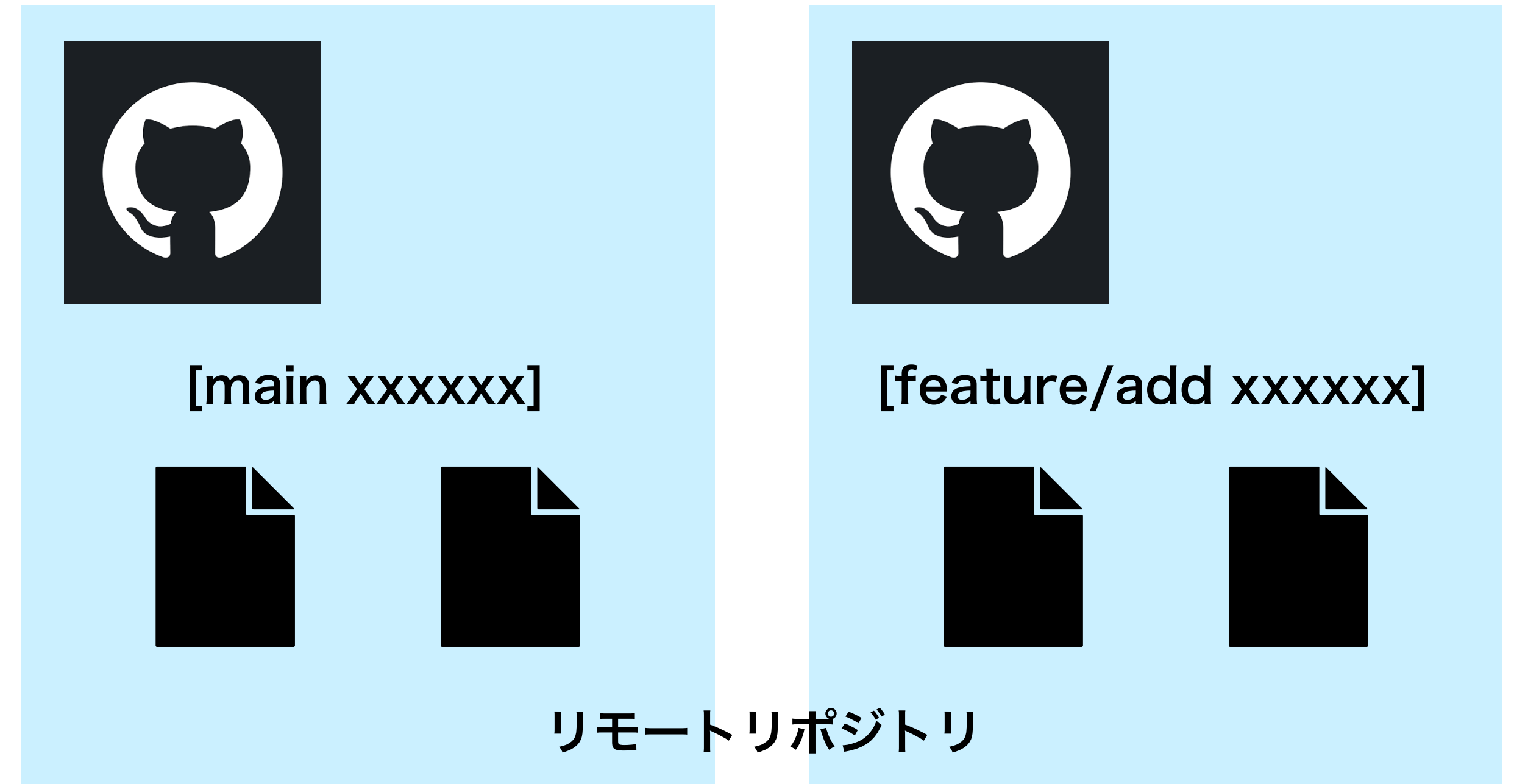


Checkout

ローカルブランチの変更

- ・ ブランチを新規作成するときは
"-b"オプションを付ける
- ・ Checkoutをする際には
今までのコミットが
全てPushされている必要がある

```
git checkout [ブランチ名]
```



Checkout

ローカルブランチの変更

- ・ ブランチを新規作成するときは
"-b"オプションを付ける
- ・ Checkoutをする際には
今までのコミットが
全てPushされている必要がある

```
git checkout [ブランチ名]
```

