



# チーム開発でのGitの使い方

## Gitとは？

Gitはバージョン管理システムの1つです。

Gitを使うことでソースコードの変更履歴が記録され、かんたんに過去のバージョンに戻したり、変更を共有することができます。

ソースコードの変更履歴を格納する場所のことを **リポジトリ** と呼びます。

## Gitのコマンド

Gitの操作はコマンドラインで行います。（ただし、GUIのGitクライアントを使うこともできます）

- [Gitチートシート](#)

## リポジトリのクローン

`git clone` コマンドを使うと、GitHub上のリポジトリをローカルにコピーすることができます。

誰かが作成したリポジトリを自分のPCにコピーして、開発を始めるときに使います。

```
git clone <GitリポジトリのURL>
```

## GUIのGitクライアント

GUIのGitクライアントを使うと、コマンドを打たずにGitの操作ができます。

EclipseでもGitの操作はできますが、専用のGitクライアントを使うことでより直感的に操作することができるので、おすすめです。

- [Sourcetree](#)（推奨）

- Fork (推奨)
- GitKraken
- GitHub Desktop

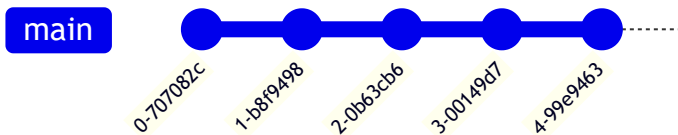
## Gitの概念

Gitの基本的な概念を図で説明します。



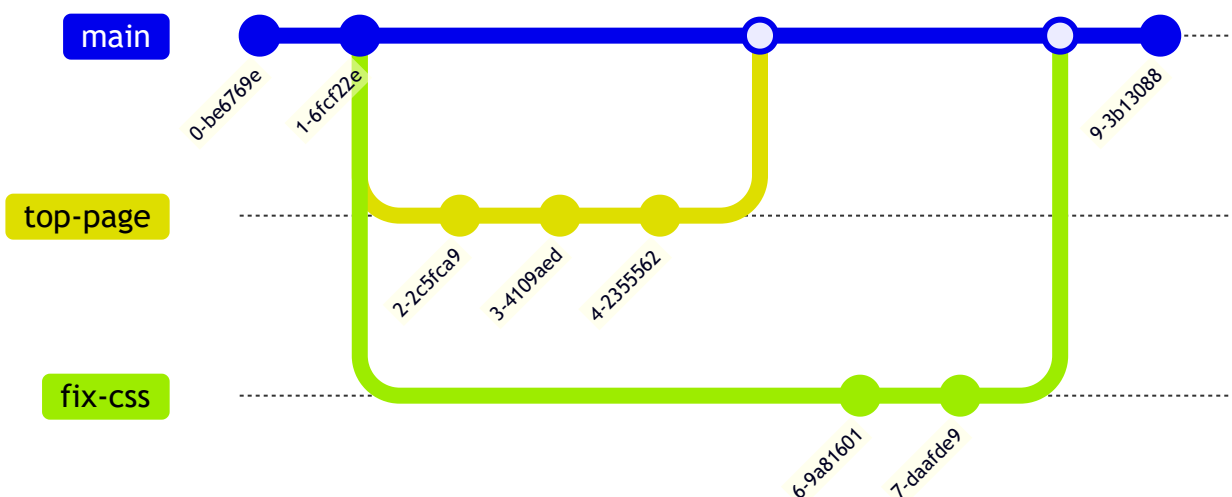
## ブランチの活用

1人で開発するときは、Gitのmainブランチで作業して、コミットするだけで問題ありませんでした。



しかし、このやり方だと、複数人で開発するときに問題が発生します。

Gitでは、ブランチを作成することで、他の人が作業しているブランチに影響を与えることなく、自分の作業を行うことができます。

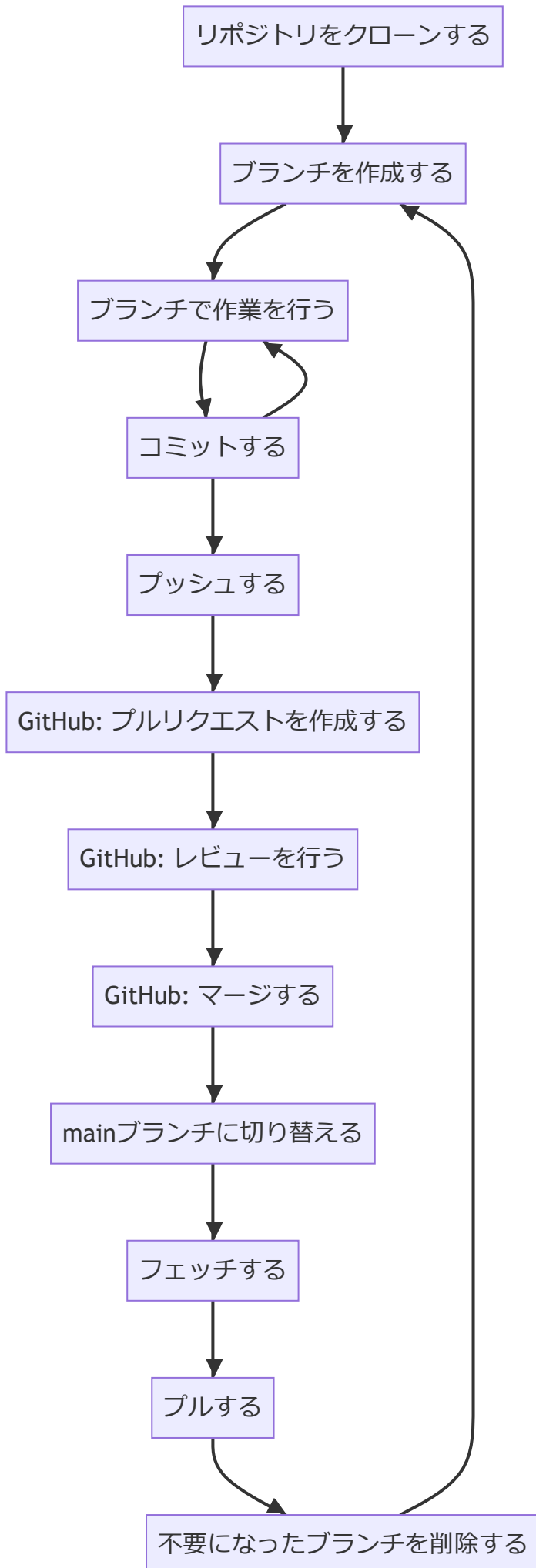


上記の例では、top-pageブランチとfix-cssブランチを作成し、それぞれで作業を行っています。

## Gitによるチーム開発の流れ

---

Gitによるチーム開発の流れを図で説明します。



# Gitの用語集

用語	説明
リポジトリ (repository)	ソースコードの変更履歴を格納する場所
クローン (clone)	リモートリポジトリをローカルにコピーすること
作業ツリー (working tree)	作業用のディレクトリ
インデックス (index)	作業ツリーの変更履歴を一時的に保存する場所
ローカルリポジトリ (local repository)	ローカルにあるリポジトリ
リモートリポジトリ (remote repository)	リモートにあるリポジトリ (GitHubなど)
ブランチ (branch)	作業用の枝分かれしたリポジトリ
コミット (commit)	変更履歴を記録すること
プッシュ (push)	ローカルリポジトリの変更履歴をリモートリポジトリに反映すること
フェッチ (fetch)	リモートリポジトリの変更履歴をローカルリポジトリに反映すること
プル (pull)	リモートリポジトリの変更履歴をローカルリポジトリに反映し、ローカルのブランチを最新の状態にすること
プルリクエスト (pull request)	作業したブランチをmainブランチにマージするためのリクエスト
マージ (merge)	ブランチを統合すること
ステージング (staging)	インデックスに変更履歴を保存すること
チェックアウト (checkout)	ブランチを切り替えること

