

Git について

インストール

Windows <https://gitforwindows.org/> からダウンロードします。画面の表示通りですが、エディタは以下を選択します。 **Use Visual Studio Code as Git's default editor**

デフォルトブランチは **Override the default branch name for new repositories** を選択します。

Mac は Homebrew でインストールします。まずは brew を使用できるようにします。 https://brew.sh/index_ja

M1 と M2 の場合は **brew -v** でバージョンが表示されない場合があります。その場合は brew がインストールされたらパスを通します。

```
echo 'eval "$(/opt/homebrew/bin/brew shellenv)"' >> ~/.zshrc
source ~/.zshrc
```

```
brew install git
```

Git でよく使う用語

- ローカルリポジトリ (local repository)
「ローカル」とは、自分のパソコンのことです。「リポジトリ」とは、保管庫のこと。ファイルは自分のパソコン内にあるローカルリポジトリで管理される
- リモートリポジトリ (remote repository)
リモートリポジトリの中に保存して、インターネット上の Web サービス (GitHub) でファイルを管理します。
- インデックス (index)
ファイルをローカルリポジトリに登録するには、まずはファイルをインデックスと呼ばれる場所に登録 (add) するところから始まります。
- コミット (commit)
インデックスに登録されたファイルをローカルリポジトリに登録することです。
- プッシュ (push)
自分のパソコン内のファイルを変更した場合は、まずはローカルリポジトリにコミットまで行う。つぎに、ローカルリポジトリの内容をリモートリポジトリにプッシュします。
- プル (pull)
プルを行うと、リモートリポジトリに保存されている内容が、パソコン内のローカルリポジトリにダウンロードされます。

- ブランチ (branch)
「ブランチ」とは、リポジトリ内で行うファイルの管理を、用途に合わせて分岐させる機能です。
- main ブランチ (main branch)
リポジトリには必ず 1 つのブランチが必要なため、まずはじめに main ブランチを作成することが一般的です。
- マージ (merge)
「マージ」とは、あるブランチで行った変更を別のブランチに適用することです。例えば、「開発用ブランチ」でバグの改修が終わった内容を「main ブランチ」にマージすることで、「main ブランチ」のバグが解消されます。
- クローン (clone)
例えば、GitHub にある既存のリモートリポジトリを更新したいが、自分のパソコンにローカルリポジトリがない場合にクローンをします。
- プルリクエスト (pull request)
「プルリクエスト」とは、自分が更新したファイルの内容を、他のチームメンバーにレビュー（更新内容が正しいかチェック）してもらう機能のことです。例えば以下のような流れです。
 1. ローカルリポジトリの main ブランチから開発用ブランチ (dev) を作成
 2. 開発用ブランチ (dev) のファイル 3 を削除してコミットした後に、開発用ブランチ (dev) をリモートリポジトリにプッシュ
 3. GitHub 上で開発用ブランチ (dev) を元にプルリクエスト (PR) を作成すると、チームメンバーにレビュー依頼の通知が届きます。

リモートリポジトリの作成

ローカルリポジトリで作成したデータをコマンドラインでリモートリポジトリに接続するハンズオンを行なっていきます。

コマンド操作の流れ

Git での管理を始める (=ローカルリポジトリ作成)

```
$ git init
```

ローカルリポジトリの設定を行う (GitHub を登録)

```
$ git config user.name GitHubのユーザーネーム  
$ git config user.email GitHubのメールアドレス
```

変更分をステージングにあげる

```
$ git add .
```

ローカルリポジトリにあげる

```
$ git commit -m "コミットメッセージ"
```

ステージング移動とコミットを一度に

```
$git commit -am "コミットメッセージ"
```

ローカルリポジトリに main ブランチを新規作成

```
$ git branch -M main
```

今いるブランチを確認

```
$ git branch
```

リモートリポジトリ (GitHub) を登録

```
$ git remote add origin リモートリポジトリのURL
```

リモートリポジトリにプッシュする

```
$ git push origin main
```

Git のプラグイン

git history

コミット履歴を確認するには、コマンドパレットから「git log」と入力し「Git: View History」を選択します。そうすると、コミット履歴が表示されます。

履歴の行を選択すると、前のコミットとの比較や、現在作業中のソースコードとの比較をすることができます。

- **コミット履歴の検索**

コミット履歴をコミットメッセージから検索することができます。

先ほどと同じように、コマンドパレットから「Git: View History」でコミットログを表示します。

- **ファイル単位の履歴が確認できる**

Git で管理しているファイルの上で右クリックします。

そして「Git: View File History」を選択します。

そうすると、そのファイルのコミット履歴が表示されます。「いつ追加した処理だったか？」など、ファイル単位の履歴が確認でき

- **行単位の履歴が確認できる** ファイル単位に確認しましたが、行単位でも履歴を確認することができます。行単位で確認する場合は「Git: View Line History」を選択します。