

チーム開発で使うGit/GitHub 講習

2023.11.12@SAPPORO OPEN DATA HACK

目次

- Git / GitHub CLI をインストールします
- Gitとは？
- GitHubとは？
- Git / GitHub 講習
- 実践GitHub

Git をインストール

For Windows

```
$ winget install --id Git.Git -e --source winget
```

For MacOS

```
$ brew install git
```

Homebrewが入っていないmacOSユーザー

以下のコードを実行してください

```
$ /bin/bash -c "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/master/install.sh)"
```

GitHub CLIをインストール

For Windows

```
winget install -e --id GitHub.cli
```

For macOS

```
brew install gh
```

インストールを待ちながら次へ..

Git / GitHub とは？

ここからの説明は、MIXIの23年度研修資料を流用します

https://speakerdeck.com/mixi_engineers/2023-git-training

実践GitHub

残りの時間で、GitHubを使ったチーム開発を体験してもらいます。

- 今日の講習参加者の中で3人のチームを作ります。
 - ハッカソンチームで参加されてる方はそのままOKです
- GitHub.ioでチームサイト仮を作ります

ミッション：チーム紹介サイトを作る

1. 各チーム、リーダーを1人決めてください
2. リーダーは、デモサイトのリポジトリをCloneし、共同開発の準備をしてください
3. 他のメンバーは完成したリポジトリをローカルにCloneする準備をしてください

デモサイトをCloneする

リーダーのみが作業してください

1. <https://github.com/mikan-foundation/sodh-git-lesson-demo>にアクセス
2. 右上の「<>Code」から、「Download Zip」でZipをダウンロード
3. 好きなフォルダに、sodh-git-lesson-demo フォルダを移動します
4. フォルダ名を好きな名前にします

リポジトリを立ち上げる

リーダーのみが作業してください

1. フォルダを右クリックし「ターミナル」または「PowerShell」を立ち上げます
2. `git init` を実行します
3. `git add .` を実行します
4. `git commit -m "first commit"` を実行します
5. `gh repo create` を実行し、すべてにエンターを押します

自分のGitHubリポジトリに、リポジトリができているはずです

開発メンバーを追加する

リーダーのみが作業してください

1. [設定]へ移動します
2. サイドバーの [アクセス] セクションで、 [コラボレーターと team] をクリックします。
3. [アクセスの管理] の右側にある [ユーザーの追加] または [Team の追加] をクリックします。
4. 検索フィールドで、招待する人のGitHubユーザー名を入力し、リストから一致する名前をクリックします。
5. [ロールの選択] で、Team または人に付与するリポジトリ ロールを選択し、 [リポジトリに名前を追加] をクリックします。

GitHub Pages の設定

リーダーのみが作業してください

1. GitHubで、サイトのリポジトリにアクセスしてください。
2. リポジトリ名の下にある [設定] をクリックします。[設定] タブが表示されない場合は、
[] ドロップダウンメニューを選び、[設定] をクリックします。
3. サイドバーの [コードと自動化] セクションで、[ページ] をクリックします。
4. [ビルドとデプロイ] の [ソース] で、[ソースからのデプロイ] を選択します。
5. [ビルドとデプロイ] で、ブランチ ドロップダウンメニューを使って、[main]を選びます。
6. 必要に応じて、フォルダー ドロップダウンメニューを使って公開元のフォルダを選択します。
7. [保存] をクリックします。

ここからは実践です

1. 各メンバーでリーダーのリポジトリをCloneして、member-X.htmlのファイル进行操作し、自分の名前と自己紹介を書き加えましょう
 - ブランチを必ず分けてみましょう。ブランチ名は `edit-{自分の名前}` にしましょう
2. コンフリクトを起こさずに、mainに対してPull Requestを作成してみましょう
3. メンバーの変更をチェックし、マージしましょう
4. GitHub Pages で変更が反映されているか確認しましょう

基本操作おさらい

1. `git branch {ブランチ名}` で作業用のブランチを作成
2. ファイルに変更を加えたら
3. `git add {ファイル名}` もしくは `git add .` で変更履歴を保存
4. `git commit -m "{コミットメッセージ}"` で変更箇所を記録
5. `git push origin {ブランチ名}` で変更をリモートにPush
6. GitHubのPull Requestからプルリクエストを作成
7. mergeしましょう
8. 作業が終わったらブランチの削除は忘れずに