【リポジトリにメンバーを招待する】

Githubのリポジトリに入る  settingをクリック

Collboratorsをクリック

Add peopleをクリックし

メンバーのアドレスを登録しAdd to repositoryボタンを押す

追加されたメンバーにメールが送られるので、  メールを開き、 acccept or decline をクリック Acccept invitationをクリックするとGithubが開く

【リモートリポジトリをローカルに複製して作業場所を作る】

\*mkdir=make directory=ディレクトリを作成

\*cd=change directory=ディレクトリを変更する

\*git clone=リモートリポジトリをローカルにまるごとコピー

これでローカル環境(自分のpc)にフォルダが作成されました。

【vs.codeでフォルダを開く】 \*各々フォルダ名は異なるので

自分のpc上に作成されたフォルダを開いてください

赤枠部分にカーソルをあてると矢印が出るので上に引っ張る →ターミナル画面が出てきます(コード入力画面と最下部の薄いグレー部分の境目)

【mainを複製した作業場所を作成(ブランチを切る)】

\*cd=change directory=指定したディレクトリに移動

\*git branch=現在のブランチ一覧を確認

これで作成したローカルリポジトリがGitHub上にもアップロードされたの で、GItHubの画面で確認してみましょう

【マージ完了済みのブランチは削除する】

リポジトリ名

\*あなたがいるブランチ(作業場所)はmainですよ

\*git checkout=指定したディレクトリに移動

\*git add . =全部のファイルを含んだフォルダを追加

\*git commt=保存する、-m ”commit message”=””部分に変更の内容を短く説明

\*git push origin=指定したブランチをリモートリポジトリ(GitHub)にアップロード

\*git checkout main =mainブランチに移動

\*git branch -d =指定したブランチを削除(delete)する

codeボタンを押す→SSHのコードをコピー

create pull requestをクリック  =どんな変更をしたのか、他の人にレビューを申請する

メンバーにURLを共有し、他のメンバーは変更内容を確認する →Merge pull requestをクリックしこの変更をマージ(結合)する

Pull request successfuly mergedとでたらマージが完了

git bashを開いて、lsを入力→enter

\*ls=list=ディレクトリの中身を確認するコマンド

mkdir test(任意のフォルダ名)→enter

cd test(任意のフォルダ名)→enter

git clone SSHのコピーしたコード→enter

vs.codeを開く>フォルダを開く> エクスプローラー(PCのフォルダ)>ローカルディスク(またはWindows) >ユーザー>user(もしくは自分の名前)>test(これがbashで作られた フォルダ) index.htmlに適当なコードを入れて保存

【Github操作画面】

GitHubの画面に戻り、赤枠部分のmainをクリックすると  先ほど作成したfeature/(任意の名前)が作成されていることが確認できました  選択を切り替えてfeature/(任意の名前)ブランチに移動する

ブランチの変更内容を、mainブランチに取り込んでよいか申請をする Compare&pull requestをクリック →create pull requestをクリック

どんな変更をしたのかタイトルと変更内容を入力する

GitHub画面でDelete branchボタンを押す vscodeに戻り、 git checkout main→enter

git branch -d feature/(任意の名前)→enter

ブランチ名がなければ削除完了

ターミナルに以下を入力していく

cd リポジトリ名→enter

git branch→enter

\*mainと表示される

git checkout -b feature/任意の名前→enter

git add .→enter

git commit -m “任意の名前(commit massge)を入力”

git push origin feature/任意の名前→enter

入力中にこんなエラーが出たら。。

git : 用語 'git' は、コマンドレット、関数、スクリプト ファイル、または操 作可能なプログラムの名前として認識されません。名前が正しく記述され ていることを確認し、パスが含まれている場合はそのパスが正しいことを確 認してから、再試行してください。 発生場所 行:1 文字:1 + git branch + ~~~ + CategoryInfo : ObjectNotFound: (git:String) [], CommandNotFoundException + FullyQualifiedErrorId : CommandNotFoundException

→ターミナルで git コマンドが認識されていない状態

確認方法: インストールされているか確認→git --versionを入力しバージョンが表示されれば 正しくインストールされている

・バージョンが出てこない場合 Git自体が正しくインストールされていない可能性があるので再インストールを実施

・バージョンが出てくる 環境変数(Path)が正しく設定されていない可能性があるので環境変数を変更する

根本対処:環境変数 PATH を再設

「スタートメニュー」→「環境変数」と検索→「システム環境変数の編集」を開 「環境変数(N)...」をクリッ Path をダブルクリック(ユーザー環境変数 / システム環境変数どちらでもOK 以下を追加(なければ)

C:\Program Files\Git\cm C:\Program Files\Git\bi OKで閉じて、PCを再起動(またはPowerShell再起動)

Pull requestの出し方例