

Git管理

Git管理について

Git管理の課題や他の課題で何度もGit管理について言ってきましたが改めてGit管理に気をつけてください。

- 作業を行う時はブランチを切る

masterへコミットしてしまうとリベートする作業が発生するなど必要以上に時間がかかってしまうようになります。

- masterにプッシュしない

作業したコードにバグがあり、それが作業全体に影響が出てしまう。

masterにプッシュできないように設定している会社、案件があればGitを使っているだけでそういった設定をしていない場合もあるので気をつける

- 作業目的とは関係のない変更を行わない

目的とは別の実装のコードを反映させる。

必要ない改行、コメントを反映させる。

`console.log()`などの不要なコードを反映させる。

Gitフロー

研修でGitを使っていましたがプルリクの作成、プルリクのコメントを元に修正と再レビュー依頼、マージ、リモートブランチからのプルと言ったことは行いませんでした。

実務では頻繁に行うので新しくリポジトリを作成して練習しておいてください。

コミット

研修の中でコミットメッセージなどまであまり言及はしていませんでしたが、こちらも非常に重要です。

多かったのが...

- Second、Thirdと言った文字でのカウント
- 課題修正

こう言った内容は第三者や後に自分で見返した時になんのコミットだったのか想像が付きません。

なのでコミットメッセージはできるだけ作業内容に沿った内容を書くようにしましょう。

また、コミットは作業がすべて終わってコミットではなく、段階ごとに分けて行うのがオススメです。

具体例

HTML: mainコンテンツの中にsectionがA、B、Cとある

CSS: HTMLの構造に沿ってスタイルが当ててある

JS: 機能E、D、Fを実装

作業手順はいくつかあると思います。

たとえば、sectionAのHTMLとCSSを作って1度コミット。それをsectionB、sectionCも同様に行なっていきます。

JSも機能ごとにコミットします。

こうすると。途中で戻りたくなった場合もキリの良いところで戻すこともできます。

また、第三者や後に自分で見返した時もこの部分の実装をしたのだと分かりやすいです。

GUIツール

研修ではコマンドを使用してGit管理を行なっていましたが、実務ではGUIツールを使用したGit管理を行うこともあります。

研修ではGitの理解を深めていただく為にコマンドを使用してGit管理を行なってもらいました。（GUIツールを使用するとクリックで簡単にできてしまうためGitフローの理解度が深まらない為）

代表的なツールが**SourceTree**です。

ブランチなどもボタンを押してブランチ名を入力すればブランチを切ることができ、ローカルのブランチもツリー表示されるので簡単に確認できます。

コミットも対象ファイルを画面上で選択、コミットしたい行だけでもコミットできるので便利です。

案件ごとにセットアップする必要があるのでご自身のPCにも入れて触ってみてください。

[SourceTree](#)