

github-flow_test

GitHub flowの練習用リポジトリ

手順

1. ブランチを作る
2. ファイルを編集
3. 不具合がなければコミット、プッシュする
4. プルリクエストを送る
5. レビューしてもらう
6. 何にもなければマージする
7. ブランチを削除する
8. ローカルにプルする

おすすめ VScode 拡張機能

1. [Git Graph](#)
2. [GitHub Pull Requests and Issues](#)
3. [GitLens -- Git supercharged](#)

GitHub Pull Requests and Issuesは、VScodeからプルリクエストを送ることと、マージすることができる。

コマンドでGitを操作するための基本コマンド一覧

実務でよく使うコマンド

> リモートリポジトリをローカルにクローンする

`git clone` コマンドで、開発リポジトリをローカルに取得します。
ディレクトリ名は、「project」とします。

```
git clone https://github.com/west-hiroaki/git-test-for-qiita.git project
```

このコマンドは、HTTPSでクローンする場合です。
SSHでクローンする場合は、[ここ](#)を参照してください。

> ブランチを確認する

`git branch -a` コマンドで、ブランチ状態を確認しておきます。

`-a` パラメータを付けないとローカルブランチのみですが、付けるとリモートブランチも確認できます。

リモートには、「master」ブランチがあることが分かります。

```
git branch -a
```

```
* master
remotes/origin/master
```

> ブランチを作成する

`git checkout -b` コマンドで、「master」ブランチから派生させた開発ブランチ「feature/add-image」をローカルに作成します。

```
git checkout -b feature/add-image remotes/origin/master
```

```
Branch 'feature/add-image' set up to track remote branch 'master' from 'origin'.
Switched to a new branch 'feature/add-image'
```



この例だと、明示的に派生元ブランチを指定していますが、省略すると現在のブランチから派生されます。

作成と同時に、アクティブブランチが作成した開発ブランチに切り替わっています。ローカルブランチを確認すると、「feature/add-image」ブランチに切り替わっているのが分かります。

```
git branch -a

* feature/add-image
  master
  remotes/origin/master
```

> 修正したファイル差分を確認する

`git diff` コマンドで、修正したファイル差分を確認します。
追加した内容が間違っていないことを確認します。

```
git diff index.html
```

```
diff --git a/index.html b/index.html
index 08b06f9..830abe4 100644
--- a/index.html
+++ b/index.html
@@ -9,5 +9,6 @@
     <hr>
     <p>github hello!!</p>
     <button>click!</button>
+    
   </body>
 </html>
```

> ワークツリーからインデックスに移動する

`git add` コマンドで、修正ファイル (`index.html`) を、ワークツリーからインデックスに移動します。

```
git add index.html
```

> インデックスの状態を確認する

`git status` コマンドで、インデックスの状態を確認します。
修正したファイル以外がインデックスに入っていないことを確認します。

```
git status
```

```
On branch feature/add-image
Your branch is up to date with 'origin/master'.

Changes to be committed:
  (use "git reset HEAD <file>..." to unstage)

        modified:   index.html
```

「index.html」以外がインデックスに入っていないことを確認しました。

> インデックスのファイル差分を確認する

`git diff --cached` コマンドで、インデックスのファイル差分を確認します。

「commit」前に、ファイルの差分を最終確認するためです。
この例だとあまり意味がないですが、実際のプロジェクトだと修正ファイルや差分も多く、間違っ
た修正を入れるわけにはいきませんので重要です。

```
git diff --cached index.html
```

```
diff --git a/index.html b/index.html
index 08b06f9..830abe4 100644
--- a/index.html
+++ b/index.html
@@ -9,5 +9,6 @@
     <hr>
     <p>github hello!!</p>
     <button>click!</button>
+    
  </body>
</html>
```

> コミットする

修正内容の確認ができたので、`git commit` コマンドで、インデックスのファイルをブランチに反
映します。

```
git commit -m '画像を表示するように修正しました'
```

```
[feature/add-image 1700755] 画像を表示するように修正しました  
1 files changed, 1 insertion(+)
```

> ブランチのログを確認する

`git log` コマンドで、ブランチのログを確認します。
ログの一番上に、さっき行ったコミットが追加されていることを確認できます。

```
git log
```

```
commit 170075502ca32ea1f37641ee22ee7cf0551a49ac (HEAD -> feature/add-image)  
Author: Souta Sugimoto <s-sugimoto@softnext.co.jp>  
Date: Sat Jul 9 10:52:26 2022 +0900
```

画像を表示するように修正しました



ちなみに、まだgitに慣れていない時は情報が多すぎると思いますので、
`--oneline` オプションを付けると見やすくなっていいかもしれません。

> プッシュする

`git push` コマンドで、リモートの開発ブランチに反映します。

```
git push origin feature/add-image
```

```
Enumerating objects: 69, done.
Counting objects: 100% (69/69), done.
Delta compression using up to 4 threads
Compressing objects: 100% (59/59), done.
Writing objects: 100% (69/69), 52.39 KiB | 2.49 MiB/s, done.
Total 69 (delta 14), reused 0 (delta 0)
remote: Resolving deltas: 100% (14/14), done.
remote:
remote: Create a pull request for 'feature/add-image' on GitHub by visiting:
remote:   https://github.com/sugimoto-0419/github-
remote:   flow_test/pull/new/feature/add-image
remote:
To github.com:sugimoto-0419/github-flow_test.git
* [new branch]      feature/add-image -> feature/add-image
```



追跡ブランチを設定しておけば、`git push` だけで「push」可能です。

> マージする

`git merge` コマンドで、開発ブランチを、「master」ブランチにマージします。
まず、「master」ブランチに移動します。

```
git checkout master
```

```
Switched to branch 'master'
Your branch is up to date with 'origin/master'.
```

次に、「feature/five-to-ten」ブランチをマージして、リモートに「push」します。

```
git merge feature/add-image
```

```
Updating 301b813..1700755
Fast-forward
 index.html      | 1 +
 1 files changed, 1 insertion(+)
```

リモートに「push」する。

```
git push origin master
```

> ブランチを削除する

`git branch -d` コマンドで、不要になったローカルの「feature/add-image」ブランチを削除します。

```
git branch -d feature/add-image
```



今回は、削除対象ブランチが「master」ブランチにマージされているので

`git branch -d` コマンドで削除できますが、マージされていない場合は削除できません。（ブランチ指定ミスで消してしまわないためです）

> リモートブランチを削除する

同様に、「`git push -d`」 コマンドで、リモートブランチも削除します。

```
git push -d origin feature/add-image
```

参考サイト

実務でどんな git コマンドを使っているか振り返ってみる