

git でブランチを使う Ver1.0

2021/12/06 Mon.

Seiichi Nukayama

1 こんな場合

想定されるケース

自宅でいろいろとコードを書いて試してみたいけれど、翌日は今日やったところから始めたいので、今日まで書いたコードは変更したくない。

つまり、今日講習で書いたコードを使っていろいろ試してみたいが、コードを変更したくない。となると、今日やったところまでのプロジェクト・フォルダを別のワークスペースにコピーして、そこでやる。そういう方法が考えられる。

もちろんそれでもいいのだけれど、Git の仕組みを使ってやることもできる。

Git の状況

Git で管理しているフォルダ: "LaLa-work"(ワークスペース)
現在のブランチ: "main"

2 計画

以下のような計画でやる。

1. 新しく "rensyu" というブランチを作成する。
2. "rensyu" ブランチでいろいろと試してみる。
3. "rensyu" ブランチに加えた変更を保存する。(Git にアップする)
4. 元の "main" ブランチに戻る。

3 新しく "rensyu" というブランチを作成する

3.1 現在のブランチ

現在のブランチは以下のコマンドで確認できる。

現在のブランチを確認

```
> git branch  
*main
```

現在は main ブランチのみである。

3.2 ブランチを作成する

ブランチ関係のコマンドは、とりあえずは以下。

ブランチの作成

```
> git branch rensyu  ("rensyu" というブランチを作成)
```

現在のブランチを "rensyu" にする

```
> git checkout rensyu
```

ブランチを作成するとともに、そのブランチをカレントブランチとする

```
> git checkout -b rensyu
```

"rensyu" ブランチを削除する

```
> git branch -d rensyu
```

"rensyu" ブランチの名前を "work" に変更する

```
> git branch -m rensyu work
```

3 番目の" ブランチを作成するとともに、そのブランチをカレントブランチとする" が手っとり早い。

```
> git checkout -b rensyu
```

確認する

```
> git branch
main
* rensyu
```

3.3 ブランチをプッシュする

このブランチでいろいろ試したあと、リモート (Github) にアップしたいとする。

`git status` とすると、現在のブランチが表示されるので、そのブランチを指定してプッシュすることになる。

(例) 現ブランチが "rensyu" である場合。

```
> git status
On branch rensyu
```

git status とすると、現在 rensyu ブランチであることがわかる。

プッシュする場合の流れは以下である。

```
> git add .(ピリオド)
> git commit -m "Main.java を追加"
> git push -u origin rensyu
```

最後の push コマンドのところで "rensyu" ブランチを指定している。

4 学校でブランチを作成し、Github にプッシュした。それを自宅でもりこみたい

こういう状況を想定している。

学校

```
> git branch
main
rensyu
```

学校でブランチを作成したので、学校で git branch とすると、main と rensyu の 2 つのブランチが表示される。

Github

```
main
rensyu
```

学校でブランチを作成したあと、Github にプッシュしたので、Github には main と rensyu が存在している。

自宅のパソコンで以下のコマンドを実行すると、以下のような表示になるはずである。

```
> git branch -r
origin/HEAD -> origin/main
origin/main
origin/rensyu
```

この `git branch -r` というコマンドは、リモート (Github) に存在するブランチを表示するコマンドである。Github には当然 main と rensyu の 2 つのブランチがある。

自宅のパソコンで以下のコマンドを実行すると、以下のような表示になる。

```
> git branch
* main
```

まだ rensyu ブランチを取り込んでいないので、当然である。

リモートの "rensyu" ブランチを取り込み、それをカレント・ブランチとする

```
> git checkout -b rensyu origin/rensyu
```

こののち、以下のコマンドを実行して、確認する。

```
> git branch
main
* rensyu
```

5 "rensyu" ブランチの変更を"main" に取り込む

"rensyu" ブランチは試しにいろいろやってみるブランチなので、それを "main" ブランチに取り込むことはないかもしれないが、そのやり方は以下である。

まず、現在のブランチを確認する。

```
git branch
```

以下のように、"main" ブランチに戻ってから "rensyu" ブランチの内容を取り込む

リモートの"rensyu" ブランチを取り込み、それをカレント・ブランチとする

```
> git checkout main
> git merge rensyu --no-ff
```

--no-ff オプションについては、以下を参照。

- git の merge --no-ff のススメ
- git merge オプションの --ff, --no-ff, --ff-only の違い

6 リモート (Github) に存在する"rensyu" ブランチを削除する

"rensyu" ブランチもその役目を果たし、削除する日がやってきた。

この場合、ローカルの"rensyu" ブランチを削除するとともに、リモートの"rensyu" ブランチも削除しなくてはならない。

手順は以下である。

リモート (Github) に存在する"rensyu" ブランチを削除する

```
> git branch -d rensyu      (ローカルの"rensyu" ブランチを削除)
> git branch -r             (リモートのブランチを確認)
origin/HEAD -> origin/main
origin/main
origin/rensyu
> git push --delete origin rensyu  (リモートのブランチを削除)
```