1. やっぱりいいものだったよGit

1.1. この記事を書くきっかけ

どうもやっとPython講習会終わりました、たろうです。
現在は就活真っ只中であります。

就職活動の情報を仕入れると、自身のポートフォリオを作っておくと良いと
言う話を聞いたので自分もポートフォリオサイトを作ろうと思いました

ただ自分にはウェブサイトの作成する経験がないため、
いろいろ探してみたらこのようなサイトを発見（Django Girls Tutorial）

このサイトを見ながらpythonのDjangoを使って
ブログのようなものを作ろうと考えました

そのサイトを読み進めると、「このサイトでは**Git**を用いて開発を進めます」と言われ、
苦手意識が強いGitでしたがこのままではいけない（おそらく社会人でも使う）
と思いGitを用いた開発を始めました

その個人開発内で、「あれGitって便利じゃん、、、、、」
となったのでそのことについてかこうかなと思います

1.2. Gitって？

Gitとはファイルなどへの変更履歴を記録しそのバージョンを管理するためのツールで、
特に複数人でシステム開発を行う際に用いられます。いつ、誰がソースコードのどこを変更したのか、
編集するべき最新バージョンはどれかなどを管理するためのものです。

ここでは概要だけと言うことで このサイトより抜粋

1.3. ここがいい

ではここから自分がいいなあと思った箇所について、3つほど取り上げたいと思います

1.3.1. 前の状態に戻ることができる

このことは、Gitを使うことにおいてはもう当たり前のことだと
思うんですけどもすごく便利だと感じました。（初心者感）

例として、開発中のシステムにエラーが発生した場合、開発のどの過程でエラーが発生したのかを切り分けて、エラーが起こる前の状態に戻す必要があります。

そんな時はGitでは特定のコミットを指定し、元の状態に戻ることが可能

```
$ git log //戻す対象のハッシュ値を調べる
commit **********************

git reset --hard ハッシュ値
```

これでね
そして一回戻ってもう一回作り直せればOK（暴論）。

このことに関して自分にとって何が良かったのか、話しておきます。

開発の際にエラーが発生、そのエラーを解決するために様々な
ライブラリをインストールし改良したところなんとか解決。
じゃあmasterブランチにmergeしましょうねといったときに、
大量のmerge conflictが発生。どうやらライブラリをインストールしたことが原因の模様？

正直解決する気力がわかないときに、Gitは戻れることを思い出し、
ライブラリをインストールする前に戻ってから作り直しました。
ちなみに解決はDjangoのアップデートすればいいだけでした。

まあ使い方として正しいか正しくないかは目を瞑ってくださいな（笑）。

1.3.2. 大人数の開発に向いている

これもGitのいいところでよく言われていることだけども、、、
大人数での開発に向いています。
大人数開発のメリットとして

- 開発ごとにブランチを切って平行で開発を進めることは強い
- また新規メンバーに対して、すぐに開発できるよう参加させることが容易
- またそれぞれの成果を反映させることも容易にできる
- レビューがしやすい

思いつく限りですとこんなところでしょうか。（まだまだたくさんあるとは思いますが）
そういえば実務先では、バージョン管理どうしてたっけなあ。

1.3.3. 本環境に移すこと、アップデートすることが簡単

開発中結構感動したこと、
Pythonで作成したDjangoやFlaskのアプリケーションをリリースして
全世界に公開できるサービスを使って実際に作ったサイトを公開しよう
とした時、実際に作ったものをその環境に移す必要がありました

そうなった場合Gitを知る前の自分だと、
「scpとか使わなきゃいけないのかなあ、面倒なこと極まりない」

と考えていたのですが、Gitを知ってからだと杞憂でした

え、じゃあどうするのかってGitHubと組み合わせながら使うのです。

手順をサクッと説明します

1. ローカル環境で構築したものをリモートのリポジトリ(Github)に `git push`します。

```
///ローカル環境
git push origin master
```

2. リモート環境(Github)で`push`したものを`merge`し、
   今度は本番環境でリモート環境(Github)のリポジトリに対して`clone`する

```
///本番環境
git clone https://github.com/Gitにおける自分の名前/リポジトリ名
```

これでバッチリ！
アップデートしたい場合はリモートリポジトリに対して

```
git pull origin master
```

このコマンドを打つだけでOK。
と、こんな感じですごく本番環境への移行と、アップデートができて楽だなあと感じました。

1.4. 結び

今回は個人開発をしながらGitについていいなと思ったことを短いながらあげてみました。
こう見返すと酒井さんに対して申し訳ない気持ちでいっぱいになる。
いや、本当にすみませんでした（@Python講習会）

次回は自分のおすすめ仮面ライダーか名場面についてまとめてみようかなと思います。