Sun. 2020/05/17 @Online

初心者向けgit講習会

福岡市 EngineerCafe

git

今回の講習会の目標

基本的なgitコマンドが使えるようになる

タイムテーブル

13:30-14:00 Gitの基礎

14:00-14:30 Gitを実際に扱ってみる

14:30-15:00 Githubとの連携

15:00~ 質問対応

自己紹介

田中聡至（福岡市EngineerCafe staff）

好きな技術、言語:

https://github.com/alt9800

アンケート

[事前]

お使いのPC(OS)は？

ご職業は？　エンジニア/学生/その他

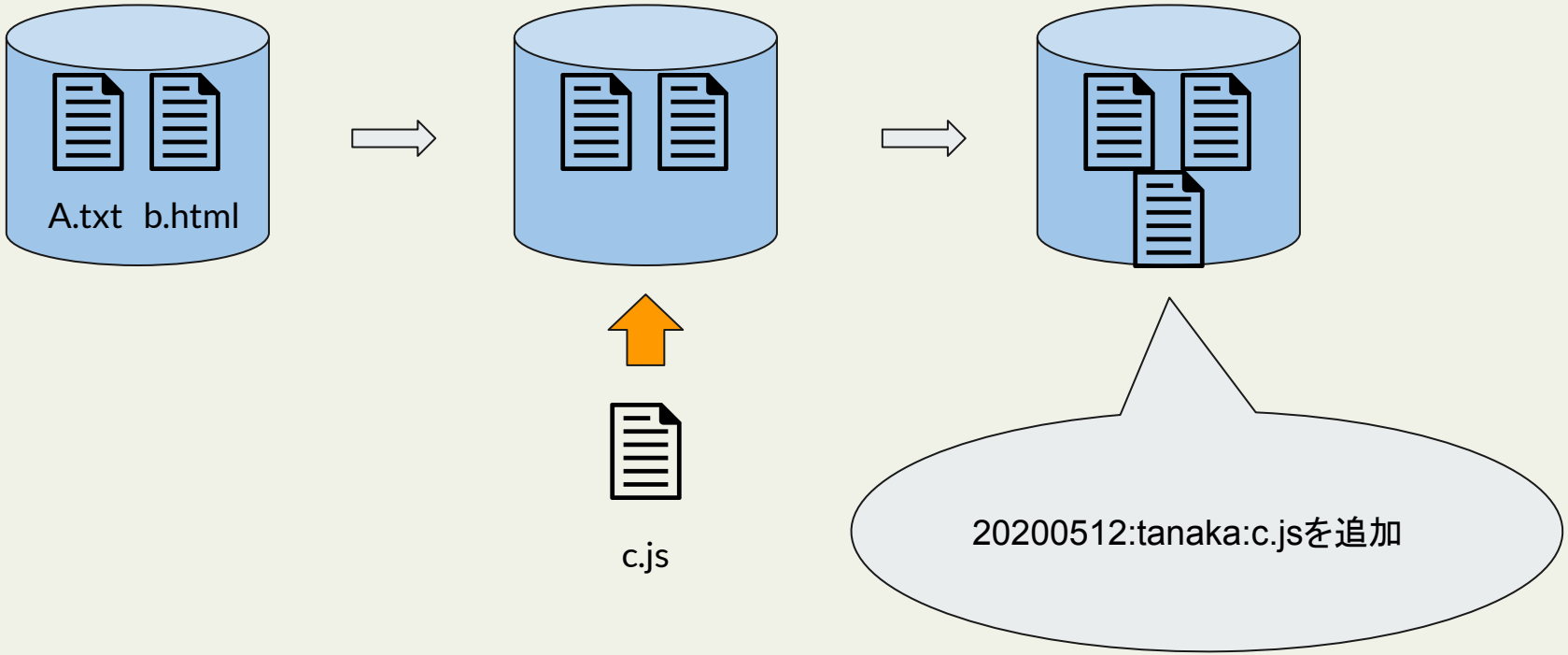
[事後]

gitについてどれくらい理解できたか

聞きたかったこと

後日コンパスページでフィードバックします

差分管理という考え



差分管理の例

- Google slide

差分管理の例

- Wikiの差分管理システム

PukiWiki / Download の変更点

Top / PukiWiki / Download

[トップ] [編集 | 差分 | バックアップ | 添付 | リロード] [新規 | 一覧 | 検索 | 最終更新 | ヘルプ | ログイン]

- 追加された行はこの色です。
- 削除された行はこの色です。
- PukiWiki/Download へ行く。
- PukiWiki/Download の差分を削除

```
#author("2019-03-01T02:30:52+09:00","default:webadmin","webadmin")
#author("2020-03-30T04:13:55+09:00","default:webadmin","webadmin")
* PukiWikiのダウンロード[#h47af29b]

|CENTER:SIZE(18):PukiWikiリリース版の最新のものは''[[1.5.2>./1.5.2]]''です。&br; → [[PukiWiki/Download/1.5.2]]|
|CENTER:SIZE(18):PukiWikiリリース版の最新のものは''[[1.5.3>./1.5.3]]''です。&br; → [[PukiWiki/Download/1.5.3]]|


#br
- 1.5系 (HTML5)
#ls2(PukiWiki/Download/1.5,reverse)

#br
- 1.4系 (XHTML 1.1)
#ls2(PukiWiki/Download/1.4,reverse)

#br
- 1.3系 (HTML 4.01 transitional)
#ls2(PukiWiki/Download/1.3,reverse)

#br
- 開発版
-- [[dev:開発版]]
```

初期設定サポート(Git for Windows)

GitをWindowから扱うためには、Linuxのエミュレータ、もしくは互換レイヤーのインストールを行うのがシンプルベターです。

[Git for Windows] > https://gitforwindows.org/

Git for WindowsにはLinuxで言うところの「端末(コンソール)」やMacの「ターミナル」に近い仕組みの「**Git bash**」が含まれています。

Windowsユーザーに限っては、最終的にはWSLを用いてUbuntu互換の環境を作る方が実りが多いと思いますのでオススメしておきます。

念のため初めてbashを扱う人のためのメモ...

```
# ディレクトリを変更する（cd .. とすると階層を一つ遡れる）

cd 〈移動したいパス〉

# 今いるディレクトリを確認する

pwd

# ディレクトリに含まれるファイルを確認する

ls

#  新しいファイルを作る

touch 〈新しいファイルの名前〉
```

Mac版初期設定の確認

```
$ git --help
```

もしくは

```
$ git --version
```

と入力し、使い方(usage)や情報が表示されることを確認

初期設定を確認(ここからは共通...)

Gitを初めて利用される場合はメールアドレス等の要素の登録が必要になります。

```
$ git config --global user.name "あなたの名前"

$ git config --global user.email "foo@example.com"

$ git config --global core.quotepath false
```

[参考]
Mac Git 初期設定 - Qiita
https://qiita.com/ucan-lab/items/aadbedcacbc2ac86a2b3

リモートリポジトリの例

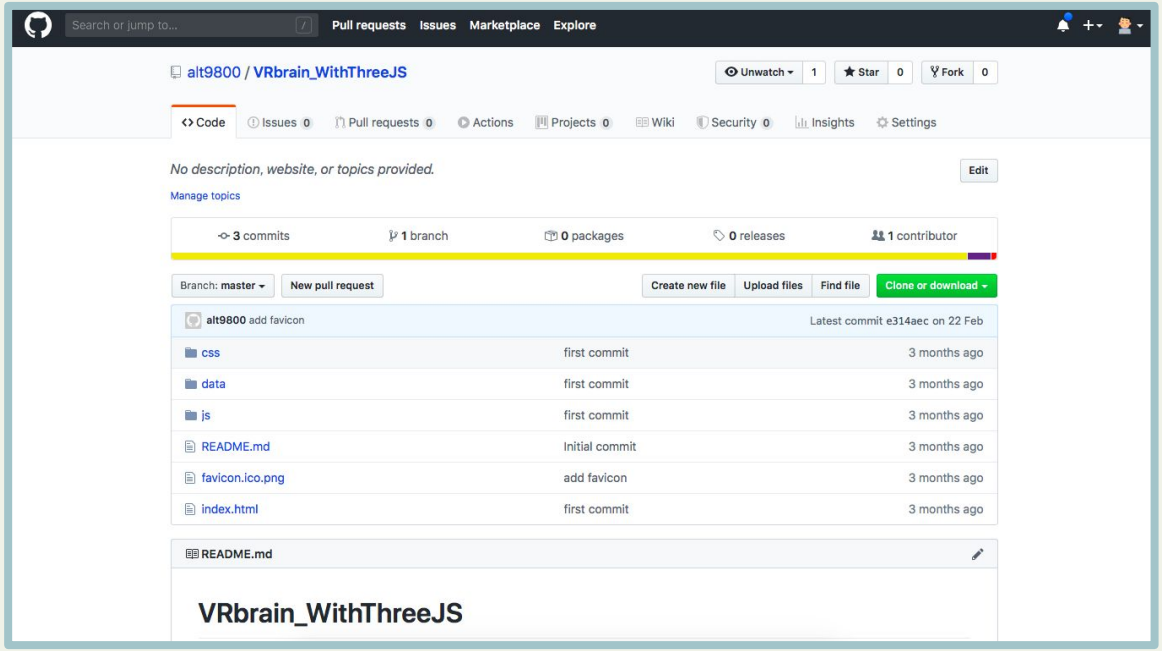
GitHub社が提供しているサービス。様々なサービスと連携できる。

OSSとして提供されている。オンプレミスでの運用もできる。

ソースコード管理および、作業進歩共有もできる。

Githubについて

Githubはgitで管理するプロジェクトを他者と共有し、多人数による開発を支援するためのプラットフォームです。



Githubとの連携

Git 自体は自身のPCやNASなどにリモートリポジトリを設けたり、ローカルリポジトリだけでソースコードの管理をすることもできますが、

今回は、多人数でリモートリポジトリを用いてソースコード共有をすることをゴールにしていますので、Githubとの連携を前提で進めていきます。

SSH認証

Githubをリモートリポジトリとして利用するためにはsshによる通信が必要になります。

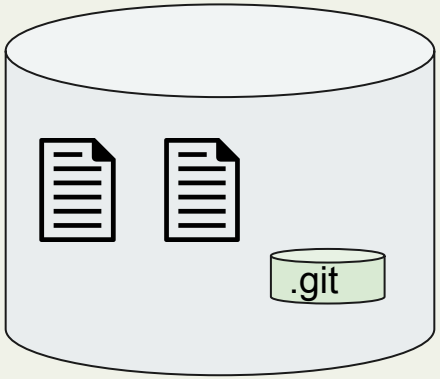
以下を参考にgithubとの連携の一歩を踏み出しましょう！！

- **Generating a new SSH key and adding it to the ssh-agent - GitHub Help**
  https://help.github.com/en/github/authenticating-to-github/generating-a-new-ssh-key-and-adding-it-to-the-ssh-agent
- **GitHubでssh接続する手順~公開鍵・秘密鍵の生成から~ - Qiita**
  https://qiita.com/shizuma/items/2b2f873a0034839e47ce

gitの用語（基本）

- リポジトリ（repository） -> gitが適用されたフォルダ（ディレクトリ）
- ブランチ（branch） -> 履歴が更新された軌跡
- コミット（commit） -> 変更履歴を残すこと
- add
- push
- pull
- merge
- fetch
- conflict
- pull request

リポジトリについて



上のような感じで、.gitが置かれたディレクトリがgitの管理下に置かれ、この中での変更の情報が.gitに差分として保持されていく

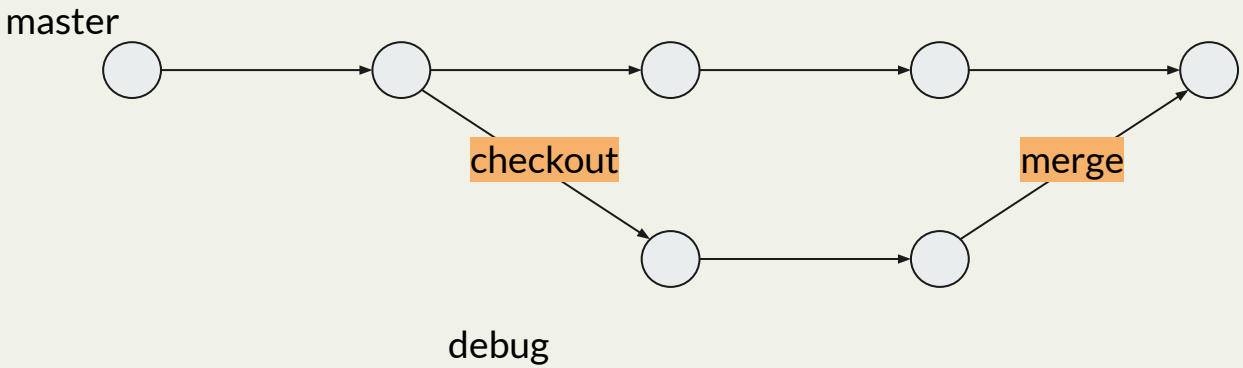
ちなみに...

不要になったリポジトリを削除したい！

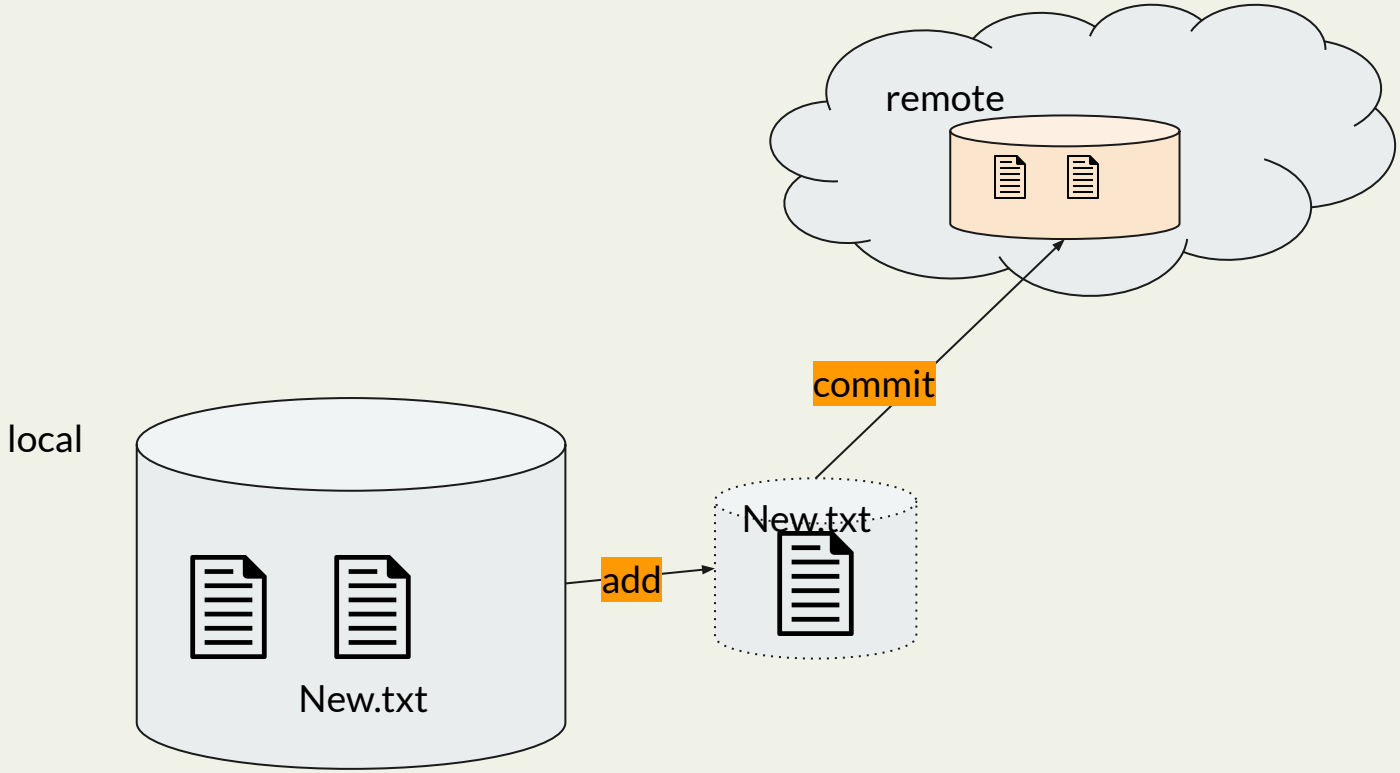
->`ls -a`と入力すると .git が隠しディレクトリとなっていることがわかると思います。これを rm するとそのプロジェクトがgitの管理下から離れます。

よく使うコマンドについての説明

Branch



add



Push / Pull

- **Push** は ローカルリポジトリからリモートリポジトリに情報をコピーするためのコマンド

  `$ git push origin <branch_name>`

  しばしば

  `$ git push origin master`

  とmaster branchの情報を取り込む

- Pull は リモートリポジトリからローカルリポジトリに情報をコピーするためのコマンド

  `$ git pull`

  厳密にはpullは fetch + merge を行なっている

“git log”あるいは “tig”

```
commit 681b3b3ecd286aa0aba28ade6f799ca2c99895bf
Author: alt9800 <[redacted]
Date:   Sun Dec 1 13:30:23 2019 +0900

    ローカルの環境にgithubからpullしたものをlolipop!用に鍵を作り直して準備

commit fea343de980f226f58d8e8a2dac3fbc036423851 (origin/master, origin/HEAD)
Merge: 9247f25 2fbd60a
Author: [redacted]
Date:   Mon Sep 30 17:18:17 2019 +0900

    Merge pull request #43 from hirok803/work01

    Work01  #1

commit 2fbd60a964aa4b32850f9794aa039c4cb5bb1901
Author: [redacted]
Date:   Mon Sep 30 16:44:16 2019 +0900

    Deviseの日本語化を実施
```

gitハッシュ

あるブランチをコミットする際、一意につけられたその地点(commit)を示すための固有名詞
<例>
commit 681b3b3ecd286aa0aba28ade6f799ca2caa23fbc34

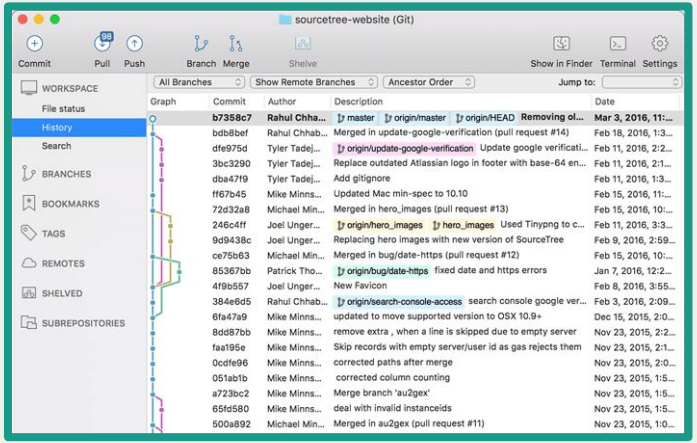
これを指定することによってその時の状態をリポジトリに再現することができる

チートシート

コマンド	操作の意味
git add X ...	ステージングエリアに変更を記録
git commit	変更を反映
git push origin master	リモートリポジトリに変更を反映
git log	変更履歴の表示

ここまでの内容でわからない概念や、もう一度聞きたい説明はありますか？

GUI での操作について



Git/Github自体は

> Sourcetree

> Github for Desktop

といったツールでGUIでの操作ができたりします。

参考になるサイト

- Progate -> https://prog-8.com/
  環境設定のドキュメントなどもアリ
- サル先生のgit入門 -> https://backlog.com/ja/git-tutorial/
- leaen git branching -> https://learngitbranching.js.org/?locale=ja
  git のブランチ操作をブラウザでトレーニングするサービス
- gitかるた ->https://whiims.github.io/git-karuta.html
  boothで購入できる副読本(¥500-)は一見の価値アリ