

# Git/GitHub 超入門

GitHubでたのしいチーム開発

# 本日のゴール

- Gitたのしい！
- GitHubでチーム開発気持ちい！



git



git

ってなんやろか...

# 分散バージョン管理システム

...よけいわからん

要すると

分散

して、

バージョンを管理する

システム

のこと

「バージョン管理」って...?

普段やつてること

# 普段やっていること

メモ.txt

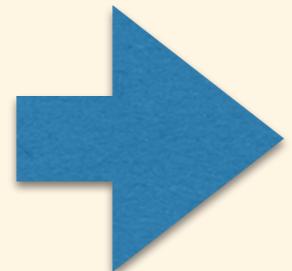
メモ

ID: hoge

# 普段やっていること

メモ.txt

メモ  
ID: hoge



メモ.txt

メモ  
ID: hoge2015

編集中…

# 普段やっていること

メモ.txt

メモ

ID: hoge

メモ.txt

メモ

ID: hoge2015

セーブ！！

# 普段やっていること

メモ.txt

メモ

ID: hoge

メモ.txt

メモ

ID: hoge2015

間違えてた！！

戻したいけど上書きしてんじゃん！！！

バージョンを管理したいとき、  
どうしますか？

...よけいわからん

# 案その1

コピーしてバックアップを  
とっておく

# コピーをとっておく

**index.html**

**編集用index.html**

**index-old.html**

**index20510620.html**

**最新版index.html**

**index.html (前のやつ)**



「バージョン管理」とは

セーブするたびに、  
バージョンを振ります

# 「バージョン管理」とは...

ver. 1

メモ.txt

変更しました！  
セーブしました！

ver. 2

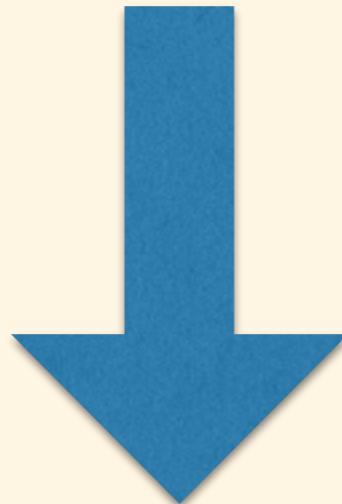
メモ.txt

# 「バージョン管理」とは...

ver. 1

やばい間違えてた！  
しかも上書きしちゃつ  
たよー！！

メモ.txt



ver. 2

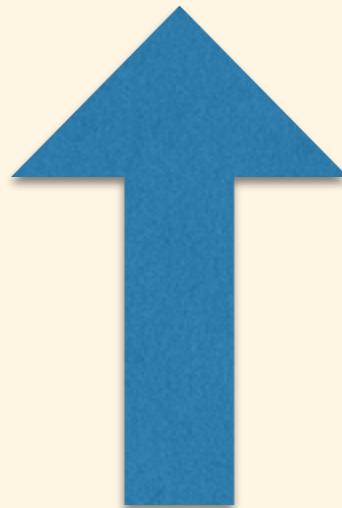
メモ.txt

# そんなときにgitなら

ver. 1

メモ.txt

これで安心！便利！



一個前にセーブした  
ところ(ver.1)に戻る！

ver. 2

メモ.txt

# gitのいいところその1

すべての変更履歴を管理できる

# gitのことば

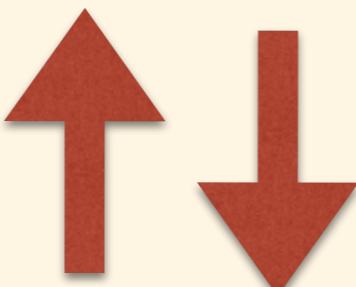
- ・ セーブ = コミット (commit)
- ・ リポジトリ = フォルダ

# gitのしくみ

リモートリポジトリ

index.html  
style.css

push



fetch

自分のPCの中

ローカルリポジトリ

index.html  
style.css

commit



merge

index.html  
style.css

## gitのいいところその2

みんながみんなのPCで  
ネットなくとも作業できる

これでgit使える！

1. ステージング・エリアに追加

```
$ git add FILENAME
```

2. ローカルで変更をセーブ

```
$ git commit -m "message"
```

3. ローカルのセーブをリモートにもセーブ

```
$ git push
```

1. ステージング・エリアに追加

```
$ git add FILENAME
```

2. ローカルで変更をセーブ

```
$ git commit -m "message"
```

# \$ git add/commit

リモートリポジトリ

index.html  
style.css

自分のPCの中

ローカルリポジトリ

**add/commit**

index.html  
style.css



index.html  
style.css

1. ステージング・エリアに追加

```
$ git add FILENAME
```

2. ローカルで変更をセーブ

```
$ git commit -m "message"
```

3. ローカルのセーブをリモートにもセーブ

```
$ git push
```

3. ローカルのセーブをリモートにもセーブ

```
$ git push
```

# \$ git push

リモートリポジトリ

index.html  
style.css

push



自分のPCの中

ローカルリポジトリ

index.html  
style.css

index.html  
style.css

# GitHubとは



**github**  
SOCIAL CODING

**GitHubは、Gitのリポジトリ  
をホスティングするサービス**

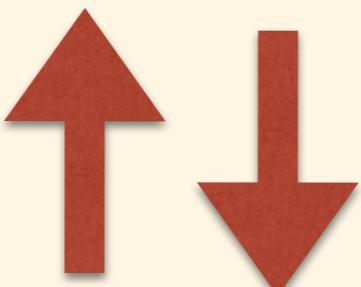
# gitのしくみ

GitHub

リモートリポジトリ

index.html  
style.css

push



fetch

自分のPCの中

ローカルリポジトリ

index.html  
style.css

commit



merge

index.html  
style.css



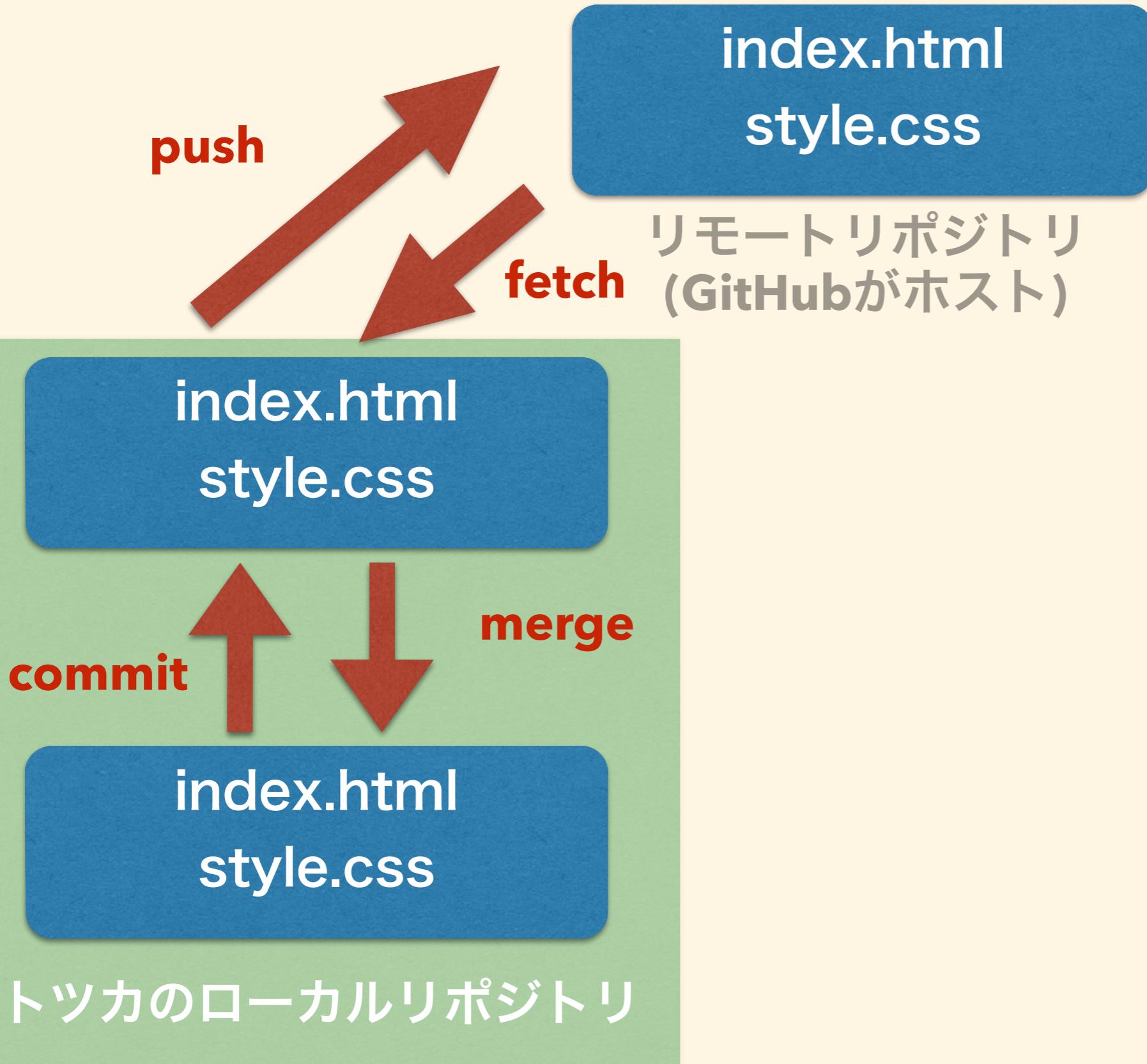
**github**  
SOCIAL CODING

Atlassian  
**Bitbucket**



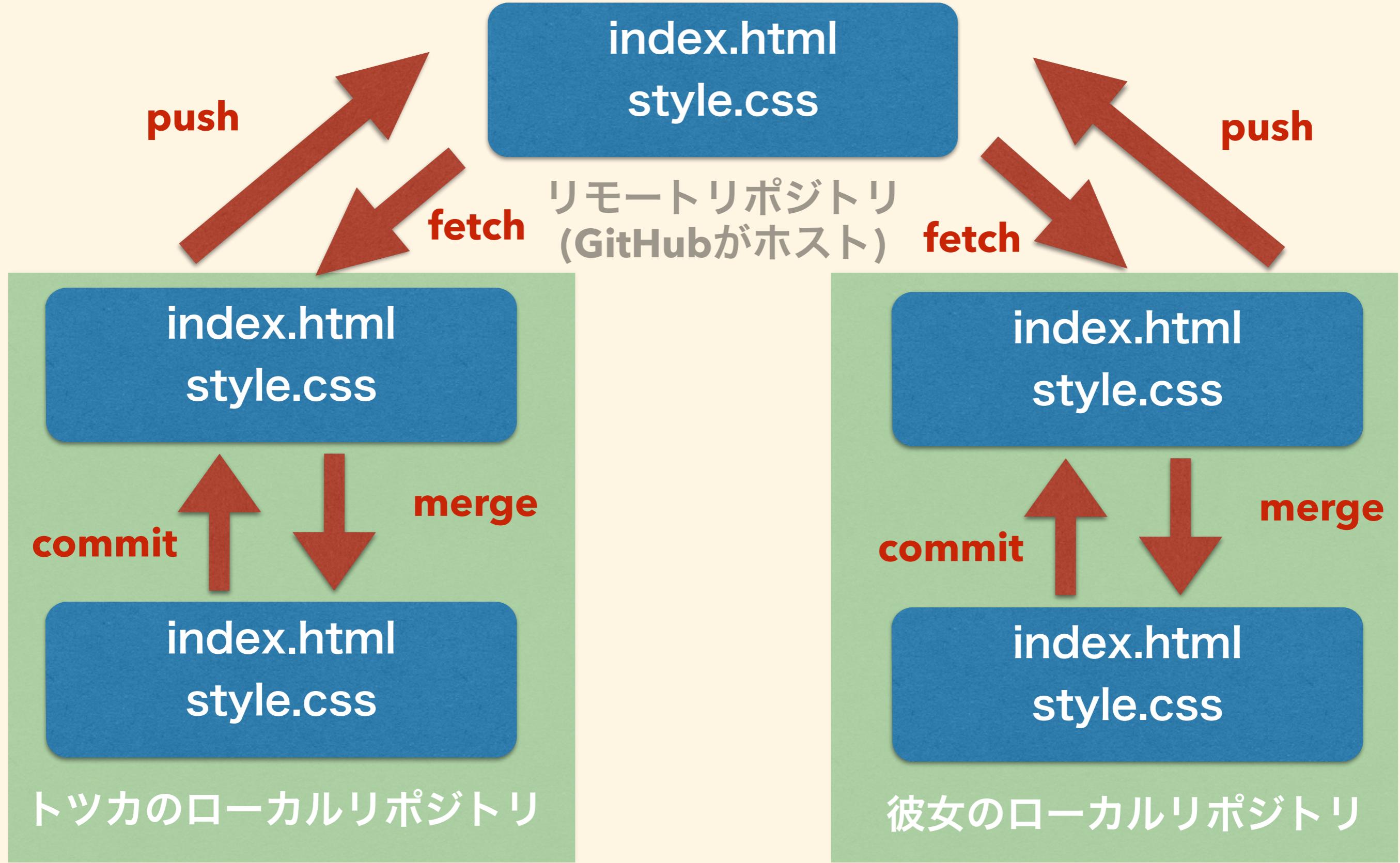
チーム開発どうやんの

# チーム開発のやりかた

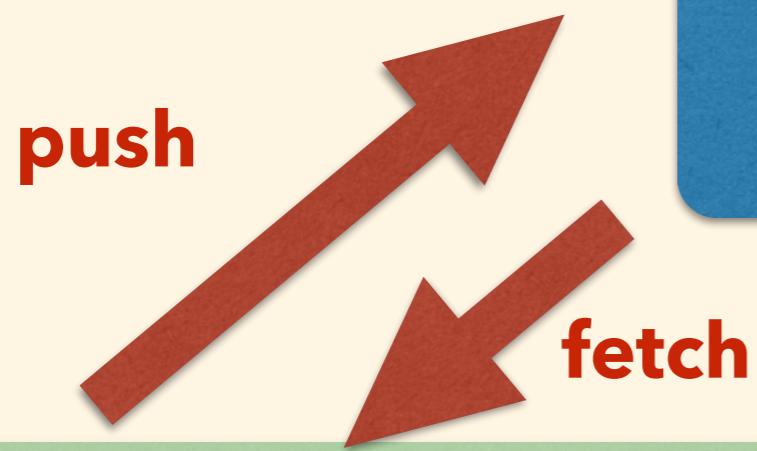


**彼女と共同作業をするとき**

# 彼女と共同作業をするとき



# 彼女と共同作業



トツカのローカルリポジトリ

リモートリポジトリ  
(GitHubがホスト)

これが分散！！！



彼女のローカルリポジトリ

# チーム開発フロー

# チーム開発フロー

- 自分のアカウントにデータを複製 (fork)
- ローカル(自分のPCの中)にデータを持ってくる (clone)
- 編集する
- 変更内容をコミット(セーブ)する (add/commit)
- コミット内容をリモート(GitHub上のソース)に反映する (push)
- Pull Requestをだす (GitHubの画面から！)

# 1. 複製する (fork)

GitHubのページから！！

編集したいデータの複製を自分の  
GitHubのアカウントにつくれます

# 1. 複製する (fork)

オリジナル

GitHub  
(自分のアカウント)

リモートリポジトリ

index.html  
style.css

自分のPCの中

ローカルリポジトリ

# 1. 複製する (fork)

オリジナル

GitHub  
(自分のアカウント)  
リモートリポジトリ

index.html  
style.css

fork

index.html  
style.css

自分のPCの中

ローカルリポジトリ

## 2. 自分のPCに持ってくる(clone)

```
$ git clone https://github...
```

\* URLはGitHubのページからコピペします

リモート(GitHub)から  
ローカル(自分のCP)にデータを持ってきます

## 2. 自分のPCに持ってくる(clone)

GitHub

(自分のアカウント)

リモートリポジトリ

index.html  
style.css

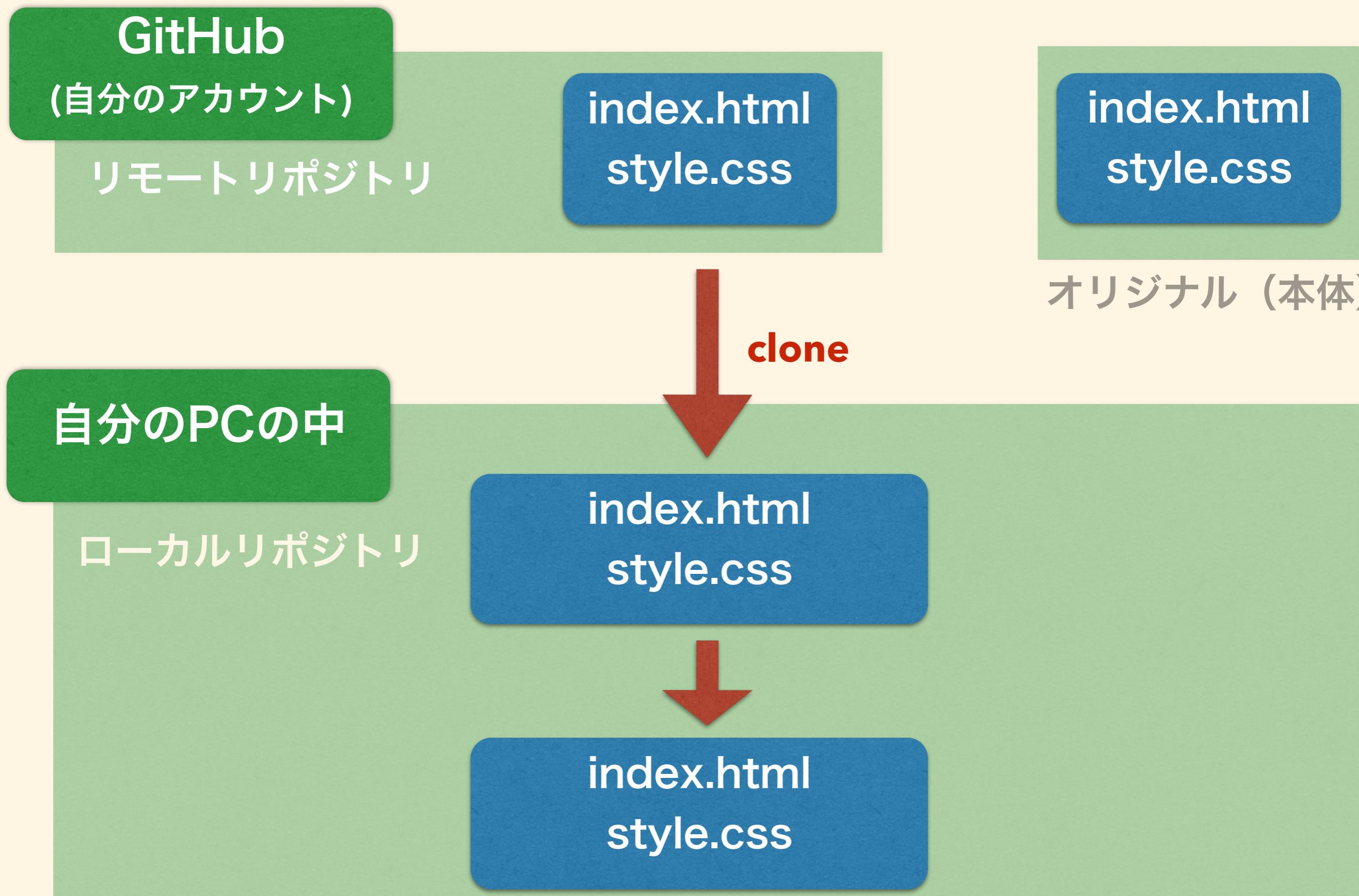
index.html  
style.css

自分のPCの中

ローカルリポジトリ

オリジナル（本体）

## 2. 自分のPCに持ってくる(clone)



### 3. 編集する



Sublime Text



### 3. 編集する

GitHub  
(自分のアカウント)

リモートリポジトリ

index.html  
style.css

index.html  
style.css

オリジナル（本体）

自分のPCの中

ローカルリポジトリ

index.html  
style.css

index.html  
style.css

### 3. 編集する

GitHub  
(自分のアカウント)

リモートリポジトリ

index.html  
style.css

index.html  
style.css

オリジナル (本体)

自分のPCの中

ローカルリポジトリ

index.html  
style.css

編集したった!!!!

index2.html  
style2.css

## 4. コミット(セーブ)する

```
$ git add FILENAME
```

commit(セーブ)するファイルを選ぶ

```
$ git commit -m "message"
```

ローカル（自分のPCの中）で  
変更内容をセーブ！

## 4. コミット(セーブ)する

GitHub

(自分のアカウント)

リモートリポジトリ

index.html  
style.css

index.html  
style.css

オリジナル (本体)

自分のPCの中

ローカルリポジトリ

index.html  
style.css

index2.html  
style2.css

## 4. コミット(セーブ)する

GitHub  
(自分のアカウント)

リモートリポジトリ

index.html  
style.css

index.html  
style.css

オリジナル (本体)

自分のPCの中

ローカルリポジトリ

index2.html  
style2.css



**add / commit**

index2.html  
style2.css

# 5. リモートに反映する(push)

```
$ git push
```

ローカル（自分のPC）の中の変更を  
リモート（GitHub上）のソースに反映します

## 5. リモートに反映する (push)

GitHub

(自分のアカウント)

リモートリポジトリ

index.html  
style.css

index.html  
style.css

オリジナル (本体)

自分のPCの中

ローカルリポジトリ

index2.html  
style2.css

index2.html  
style2.css

## 5. リモートに反映する (push)

GitHub

(自分のアカウント)

リモートリポジトリ

index2.html  
style2.css

index.html  
style.css

push



自分のPCの中

ローカルリポジトリ

index2.html  
style2.css

index2.html  
style2.css

オリジナル (本体)

# 6. 変更を取り込んでもらう (pull request)

GitHubのページから！！

オリジナルのレポジトリに  
「僕の変更を取り込んでくれ！」  
とリクエストします

## 5. 変更を取り込んでもらう (pull request)

GitHub

(自分のアカウント)

リモートリポジトリ

index2.html  
style2.css

index.html  
style.css

オリジナル (本体)

自分のPCの中

ローカルリポジトリ

index2.html  
style2.css

index2.html  
style2.css

## 5. リモートに反映する (push)

GitHub  
(自分のアカウント)  
リモートリポジトリ

index2.html  
style2.css

PR

index.html  
style.css

自分のPCの中  
ローカルリポジトリ

index2.html  
style2.css

変更したから取  
り込んでよー！！

オリジナル (本体)

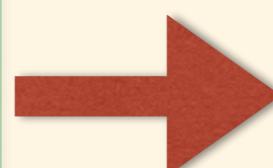
index2.html  
style2.css

## 5. リモートに反映する (push)

GitHub  
(自分のアカウント)  
リモートリポジトリ

index2.html  
style2.css

PR



index2.html  
style2.css

自分のPCの中  
ローカルリポジトリ

index2.html  
style2.css

変更したから取  
り込んでよー！！

OK!!!

index2.html  
style2.css

# 0. みんなの変更を取り込む

```
$ git remote add upstream https://github.com/okadai-techlab/okayama-u-techlab.com.git
```

てくらぼのオリジナルの方もリモート先として登録する

\* 登録は最初の一回だけでいいよ！‘upstrem’って名前じゃなくてもいいよ！

```
$ git pull upstream master
```

オリジナルの方のリモート（GitHub上）が変更され  
ていたら最新版を自分のソースに取り込む

## 5. 変更を取り込んでもらう (pull request)

GitHub

(自分のアカウント)

リモートリポジトリ

index2.html  
style2.css

誰かが変更した  
(最新版)

index3.html  
style3.css

複製  
origin

オリジナル (本体)  
upstream

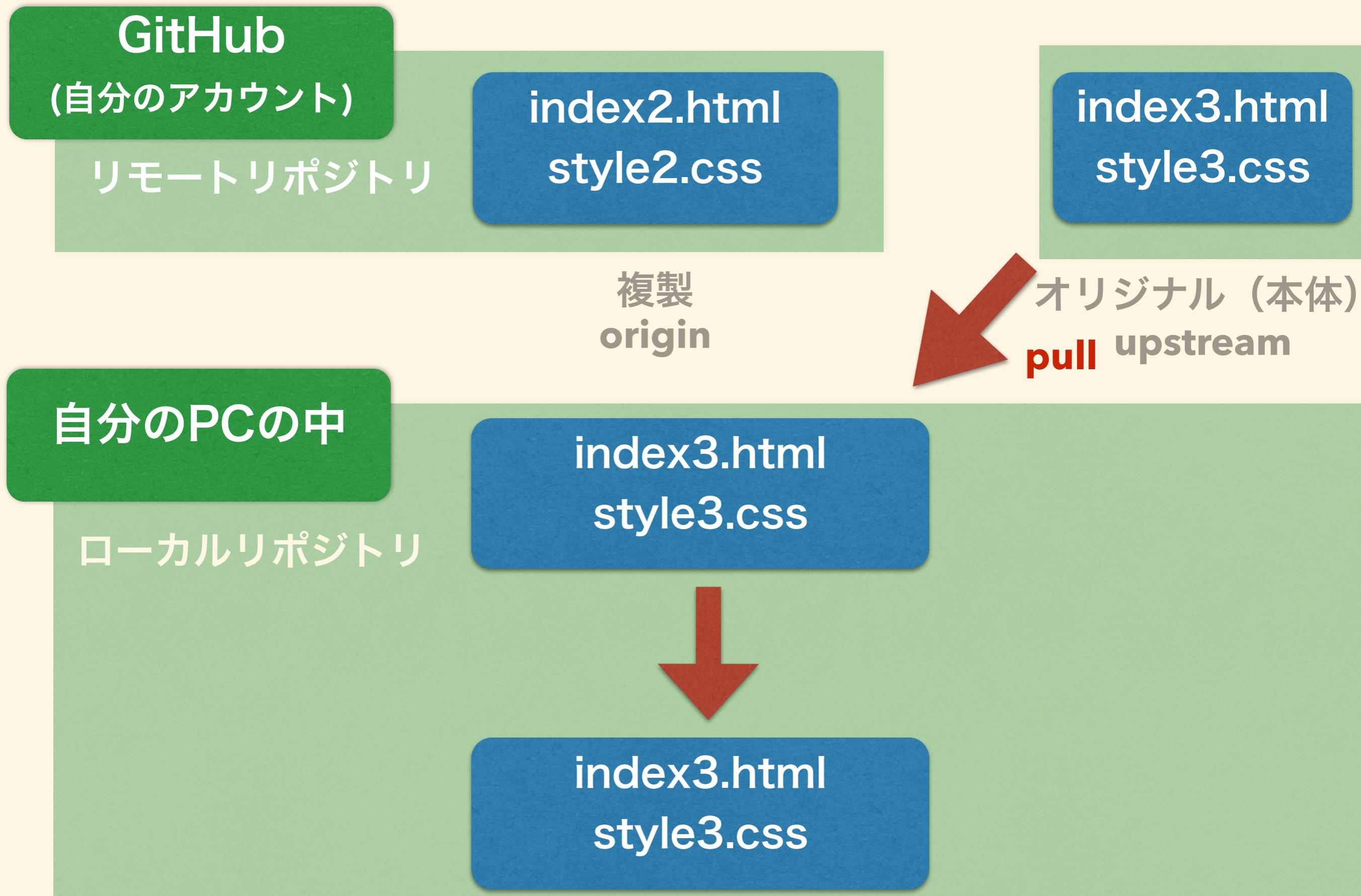
自分のPCの中

ローカルリポジトリ

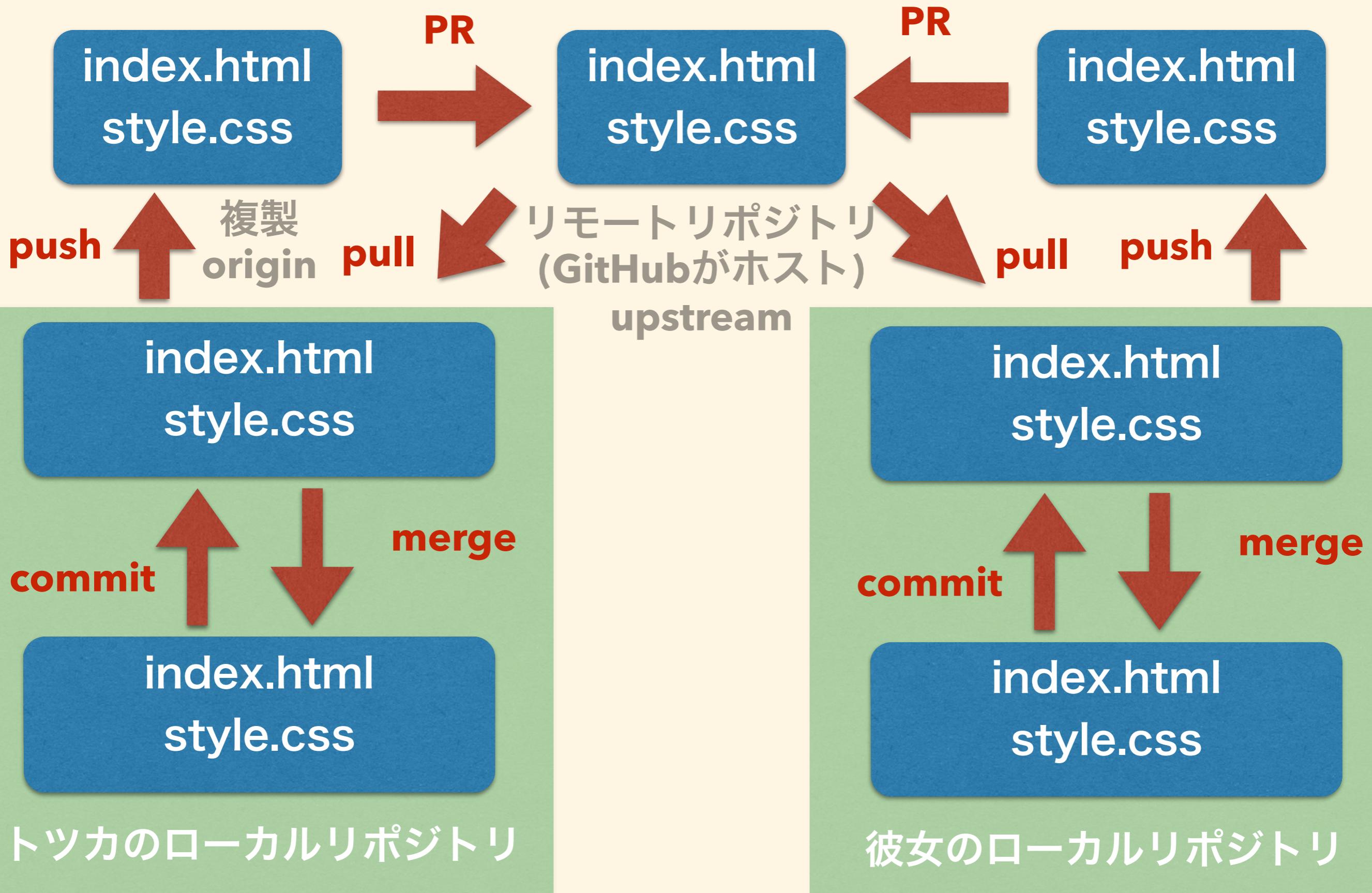
index2.html  
style2.css

index2.html  
style2.css

## 5. 変更を取り込んでもらう (pull request)



# 彼女と共同作業をするとき



# チーム開発フロー

- 自分のアカウントにデータを複製 (fork) ... 1回目だけ
- ローカル(自分のPCの中)にデータを持ってくる (clone) ... 1回目だけ
- 最新版を取り込む (pull)
- 編集する
- 変更内容をコミット(セーブ)する (add/commit)
- コミット内容をリモート(GitHub上のソースorigin)に反映する (push)
- みんなの変更を取り込む (Pull Requestをだす) -> pullに戻る

# Demo