



WAV

Git, GitHub

本日のスケジュール

- ・ Git, GitHubとは
- ・ 基本の使い方
- ・ コマンド集

WAV

Git, GitHubとは

WAV

Git, GitHubとは

ソースコードのバージョン管理ツール

コードの変更履歴(誰が、いつ、何を)を管理してくれます



共同開発ではほぼ必須！！

Git, GitHubとは

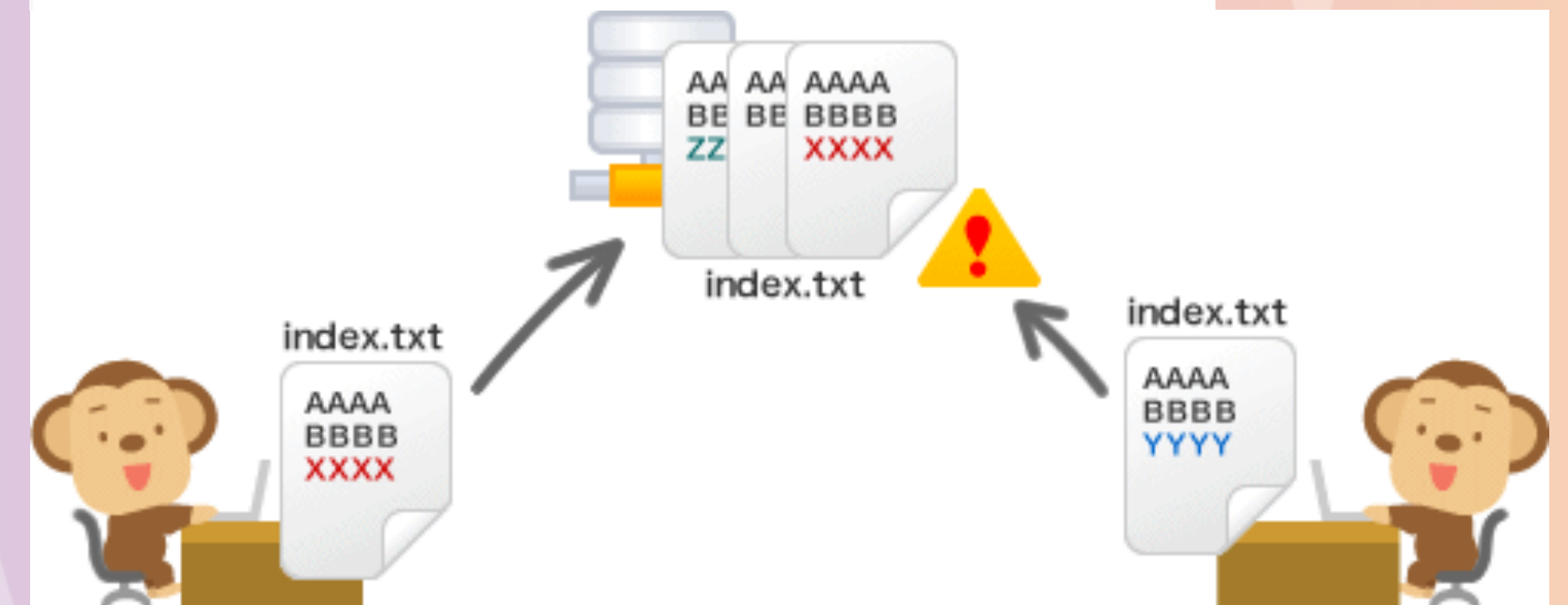
例えば、

- ・ファイルの更新履歴が。。。
- ・共同開発したら。。。



Gitで管理しよう！

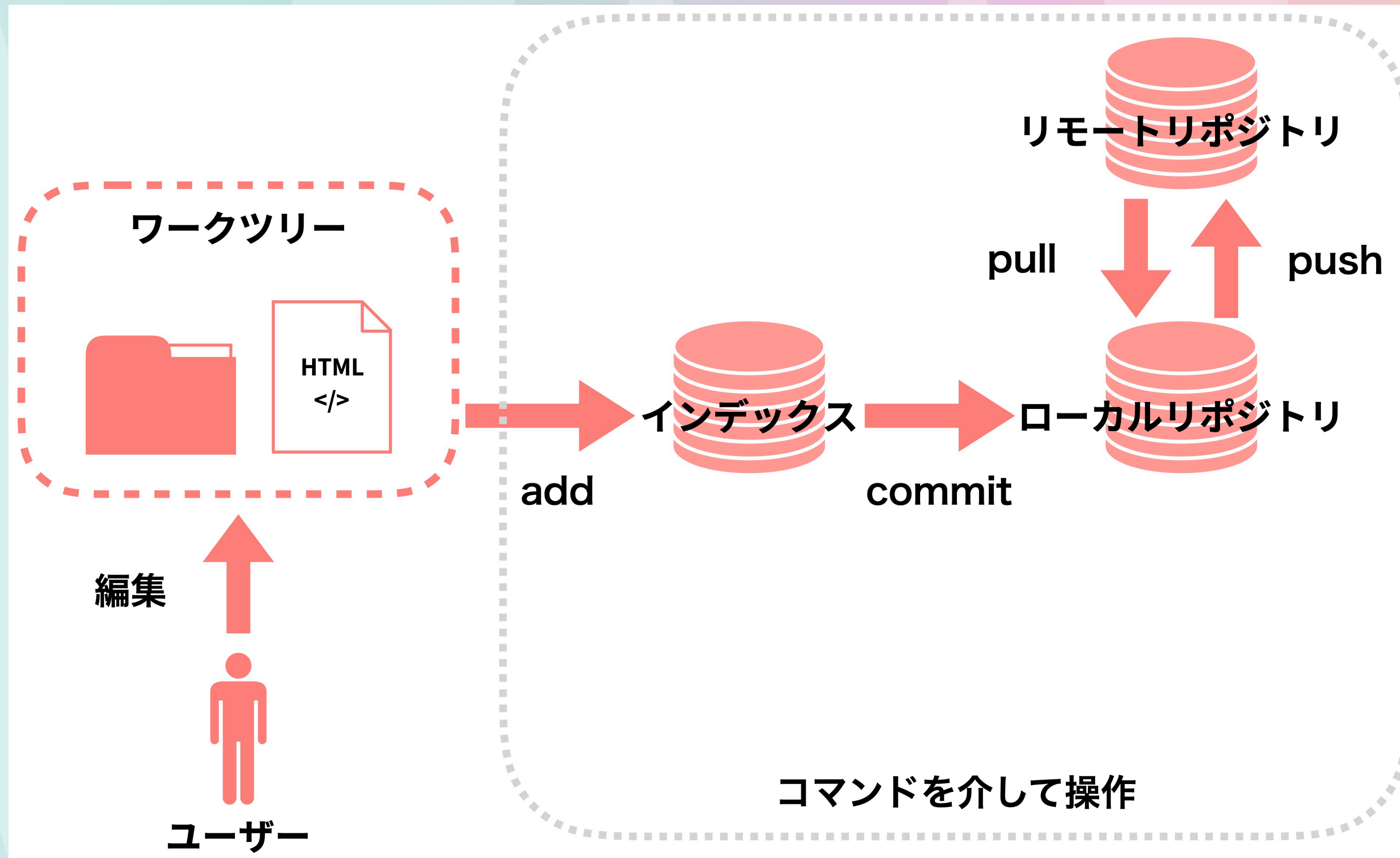
<input type="checkbox"/> 名前	状態
A社見積り-加藤作成-仮-修正済み	✓
A社見積り-加藤作成-仮	✓
A社見積り-斎藤チェック済み	✓
A社見積り-斎藤チェック済み-修正済み	✓
A社見積り-提出前	✓
A社見積り-提出前-最終	✓



基本の使い方

WAV

Gitの仕組み



Gitの用語

『リポジトリ』

過去のソースコードのデータベース

「ローカルリポジトリ」・・・各々のPCにあるデータベース

「リモートリポジトリ」・・・ローカルリポジトリをサーバー上にバックアップしたもの。GitHubやBitbucketなど

『インデックス』

リポジトリにコミットする準備をするための場所

『ワークツリー』

Finderから見えるファイル

Gitの用語

『add』

コミットの対象となるファイルをインデックスに登録(ステージング)する操作

『commit』

その時のワークツリーの状態をgitに登録する操作

エラー時などは、データベースを読み込むことで前の状態に戻ることができる

『push』

ローカルリポジトリの内容をリモートリポジトリに反映させる操作

『pull』

リモートリポジトリの内容をローカルリポジトリに反映させる操作

補足

インデックスの存在意義

コミットを意味のあるまとまりにできる
→1つのコミットは1つの意味を束ねたものが良い

例えば、X機能開発中に、別でY機能も実装した。
→コミットはX機能に関連するファイルだけ

補足

1コミット1つの意味

コミットはできるだけ細かくすると良い

例えば、

X機能実装→コミット→Y機能実装→コミット→...

一連の流れ

- GitHubでリポジトリを作成
- リモートリポジトリをクローン(git clone)
- ローカルリポジトリを最新にする(git pull) – 2回目以降
- 開発後、ファイルをステージング(git add)
- コミットする(git commit)
- リモートリポジトリにプッシュ(git push)

】初回のみ

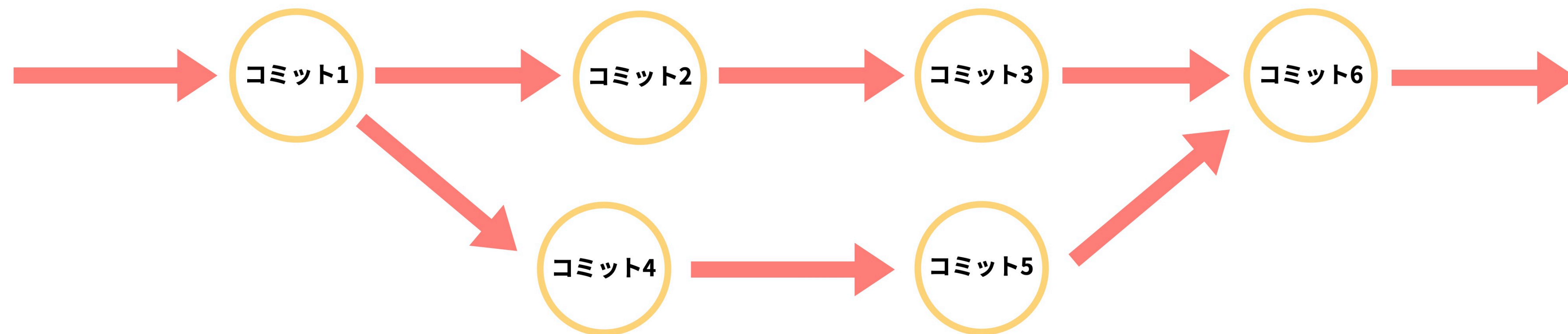
WAV

ブランチ

『ブランチ』とは…

訳：木の枝

1つのプロジェクトから分岐させることで、本体に影響を与えずに作業できる



現在のバージョンを直接書き換えることなく、新機能の開発を行ったりできる！

共同作業をしている際に、同時に同じ箇所を書き換えてコードの整合性が取れなくなることを防ぐ！

一連の流れ(ブランチ)

- ブランチを作成(`git checkout -b`)
- 開発後、ファイルをステージング(`git add`)
- コミットする(`git commit`)
- リモートリポジトリにプッシュ(`git push`)
- ブランチをマージ(`git marge`)

WAV

$+\alpha$

もっと実践的にしっかり学びたい人は

<https://backlog.com/ja/git-tutorial/>

WAV

コマンド集

WAV

基本コマンド

git init <リポジトリの名前> (リポジトリを作成)

git add <ファイル名.拡張子> (ファイルをステージング)

git add . (変更等あった全ファイルをステージング)

git commit -m “コミットの要約” (コミットする)

git push origin master (リモートリポジトリにプッシュ)

git pull origin master (リモートリポジトリをプル)

基本コマンド

git checkout -b <ブランチの名前> (ブランチを作成)

git merge <マージ対象のブランチ> (ブランチをマージ)

git diff (変更部分を表示)

git diff --name-only (変更ファイルを表示)

git log (コミット履歴を表示)

git checkout <ブランチ名> (ブランチを切り替え)