1 Gitとはプログラム開発の作業履歴を記録するツール

システム開発の現場ではかかせないもの

複数メンバーで共同開発すると、誰が、いつ、どこを、何故修正したのか

把握しづらくなる→作業効率の低下、バグも発生しやすい

↓それを防ぐのが

バージョン管理システム

バージョン管理システムを利用すれば、ファイルなどいつどこで何を修正したのかなどを

記録しておける

色々なバージョン管理のツールがある

・CVS

・**Git**

・Mercurial

・Subversion

中でもよく使われているのがGit

Gitには様々なwebサービスがある（更に使いやすくしたもの）

Gitのwebサービス

・Github

・Gitlab

・BitBucket

Gitの特徴

・分散型のバージョン管理システム

・シェルコマンドで操作

・多くの企業やコミュニティが採用

主な採用している企業

Google

Microsoft

Facebook

主な採用しているコミュニティ

Linux

Ruby

Ruby on Rails

事実上のバージョン管理システムの標準ツールと言ってもいいぐらい採用されている

基本のシェルコマンド

カレントディレクトリを表示する

pwd

ファイル一覧を表示する

ls

隠しファイルも含めてファイル一覧を表示する

ls -a

gitの基本コマンド

オプションを付けないで実行すると簡易ヘルプが表示される

git

バージョンを表示する

git –version

ユーザー情報を登録する（名前とメールアドレスを登録する

作業履歴の記録時にこの名前が使われる

git config –global user.name “名前”

git config –global user.email”メアド”

登録情報を確認する

git config –list

リポジトリを作成する

myworkリポジトリを作ってそこにリポジトリを作成する、作業履歴は

このディレクトリの「.git」という隠しディレクトリに記録される

mkdir リポジトリ名

cd リポジトリ名

git init リポジトリを作成する（登録）

リポジトリやステージングエリアの状態を確認する

git status

ステージングエリアに仮登録する

git add readme.txt

ステージングエリアの内容をコミットする

git commit

コミット履歴を確認する

git log

コミットメッセージだけ表示する

git log –oneline

指定した件数だけコミットログを表示する

git log -2

ステータスを簡易的に表示

git status -s

ステージングとワーキングディレクトリを比較する

※ステージングに仮登録されていない新規ファイルは表示されない

git diff

最新コミットとステージングエリアを比較する

git diff –cached

最新コミットとワーキングディレクトリを比較する

git diff HEAD

シェルで利用できるテキストエディタ

gitでコミットしようとすると、コミットメッセージの入力のために

自動的に起動する

vim

vimの操作コマンド

- a 現在のカーソル位置の右から入力する

- A 現在の行の末尾に入力する

- 編集モードからコマンドモードに切り替える

- u 元に戻す

- :wq 保存して終了

- :q! 保存しないで終了

ステージングエリアにまとめて仮登録する

git add .

仮登録とコミットを同時に実行する

git commit -a

コミットメッセージを同時に指定する

git commit -m “commit message”