

前提:

README.md に示した使用ソフトのインストールおよび初期設定は完了しているものとする。

問題 1

Git はファイルやプログラムの(ア)を行うためのシステムである。Git などの(ア)システムを導入することによって、(イ|₂つ挙げよ)ができるようになる。

(ア)システムは(ウ)システムと(エ)システムの2パターンに分けられるがGitは(エ)である。

(ウ)システムでは、複数人でファイルなどを共同編集する場合、中央サーバに直接変更後のファイルを送るが、(エ)システムでは(オ)。こうすることで、共同編集時に生じる混乱を防いでいる。また(カ)というメリットもある。

問題 2

Git の特徴としては

- ・ネットワークに接続していなくても作業ができる(一つ目)
- ・履歴を保存する前に Staging を行う(二つ目)
- ・他のユーザと柔軟なやり取りができる(三つ目)

が挙げられる。

一つ目…問題 1 で述べた通り、分散型はバージョン管理システムの利点である。

二つ目…Git では、(ア)を Commit という。Commit を行う際の(イ)を Staging という。

Staging を行うことによって、(ウ)を一つの Commit にまとめたり、逆に単一のファイルに施した変更を(エ)たりなど、Commit を柔軟に加工し、(オ)ことが可能となる。

(オ)が可能になることによって、(カ|バグ対策に役立つ点を 2 つ挙げよ)などのメリットが生まれる。

三つ目…Git を用いたツール「GitHub」はみんなで使う SNS であるため、他人のソースを自由に見ることができる。また、そのソースに問題が見つかったなどの場合、修正を加えてあげて、それを反映するようにリクエストを送ることができる。これを(キ)という。

問題 3

参考: <https://wa3.i-3-i.info/word15664.html>

リポジトリ (repository) とは—「分かりそう」で「分からない」でも「分かった」気になれる IT 用語辞典 (2018 年 11 月 24 日閲覧)

リポジトリとは何か。簡単にいえ。

問題 4

Git はファイルを共同編集する際にも変更点を適切に管理でき、大変便利であるが、それでも非常の多くの人が参加するプロジェクトで、かつ(ア)場合、リポジトリの管理が複雑になる。そこで、Git において(ア)のを簡単にできるようにするための Hub(中継地点)として生れたのが GitHub である。

問題 5

GitHub において、Push と Pull の意味をそれぞれ説明せよ。

問題 6

Fork と Pull Request は Git にはない GitHub の特徴的な機能の例である。Fork と Pull Request の意味に触れながら、さらに Fork、Pull Request、Merge によって「プロジェクトに参加できる」ことを説明する文章を書け。

問題 7

Clone とは何か

問題 8

コマンドプロンプトを立ち上げた。

GitHub ユーザ 17ec084 のリポジトリ test を一つ下のディレクトリ cloneHere に Clone せよ。

(git コマンドを使うためのパスは通っていて、windows または Linux の環境とする)

問題 9

ローカルディレクトリ a はローカルリポジトリである。a/b.java と a/c.c を書き換えた。この変更の内、b.java に施した変更のみを共有リポジトリ A に反映させたい。コマンドプロンプトから行いたい。コマンドプロンプトを起動するところから方法を説明せよ。

問題 10

現在のディレクトリ内の変更すべてを共有リポジトリに反映させたいとき、どのようなコマンドによって Staging できるかいえ

問題 11

問題 9 に出てきた master について説明せよ。

問題 12

ブランチ(Branch)とは何か。

答え 1

(ア):バージョン管理

(イ):変更の履歴の確認と、過去の特定の状態へ戻すこと

(ウ):中央集権型バージョン管理

(エ):分散型バージョン管理

(オ):全員がそれぞれローカル環境に「中央サーバのコピー」を作り、その差分を本物の中央サーバに送る

(カ):ネットワークに接続していなくても作業ができる

答え 2

(ア):変更履歴の単位

(イ):準備

(ウ):複数のファイルに施した変更

(エ):複数の Commit に分け

(オ):各 Commit に意味を持たせる

(カ):

- ・バグが発見されたときの切り戻しが容易になる
- ・どの時点でバグが入ったのかが確かめやすくなる

(キ):Pull Request

答え 3

データの保管場所

答え 4

(ア):複数人でリポジトリの管理を行う

答え 5

Push・・・ローカルで加えた変更を共有リポジトリに反映

Pull・・・最新の共有リポジトリをローカルに反映

答え 6

興味のある(他人の)プロジェクトを Fork することで、そのプロジェクトのコピーを自分のリポジトリに作る。そして加筆・修正したものを、元のプロジェクトの開発者に Pull Request

として送る。開発者は送られた Pull Request に問題がないと判断すればその変更を反映 (Merge)する。この方法で、開発者でなくても興味あるプロジェクトに参加することができる。

補足:ここでいう「開発者」というのは、プロジェクトを編集する権利(コミット権)が与えられている者のことをいう。

答え 7

共有リポジトリを各開発者のローカル環境にコピーすること。

答え 8

cd cloneHere

git clone git@github.com:17ec084/test.git

答え 9

1. ディレクトリ a でコマンドプロンプトを起動する。
2. git remote -v を実行し、共有リポジトリの「名前」を確認する。
今回は origin であったと仮定。
2. git add b.java
3. git commit -m (コミットメッセージ)
4. git push origin master

git push コマンドについて紹介したので、git pull コマンドの紹介は省く。引数は同じ。

答え 10

git add .

答え 11

master は、レポジトリにデフォルトで与えられるブランチの名前である。

補足:とりあえず、「ブランチ」をいじっていないのであれば今のブランチは絶対 master ブランチなので、言われた通りに master と打てばよい。

答え 12

開発中のリポジトリの本流から分岐し、本流に影響を与えない形で独自の変更を加えるための「枝分かれ」における「枝」である。