はじめに

この資料は、「著者(俺)がわかる」をベースに書いています。

なのでこの資料を読んで「わかりにくい」と思うかもしれません。

人によって理解の仕方は違うので仕方がないことです。

そういうときは、自分に合う資料を見つけてください。

Git/GitHubに関する資料はインターネット上を探せばいくらでもあります。

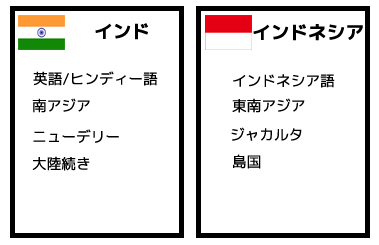
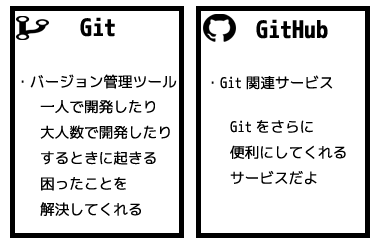
自分で探すということも大事です。つまり、何が言いたいかというとこの資料を読んでも必ず理解できるわけではないので理解できるよう自分で努力してください、ということです。

それでは、こういう投げやりな資料でも良ければ、読んでいってくださいな。

Git/GitHubとは

まずはじめに、GitとGitHubは**別物**です。どれくらい違うかというと、JavaとJavaScript、

インドとインドネシアくらい違います。



Gitとは

Gitとはプログラムのソースコードやドキュメントの変更,その履歴を管理,追跡するための分散型バージョン管理ツールです。開発者はLinuxカーネルの開発者でおなじみのリーナストーバルズらである。

バージョン管理ツールとは

名前の通りです。例えば、ソースコードを書いていて「以前の状態のほうが良かった」とか「間違って変更してしまった」、「新しい機能を試したいからコピーしてそっちで試そう」など思うことがありますよね?そういう開発中の困ったことを解決してくれるのがバージョン管理ツールです。コマンド、またはGUIによる直感的な操作で、以前の状態に戻したり状態を分岐させて新しい機能を試したりということができます。

バージョン管理ツールはGitだけでなくApacheソフトウェア財団のSVN(Apache・Subversion)やThe CVS TeamのCVS(Concurrent Versions System)などがあります。

バージョン管理ツールの種類

バージョン管理ツールには大きく分けて2種類があります。

1つはCVSやSVNなどの集中型バージョン管理システム、もう1つはGitの分散型バージョン管理システムです。

集中型バージョン管理システムと分散型バージョン管理システム、これらの違いを説明するにあたり覚えておいてほしい用語があります。

リポジトリ(repository) - ファイルの変更履歴や、ファイルそのものなどを保存する場所

ローカルリポジトリ(local repository) : 自分のコンピュータ内にあるリポジトリ

リモートリポジトリ(remote repository) : 外部のコンピュータ(サーバ)内にあるリポジトリ

コミット(commit) : リポジトリに履歴を登録すること

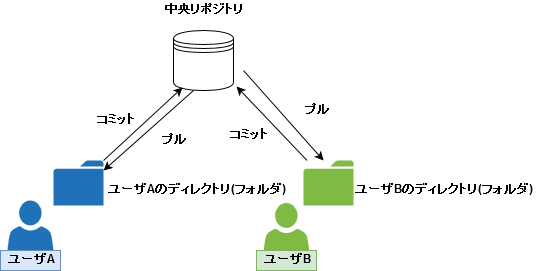
プル(pull) : リポジトリから履歴を取得すること

プッシュ(push) : ローカルリポジトリをリモートリポジトリに反映すること

集中型バージョン管理システム

集中型バージョン管理システムはいわゆる「クライアント・サーバモデル」を使ったバージョン管理のことです。中央リポジトリ(リモートリポジトリ)から各クライアントがデータを持ってきて編集し、編集が終わったら中央リポジトリを更新するという管理方法です。

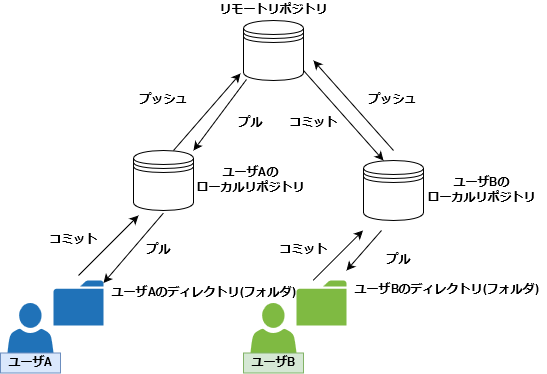
よってリポジトリは一つとなります。



分散型バージョン管理システム

分散型バージョン管理システムは集中型バージョン管理システムと違いリポジトリを複数持つことができます.例えばユーザが自分のリポジトリ(ローカルリポジトリ)を持っていて

普段はローカルリポジトリを用いて作業を行い、ある程度作業が完結し作業を他のユーザと共有したい時はリモートリポジトリに反映するといったことが可能です。



GitHubとは

その名の通りGitのHub(ハブ : 中心部)です.リモートリポジトリを運用しているサービスです.所謂、クラウドと呼ばれるものでGoogle DriveなどのGit版。

類似サービスに

GitLab, Bitbucketなどがある。

Gitの導入

公式サイトから<https://git-scm.com/downloads> OSに応じてダウンロード・インストールを行ってください。

Gitを使おう

1. リポジトリを作ろう

以下のコマンドを入力するとカレントディレクトリ(今自分が開いているディレクトリ)がリポジトリとして登録されます.おそらくinitはinitialize(初期化)の略称です.

$ git init

1. ファイルをステージングしよう

Gitにコミット(変更履歴を登録)するためにはコミットするファイルを登録してあげなければいけません。以下のコマンドでファイルを登録しましょう。

$ git add 登録したいファイル名

また、以下のコマンドでステージングを取り下げることができます

$ git reset ファイル名

1. ファイルをコミットしよう

Gitにコミット(変更履歴を登録)します。以下のコマンドでファイルをコミットしましょう。

コミットメッセージとはどんな変更を加えたかなどをわかりやすくするためのラベルです。

書かなくともコミットはできますが、できるだけ書くようにしましょう。

$ git commit –m “コミットメッセージ”

1. コミットを確認してみよう

以下のコマンドで今までのコミット履歴が確認できます。

$ git log

1. ブランチを切ってみよう

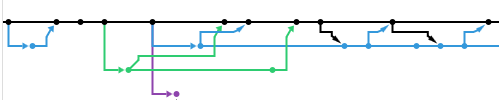
**ブランチ(Branch)とは**

日本語で枝という意味です。

何かプログラムを作成していて別の機能を追加したいけどこのファイルは残しておきたい

とか機能追加前のバージョンと機能追加後のバージョン両方とも残しておいて並行して作業を行いたい時に作成します。

ブランチの中でも特に主となるブランチのことをマスターブランチ(master branch)と呼びます。下の画像でいうと黒い線がマスターブランチです。



**ブランチを作成する**

以下のコマンドでブランチを作成します。

$ git branch 作成したいブランチ名

**ブランチを確認する**

以下のコマンドで現在いるブランチと存在するブランチを確認します。

$ git branch

**ブランチを移動する**

以下のコマンドで現在いるブランチからブランチを移動します。

$ git checkout 移動先ブランチ名

**ブランチをマージする**

ブランチを統合することをマージ(merge)といいます。

マージ先のブランチに移動を行ってから以下のコマンドでブランチをマージできます。

$ git merge マージするブランチ名

**要らないブランチを削除する**

以下のコマンドでブランチを削除します。

$ git branch –d 削除したいブランチ名

1. コミットを戻してみよう

git logコマンドで戻したいコミットのidを取得してから以下のコマンドでコミットを戻せます。

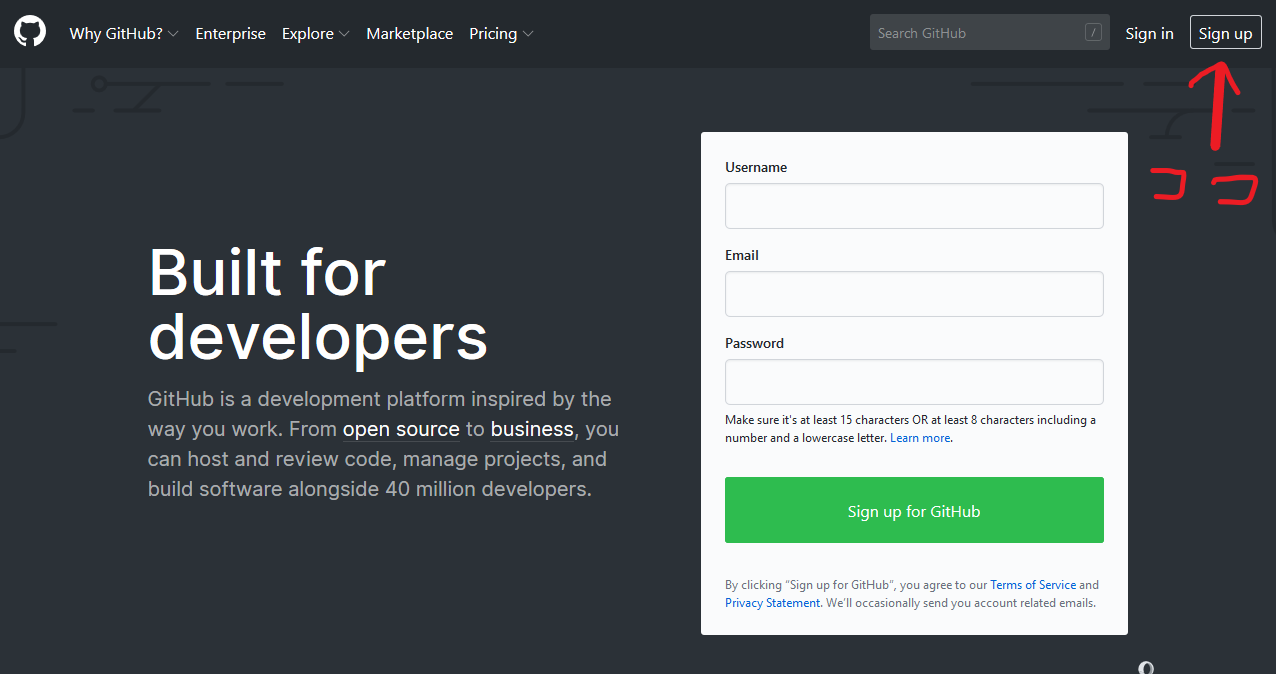
$ git reset –-hard コミットid

GitHubを使おう

<https://github.com/> にアクセスしてください。

1. 登録してみよう

サイトの案内に従って登録してください。



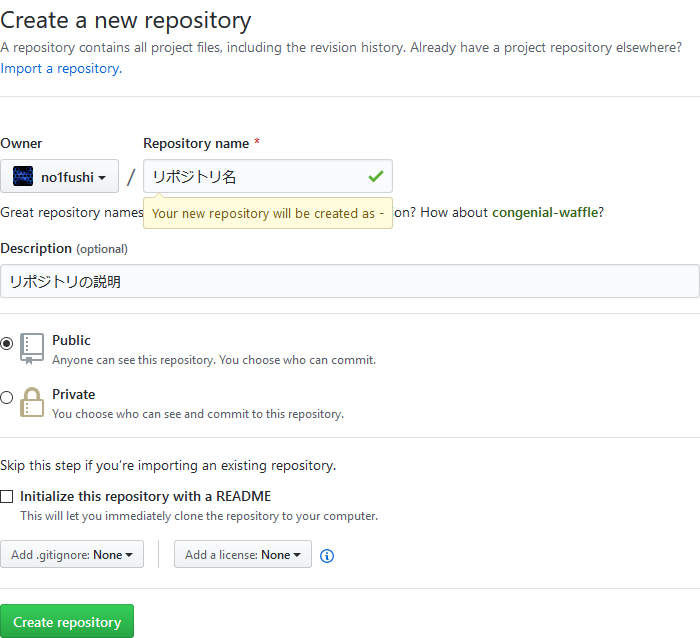
1. ログインしてみよう

<https://github.com/login> からログインできます

1. リポジトリを作成しよう

<https://github.com/new> からリポジトリの作成を行えます。

必要事項を埋めてCreate repositoryボタンを押しましょう。



1. リモートリポジトリにプッシュしよう

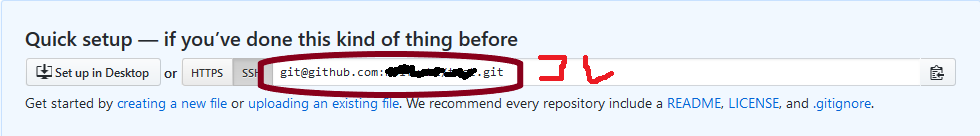
ローカルリポジトリをリモートリポジトリに反映することをプッシュ(Push)と呼びます。

**ローカルリポジトリとリモートリポジトリの関連付けをしよう**

この作業は最初の一度だけ行えばいい作業です。以下のコマンドを実行してください。

$ git remote add origin リモートリポジトリパス

リモートリポジトリパスはリモートリポジトリを作成した際に表示されます。



**リモートリポジトリにプッシュしよう**

以下のコマンドを実行してください。

リモートリポジトリのブランチは標準ではoriginローカルリポジトリのブランチはmasterです。

$ git push –u リモートリポジトリブランチ ローカルリポジトリのブランチ

1. リモートリポジトリの状態をローカルリポジトリに反映しよう

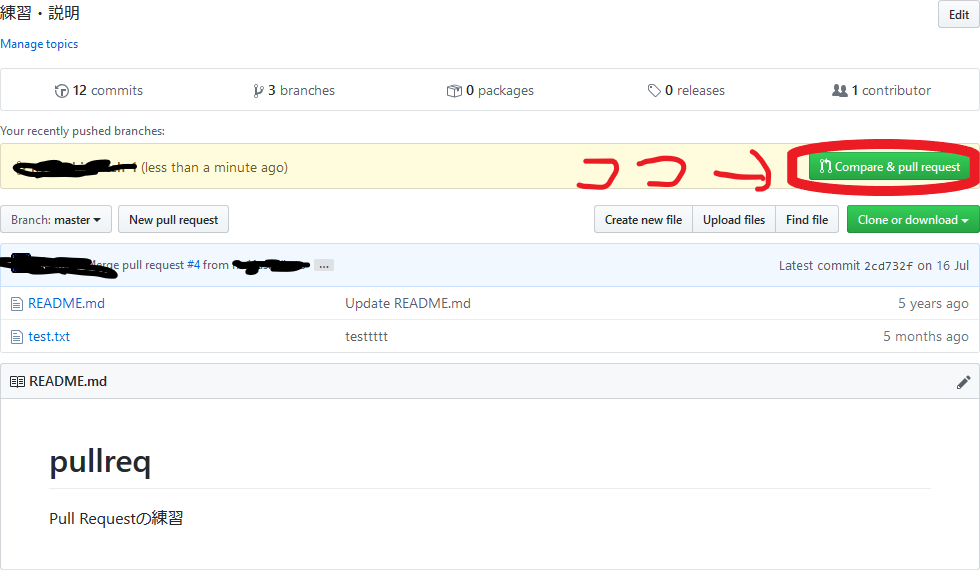
以下のコマンドを実行してください

$ git pull。リモートリポジトリブランチ ローカルリポジトリのブランチ

1. プルリクエストを送ってみよう

Pushするときにローカルリポジトリのブランチをmaster以外のブランチでoriginにpushするとリモートリポジトリでもそのブランチが作成され、更新内容はoriginブランチではなくその作成されたブランチに反映されます。

なので、そのブランチの更新内容をoriginブランチにマージしてくださいとお伺いを立てる作業をプルリクエスト(pull request)と呼びます。



1. 既にあるリモートリポジトリをローカルに持ってこよう

持ってくるリポジトリ用のディレクトリを作成し以下のコマンドを実行してください。

$ git clone リモートリポジトリパス