**git**

**初期設定**

|  |  |
| --- | --- |
| **ユーザー名の設定** | git config --global user.name "First-name Family-name" |
| **メールの設定** | git config --global user.email "username@example.com" |
| **エディタをvimにする** | git config --global core.editor 'vim -c "set fenc=utf-8"' |
| **改行の設定** | git config --global core.autocrlf true |

**core.autocrlf**

|  |  |  |
| --- | --- | --- |
| 設定 | チェックアウト時 | コミット時 |
| TRUE | CRLF | LF |
| input | 変換しない | LF |
| FALSE | 変換しない | 変換しない |

**プロキシ**

|  |  |
| --- | --- |
| **http.proxyの設定** | git config --global http.proxy http://proxy.example.com:8080 |
| git config --global http.proxy http://ID:PW@プロキシサーバのURL:ポート番号 |
| **https.proxyの設定** | git config --global https.proxy http://proxy.example.com:8080 |
| git config --global https.proxy http://ID:PW@プロキシサーバのURL:ポート番号 |

※ @ ⇒ %40

**その他**

|  |  |  |
| --- | --- | --- |
| git clone 時に SSL証明書エラー が出た時の対処方 | git config --global http.sslVerify false | |
| git status時に日本語のファイル名が文字化けしない | git config --global core.quotepath false |

**コミットメッセージ**

1行目は概要

2行目は空行

3行目は詳細

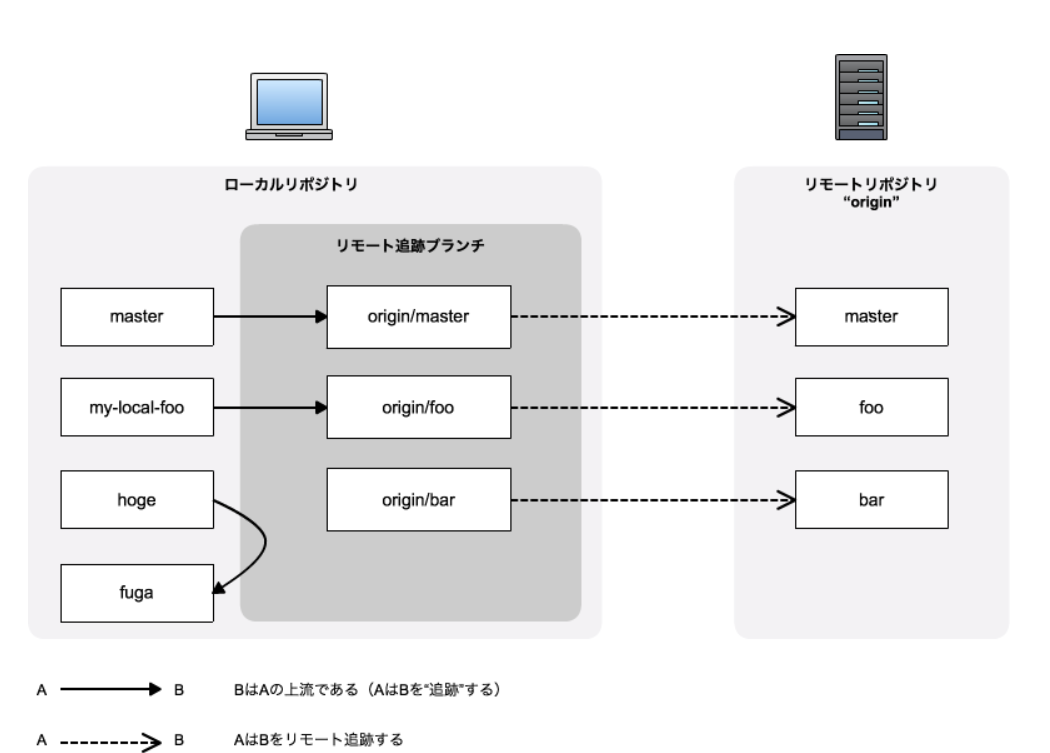
**リモート追跡ブランチ**

ローカルリポジトリにあり、リモートブランチを追跡するブランチ

例 origin/master

**上流ブランチ**

追跡対象のブランチ



<https://qiita.com/uasi/items/69368c17c79e99aaddbf>

**コンフィグ**

|  |  |
| --- | --- |
| 設定を表示する | git config -l |
| git config --list |

オプション

--system

--global

--local

**コミット**

|  |  |
| --- | --- |
| コミットする | git commit |
| メッセージ付きでコミット | git commit -m |
| 直前のコミットメッセージの修正 | git commit --amend |

**インデックス**

|  |  |
| --- | --- |
| カレントディレクトリ以下のファイルをインデックスに追加 | git add . |
| バージョン管理対象のファイルをインデックスに追加 | git add -u |
| 全てのファイルをインデックスに追加 | git add -A |
| <file>をインデックスに追加 | git add -- <file> |
| <file>をインデックスから削除 | git reset -- <file> |

**削除とリネーム**

|  |  |
| --- | --- |
| バージョン管理対象のファイルを削除 | git rm -- <file> |
| バージョン管理対象のファイル名を変更 | git mv -- <file-name> <new-file-name> |

**ステータス**

|  |  |
| --- | --- |
| ステータスを表示 | git status |
| ステータスを短く表示 | git status -s |

$ git status -s

A test1.txt

M test2.txt

MM test3.txt

D test4.txt

R test5.txt

?? test6.txt

**リセット**

|  |  |
| --- | --- |
| 全インデックスを取り消す | git reset |
| <file>をインデックスから削除 | git reset -- <file> |
| インデックスと作業ツリーを最新コミットに戻す | git reset --hard HEAD |
| インデックスと作業ツリーを<commit-id>に戻す | git reset --hard <commit-id> |

**ブランチ**

|  |  |
| --- | --- |
| ローカルブランチを表示 | git branch |
| リモート追跡ブランチを表示 | git branch -r |
| ローカルブランチとリモート追跡ブランチを表示 | git branch -a |
| currentブランチを起点に新しいブランチを作成する。  リモートブランチ、もしくはリモート追跡ブランチに同名のブランチがあっても無関係のブランチが作成される | git branch <branch> |
| ブランチの切り替え | git checkout <branch> |
| 同名のリモート追跡ブランチがあればそれを起点にローカルブランチを作成 |  |

**ファイル修正**

|  |  |
| --- | --- |
| ファイルの変更の取り消し | git checkout -- <file> |

**クローン**

|  |  |
| --- | --- |
| クローンする | git clone <url> |
| ブランチを指定してクローンする | git clone -b <branch> <url> |

**エラー**

・SSL証明書エラー git config --global http.sslVerify false

・認証エラー

・不正なURL

**フェッチ**

|  |  |
| --- | --- |
| git fetch | **リモートにある全てのブランチをリモート追跡ブランチとして取得する** |
| **全てのリモート追跡ブランチを更新する** |

**プル**

|  |  |
| --- | --- |
| git pull | **リモートにある全てのブランチをリモート追跡ブランチとして取得する** |
| **全てのリモート追跡ブランチを更新する** |
| **currentブランチに対応するリモート追跡ブランチをマージする** |
| git pull --rebase | **マージコミットしない** |

**プッシュ**

|  |  |
| --- | --- |
| git push | **currentブランチをリモートにプッシュする** |
| git push -f | **強制プッシュ** |
| git push --delete origin <branch> | **リモートブランチを削除** |

**クリーン**

|  |  |
| --- | --- |
| git clean -f | **remove untracked files** |
| git clean -f -d | **remove untracked files and directories** |

※pullの場合はブランチをcheckoutしないとマージされない

※フェッチもプルもプッシュも引数なしで使うべき

**リモート**

|  |  |
| --- | --- |
| リモートブランチのurlを表示 | **git remote -v** |

**revert**

|  |  |
| --- | --- |
| git revert <commit-id> | **コミットの打ち消し（指定したコミットとは全く逆のコミットをする）** |
| git revert -m 1 <commit-id> | **マージコミットの打ち消し** |

**1 ⇒ マージされる側のブランチ**

**2 ⇒ マージする側のブランチ**

<https://qiita.com/shizuma/items/313ed581e071f53a4d2e>

**エラー**

・コンフリクト

・コミット忘れ

$ git pull

error: Your local changes to the following files would be overwritten by merge:

test6.txt

$ git pull --rebase

error: cannot pull with rebase: Your index contains uncommitted changes.

error: please commit or stash them.

⇒・commitしたいファイルはcommitする

・その他のファイルはreset hardする

・バージョン管理対象でないファイルを削除する