


基本的なGitコマンドまとめ

🔖 Git(/tags/Git) 5346

 2m1tsu3 (/2m1tsu3) 2015年07月02日に更新

🔄 (/2m1tsu3/items/6d49374230afab251337/revisions) 4

📝 編集リクエスト (/2m1tsu3/items/6d49374230afab251337/patches) 1

👍 430 いいね 💬 2 コメント

👍 いいね



(/JUNKEI)



(/kusokamayrou)



(/clubtrack9999)



(/kazuhikoyamashita)



(/mki_skt)



(/sasayabaku)



(/3to_day) (/haneru)



(/chanchihiro) (/flanny)



... (/2m1tsu3/items/6d49374230afab251337/likers)

📁 ストック ...

🕒 この記事は最終更新日から1年以上が経過しています。

基本的なgitコマンドまとめ。

そもそも自分用に作ったものなので見づらいです。

init

```
$ git init
```

ディレクトリにリポジトリを作成。

```
$ git init --bare
```

ベアリポジトリの作成

```
$ git init --shared
```

グループ書き込み権限を有効にする。

Tweet 0



👍 Like 0

118



(/2m1tsu3)

2m1tsu3 (/2m1tsu3)

524 Contribution

👤 フォロー

人気の投稿

- 基本的なGitコマンドまとめ (/2m1tsu3/items)

add

```
$ git add [filename]
```

ファイルやディレクトリをインデックスに登録.

```
$ git add -A
```

すべての変更を含むワークツリーの内容をインデックスに追加.

```
$ git add -u
```

以前コミットしたことがあるファイルだけインデックスに追加.

commit

```
$ git commit
```

インデックスに追加されたファイルをコミットする.

```
$ git commit -m "[comment]"
```

コミットメッセージを同時に指定.

```
$ git commit -a
```

変更されたファイル(新規を除く)をインデックスに追加し,コミットする.

```
$ git commit --amend
```

直前のコミットを修正する.

status

```
$ git status
```

前回のコミットと比較してどのファイルが変更されたかを表示.

log

```
$ git log
```

コミットログを参照する.

```
$ git log --oneline
```

コミットログの先頭7桁のコミットIDを表示する.

```
$ git log --decorate
```

コミットログのHEADの位置を明示する.

```
$ git log --graph
```

/6d49374230afab251337)

- RedmineとGitの連携メモ (/2m1tsu3/items/0e7a111543f52ebe4ff1)
- Redmineインストールメモ (/2m1tsu3/items/e67d9fe7985d710c958e)
- CentOSでJenkinsインストール&プロキシ設定 (/2m1tsu3/items/3ec797b7f4a50a123f2c)
- OpenStack on CentOS7引っかけりリスト (/2m1tsu3/items/35d34df6f57271679461)

init

add

commit

status

log

diff

checkout

show

reset

rm

mv

revert

rebase

clone

push

pull

remote

fetch

branch

merge

tag

stash

コミットログを縦グラフで表示する.

```
$ git log --grep [filter]
```

指定した文字がコミットログに含まれるコミットを表示する.

reflog

cherry-pick

config

diff

```
$ git diff
```

インデックスとワーキングツリーの差異を表示.

```
$ git diff --cache
```

HEADとインデックスの差分を表示.

```
$ git diff [commit id 1] [commit id 2]
```

コミット間の差異を表示する.

checkout

```
$ git checkout [commit id] [filename]
```

コミットされた過去のファイルを復元する.

```
$ git checkout [branch]
```

ブランチを変更する.

```
$ git checkout --ours [filename]
```

マージでコンフリクトしたときに情報を指定してファイル内容を採用する.

```
$ git checkout --theirs [filename]
```

マージでコンフリクトしたときに下方を指定してファイル内容を採用する.

show

```
$ git show
```

最新のコミット内容を表示.

```
$ git show [tagname]
```

タグを指定してコミット内容を表示.

reset

```
$ git reset [commit id]
```

インデックスを現在のHEADの状態にする.

```
$ git reset HEAD [filename]
```

インデックスからファイルをアンステージする.

```
$ git reset --hard [commit id]
```

ワークツリーを含めたすべてをコミットIDの状態に戻す.

```
$ git reset --hard HEAD@[number]
```

git reflogで確認した番号の状態に戻す.

```
$ git reset --hard ORIG_HEAD
```

直前の状態に戻す.

rm

```
$ git rm [filename]
```

ワークツリーとインデックスからファイルを削除.

```
$ git rm --cache [filename]
```

インデックスのファイルを削除.

mv

```
$ git mv [filename 1] [filename 2]
```

ファイル名を変更(インデックスとワークツリーに同ファイル存在時)

revert

```
$ git revert [commit id]
```

コミットIDのコミットを取り消す.

rebase

```
$ git rebase -i [commit id]
```

コミットIDから古い順にコミットが表示され,コミットの行を消すとコミットの取り消し,先頭のpickをほかのものに置き換えるとコミットメッセージなどの編集が可能になる.

```
$ git rebase --abort
```

直前のgit rebaseの編集を中止する.

```
$ git rebase --continue
```

git rebaseの変更を適応する.

clone

```
$ git clone [repository PATH] [new repository PATH]
```

リポジトリをコピーする.

push

```
$ git push [remote repository] [branch]
```

リモートリポジトリに変更を書き込む.

```
$ git push [remote repository] --tags
```

リモートリポジトリにすべてのタグをアップロードする.

```
$ git push [remote repository] [tagname]
```

リモートリポジトリに指定したタグをアップロードする.

```
$ git push [remote repository] :[branch or tagname]
```

指定したブランチ,もしくはタグをリモートリポジトリから削除する.

pull

```
$ git pull [remote repository] [branch]
```

リモートリポジトリの変更を取り込む.

remote

```
$ git remote
```

リモートリポジトリの一覧表示

```
$ git remote add [username] [remote repository PATH]
```

名前とリモートリポジトリを関連付けする(リモートリポジトリの追加).

```
$ git remote rename [remoterepository] [new name]
```

リモートリポジトリの名前を変更する.

```
$ git remote show [remote repository]
```

リモートリポジトリの情報を見る.

```
$ git remote prune [remote repository]
```

リモートリポジトリで排除されたブランチをローカルからも削除する.

fetch

```
$ git fetch [remote repository]
```

リモートリポジトリの最新情報を追加する.

```
$ git fetch --prune
```

リモートリポジトリの削除情報をローカルに更新する.

branch

```
$ git branch &[new branch]
```

現在のブランチの確認.&新しいブランチを作成する.

```
$ git branch -a
```

すべてのブランチを確認する.

```
$ git branch -r
```

リモートブランチを確認する.

```
$ git branch -d [branch]
```

ブランチを削除する.

```
$ git branch -m [branch] [new branchname]
```

ブランチの名前を変更する.

```
$ git branch --set-upstream [my branch] [other branch]
```

他のユーザーのブランチと自分のブランチを関連付ける.

merge

```
$ git merge [branch]
```

現在のブランチをほかのブランチとマージする.

tag

```
$ git tag
```

タグの一覧を表示.

```
$ git tag -n[number]@
```

タグとそのメッセージ[行数指定]の一覧を表示する(行数指定なしの場合 1 行).

```
$ git tag -l [filter]
```

タグを,フィルターをかけて表示する.

```
$ git tag [tagname]
```

現在のコミットIDにタグを関連付けする.

```
$ git tag [tagname] [commit id]
```

コミットIDを指定してタグを関連付けする.

```
$ git tag -a [tagname]
```

現在のコミットIDにメッセージ付きのタグを関連付けする.

```
$ git tag -d [tagname]
```

指定したタグを削除する.

stash

```
$ git stash
```

現在の状態を保存する.

```
$ git stash save "[message]"
```

メッセージ付きで現在の状態を保存する.

```
$ git stash list
```

保存した状態の一覧を表示する.

```
$ git stash pop
```

最新の保存状態を復元する.

```
$ git stash pop stash@[number]
```

番号を指定して保存状態を復元する.

```
$ git stash apply
```

保存状態をリストに残したまま最新の保存状態を復元する.

```
$ git stash apply stash@[number]
```

保存状態をリストに残したまま指定した番号の保存状態を復元する.

```
$ git stash drop stash@[number]
```

保存状態を削除する.

```
$ git stash clear
```

保存状態をすべて削除する.

reflog

```
$ git reflog
```

過去にHEADが指していたコミット一覧を表示する。

```
$ git reflog [branch]
```

ブランチを指定して過去にHEADが指していたコミット一覧を表示する。

cherry-pick

```
$ git cherry-pick [commit id]
```

別のブランチのコミットを現在のブランチにコピーする。

config

```
$ git config -l
```

使用されるリポジトリの設定を表示する。

```
$ git config --global user.name [username]
```

ユーザ名の設定。

```
$ git config --global user.email [email address]
```

メールアドレスの設定。

```
$ git config --global color.ui auto
```

出力結果を色づけする。

👍 いいね

430

🔖

ストック

👤

(/Reds) (/habu0104)

(/zaoriku0) (/kasumani)

(/zokka) (/Runfun)

(/jjjiii26) (/mimosafa)

(/akira-hamada)

(/sogaoh)


... (/2m1tsu3/items/6d49374230afab251337/likers)

📁


編集リクエスト (/drafts/6d49374230afab251337/edit)

...


🔗 この記事は以下の記事からリンクされています



RedmineとGitの連携メモ (/2m1tsu3/items/0e7a111543f52ebe4ff1#_reference-1fdcb938bfd47009a55a) からリンク
3年以上前



git まとめ ※自分用メモ (/dawn_628/items/3431a36228547e4df3e9#_reference-dde6ac3f819f65b24df4) からリンク
約1年前



GitHubのマージとリベースについて (/masatomix/items/4bdb5f25de632c97f37f#_reference-551e584b2477d80c70fe) からリンク
9ヶ月前

🌐

yasumodev (/yasumodev) 1186 contribution

2015-06-30 10:12

👍 いいね

2

```
$git commit -m "[comment]"

$git commit -a
```

commitの章にある、上記コマンド例の「-」が本当のハイフン「-」ではないようです。コピペして実行するとエラーになりました。

👤

2m1tsu3 (/2m1tsu3) 524 contribution

2015-07-02 15:55

👍 いいね

2

ご指摘ありがとうございます。
修正しました。

9 / 9

2017/10/06 14:29