

Система контролю версій Git: розширена шпаргалка

🕒 24.04.2019 👤 Cinex 📁 Linux, Windows, Веб-розробка 💬 0

ПОШУК

ПОЗНАЧКИ

[APACHE](#)[AUTORUN](#)[BASH](#)[BITCOIN](#)[BLOCKCHAIN](#)[COMPOSER](#)[CRYPTOCURRENCY](#)[CSS](#)[DATABASE](#)[DJANGO](#)[DOCKER](#)[ELOQUENT](#)[ENGLISH](#)[ENUM](#)[GAMES](#)[GIMP](#)[GIT](#)[HTML](#)[JAVA](#)[JAVASCRIPT](#)[JQUERY](#)[LARAVEL](#)[MERCURIAL](#)[PARSING](#)[PHOTOSHOP](#)[PHP](#)[PHPSTORM](#)

Most Frequently Used Git Commands

Git

Git — це популярна система контролю версій. Її використовує практично кожен проект, чия діяльність так чи інакше пов'язана з ІТ.

Запам'ятати всі команди з першого разу навряд чи вийде, тому ми для вас приготували розширену шпаргалку.

Корисні керівництва по Git

```
1 git help -g
```

Пошук за змістом

[PLAYONLINUX](#)[PLUGINS](#)[PROBLEM SOLVING](#)[PYTHON](#)[REGEXP](#)[SEO](#)[SOFTWARE ENGINEERING](#)[SPRING](#)[SQL](#)[TELEGRAM](#)[TIPS AND TRICKS](#)[TIPS AND TRICKS](#)[UNIT TESTING](#)[WORDPRESS](#)[ВІРШІ](#)[ШПАРГАЛКИ](#)

ОСТАННІ НОТАТКИ

20 Laravel Eloquent порад і трюків

15.12.2019

Як зробити скріншот сайту по URL на PHP

04.08.2019

Docker Cheat Sheet

05.07.2019

Гарячі клавіші Ubuntu Linux

01.07.2019

Git happens! 6 типових помилок Git і як їх виправити

22.06.2019

НЕДАВНІ КОМЕНТАРІ

```
1 git log -S'<a term in the source>'
```

Віддалена синхронізація і перезапис локальних змін

```
1 git fetch origin && git reset --hard origin/master && git clean -f -d
```

Список всіх файлів до комітів

```
1 git ls-tree --name-only -r <commit-ish>
```

Скасування комітів

```
1 git update-ref -d HEAD
```

Список всіх конфліктуючих файлів

```
1 git diff --name-only --diff-filter=U
```

Список всіх файлів, змінених комітів

```
1 git diff-tree --no-commit-id --name-only -r <commit-ish>
```

Зміни з моменту останнього коміта

```
1 git diff
```

Зміни, виконані для коміта

```
1 git diff --cached
```

Альтернатива:

```
1 git diff --staged
```

Показати підготовлені / непідготовлені файли для комітів

```
1 git diff HEAD
```

Всі гілки, які вже з'єднані з гілкою master

```
1 git branch --merged master
```

Швидкий перехід до попередньої гілки

```
1 git checkout -
```

Альтернатива:

```
1 git checkout @{-1}
```

Видалити гілки, які вже об'єднані з master

```
1 git branch --merged master | grep -v '^*' | xargs -n 1 git branch -d
```

Альтернатива:

```
1 git branch --merged master | grep -v '^*\| master' | xargs -n 1 git branch -d
```

Всі гілки і які виходять з них, а також останні коміти на гілці

```
1 git branch -vv
```

Відстеження гілки

```
1 git branch -u origin/mybranch
```

Видалення локальної гілки

```
1 git branch -d <local_branchname>
```

Видалення гілки

```
1 git push origin --delete <remote_branchname>
```

Альтернатива:

```
1 git push origin :<remote_branchname>
```

Видалення локальної мітки

```
1 git tag -d <tag-name>
```

Видалення мітки

```
1 git push origin :refs/tags/<tag-name>
```

Скасування локальних змін

```
1 git checkout -- <file_name>
```

Скасування коміта, створення нового коміта

```
1 git revert <commit-ish>
```

Скасування коміта, переважно для приватних гілок

```
1 git reset <commit-ish>
```

Повтор попереднього повідомлення коміта

```
1 git commit -v --amend
```

Перегляд історії комітів для поточної гілки

```
1 git cherry -v master
```

Змінити автора

```
1 git commit --amend --author='Author Name <email@address.com>'
```

Скидання автора, після того як автор був змінений в глобальній конфігурації

```
1 git commit --amend --reset-author --no-edit
```

Зміна віддаленого URL

```
1 git remote set-url origin <URL>
```

Отримання списку всіх віддалених посилань

```
1 git remote
```

Альтернатива:

```
1 git remote show
```

Отримання списку всіх локальних і віддалених гілок

```
1 git branch -a
```

Отримання списку тільки віддалених гілок

```
1 git branch -r
```

Отримання списку змін файлу, а не весь файл

```
1 git add -p
```

Отримання git bash

```
1 curl http://git.io/vfhol > ~/.git-completion.bash && echo '[ -f ~/.git-completion.bash ]
```

Що змінилося за два тижні?

```
1 git log --no-merges --raw --since='2 weeks ago'
```

Альтернатива:

```
1 git whatchanged --since='2 weeks ago'
```

Подивитися всі коміти, зроблені з моменту створення гілки майстра

```
1 git log --no-merges --stat --reverse master..
```

Вибір комітів по гілках за допомогою cherry-pick

```
1 git checkout <branch-name> && git cherry-pick <commit-ish>
```

Показати гілки, що містять commit-hash

```
1 git branch -a --contains <commit-ish>
```

Альтернатива:

```
1 git branch --contains <commit-ish>
```

Git-аліаси

```
1 git config --global alias.<handle> <command>
2 git config --global alias.st status
```

Збереження поточного стану відслідковуються файлів без комітів

```
1 git stash
```

Альтернатива:

```
1 git stash save
```

Збереження поточного стану змін в файлах, які відслідковуються


```
1 git stash -k
```

Альтернатива:

```
1 git stash --keep-index  
2 git stash save --keep-index
```

Збереження поточного стану, включаючи невикористовувані файли

```
1 git stash -u
```

Альтернатива:

```
1 git stash save -u  
2 git stash save --include-untracked
```

Збереження поточного стану з повідомленням

```
1 git stash save <message>
```

Збереження поточного стану всіх файлів (ігноруються, не відслідковуються і відслідковуються)

```
1 git stash -a
```

Альтернатива:

```
1 git stash --all  
2 git stash save --all
```

Показати список всіх даних, збережених в схованці

```
1 git stash list
```

Ховання без видалення зі списку

```
1 git stash apply <stash@{n}>
```

Взяти файл зі схованки

```
1 git checkout <stash@{n}> -- <file_path>
```

Альтернатива:

```
1 git checkout stash@{0} -- <file_path>
```

Показати всі файли, які відслідковують

```
1 git ls-files -t
```

Показати всі файли, які не відслідковуються

```
1 git ls-files --others
```

Показати всі проігноровані файли

```
1 git ls-files --others -i --exclude-standard
```

Створення нового робочого дерева зі репозиторію (git 2.5)

```
1 git worktree add -b <branch-name> <path> <start-point>
```

Відстеження файлів без видалення

```
1 git rm --cached <file_path>
```

Альтернатива:

```
1 git rm --cached -r <directory_path>
```

Перед видаленням непотрібних файлів / каталогів, отримати список цих файлів / каталогів

```
1 git clean -n
```

Видалення не відслідковуваних файлів

```
1 git clean -f
```

Видалення каталогу, який не відслідковується

```
1 git clean -f -d
```

Оновлення всіх підмодулів

```
1 git submodule foreach git pull
```

Альтернатива:

```
1 git submodule update --init --recursive  
2 git submodule update --remote
```

Показати всі коміти в поточній гілці

```
1 git cherry -v master
```

Альтернатива:

```
1 git cherry -v master <branch-to-be-merged>
```

Перейменування гілки

```
1 git branch -m <new-branch-name>
```

Альтернатива:

```
1 git branch -m [<old-branch-name>] <new-branch-name>
```

Архівация гілки master

```
1 git archive master --format=zip --output=master.zip
```

Змінити попередній коміт без повідомлення про редагування

```
1 git add --all && git commit --amend --no-edit
```

Видалення посилань, які посилаються на віддалені гілки

```
1 git fetch -p
```

Альтернатива:

```
1 git remote prune origin
```

Візуалізація дерева версій

```
1 git log --pretty=oneline --graph --decorate --all
```

Альтернатива:

```
1 gitk --all
```

Розгортання вкладеної папки git в gh-pages

```
1 git subtree push --prefix subfolder_name origin gh-pages
```

Додавання проекту до сховища з використанням піддерева

```
1 git subtree add --prefix=<directory_name>/<project_name> --squash git@github.com:<username>
```

Отримання останніх змін в репозиторії проекту з використанням піддерева

```
1 git subtree pull --prefix=<directory_name>/<project_name> --squash git@github.com:<username>
```

Експорт історії гілки в файл

```
1 git bundle create <file> <branch-name>
```

Імпорт з пакета

```
1 git clone repo.bundle <repo-dir> -b <branch-name>
```

Отримання імені поточної гілки

```
1 git rev-parse --abbrev-ref HEAD
```

Ігнорувати один файл при коміті

```
1 git update-index --assume-unchanged Changelog; git commit -a; git update-index --no-assun
```

Запит за ідентифікатором в локальну гілку

```
1 git fetch origin pull/<id>/head:<branch-name>
```

Альтернатива:

```
1 git pull origin pull/<id>/head:<branch-name>
```

Показати саму використовувану мітку в поточній гілці

```
1 git describe --tags --abbrev=0
```

Показати вбудоване слово diff

```
1 git diff --word-diff
```

Не зважати на зміни відслідковується файлу

```
1 git update-index --assume-unchanged <file_name>
```

Скасування без змін

```
1 git update-index --no-assume-unchanged <file_name>
```

Видалення файлів з .gitignore

```
1 git clean -X -f
```

Відновлення видаленого файлу

```
1 git checkout <deleting_commit>^ -- <file_path>
```

Відновлення файлів до певного коміту

```
1 git checkout <commit-ish> -- <file_path>
```

Список всіх аліасів і конфігів

```
1 git config --list
```

Додавання кастомного редактора

```
1 git config --global core.editor '$EDITOR'
```

Автоматичне виправлення помилок

```
1 git config --global help.autocorrect 1
```

Перевірте, чи є зміна частиною релізу

```
1 git name-rev --name-only <SHA-1>
```

Позначає ваш коміт, як виправлення попереднього коміту

```
1 git commit --fixup <SHA-1>
```

Пропустити область під час коміту

```
1 git commit --only <file_path>
```

Список ігнорованих файлів

```
1 git check-ignore *
```

Статус ігнорованих файлів

```
1 git status --ignored
```

Список n останніх комітів

```
1 git log -<n>
```

Альтернатива:

```
1 git log -n <n>
```

Відкрити всі конфліктуючі файли в редакторі

```
1 git diff --name-only | uniq | xargs $EDITOR
```

Миттєвий перегляд робочого сховища в gitweb

```
1 git instaweb [--local] [--httpd=<httpd>] [--port=<port>] [--browser=<browser>]
```


Перегляд GPG підпису в журналі комітів

```
1 git log --show-signature
```

Видалення запису в глобальній конфігурації

```
1 git config --global --unset <entry-name>
```

Створення нової гілки без історії

```
1 git checkout --orphan <branch_name>
```

Витягти файл з іншої гілки

```
1 git show <branch_name>:<file_name>
```

Змінити попередні два коміти з інтерактивною перестановкою

```
1 git rebase --interactive HEAD~2
```

Список всіх гілок — WIP

```
1 git checkout master && git branch --no-merged
```

Бінарний пошук помилок

```
1 git bisect start
2 git bisect bad
3 git bisect good v2.6.13-rc2
4 git bisect bad
5 git bisect good
```

```
6 git bisect reset
```

Створення списку комітів і змін в конкретному файлі

```
1 git log --follow -p -- <file_path>
```

Клонувати гілку

```
1 git clone -b <branch-name> --single-branch https://github.com/user/repo.git
```

Створення та редагування нової гілки

```
1 git checkout -b <branch-name>
```

Альтернатива:

```
1 git branch <branch-name> && git checkout <branch-name>
```

Відключити кольоровий вивід git в термінал

```
1 git config --global color.ui false
```

Налаштування кольору

```
1 git config --global <specific command e.g branch, diff> <true, false or always>
```

Клонувати репозиторій

```
1 git clone https://github.com/user/repo.git --depth 1
```

Пошук комітів по всіх гілках

```
1 git log --all --grep='<given-text>'
```

Отримання першого коміту в гілці

```
1 git log master..<branch-name> --oneline | tail -1
```

Показати коміти по авторам і назві

```
1 git shortlog
```

Кількість коммітов в гілці

```
1 git rev-list --count <branch-name>
```

Аліас: git undo

```
1 git config --global alias.undo '!f() { git reset --hard $(git rev-parse --abbrev-ref HEAD); }
```

Додавання приміток до об'єкту

```
1 git notes add -m 'Note on the previous commit....'
```

Показати всі git-notes

```
1 git log --show-notes='*'
```

Застосування комітів з іншого сховища

```
1 git --git-dir=<source-dir>/.git format-patch -k -1 --stdout <SHA1> | git am -3 -k
```

Список заборонених git комітів

```
1 git log --branches --not --remotes
```

Альтернатива:

```
1 git log @{u}..  
2 git cherry -v
```

Змінити git конфігурацію

```
1 git config [--global] --edit
```

Показати логічну змінну Git

```
1 git var -l | <variable>
```

Дізнатися назву репозиторію

```
1 git rev-parse --show-toplevel
```

Показати логи між діапазоном дат

```
1 git log --since='FEB 1 2017' --until='FEB 14 2017'
```

Виключити автора з логів

```
1 git log --perl-regexp --author='^((?!excluded-author-regex).*)'
```

Створення зведення очікують змін

```
1 git request-pull v1.0 https://git.ko.xz/project master:for-linux
```

Отримання списку посилань в віддаленому репозиторії

```
1 git ls-remote git://git.kernel.org/pub/scm/git/git.git
```

Список всіх git аліасів

```
1 git config -l | grep alias | sed 's/^alias\././g'
```

Альтернатива:

```
1 git config -l | grep alias | cut -d '.' -f 2
```

Перенесення локальної гілки в віддалений репозиторій

```
1 git push -u origin <branch_name>
```

Джерело



GIT

TIPS AND TRICKS

ШПАРГАЛКИ



« ПОПЕРЕДНІЙ

10 корисних Git команд, які
полегшать роботу

ДАЛІ »

Laravel - створення і використання
зв'язків між таблицями БД MySQL.



ЗАЛИШТЕ ПЕРШИЙ КОМЕНТАР

Залишити коментар

Вашу адресу електронної пошти не буде опубліковано

Коментувати

Ім'я*

Email *

☐ Збережіть моє ім'я, електронну пошту у цьому веб-переглядачі наступного разу, коли я коментуватиму.

ОПУБЛІКУВАТИ КОМЕНТАР

Copyright © 2020