# Полезные руководства по Git

|  |  |
| --- | --- |
| 1 | git help -g |

# Поиск по содержанию

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

# Удаленная синхронизация и перезапись локальных изменений

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

# Список всех файлов до коммита

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

# Отмена коммита

|  |  |
| --- | --- |
| 1 | git update-ref -d HEAD |

* [Про Git, Github и Gitflow простыми словами](https://proglib.io/p/git-github-gitflow/)

# Список всех конфликтующих файлов

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

# Список всех файлов, измененных коммитом

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

# Изменения с момента последнего коммита

|  |  |
| --- | --- |
| 1 | git diff |

# Изменения, выполненные для коммита

|  |  |
| --- | --- |
| 1 | git diff --cached |

## Альтернатива:

|  |  |
| --- | --- |
| 1 | git diff --staged |

# Показать подготовленные/неподготовленные файлы для коммита

|  |  |
| --- | --- |
| 1 | git diff HEAD |

# Все ветки, которые уже соединены с веткой master

|  |  |
| --- | --- |
| 1 | git branch --merged master |

# Быстрый переход к предыдущей ветке

|  |  |
| --- | --- |
| 1 | git checkout - |

## Альтернатива:

|  |  |
| --- | --- |
| 1 | git checkout @{-1} |

# Удалить ветки, которые уже объединены с 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 |

# Все ветки и выходящие из них, а также последние коммиты на ветке

|  |  |
| --- | --- |
| 1 | git branch -vv |

# Отслеживание ветки

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

# Удаление локальной ветки

|  |  |
| --- | --- |
| 1 | git branch -d <local\_branchname> |

# Удаление ветки

|  |  |
| --- | --- |
| 1 | git push origin --delete <remote\_branchname> |

## Альтернатива:

|  |  |
| --- | --- |
| 1 | git push origin :<remote\_branchname> |

# Удаление локальной метки

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

# Удаление метки

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

# Отменена локальных изменений

|  |  |
| --- | --- |
| 1 | git checkout -- <file\_name> |

# Отмена коммита, создание нового коммита

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

# Отмена коммита, предпочительно для приватных веток

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

# Повтор предыдущего сообщение коммита

|  |  |
| --- | --- |
| 1 | git commit -v --amend |

# Просмотр истории коммитов для текущей ветки

|  |  |
| --- | --- |
| 1 | git cherry -v master |

# Изменить автора

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

* [Модели ветвления в Git: какую выбрать?](https://proglib.io/p/git-workflow/)

# Сброс автора, после того как автор был изменен в глобальной конфигурации

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

# Изменение удаленного URL

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

# Получение списка всех удаленных ссылок

|  |  |
| --- | --- |
| 1 | git remote |

## Альтернатива:

|  |  |
| --- | --- |
| 1 | git remote show |

# Получение списка всех локальных и удаленных веток

|  |  |
| --- | --- |
| 1 | git branch -a |

# Получение списка только удаленных веток

|  |  |
| --- | --- |
| 1 | git branch -r |

# Получение списка изменений файла, а не весь файл

|  |  |
| --- | --- |
| 1 | git add -p |

# Получение git bash

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

* [Git за полчаса: руководство для начинающих](https://proglib.io/p/git-for-half-an-hour/)

# Что изменилось за две недели?

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

## Альтернатива:

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

# Просмотреть все коммиты, сделанные с момента создания ветки мастера

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

# Выбор коммитов по веткам с помощью cherry-pick

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

# Показать ветки, содержащие commit-hash

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

## Альтернатива:

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

# Git-алиасы

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

# Сохранение текущего состояния отслеживаемых файлов без коммитов

|  |  |
| --- | --- |
| 1 | git stash |

## Альтернатива:

|  |  |
| --- | --- |
| 1 | git stash save |

# Сохранение текущего состояния изменений в отслеживаемых файлах

|  |  |
| --- | --- |
| 1 | git stash -k |

## Альтернатива:

|  |  |
| --- | --- |
| 1 | git stash --keep-index |

|  |  |
| --- | --- |
| 1 | git stash save --keep-index |

# Сохранение текущего состояния, включая неиспользуемые файлы

|  |  |
| --- | --- |
| 1 | git stash -u |

## Альтернатива:

|  |  |
| --- | --- |
| 1 | git stash save -u |

|  |  |
| --- | --- |
| 1 | git stash save --include-untracked |

# Сохранение текущего состояния с сообщением

|  |  |
| --- | --- |
| 1 | git stash save <message> |

# Сохранение текущего состояния всех файлов (игнорируются, не отслеживаются и отслеживаются)

|  |  |
| --- | --- |
| 1 | git stash -a |

## Альтернатива:

|  |  |
| --- | --- |
| 1 | git stash --all |

|  |  |
| --- | --- |
| 1 | git stash save --all |

# Показать список всех данных, сохраненных в тайнике

|  |  |
| --- | --- |
| 1 | git stash list |

# Прятанье без удаления из списка

|  |  |
| --- | --- |
| 1 | git stash apply <stash@{n}> |

# Взять файл из тайника

|  |  |
| --- | --- |
| 1 | git checkout <stash@{n}> -- <file\_path> |

## Альтернатива:

|  |  |
| --- | --- |
| 1 | git checkout stash@{0} -- <file\_path> |

# Показать все отслеживаемые файлы

|  |  |
| --- | --- |
| 1 | git ls-files -t |

# Показать все неотслеживаемые файлы

|  |  |
| --- | --- |
| 1 | git ls-files --others |

# Показать все проигнорированные файлы

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

# Создание нового рабочего дерева из репозитория (git 2.5)

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

# Отслеживание файлов без удаления

|  |  |
| --- | --- |
| 1 | git rm --cached <file\_path> |

## Альтернатива:

|  |  |
| --- | --- |
| 1 | git rm --cached -r <directory\_path> |

# Перед удалением ненужных файлов / каталогов, получить список этих файлов / каталогов

|  |  |
| --- | --- |
| 1 | git clean -n |

# Удаление неотслеживаемых файлов

|  |  |
| --- | --- |
| 1 | git clean -f |

# Удаление неотслеживаемого каталога

|  |  |
| --- | --- |
| 1 | git clean -f -d |

# Обновление всех подмодулей

|  |  |
| --- | --- |
| 1 | git submodule foreach git pull |

## Альтернатива:

|  |  |
| --- | --- |
| 1 | git submodule update --init --recursive |

|  |  |
| --- | --- |
| 1 | git submodule update --remote |

# Показать все коммиты в текущей ветке

|  |  |
| --- | --- |
| 1 | git cherry -v master |

## Альтернатива:

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

# Переименование ветки

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

## Альтернатива:

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

# Архивация ветки master

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

# Изменить предыдущий коммит без сообщения об редактировании

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

# Удаление ссылок, ссылающихся на удаленные ветки

|  |  |
| --- | --- |
| 1 | git fetch -p |

## Альтернатива:

|  |  |
| --- | --- |
| 1 | git remote prune origin |

# Визуализация дерева версий

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

## Альтернатива:

|  |  |
| --- | --- |
| 1 | gitk --all |

# Развертывание вложенной папки git в gh-pages

|  |  |
| --- | --- |
| 1 | git subtree push --prefix subfolder\_name origin gh-pages |

# Добавление проекта к репозиторию с использованием поддерева

|  |  |
| --- | --- |
| 1 | git subtree add --prefix=<directory\_name>/<project\_name> --squash git@github.com:<username>/<project\_name>.git master |

# Получение последних изменений в репозитории проекта с использованием поддерева

|  |  |
| --- | --- |
| 1 | git subtree pull --prefix=<directory\_name>/<project\_name> --squash git@github.com:<username>/<project\_name>.git master |

# Экспорт истории ветки в файл

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

# Импорт из пакета

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

# Получение имени текущей ветки

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

# Игнорировать один файл при коммите

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

# Запрос по идентификатору в локальную ветку

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

## Альтернатива:

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

# Показать самую используемую метку в текущей ветке

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

# Показать встроенное слово diff

|  |  |
| --- | --- |
| 1 | git diff --word-diff |

# Не учитывать изменения отслеживаемого файла

|  |  |
| --- | --- |
| 1 | git update-index --assume-unchanged <file\_name> |

# Отмена без изменений

|  |  |
| --- | --- |
| 1 | git update-index --no-assume-unchanged <file\_name> |

# Удаление файлов из .gitignore

|  |  |
| --- | --- |
| 1 | git clean -X -f |

# Восстановление удаленного файла

|  |  |
| --- | --- |
| 1 | git checkout <deleting\_commit>^ -- <file\_path> |

# Восстановление файлов к определенному коммиту

|  |  |
| --- | --- |
| 1 | git checkout <commit-ish> -- <file\_path> |

# Список всех алиасов и конфигов

|  |  |
| --- | --- |
| 1 | git config --list |

# Добавление пользовательских редакторов

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

# Автоматическое исправление опечаток

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

# Проверьте, является ли изменение частью релиза

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

# Помечает ваш коммит, как исправление предыдущего коммита

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

# Пропустить область во время коммита

|  |  |
| --- | --- |
| 1 | git commit --only <file\_path> |

# Список игнорируемых файлов

|  |  |
| --- | --- |
| 1 | git check-ignore \* |

# Статус игнорируемых файлов

|  |  |
| --- | --- |
| 1 | git status --ignored |

# Список n последних коммитов

|  |  |
| --- | --- |
| 1 | git log -<n> |

## Альтернатива:

|  |  |
| --- | --- |
| 1 | git log -n <n> |

# Открыть все конфликтующие файлы в редакторе

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

# Мгновенный просмотр рабочего репозитория в gitweb

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

# Просмотр GPG подписи в журнале коммитов

|  |  |
| --- | --- |
| 1 | git log --show-signature |

# Удаление записи в глобальной конфигурации

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

# Создание новой ветки без истории

|  |  |
| --- | --- |
| 1 | git checkout --orphan <branch\_name> |

# Извлечь файл из другой ветки

|  |  |
| --- | --- |
| 1 | git show <branch\_name>:<file\_name> |

# Изменить предыдущие два коммита с интерактивной перестановкой

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

# Список всех веток — WIP

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

# Бинарный поиск ошибок

|  |  |
| --- | --- |
| 1  2  3  4  5  6 | git bisect start  git bisect bad  git bisect good v2.6.13-rc2  git bisect bad  git bisect good  git bisect reset |

# Создание списка коммитов и изменений в конкретном файле

|  |  |
| --- | --- |
| 1 | git log --follow -p -- <file\_path> |

# Клонировать ветку

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

# Создание и редактирование новой ветки

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

## Альтернатива:

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

# Отключить цветной вывод git в терминал

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

# Настройки цвета

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

# Клонировать репозиторий

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

# Поиск коммитов по всем веткам

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

# Получение первого коммита в ветке

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

# Показать коммиты по авторам и названию

|  |  |
| --- | --- |
| 1 | git shortlog |

# Количество коммитов в ветке

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

# Алиас: git undo

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

# Добавление примечаний к объекту

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

# Показать все git-notes

|  |  |
| --- | --- |
| 1 | git log --show-notes='\*' |

# Применение коммита из другого репозитория

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

# Список запрещенных git коммитов

|  |  |
| --- | --- |
| 1 | git log --branches --not --remotes |

## Альтернатива:

|  |  |
| --- | --- |
| 1 | git log @{u}.. |

|  |  |
| --- | --- |
| 1 | git cherry -v |

# Изменить git конфигурацию

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

# Показать логическую переменную Git

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

# Узнать название репозитория

|  |  |
| --- | --- |
| 1 | git rev-parse --show-toplevel |

# Показать логи между диапазоном дат

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

# Исключить автора из логов

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

# Создание сводки ожидающих изменений

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

# Получение списка ссылок в удаленном репозитории

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

# Список всех git алиасов

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

## Альтернатива:

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

* [Исходная статья](https://github.com/git-tips/tips)

# Перенос локальной ветки в удаленный репозиторий

Shell

|  |  |
| --- | --- |
| 1 | git push -u origin <branch\_name> |