

[КАК СТАТЬ АВТОРОМ](#)[Неделя тестировщиков](#)[Пан или пропал: устраива...](#)**97.51**

Рейтинг

## Skillbox

Образовательная платформа Skillbox



fokus-lop 1 марта 2019 в 19:30

# 10 Git-команд, которые стоит знать разработчику

Блог компании Skillbox, Программирование\*, Git\*, Учебный процесс в IT

[Перевод](#)

Автор оригинала: Jeff Hale



В этой статье мы обсудим разные Git-команды, которые могут оказаться полезными для разработчика или специалиста по Big Data. Вы узнаете, как проверять, удалять и приводить код в порядок. А еще рассмотрим способы выхода из Vim и экономию времени с помощью псевдонимов Bash и конфигурации редактора Git.

**Skillbox рекомендует:** Практический курс «Мобильный разработчик PRO».

**Напоминаем:** для всех читателей «Хабра» — скидка 10 000 рублей при записи на любой курс Skillbox по промокоду «Хабр».

## Проверяем все и вся



- `git diff` — Посмотреть все изменения файла локально. При указании имени файла изменения будут показаны только для него.
- `git log` — Просмотреть историю коммита. Может также использоваться для файла с `git log -p my_file`. Введите `q`, чтобы выйти.
- `git blame my_file` — Просмотреть, кто, что и когда изменил в `my_file`.
- `git reflog` — Показать журнал изменений в заголовке локального репозитория. Отличный вариант для поиска утраченных данных.

Эти команды позволяют быстро проверить все, что нужно, без всяких проблем. Если что-то не так, Git дает большое количество возможностей для удаления или отката коммитов и изменений в файлах.

## Вернуть, как было

- `git reset`, `git checkout` и `git revert` — команды, которые используются, чтобы отменить какие-либо действия. Но они не такие и простые, с ними надо уметь обращаться.
- `git reset`, `git checkout` могут использоваться как для коммитов, так и для обычных файлов.

- `git revert` используется только для работы с коммитами.

Если вы работаете с собственными локальными коммитами, которые никак не связаны с командной работой, можно без проблем использовать все перечисленные команды.

Если вы работаете в коллективе и коммиты общие, тогда ваш выбор — `git revert`.



У каждой команды есть целый набор опций. Вот наиболее употребляемые:

`git reset --hard HEAD` — отмена проиндексированных и непроиндексированных изменений с момента последнего коммита.

Указываем вместо `HEAD` определенный коммит, чтобы отменить изменения, произошедшие после него. `--hard` отбрасываются оба типа изменений, о которых говорилось выше.

Не забывайте убедиться в том, что вы не отменяете коммит из опубликованной ветки, от которой зависят другие члены команды.

`git checkout my_commit` — отмена изменений `my_commit`.

`HEAD` часто используется для `my_commit`, чтобы отменить изменения в вашем локальном рабочем каталоге с момента последней фиксации.

`checkout` лучше всего использовать для локальных отмен. В этом случае коммиты из удаленной ветки, от которой зависят ваши коллеги, не будут затронуты!

Если вы используете `checkout` с веткой вместо коммита, `HEAD` переключается на

указанную ветвь, а рабочий каталог обновляется для соответствия изменениям. Это самое распространенное использование этой команды.

`git revert my_commit` — отмена последствий изменений в `my_commit`. `revert` выполняет новый коммит после отмены изменений.

`revert` безопасен для общих проектов, поскольку команда не перезаписывает изменения, от которых могут зависеть другие ветки.



Иногда вы просто хотите удалить неотслеживаемые файлы в вашем локальном каталоге. К примеру, запустив какой-то код, который создал много разных типов файлов, которые вам не нужны. К сожалению, `Clean` поможет мгновенно удалить их!

`git clean -n` — удаление неотслеживаемых файлов в локальной рабочей директории.

- n — флаг для пробного запуска, ничего не удаляется.
- f — флаг для удаления файлов.
- d — флаг для удаления неотслеживаемых директорий.

По умолчанию неотслеживаемые файлы .gitignore не будут удалены, но это можно изменить.



## Наводим порядки

`git commit --amend` — добавляем поэтапные изменения в последний коммит.

Если ничего не проиндексировано, команда позволяет вам редактировать последнее сообщение коммита. Используйте команду только в том случае, если коммит не был объединен с удаленной master-веткой.

`git push my_remote --tags` — отправка локальных тэгов в удаленный репозиторий.  
Хороший вариант для присвоения версий изменениям.

## Помогите, я застрял в Vim и не могу выбраться!

Git в некоторых случаях открывает сессию редактора Vim. И если вы не слишком

хорошо знакомы с ним, то можете оказаться в затруднительной ситуации. Да и не только вы — к примеру, на Stack Overflow более 4 тысяч пользователей хотят знать, как выбраться из Vim.



Вот четырехэтапный план, который поможет закрыть Vim и сохранить изменения:

- Нажимаем `i`.
- Вводим сообщение коммита в первой строке.
- `Esc`.
- Вводим `:x`.

Все, вы свободны!

Изменяем редактор по умолчанию.

Вы можете избавиться от Vim совсем, если смените редактор по умолчанию. Вот [команды](#) для работы с популярными редакторами. Пример выбора другого редактора, в нашем случае Atom:

```
git config --global core.editor «atom --wait»
```

## Ярлыки для команд Git

А вот способ, который позволяет добавлять ярлыки для Git-команд, для вашего `.bash_profile`.

```
alias gs='git status '  
alias ga='git add '  
alias gaa='git add -A '  
alias gb='git branch '  
alias gc='git commit '  
alias gcm='git commit -m '  
alias go='git checkout '
```

Больше информации о `.bash_profile` можно получить [здесь](#).

Что касается способа, приведенного выше, то теперь вы можете использовать `gs` вместо `git status`.

Собственно, это все на сегодня. Если есть возможность, укажите в комментариях, какие Git-команды используете вы и почему.

#### Skillbox рекомендует:

- Двухлетний практический курс «Я — веб-разработчик PRO».
- Онлайн-курс «C#-разработчик с 0».
- Практический годовой курс «PHP-разработчик с 0 до PRO».

**Теги:** [git](#), [команды git](#), [программирование](#), [обучение](#), [skillbox](#)

**Хабы:** [Блог компании Skillbox](#), [Программирование](#), [Git](#), [Учебный процесс в IT](#)

◆ +11

🔖 380

💬 26



#### Редакторский дайджест

Присылаем лучшие статьи раз в месяц







Skillbox

Образовательная платформа Skillbox

[Сайт](#) [Facebook](#) [ВКонтакте](#) [Instagram](#) [Telegram](#)

34

0

Карма

Рейтинг

@fokus-lop

Пользователь

## Комментарии 26

 **ilyapirogov**  
01.03.2019 в 20:20

`git stash` — отложить все текущие изменения в локальное хранилище  
`git stash pop` — забрать последние сохраненные изменения из хранилища

Полезно если вы не хотите пока комитить код, но вам срочно нужно переключиться на другую ветку.

 +7 [Ответить](#) **Graf54r**  
01.03.2019 в 22:09

помню я как то думал что если переключится на другую ветку, то в этой все останется также, а комитить я часто не любил. Эх...

 +2 [Ответить](#) **SVlad**  
02.03.2019 в 23:03

Полезно если вы не хотите пока комитить код, но вам срочно нужно переключиться на другую ветку.

Всегда в таких случаях делаю `commit`, озаглавленный WIP — нагляднее и не забудешь.  
Потом просто делаю `amend`.  
`stash` скорее нужен для упрощённого `rebase` между ветками.

 0 [Ответить](#) **Prost971**  
01.03.2019 в 20:21



+1 статья в закладки. Хорошие шпаргалки по гиту на весь — золото)

 -2 [Ответить](#)



 **JTG**  
01.03.2019 в 20:36

Отвратительная статья. Если бы я не умел работать с git, ничего бы не понял. Даже хуже, после этой статьи:

- У меня есть изменённые файлы, но git diff ничего не показывает ;( Что ещё за staging area?
- Я сделал git reset --hard и всё пропало!
- "#1234 foo" и «Revert „#1234 foo“» потому что «работаю в коллективе и коммиты общие, мой выбор — git revert».
- git commit --amend не работает! Но ведь я читал статью, мой коммит не был объединен с удаленной master-веткой! Я же пушил его только в develop!

 +17 [Ответить](#)



 **fireSparrow**  
01.03.2019 в 21:20

Так нигде и не сказано, что это вводная статья для чайников.

Статья называется «10 Git-команд, которые стоит знать разработчику». И то, что обещает название, в тексте полностью раскрыто.

 0 [Ответить](#)



 **JTG**  
01.03.2019 в 21:48

Тогда выходит, что для чайников эта статья вредна, а для не-чайников — бесполезна. Например, под заголовком «наводим порядки» ожидаешь увидеть что-то про rebase, а не «как запустить метку» (при этом как поставить эту-самую метку — не уточняется).

То же самое с

Не забывайте убедиться в том, что вы не отменяете коммит из удаленной ветки, от которой зависят другие члены команды.

Убедиться при помощи телепатии. Про «git branch -r --contains» разработчику, видимо, знать не стоит. Спасибо, Jeff Hale, очень полезная статья!

 +8 [Ответить](#)




 **fireSparrow**  
01.03.2019 в 22:01

Это просто шпаргалка.

Если она ничего нового вам не сообщила — то просто поищите другие материалы по гиту. Есть пользователи, которые посчитали её полезной для себя, если судить по оценкам.

 0 Ответить

 **usego**  
01.03.2019 в 20:46

Оно всё надо при пользовании Source Tree? До сих пор в командной строке использую максимум 5 команд и то нечасто. Всё остальное на много удобнее в UI смотреть и делать.

 -5 Ответить



 **nick\_gabpe**  
01.03.2019 в 23:04

Вопрос "[How to exit vim?](#)" имеет всего 4193 «голоса за», а вот просматривается гораздо чаще viewed 1,853,728. Так что скорее не 4 тысячи, а почти 2 миллиона человек :)

 +3 Ответить



 **vyo**  
01.03.2019 в 23:19

Полному нубу — бесполезно. Новичку — вредно. Немного попользовавшемуся гитом — снова бесполезно.

Даже не знаю, на кого статья ориентирована. Сам гит освоил по чему-то вроде офф. мануала, а дальше — StackOverflow (не забывая читать непринятые ответы, конечно).

 +6 Ответить



 **AnROm**  
02.03.2019 в 02:04

И еще одна очередная статья про N Git-команд, которые нужно знать разработчику

 +5 Ответить



 **netch80**  
02.03.2019 в 03:58

Вот так лучше:

> git blame my\_life — Просмотреть, кто, что и когда изменил в my\_life.

всего две буквы переставил, а какой эффект :)

 +4 Ответить



 **KvanTTT**  
02.03.2019 в 04:57

Тоже сначала так прочитал :)

 0 [Ответить](#)

 **KvanTTT**  
02.03.2019 в 04:58

Чтобы не использовать ярлыки для команд Git, можно просто использовать GUI. Команды вводить не придется, да и работать так более наглядно.

 +3 [Ответить](#)

 **immaculate**  
02.03.2019 в 08:52

Я не понимаю, зачем нужен GUI для Git. Когда-то поставил GitKraken, помучился пару дней, и снес.

Git — из категории настолько мощных инструментов, для которых GUI скорее помеха, чем помощь. И наделать проблем проще скорее из GUI, чем из командной строки.

По опыту, который есть у меня, GUI пользуются только те, кто сумел освоить лишь `pull/commit/push` и ничего более (<https://xkcd.com/1597/>). И в случае любой реальной проблемы (даже проблемки), эти люди беспомощны. И их все время приходится водить за руку объясняя даже такие элементарные вещи, как правильно называть ветки.

Как только человек учится пользоваться git в полной мере, он уже переходит в командную строку. Есть множество ситуаций в жизни, которые никак не могут быть предусмотрены заранее создателями любого GUI. Во многих GUI инструментах даже `rebase` сделать сложно (требуются минуты времени и десятки кликов мыши), тогда как в командной строке достаточно написать `git rebase -i HEAD~3`.

 -3 [Ответить](#)

 **gturk**  
02.03.2019 в 16:32

В гуге гораздо удобнее перед комитом ещё раз критическим взглядом посмотреть на то что ты там наколбасил и точно отменить непонравившиеся куски.

Да и вообще для каждодневной работы, лично мне, удобнее гуй — набивать руками имена файлов и веток сомнительное удовольствие. Ну а если нужно хотябы заребейзиться или что-то более экзотичное, то да, там уже в консоль.

Плюс гуй всётаки сглаживает кривую обучения

 +2 [Ответить](#)

 **Danik-ik**  
02.03.2019 в 18:23

Пользуюсь Смартгитом, плачу деньги (так как для работы). За пару лет единственное, что не смог в нём сделать — какую-то экзотику вроде мержа поддереьев без общего корня. Всегда вижу, какая команда выполняется в результате моего действия (помогает понимать и автоматизировать). Классные дифф, редактор индекса (трёхоконный), конфликт резолвер. Прекрасное, наглядное представление всего, что надо. Они достигли уровня, когда инструмент намного больше помогает, чем мешает.

А Кракена тоже снёс в тот же день, как поставил — кажется, он посмел сыпаться с ошибками. Хотя картинка у него красивая, да...

Кажется, важно не только gui или cli, но и то, что это конкретно за gui — то ли для тех, кто ничего не хочет видеть лишнего, то ли для тех, кто хочет повысить свою эффективность, не теряя при этом контроля за процессом. И для самостоятельного обучения, к примеру, наглядный результат — вообще очень полезная вещь.

0 Ответить



KvanTTT

03.03.2019 в 07:22

Ну а если нужно хотябы заребейзиться

Это как раз тоже удобней делать через GUI. Ну во всяком случае GitExtensions поддерживает.

0 Ответить



SVlad

02.03.2019 в 23:01

Собственный git gui самого гита вполне удобен для повседневной работы. Он наглядно показывает изменения в рабочей области, в staging и дерево коммитов. Ну и некоторые частые команды там проще, чем в консоли. Например, есть отдельная команда revert для файлов, и для строк в файлах.

0 Ответить



GLeBaTi

04.03.2019 в 17:22

Для меня самым удобным и ёмким оказался GitExtensions. Не приходится в очередной раз гуглить команды для git

+1 Ответить



ivan\_suroegin

02.03.2019 в 19:12

А зачем пробелы перед закрывающимися кавычками в ярлыках для команд Git? При наборе, например,

ga .

перед точкой будет два пробела, это пофиг, согласен, отработает как надо, но это немного странно, мне кажется.

0 Ответить



 **WebMonet**  
04.03.2019 в 15:06

ИМХО лучшая шпаргалка по GIT ever!

0 Ответить



 **Magikan**  
05.03.2019 в 07:55

стесняюсь спросить, а progit разве отменили?

p.s: stackoverflow уже года как 2 назад опубликовал инфу о том, что вопрос «как выйти из vim» набрал миллион просмотров. )))

0 Ответить



 **grigorym**  
10.02.2020 в 22:16

«git checkout my\_commit — отмена изменений my\_commit»... эээ, что?

0 Ответить



 **GLeBaTi**  
11.02.2020 в 14:30

наверное имелось ввиду с флагом --force. Тогда да, локальные изменения откатятся.

0 Ответить



Только полноправные пользователи могут оставлять комментарии. [Войдите](#), пожалуйста.

#### ПОХОЖИЕ ПУБЛИКАЦИИ

25 декабря 2020 в 17:17

**Git, я хочу все отменить! Команды исправления допущенных ошибок**

 +20 28K 301 20 +20

31 августа 2018 в 16:08

**Пятничные вебинары от Skillbox: разработка для начинающих** +8 2.7K 25 0

28 августа 2018 в 18:29

**Выбор языка программирования: 3 совета от программиста из Apple** +4 35K 118 36 +36

## ЛУЧШИЕ ПУБЛИКАЦИИ ЗА СУТКИ

вчера в 11:16

**Bash отладчик с поддержкой произвольных точек останова** +44 4K 74 5 +5

вчера в 14:40

**Новые нули дзета-функции** +30 3.9K 14 10 +10

вчера в 21:20

**Личный опыт выгорания** +25 7.6K 29 22 +22

вчера в 17:00

**DIY квантовые вычисления: как я начал собирать квантовые схемы** +21 8K 41 22 +22

вчера в 21:00

**Антиматерия и бариогенезис. Три причины, почему нет антивещества, но есть мы** +15 2.6K 18 1 +1

## ИНФОРМАЦИЯ

|                  |   |
|------------------|---|
| Дата основания   | 7 июня 2016                                   |
| Местоположение   | Россия  |
| Сайт             | <a href="https://skillbox.ru">skillbox.ru</a> |
| Численность      | 501–1 000 человек                             |
| Дата регистрации | 18 мая 2018                                   |

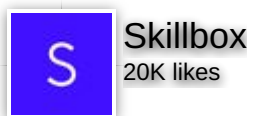
## ССЫЛКИ

---

[Образовательная платформа Skillbox](https://skillbox.ru)  
[skillbox.ru](https://skillbox.ru)

## FACEBOOK


---



## ВКОНТАКТЕ

---



 Skillbox: образовательная пл...

451,754 followers

☐ Follow on VK

## БЛОГ НА ХАБРЕ

30 июня в 23:14

### Устраняем популярные изъяны в коде — чек-лист ошибок junior-разработчиков

 13K  11 +11

9 июня в 16:59

### Разбираемся с Qwik — новым витком эволюции фреймворков

#### Ваш аккаунт

Войти

Регистрация

#### Разделы

Публикации

Новости

Хабы

Компании

Авторы

Песочница

#### Информация

Устройство сайта

Для авторов

Для компаний

Документы

Соглашение

Конфиденциальность

#### Услуги

Корпоративный блог

Медийная реклама

Нативные проекты

Образовательные

программы

Мегапроекты



Настройка языка

[Техническая поддержка](#)

[Вернуться на старую версию](#)

© 2006–2022, Habr