

- Общие ключи
 - `-q/--quite` - вывод только ошибок и предупреждений, игнорируя остальную информацию
 - `-v/--verbose` - вывод более детальной информации
 - `-z/--null` - заканчивать вывод информации NUL-charом вместо переноса строки
 - `-p/--patch` - открывает интерфейс патча
 - `-f/--force` - игнорирование конфликтов, несмотря на возможную потерю данных
 - `--progress` - выводить прогресса в stderr
 - `--dry-run` - имитирует выполнение команды и выводит логи
- `config` - конфигурация гита (имя пользователя, текстовый редактор) - `--replace-all` (замена всех переменных, подходящих под данную регулярку) - `--add` (добавляет строку настройки без изменения уже существующих настроек) - `--get` / `--get-all` (получение значения / всех значений по ключу) - `--unset/--unset-all` (удалить переменную / все переменные, подходящие под данную регулярку) - `--rename-section/--remove-section` (переименование/удаление секции) - `-l/--list` (список всех настроек в конфиг-файле) - `-e/--edit` (открывает редактор для редактирования конфиг-файла) - `--global` (установка настроек для пользователя, файл `~/.gitconfig`) - `--system` (установка настроек системы, файл `/etc/.gitconfig`) - `--local` (установка настроек в локальном репозитории)
- `init` - создание пустого git-репозитория
 - `--bare` - создание bare-репозитория
 - `--template=<template_directory>` - директория с которой будут скопированы файлы перед клонированием
 - `--separate-git-dir=<git_dir>` - указание пути для папки `.git/`. По-умолчанию создается в корне репозитория.
 - `clone <url> <dir>` - клонирование репозитория в новую папку
 - `-l/--local` - клонирование локального репозитория
 - `--reference <repository>` - использовать ссылки на другой локальный репозиторий вместо клонирования
 - `-n/--no-checkout` - не выполнять команду checkout по завершению клонирования
 - `--bare` - создание bare-репозитория
 - `-o/--origin <name>` - указание имя для < upstream >. По-умолчанию - это origin
 - `-b/--branch` - указать ветку на которую будет ссылаться HEAD. По-умолчанию делает такой же, как и в клонируемом репозитории
 - `--recurse-submodules` - рекурсивная инициализация и клонирование сабмодулей
 - `--separate-git-dir=<git_dir>` - указание пути для папки `.git/`. По-умолчанию создается в корне репозитория.
 - `--dissociate` (позволяет отделить репозиторий, клонированный через `--reference`)
- `add <filename>` - добавление файлов в индекс
 - `-i/--interactive` (смотреть пикчу(!); интерактивно добавить модифицированный контент в рабочее дерево к индексу)
 - `--edit` - открывает diff с индексом в редакторе и дает пользователю отредактировать его
 - `--update` - обновляет сущности, которые уже в индексе или удаляет их, но не добавляет новые
 - `-A/--all/--no-ignore-removal` - добавить изменение всех индексируемых и не индексируемых файлов
 - `--no-all/--ignore-removal` - добавить изменение всех индексируемых, но игнорировать удалённые из индекса файлы

- `--refresh` - не добавляет файлы, только обновляет информацию в индексе
- `--ignore-errors` - игнорировать ошибки при добавлении и продолжать добавлять
- `--renormalize` - удалить и добавить все файлы индекс
- `status` - показывает статус рабочего дерева и состояния файлов
 - `-s/--short` - вывод в коротком формате
 - `-b/--branch` - отобразить информацию о ветке игнорирую `-s`
 - `--long` - вывод всех информации. По-умолчанию
 - `--ignore-submodules[=<when>]` - игнорировать изменения в сабмодулях
 - `--[no-]renames` - использует/не использует трекинг переименований
- `diff <objname> <objname>` - показывает разницу между коммитами/ветками
 - `-s/--no-patch` - не выводить патч
 - `--raw` - вывод diff в формате plain text
 - `--minimal` - более долгий анализ для генерации минимально возможного diff
 - `--stat` - показывает список файлов с изменениями
 - `--summary` - вывод расширенной информации о файле вроде создания, переименования или изменения прав на доступ
 - `--name-only` - отображать только название измененных файлов
 - `--name-status` - отображать только имя и статус измененных файлов
 - `--no-renames` - отключить поиск переименованных файлов
 - `--check` - предупреждает, если существуют ошибки с пробелами (например строки исключительно из пробелов)
 - `--cached/--staged` - вывод diff между коммитом и заиндексированным состоянием (git add)
 - `--no-index` - позволяет сравнить неиндексируемые файлы
- `commit` (создание коммита т.е. фиксация некоего состояние репозитория)
 - `-a/--all` - автоматически застейджить и закоммитить все файлы, которые были изменены или удалены
 - `-c/--reedit-message=<commit>` - изменить сообщение коммита
 - `--reset-author` - при изменении коммита меняется автор на того, кто менял коммит
 - `--author=<author>` (указать автора)
 - `--date=<date>` (указать дату коммита)
 - `-m < msg > /--message < msg >` (указание сообщение коммита)
 - `-F < file > /--file=< file >` (взять сообщение коммита из файла)
 - `--allow-empty` (возможность создания пустого коммита)
 - `--allow-empty-message` (разрешает коммиты без сообщений)
 - `-e/--edit` (позволяет изменить сообщение, которое получено с `-F/-m/-C`)
 - `--no-edit` (не открывать редактор сообщения коммита)
 - `--amend` (отмена предыдущего коммита, создание нового на основании сделанного и новых изменений)
 - `-o/--only` (коммитить только указанный файл) - `--[no-]status` - выводит/не выводит статус как в git status
- `reset <hash>` - перемещение head, индекса и рабочего дерева до указанного коммита
 - `--soft` - сбросить только HEAD
 - `--mixed` - сбросить HEAD, индекс не переносится (но отображается, как unstaged)
 - `--hard` - сбросить HEAD и индекс и рабочее дерево
 - `--keep` - сбросить HEAD, но сохранить изменения локально

- **rm** <filename> - удаление файла из рабочего дерева и индекса
 - **-r** - разрешает рекурсивное удаление, если указано название папки
 - **--cached** - удалить только из индекса
- **rebase** <branchname> - перемещение коммитов из ветки на последний коммит другой ветки
 - **--autostash** - сохраняет состояние рабочего дерева перед rebase
 - **--fork-point** - ищет наилучшего предка для веток
 - **-r/--rebase-merges** - попытка пересоздать слияния после rebase, чтобы сохранить структуру дерева
 - **-p/--preserve-merges** - попытка сохранить структуру дерева со слияниями
 - **--no-ff, -f, --force-rebase** - перемещение коммитов по одному, сохраняя историю коммитов
 - **--allow-empty-message** - позволяет пустые сообщения
 - **--[no-]stat** - показывать/ не показывать diff
 - **--verify** - разрешает прекоммитные хуки
 - **-c** - продолжить после конфликта
 - Действия: **--continue, abort, skip, quit**
- **branch** (выполнение различных операций с ветками)
 - **-a/--all** - выводит список всех локальных и удалённых-индексируемых веток
 - **-r** - remote ветки
 - **-d/--delete** удалить смёрженную ветку
 - **-D** - удалить ветку, даже если не было merge
 - **-m/--move** - перемещает/переименовывает ветку
 - **-i** - игнорирует регистр
 - **-u [remote branch]** - связывает текущую ветку с удаленной
 - **--list** - вывести список названий веток
 - **--create-reflog** - создаёт историю перемещения HEAD для ветки
- **checkout** - переключение между ветками
 - **-b < new_branch >** - создание новой ветки и переход в неё
 - **-t/--track < remote_branch >** - создаст новую ветку связанную с удаленной (с таким же именем)
 - **-l** - создание истории изменения HEAD (reflog)
 - **--merge** - переходит в ветку и сливает изменения с ней
- **merge** <branchname> - слияние веток (историй разработки)
 - **--[no-]commit** - установка флага, нужно ли коммитить изменения после вынужденного мерджа при пуле.
 - **-e/--[no-]edit** - открыть редактор до комита после мерджа
 - **--[no-]ff** - если можно резолвить конфликт fast-forward'ом, то обновляет указатель на ветку не создавая мердж-коммита
 - **--ff-only** - создавать мердж-комит, даже если можно резолвить fast-forward'ом
 - **-n/--[no-]stat** - выводить diff-статистику в конце мерджа. **-n = --no-stat**
 - **--[no-]squash** - сквошить комиты один вместо мерджа
 - **--[no-]autostash** - делает stash перед выполнением merge
 - **-m <msg>** - сообщение для комита, что будет создан при мердже
 - **--abort, --continue** - прервать/ продолжить текущий резолвинг конфликта
- **stash** - сохранение изменений локально без комита
 - **list** - список stash'ей
 - **apply** - применить изменения из stash

- **drop** - удалить stash
- **show** [**<stash>**] - показывает diff между stash и последним коммитом
- **branch** **<branchname>** [**<stash>**] - создает ветку **branchname** и применяет туда изменения из **stash**
- **clear** - удаление все stash'ы
- **tag** (создание, удаление и проверка тега объекта)
 - **-a/--annotate** - создаёт аннотированный объект без подписи
 - **-d/--delete** - удаление тегов с указанными именами
 - **-l/--list** **<pattern>** - вывод тегов по шаблону
 - **--sort=<key>** - сортировка на основе заданного ключа
 - **--points-at** **<object>** - выбрать теги, которые ссылаются на данный объект
 - **-m** **<msg>/--message=<msg>** - сообщение для тега
 - **-e/--edit** - открыть редактор перед созданием тега для редактирования сообщения
- **fetch** - скачивание объектов и ссылок из другого репозитория
 - **--all** - получить все данные с удалённого репозитория
 - **-a/--append** - добавление **ref names and object names** в **.git/FETCH_HEAD**. Без этого **атрибута** данные в этом файле будут перезаписаны
 - **--depth=<depth>** - ограничение на кол-во комитов, которые будут скачены с каждой ветки
 - **-p/--prune** - перед fetch удаляет ссылки на несуществующие объекты на удаленном репозитории
 - **-t/--tags** - fetch всех тегов из удаленного репозитория
 - **--recurse-submodules** **[=yes|on-demand|no]** - выполнять ли fetch для всех сабмодулей
 - **-j/--jobs=<n>** - максимально число fetch'ей, которые могут выполняться одновременно
- **pull** **<remote>** - выполнение fetch и интеграция с другим репозиторием или локальной веткой
 - **--recurse-submodules** **[=yes|on-demand|no]** - выполнять ли fetch для всех сабмодулей
 - **-r/--[no-]rebase** **[=false|true|merges|preserve|interactive]** - после fetch выполняется rebase на последний комит с upstream'a.
 - Опции, связанные с **merge** (**--ff**, **squash**, **edit**, **autostash**)
 - Опции, связанные с **fetch** (**--all**, **append**, **--no-tags**)
- **push** **[remote]** **[branch]** - отправка объектов на удалённый репозиторий
 - **--all** - пуш всех веток
 - **--prune** - удаляет удалённые ветки, у которых нет локальных копий
 - **-d/--delete** - удаляет все перечисленные ссылки из удаленного репозитория
 - **--tags** - отправляет перечисленные теги
 - **-u/--set-upstream** - добавление upstream'a для каждой ветки
 - **--recurse-submodules** **=check|on-demand|only|no** - проверка, обновлены ли сабмодули до последнего комита. Если нет - abortит выполнение
 - **--[no-]verify** - включает/отключает **pre-push hook**
- **remote** - управление удалёнными репозиториями
 - **--list** - отображает все связанные удаленные репозитории
 - **--add** **[name]** **[url]** - позволяет создать имя для репозитория, чтобы было удобнее к нему обращаться
 - **--rename** - изменяет имя репозитория
 - **--rm** - удалить ссылку на репозиторий
 - **--show** - показать информацию о репозитории
- **show** **<objhash>** (отображает различные типы объектов или внутренности коммита)

- `--format=<format>` - выбор формата вывода (oneline, short, medium...)
- `--encoding=<encoding>` - использовать кодировку
- **log** - вывод логов коммитов
 - `--pretty` - форматирование
 - `-p` - показывает diff
 - `--stat` - показывает статистику по каждому коммиту
 - `--graph` - вывод дерева коммитов в виде графа
- **fsck** - проверяет объекты в базе данных
 - `--unreachable` - вывод всех объектов, которые существуют, но не доступны ни с одного узла(reference nodes)
 - `--[no-]dangling`: вывод всех объектов, которые существуют, но никогда не используются напрямую
- **gc** - удаление ненужных файлов и оптимизация репозитория
 - `--aggressive` - агрессивная(более детальная) оптимизация по памяти
 - `--auto` - проверяет, нужно ли выполнять сборку. Если нет - завершает выполнение команды
 - `--[no-]prune=<date>` - удаляет свободные объекты старше даты (по дефолту 2 недели) / не удалять объекты
- **instaweb** (встроенная утилита, для просмотра состояния репозитория в браузере)
 - Возможные опции: `--local`, `--port`, `--httpd`(команда для запуска), `--browser`
 - Действия для сервера: `--stop/start/restart`
- **archive** - создание архива файлов из указанного дерева
 - `--format=<fmt>` - формат архива, tar по-умолчанию
 - `-l/--list` - все доступные типы архива
 - `-o <file>/--output=<file>` - запись архива в указанный файл вместо stdout
 - `--remote=<repo>` - создание архива с удаленного сервера, а не локального дерева
- **prune** - удаляет все недостижимые объекты с базы данных объектов
 - `--expire <time>` - удалять только те файлы, которые старше чем указанное время
- **cat-file <objhash>** (получение содержимого объекта по хешу)
 - `-t` - вместо содержимого вывести тип объекта
 - `-s` - вместо содержимого вывести размер объекта
 - `-p` - вывод содержимого `<object>` в удобном виде в зависимости от его типа.
 - `-e` - выход с нулевым статусом, если `<object>` существует и является допустимым объектом, иначе выход с ненулевым значением и запуск ошибки в stderr.
- **ls-tree <treedir>** - список элементов дерева объектов
 - `-d` - показывать только деревья
 - `-r` - рекурсивно выводить поддеревья
 - `-t` - выводить также информацию о дереве при `-r`
 - `-l/--long` - выводит размер объекта
 - `--name-only/--name-status` - выводить только названия файлов
 - `--full-name` - выводить полный путь до файла
- **grep <regex>** (вывод строк, подходящих под указанный паттерн)
 - `--cached` - поиск blob'ов, зарегистрированных в индексном файле
 - `--no-index` - поиск файлов в текущем каталоге, который не управляется Git
 - `--exclude-standard` - игнорирование файлов, описанных в .gitignore