

## Создать

Клонировать существующий репозиторий

```
$ git clone ssh://user@domain.com/repo.git
```

Создать новый локальный репозиторий

```
$ git init
```

## Локальные изменения

Измененные файлы в вашем рабочем каталоге

```
$ git status
```

Изменения в отслеживаемых файлах

```
$ git diff
```

Добавление всех текущих изменений в следующий коммит (обратите внимание на то, что указана точка)

```
$ git add .
```

Добавление изменений в <файле> в следующий коммит

```
$ git add -p <файл>
```

Коммит всех локальных изменений в отслеживаемых файлах

```
$ git commit -a
```

Коммит ранее подготовленных изменений

```
$ git commit
```

Изменить последний коммит

*Не исправляйте опубликованные коммиты!*

```
$ git commit —amend
```

## История коммитов

Показать все коммиты, начиная с самого нового

```
$ git log
```

Показать когда в конкретном файле делались изменения

```
$ git log -p <файл>
```

Кто, когда и что изменял в <файле>

```
$ git blame <файл>
```

## Ветвление и теги

Список всех существующих ветвей

```
$ git branch
```

Переключение на ветвь HEAD

```
$ git checkout <ветвь>
```

Создание новой ветви на основе вашей текущей ветви HEAD

```
$ git branch <новая-ветвь>
```

Создание новой ветви на основе ветви, находящейся на удаленном репозитории

```
$ git checkout --track <репозиторий/ветвь>
```

Удаление локальной ветви

```
$ git branch -d <ветвь>
```

Добавление тега в текущий коммит

```
$ git tag <название тега>
```

## Обновление и публикация

Список все сконфигурированных удаленных репозиториев

```
$ git remote -v
```

Информация об удаленном репозитории

```
$ git remote show <репозиторий>
```

Добавить новый удаленный репозиторий с именем <имя>

```
$ git remote add <имя> <url>
```

Скачать все изменения из <репозитория>, но не интегрировать их в HEAD

```
$ git fetch <remote>
```

Скачать изменения и непосредственно слить / интегрировать их в HEAD

```
$ git pull <репозиторий> <ветвь>
```

Публикация локальных изменений в удаленном репозитории

```
$ git push <репозиторий> <ветвь>
```

Удаление ветви на удаленном репозитории

```
$ git branch -dr <репозиторий/ветвь>
```

Публикация ваших тегов

```
$ git push --tags
```

## Слияние и перебазируание

Слияние <ветви> с вашей текущей ветвью HEAD

```
$ git merge <ветвь>
```

Перебазирование текущей ветви HEAD на <ветвь>  
*Не перебазируйте опубликованные коммиты!*

```
$ git rebase <ветвь>
```

Отменить перебазирование

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

Продолжить перебазирование после разрешения конфликтов

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

Используйте следующую команду для разрешения конфликтов

```
$ git mergetool
```

Используйте редактор для ручного разрешения конфликтов и (после разрешения) пометьте файл как файл, у которого конфликты разрешены

```
$ git add <файл с разрешенными конфликтами>
```

```
$ git rm <файл с разрешенными конфликтами>
```

## Отмена действий

Отмена в рабочем каталоге все локальные изменения

```
$ git reset --hard HEAD
```

Отмена локальных изменений в конкретном файле

```
$ git checkout HEAD <файл>
```

Вернуть коммит (создать новый коммит с обратными изменениями)

```
$ git revert <коммит>
```

Переустановить указатель HEAD на предыдущий коммит

... и затем отменить все сделанные изменения

```
$ git reset --hard <коммит>
```

... и сохранить все изменения как не подготовленные к коммиту

```
$ git reset <коммит>
```

... и сохранить незакомитченные локальные изменения

```
$ git reset --keep <коммит>
```