

Git

Установка:

```
sudo apt-get install git
```

Глобальные настройки:

```
git config --global user.name "FooBar"
```

```
git config --global user.email "foo@bar.ru"
```

или в файле: ~/.gitconfig

Инициализация проекта:

```
mkdir git-project && cd git-project
```

```
git init
```

Добавить файл на коммит кандидат - индексация файлов:

```
git add foo.txt
```

Дефолтный UI редактор:

```
gitk
```

Коммит:

```
git commit
```

или

```
git commit -m "First Commit"
```

В gitk -> Shift + F5 - посмотреть изменения

вносим изменения в файл, и чтобы его опять закоммитить надо сначала сделать

```
git add foo.txt
```

 (добавление файла в индекс)

Статус - есть ли что коммитить, что поменялось

```
git status
```

Откат к предыдущей версии файла

```
git checkout -- foo.txt
```

Ещё про индексацию файлов (добавить все файлы в индексацию):

```
git add *
```

 - все файлы

или

```
git add .
```

 - файлы только в текущем каталоге

Копия репозитория с github:

```
git clone https://foo.bar
```

работаем с файлами...

```
git status
```

```
git add foo.txt
```

```
git commit -m "Comment"
```

Загрузка на github:

```
git push
```

Получить изменения из github

```
git pull
```

Индексировать все файлы можно ещё так:

```
git add -A
```

Отменить все изменения до commit, но уже с add:

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

Отменить добавление к индексации файла до коммита

```
git reset HEAD foo.txt
```

Удалить файлы и каталоги вне индекса:

```
git clean -n - посмотреть, что будет удалено
```

```
git clean -nd
```

```
git clean -fd
```

Команда выше не удаляет файлы которые перечислены в .gitignore.

Просмотреть, а затем удалить ВСЕ непроиндексированные файлы и папки, включая игнорируемы Git-ом:

```
git clean -ndx
```

```
git clean -fdx
```

История репозитория:

```
git log
```

```
git log --pretty=oneline - покажет историю в 1 строку
```

```
git log --pretty=oneline -max-count=3
```

```
git log --pretty=oneline --all
```

```
git log --pretty=oneline --author="krdprog"
```

```
git log --pretty=format:"%h - %s : %ad [%an]" --date=short
```

- короткая запись hash, комментария, даты и автора

```
git log --pretty=format:"%h %ad | %s" --date=short
```

сделал себе алиас `gitl`

Пример ветки дерева, снимшоты:

```
2d67e51 2018-04-04 night edition
```

```
a813a4d 2018-04-04 Two commit
```

Возврат к предыдущему коммиту:

```
git checkout a813a4d
```

где a813a4d - это хеш нужного коммита



таким образом ходим по истории коммитов

Переключение на ветку master:

```
git checkout master
```

Краснодарский программист

Краснодарский программист
info@krdprog.ru

 [krdprog](#)
 [krdprog](#)

Рабочие рецепты и способы решения
возникающих задач + выжимки из
учебных материалов.