

```
git log --graph --oneline --all
```

1. Начало работы

```
git init (.git Cmd+Shift+.)
git clone
git status
touch 1.py
vim 1.py
добавление ненужного файла + .gitignore
git add ./*/file_name
git commit [-m]
git commit -a -m - adds all TRACKED files to the staging area and
commits them in one step.
2 картинки
попытка коммита untracked
попытка коммита modified
git config --list
$ git config --global user.name "Kate"
$ git config --global user.email kate@mail.ru
git log
git diff
rm vs git rm
=====Тест 1=====
```

2. Отмена изменений

```
- Отмена индексации файла (tracked -> untracked)
git reset HEAD file_name
- Отмена изменений файла (опасная команда, удаляет все текущие изменения,
перезаписывая файл)
git checkout --file_name
- Изменение последнего коммита (слишком рано закоммитили или написали
неправильный комментарий)
git commit --amend
- Отмена последнего коммита
(навсегда, без сохранения последних изменений)
git reset --hard HEAD~1 / идентификатор_коммита
(навсегда, с сохранением последних изменений в индексе, которые можно
сразу же коммитить)
git reset --soft HEAD~1 / идентификатор_коммита
(навсегда, с заначкой)
git stash
git reset --hard HEAD~1 / идентификатор_коммита
git stash pop
(просто посмотреть)
git checkout HEAD~1 / идентификатор_коммита
(создание нового коммита, в котором нет тех нежелательных частей,
присутствующих в старом коммите)
git revert HEAD
(undo the last commit, but don't create a revert commit)
git revert -n HEAD
```

3. Работа с удаленным репозиторием

git remote

git fetch – забирает данные в ваш локальный репозиторий, но не сливает их с какими-либо вашими наработками и не модифицирует то, над чем вы работаете в данный момент.

git pull = git fetch + git merge

git push [удал. сервер] [ветка]

!!!Эта команда срабатывает только в случае, если вы клонировали с сервера, на котором у вас есть права на запись, и если никто другой с тех пор не выполнял команду push. Если вы и кто-то ещё одновременно клонируете, затем он выполняет команду push, а затем команду push выполняете вы, то ваш push точно будет отклонён. Вам придётся сначала вытянуть (pull) их изменения и объединить с вашими. Только после этого вам будет позволено выполнить push.

=====Тест 2=====

4. Ветки

<http://git-school.github.io/visualizing-git/>

Ветка в Git'e – это просто легковесный подвижный указатель на один из этих коммитов.

git branch имя_ветки

git checkout имя_ветки

git checkout -b имя_ветки

git branch

история с работой в одной ветке и срочным фиксом бага в мастере
слияния fast forward (без нового коммита) и коммит-слияние (с новым коммитом)

master: git branch b1, git checkout b1

b1: git commit, git commit, git commit..., git checkout master

master: git branch bug_fix, git checkout bug_fix

bug_fix: git commit, git commit, git commit..., git checkout master

master: git merge bug_fix (fast forward (без нового коммита)), git branch -d bug_fix, git checkout b1

b1: git commit, git commit, git commit..., git checkout master

master: git merge b1

git branch -d/-D имя_ветки – удаление ветки