Министерство науки и высшего образования Российской Федерации

Федеральное государственное автономное образовательное учреждение

высшего образования

«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт цифрового развития

Кафедра инфокоммуникаций

**ОТЧЕТ**

**ПО ЛАБОРАТОРНОЙ РАБОТЕ №2**

**дисциплины «****Программирование на python»**

|  |  |  |  |
| --- | --- | --- | --- |
|  | | Выполнил:  Кожуховский Виктор Андреевич  1 курс, группа ИВТ-б-о-22-1,  09.03.01 «Информатика и вычислительная техника», направленность (профиль) «Программное обеспечение средств вычислительной техники и автоматизированных систем  », очная форма обучения  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (подпись) | |
|  | | Руководитель практики:  Воронкин Роман Александрович  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (подпись) | |
|  | |  | |

Отчет защищен с оценкой \_\_\_\_\_\_\_\_\_\_\_ Дата защиты\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Ставрополь, 2023 г.

**Тема:** Исследование возможностей Git для работы с локальными репозиториями.

**Цель работы:** *исследовать базовые возможности системы контроля версий Git для работы с локальными репозиториями.*

# Методика и порядок выполнения работы

1. Создал общедоступный репозиторий на GitHub.

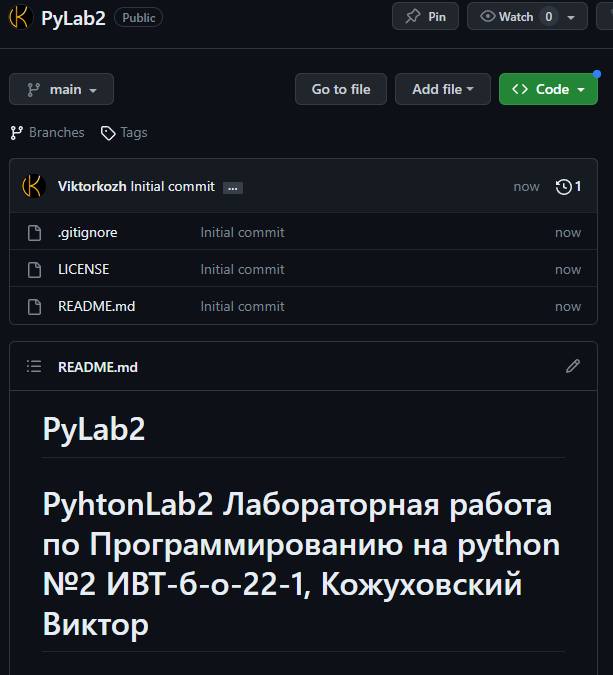


Рисунок 1. Созданный репозиторий

1. Проработайте примеры лабораторной работы. Отразите вывод на консоли при выполнении команд git в отчете для лабораторной работы.

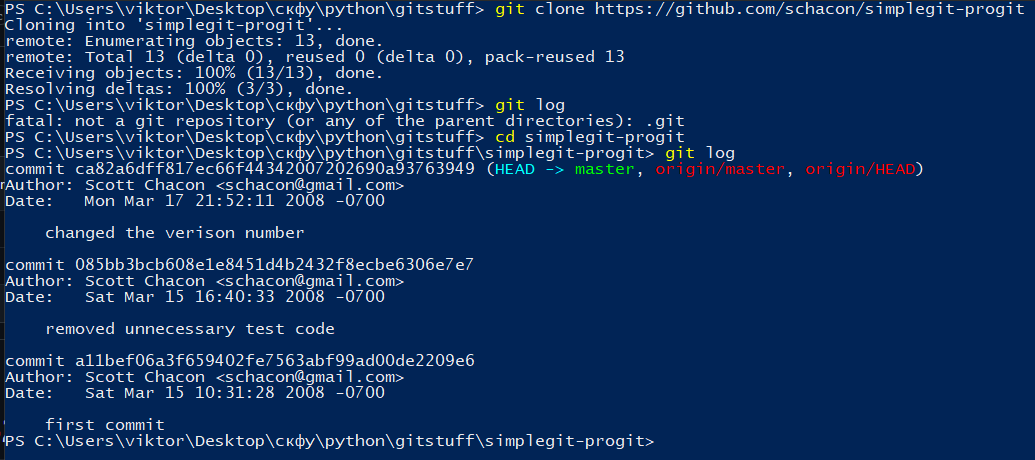


Рисунок 2. Результат выполнения команды git clone

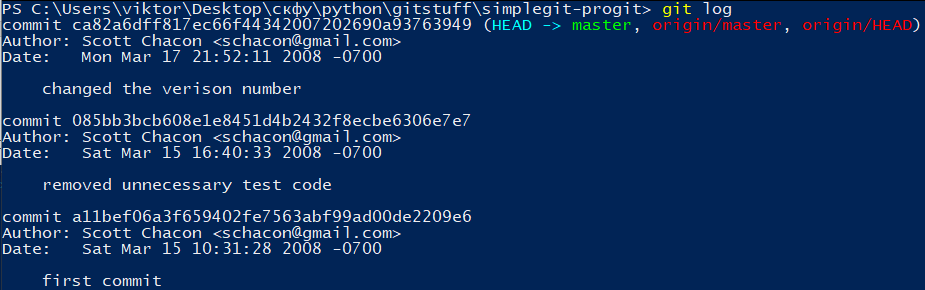


Рисунок 3. Результат выполнения команды git log

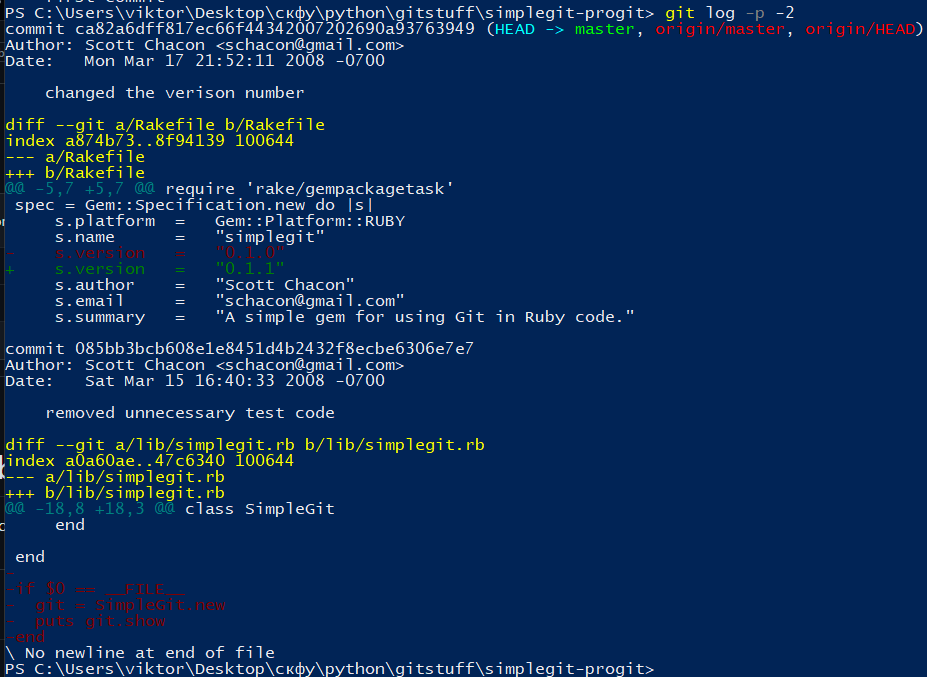


Рисунок 4. Результат выполнения команды git log -p -2

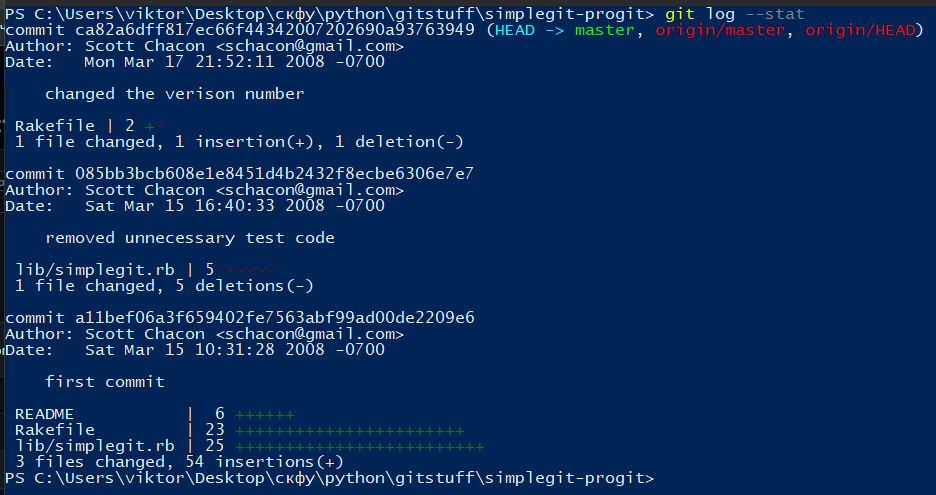


Рисунок 5. Результат выполнения команды git log --stat

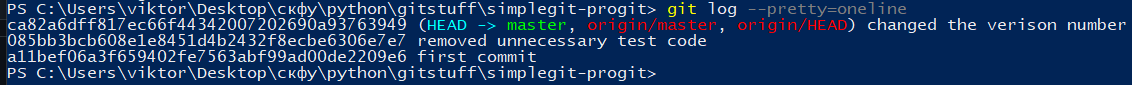


Рисунок 6. Результат выполнения команды git log --pretty=oneline

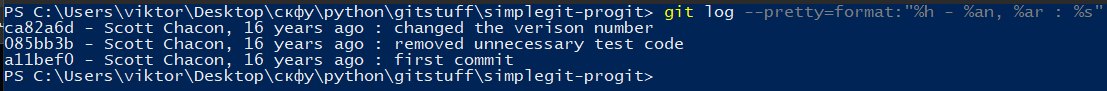


Рисунок 7. Результат выполнения команды git log --pretty=format:"%h - %an, %ar : %s"

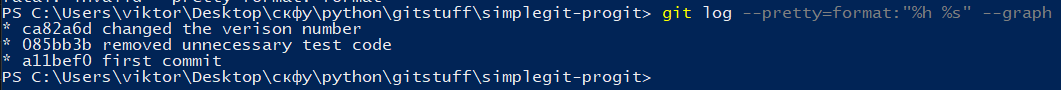


Рисунок 8. Результат выполнения команды git log --pretty=format:"%h %s" –graph

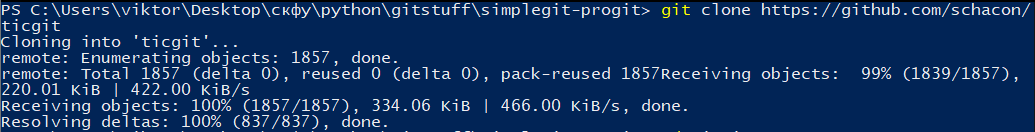


Рисунок 9. Клонирование репозитория ticgit



Рисунок 10. Результат выполнения команды git remote



Рисунок 11. Результат выполнения команды git remote -v



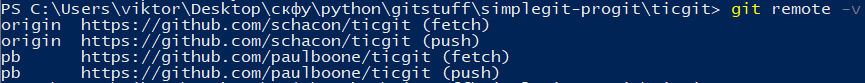


Рисунок 12. Результат выполнения команды

git remote add pb https://github.com/paulboone/ticgit

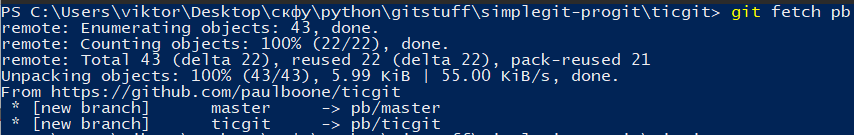


Рисунок 13. Результат выполнения команды git fetch pb

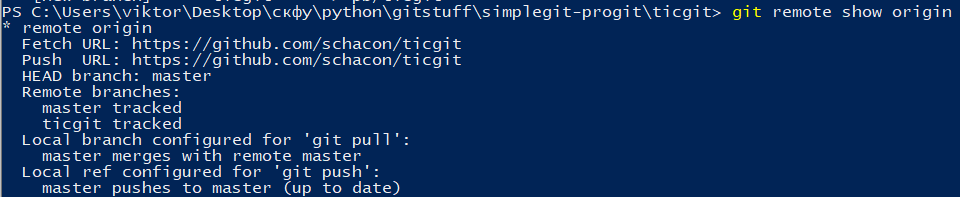


Рисунок 14. Результат выполнения команды git remote show origin



Рисунок 15. Результат выполнения команды git remote rename pb paul



Рисунок 16. Результат выполнения команды git remote remove paul

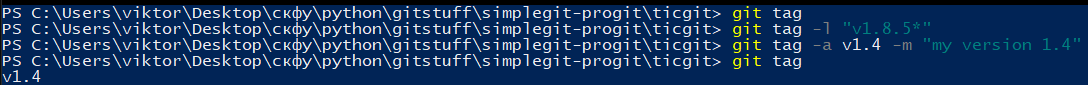


Рисунок 17. Создал аннотированный тег

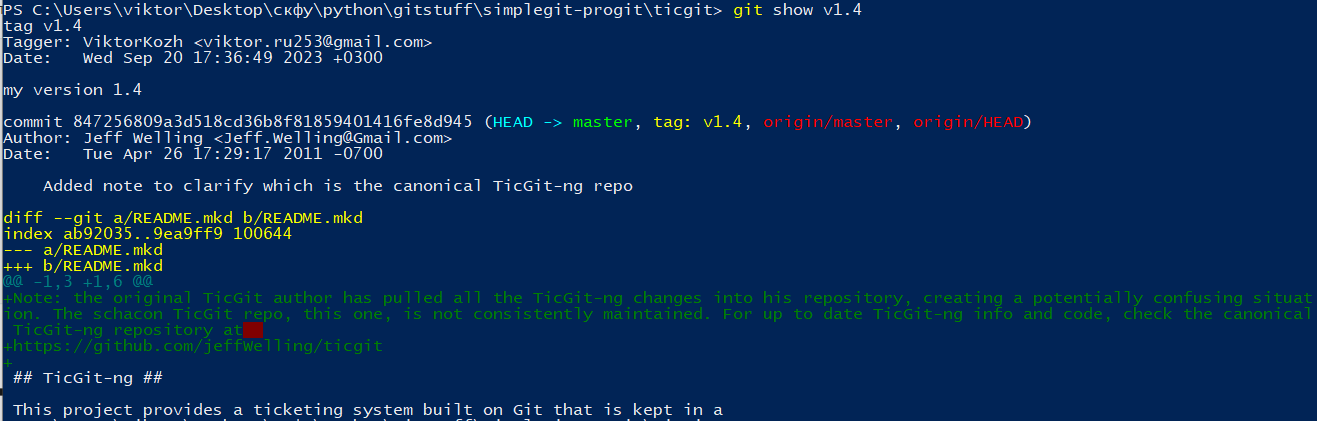


Рисунок 18. Результат выполнения команды git show v1.4



Рисунок 19. Удаление тега

1. Выполнил клонирование созданного репозитория на рабочий компьютер.

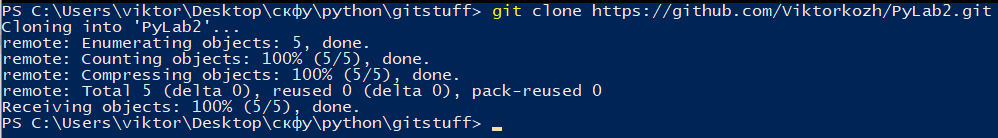


Рисунок 20. Клонирование репозитория

1. Дополнил файл .gitignore необходимыми правилами для выбранного языка программирования и интегрированной среды разработки.



Рисунок 21. Дополненный .gitignore

1. Добавил в файл README.md информацию о дисциплине, группе и ФИО студента, выполняющего лабораторную работу.

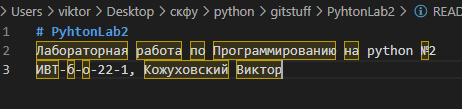


Рисунок 22. Дополненный README.md

1. Просмотрел историю (журнал) хранилища командой git log и git log --graph --pretty=oneline --abbrev-commit.

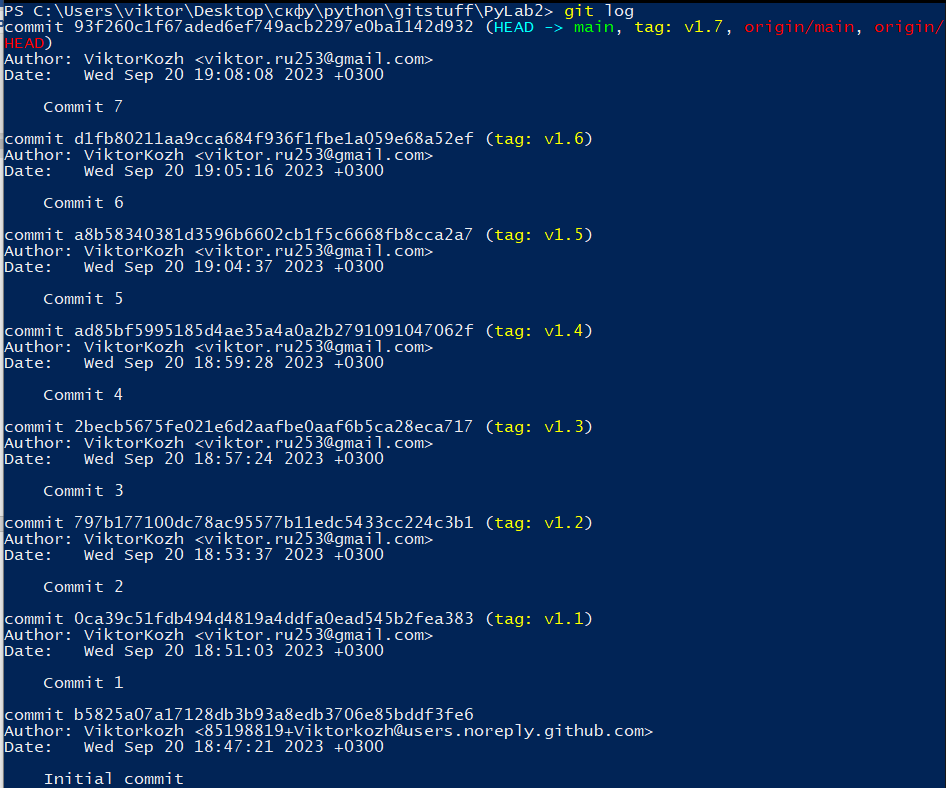


Рисунок 23. Журнал коммитов

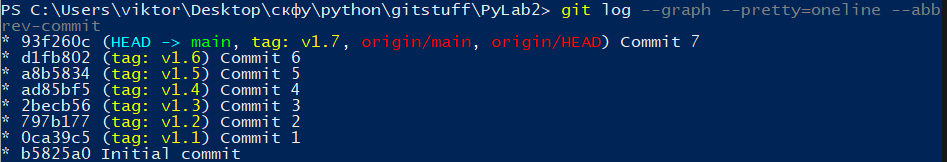


Рисунок 24. Уменьшенный журнал коммитов

1. Просмотрел содержимое коммитов командой git show HEAD и HEAD~1

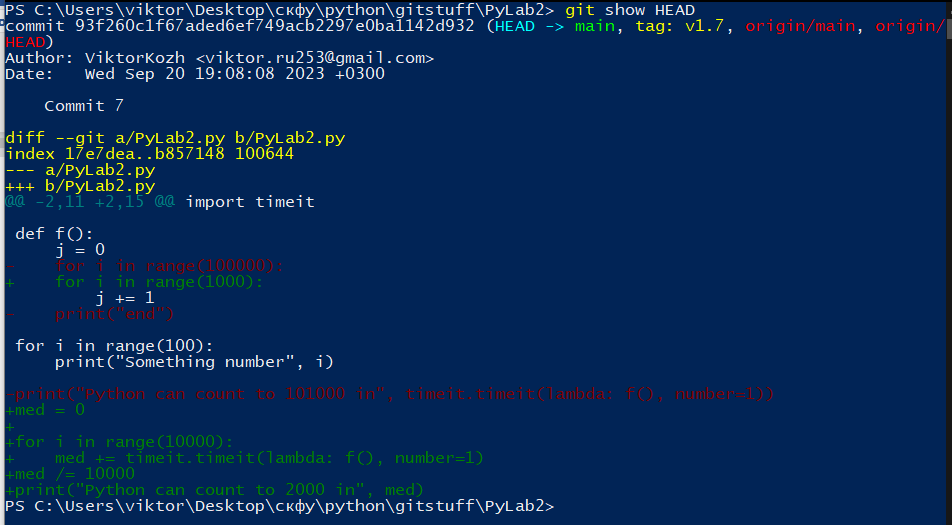


Рисунок 25. Результат выполнения команды git show HEAD

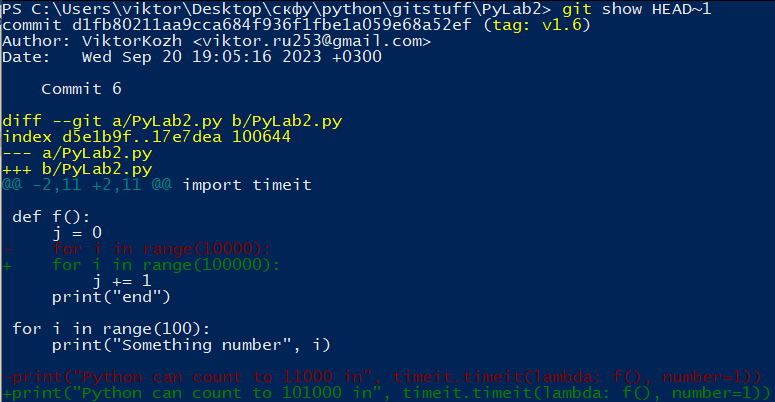


Рисунок 26. Результат выполнения команды git show HEAD ~1

1. Освоил возможность отката к заданной версии.
   1. Удалил весь код из одного из файлов программы репозитория.
   2. Удалил все несохраненные изменения в файле командой: git checkout -- PyLab2.py.
   3. Повторил пункт 8.1 и сделайте коммит.
   4. Откатить состояние хранилища к предыдущей версии командой: git reset --hard HEAD~1 .

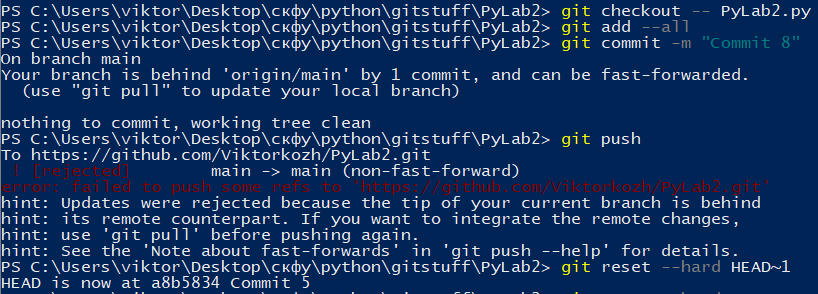


Рисунок 27. Результат выполнения действий из задания 8.

Сделайте выводы об изменении содержимого выбранного Вами файла программы после выполнения пунктов 8.1–8.4. Отразите эти выводы в отчете.

Вывод: GitHub позволяет вернуть удаленный, по ошибке или невнимательности, код и начать с предыдущего коммита.

# Вопросы для защиты работы

1. **Как выполнить историю коммитов в Git? Какие существуют дополнительные опции для просмотра истории киммитов?**

Просмотр истории коммитов осуществляется при помощи команды git log. Эта команда имеет большое количество опций для поиска коммитов по разным критериям.

|  |  |
| --- | --- |
| -p | Показывает патч для каждого коммита. |
| --stat | Показывает статистику измененных файлов для каждого коммита. |
| --shortstat | Отображает только строку с количеством изменений/вставок/удалений для команды --stat. |
| --name-only | Показывает список измененных файлов после информации о коммите. |
| --name-status | Показывает список файлов, которые добавлены/изменены/удалены. |
| --abbrevcommit | Показывает только несколько символов SHA-1 чек-суммы вместо всех 40. |
| --relativedate | Отображает дату в относительном формате (например, «2 weeks ago») вместо стандартного формата даты. |
| --graph | Отображает ASCII граф с ветвлениями и историей слияний. |
| --pretty | Показывает коммиты в альтернативном формате. Возможные варианты опций: oneline, short, full, fuller и format (с помощью последней можно указать свой формат). |
| --oneline | Сокращение для одновременного использования опций --pretty=oneline --abbrev-commit. |

1. **Как ограничить вывод при просмотре истории коммитов?**

Ограничить вывод при просмотре истории коммитов можно при помощи оператора –(желаемое число записей), например -2.

1. **Как** **внести изменения в уже сделанный коммит?**

Внести изменения в уже сделанный коммит можно при помощи команды git commit –amend.

1. **Как отменить индексацию файла в Git?**

Для отмены индексации (unstage) файла в Git, можно использовать команду git reset.

1. **Как отменить изменения в файле?**

Для этого можно использовать команду git checkout (называние файла).

1. **Что такое удаленный репозиторий Git?**

Удалённые репозитории представляют собой версии вашего проекта, сохранённые в интернете или ещё где-то в сети.

1. **Как выполнить просмотр удаленных репозиториев данного локального репозитория?**

Для того, чтобы просмотреть список настроенных удалённых репозиториев, можно запустить команду git remote .

1. **Как добавить удаленный репозиторий для данного локального репозитория?**

Для того, чтобы добавить удалённый репозиторий и присвоить ему имя (shortname), необходимо выполнить команду git remote add <shortname> <url>

1. **Как выполнить отправку/получение изменений с удаленного репозитория?**

Для получения данных из удалённых проектов, следует выполнить git fetch [remote-name]. Для отправки используется git push origin master.

1. **Как выполнить просмотр удаленного репозитория?**

Для просмотра удаленного репозитория используется команда git remote show <remote>.

1. **Каково назначение тэгов Git?**

Тэги используются для фиксации конкретных версий проекта.

1. **Как осуществляется работа с тэгами Git?**

Работа с тэгами Git осуществляется при помощи команды git tag.

1. **Самостоятельно изучите назначение флага --prune в командах git fetch и git push. Каково назначение этого флага?**

Флаг --prune в команде git fetch используется для удаления удаленных веток и ссылок, которые больше не существуют на удаленном репозитории.

В git push он удаляет ветки на удаленном репозитории, которые не существуют в локальном.