**GIT Task 1 .**

Все действия выполняются индивидуально и направлены на получение навыков работы с GIT.

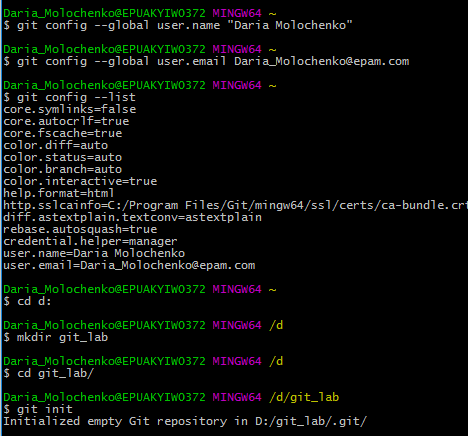
Отчет оформляется в виде скриншотов с комментариями действий при необходимости.

Последовательность выполнения работы:

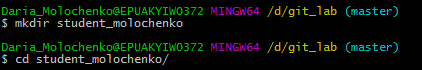
Часть 1. Инициализация репозитория и первый коммит.

1. Создать рабочий каталог.

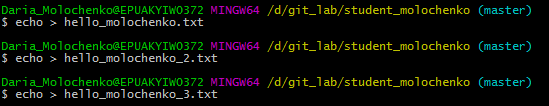
Инициализировать в нем репозиторий. Настроить файл config.



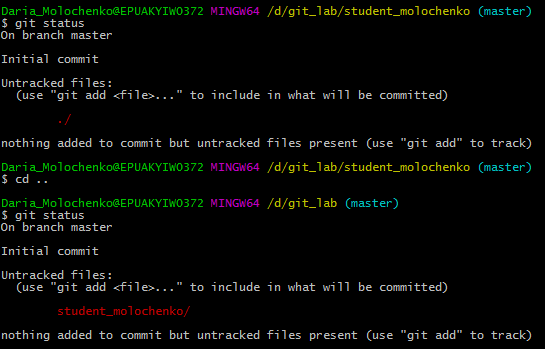
1. Создать в ней рабочий каталог c именем student\_name (где name – Ваша фамилия).



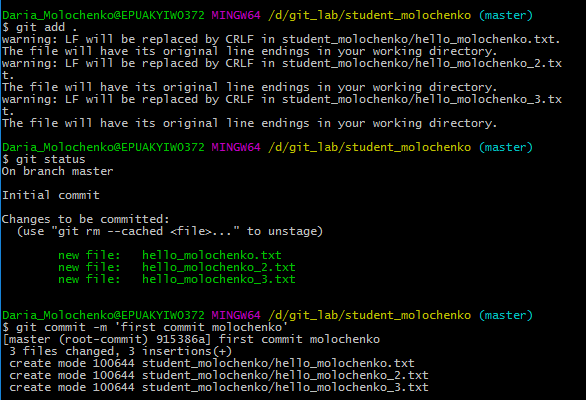
1. Создать в рабочем каталоге 3 файла проекта: (пример hello\_name.rb) (где name – Ваша фамилия).



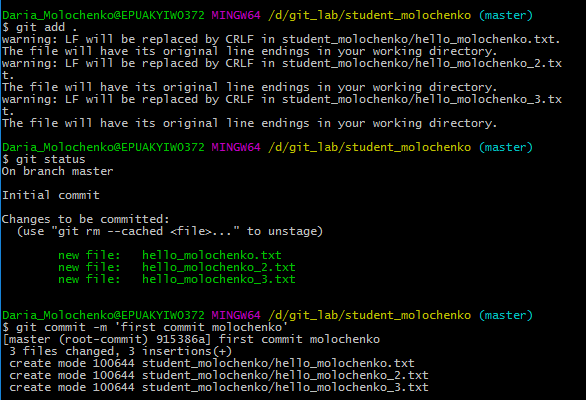
- проверить статус репозитория



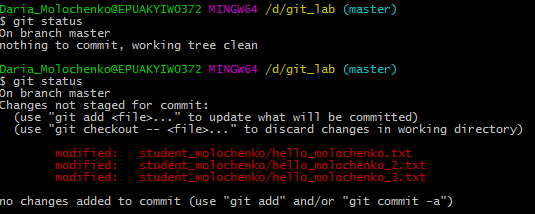
1. Проиндексировать файлы.



1. Сделать первый commit с комментарием ‘’first commit name” (где name – Ваша фамилия)

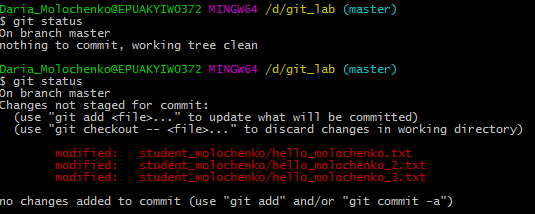


***После коммита git status показывает, что в репозитории нет неотслеживаемых или измененных файлов:***

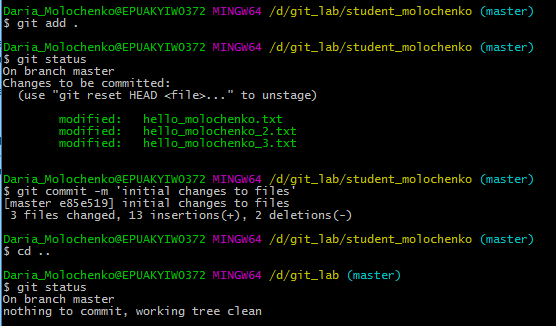


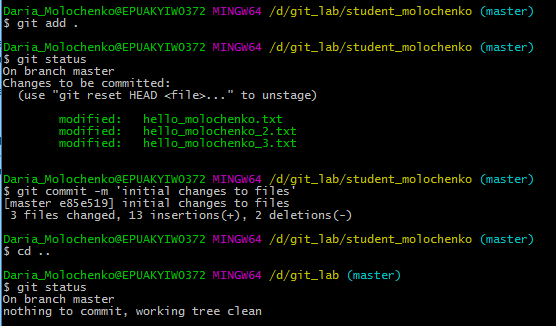
1. Сделать изменения в файлах.

***После внесения изменений в файлы git status показывает, что есть измененные файлы:***



- cделать commit изменений.





Часть 2. Работа с изменениями.

1. Сделать изменения в одном из файлов – файл 1.

Проиндексировать файл

1. Сделать новые изменения в файле1.

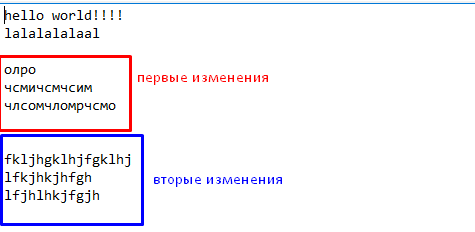
- проверить статус

1. Сделать commit

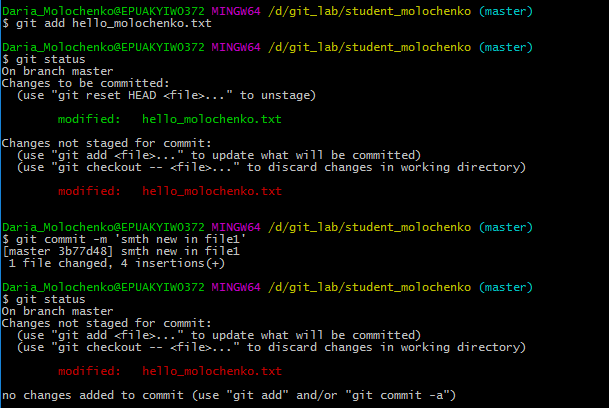
- проверить статус

Объяснить результат.

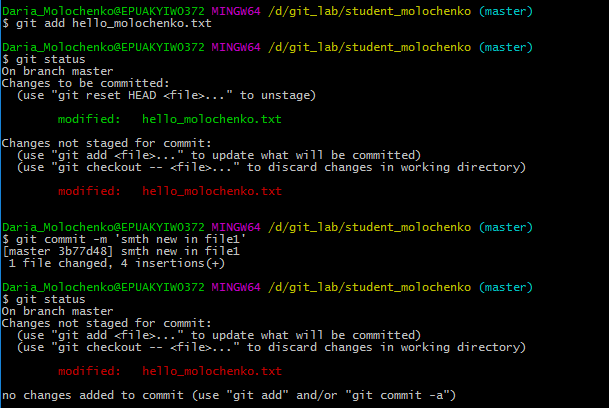
**Наши изменения в файле:**



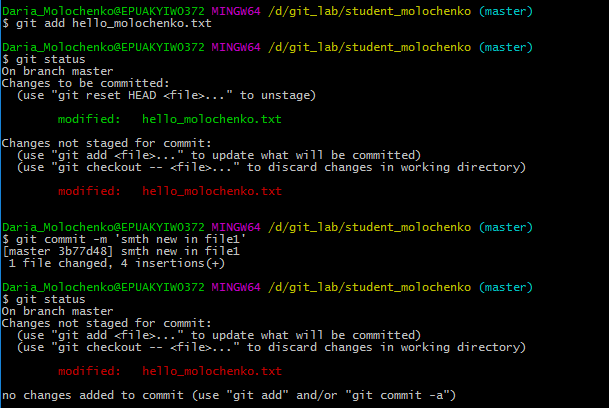
***Исходя из скриншота ниже мы видим, что после внесения и индексирования первых изменений, а также внесения вторых изменений, git status показывает, что есть как изменения, которые проиндексированы и готовы к внесению их в коммит, так и те, которые не будут внесены в коммит (наши вторые изменения).***



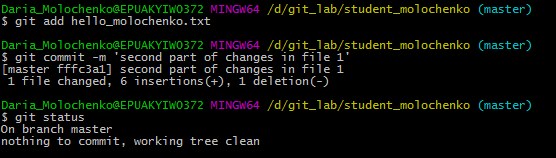
***Результат выполнения команды git commit также показывают, что было внесено только 4 новых строки, т.е. наши первые изменения .***



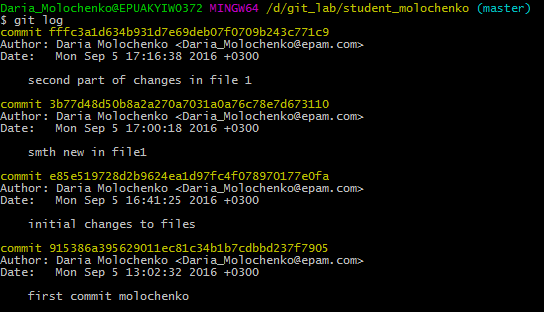
***Выполняя git status снова, видим, что у нас остались непроиндексированные изменения. Чего и стоило ожидать.***



***Проиндексируем наши вторые изменения и закоммитим их:***



1. Вывести историю проекта. $ git log \



Затем

1. Используйте команду $ git log --pretty=oneline , настраивая соответствующие параметры выведите на экран последние commits за 5 минут.



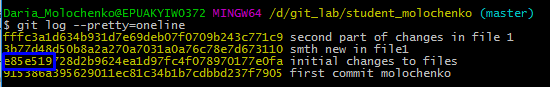
***За 5 минут коммитов не было. А за 30 было:***



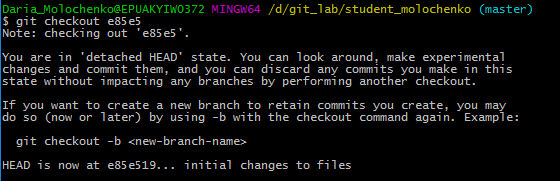
Часть 3. Работа с предыдущими состояниями репозитория.

1. Выполните возврат состояния файлов рабочего каталога на второй commit со времени начала выполнения таска ( ОБЯЗАТЕЛЬНО используйте команду git checkout <hash>)

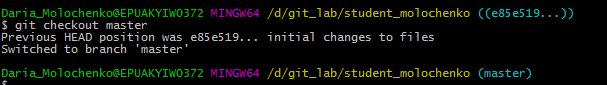
***Смотрим SHA1 нашего второго коммита:***



***Переставляем HEAD:***



1. Произведите возврат состояния файлов к последнему commit .

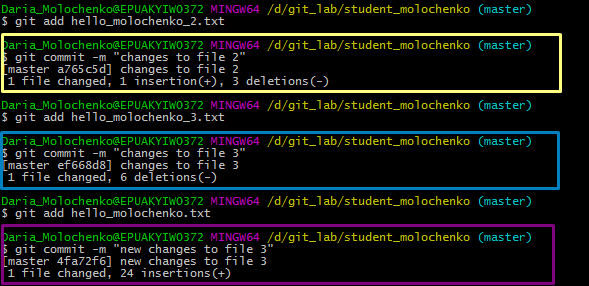


Часть 4. Работа с тэгами.

1. Создайте тэг в текущем snapshot репозитория. Название тэга - V1\_name (где name – Ваша фамилия)/

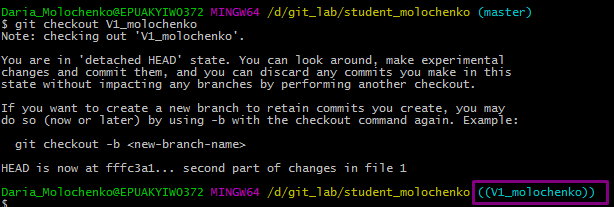


1. Выполните изменения и как минимум два commits в текущем каталоге.

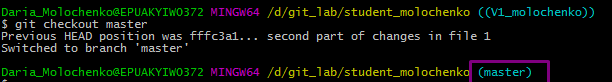


Сделали изменения и закоммитили их.

1. Продемонстрируйте возврат к состоянию репозитория помеченную тегом.

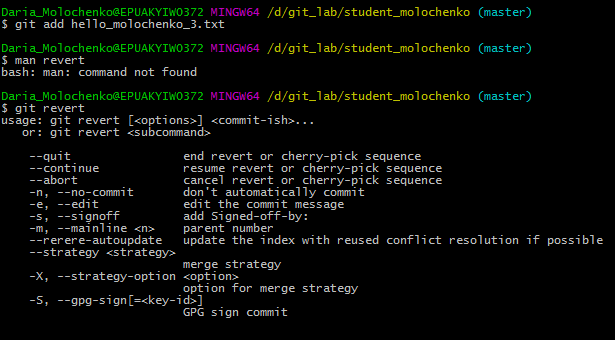


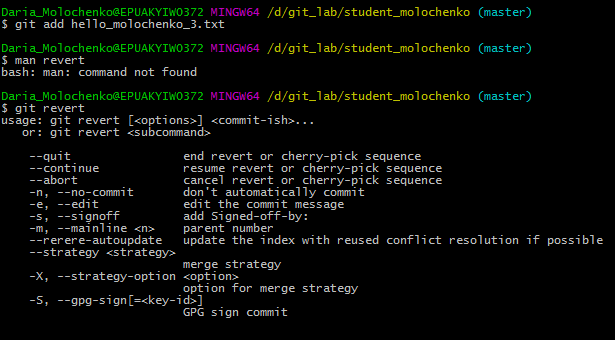
1. Выполните возврат к последнему состоянию master.



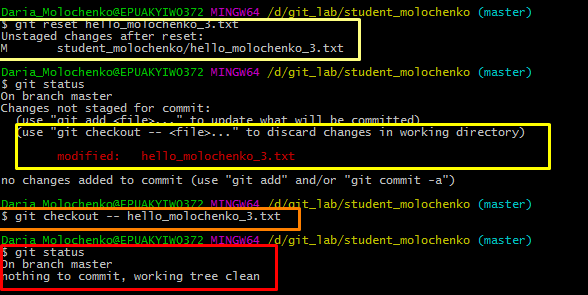
Часть 5. Работа с функцией revert.

1. Выполните изменения текущего состояния рабочего каталога.
2. Проиндексируйте изменения.
3. Командой revert верните исходное состояние файлов.

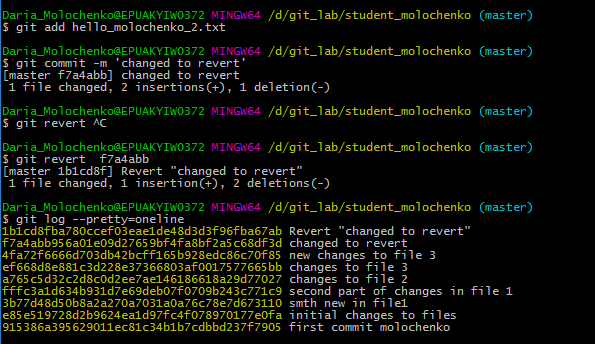




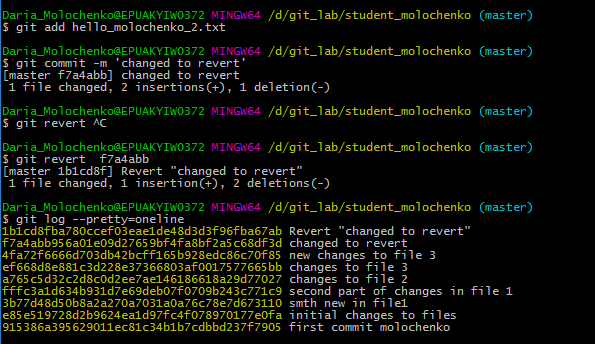
***Git revert позволяет отменять изменения, произведенные в прошлом отдельным коммитом, но не позволяет снять индексацию с определенных файлов. Для этого используется команда git reset. А для того, чтобы отменить изменения в отдельном файле, который не проиндексирован, используется команда git checkout -- <file name>.***



1. Сделайте изменения состояния каталога.
2. Проиндексируйте и выполните commit.



1. Выполните revert последнего commit



1. Продемонстрируйте log.

