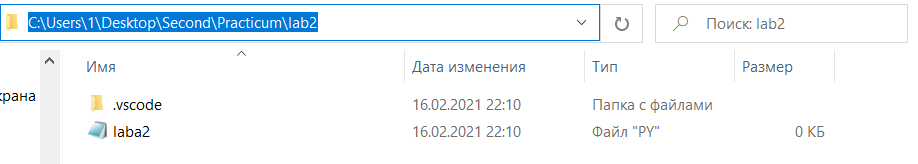
**Работа с Git в терминале**

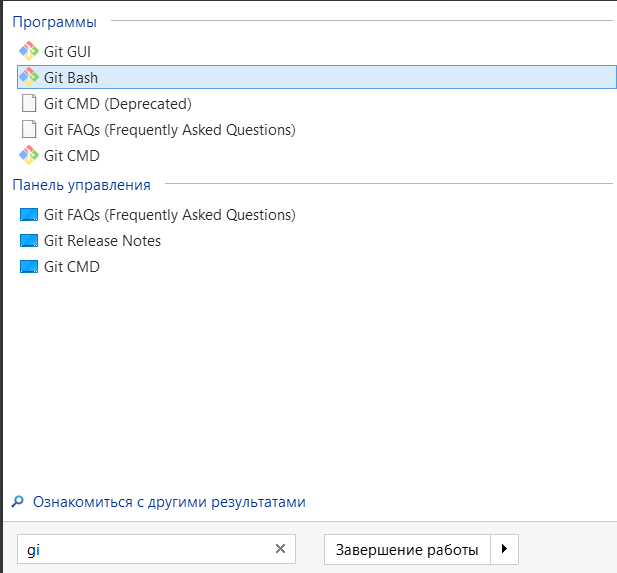
1. Выберите тематику программы, которую собираетесь написать. Создайте для нее рабочую директорию

Тематика программы как в прошлой методичке – различные вариации «Hello World», рабочая директория для неё такая:



1. Инициализируйте в рабочей директории репозиторий при помощи команды git init.

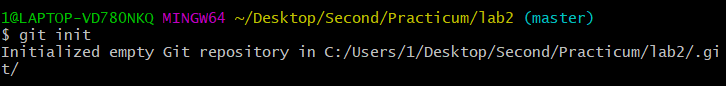
Теперь откроем Git Bash:



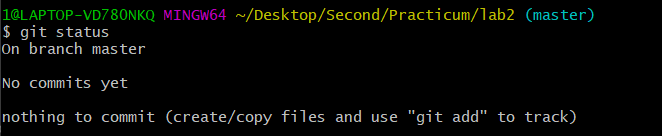
И затем переходим в рабочую директорию:



Затем инициализируем репозиторий в рабочей директории:

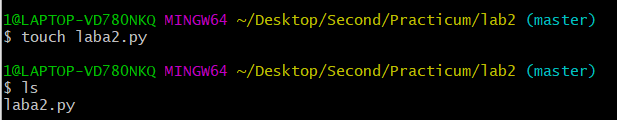


1. Выполните в репозитории команду git status. Проинтерпретируйте полученное сообщение.

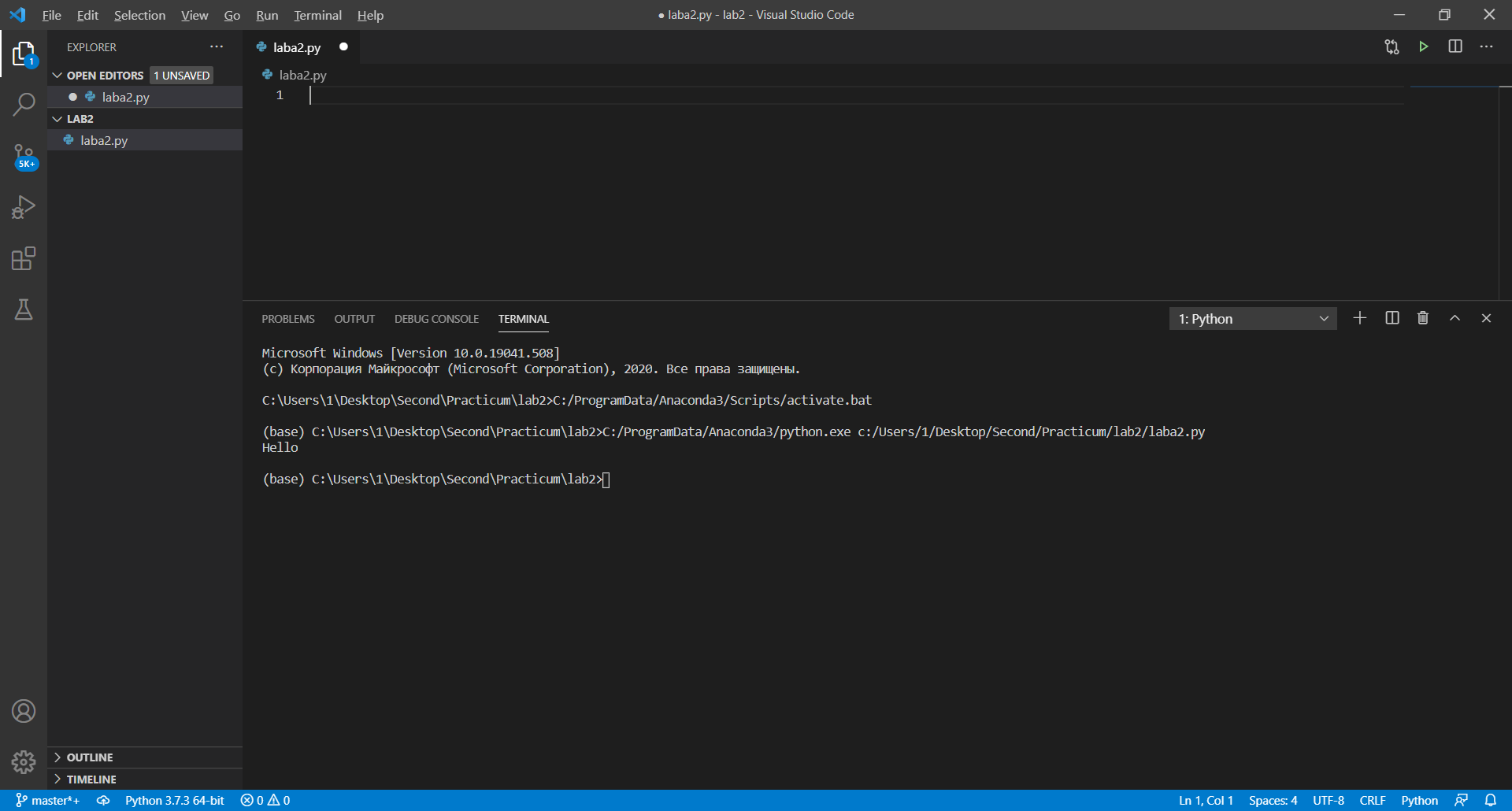


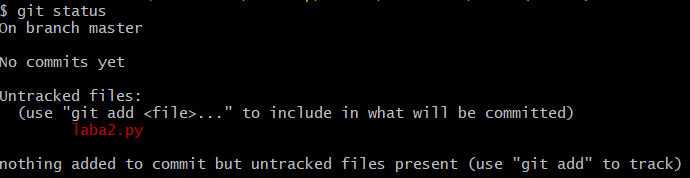
Интерпретируется это однозначно: у нас нет ещё ни проекта, ни коммитов для этого проекта.

1. Создайте файл для исходного текста программы. Выполните команду git status.



Вот мой созданный файл:

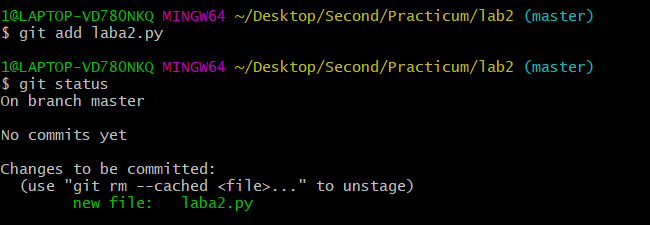




То есть, мы сделали изменение в репозитории – создали пустую программу, но не сохранили текущую версию нашего проекта – директорию с этим пустым файлом.

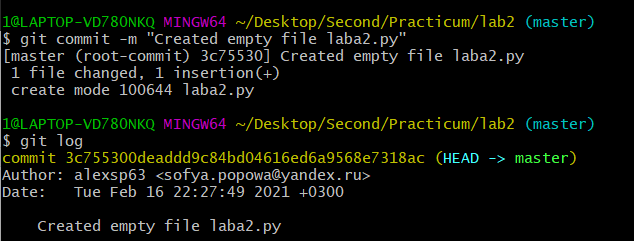
1. Добавьте созданный файл под версионный контроль при помощи команды git add. Еще раз выполните git status.

Добавили созданный файл под версионный контроль, используя нужную комманду, и проверили это через git status:



1. Сделайте начальный коммит при помощи команды git commit с опцией -m.

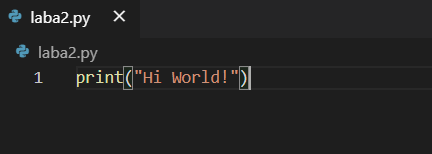
Сделала коммит, при этом в логах он отображается:



1. Сделайте еще несколько коммитов. Выполните команду git log для просмотра истории коммитов.

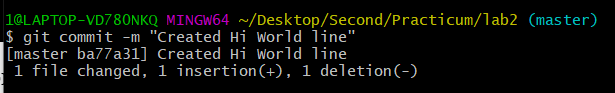
Сделаю первый коммит:

Изменю текст программы и сохраню:

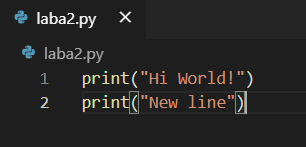


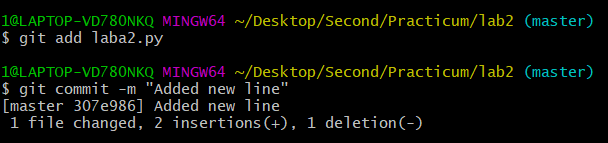
Затем сделаю сам коммит:



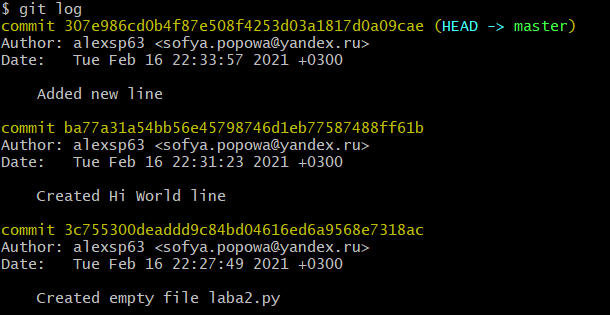


Аналогично сделаю ещё один коммит:





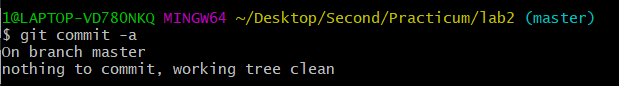
А затем посмотрим историю моих коммитов:



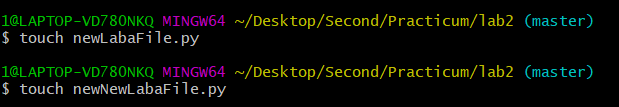
Здесь отображается всё так, как и должно быть.

1. Сделайте так, чтобы при коммите измененные файлы автоматически добавлялись в коммит.

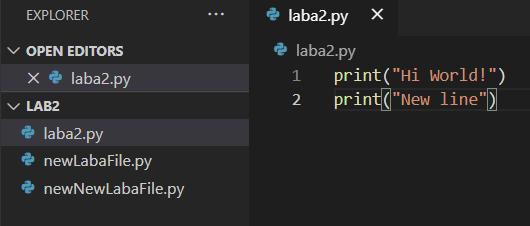
Это делается при помощи этой команды:



1. Добавьте еще несколько файлов с исходным текстом программы.

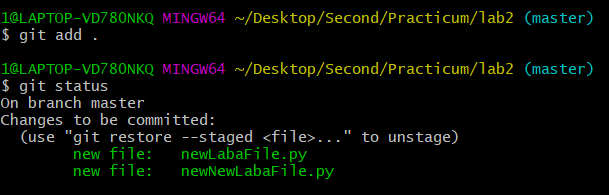


И вот они у меня появились:



1. Добавьте все новые файлы под версионный контроль одной командой.

Это делается так:



Здесь . значит, что под версионный контроль попадают все файлы директории.

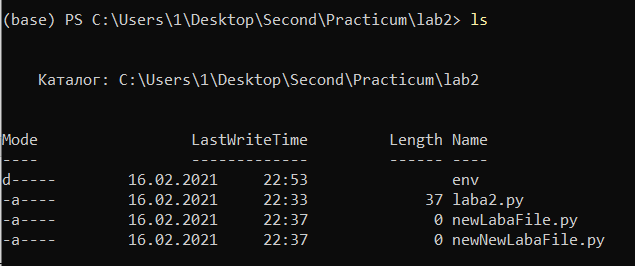
1. На всех стадиях работы пользуйтесь командой git status.

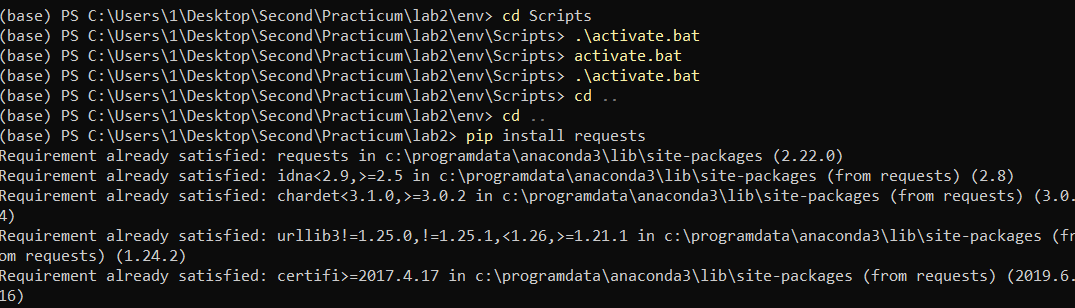
Хорошо.

1. Инициализируйте в рабочей директории виртуальное окружение (Если вы пишите не на Python, то можете инициализировать какой-либо программный фреймворк, либо начать работать в IDE, которая создает скрытую папку с настройками в рабочем каталоге).

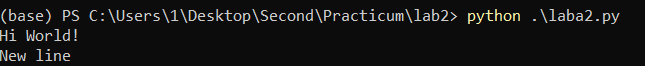
Теперь попробую установить виртуальное окружение, для этого пойду в Анакондовскую консоль (у нас питон всё устанавливает через неё), и напишем нужные команды:





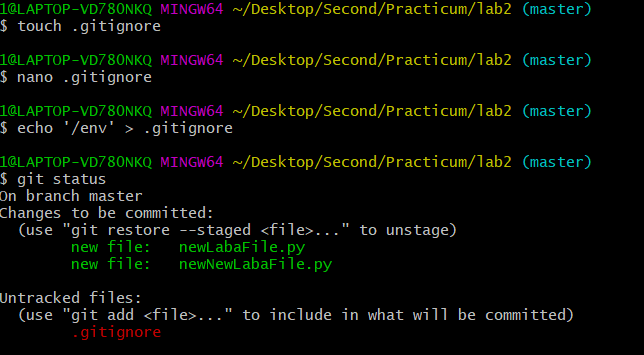


Теперь всё работает:



1. Добавьте созданную служебную папку в файл .gitignore. Проверьте, что они не добавляются в репозитории при добавлении новых файлов с исходным кодом.

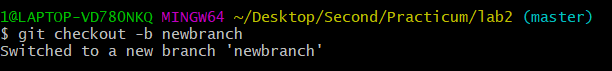
Делаем следующие команды:



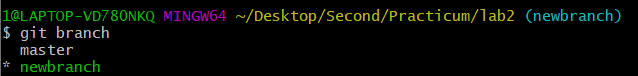
Теперь они не добавляются в репозитории.

1. Создайте новую тематическую ветку git branch. Перейдите в нее с помощью git checkout. Выведите на экран список всех веток.

Теперь создаём новую тематическую ветку и сразу переходим в неё:

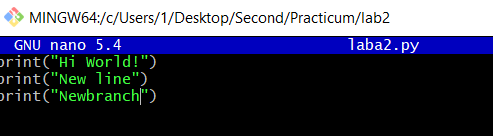


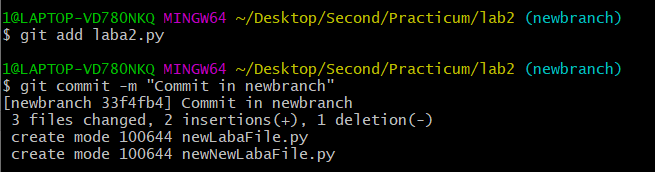
Теперь выведем на экран список всех веток:



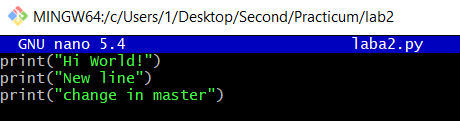
1. Сделайте несколько коммитов в основную и тематическую ветки.

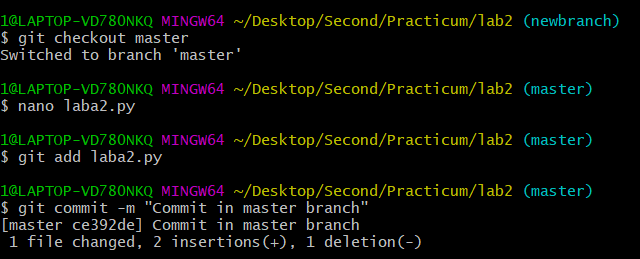
Сделаю изменение и коммит в новой ветке:





Теперь иду в master и меняю и коммичу что-нибудь там:

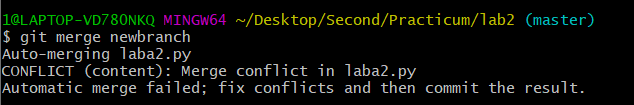




1. Слейте изменения в основную ветку с помощью git merge. Если произошел конфликт слияния, разрешите его и завершите слияние с помощью git commit.

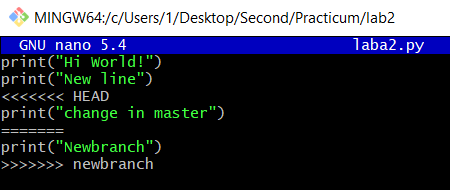
Теперь сделаем слияние: для этого в мастере напишем команду и имя ветки, с которой будем сливаться (в данном случае - newbranch).

У меня конфликт слияния:

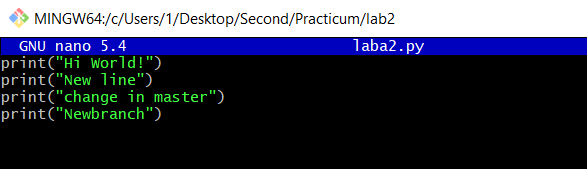


Надо его отрегулировать вручную.

Для этого зайдём в редактор и увидим типичный синтаксис конфликта: разные значки:



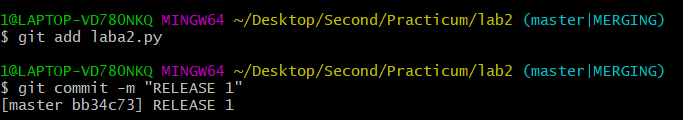
И теперь я вручную отредактирую данный файл, приведу его к такому виду, каким я хочу его видеть, и удалю метки:



И сохраню.

А теперь даже гит оповестил нас при слиянии, что нужно сделать коммит вручную, что я и сделаю.

1. При получении в процессе разработки программы в стабильно работающем состоянии, слейте это состояние в основную ветку и добавьте к коммиту слияния пометку с номером релиза.



И вот последние записи из истории коммитов, всё так, как и должно быть:

