

# **Введение в работу в среде программирования Jupyter .**

С помощью Jupyter можно редактировать и запускать код на языке программирования Python 3 и не только.

## **Запуск Jupyter**

Для запуска Jupyter необходимо:

1. Перейти по ссылки: <https://jupyter.dns-shop.ru/hub/login>
2. Ввести логин и пароль, авторизация доменная (логин и пароль, как при входе в систему).

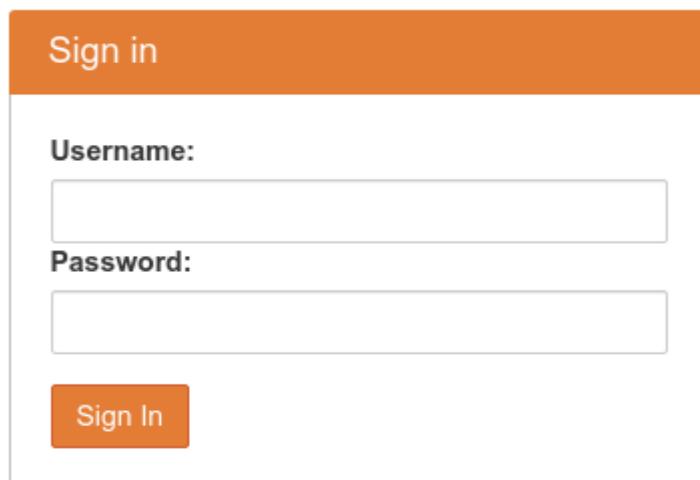


Рисунок 1 — окно авторизации

3. Нажать кнопку «Sign In».

В случае возникновения ошибки сотрудникам сервисной службы необходимо обращаться к Бровко Н.С.

## **Загрузка файлов в Jupyter**

Для загрузки файлов можно:

1. Перетащить и бросить файл в окно Jupyter .

## 2. Воспользоваться кнопкой upload

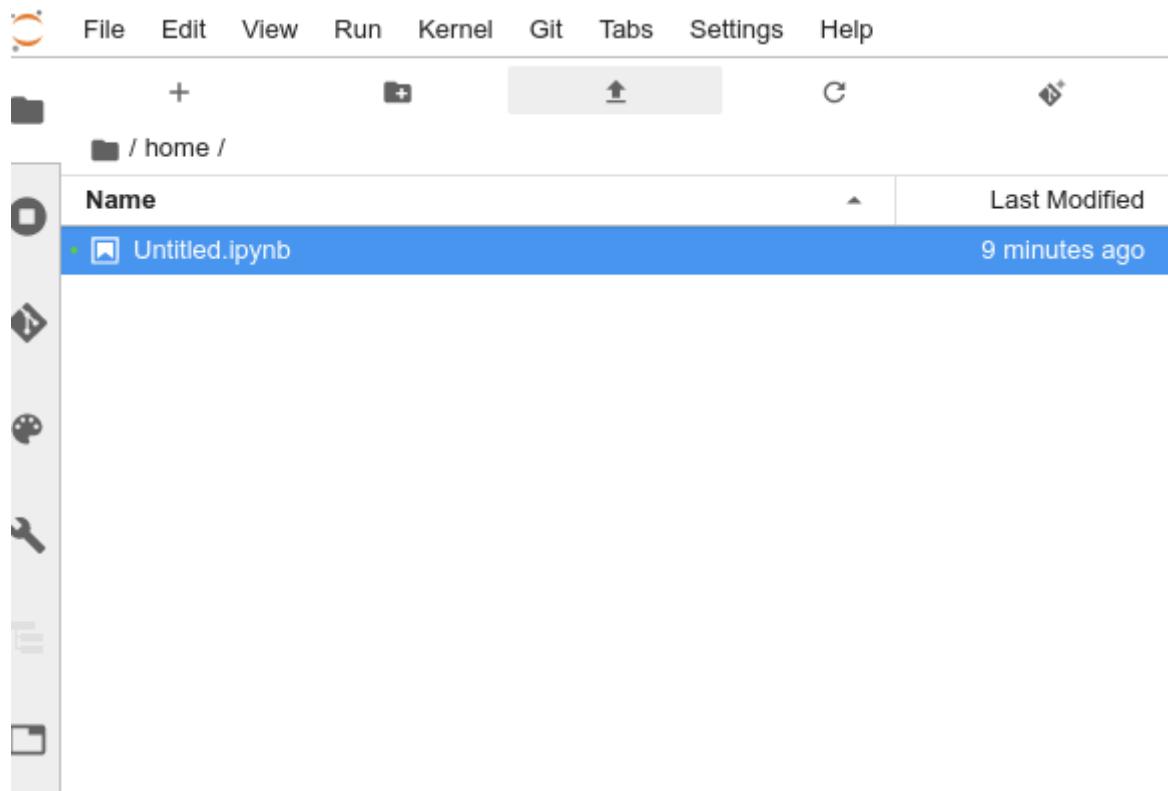


Рисунок 2 — загрузка файлов

## Как создать Jupyter Notebook для работы

Для того, чтобы создать новый блокнот необходимо:

1. Если лаунчер закрыт, то открыть новый лаунчер, воспользовавшись кнопкой
2. Создать блокнот

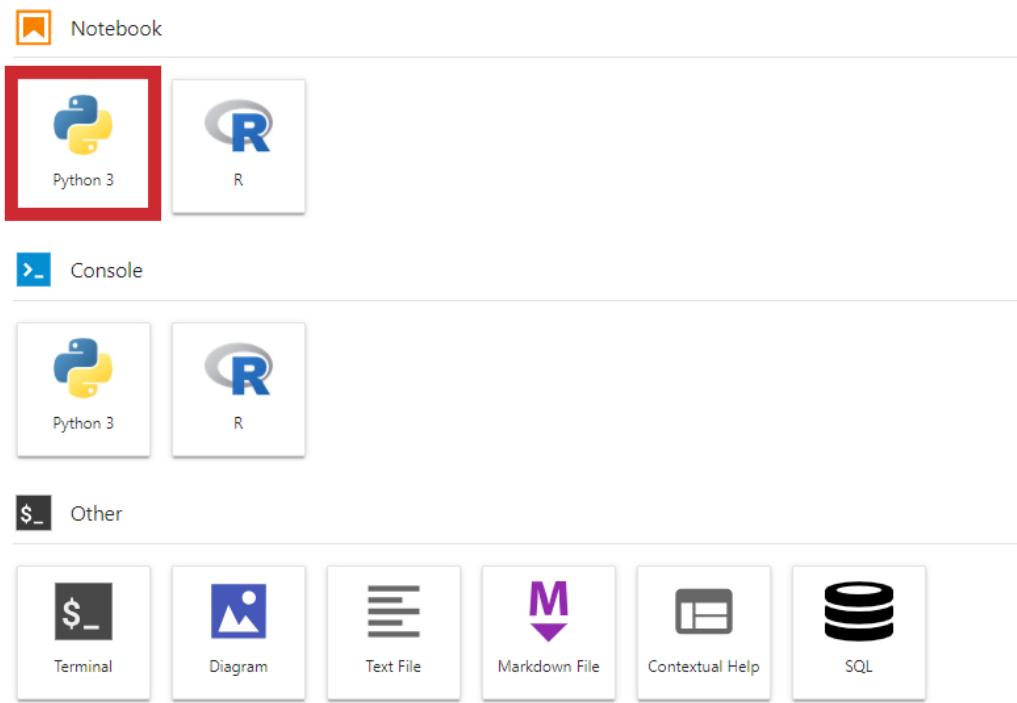


Рисунок 3 — создание блокнота

## Как переименовать Блокнот

Для того, чтобы переименовать блокнот необходимо:

1. Нажать по нему правой кнопкой мыши
2. Выбрать пункт «Rename»
3. Ввести новое название

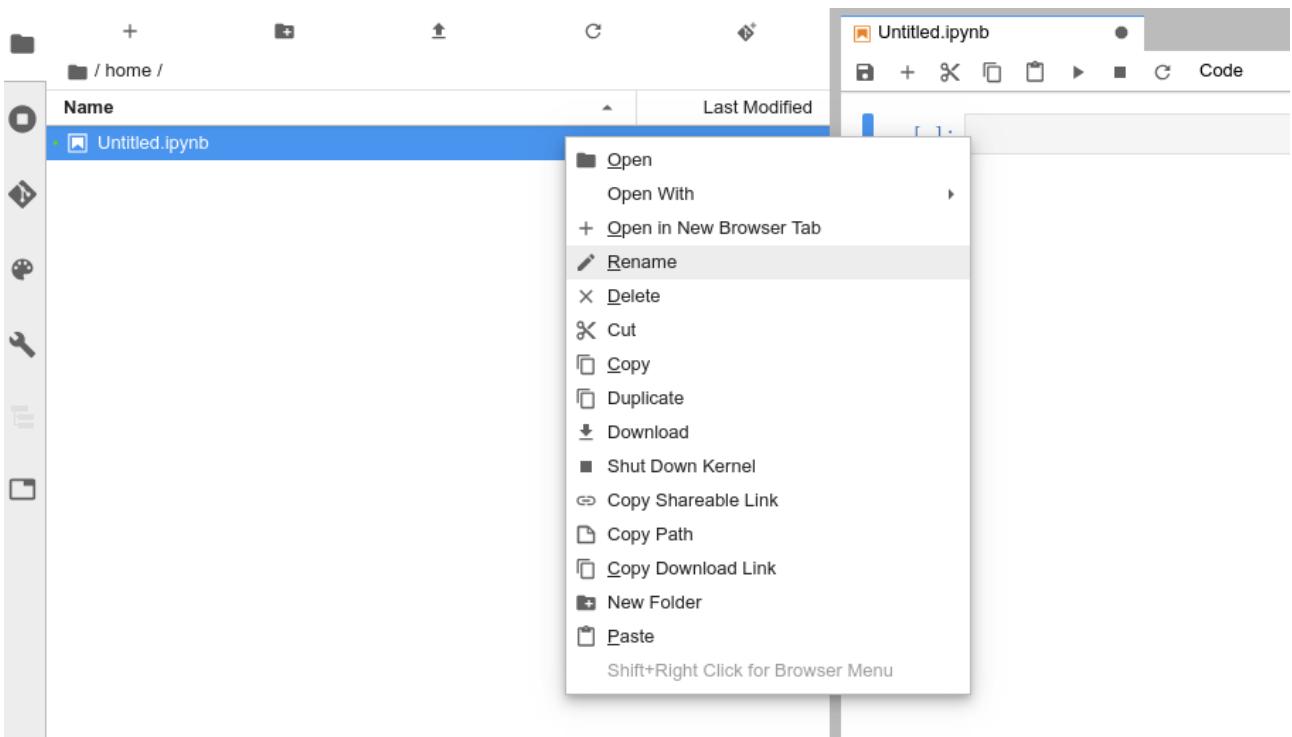


Рисунок 4 — переименования блокнота

## Основные кнопки в интерфейсе JupyterNotebook

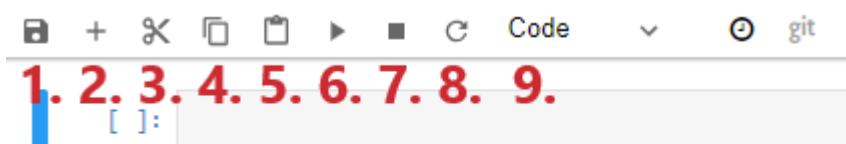


Рисунок 5 — основные кнопки интерфейса

1. Сохранить изменения в блокноте
2. Создать новую ячейку
3. Вырезать ячейку
4. Копировать ячейку
5. Вставить ячейку
6. Запустить код из ячейки
7. Перезапустить ядро, необходимо использовать если программа начала тормозить
8. Выбор типа ячейки:
  1. Code — ячейка с кодом

## 2. Markdown — ячейка с некоторым текстовым описанием

### **Основной синтаксис MarkDown**

#### **Параграфы**

*<пустая строка>*

*Текст параграфа. Может быть  
разбит на несколько строк.*

*<пустая строка>*

#### **Переводы строк**

*Текст параграфа.<пробел><пробел>*

*Этот текст будет с новой строки.*

#### **Заголовки**

*# Заголовок первого уровня H1*

*## Заголовок второго уровня H2*

*...*

*##### Заголовок шестого уровня H6*

#### **Цитаты**

*> Это*

*> многострочная*

*> цитата*

#### **Списки**

##### **Ненумерованные**

*+ Первая строка*

*+ Вторая строка*

*+ Третья строка*

##### **Нумерованные**

*1. Первая строка*

*1. Вторая строка*

*1. Третья строка*

## Горизонтальная линия

\* \* \*

## Ссылки

Это простая [ссылка](<https://gist.github.com/neretin-trike/>) в тексте.

Это [ссылка](<https://gist.github.com/neretin-trike/> "Gists") с атрибутом title.

## Ссылка в тексте

По адресу <<https://gist.github.com/neretin-trike/>> находится мой профиль на GitHub.

## Выделение строк

\* Наклонный текст \*

\*\* Полужирный текст \*\*

`Код в строке`

~ Зачеркивание

## Изображения

![alt-текст](<http://example.com/image.jpg> "Необязательный титул")

## Основные быстрые клавиши в JupyterNotebook

Ctrl + Enter — выполнить код в ячейки

Ctrl + F — поиск в блокноте

Shift + Tab — открыть документацию модуля

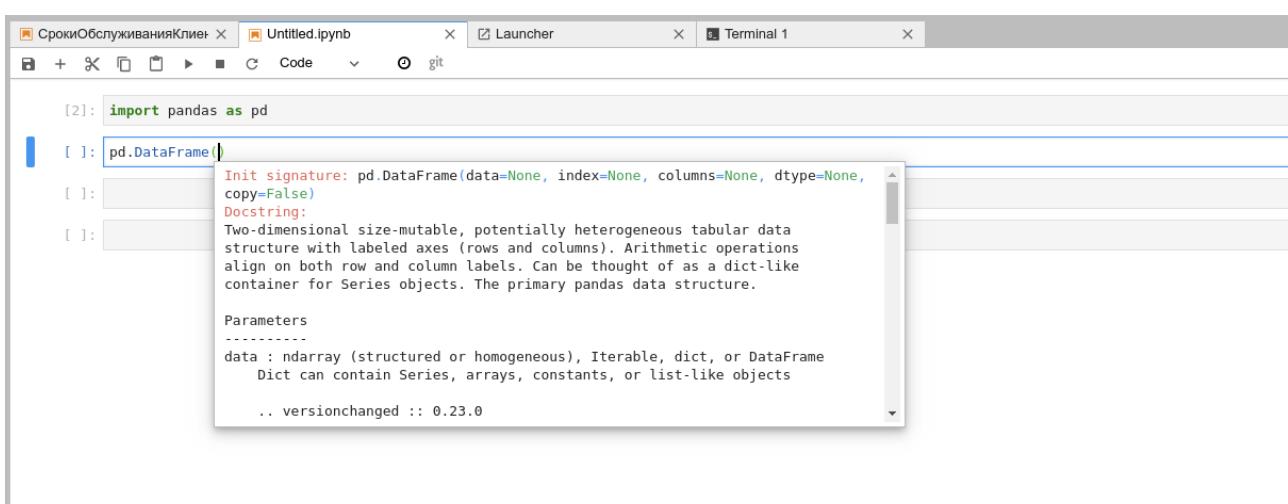


Рисунок 6 — открытие документации модуля

# Библиотеки в Python

Библиотеки Python — это файлы с шаблонами кода. Их придумали для того, чтобы людям не приходилось каждый раз заново набирать один и тот же код: они просто открывают файл, вставляют свои данные и получают нужный результат.

Достаточно большое количество библиотек автоматически установлены в Jupyter.

## Как установить библиотеку в Python

Для того, чтобы установить библиотеку в Jupyter необходимо из JupyterNotebook выполнить команду `!pip3 install НазваниеБиблиотеки --user`.

Список библиотек для установки, они понадобятся в рамках обучающего курса:

- `!pip3 install xlrd --user`
- `!pip3 install openpyxl --user`

## Как импортировать библиотеку в свой блокнот

Для подключения библиотеки необходимо в JupyterNotebook выполнить команду: `import НазваниеБиблиотеки as МаскаНазванияБиблиотеки`. Пример, `import pandas as pd` — импорт библиотеки для работы с таблицами. В следующих уроках она будет активно использоваться.



Рисунок 7 — импорт библиотеки

## Как перезапустить сервер

Для перезапуска сервера необходимо:

- Зайти в панель управления сервером

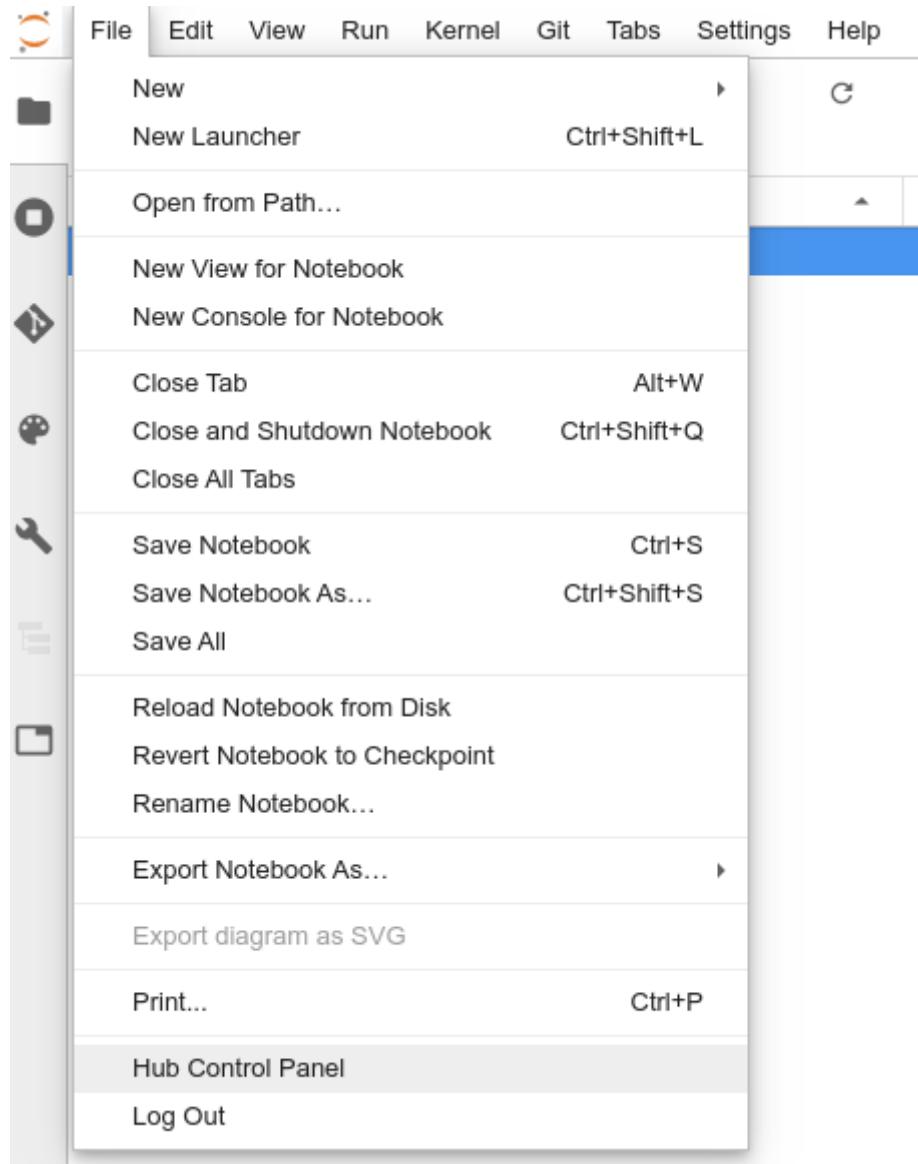


Рисунок 8 — Hub Control Panel

- Нажать на кнопку Stop My Server



Рисунок 9 — Остановка сервера

- А после нажать Start My Server

## Система контроля версий Git

Система контроля версий — это система, записывающая изменения в файл или набор файлов в течение времени и позволяющая вернуться позже к определённой версии.

Репозиторий или “геро” это коллекция всех изменений, которые были совершены на протяжении всего времени после инициализации репозитория.

## Как клонировать репозиторий в Jupyter

- Перейти на страницу репозитория в git  
(<https://git.dns-shop.ru/Brovko.NS/python3-course>)

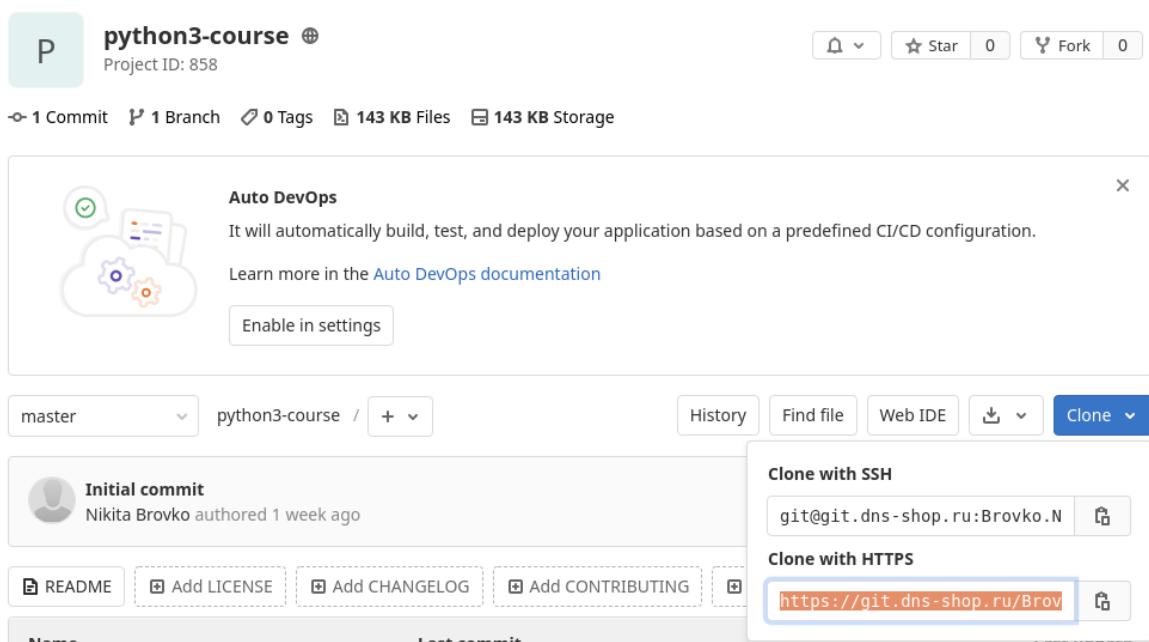


Рисунок 10 — обучающий репозиторий

- Скопировать ссылку для клонирования, как на скриншоте

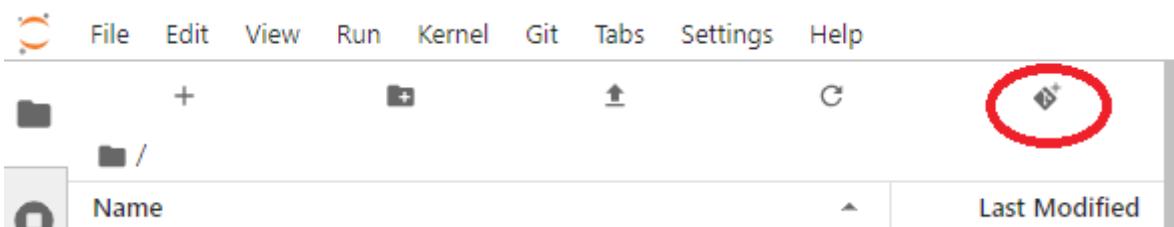


Рисунок 11 — клонирование репозитория

- Зайти в Jupyter и нажать на кнопку в верхней части экрана, как на рисунке ниже
- В появившиеся окно вставить ссылку <https://git.dns-shop.ru/Brovko.NS/python3-course.git> и нажать на кнопку CLONE.

## Как получить изменения из репозитория

- В Jupyter зайти во вкладку Git
- Нажать на кнопку «Pull latest changes»

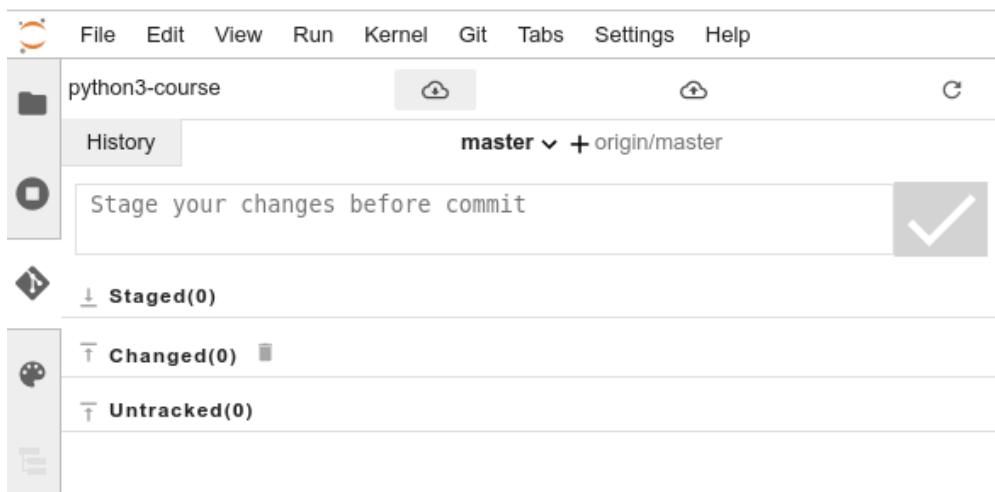


Рисунок 11 — получения изменения из репозитория

## Задания к самостоятельному выполнению

1. Зайти в Jupyter (<https://jupyter.dns-shop.ru/hub/login>)
2. Установить необходимые библиотеки xlrd, openpyxl
3. Клонировать обучающий репозиторий