

Работа с IPython и Jupyter Notebook

IPython представляет собой мощный инструмент для работы с языком Python. Базовые компоненты IPython – это интерактивная оболочка с широким набором возможностей и ядро для Jupyter. Jupyter Notebook является графической веб-оболочкой для IPython, которая расширяет идею консольного подхода к интерактивным вычислениям.

Основные отличительные особенности данной платформы – это комплексная интроспекция объектов, сохранение истории ввода на протяжении всех сеансов, кэширование выходных результатов, расширяемая система “магических” команд, логирование сессии, дополнительный командный синтаксис, подсветка кода, доступ к системной оболочке, стыковка с pdb отладчиком и Python профайлером.

IPython позволяет подключаться множеству клиентов к одному вычислительному ядру и, благодаря своей архитектуре, может работать в параллельном кластере.

В Jupyter Notebook вы можете разрабатывать, документировать и выполнять приложения на языке Python, он состоит из двух компонентов: веб-приложение, запускаемое в браузере, и ноутбуки – файлы, в которых можно работать с исходным кодом программы, запускать его, вводить и выводить данные и т.п.

Веб приложение позволяет:

- редактировать Python код в браузере, с подсветкой синтаксиса, автоотступами и автодополнением;
- запускать код в браузере;
- отображать результаты вычислений с медиапредставлением (схемы, графики);
- работать с языком разметки Markdown и LaTeX.

Ноутбуки – это файлы, в которых сохраняются исходный код, входные и выходные данные, полученные в рамках сессии. Фактически, он является записью вашей работы, но при этом позволяет заново выполнить код, присутствующий на нем. Ноутбуки можно экспортировать в форматы PDF, HTML.

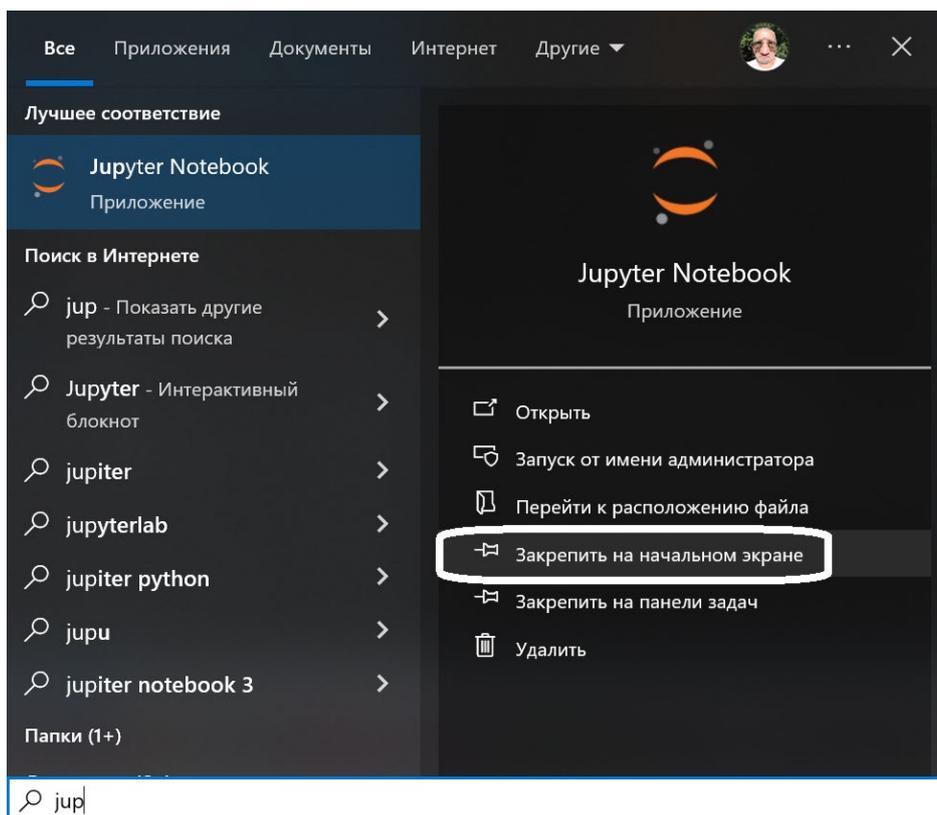
Установка и запуск Jupyter Notebook

Jupyter Notebook входит в состав Anaconda. Описание процесса установки можно найти в предыдущей инструкции.

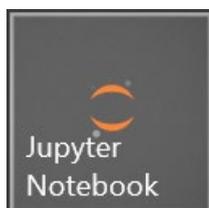
Запустить Jupyter Notebook можно из окна Anaconda Navigator. Для этого нужно нажать кнопку «**Launch**» (Запуск) на соответствующей иконке:



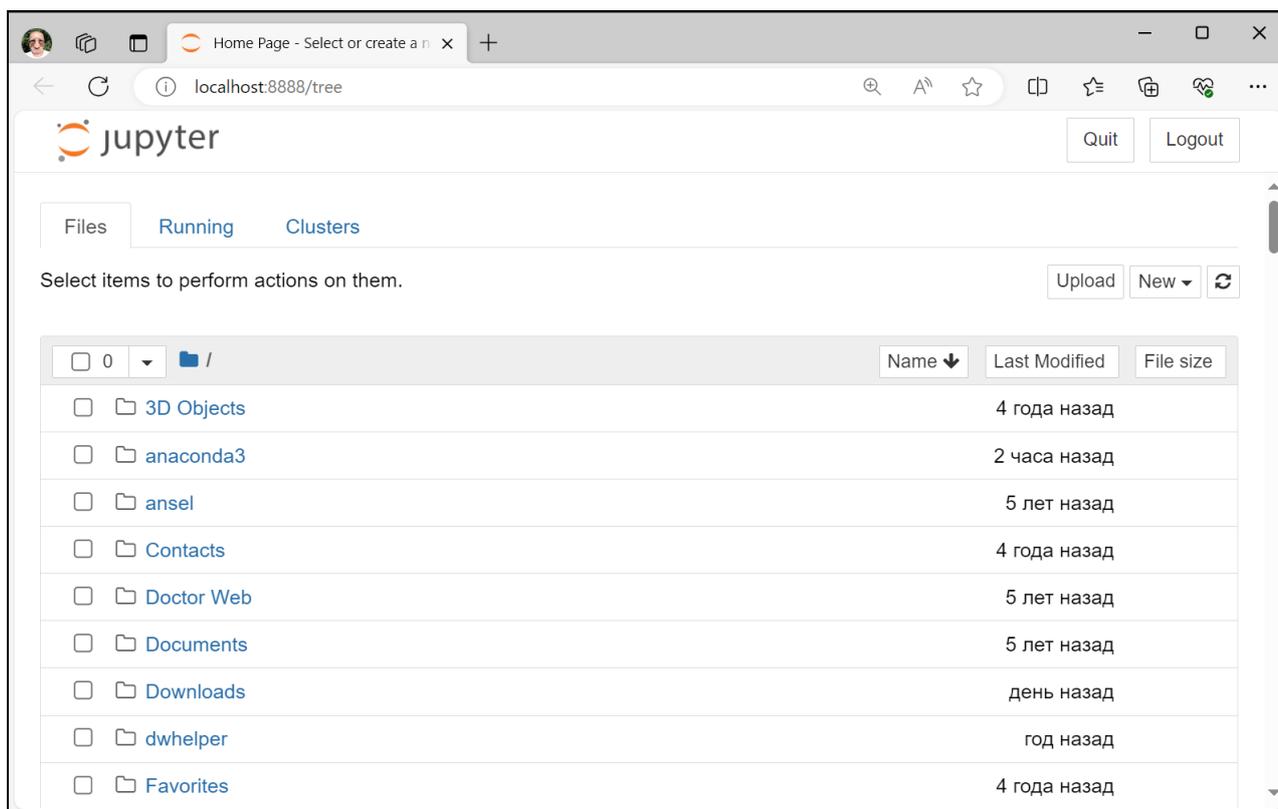
Также можно найти приложение через строку поиска Windows 10 и закрепить на начальном экране:



Теперь можно быстро запускать Jupyter Notebook иконкой через меню Пуск, не запуская Anaconda Navigator:



При запуске открывается новая вкладка вашего браузера, установленного по умолчанию, которая будет выглядеть примерно так, как показано на следующем скриншоте:



Это пока не блокнот. Это **панель инструментов ноутбука**, специально разработанная для управления ноутбуками Jupyter.

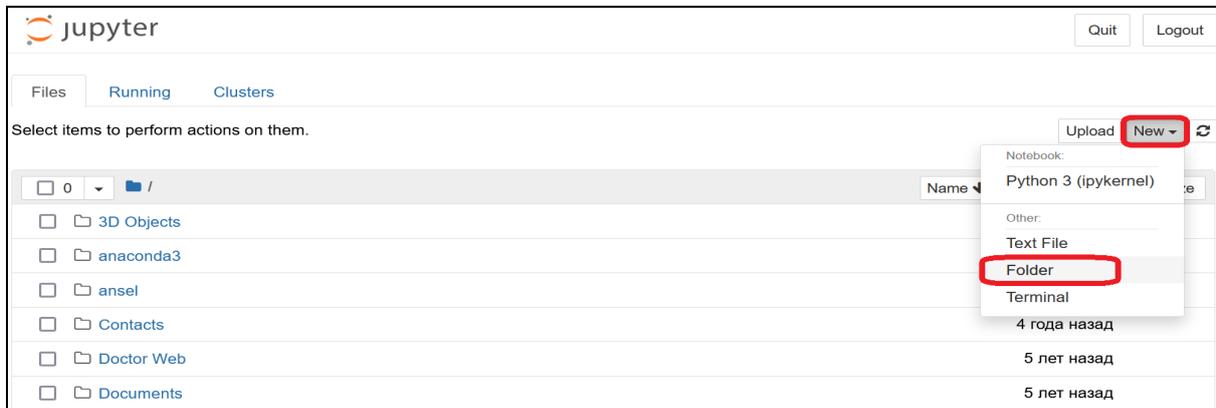
Имейте в виду, что панель управления предоставит вам доступ только к файлам и подпапкам, содержащимся в каталоге запуска Jupyter.

Обратите внимание на URL-адрес панели управления: **http://localhost:8888/tree**. Localhost не является веб-сайтом, но этот адрес указывает, что контент обслуживается с вашего собственного локального компьютера. Ноутбуки и панель инструментов Jupyter – это веб-приложения, и Jupyter запускает локальный веб-сервер Python для обслуживания этих приложений в веб-браузере, что делает его практически независимым от платформы и открывает возможности для более удобного обмена в Интернете.

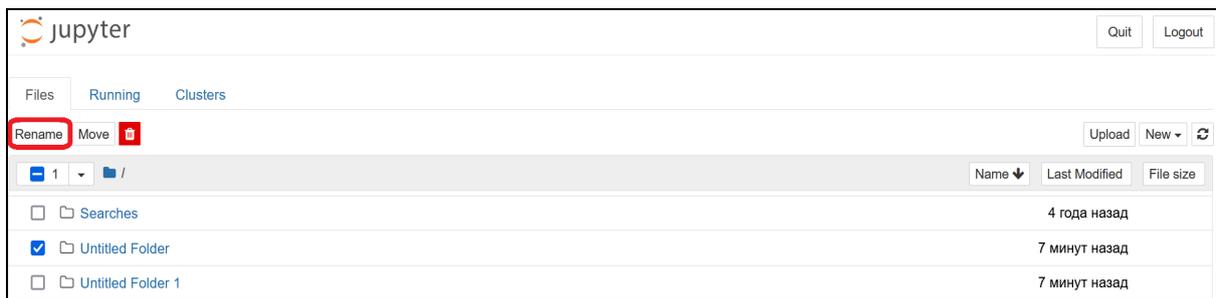
Примеры работы

Будем следовать правилу: лучше один раз увидеть... Рассмотрим несколько примеров, выполнив которые, вы сразу поймете принцип работы с Jupyter Notebook.

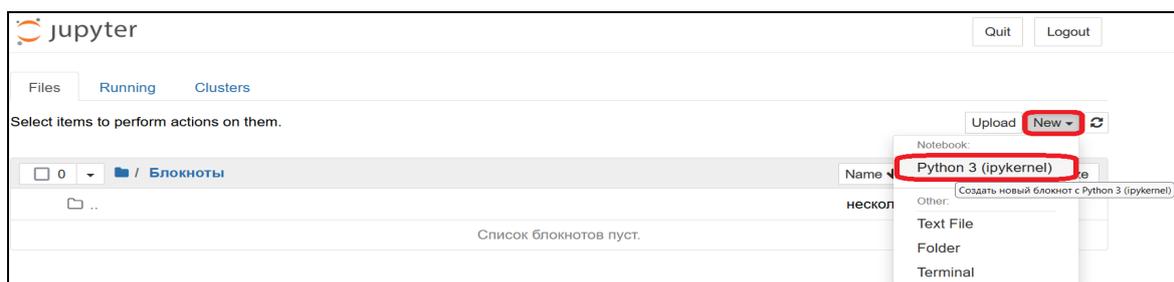
Запустите Jupyter Notebook и создайте папку для наших примеров, для этого нажмите на «New» (Новый) в правой части экрана наверху и выберите в выпадающем списке «Folder» (Папка):



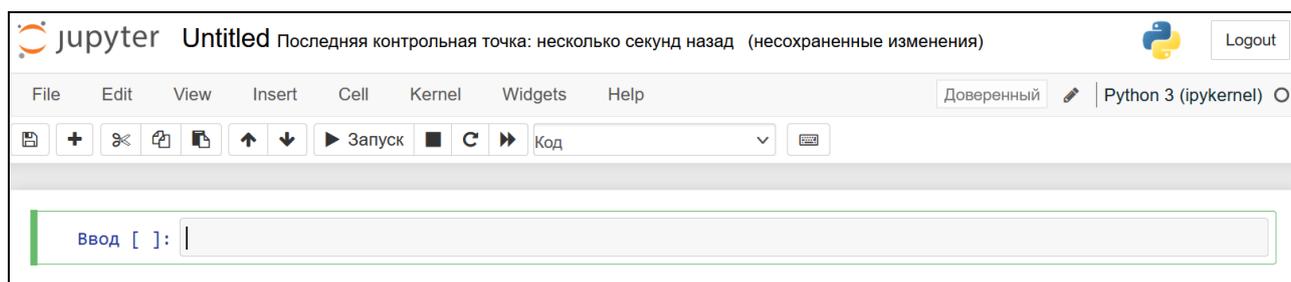
По умолчанию папке присваивается имя “Untitled Folder”. Давайте переименуем ее в “Блокноты”, для этого поставьте галочку напротив имени папки и нажмите на кнопку «Rename» (Переименовать):



Зайдите в эту папку и создайте в ней новый ноутбук, воспользовавшись той же кнопкой «New», только на этот раз нужно выбрать «Python 3 (ipykernel)»:



Ваш первый блокнот откроется в новой вкладке – каждый блокнот использует свою вкладку, поэтому вы можете открывать несколько блокнотов одновременно:



Если вы переключитесь обратно на панель инструментов, вы увидите новый файл **Untitled.ipynb** и зеленый текст, который говорит о том, что ваш ноутбук работает.

Что за файл ipynb?

Будет полезно понять, что это за файл на самом деле. Каждый файл **.ipynb** представляет собой текстовый файл, который описывает содержимое вашей записной книжки в формате **JSON**. Каждая ячейка и ее содержимое, включая вложения изображений, которые были преобразованы в строки текста, перечислены в нем вместе с некоторыми метаданными.

Код на языке Python или текст в нотации языка разметки **Markdown** нужно вводить в ячейки.

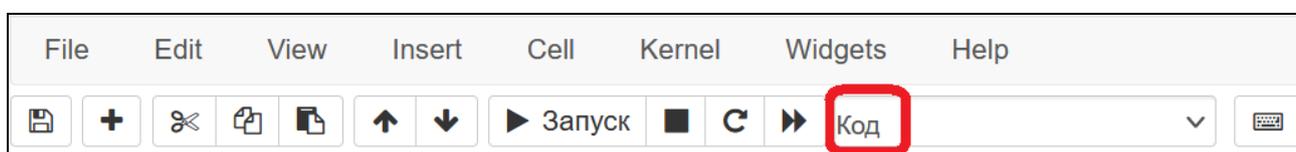
Ячейки (Cells)

Ячейки образуют структуру ноутбука. На скриншоте нового блокнота это поле с зеленым контуром. Есть два основных типа ячеек, которые мы рассмотрим:

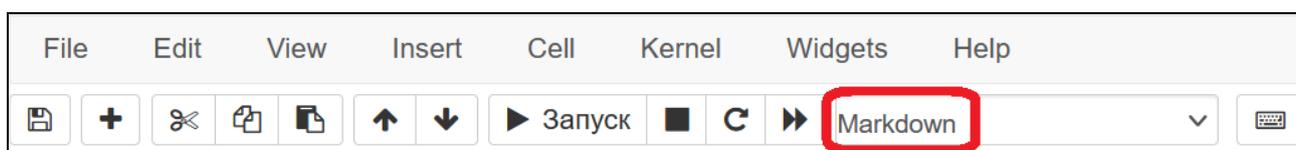
- **Ячейка кода** содержит код, который должен быть выполнен в ядре, и отображает его вывод ниже.
- **Ячейка Markdown** содержит текст, отформатированный с использованием языка разметки Markdown, и отображает его вывод.

Первая ячейка в новом блокноте всегда является ячейкой **кода**.

Если вы вводите в ячейку код Python, то на панели инструментов нужно выставить свойство **Code** (или **Код**):



Если это Markdown текст – выставить **Markdown**:



Вы всегда можете определить в каком режиме вы редактируете ячейку, поскольку ячейки кода всегда имеют метку **Ввод []** (или **In []**) слева, а ячейки Markdown – нет:

Ввод []:

ИЛИ

In []:

Часть «In» метки просто короткая запись слова **Input**, а номер метки в квадратных скобках указывает, когда ячейка была выполнена в ядре.

Для начала решим простую арифметическую задачу. Выставьте режим ячейки Code, введите в ячейке **2 + 3** и нажмите кнопку «▶ Запуск» («▶ Run»). Можно использовать горячие клавиши **Ctrl+Enter** или **Shift+Enter**, в первом случае введенный вами код будет выполнен интерпретатором Python, во втором – будет выполнен код и создана новая ячейка, которая расположится уровнем ниже так, как показано на рисунке:

```
Ввод [4]: 2 + 3
Out[4]: 5
Ввод [ ]:
```

Если у вас получилось это сделать, выполните еще несколько примеров:

```
In [2]: 3 + 2
Out[2]: 5

In [3]: a = 5
        b = 7
        print(a + b)
12

In [4]: n = 7
        for i in range(n):
            print(i*10)
0
10
20
30
40
50
60

In [5]: i = 0
        while True:
            i += 1
            if i > 5:
                break
            print("Test while")
Test while
Test while
Test while
Test while
Test while
```