

Лекція №1

Опис, встановлення та основи роботи з Jupiter Notebook

Jupyter Notebook - неймовірно потужний інструмент для інтерактивної розробки та подання проектів в області наук про дані. У цій лекції ви дізнаєтеся, як налаштувати Jupyter Notebooks на локальному комп'ютері і як почати використовувати його в ваших проектах.

Почнемо з визначення: що таке «notebook» (блокнот)? Блокнот об'єднує код і результат його виведення в єдиний документ, який об'єднує візуалізацію, розмічений текст, математичні рівняння та інші мультимедійні дані. Цей інтуїтивно зрозумілий робочий процес сприяє ітеративній і швидкій розробці, що робить ноутбуки все більш популярним вибором для подання в даних і їх аналізу.

Проект Jupyter є наступником більш раннього проекту IPython Notebook, який вперше був опублікований в якості прототипу в 2010 році. Хоча в Jupyter Notebooks можна використовувати з багатьма різними мовами програмування, саме Python, є найбільш поширеним варіантом використання.

Встановлення:

Найпростіше почати роботу з Jupyter Notebooks, встановивши дистрибутив Anaconda.



Anaconda є найбільш широко використовуваним дистрибутивом Python для роботи з даними і поставляється з попередньо встановленими найбільш популярними бібліотеками та інструментами. Деякі з найбільших бібліотек Python, включених в Anaconda, включають NumPy (<https://numpy.org/>), pandas (<https://pandas.pydata.org/>) і Matplotlib (<https://matplotlib.org/>).

Повний список - <https://docs.anaconda.com/anaconda/packages/pkg-docs/>

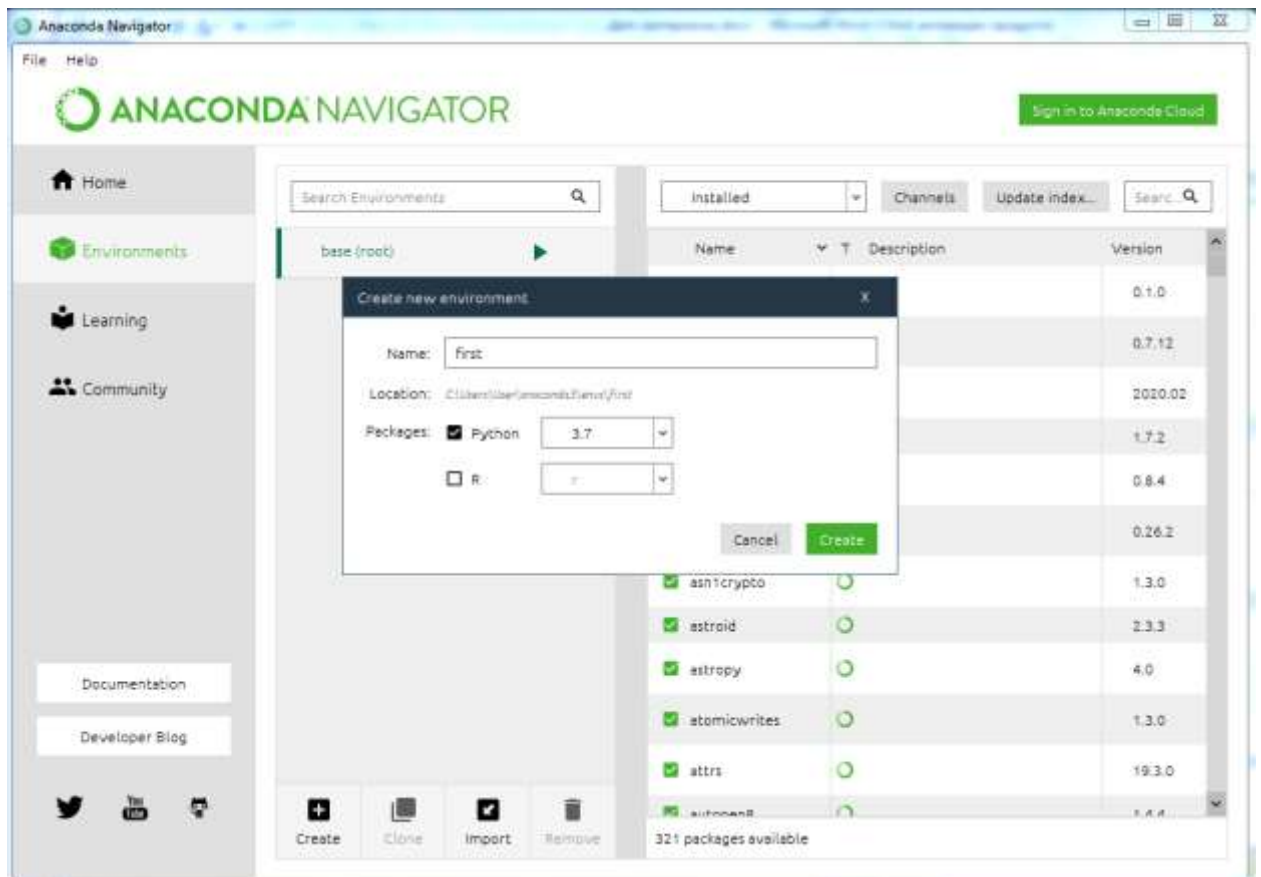
Це дозволяє приступити до роботи, без зайвого управління установками або занепокоєння про залежності і проблемах установки, пов'язаних з ОС.

Для встановлення Anaconda:

1. Завантажте останню версію Anaconda
<https://www.anaconda.com/products/individual>

2. Встановіть Anaconda виконуючи інструкціям.

Після встановлення буде відкрито Anaconda Navigator:



Для зручності роботи рекомендується створити окреме оточення - це дозволить уникнути конфліктів версій (наприклад коли потрібна інша версія пітона або бібліотеки). Для цього в Anaconda Prompt потрібно запустити команду:

```
conda create --name first python=3.7
```

де **first** - назва оточення. Можна також вказати потрібну версію Python.

Після створення оточення, його необхідно активувати командою

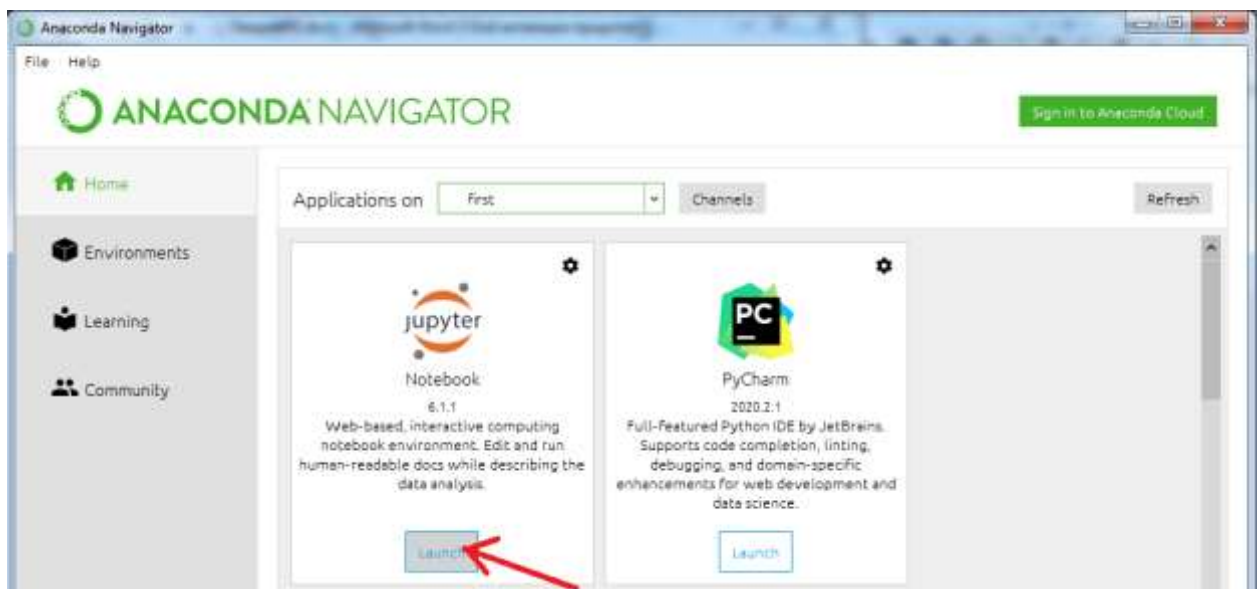
```
conda activate first
```

Створити нове оточення можна також і в вкладці environments, натиснувши на Create в Anaconda Navigator.

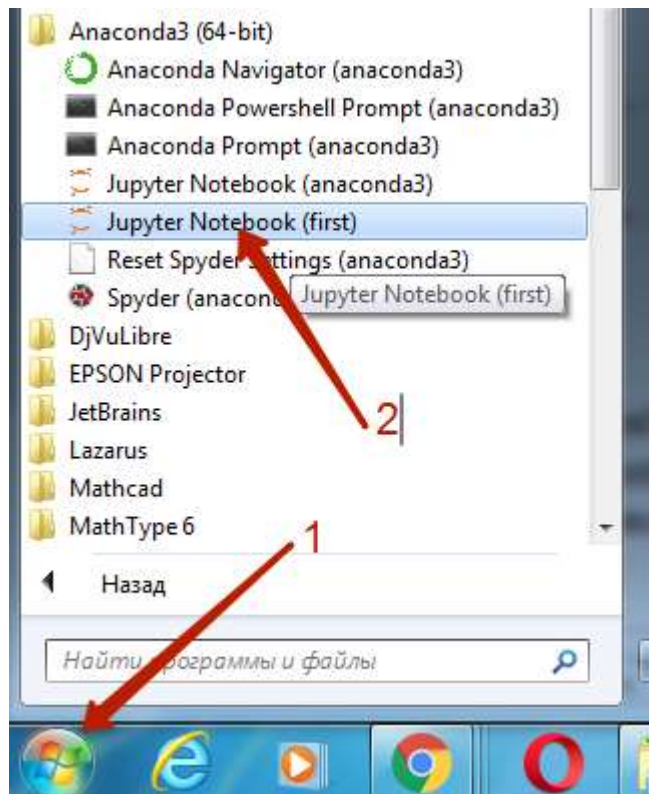
Створення оточення може зайняти кілька хвилин. Після установки в Anaconda Navigator можна вибрати оточення:



Далі потрібно встановити і запустити в обраному оточенні Jupyter Notebook.



Можна також запустити блокнот із меню Пуск в групі застосунків Anaconda



Для просунутих користувачів

1. Встановлення Python – скачати файл завантаження з сайту <https://www.python.org/>

2. Запустити файл. Встановити PATH – шлях до папки, в якій знаходиться файл python.exe: Панель управління→Система і безпека→Система → Дополнительные параметры системы, вкладка «Дополнительно», кнопка «Переменные среды».

3. Встановити Jupyter Notebook

```
pip3 install jupyter
```

4. Запуск блокнота

```
jupyter notebook
```

Подальшим розвитком Jupyter Notebook став Jupyter Lab, гнучке, інтегроване, легко розширюване середовище, що підтримує одночасну роботу з декількома блокнотами Jupyter, текстовими файлами, датасетами, терміналами та іншими компонентами. Аналогічно класичним IDE в JupyterLab можна впорядковувати документи в робочій області в зручному порядку за допомогою вкладок і роздільників.

Встановлення Jupyter Lab

- за допомогою conda

```
conda install -c conda-forge jupyterlab
```

- за допомогою pip

```
pip3 install jupyterlab
```

Виклик Jupyter Lab із Anaconda Prompt

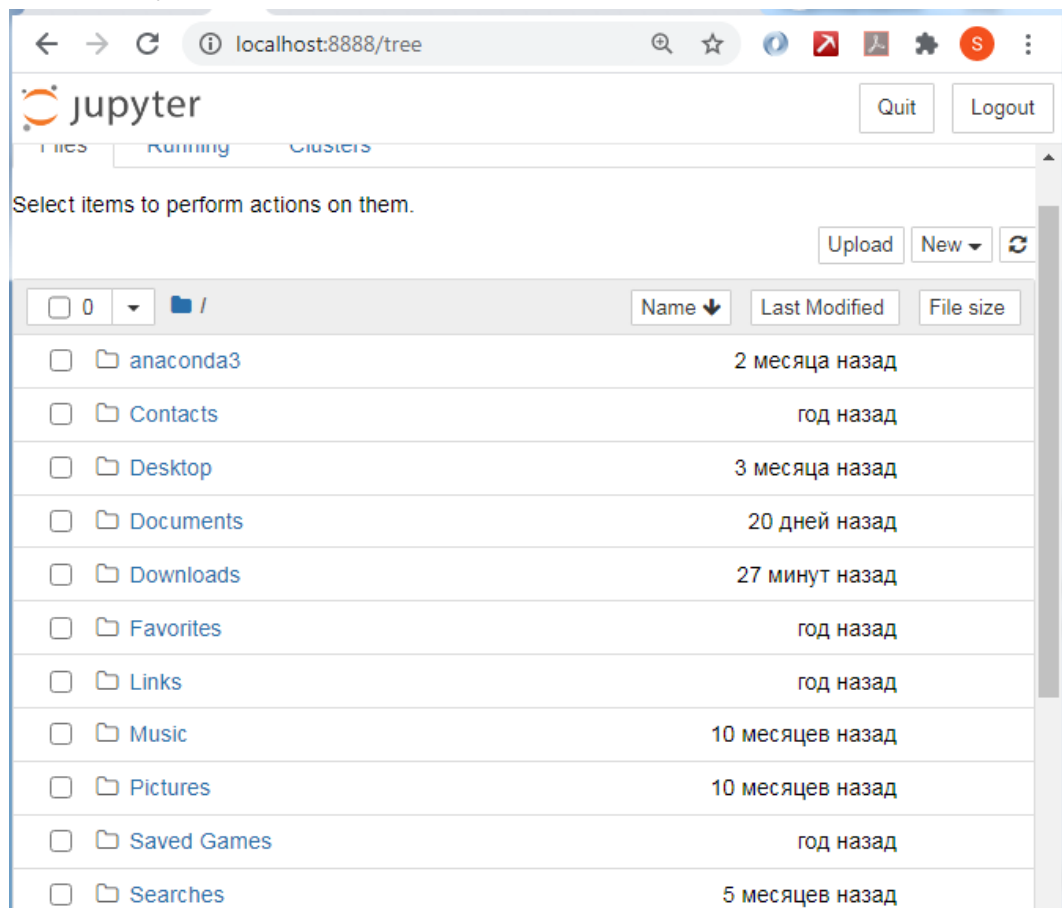
```
jupyter lab
```

Про роботу в Jupyter Lab можна прочитати тут - <https://jupyterlab.readthedocs.io/en/stable/index.html>.

Далі описується робота в Jupyter Notebook

Створення першого блокнота

Після запуску блокнота в браузері відкриється нова вкладка, що буде виглядати наступним чином:



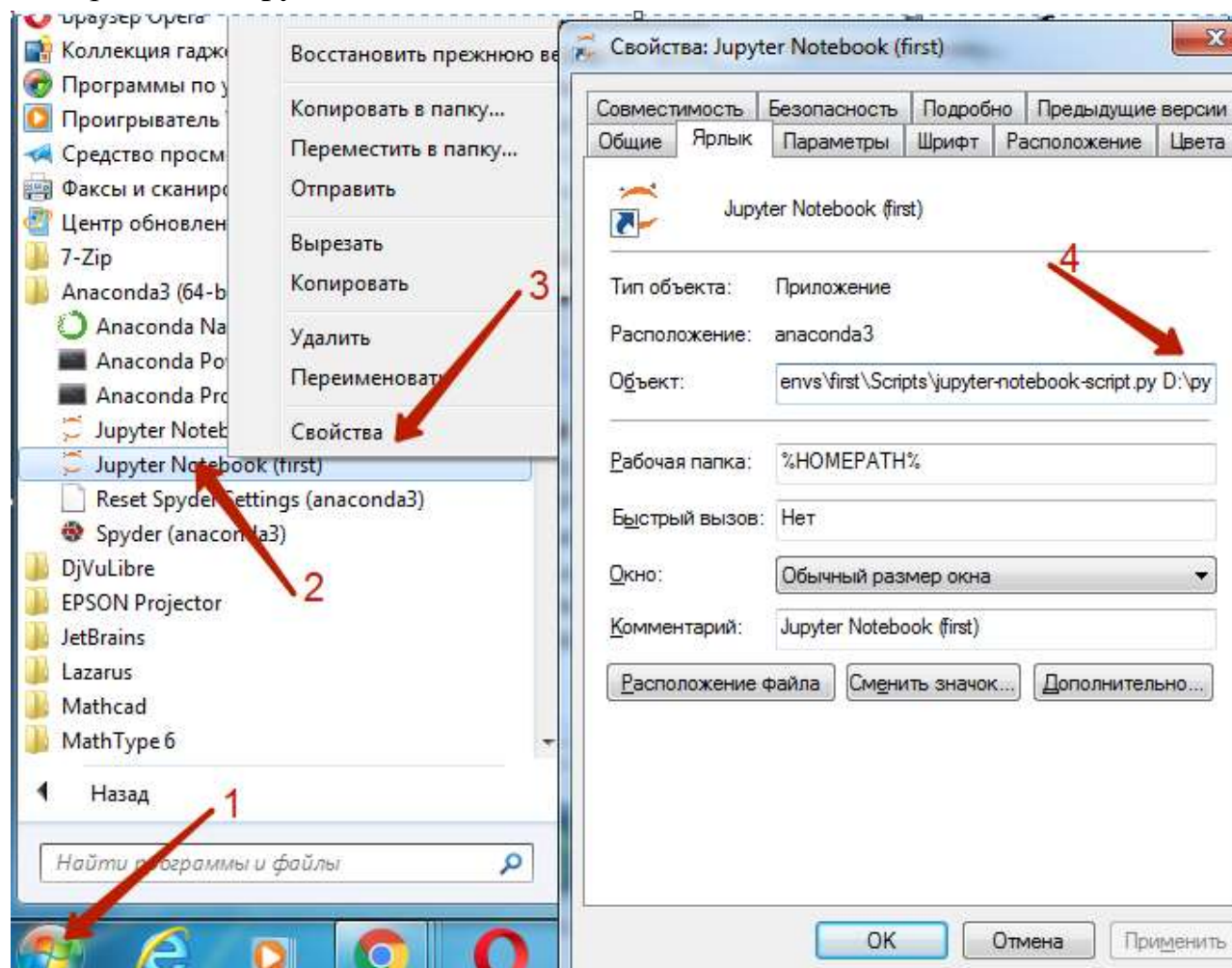
Це – панель інструментів ноутбука, створена для управління ноутбуками Jupyter.

Майте на увазі, що панель управління надасть вам доступ тільки до файлів і підпапок, що містяться в каталозі запуску Jupyter; проте каталог запуску може бути змінений.

Зміна стартового каталогу Jupyter Notebook

При запуску із панелі Пуск:

1. Клацнути мишею Пуск>Все программы>Anaconda
2. Правою кнопкою миші клацнути на назві потрібного блокноту та обрати пункт «Свойства»
3. Замінити в полі «Объект» %USERPROFILE% на шлях до потрібної папки, Наприклад, D:\py



При запуску з терміналу (вікна *cmd* або *anaconda prompt*)

Спосіб 1. (працює якщо не налаштований шлях в *jupyter_notebook_config.py*)

- 1a Перейти в потрібну папку, використовуючи команду *cd* (наприклад, `cd /d d:\py`).
- 1b. Визвати команду *jupyter notebook*


```
ca: C:\Windows\system32\cmd.exe - jupyter notebook
Microsoft Windows [Version 6.1.7601]
(c) Корпорация Майкрософт (Microsoft Corp.), 2009. Все права защищены.
C:\Users\User>cd /d d:\py
d:\py>jupyter notebook
[I 19:23:34.385 NotebookApp] The port 8888 is already in use, trying another port.
[I 19:23:34.387 NotebookApp] The port 8889 is already in use, trying another port.
[I 19:23:34.388 NotebookApp] The port 8890 is already in use, trying another port.
[I 19:23:34.389 NotebookApp] The port 8891 is already in use, trying another port.
[I 19:23:34.391 NotebookApp] The port 8892 is already in use, trying another port.
[I 19:23:34.392 NotebookApp] The port 8946 is already in use, trying another port.
[W 19:23:34.406 NotebookApp] Terminals not available (error was No module named
'winpty.cpython')
```

Список 2.

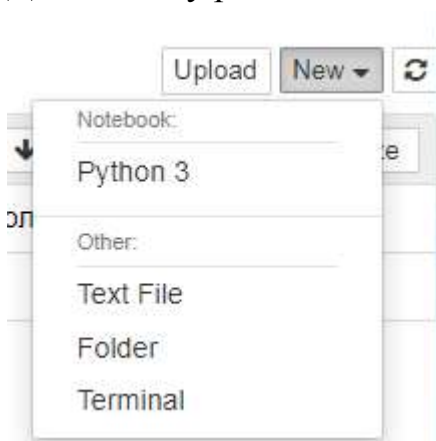
Ввести команду

```
jupyter notebook --notebook-dir="повний_шлях_до_папки"
```

Інші способи можна переглянути за посиланням <https://stackoverflow.com/questions/35254852/how-to-change-the-jupyter-start-up-folder>

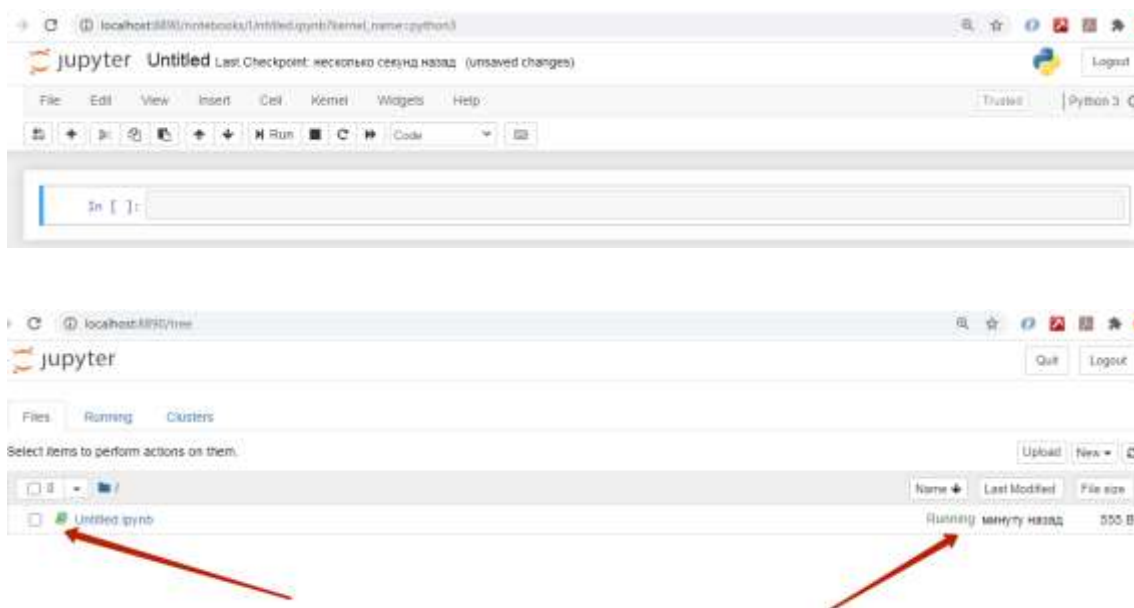
URL-адреса в адресному рядку схожа на `http://localhost:8888/tree`. Localhost не є веб-сайтом, але вказує, що контент обслуговується з вашого локального комп'ютера: з вашого власного комп'ютера. Ноутбуки і панель інструментів Jupyter - це веб-додатки, і Jupyter запускає локальний веб сервер Python для обслуговування цих додатків в веб-браузері, що робить його практично незалежним від платформи і відкриває можливості для більш зручного обміну в Інтернеті.

Для початку роботи натисніть на кнопку New і оберіть Python3



Ваш перший блокнот Jupyter відкриється в новій вкладці - кожен блокнот використовує свою вкладку, тому ви можете відкривати кілька блокнотів одночасно. Якщо ви перейдете назад на панель інструментів, ви

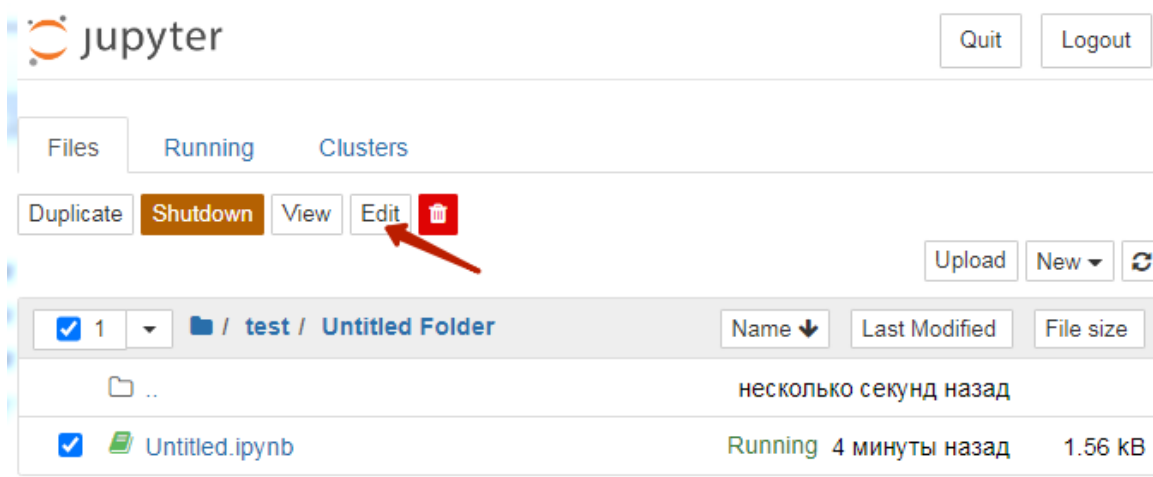
побачите новий файл `Untitled.ipynb` і зелений колір піктограми і тексту **Running**, який говорить про те, що ваш ноутбук працює.



Файл `ipynb`

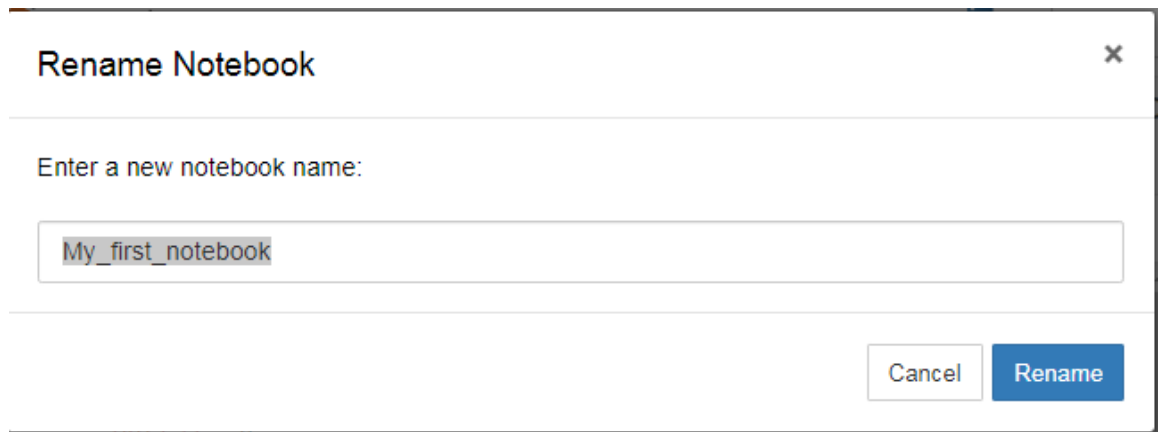
Буде корисно зрозуміти, що це за файл насправді. Кожен файл `.ipynb` є текстовим файлом, який описує вміст вашого блокноту у форматі JSON. Кожна комірка й її вміст, включаючи вкладення зображень, які були перетворені в рядки тексту, перераховані в ньому разом з деякими метаданими. Ви можете редагувати їх самостійно - якщо знаєте, що робите! – вибравши «**Edit**» **Edit Notebook Metadata** в рядку меню в блокноті.

Можна подивитися текст файлу в панелі навігації, натиснувши на кнопку **Edit**.



Перейменування файлу

Двічі клацнувши на назві файлу можна її змінити.



Ядра (kernels) та комірки (cells)

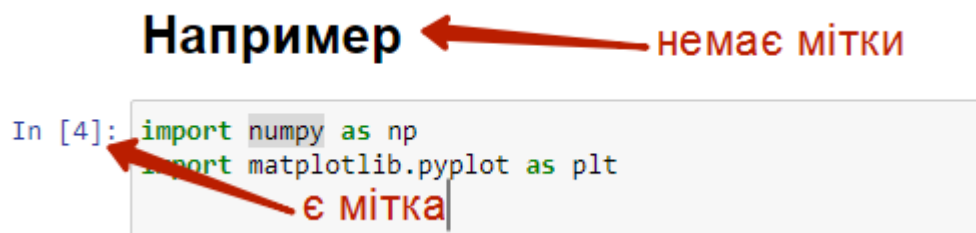
- kernel (ядро) - це «обчислювальний движок», який виконує код, що міститься в документі ноутбука.
- cell (комірка) - це контейнер для тексту, який буде відображатися в записнику, або код, який буде виконуватися ядром ноутбука.

Комірка (Cell)

Комірки утворюють структуру ноутбука. На скріншоті нового блокнота в наведеному вище розділі це поле з зеленим контуром. Є два основних типи комірок, які ми розглянемо:

- Комірка коду (Code) містить код, який повинен бути виконаний в ядрі, і відображає його виведення нижче.
- Комірка Markdown містить текст, відформатований з використанням Markdown, і відображає його висновок на місці під час запуску.

Коли ви запустите комірку коду, її висновок буде відображено нижче, а мітка зліва зміниться з `In []` на `In [1]`. Виведення осередку коду також є частиною документа. Ви завжди можете визначити різницю між кодом і комітками Markdown, тому що комірки коду мають цю мітку зліва, а комірки Markdown – не мають.



Частина «In» мітки просто короткий запис слова «Input», а номер мітки вказує, коли комірка була виконана в ядрі - в нашому випадку комірка була виконана першою. Запустіть комірку знову, і мітка зміниться на In [2], тому що тепер комірка була другою, запущеною в ядрі.

Введемо наступний код в комірку:

```
In [*]: import time
        time.sleep (3) # сон у 3 секунди
```

Ця комірка не виводить нічого, але для її виконання потрібно три секунди. При цьому Jupyter показує, що комірка в даний момент працює, змінивши мітку на In [*].

Як правило, дані у вихідні комірки надходять з будь-яких текстових даних, спеціально надрукованих під час виконання комірок, а також з значення останнього рядка в комірці, будь то змінна, виклик функції або щось ще. Наприклад:

```
In [9]: def say_hello(name):
        return 'Hello, {}'.format(name)
        say_hello('Vasya')
```

```
Out[9]: 'Hello, Vasya!'
```

Режими ноутбука

Блокноти Jupyter мають два різні режими EDIT і COMMAND.

Режим редагування:

Щоб увійти в режим EDIT, натисніть клавішу Enter на клавіатурі або клацніть клітинку. Режим редагування можна визначити за зеленою рамкою навколо комірки із зеленим лівим полем. Коли ви перебуваєте в режимі редагування, ви можете вводити текст в клітинки.

```
In [ ]:
```

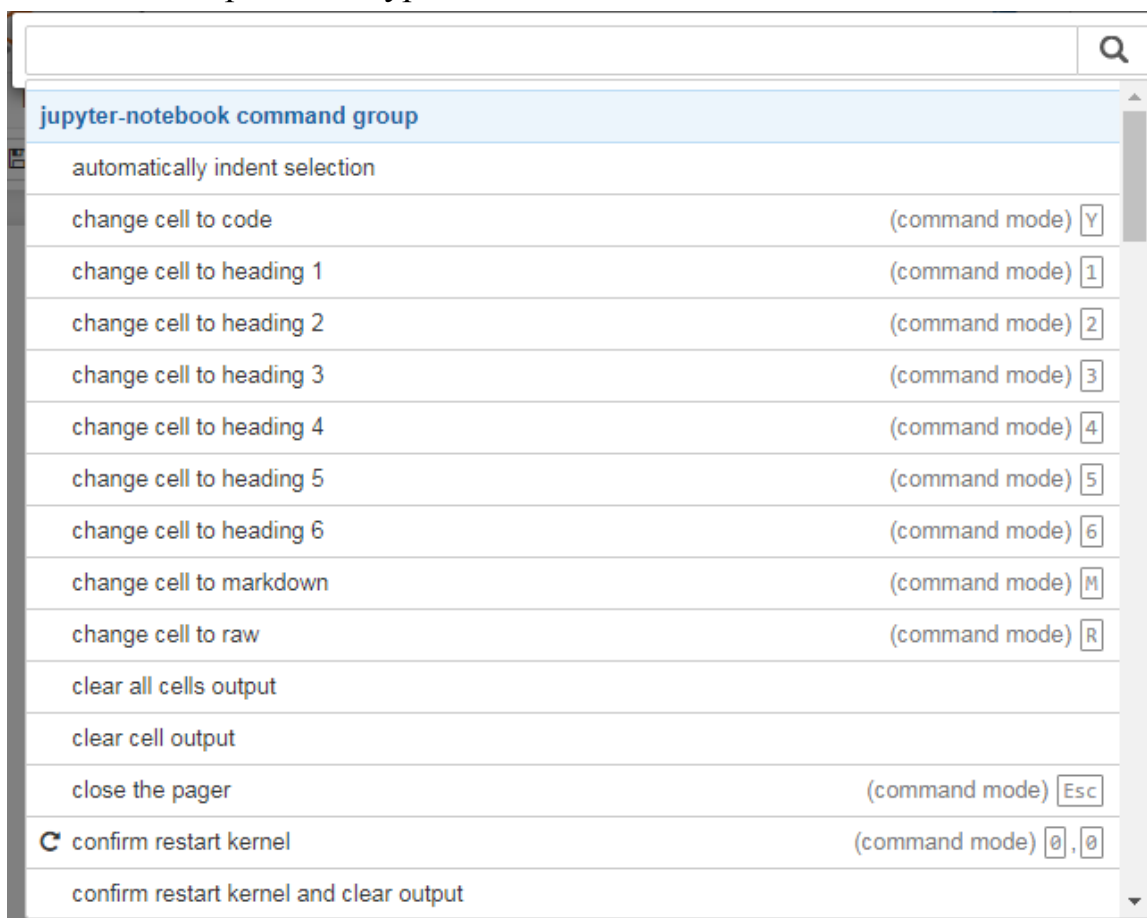
Командний режим:

Щоб увійти в режим COMMAND, натисніть Esc або клацніть де-небудь за межами комірки. Ви побачите сіру рамку навколо комірки із синім лівим краєм. Коли ви перебуваєте в командному режимі, ви можете редагувати свій блокнот, але не можете вводити клітинки.

```
In [ ]:
```

Палітра команд

Натиснувши піктограму з клавіатурою, або **Ctrl+Shift+P**, відкривається палітра клавіатурних команд.



Приклади команд

- Запустити вибрану комірку: **Ctrl + Enter**
- Запустити комірку та вставити нову комірку нижче: **Alt + Enter**
- Запустити комірку та перейти до клітинки нижче: **Shift + Enter**
- Вставте клітинку зверху: **Esc+A**
- Вставте комірку внизу: **Esc+B**
- Вирізати виділені клітинки: **Esc+X**
- Копіювати виділені комірки: **Esc+C**

Як поділитись блокнотом.

1. Спочатку очистити всі виведення **Cell>All output>Clear** та запустити ядро для всіх клітинок **Kernel>Restart & Run All**

2. Завантажити блокнот, наприклад на GitHub

3а. Ввести в адресний рядок https://colab.research.google.com/path_to_your_file_on_GitHub або

3б. Перейти на сайт nbviewer <https://nbviewer.jupyter.org/> і у відповідному полі ввести адресу репозиторію GitHub. У відкритому вікні репозиторію обрати необхідний файл.

Але nbviewer не виконує код блокнотів, а тільки відображає вхід та вихід збережених ноутбуків як веб-сторінку.

Для виконання коду ноутбуку використовується сторонній веб-сервіс Binder, що відкриває середовище виконання. Щоб завантажити ноутбук в mybinder.org необхідно натиснути кнопку *Execute on Binder* на панелі навігації.



Посилання на Jupyter Notebook з матеріалами до цієї лекції – https://github.com/svniko/python/blob/master/Lecture_1_ua.ipynb. Зверніть увагу, що не всі комірки `Out[]` відображаються (render) як слід.

Якщо відкрити цей же ноутбук в nbviewer, то все буде відображено вірно

– https://nbviewer.jupyter.org/github/svniko/python/blob/master/Lecture_1_ua.ipynb

Крім Binder код ноутбука, що завантажено на GitHub можна також відкрити в Google Colab. Для цього можна скористатися розширенням для Google Chrome Open in Colab, або ж просто додати спереду до адреси блокноту на GitHub адресу <https://colab.research.google.com/github/> шлях_до_блокноту_на_GitHub. Наприклад, щоб відкрити блокнот https://github.com/svniko/python/blob/master/Lecture_1_ua.ipynb в Google Colab, необхідно в адресному рядку ввести https://colab.research.google.com/github/svniko/python/blob/master/Lecture_1_ua.ipynb