

Комп'ютерна грамотність та інструменти для роботи з кодом

Інструменти аналізу даних та Data Science

1. Python і як можна запускати код на Python
2. Термінал та командний рядок
3. Файлова система
4. Interactive Python
 - a. Jupyter Notebook
 - b. Google Colab
 - c. Встановлення локально
 - d. Anaconda та Jupyter Lab
5. Віртуальні оточення - Virtual Envs

Основні аспекти Python

Python - це *високорівнева* мова програмування, яка відома своєю *простотою* і *читабельністю*. Основні аспекти про Python:

1. **Читабельний Синтаксис:** Python використовує простий і читабельний синтаксис. Він не вимагає від програміста використовувати складні конструкції або символи. Код написаний так, щоб його було легко читати і розуміти.
2. **Інтерпретована Мова:** Python є інтерпретованою мовою. Це означає, що код виконується покроково, і не потрібно його *компілювати*. Це спрощує процес написання та тестування коду.
3. **Бібліотеки та Фреймворки:** У Python є велика кількість бібліотек і фреймворків. Це готові до використання частини коду, які можна використовувати для вирішення різноманітних задач, від веб-розробки до аналізу даних.
4. **Об'єктно-Орієнтована Мова:** Python підтримує об'єктно-орієнтоване програмування, що дозволяє створювати класи та об'єкти, спрощуючи управління та організацію коду.
5. **Платформонезалежність:** Python можна використовувати на різних операційних системах. Це означає, що код, написаний на одній системі, буде працювати і на іншій без змін.

Як Python обробляє код

- **Написання коду:** Програміст пише код, використовуючи *синтаксис* Python.
- **Виконання коду:** Код виконується *інтерпретатором* Python покроково. Якщо в коді є помилки, інтерпретатор покаже повідомлення про помилку.
- **Виведення результату:** Якщо код написаний правильно, Python виконає його і покаже результат.

Приклад роботи Python

Уявімо, ви хочете написати програму, яка додає два числа. На Python це буде виглядати так:

```
# Вводимо два числа
```

```
a = 5
```

```
b = 10
```

```
# Додаємо числа
```

```
сума = a + b
```

```
# Виводимо результат
```

```
print("Сума:", сума)
```

Коли ви запустите цю програму, інтерпретатор Python прочитає та виконає код покроково і виведе результат "Сума: 15".

Синтаксис мови програмування

Ми пишемо код використовуючи синтаксис мови програмування.

Синтаксис мови програмування — це *правила* написання програм цією мовою. Це набір правил та конвенцій, які визначають, як програми, написані цією мовою, повинні бути структуровані та оформлені. Кожна мова програмування має свій власний унікальний синтаксис. Але є спільні принципи роботи мов програмування.

Як встановити Python

Офіційний сайт Python: <https://www.python.org/>

Для початку перевіримо, чи в нас вже є Python. Для цього відкриваємо Термінал (Mac/Linux) або командний рядок на Windows і вводимо команду `python --version` .

Як увійти в командний рядок:

https://tutorial.djangogirls.org/uk/intro_to_command_line/

Інструкція як встановити Python: https://tutorial.djangogirls.org/uk/python_installation/

Інший варіант

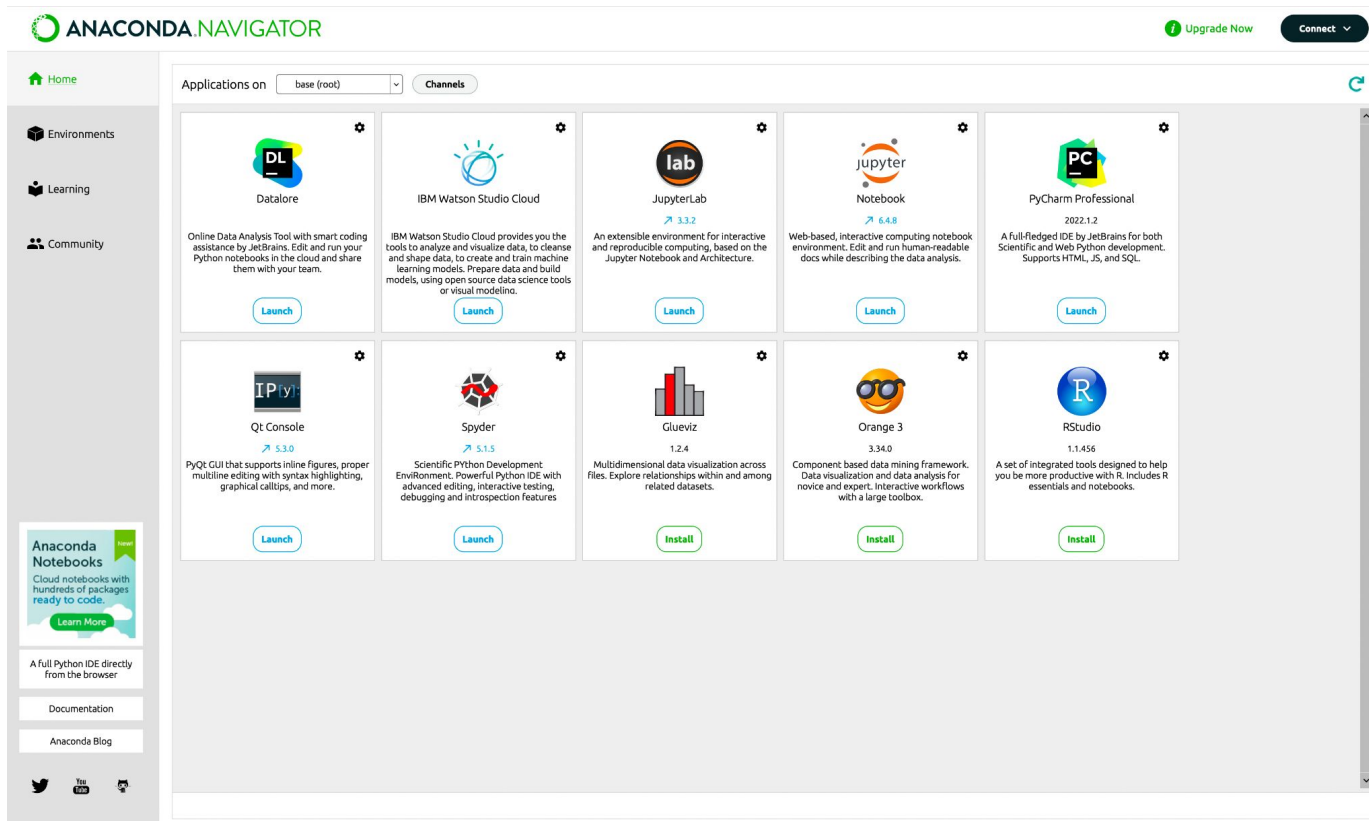
Можна встановити пакет [Anaconda](#).

Пакет = збірка програм.

Вона вже містить Python і багато корисних бібліотек для роботи. Також вона дозволить нам зручно керувати віртуальними оточеннями.

Як її встановити для різних ОС — є в документі: “Як встановити Anaconda.pdf” або за вказівками на [офіційному сайті](#) (взагалі завжди трохи краще використовувати інформацію з оф. сайту, адже там вона максимально актуальна).

Анаconda після встановлення



Домашнє завдання 1

Встановити Anaconda або Python на свій комп'ютер.

Якщо не вийшло, це не велика проблема і ви все одно зможете програмувати!

Далі розкажу, де і як.

Де виконувати код Python

Є кілька варіантів:

- командний рядок
- IDEs - integrated development environment. Найпопулярніші PyCharm і VSCode

Jupyter Notebook

Jupyter Notebook – середовище розробки, де одразу можна побачити результат виконання коду та його окремих фрагментів. На відміну від традиційного середовища розробки, код можна розбити на шматки та виконувати їх у довільному порядку.

Запуск – локально або на віддаленому сервері.

Jupyter Notebook

Підтримує наступні мови:

- Python
- R
- Perl
- Bash (команди до операційної системи для роботи з файловою системою, наприклад)
- Julia

Запустити код довільною мовою, не Python, – за допомогою `magic` (магічної) команди - відсоток.

Jupyter Notebook

Jupyter notebook - фактично це спосіб інтерактивної взаємодії з json-подібним файлом. Тобто це програмка, яка допомагає нам зручно працювати з певним типом файлу. І ця програмка є бібліотекою для Python. Бібліотека — набір функціонала.

Аби встановити jupyter notebook (якщо її немає), треба **після встановлення Python** виконати в командному рядку

```
pip install notebook
```

Аби перевірити наявність Jupyter notebook - в командному рядку:

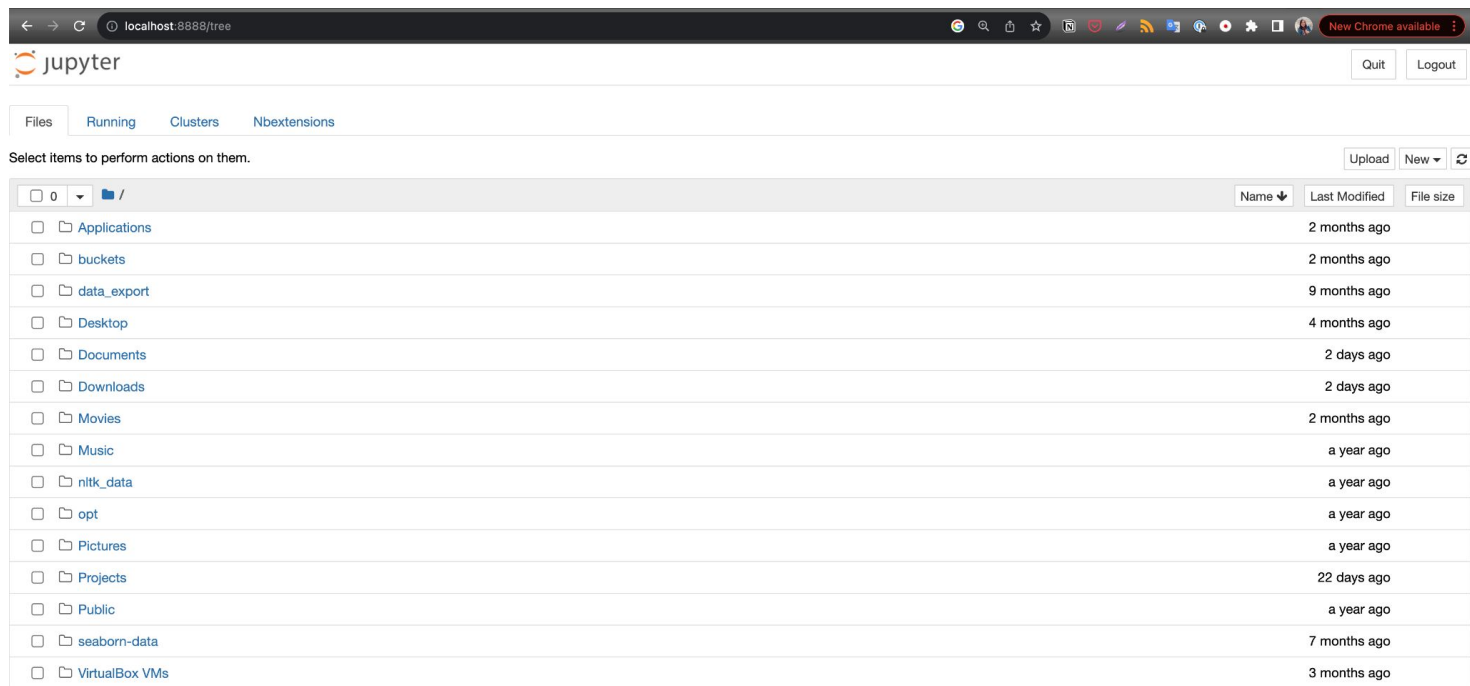
```
jupyter notebook
```

Після цього у вас має відкритись в браузері вікно з адресою <http://localhost:8888/tree>

Аби завершити роботу — в ком.рядку натисніть Ctrl+C і далі оберіть “yes”.

Вікно, яке у вас відкриється після запуску jupyter

Це — ваша файлова система. У кожного вона має свою структуру. Тож, у вас тут можуть бути інші файли і папки. У мене така, у вас може бути інша.



The screenshot shows the JupyterLab web interface in a browser window. The address bar indicates the URL is `localhost:8888/tree`. The JupyterLab logo is in the top left, and 'Quit' and 'Logout' buttons are in the top right. Below the logo are tabs for 'Files', 'Running', 'Clusters', and 'Nbextensions'. A message 'Select items to perform actions on them.' is displayed above a list of files and folders. The list has columns for 'Name', 'Last Modified', and 'File size'. The files and folders listed are:

Name	Last Modified	File size
0		
/		
Applications	2 months ago	
buckets	2 months ago	
data_export	9 months ago	
Desktop	4 months ago	
Documents	2 days ago	
Downloads	2 days ago	
Movies	2 months ago	
Music	a year ago	
nlTK_data	a year ago	
opt	a year ago	
Pictures	a year ago	
Projects	22 days ago	
Public	a year ago	
seaborn-data	7 months ago	
VirtualBox VMs	3 months ago	

Файли і папки в комп'ютері

Кожен файл має свій шлях. Аби відкрити файл з Python ми маємо прописати шлях до файлу.

Шлях може бути *абсолютним* і *реферальним* (відносним).

Коли ми будемо працювати з файлами в курсі — у мене вони лежать в одному місці, а у вас можуть лежати в іншому, в залежності від того, як ви організували свою *файлову систему*.

З файловою системою ми можемо працювати як з *графічного інтерфейсу* (коли ми бачимо папки), так і з командного рядка чи з jupyter notebook.

До речі, команди для роботи з файловою системою з командного рядка для різних ОС зібрані в документі “Основні команди в Bash та PowerShell”.

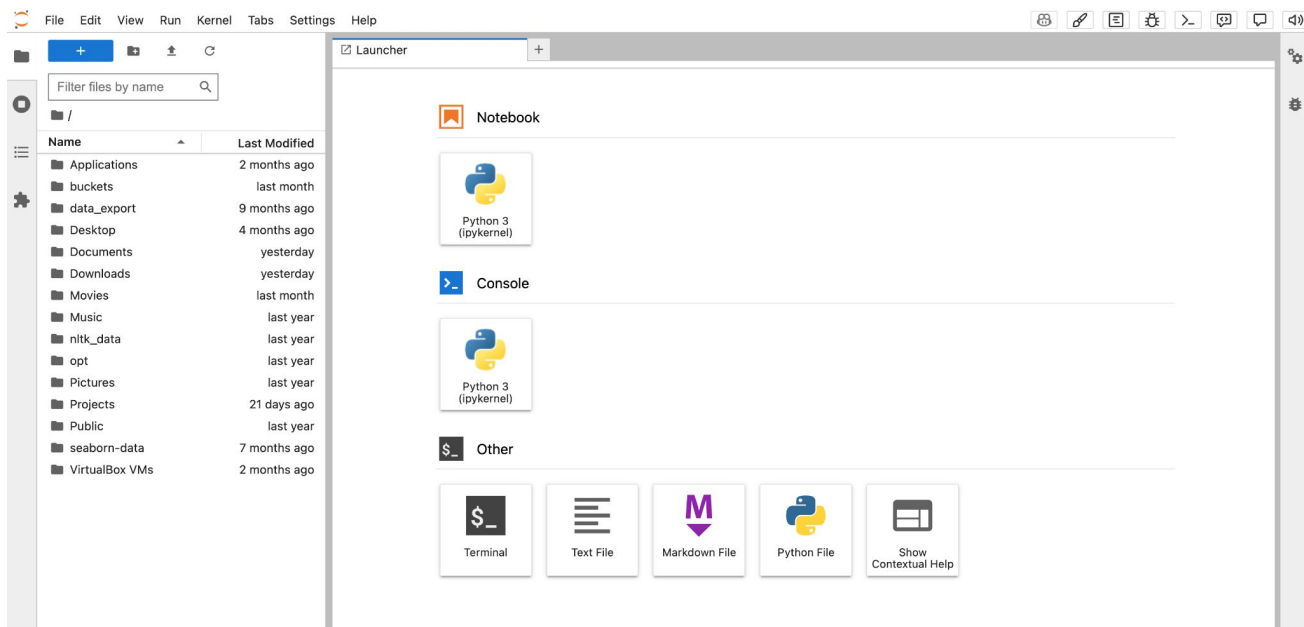
Що таке це `pip`

`pip` - це інструмент для встановлення та керування пакетами Python, який дозволяє користувачам легко встановлювати, оновлювати та видаляти бібліотеки та інші ресурси мови програмування Python.

Всі бібліотеки ми будемо встановлювати через `pip`.

Jupyter Lab

Це jupyter notebook з розширеним функціоналом. Можна працювати в ньому на курсі та в роботі. На мій погляд, це навіть зручніше. Але в наших лекціях я працюю в jupyter notebook - jFYI.



Як встановити Jupyter Lab

Після встановлення Python виконати в командному рядку

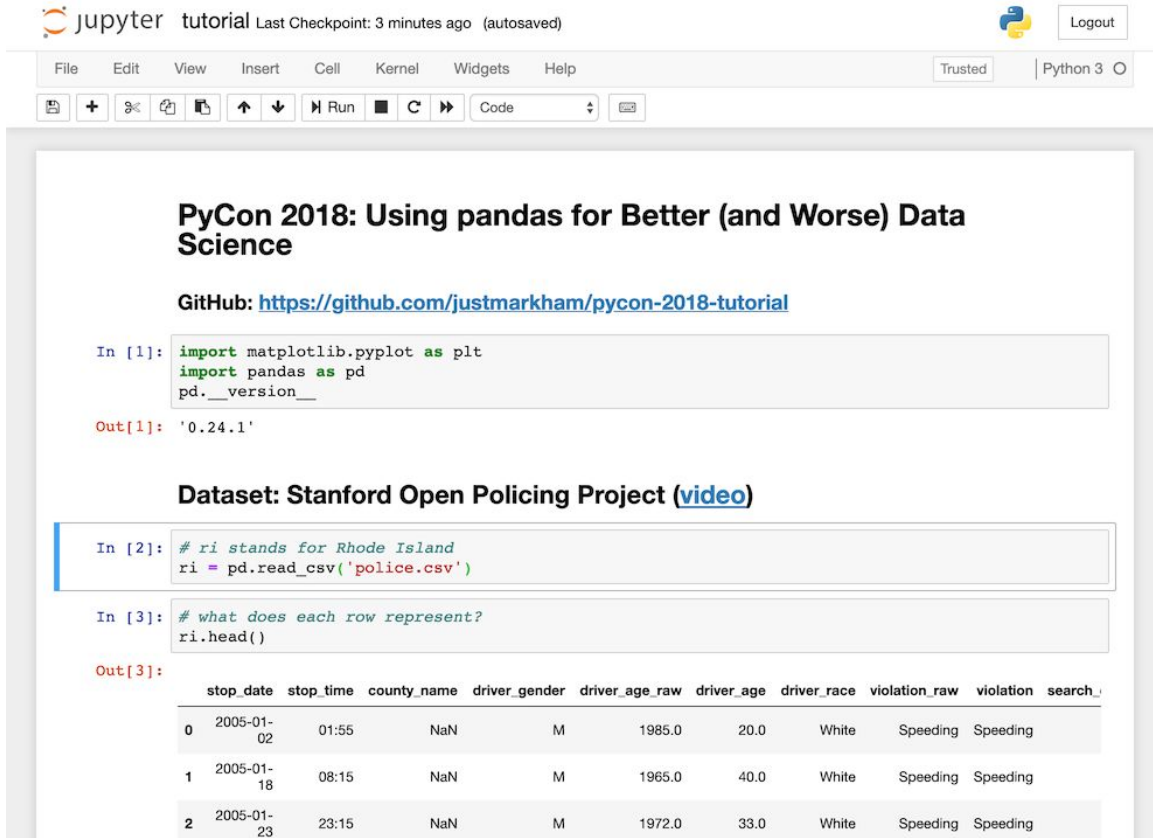
```
pip install jupyterlab
```

Як працює Jupyter notebook локально

Демо структури середовища та функціоналу, гарячих клавіш: [Lecture 0. Jupyter Demo.ipynb](#)

Важливо! В мене встановлені jupyter notebook extensions. Це бібліотека для розширення функціонала jupyter notebook. Вона не підтримується: тобто баги (помилки) не фіксяться (не лагодяться) девелоперами (розробниками цієї програми). Встановлення цих розширень опціональне і зовсім необов'язкове для проходження курсу. Але якщо хочете встановити - розкажу як.

Jupyter Notebook 6e3 Extensions



PyCon 2018: Using pandas for Better (and Worse) Data Science

GitHub: <https://github.com/justmarkham/pycon-2018-tutorial>

```
In [1]: import matplotlib.pyplot as plt
import pandas as pd
pd.__version__
```

```
Out[1]: '0.24.1'
```

Dataset: Stanford Open Policing Project ([video](#))

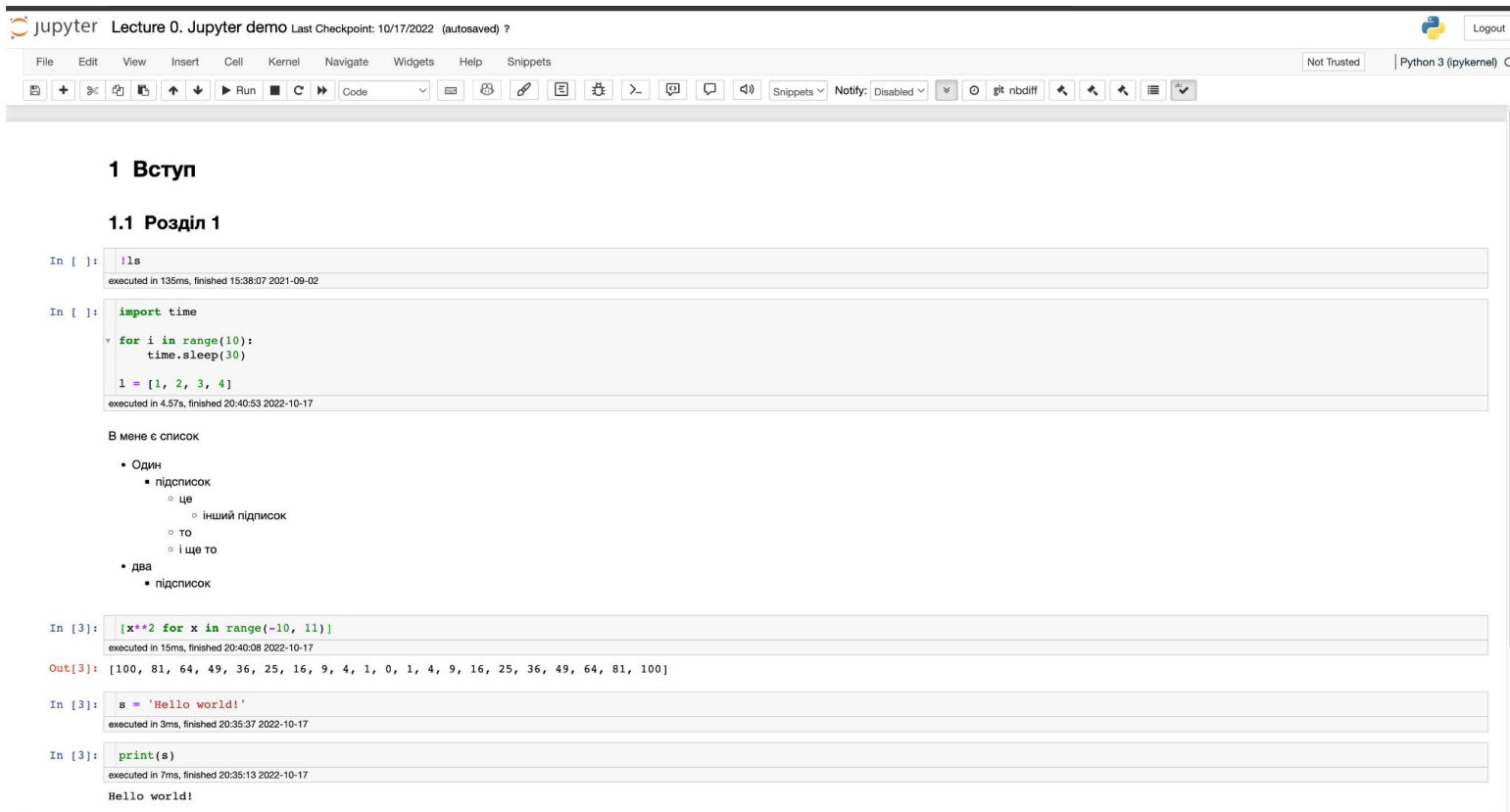
```
In [2]: # ri stands for Rhode Island
ri = pd.read_csv('police.csv')
```

```
In [3]: # what does each row represent?
ri.head()
```

```
Out[3]:
```

	stop_date	stop_time	county_name	driver_gender	driver_age_raw	driver_age	driver_race	violation_raw	violation	search_
0	2005-01-02	01:55	NaN	M	1985.0	20.0	White	Speeding	Speeding	
1	2005-01-18	08:15	NaN	M	1965.0	40.0	White	Speeding	Speeding	
2	2005-01-23	23:15	NaN	M	1972.0	33.0	White	Speeding	Speeding	

Jupyter Notebook 3 Extensions



The screenshot shows a Jupyter Notebook interface with the following components:

- Header:** "jupyter Lecture 0. Jupyter demo Last Checkpoint: 10/17/2022 (autosaved) ?" and a "Logout" button.
- Menu Bar:** File, Edit, View, Insert, Cell, Kernel, Navigate, Widgets, Help, Snippets.
- Toolbar:** Includes icons for file operations, running cells, and a status bar showing "Not Trusted" and "Python 3 (pykernel)".
- Section 1: Вступ**
 - Section 1.1: Розділ 1**
 - Code Cell 1:** `ls`. Output: "executed in 135ms, finished 15:38:07 2021-09-02".
 - Code Cell 2:** `import time`, `for i in range(10):`, `time.sleep(30)`, `l = [1, 2, 3, 4]`. Output: "executed in 4.57s, finished 20:40:53 2022-10-17".
 - Text:** "В мене є список"
 - List:**
 - Один
 - підписок
 - це
 - інший підписок
 - то
 - і ще то
 - два
 - підписок
 - Code Cell 3:** `[x**2 for x in range(-10, 11)]`. Output: "executed in 15ms, finished 20:40:08 2022-10-17".
 - Output:** `[100, 81, 64, 49, 36, 25, 16, 9, 4, 1, 0, 1, 4, 9, 16, 25, 36, 49, 64, 81, 100]`
 - Code Cell 4:** `s = 'Hello world!'`. Output: "executed in 3ms, finished 20:35:37 2022-10-17".
 - Code Cell 5:** `print(s)`. Output: "executed in 7ms, finished 20:35:13 2022-10-17".
 - Text:** "Hello world!"

Jupyter Notebook Extensions: встановлення 1/3

Встановлення [Jupyter Notebook Extensions](#) - в терміналі/консолі наступний набір команд:

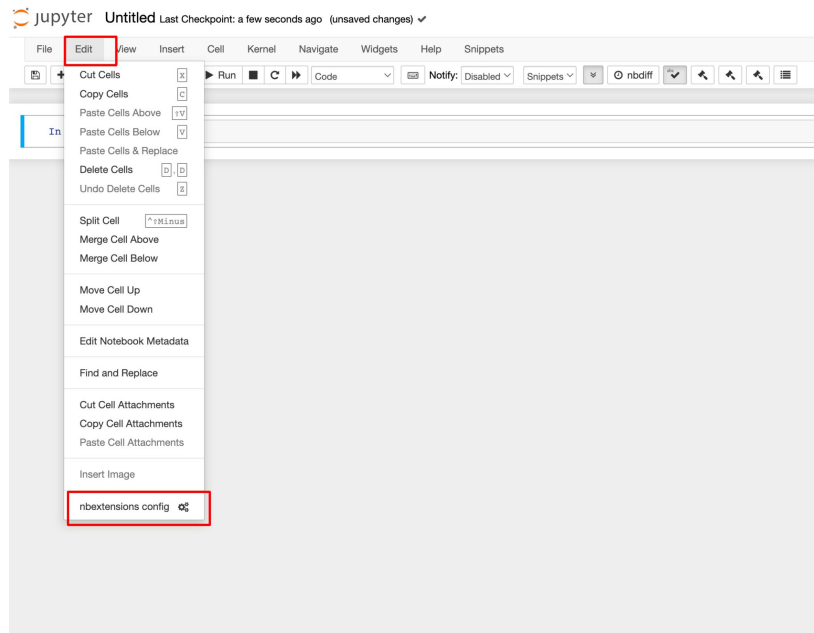
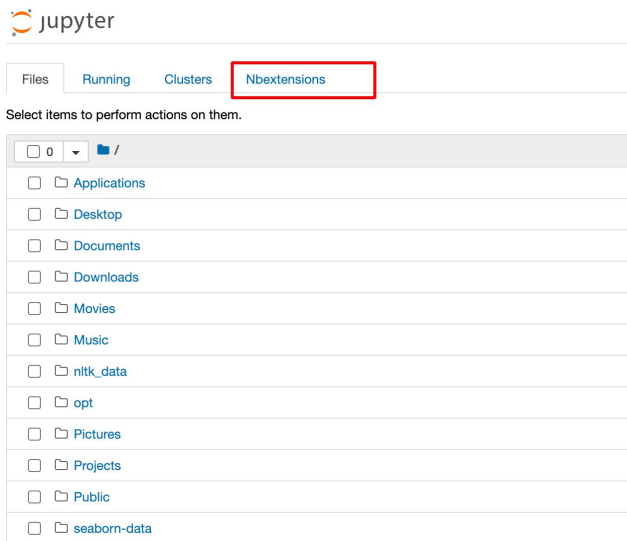
```
pip install jupyter_contrib_nbextensions  
jupyter contrib nbextension install
```

Далі запускаємо:

```
jupyter notebook
```

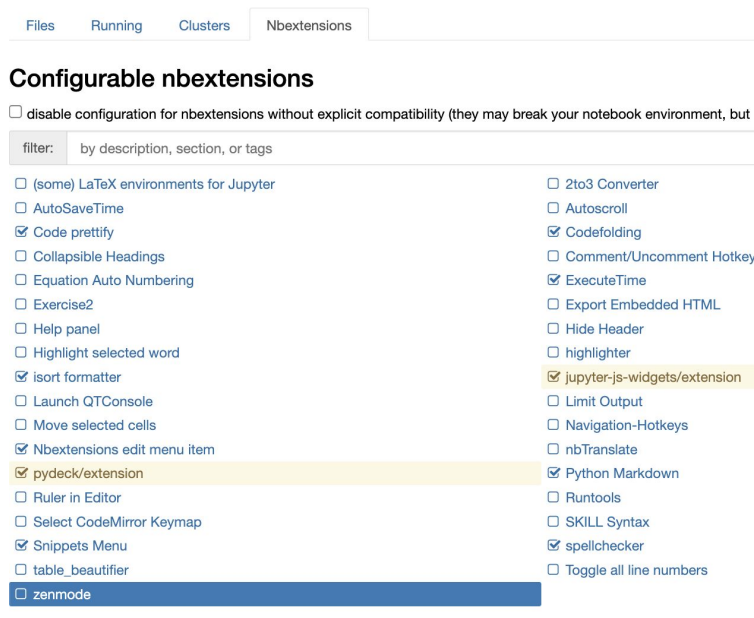
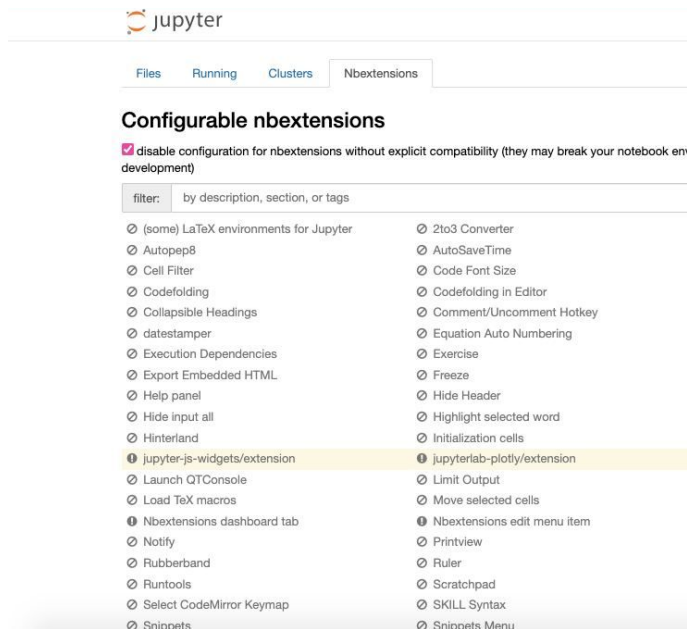
Jupyter Notebook Extensions: встановлення 2/3

Має з'явитись Nbextensions розділ меню (лівий скріншот) або ж створити новий .ipynb та перейти там в Edit -> nbextensions config (правий скріншот)



Jupyter Notebook Extensions: встановлення 3/3

В Configurable nbextensions треба **зняти** (вимкнути) галочку біля “disable configuration for ...” і все - ви можете налаштовувати ті розширення, які хочете мати.



Мої налаштування Nbextensions

Configurable nbextensions

☐ disable configuration for nbextensions without explicit compatibility (they may break your notebook environment, but can be useful to show for nbextension development)

filter: by description, section, or tags			
<input type="checkbox"/> (some) LaTeX environments for Jupyter	<input type="checkbox"/> 2to3 Converter	<input type="checkbox"/> AddBefore	<input checked="" type="checkbox"/> Autopep8
<input type="checkbox"/> AutoSaveTime	<input type="checkbox"/> Autoscroll	<input type="checkbox"/> Cell Filter	<input type="checkbox"/> Code Font Size
<input checked="" type="checkbox"/> Code prettify	<input checked="" type="checkbox"/> Codefolding	<input type="checkbox"/> Codefolding in Editor	<input type="checkbox"/> CodeMirror mode extensions
<input type="checkbox"/> Collapsible Headings	<input type="checkbox"/> Comment/Uncomment Hotkey	<input checked="" type="checkbox"/> contrib_nbextensions_help_item	<input type="checkbox"/> daterstamp
<input type="checkbox"/> Equation Auto Numbering	<input checked="" type="checkbox"/> ExecuteTime	<input type="checkbox"/> Execution Dependencies	<input type="checkbox"/> Exercise
<input type="checkbox"/> Exercise2	<input type="checkbox"/> Export Embedded HTML	<input type="checkbox"/> Freeze	<input type="checkbox"/> Gist-it
<input type="checkbox"/> Help panel	<input type="checkbox"/> Hide Header	<input type="checkbox"/> Hide input	<input type="checkbox"/> Hide input all
<input type="checkbox"/> Highlight selected word	<input type="checkbox"/> highlighter	<input type="checkbox"/> Hinterland	<input type="checkbox"/> Initialization cells
<input checked="" type="checkbox"/> isort formatter	<input checked="" type="checkbox"/> jupyter-js-widgets/extension	<input checked="" type="checkbox"/> jupyterlab-plotly/extension	<input type="checkbox"/> Keyboard shortcut editor
<input type="checkbox"/> Launch QTConsole	<input type="checkbox"/> Limit Output	<input type="checkbox"/> Live Markdown Preview	<input type="checkbox"/> Load TeX macros
<input type="checkbox"/> Move selected cells	<input type="checkbox"/> Navigation/Hotkeys	<input checked="" type="checkbox"/> nbdtme/index	<input checked="" type="checkbox"/> Nbextensions dashboard tab
<input checked="" type="checkbox"/> Nbextensions edit menu item	<input type="checkbox"/> nbTranslate	<input checked="" type="checkbox"/> Notify	<input type="checkbox"/> Printview
<input checked="" type="checkbox"/> pydeck/extension	<input checked="" type="checkbox"/> Python Markdown	<input type="checkbox"/> Rubberband	<input type="checkbox"/> Ruler
<input type="checkbox"/> Ruler in Editor	<input type="checkbox"/> Runtools	<input type="checkbox"/> Scratchpad	<input checked="" type="checkbox"/> ScrollDown
<input type="checkbox"/> Select CodeMirror Keymap	<input type="checkbox"/> SKILL Syntax	<input type="checkbox"/> Skip-Traceback	<input checked="" type="checkbox"/> Snippets
<input checked="" type="checkbox"/> Snippets Menu	<input checked="" type="checkbox"/> spellchecker	<input type="checkbox"/> Split Cells Notebook	<input checked="" type="checkbox"/> Table of Contents (2)
<input type="checkbox"/> table_beautifier	<input type="checkbox"/> Toggle all line numbers	<input type="checkbox"/> Tree Filter	<input type="checkbox"/> Variable Inspector
<input type="checkbox"/> zenmode			

Google Colab = Jupyter Notebook на хмарних сервісах

Доступ – через Google Account.

Має вже встановлені бібліотеки (більшість, які нам необхідні) та GPU (потрібно для важких обчислень).

Є обмеження: до 12 годин безперервної роботи.

Google Colab = Jupyter Notebook на хмарних сервісах

Демо функціонала:

https://colab.research.google.com/notebooks/basic_features_overview.ipynb

- + які бібліотеки встановлені можемо перевірити через (!pip freeze)
- + як створити ноутбук у Google Drive

Virtual environments

Віртуальне оточення необхідне, аби створити різне оточення для різних програм. До того ж, це — безпека Python, встановленого у вас на комп'ютері. Зручно керувати вірт. оточеннями з Anaconda. Якщо вона у Вас встановлена, то працюємо з вірт. оточеннями наступним чином.

Створити вірт. оточення з певною версією Python

```
conda create --name py312 python==3.12
```

Зайти в це оточення

```
conda activate py312
```

Вийти з цього оточення

```
conda deactivate
```

Як працювати з вірт. оточеннями без Anaconda

Інструкція тут: <https://docs.python.org/uk/3/tutorial/venv.html>

Домашнє завдання 2

Запустити свій перший jupyter notebook.

Програма максимум — це спробувати попрацювати з jupyter notebook в різних оточеннях.