


ЗВІТ З ЛАБОРАТОРНОЇ РОБОТИ №1  
за курсом "Програмування на Python"  
студента/студентки групи ПА-  
HarryJamesPotter/HermioneJeanGranger  
кафедра комп'ютерних технологій, ДНУ  
2025/2026

Тема: «Налаштування програмного середовища для запуску першої програми»


Для розробки програм на мові Python, необхідна установка додаткового програмного забезпечення, а саме редактора коду, компілятора або інтерпретатора, відладчика та іншого програмного забезпечення.

Існує кілька варіантів розробки програм на Python. Розглянемо три основні підходи, а саме:

- PyCharm  IDE;
- Jupyter Notebook (Anaconda Navigator);
- текстовий редактор (notepad++, sublime, visual studio code) та інтерпретатор Python.
- Або онлайн інтерпретатори (replit.com, onecompiler.com/python та інші)

Усі програми, описані вище, є безкоштовними, а PyCharm IDE має версію Community. Далі розглянемо середовища розробки, а також їх переваги та недоліки. Найкращий варіант, встановити всі три варіанти, а потім вибрати, який буде найзручнішим у використанні.

### ***PyCharm IDE.***

PyCharm  IDE – інтегроване середовище розробки для мови програмування Python. Надає засоби для аналізу коду, графічний відладчик, інструмент для запуску юніт-тестів та підтримує веб-розробку на Django. PyCharm розроблена компанією JetBrains на основі IntelliJ IDEA. Зазвичай IDE має текстовий редактор, компілятор або інтерпретатор, відладчик та інше програмне забезпечення. IDE дозволяє збільшити швидкість розробки (за умови попереднього навчання роботі з IDE). Безкоштовна версія має відкритий вихідний код і розповсюджується під ліцензією Apache 2. Це полегшене середовище, яке підходить для розробки лише на Python. PyCharm дозволяє швидко проводити рефакторинг коду, а також використовувати зручний графічний відладчик. Утиліта підтримує всі нові версії Django, а також IronPython, Jython, Cython, PyPy wxPython, PyQt, PyGTK та багато інших інструментів. У PyCharm можна проводити інтегроване Unit тестування, використовувати інтерактивні консолі для Python, Django, SSH, відладчика та баз даних. PyCharm має велику колекцію плагінів, і його можна

використовувати у зв'язці з різними трекерами на зразок JIRA, Youtrack, Lighthouse, Redmine, Trac та інших. PyCharm крос-платформне середовище розробки: можна використовувати на Linux, Windows та Mac OS.

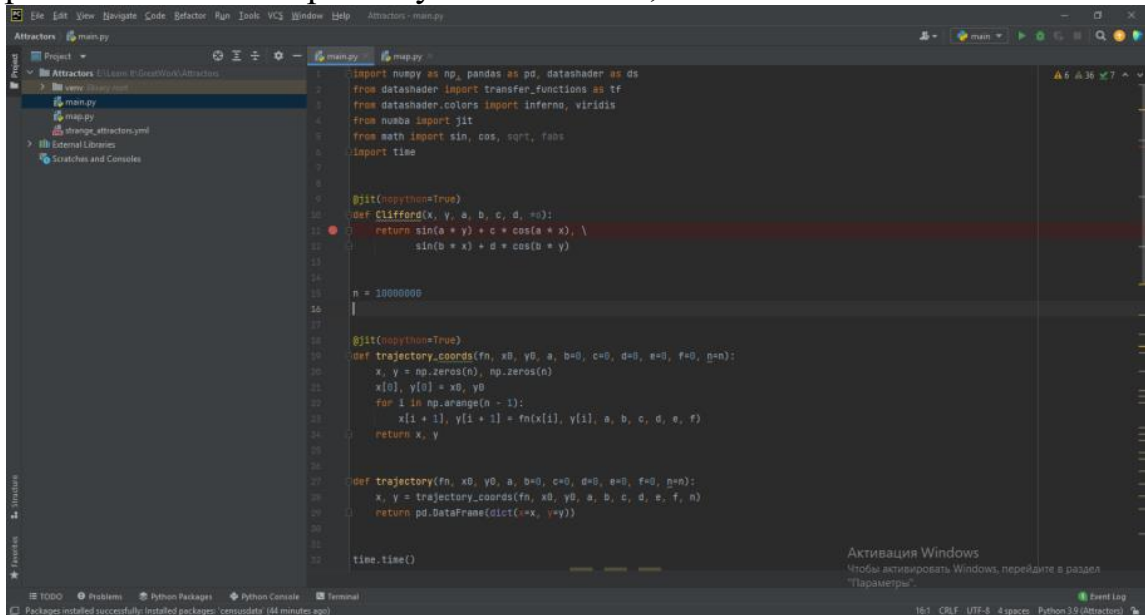


Рис. 1.1. Інтерфейс IDE PyCharm.

Основні переваги:

- можливість перегляду всього вихідного коду одним клацанням миші;
- безліч плагінів;
- автодоповнення коду (у тому числі django-шаблони, на порядок краще за аналоги, так само хороше автодоповнення js, css, html і т.д.);
- підтримка спільноти;
- просте встановлення;
- хороші можливості для налагодження програми;
- PyCharm крос-платформне середовище розробки: можна використовувати на Linux, Windows та Mac OS.

Основні недоліки:

- може виникнути проблема при спробі виправити такі інструменти, як venv;
- переналаштування компілятора-інтерпретатор може вимагати масу зусиль;
- ресурсомісткий - PyCharm займає досить багато місця на диску, а також написаний на Java, а отже, вимагає багато ресурсів оперативної пам'яті.

## Використання Jupyter Notebook за допомогою Anaconda Navigator.

Jupyter Notebook – це середовище розробки, де можна бачити результат виконання коду та її окремих фрагментів. На відміну від традиційного середовища розробки, код можна розбити на частини та виконувати їх у довільному порядку. У такому середовищі розробки можна, наприклад, написати функцію та одразу перевірити її роботу, без запуску програми повністю. А ще можна змінити порядок виконання коду. Можна окремо завантажити файл у пам'ять, окремо перевірити вміст. А ще в jupyter-ноутбуках є виведення результату одразу після фрагмента коду. Наприклад, можна всередині коду побачити побудований графік, отримати попередні цифри або будь-яку іншу візуалізацію.

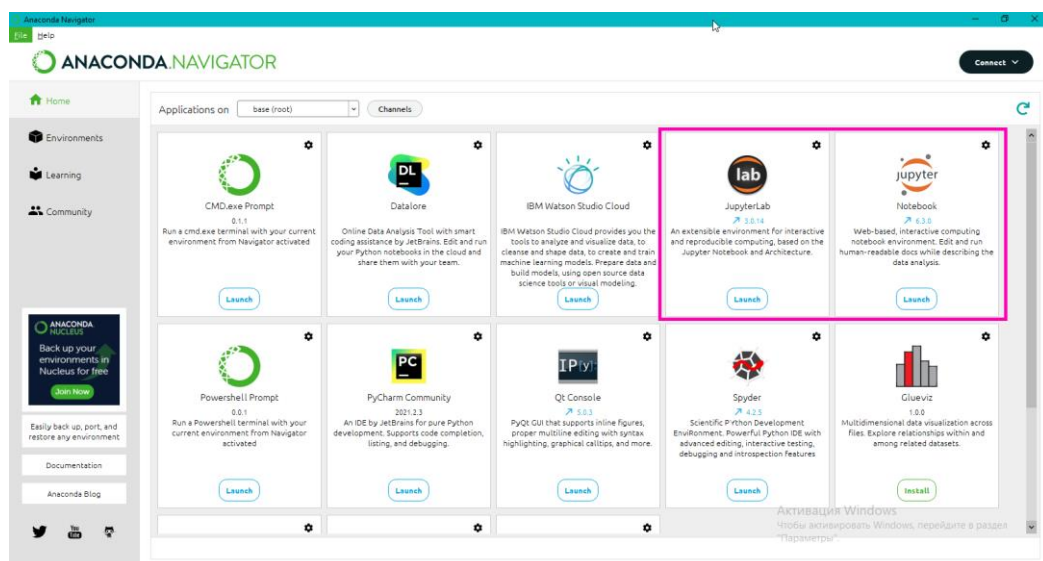


Рис. 1.2. Інтерфейс Anaconda Navigator.

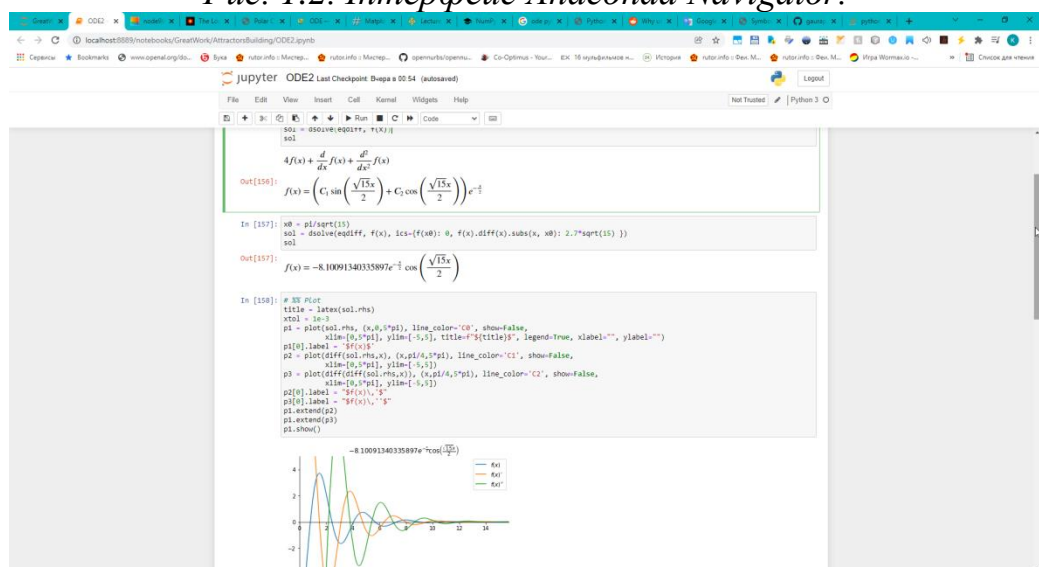


Рис. 1.3: Інтерфейс Jupyter Notebook.

Основні переваги Jupyter Notebook (Anaconda Navigator):

- Jupyter Notebooks допускає поетапне виконання блоків коду, деякі програмісти вважають вигідним, оскільки дозволяє зручно тестувати;





- Jupyter Notebooks дозволяє використовувати Matplotlib inline (%matplotlib inline), який виводить Matplotlib у Jupyter Notebook і дає можливість інтерактивно керувати графіками, що зручно та дозволяє швидше аналізувати результати.
- велика підтримка спільноти;
- просте встановлення;
- найчастіше, коли говорять про jupyter-ноутбуки, мають на увазі роботу з Python. Але насправді можна працювати з іншими мовами: Ruby, Perl, R, bash-скрипти;
- Jupyter Jupyter кросс-платформенная среда разработки: можно использовать на Linux, Windows и Mac OS.

Основні недоліки:

- ще один інструмент, який добре мати у вашому наборі інструментів для розробників, але, ймовірно, не найкращий інструмент для розробки складної програми за межами галузі науки про дані.

### ***Використання текстового редактора (Notepad++, Geany, Visual Studio Code, Sublime) та інтерпретатора Python.***

Для цього способу розробки, потрібен один із наведених нижче текстових редакторів і встановлений окремо з офіційного сайту інтерпретатор мови Python.

- Notepad++  — вільний текстовий редактор з відкритим вихідним кодом для Windows з підсвічуванням синтаксису, розмітки та мов опису апаратури VHDL і Verilog. Базується на компоненті Scintilla, написаному на C++ з використанням STL, а також Windows API, і розповсюджується під ліцензією GNU General Public License. Базова функціональність програми може бути розширена як за рахунок плагінів, так і сторонніх модулів, таких як компілятори та препроцесори. Підтримує понад 100 форматів;
- Geany  — середовище розробки програмного забезпечення, написане за допомогою бібліотеки GTK+. Доступна для наступних операційних систем: BSD, GNU/Linux, Mac OS X, Solaris та Windows. Geany поширюється згідно з GNU General Public License;
- Visual Studio Code  — кроссплатформенний редактор коду від Microsoft з відкритим вихідним кодом;
- Sublime Text  — кроссплатформенний текстовий редактор на C++ та Python. Спочатку розроблявся як розширення для Vim. Починаючи з

версії 2.0, підтримує 44 основні мови програмування, включаючи Python. Його головні принципи: мінімалістичний інтерфейс та акцент на коді. Закритий вихідний код, потрібна ліцензія, хоча є безкоштовна версія.

Основні переваги використання текстового редактора та інтерпретатор Python:

- висока продуктивність;
- велика кількість розширень та плагінів;
- простий інтерфейс;
- легкий – дані варіант займає найменше місця на диску та в оперативній пам'яті;
- підтримує багато мов програмування.

Основні недоліки:

- Складність налаштування ПЗ для розробників-початківців.

**Інші IDE:**

<https://www.spyder-ide.org/>

Постановка задачі:

- 1) Обрати IDE
- 2) Встановити інтерпретатор мови Python
- 3) Налаштувати IDE для запуску першої програми
- 4) Написати просту програму «Hello World»
- 5) Додати до програми «Hello World» можливість запитувати в користувача ім'я та вітати користувача
- 6) Вивести та проаналізувати розміри базових типів у мові Python
- 7) Навчитися вгадувати код (break points, debug, print, printf, alert – будь-який з цих варіантів)