

Інтерактивний комп'ютинг з Jupyter Notebook

Короткий огляд

Jupyter notebook – графічна вебоболонка для IPython¹ для інтерактивних обчислень та спільної розробки додатків (IPython підтримує паралельні підключення до одного обчислювального ядра).

Особливості Jupyter notebook:

- комплексна інтроспекція об'єктів;
- збереження історії введення у сеансах, логування сесії;
- кешування вихідних результатів;
- графічне представлення результатів;
- підтримка мови розміток Markdown і LaTeX;
- розширювана система магічних команд.

Компоненти Jupyter notebook:

- браузерний вебзастосунок;
- ноутбуки – файли для роботи з кодом програми.

Ноутбук – файл, у якому зберігається код та отримані в межах сесії вхідні та вихідні дані. Він є документованим записом роботи з можливістю багаторазового запуску частин коду. Ноутбук можна експортувати в формати PDF і HTML.

Встановлення та запуск

Jupyter Notebook входить до складу Python-дистрибутиву **Anaconda**. Процес встановлення та запуску описаний на офіційному сайті anaconda.com та на багатьох інших онлайн-ресурсах.

Основи роботи в оболонці

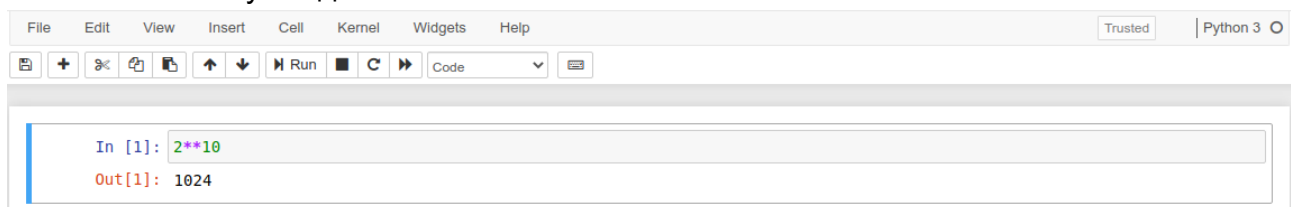
¹ IPython – інтерактивна оболонка мови програмування Python



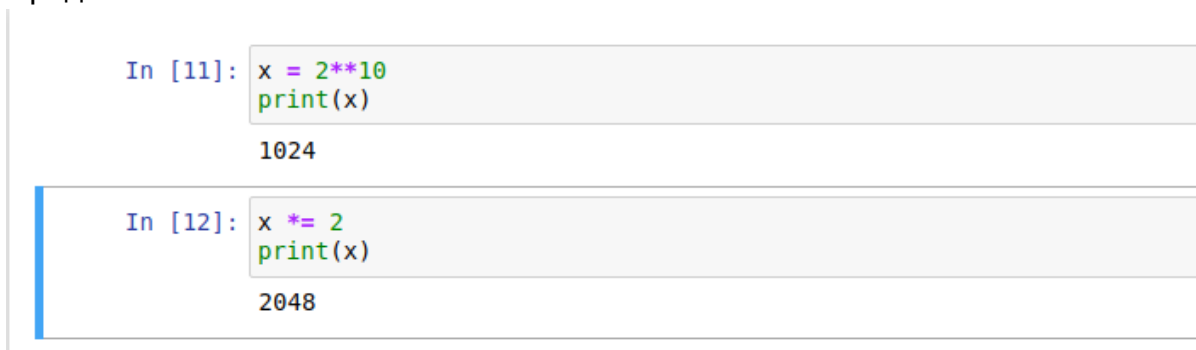
Створення ноутбука (текстового файлу, папки, вікна командної оболонки):



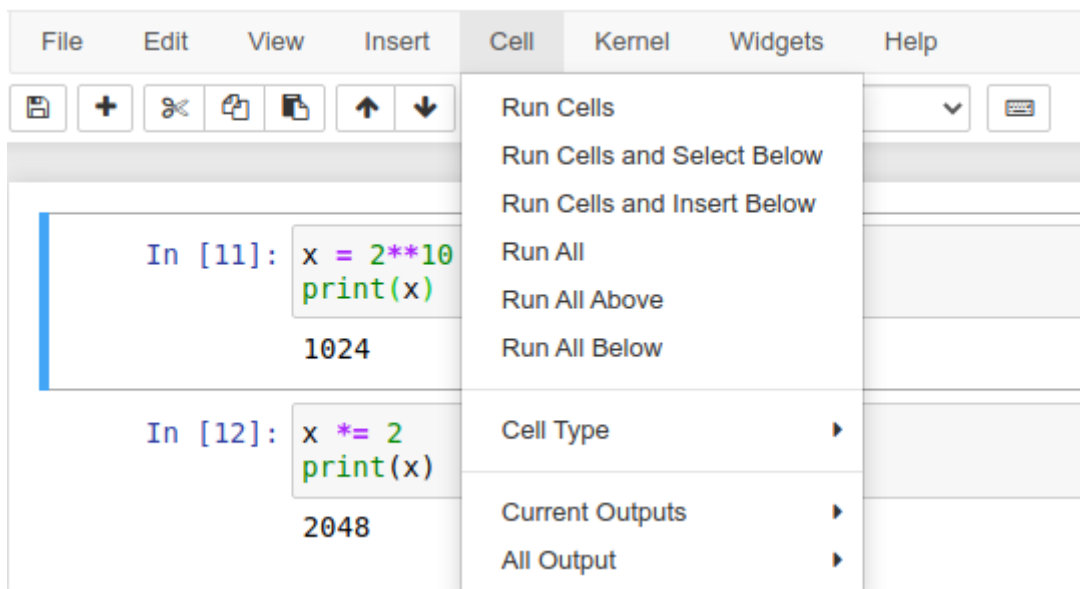
Ctrl+Enter виконує код:



Shift+Enter виконує код і створює нову комірку, у якій можна використовувати дані попередньої:



Керування комірками:



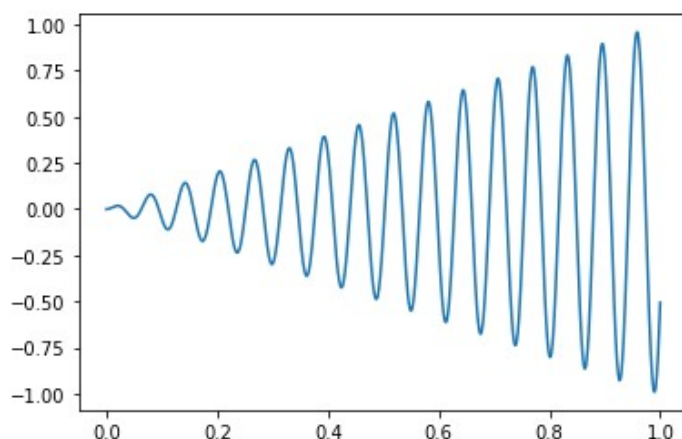
Виведення зображень

За замовчуванням зображення не виводяться у ноутбучі. Для їх відображення потрібно виконати спеціальну команду, яка починається символом %. Наприклад, для бібліотеки `matplotlib`:

```
In [6]: %matplotlib inline
from matplotlib import pyplot as plt
from math import sin

x = [i/1000 for i in range(1001)]
y = [i*sin(100*i) for i in x]

plt.plot(x, y)
plt.show()
```



“Магічні” команди

Список доступних команд:

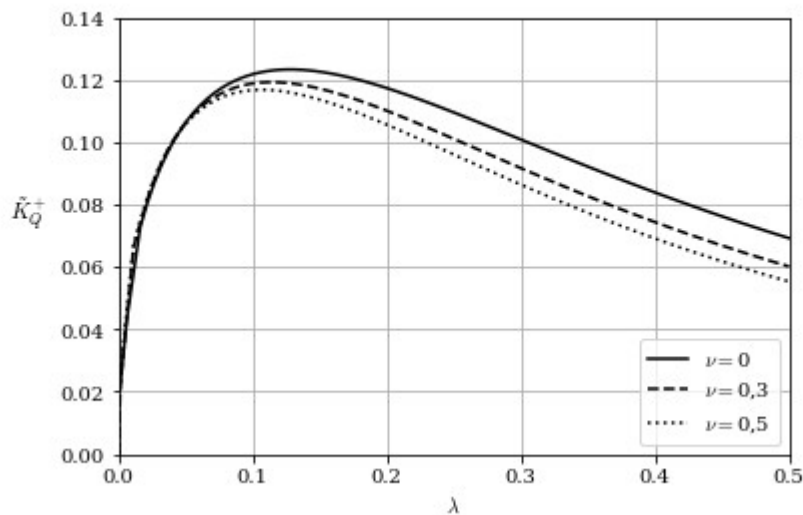
```
In [2]: %lsmagic
Out[2]: Available line magics:
%alias %alias_magic %autoawait %autocall %automagic %autosave %bookmark %cat %cd %clear %colors %conda
%config %connect_info %cp %debug %dhist %dirs %doctest_mode %ed %edit %env %gui %hist %history %killb
%gscripts %ldir %less %lf %lk %ll %load %load_ext %loadpy %logoff %logon %logstart %logstate %logstop
%ls %lsmagic %lx %macro %magic %man %matplotlib %mkdir %more %mv %notebook %page %pastebin %pdb %pde
%f %pdoc %pfile %pinfo %pinfo2 %pip %popd %pprint %precision %prun %psearch %psource %pushd %pwd %pyc
%at %pylab %qtconsole %quickref %recall %rehashx %reload_ext %rep %rerun %reset %reset_selective %rm %r
%mdir %run %save %sc %set_env %store %sx %system %tb %time %timeit %unalias %unload_ext %who %who_ls
%whos %xdel %xmode

Available cell magics:
%%! %%HTML %%SVG %%bash %%capture %%debug %%file %%html %%javascript %%js %%latex %%markdown %%perl %
%prun %ppypy %%python %%python2 %%python3 %%ruby %%script %%sh %%svg %%sx %%system %%time %%timeit %%
writefile

Automagic is ON, % prefix IS NOT needed for line magics.
```

Виконання скрипта .py та інших ноутбуків:

```
In [1]: %run ./draw_graph.py
```



Час виконання коду в комірці:

```
In [4]: %%time
x = 2**100000

CPU times: user 445 µs, sys: 67 µs, total: 512 µs
Wall time: 520 µs
```

`%timeit` запускає код багато раз (кількість можна задати) та виводить середній час з кількох найшвидших виконань:












```
In [7]: %timeit x = 2**100000

501 µs ± 10 µs per loop (mean ± std. dev. of 7 runs, 1000 loops each)
```


Документація з магічних команд:

<https://ipython.org/ipython-doc/3/interactive/magics.html>

Супровід коду (розмітка)












          

Markdown ▾




Диференціальне рівняння:
$$\begin{aligned} \dot{x} &= \sigma(y-x) \end{aligned}$$

Результат:

Markdown ▾



Диференціальне рівняння:
$$\dot{x} = \sigma(y - x)$$


Віджети

Використання interact

```
In [48]: from ipywidgets import interact
```

```
In [49]: def show_factorial(x):  
         f = 1  
         for i in range(1,x+1):  
             f *= i  
         return str(x)+'! = '+str(f)
```

```
In [50]: interact(show_factorial,x=100)
```

x  53

'53! = 4274883284060025564298013753389399649690343788366813724672000000000000'