

BÀI 1: CÀI ĐẶT VÀ LÀM QUEN MÔI TRƯỜNG LẬP TRÌNH PYTHON

1. Mục Tiêu

- Cài đặt môi trường lập trình: Anaconda, Pycharm
- Lập trình trên môi trường Jupyter Notebook
- Làm quen với ngôn ngữ Python

2. Bài tập thực hành




2.1 Cài đặt Python dùng bản phân phối Anaconda

Bài 1. Cài đặt môi trường Anaconda cho lập trình Python

- Tải bản phân phối Anaconda mới nhất về máy
- Cài đặt trong thư mục: **C:\ProgramData\Anaconda3**
- Gắn đường dẫn liên kết của Anaconda đến biến môi trường **PATH**:
C:\ProgramData\Anaconda3; C:\ProgramData\Anaconda3\Scripts
- Kiểm tra môi trường vừa cài đặt bằng cách mở console và gõ lệnh **python**, ghi chú lại phiên bản của python

Hướng dẫn:

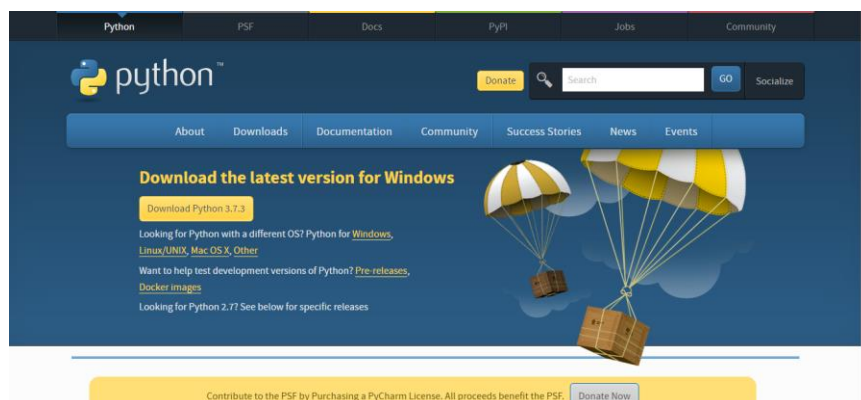
- Anaconda là một bản phân phối mã nguồn mở miễn phí của ngôn ngữ lập trình Python
 - Tích hợp sẵn các phiên bản python.
 - Phục vụ cho machine learning, deep learning, xử lý dữ liệu lớn, phân tích dự đoán và tính toán khoa học.
 - Được hơn 6 triệu người sử dụng.
 - Hơn 1,500+ Python/R data science packages
- Tải Anaconda tại website: <https://www.anaconda.com/products/individual>

Windows 	MacOS 	Linux 
Python 3.8 64-Bit Graphical Installer (457 MB) 32-Bit Graphical Installer (403 MB)	Python 3.8 64-Bit Graphical Installer (435 MB) 64-Bit Command Line Installer (428 MB)	Python 3.8 64-Bit (x86) Installer (529 MB) 64-Bit (Power8 and Power9) Installer (279 MB)

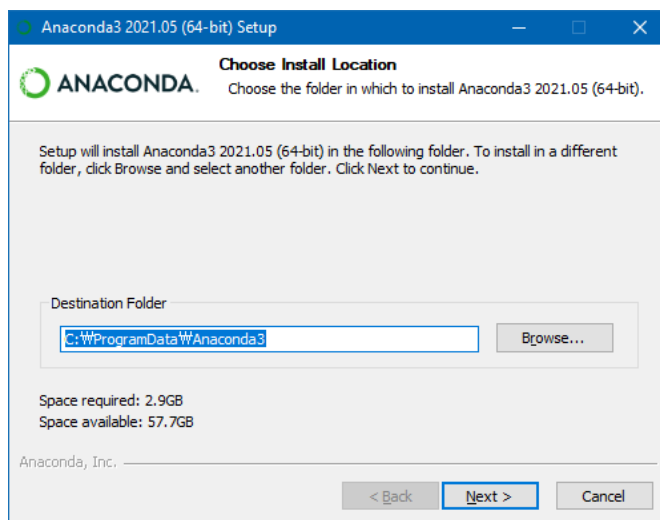
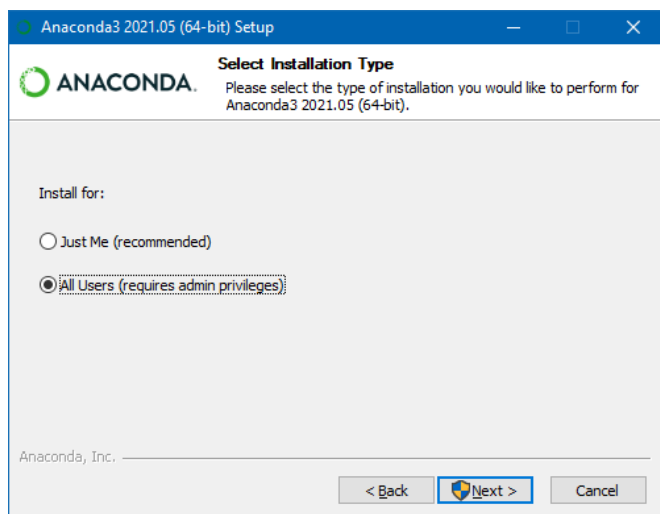
Lưu ý: nếu để cho gọn nhẹ, chúng ta có thể tải phiên bản thu gọn Miniconda chỉ cài đặt ban đầu những gói thư viện cần thiết (<https://docs.conda.io/en/latest/miniconda.html>).

Python version	Name	Size
Python 3.9	Miniconda3 Windows 64-bit	57.7 MiB
	Miniconda3 Windows 32-bit	54.9 MiB
Python 3.8	Miniconda3 Windows 64-bit	57.0 MiB
	Miniconda3 Windows 32-bit	54.2 MiB
Python 2.7	Miniconda2 Windows 64-bit	54.1 MiB
	Miniconda2 Windows 32-bit	47.7 MiB

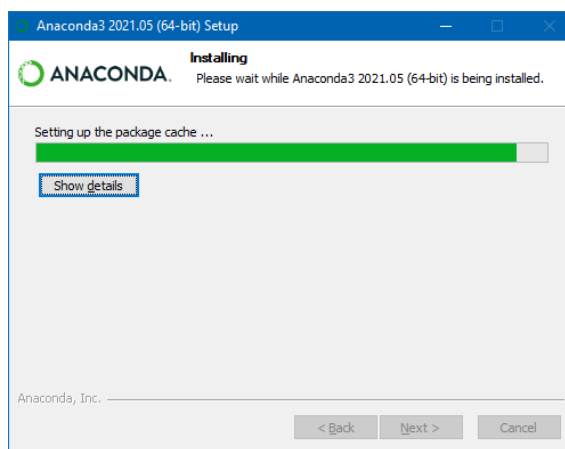
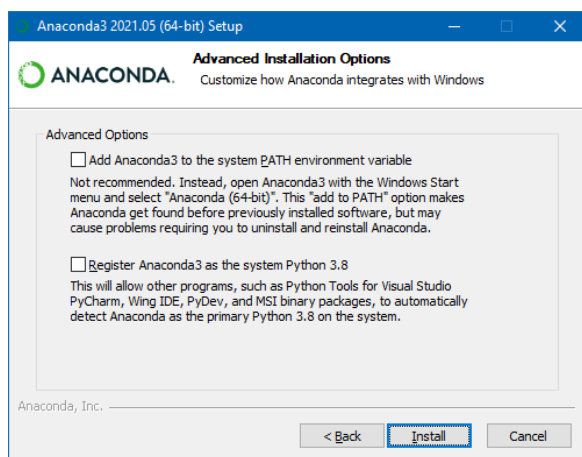
Bên cạnh đó, chúng ta có bản phân phối ngôn ngữ lập trình Python chính thức tại website: <http://www.python.org/downloads>. Tuy nhiên, sử dụng bản phân phối chính thức có nhiều điểm hạn chế tại một số thư viện đòi hỏi cài thêm một số thư viện phụ thuộc tùy biến trên từng môi trường hệ điều hành (Linux, Windows, và Mac).



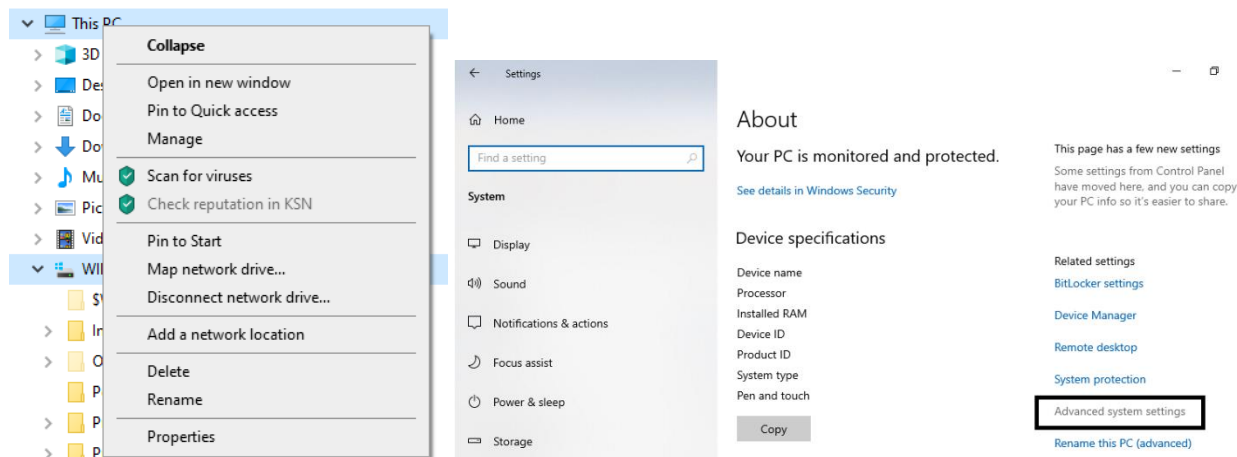
- Trong phạm vi môn học, chúng ta sử dụng bản phân phối Anaconda cho ngôn ngữ lập trình Python.
- Chúng ta cài phiên bản mới nhất dùng cho windows (Anaconda3-2021.05-Windows-x86_64.exe) theo các bước:



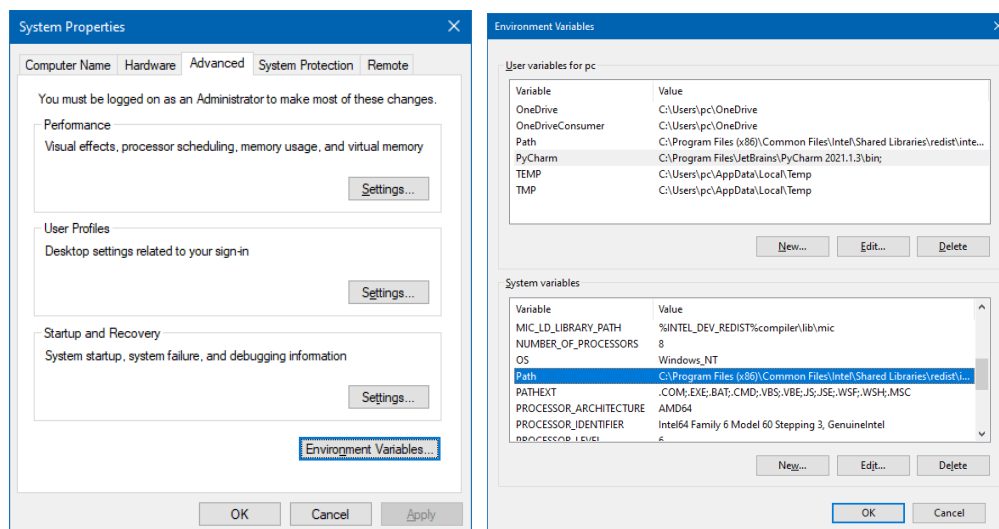
Lưu ý: đường dẫn cài đặt mặc định là **C:\ProgramData\Anaconda3**



- Thiết lập biến môi trường **PATH** dẫn đến thư mục hệ thống Anaconda:
 - Click phải lên biểu tượng This PC trong Window Explorer, chọn Properties. Sau đó, chọn Advanced System Settings trên phía phải của cửa sổ Settings cửa xuất hiện.



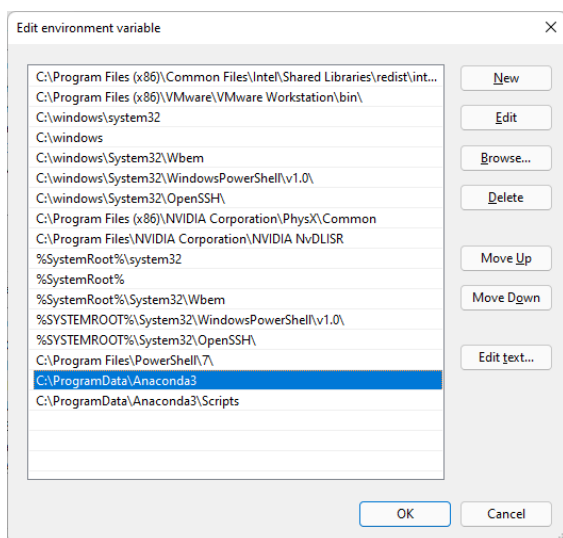
- Chọn Enviroments Variables:



- Chọn biến môi trường PATH, click Edit và bổ sung thêm đường dẫn sau:

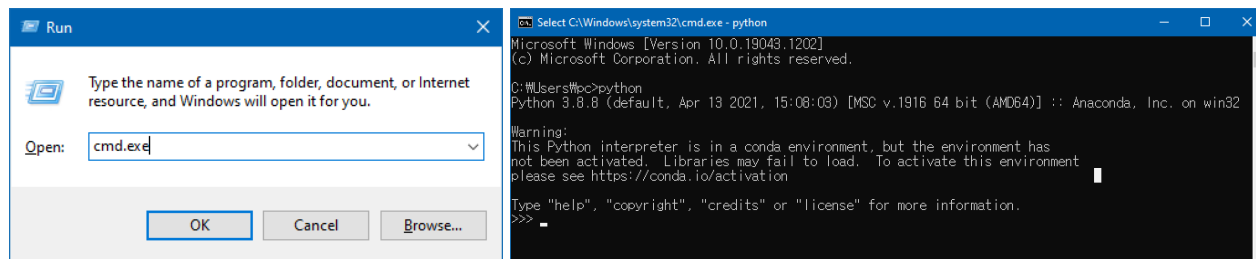
C:\ProgramData\Anaconda3

C:\ProgramData\Anaconda3\Scripts



- Kiểm tra quá trình cài đặt thành công:

Nhấn Windows + R, gõ cmd.exe và gõ lệnh python --version hoặc python



2.2 Quản trị môi trường Anaconda

Bài 2. Quản trị môi trường Anaconda:

- Tạo thư mục `ai_practice_prj` trong ổ đĩa `D:\`
- Mở console và chuyển đến thư mục `D:\ai_practice_prj`
- Kích hoạt môi trường base của Anaconda:
 - Liệt kê xem hiện tại có bao nhiêu môi trường làm việc
 - Liệt kê xem các thư viện đã cài đặt trong môi trường base
 - Cập nhật thư viện conda, pip, ipykernel, jupyter, jupyterlab
 - Cài đặt các thư viện tính toán khoa học: scikit-learn, matplotlib, scipy
- Phục hồi lại về môi trường ban đầu

Hướng dẫn

- Dùng Window Explorer để tạo thư mục `D:\ai_practice_prj`
- Nhấn phím Windows + R và gõ `cmd.exe`
- Thực hiện các lệnh sau:

```
[init_path]>cd /d D:\ai_practice_prj
D:\ai_practice_prj>activate base
(base) D:\ai_practice_prj>conda env list
# conda environments:
#
base                  *  C:\ProgramData\Anaconda3

(base) D:\ai_practice_prj>conda list
# packages in environment at C:\ProgramData\Anaconda3:
#
# Name                        Version           Build    Channel
_ipyw_jlab_nb_ext_conf       0.1.0             py38_0
alabaster                     0.7.12            pyhd3eb1b0_0
anaconda                     2021.05            py38_0
anaconda-client               1.7.2             py38_0
...
(base) D:\ai_practice_prj>python -m pip install --upgrade pip
(base) D:\ai_practice_prj>conda update conda
(base) D:\ai_practice_prj>conda install ipykernel jupyter
(base) D:\ai_practice_prj>pip install jupyterlab==3.1.12
(base) D:\ai_practice_prj>pip install matplotlib scipy scikit-learn
(base) D:\ai_practice_prj>conda.bat deactivate
```

2.3 Làm quen với công cụ lập trình JupyterLab

Bài 3. Lập trình trên JupyterLab:

- Mở console và chuyển đến thư mục `D:\ai_practice_prj`
- Kích hoạt môi trường base và khởi động JupyterLab tại thư mục `D:\ai_practice_prj`
- Tạo cấu trúc thư mục cho thực hành môn học:

```
D:\AI_PRACTICE_PRJ
├─bin
└─data
```

```
└─ lesson1
└─ libs
```

- Vào thư mục **lesson1** và tạo tập tin **my_lesson1.ipynb** như sau:
 - Ô 1 (Markdown Cell) có nội dung sau:


```
# "Thuc hanh tri tue nhan tao Buo i 1"
## "Noi dung thuc hanh"
```
 - Ô 2 (Markdown Cell) có nội dung sau:


```
### In chu ra man hinh
```
 - Ô 3 (Code Cell) có nội dung sau:


```
# lenh 1 - xuat chu
print("Hello, Python World!")
```
 - Ô cuối (Markdown Cell) có nội dung sau:


```
# Ket Thuc
```
- Tạo tập tin **ex0_hello_world.py** in ra chữ "Hello, Python World!" có lệnh sau:


```
print("Hello, Python World!")
```
- Dùng python console, ipython và jupyter notebook để chạy tập tin **ex0_hello_world.py**.

Hướng dẫn

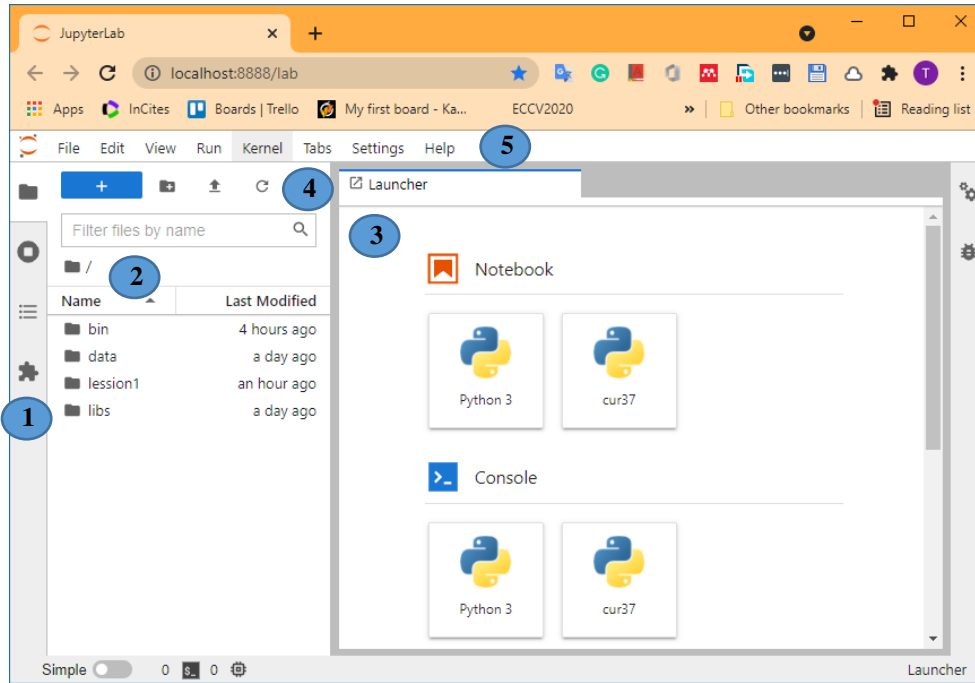
- Nhấn phím Windows + R và gõ cmd.exe
- Thực hiện các lệnh sau:

```
[init_path]>cd /d D:\ai_pratice_prj
D:\ai_pratice_prj>activate base
(base) D:\ai_pratice_prj>jupyter lab
[I 2021-09-17 15:31:46.541 ServerApp] jupyterlab | extension was successfully linked.
...
To access the server, open this file in a browser:
  file:///C:/Users/pc/AppData/Roaming/jupyter/runtime/jpserver-33128-open.html
Or copy and paste one of these URLs:
  http://localhost:8888/lab?token=9c0e9f3638fb79767966caca53d8ac6edeba389536819884
  or http://127.0.0.1:8888/lab?token=9c0e9f3638fb79767966caca53d8ac6edeba389536819884
[I 2021-09-17 15:31:56.008 LabApp] Build is up to date
```

- Jupyter-Lab hay Jupyter-Notebook là môi trường tương tác để chạy mã nguồn Python. Khi chạy lệnh **jupyter lab** tại nơi đặt thư mục gốc, chúng ta sẽ mở trình duyệt và dùng url trên để truy cập:

<http://localhost:8888/lab?token=9c0e9f3638fb79767966caca53d8ac6edeba389536819884>

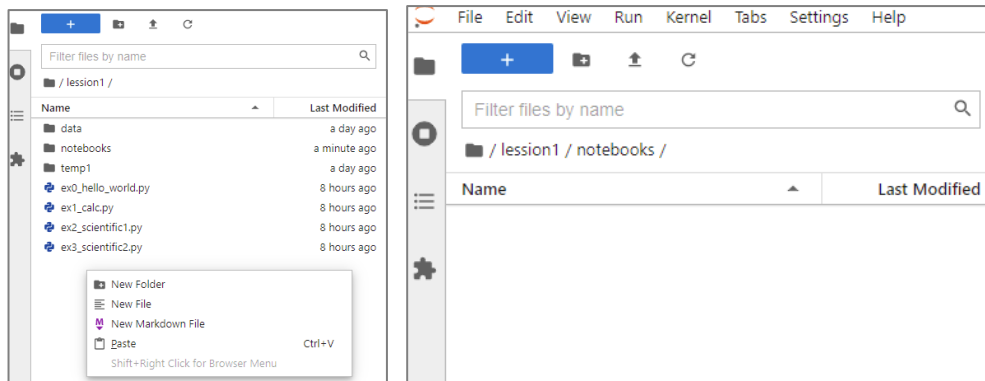
Token [9c0e9f3638fb79767966caca53d8ac6edeba389536819884] là mã tường tự như mật khẩu để truy cập
- Giao diện website của Jupyter-Lab:



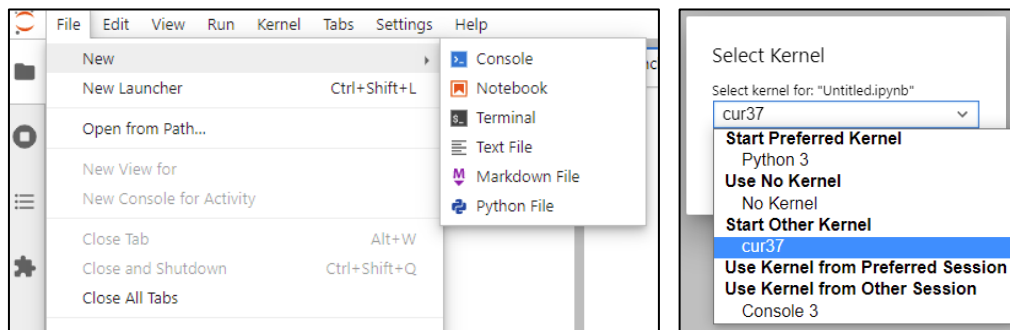
- Vùng danh sách công cụ (1):
 - Quản lý thư mục: Vùng (2) sẽ hiển thị cấu trúc cây thư mục, Vùng (4) bao gồm thêm tập tin, tạo thư mục, tải tập tin lên và cập nhật nội dung
 - Danh sách notebook đang chạy: Vùng (2) dùng để tắt hoặc khởi động lại kernel chạy
 - Table of Contents: Vùng (2) hiển thị cấu trúc tập tin notebook đang mở
 - Các công cụ khác (được cài qua plugins của jupyter-lab)
- Vùng làm việc (3) hiển thị nội dung tập tin và các cửa sổ trực quan dữ liệu
- Thanh thực đơn (5) chứa chức năng chương trình

• Các thao tác trên Jupyter-Lab:

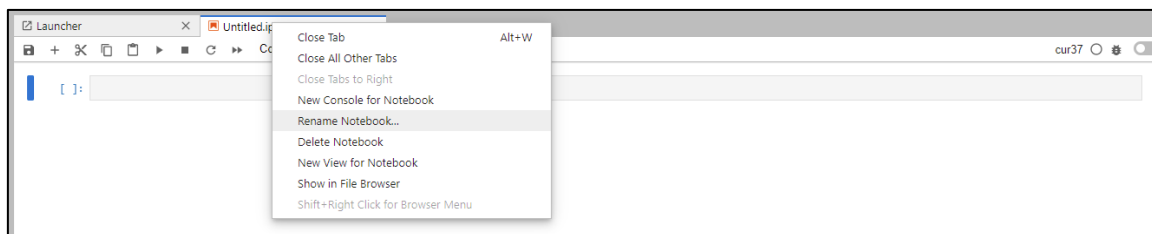
- Di chuyển đến thư mục lesson1, và tạo thư mục notebooks dùng File Browser:



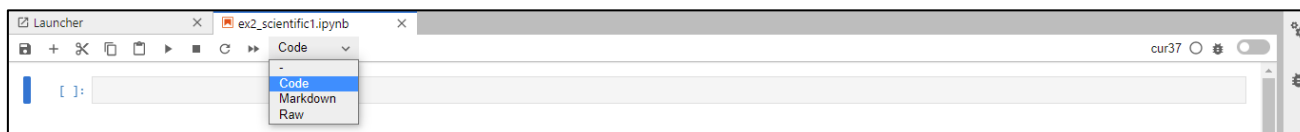
- Mở một tập tin notebook bằng cách chọn Menu File → New → Notebook, chọn kernel Python3



- Đổi tên tập tin thành **ex2_scientific1.ipynb**:



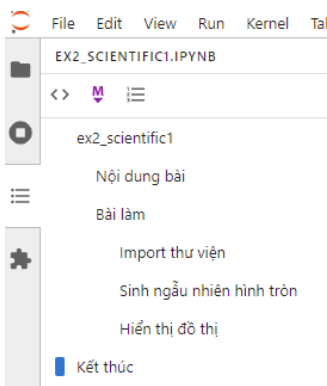
- Vùng làm việc của notebook chứa các cells gồm ba loại **Code**, **Markdown**, **Raw**



- **Markdown cell** chứa các mã markdown để trang trí và ghi chú cho vùng làm việc. Markdown là cú pháp hiển thị định dạng văn bản với cú pháp đơn giản, dễ dàng chuyển đổi sang các kiểu định dạng HTML, LaTeX,

...

- Vào website <https://www.markdownguide.org/cheat-sheet> để xem cú pháp markdown
- Có một số cú pháp chính:
 - Heading: #, ##, ### (tạo cấu trúc tài liệu với Table of Content)
 - Định dạng văn bản: ****Bold****, **italic**, >quote, 1. 2. 3. ordered list, - unorder list, `code`
 - Khối định dạng: Link [title](https://www.example.com), Ảnh ![alt text](image.jpg)
 - Công thức toán: $\$latex equation\$$ → <https://latex.codecogs.com/eqneditor/editor.php>
 - Table: <https://www.tablesgenerator.com/>
- Cấu trúc trình bày của Markdown cho bài tập **ex2_scientific1.ipynb** như sau:



- Giao diện gõ layout bài tập cho ô **Markdown**:

Markdown

```
# ex2_scientific1

## Nội dung bài
Hiển thị ngẫu nhiên $N=50$ hình tròn:
+ Tâm $c_i = (x_i, y_i)$ ngẫu nhiên với $x_i, y_i \in [0, 1]$
+ Màu sắc $color_i$ ngẫu nhiên với $color_i \in [0, 1]$
+ Diện tích $area_i = \pi * r_i^2$ với sinh ngẫu nhiên $r_i \in [0, 1]$

## Bài làm

### Import thư viện

[ ]:

### Sinh ngẫu nhiên hình tròn

[ ]:

[ ]:

### Hiển thị đồ thị

[ ]:

# Kết thúc
```

ex2_scientific1

Nội dung bài

Hiển thị ngẫu nhiên \$N = 50\$ hình tròn:

- Tâm \$c_i = (x_i, y_i)\$ ngẫu nhiên với \$x_i, y_i \in [0, 1]\$
- Màu sắc \$color_i\$ ngẫu nhiên với \$color_i \in [0, 1]\$
- Diện tích \$area_i = \pi * r_i^2\$ với sinh ngẫu nhiên \$r_i \in [0, 1]\$

Bài làm

Import thư viện

[]:

Sinh ngẫu nhiên hình tròn

[]:

[]:

Hiển thị đồ thị

[]:

Kết thúc

- **Code cell:** ô chứa mã nguồn tương tác với python bao gồm các lệnh magics cũng như mã python tương tác
 - Các lệnh magics thông dụng là %run, %matplotlib, ... (xem thêm phần IPython)
 - Các mã chương trình python trên từng ô code sẽ được chạy và hiển thị kết quả trực quan ngay phía dưới dạng inline.
 - Các mã cho cell code trong ex2_scientific1.ipynb như sau:

▼ Import thư viện

```
[1]: import numpy as np
import matplotlib.pyplot as plt
from IPython import display # hiển thị rich output
```

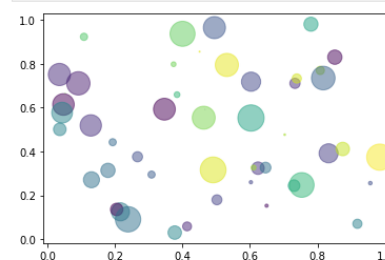
▼ Sinh ngẫu nhiên hình tròn

```
[2]: %time
# generate randomly the center (x,y) of circle, color, area
N = 50
x = np.random.rand(N)
y = np.random.rand(N)
colors = np.random.rand(N)
area = np.pi * (15 * np.random.rand(N)) ** 2 # 0 to 15 point radii

Wall time: 0 ns
```

▼ Hiển thị đồ thị

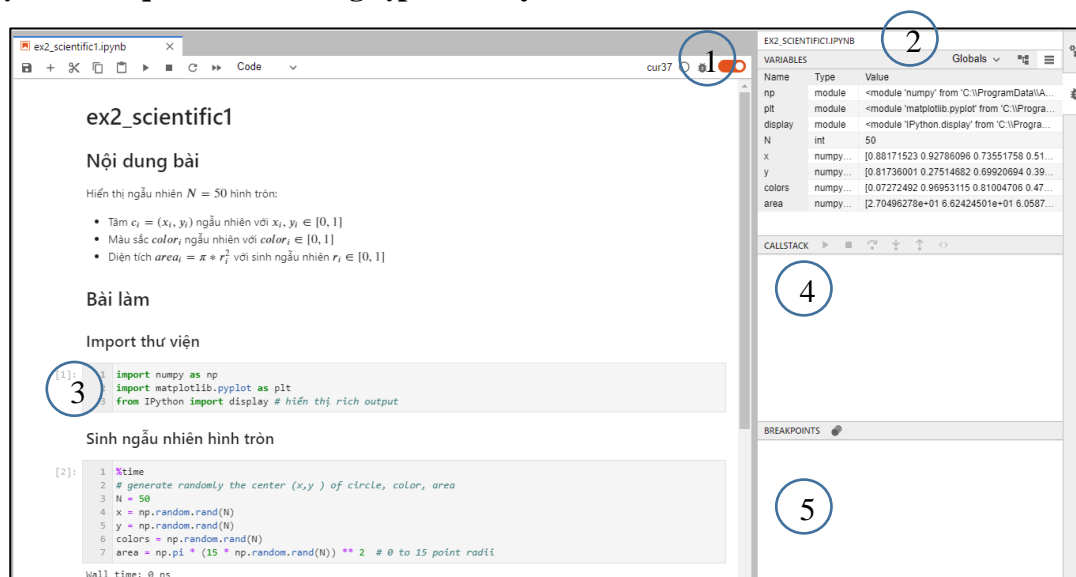
```
[4]: %matplotlib inline
plt.scatter(x, y, s=area, c=colors, alpha=0.5)
plt.show()
```



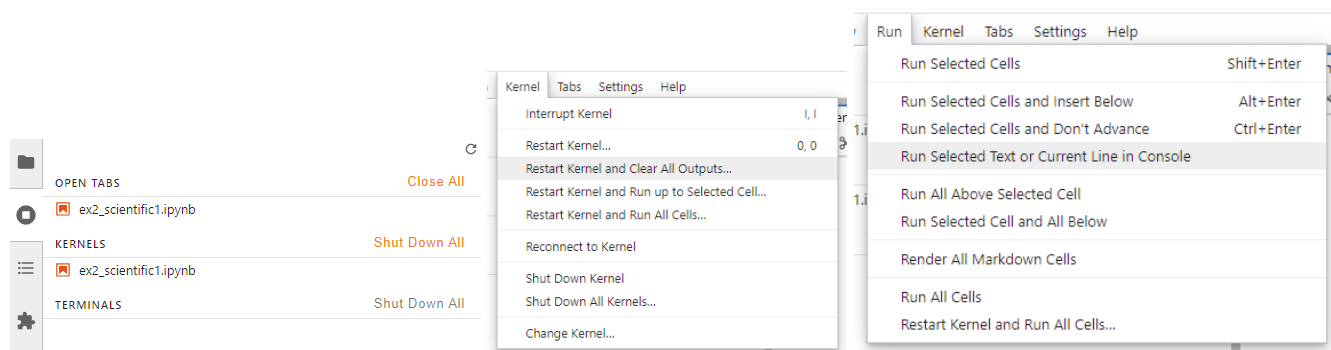
```
[3]: # Mã kiểm tra dữ liệu
print(f' x: len={len(x):d}, [{min(x):.2f}, {max(x):.2f}]: 10 samples [{x[0:10]}]')
print(f' y: len={len(y):d}, [{min(y):.2f}, {max(y):.2f}]: 10 samples [{y[0:10]}]')
print(f' colors: len={len(colors):d}, [{min(colors):.2f}, {max(colors):.2f}]: 10 samples [{colors[0:10]}]')
print(f' area: len={len(area):d}, [{min(area):.2f}, {max(area):.2f}]: 10 samples [{area[0:10]}]')

+ x: len=50, [0.04, 0.98]: 10 samples [[0.73356804 0.95685569 0.60347658 0.53174887 0.21627315 0.83325355
0.04740813 0.20526461 0.46647076 0.4948485 ]]
+ y: len=50, [0.03, 0.98]: 10 samples [[0.71104027 0.25563421 0.71851312 0.79492536 0.12463443 0.39170714
0.61448905 0.13565557 0.59134363 0.96525805]]
+ colors: len=50, [0.03, 0.98]: 10 samples [[0.15535275 0.25578224 0.23576306 0.95040495 0.4213997 0.20419906
0.03035512 0.02819051 0.28668777 0.280979 ]]
+ area: len=50, [0.02, 625.15]: 10 samples [[9.61908582e+01 1.20734350e+01 3.34727285e+02 4.84023376e+02
2.98636328e+02 3.30774365e+02 4.21557366e+02 1.42886939e+02
2.20982924e-02 4.41972884e+02]]
```

- Mở tính năng debug (1) để xem sự thay đổi các biến (2) cũng như đặt breakpoint bằng cách click vào dòng cần debug (3) được đánh số và quan sát call stacks (4) và các breakpoints (5):



- Quan sát các kernel đang chạy, thao tác trên kernel và chạy lại cũng như chạy lệnh:



- Xóa nội dung chạy Edit → Clear All Outputs
- Kết xuất ra các định dạng khác: File → Export Notebook As → HTML

2.4 Lập trình với JupyterLab

Bài 4. Tính toán đơn giản

- Tạo tập tin **ex1_calc.py** trong thư mục **lesson1**:

```
print("Calculator Basic")
a = int(input("Moi ban nhap so a: "))
b = int(input("Moi ban nhap so b: "))
c = a + b
print("c = %d + %d = %d" % (a, b, c))
```

- Thêm một code cell trong **my_lesson1.ipynb** để chạy tập tin **ex1_calc.py**.

Hướng dẫn

- Gõ **%run ex1_calc.py**

Bài 5. Tính toán khoa học

- Tạo một tập tin **ex2_scientific1.ipynb** trong thư mục **lesson1** được trang trí như sau:

Markdown

```
# ex2_scientific1

## Nội dung bài
Hiển thị ngẫu nhiên $N=50$ hình tròn:
+ Tâm $c_i = (x_i, y_i)$ ngẫu nhiên với $x_i, y_i \in [0, 1]$
+ Màu sắc $color_i$ ngẫu nhiên với $color_i \in [0, 1]$
+ Diện tích $area_i = \pi * r_i^2$ với sinh ngẫu nhiên $r_i \in [0, 1]$

## Bài làm

### Import thư viện

[ ]:

### Sinh ngẫu nhiên hình tròn

[ ]:

[ ]:

### Hiển thị đồ thị

[ ]:

# Kết thúc
```

ex2_scientific1

Nội dung bài

Hiển thị ngẫu nhiên $N = 50$ hình tròn:

- Tâm $c_i = (x_i, y_i)$ ngẫu nhiên với $x_i, y_i \in [0, 1]$
- Màu sắc $color_i$ ngẫu nhiên với $color_i \in [0, 1]$
- Diện tích $area_i = \pi * r_i^2$ với sinh ngẫu nhiên $r_i \in [0, 1]$

Bài làm

Import thư viện

[]:

Sinh ngẫu nhiên hình tròn

[]:

[]:

Hiển thị đồ thị

[]:

Kết thúc

- Gõ và chạy Code Cell Import thư viện như sau:

Import thư viện

```
[1]: import numpy as np
import matplotlib.pyplot as plt
from IPython import display # hiển thị rich output
```

- Gõ và chạy Code Cell sinh ngẫu nhiên hình tròn:

Sinh ngẫu nhiên hình tròn

```
[2]: %time
# generate randomly the center (x,y) of circle, color, area
N = 50
x = np.random.rand(N)
y = np.random.rand(N)
colors = np.random.rand(N)
area = np.pi * (15 * np.random.rand(N)) ** 2 # 0 to 15 point radii

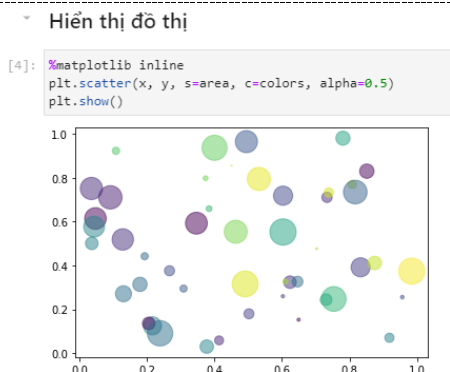
Wall time: 0 ns
```

- Gõ và chạy Code Cell kiểm tra dữ liệu:

```
[3]: # Mã kiểm tra dữ liệu
print(f'+ x: len={len(x):d}, [{min(x):.2f}, {max(x):.2f}]: 10 samples [{x[0:10]}]')
print(f'+ y: len={len(y):d}, [{min(y):.2f}, {max(y):.2f}]: 10 samples [{y[0:10]}]')
print(f'+ colors: len={len(colors):d}, [{min(colors):.2f}, {max(colors):.2f}]: 10 samples [{colors[0:10]}]')
print(f'+ area: len={len(area):d}, [{min(area):.2f}, {max(area):.2f}]: 10 samples [{area[0:10]}]')

+ x: len=50, [0.04, 0.98]: 10 samples [[0.73356804 0.95685569 0.60347658 0.53174887 0.21627315 0.83325355
0.04740813 0.20526461 0.46647076 0.4948485 ]]
+ y: len=50, [0.03, 0.98]: 10 samples [[0.71104027 0.25563421 0.71851312 0.79492536 0.12463443 0.39170714
0.61448985 0.13565557 0.59134363 0.96525805]]
+ colors: len=50, [0.03, 0.98]: 10 samples [[0.15535275 0.25578224 0.23576306 0.95040495 0.4213997 0.20419906
0.03035512 0.02819051 0.28668777 0.280979 ]]
+ area: len=50, [0.02, 625.15]: 10 samples [[9.61908582e+01 1.20734358e+01 3.34727285e+02 4.84023376e+02
2.98636328e+02 3.30774365e+02 4.21557366e+02 1.42886939e+02
2.20982924e-02 4.41972884e+02]]
```

- Gõ và chạy Code Cell hiển thị đồ thị:



- Mở Toolbox Debug và quan sát biến dữ liệu
- Xuất toàn bộ notebook ra định dạng html

3. Bài tập tự thực hành

Bài 6. Cài đặt thêm thư viện cho môi trường base

- Mở console và chuyển đến thư mục **D:\ai_practice_prj**
- Kích hoạt môi trường base của Anaconda:
 - Kiểm tra số lượng môi trường cài đặt, và các gói thư viện đã cài đặt trong môi trường base
 - Cài đặt thư viện xử lý ảnh: opencv-python, opencv-contrib-python, pillow, scikit-image
 - Cài đặt các thư viện khác: tqdm, pandas, ipywidgets, pytables, graphviz, pydot, xlrd
 - Cài đặt thư viện widgetsnbextension (từ channel anaconda), nodejs (từ channel conda-forge)
 - Mở tính năng nbextension và cài đặt một số plugins cho jupyterlab: @jupyter-widgets/jupyterlab-manager, @jupyterlab/toc
- Phục hồi lại môi trường ban đầu

Hướng dẫn

- Nhấn phím Windows + R và gõ cmd.exe
- Thực hiện các lệnh sau:

```
[init_path]>cd /d D:\ai_practice_prj
D:\ai_practice_prj>activate base
(base) D:\ai_practice_prj>pip install opencv-python opencv-contrib-python pillow scikit-image
(base) D:\ai_practice_prj>pip install tqdm pandas ipywidgets pytables graphviz pydot xlrd
(base) D:\ai_practice_prj>conda install -c conda-forge nodejs
(base) D:\ai_practice_prj>conda install -c anaconda widgetsnbextension
(base) D:\ai_practice_prj>jupyter nbextension enable --py widgetsnbextension
(base) D:\ai_practice_prj>jupyter labextension install @jupyter-widgets/jupyterlab-manager
(base) D:\ai_practice_prj>jupyter labextension install @jupyterlab/toc
(base) D:\ai_practice_prj>jupyter lab clean
(base) D:\ai_practice_prj>jupyter lab build
(base) D:\ai_practice_prj>conda.bat deactivate
D:\ai_practice_prj>
```

Bài 7. Tính chu vi và diện tích hình tròn:

- Tạo tập tin **ex3_circle.py** có chức năng nhập vào bán kính r là số thực.
- Sau đó, tính chu vi và diện tích hình tròn biết: $cv = 2\pi r$, $dt = \pi r^2$
- Thêm một code cell trong **my_lesson1.ipynb** để chạy tập tin **ex3_circle.py**.

Bài 8. Hiển thị đồ thị $\sin(x)$ và $\cos(x)$:

- Tạo tập tin `ex4_scientific2.ipynb` có nội dung sau: (Lưu ý: `%% md` là ô Markdown, còn `%%` là ô Code)

```
%% md
# ex4_scientific2
## Nội dung bài
Hiển thị 2 đồ thị  $\sin(x)$ ,  $\cos(x)$  trong khoảng  $x \in [-\pi, \pi]$ 
## Bài làm

%% md
### Import thư viện

%%
import numpy as np
import matplotlib.pyplot as plt

%% md
### Sinh dữ liệu hàm số

%%
X = np.linspace(-np.pi, np.pi, 256, endpoint=True) # generate x in  $[-\pi, \pi]$ 
C, S = np.cos(X), np.sin(X)

%%
print(f'+ X: |X|={len(X)}, x in [{np.min(X):.2f}, {np.max(X):.2f}], 10 samples: {X[0:10]}')
print(f'+ Cos(X): |C|={len(C)}, C in [{np.min(C):.2f}, {np.max(C):.2f}], 10 samples: {C[0:10]}')
print(f'+ Sin(X): |S|={len(S)}, S in [{np.min(S):.2f}, {np.max(S):.2f}], 10 samples: {S[0:10]}')

%% md
### Hiển thị đồ thị

%%
# visualize data
plt.plot(X, C, color="blue", linewidth=2.5, linestyle="-");
plt.plot(X, S, color="red", linewidth=2.5, linestyle="-");

# limit on x axis
plt.xlim(X.min() * 1.1, X.max() * 1.1);
plt.xticks([-np.pi, -np.pi / 2, 0, np.pi / 2, np.pi],
           [r'$-\pi$', r'$-\pi/2$', r'$0$', r'$+\pi/2$', r'$+\pi$']);

# limit on y axis
plt.ylim(C.min() * 1.1, C.max() * 1.1);
plt.yticks([-1, 0, +1], [r'$-1$', r'$0$', r'$+1$']);

plt.show()

%% md
# Kết thúc
```

- Kết quả hiển thị như sau:

ex4_scientific2

Nội dung bài

Hiển thị 2 đồ thị $\sin(x)$, $\cos(x)$ trong khoảng $x \in [-\pi, \pi]$

Bài làm

Import thư viện

```
In [1]: import numpy as np
import matplotlib.pyplot as plt
```

Sinh dữ liệu hàm số

```
In [2]: X = np.linspace(-np.pi, np.pi, 256, endpoint=True) # generate x in  $[-\pi, \pi]$ 
C, S = np.cos(X), np.sin(X)
```

```
In [16]: print(f'+ X: |X|={len(X)}, x in [{np.min(X):.2f}, {np.max(X):.2f}], 10 samples: {X[0:10]}')
print(f'+ Cos(X): |C|={len(C)}, C in [{np.min(C):.2f}, {np.max(C):.2f}], 10 samples: {C[0:10]}')
print(f'+ Sin(X): |S|={len(S)}, S in [{np.min(S):.2f}, {np.max(S):.2f}], 10 samples: {S[0:10]}')

+ X: |X|=256, x in [-3.14, 3.14], 10 samples: [-3.14159265 -3.11895271 -3.09231277 -3.06767283 -3.04303288 -3.01839294
-2.993753 -2.96911306 -2.94447311 -2.91983317]
+ Cos(X): |C|=256, C in [-1.00, 1.00], 10 samples: [-1.          -0.99906645 -0.99878599 -0.99726917 -0.99514692 -0.99242051
-0.98909361 -0.9848223 -0.9803477 -0.97551197]
+ Sin(X): |S|=256, S in [-1.00, 1.00], 10 samples: [-1.22464680e-16 -2.46374402e-02 -4.92599411e-02 -7.38523275e-02
-9.84862783e-02 -1.22882251e-01 -1.47301680e-01 -1.71625679e-01
-1.95949467e-01 -1.12040350e-01]
```

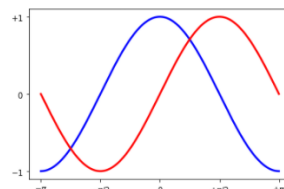
Hiển thị đồ thị

```
In [22]: # visualize data
plt.plot(X, C, color="blue", linewidth=2.5, linestyle="-");
plt.plot(X, S, color="red", linewidth=2.5, linestyle="-");

# limit on x axis
plt.xlim(X.min() * 1.1, X.max() * 1.1);
plt.xticks([-np.pi, -np.pi / 2, 0, np.pi / 2, np.pi],
           [r'$-\pi$', r'$-\pi/2$', r'$0$', r'$+\pi/2$', r'$+\pi$']);

# limit on y axis
plt.ylim(C.min() * 1.1, C.max() * 1.1);
plt.yticks([-1, 0, +1], [r'$-1$', r'$0$', r'$+1$']);

plt.show()
```



Kết thúc

- Lưu notebook ra html

--- Hết ---