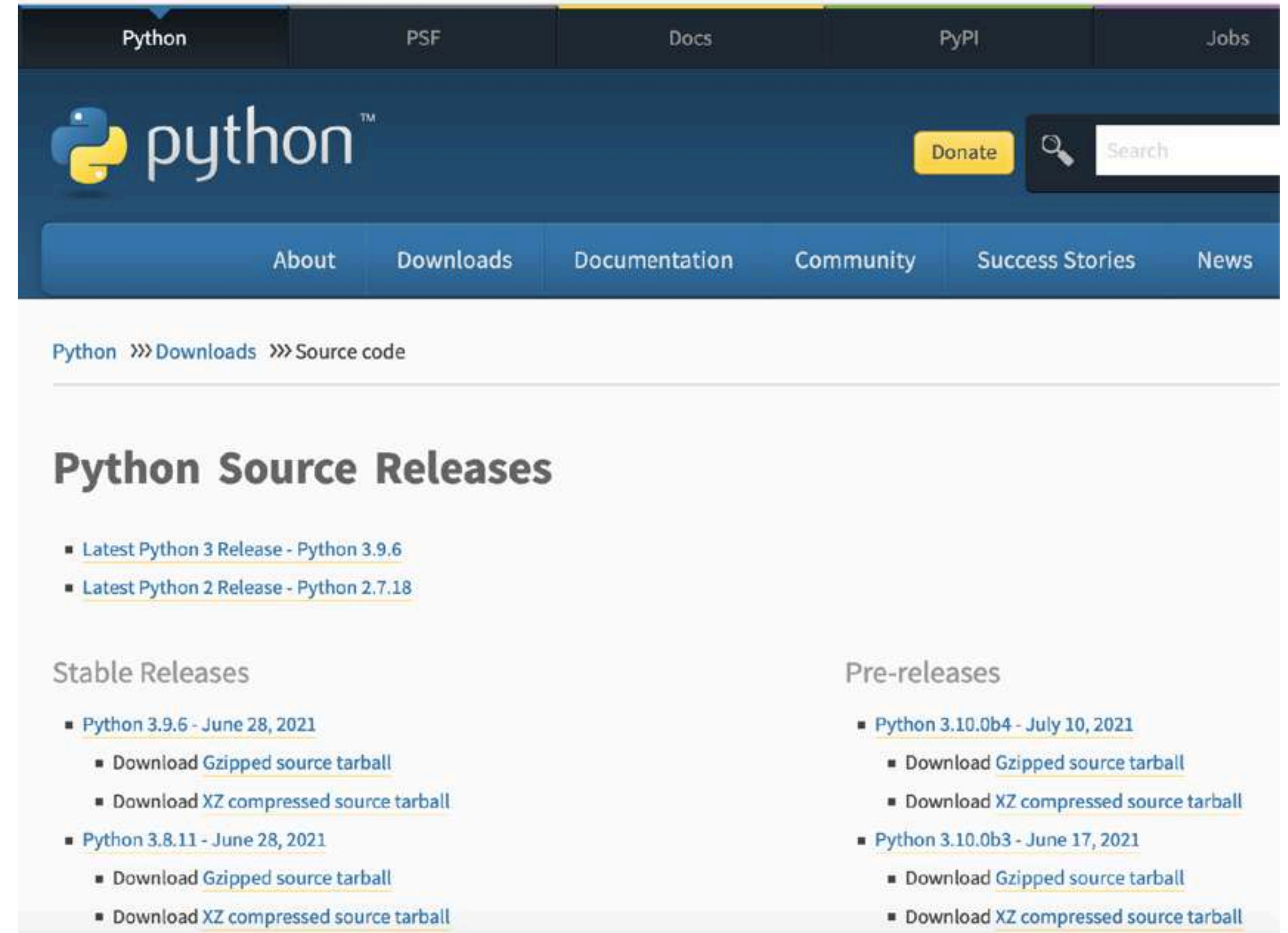


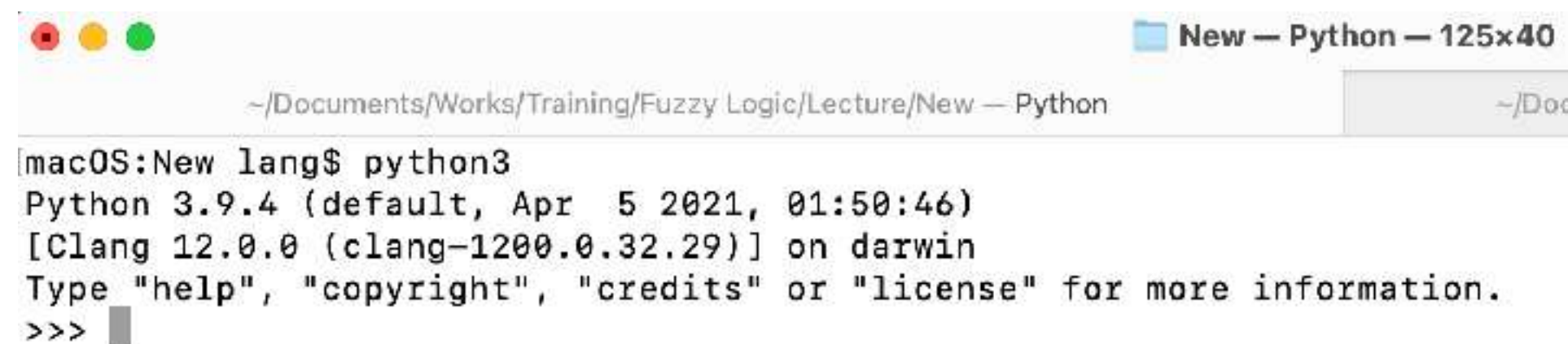
5. Một ít về Python

- Phiên bản tháng 02/2023 là Python 3.11.2
- Có thể cập nhật tại <https://www.python.org/downloads/source/>



Sử dụng Python

- Dùng ở chế độ console với dòng lệnh theo kiểu thông dịch



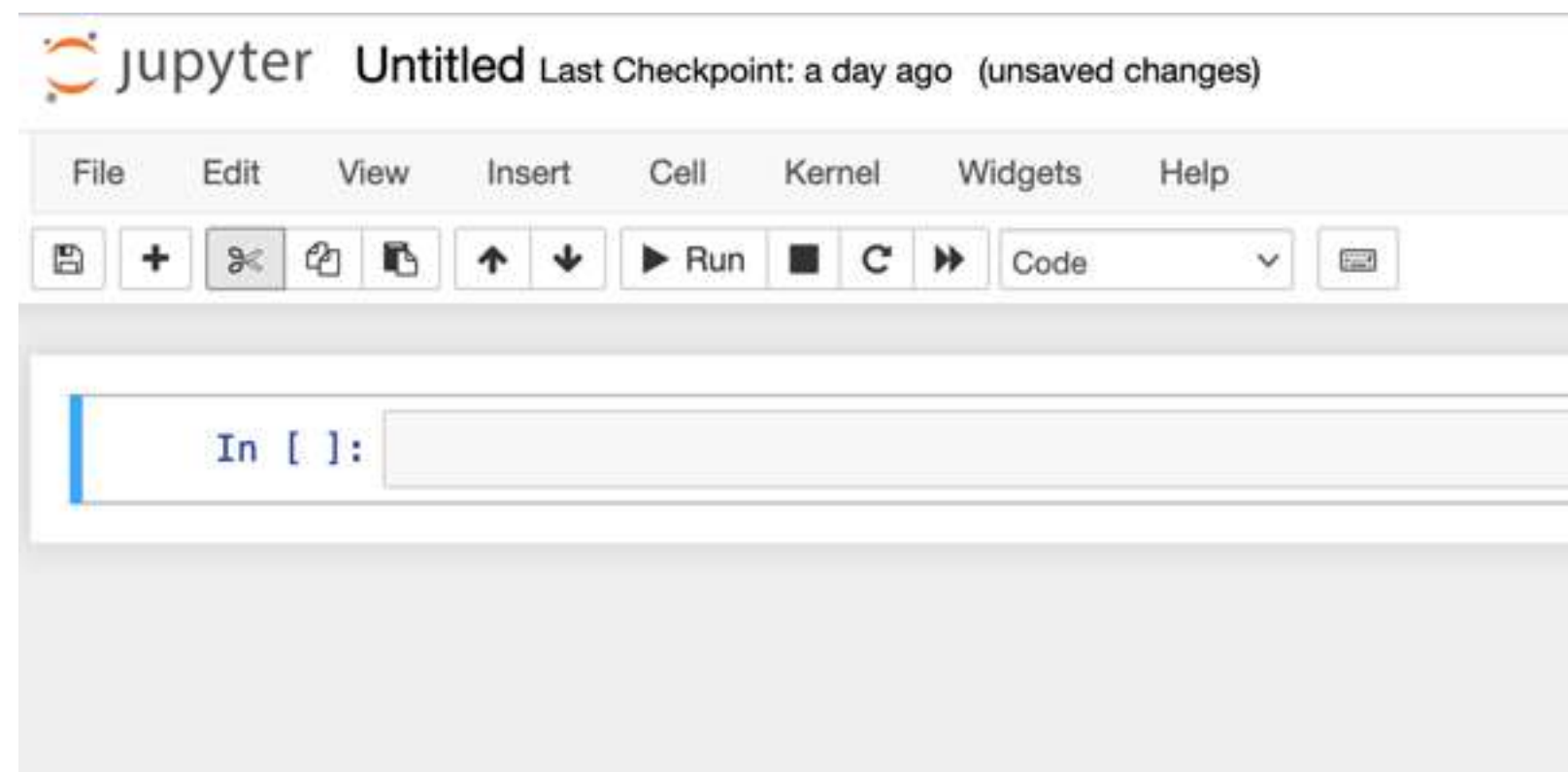
```
~/Documents/Works/Training/Fuzzy Logic/Lecture/New — Python
macOS:New lang$ python3
Python 3.9.4 (default, Apr 5 2021, 01:50:46)
[Clang 12.0.0 (clang-1200.0.32.29)] on darwin
Type "help", "copyright", "credits" or "license" for more information.
>>>
```

- Sử dụng một trình soạn thảo đơn giản, chẳng hạn Notepad (Windows), TextEdit (macOS), Nano (Linux) sau đó biên dịch và thực thi ở chế độ Command Line

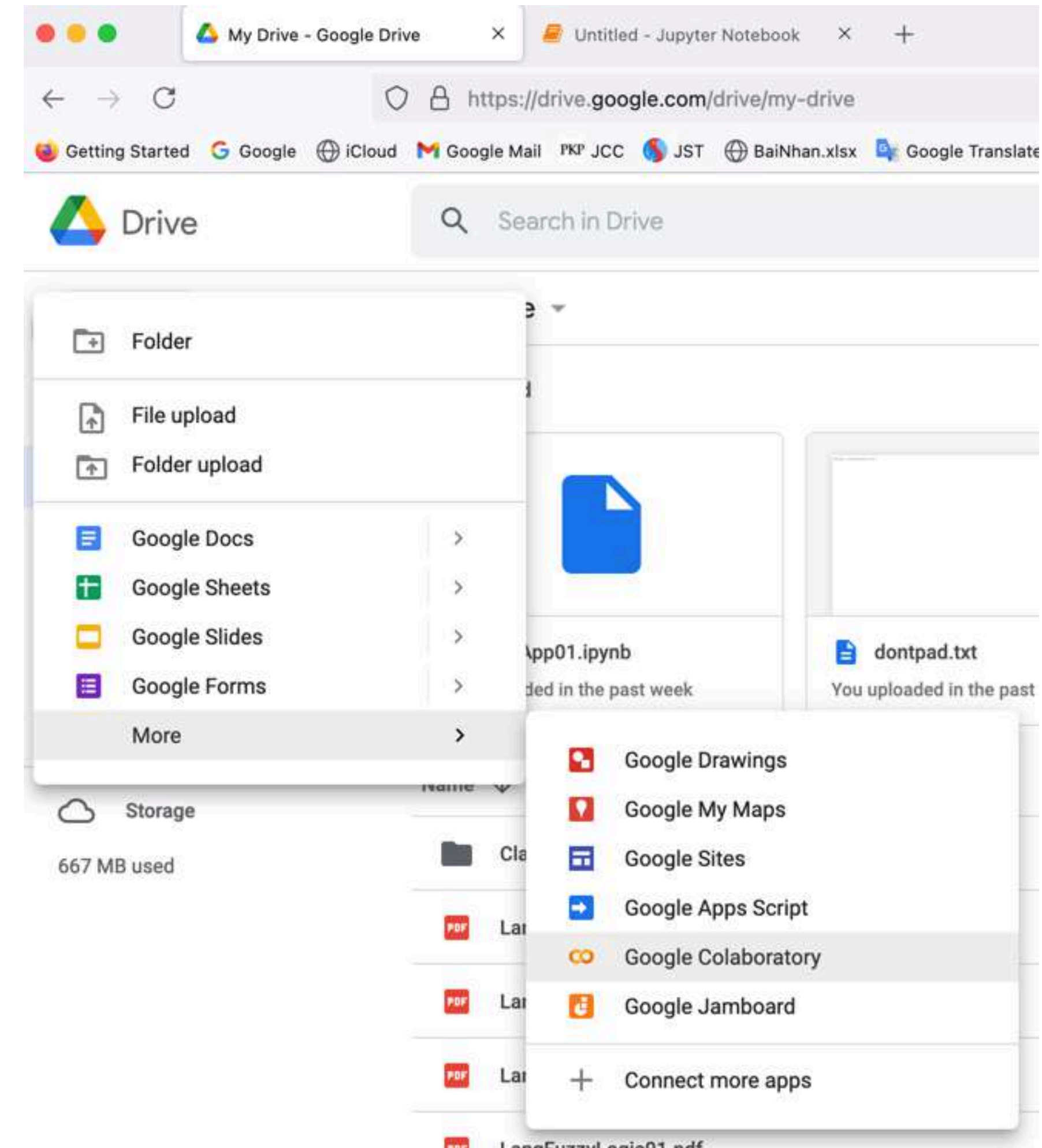
- Thông qua một IDE nào đó, chẳng hạn PyCharm



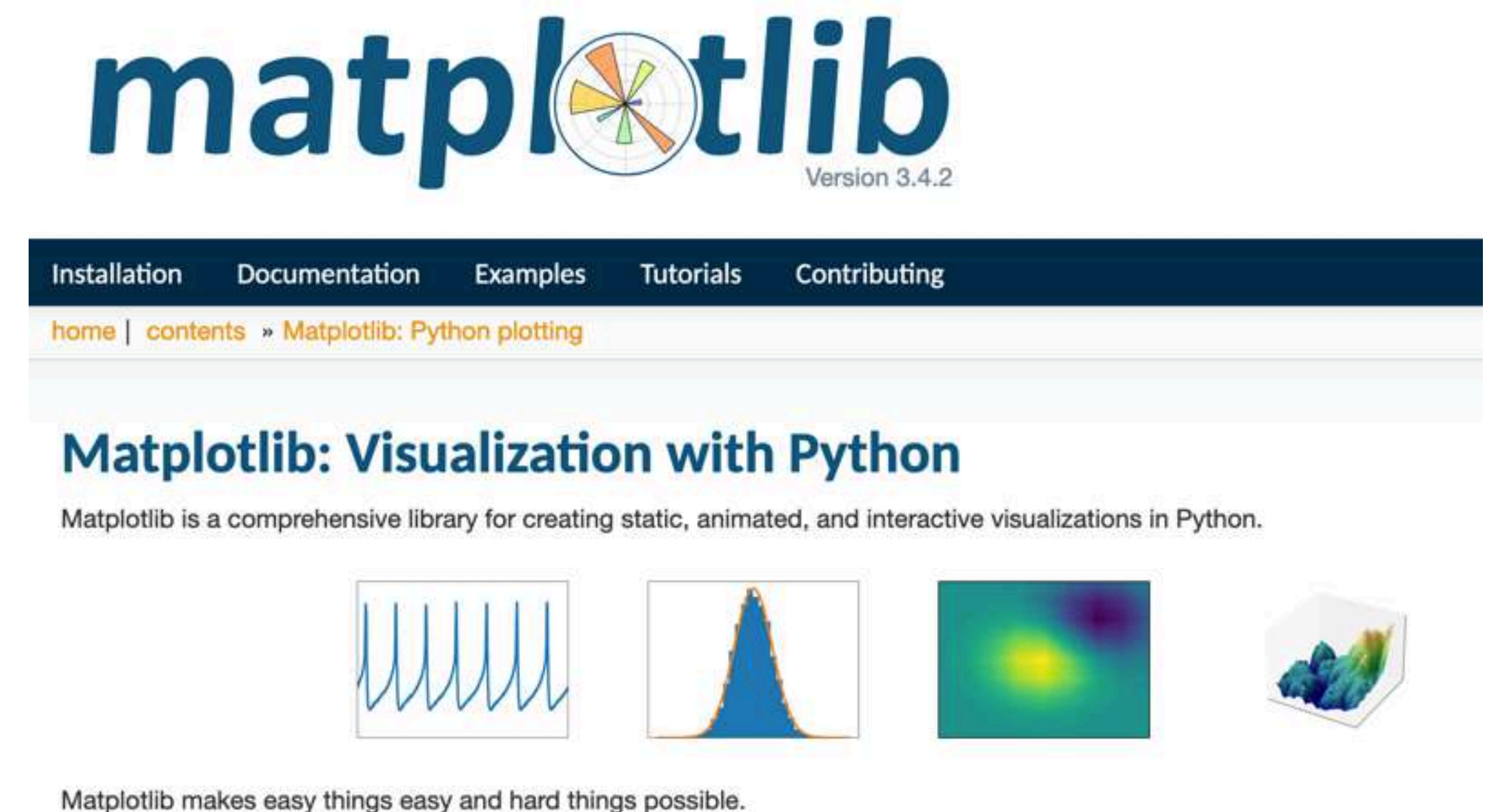
- Hoặc sử dụng Jupyter-Lab để dùng giao diện web bằng cách khởi động Jupyter Server trên máy cá nhân thông qua việc khởi động trực tiếp jupyter notebook.
- Lưu ý, để dùng được, trước đó phải install gói jupyter



- Khi cần sử dụng nguồn tài nguyên ảo trên internet, có thể dùng Google Colab có trong Google Drive



- Để thuận lợi, các gói thư viện tối thiểu cần có cho học phần đó là:
 - Numpy để sử dụng các công cụ của toán học tính toán, đại số tuyến tính
 - Scikit-Fuzzy cho những API liên quan đến tập hợp mờ
 - Pandas để xử lý dữ liệu dạng bảng
 - Matplotlib để trực quan hoá dữ liệu



Tập hợp mờ trong Python

- Tập hợp mờ được xây dựng qua hàm thành viên, trên máy tính hàm thành viên chính là giá trị phụ thuộc của các phần tử trong không gian nền lên tập mờ.
- Từ đó, có thể coi hàm thành viên là mảng các giá trị có kích thước giống với mảng không gian nền.
- Chẳng hạn với tập mờ phòng học rộng A trên không gian nền các phòng học

$$X = \{30, 50, 75, 90, 100, 125, 150, 200\} \text{ với}$$

$$A = \{(30, 0), (50, 0), (75, 0), (90, 0.2), (100, 0.5), (125, 0.8), (150, 1), (200, 1)\}$$

- Trước tiên cần include các thư viện sử dụng

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

```
1 import numpy as np
2 import matplotlib.pyplot as plt
```

```
1 X = np.array( [30,50,75,90,100,125,150,200] )
2 A = np.array( [0,0,0,0.1,0.5,0.8,1,1] )
```

- Sau đó tạo các mảng một chiều cho 2 tập hợp X và A .

```
X = np.array( [30,50,75,90,100,125,150,200] )
A = np.array( [0,0,0,0.1,0.5,0.8,1,1] )
```

- Để hình dung, có thể trực quan hoá thông qua các hàm của Matplotlib

```

plt.figure( figsize=(12,6) )
plt.title( "ĐỒ THỊ BIỂU DIỄN HÀM THUỘC CỦA TẬP MỜ" )
plt.plot( X,A,label="Tập mờ A = " + r"$\{\frac{0}{30},\frac{0}{50},\frac{0}{75},\frac{0.1}{90},\frac{0.5}{100},\frac{0.8}{125},\frac{1}{150},\frac{1}{200},\}$",color="blue",marker="+" )
plt.plot( X,B,label="Tập mờ B = " + r"$\{\frac{1}{30},\frac{1}{50},\frac{0.9}{75},\frac{0.8}{90},\frac{0.7}{100},\frac{0.3}{125},\frac{0.1}{150},\frac{0}{200},\}$",color="red",marker="*")
plt.plot( X,C,label="Tập mờ C = " + r"$\{\frac{0}{30},\frac{0.3}{50},\frac{0.9}{75},\frac{1}{90},\frac{1}{100},\frac{0.9}{125},\frac{0.5}{150},\frac{0}{200},\}$",color="green",marker="o" )
plt.xlabel( "Không gian nền X = " + str(X) )
plt.ylabel( "Mức độ phụ thuộc: " + r"$\mu_A(x)$" )
plt.legend( loc="best" )
plt.show()

```

```

1 X = np.array( [30,50,75,90,100,125,150,200] )
2 A = np.array( [0,0,0,0.1,0.5,0.8,1,1] )
3 B = np.array( [1,1,0.9,0.8,0.7,0.3,0.1,0] )
4 C = np.array( [0,0.3,0.9,1,1,0.9,0.5,0] )

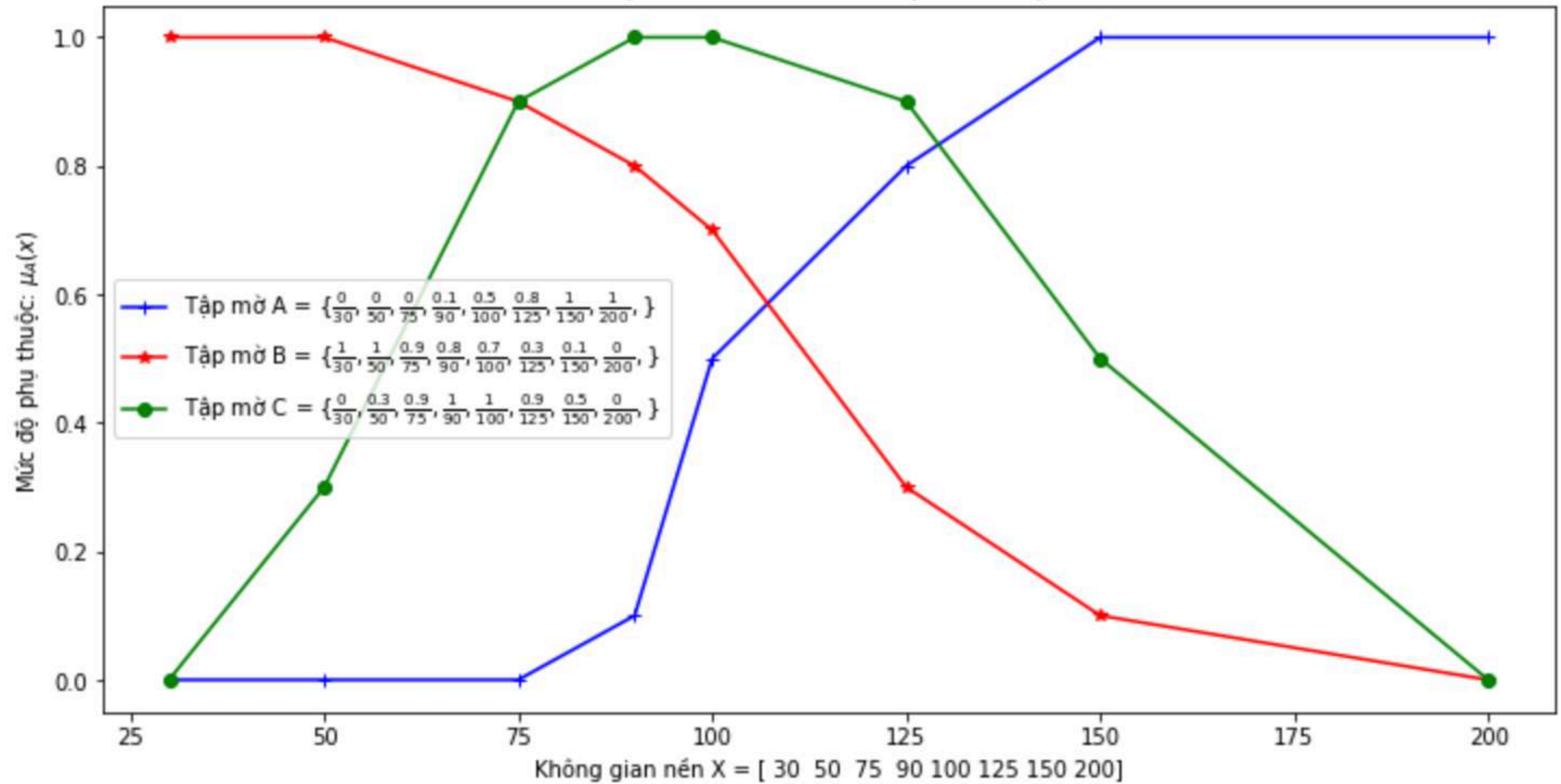
```

```

1 plt.figure( figsize=(12,6) )
2 plt.title( "ĐỒ THỊ BIỂU DIỄN HÀM THUỘC CỦA TẬP MỜ" )
3 plt.plot( X,A,label="Tập mờ A = " + r"$\{\frac{0}{30},\frac{0}{50},\frac{0}{75},\frac{0.1}{90},\frac{0.5}{100},\frac{0.8}{125},\frac{1}{150},\frac{1}{200},\}$",color="blue",marker="+" )
4 plt.plot( X,B,label="Tập mờ B = " + r"$\{\frac{1}{30},\frac{1}{50},\frac{0.9}{75},\frac{0.8}{90},\frac{0.7}{100},\frac{0.3}{125},\frac{0.1}{150},\frac{0}{200},\}$",color="red",marker="*")
5 plt.plot( X,C,label="Tập mờ C = " + r"$\{\frac{0}{30},\frac{0.3}{50},\frac{0.9}{75},\frac{1}{90},\frac{1}{100},\frac{0.9}{125},\frac{0.5}{150},\frac{0}{200},\}$",color="green",marker="o" )
6 plt.xlabel( "Không gian nền X = " + str(X) )
7 plt.ylabel( "Mức độ phụ thuộc: " + r"$\mu_A(x)$" )
8 plt.legend( loc="best" )
9 plt.show()

```


ĐỒ THỊ BIỂU DIỄN HÀM THUỘC CỦA TẬP MỜ



- Ở trên dùng Jupyter.
- Trong trường hợp thực thi ở chế độ dòng lệnh, có thể lưu các lệnh vào một tập tin - chẳng hạn FuzzySetFromNumpy.py với nội dung như sau:

```
import numpy as np
import matplotlib.pyplot as plt
plt.figure( figsize=(12,6) )
plt.title( "ĐỒ THỊ BIỂU DIỄN HÀM THUỘC CỦA TẬP MỜ" )
plt.plot( X,A,label="Tập mờ A = " + r"$\{\frac{0}{30},\frac{0}{50},\frac{0}{75},\frac{0.1}{90},\frac{0.5}{100},\frac{0.8}{125},\frac{1}{150},\frac{1}{200},\}$",color="blue",marker="+" )
plt.plot( X,B,label="Tập mờ B = " + r"$\{\frac{1}{30},\frac{1}{50},\frac{0.9}{75},\frac{0.8}{90},\frac{0.7}{100},\frac{0.3}{125},\frac{0.1}{150},\frac{0}{200},\}$",color="red",marker="*")
plt.plot( X,C,label="Tập mờ C = " + r"$\{\frac{0}{30},\frac{0.3}{50},\frac{0.9}{75},\frac{1}{90},\frac{1}{100},\frac{0.9}{125},\frac{0.5}{150},\frac{0}{200},\}$",color="green",marker="o" )
plt.xlabel( "Không gian nền X = " + str(X) )
plt.ylabel( "Mức độ phụ thuộc: " + r"$\mu_A(x)$" )
plt.legend( loc="best" )
plt.show()
```

- Lưu ý rằng, Jupyter tạo ra tập tin có tên là *name.ipynb*, khi cần chuyển sang tập tin *name.py* dùng lệnh
jupyter nbconvert --to script name.ipynb

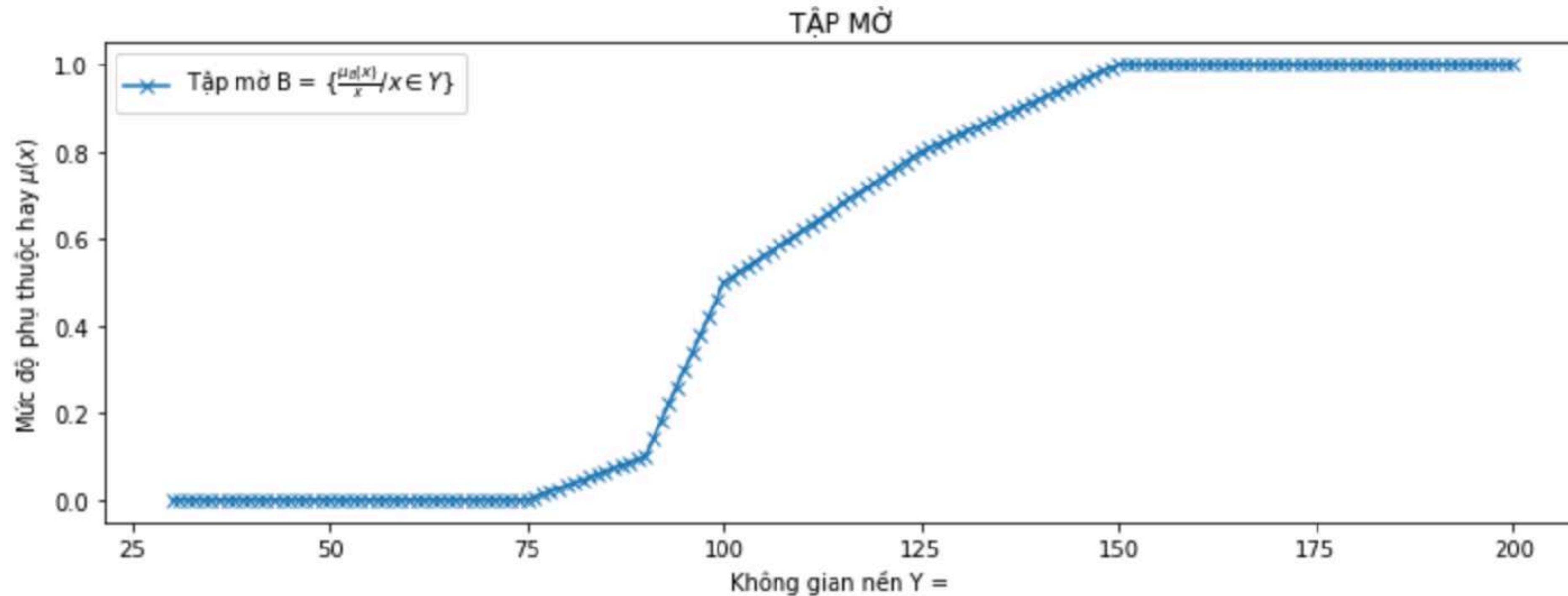
Mở rộng thêm

- Như ở Ví dụ 2 nhưng khi không gian nền dày đặc hơn, để có được hàm thuộc tương ứng, có thể sử dụng hàm nội suy trong gói *numpy* để có tập mờ
- Chẳng hạn, thay vì X như trên, ta dùng không gian nền Y có nhiều phần tử hơn, mỗi phần tử cách nhau 1 giá trị, từ đó có thể tạo ra tập mờ B được nội suy từ X và A bằng cách

```
Y = np.arange( 30, 201, 1 )  
B = np.interp( Y, X, A )
```



```
plt.figure( figsize=(12,4) )|
plt.title( "TẬP MỜ" )
plt.plot( Y, B,marker='x',label="Tập mờ B = " + r"$\{ \frac{\mu_B(x)}{x} / x \in Y \}$")
plt.xlabel( "Không gian nền Y = " )
plt.ylabel( "Mức độ phụ thuộc" + " hay " + r"$\mu(x)$" )
plt.legend( loc="best" )
plt.show()
```



Thư viện Scikit-Fuzzy

- Là một trong các gói chứa hàm phổ biến nhất để giải quyết những vấn đề cơ bản của tập mờ, logic mờ.
- Có thể tìm hiểu chi tiết tại <https://pythonhosted.org/scikit-fuzzy/overview.html>
- Để install hoặc upgrade gói này vào bằng cách
`python3 -m pip install scikit-fuzzy`
`python3 -m pip install --upgrade scikit-fuzzy`
- Hoặc kiểm tra phiên bản cập nhật của các gói đã install
`python3 -m pip list —outdated`

Sử dụng Scikit-Fuzzy

- Trước hết, cần include thêm gói thư viện *skfuzzy*, hoặc từ các module tùy theo việc sử dụng trong chương trình
- Chẳng hạn để nội suy tập mờ trên không gian nền dày đặc hơn như trong ví dụ trước, sử dụng các hàm trong modul *fuzzymath*

```
import skfuzzy as fz
Y = np.arange( 30, 201, 1 )
B = fz.interp_membership( X, A, Y )
```

- Khi đó vẫn nhận được kết quả như trường hợp dùng hàm nội suy *interp()* của thư viện *numpy*.

Một số hàm thành viên trong *skfuzzy*

- Ở phần trước xây dựng hàm thành viên trên cơ sở mảng dữ liệu là các giá trị thuộc. Khi dùng Scikit-Fuzzy, cũng có những hàm thành viên đã được tạo sẵn dưới dạng API có trong thư viện để sử dụng.
- Chẳng hạn, hình cong của các chữ như S , Z , Π ; hoặc theo hình dáng như hình tam giác, hình thang, quả chuông; hoặc theo đặc tính như hàm Sigmoid, phân phối Gauss, ...

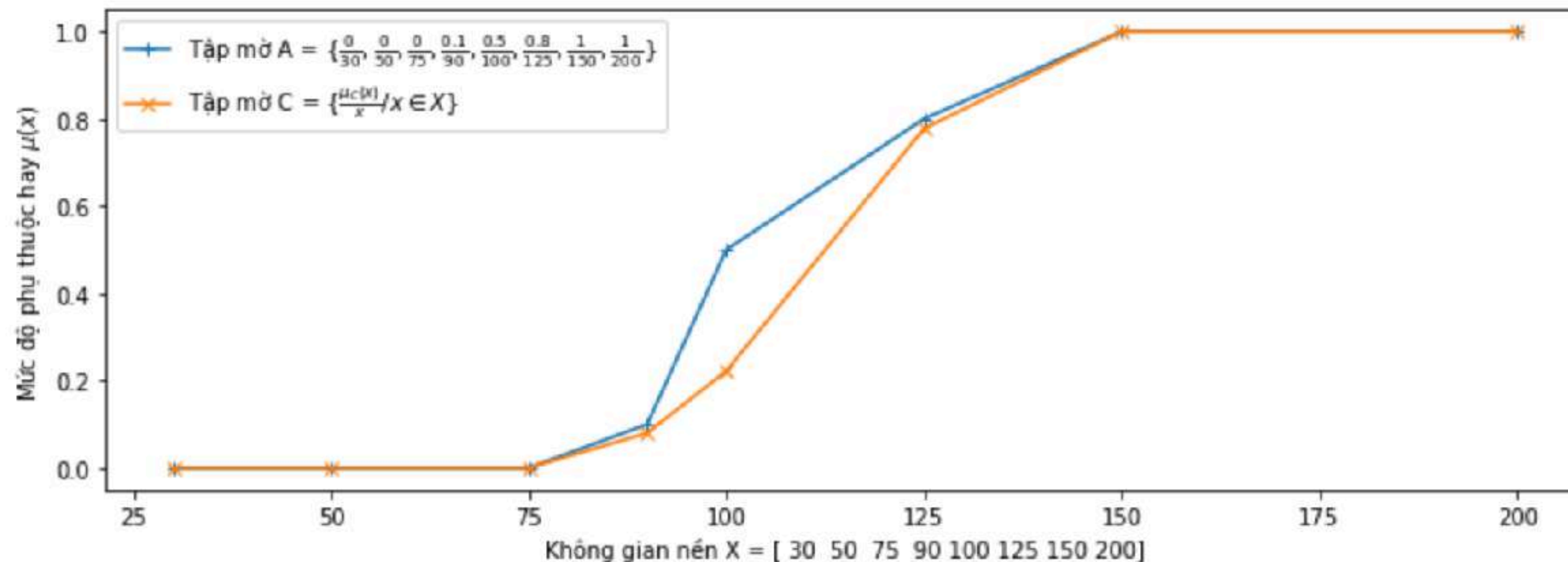
Hàm $smf()$

- Hàm $smf(X, a, b)$ này để tạo hàm thành viên của tập mờ trên không gian nền X với đồ thị có phần tử $a \in X$ có độ thuộc gần 0 và phần tử $b \in X$ có độ thuộc gần 1.
- Hình dáng của tập mờ A và B trong chương trình trước có dạng tương tự, nên có thể viết:
 - ▶ $C = smf(X, 75, 150)$ để đầu thấp của chữ S có giá trị tiến về 0, còn phía cao giá trị tiến về 1

- Kết quả tập mờ C được trực quan hoá như hình

```
C = fz.smf( X,75,150 )
```

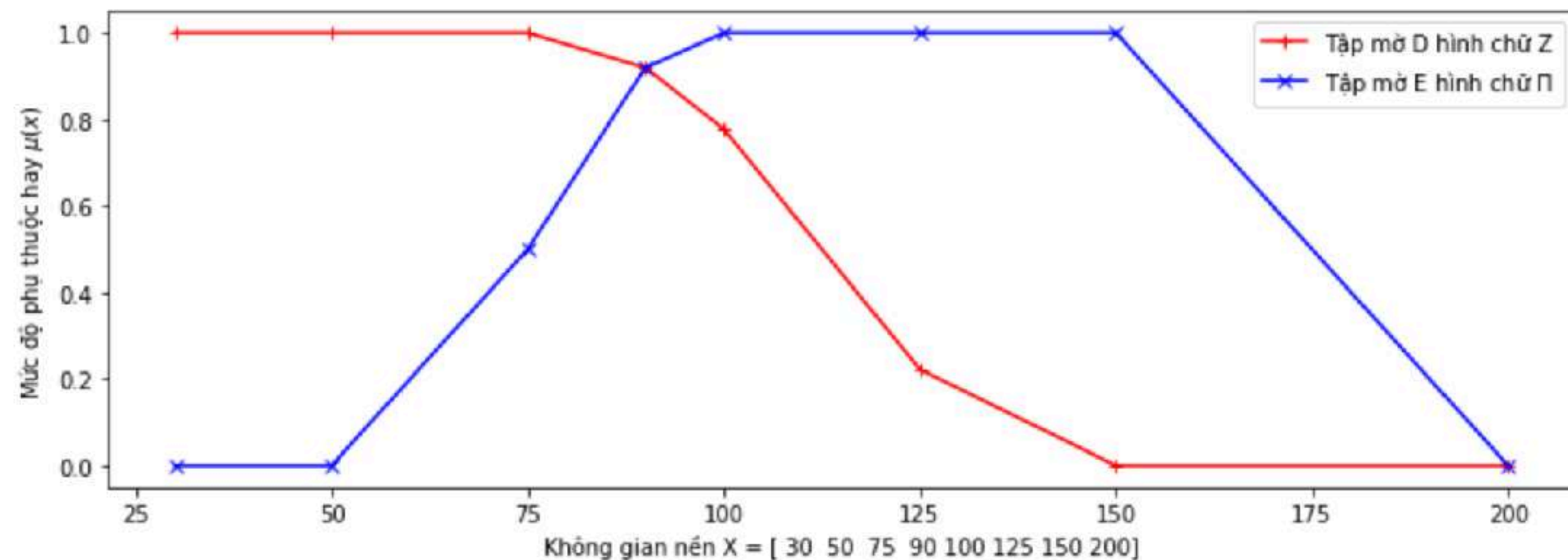
```
plt.figure( figsize=(12,4) )
plt.plot( X, A,marker='+',label="Tập mờ A = " + r"$\{\frac{0}{30},\frac{0}{50},\frac{0}{75},\frac{0.1}{90},\frac{0.5}{100},\frac{0.8}{125},\frac{1}{150},\frac{1}{200}\}$" )
plt.plot( X, C,marker='x',label="Tập mờ C = " + r"$\{\frac{\mu_C(x)}{x} / x \in X \}$" )
plt.xlabel( "Không gian nền X = " + str(X) )
plt.ylabel( "Mức độ phụ thuộc" + " hay " + r"$\mu(x)$" )
plt.legend( loc="best" )
plt.show()
```



- Tương tự như vậy, có hàm thành viên hình dạng chữ Z, chữ Π trên cùng không gian nền X : $zmf(X, a, b)$, $pimf(X, a, b, c, d)$

```
D = fz.zmf( X,75,150 )
E = fz.pimf( X,50,100,150,200 )
```

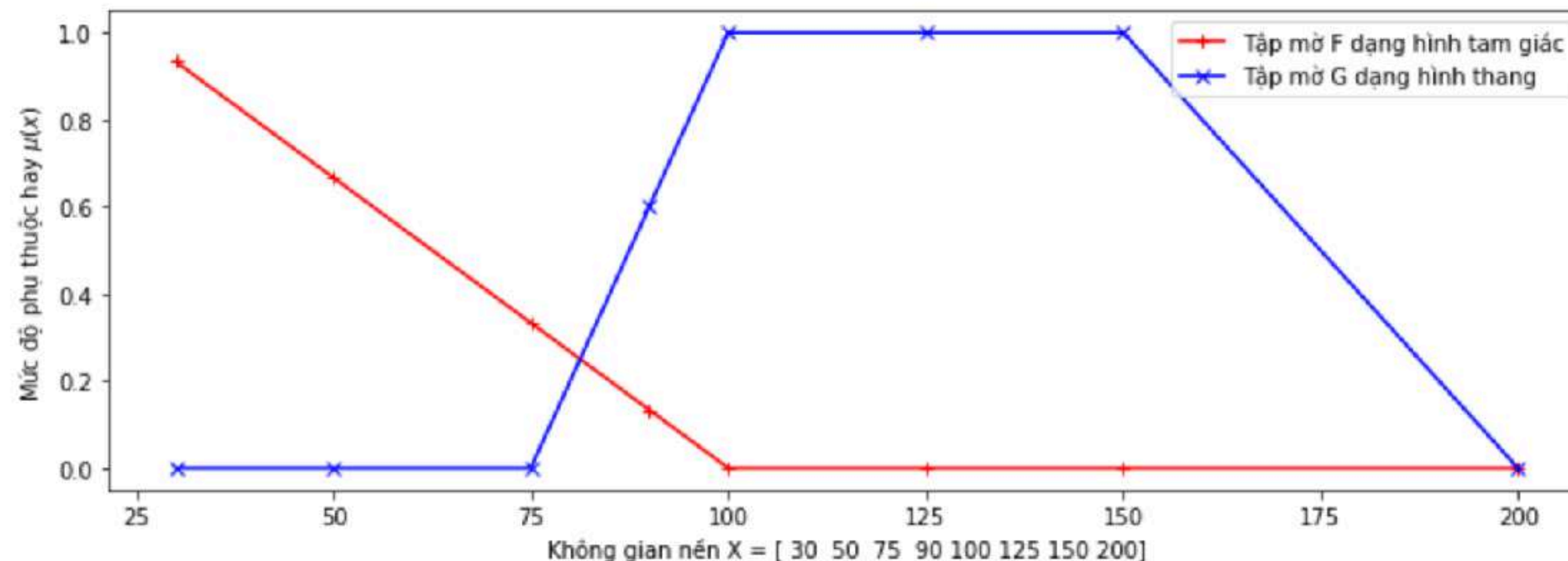
```
plt.figure( figsize=(12,4) )
plt.plot( X,D,marker='+',label="Tập mờ D hình chữ Z",color='red' )
plt.plot( X,E,marker='x',label="Tập mờ E hình chữ " + r"$\Pi$",color='blue' )
plt.xlabel( "Không gian nền X = " + str(X) )
plt.ylabel( "Mức độ phụ thuộc" + " hay " + r"$\mu(x)$" )
plt.legend( loc="best" )
plt.show()
```



- Hàm thành viên dạng hình tam giác, hình thang, hình quả chuông trên không gian nền X : $trimf(x, [a, b, c])$, $trapmf(X, [d, e, f, g])$, trong đó a, b, c là hoành độ của 3 đỉnh tam giác; d, e, f, g là hoành độ của 4 đỉnh tạo nên hình thang

```
F = fz.trimf( X, [25, 25, 100] )  
G = fz.trapmf( X, [75, 100, 175, 200] )
```

```
plt.figure( figsize=(12,4) )  
plt.plot( X, F, marker='+', label="Tập mờ F dạng hình tam giác", color='red' )  
plt.plot( X, G, marker='x', label="Tập mờ G dạng hình thang", color='blue' )  
plt.xlabel( "Không gian nền X = " + str(X) )  
plt.ylabel( "Mức độ phụ thuộc" + " hay " + r"$\mu(x)$" )  
plt.legend( loc="best" )  
plt.show()
```



Hàm $gbellmf(Y, u, v, a)$

- Hàm thành viên hình quả chuông $gbellmf(Y, u, v, a)$ được hiện thực từ hàm
$$\mu(x) = \frac{1}{1 + \left| \frac{x - a}{u} \right|^{2v}}$$
 như là hàm phối chuẩn (*Normal Distribution*) để chỉ sự phối đều đặn đối xứng về 2 phía như quả chuông.
- Đặc điểm:
 - ▶ $\lim_{x \rightarrow a} \mu(x) = 1$, hoặc $\lim_{u \rightarrow \infty} \mu(x) = 1$
 - ▶ Và khi $\left| \frac{x - a}{u} \right| > 1$ & $v \rightarrow \infty \Rightarrow \mu(x) \rightarrow 0$

- Hàm hình quả chuông và phân phối Gauss

```
H = fz.gbellmf( Y,20,2,125)  
I = fz.gaussmf( Y,125,20)
```

```
plt.figure( figsize=(12,4) )  
plt.plot( Y,H,label="Tập mờ H dạng quả chuông (phân phối chuẩn)",color='red' )  
plt.plot( Y,I,label="Tập mờ I dạng phân phối Gauss" )  
plt.xlabel( "Không gian nền X = " + str(X) )  
plt.ylabel( "Mức độ phụ thuộc" + " hay " + r"$\mu(x)$" )  
plt.legend( loc="best" )  
plt.show()
```

