



GIỚI THIỆU NGÔN NGỮ LẬP TRÌNH PYTHON

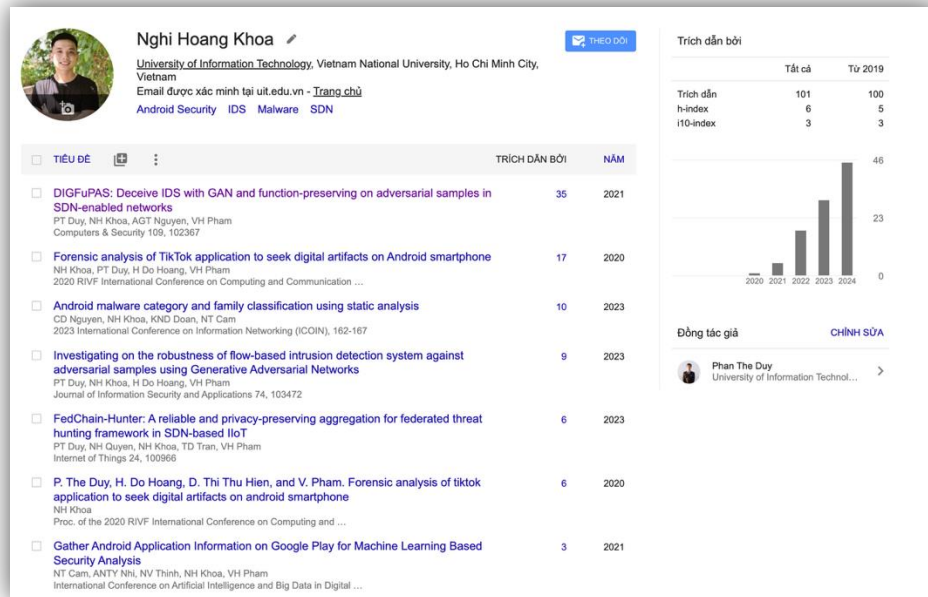
Kỹ Thuật Lập Trình Python

Giảng viên: ThS. Nghi Hoàng Khoa | Email: khoanh@uit.edu.vn



Giới thiệu bản thân

- Giảng viên: ThS. Nghi Hoàng Khoa
- Email: khoanh@uit.edu.vn
- Phòng thí nghiệm An toàn Thông tin – E8.1 UIT InsecLab
- Chuyên môn nghiên cứu :
 - Windows/Android (Malware, Forensics, Penstesting)
 - ML/AI cho Cybersecurity: Fileless, Web Assembly
 - (<https://insecLab.uit.edu.vn/cong-bo-khoa-hoc/>)
- Khoá 8 năm 2013 - UIT@VNU-HCM
- Chủ nhiệm TeamT - Câu lạc bộ An toàn Thông tin – UIT Wanna.W1n (<https://ctftime.org/team/138431>)
- Tham gia hoạt động giảng dạy thực hành các môn: An toàn Mạng, Bảo mật Web và Ứng dụng, Cơ chế hoạt động của mã độc...



- Trang course : <https://elearning.citd.vn/>
- Những gì sẽ được đăng ở đây?
 - Thông báo
 - Thông tin khóa học, quy định lớp học
 - Slide, sách, bài báo
 - Bài tập, bài lab

- Chương 01 - **GIỚI THIỆU NGÔN NGỮ LẬP TRÌNH PYTHON**
- Chương 02 - **CÁC KIỂU DỮ LIỆU CƠ BẢN**
- Chương 03 - **CẤU TRÚC Rẽ NHÁNH**
- Chương 04 - **CẤU TRÚC LẶP**
- Chương 06 - **MODULE - PACKAGE**
- Chương 07 - **XỬ LÝ NGOẠI LỆ**
- Chương 08 - **PYTHON OBJECT-ORIENTED PROGRAMMING**
- Chương 09 - **PYTHON GRAPHICAL USER INTERFACE**
- Chương 10 - **KẾT NỐI HỆ QUẢN TRỊ CƠ SỞ DỮ LIỆU**

- Vui lòng đến lớp đúng giờ. Nếu việc đi học trở thành vấn đề (ít hơn 2/3 số buổi) sẽ chuyển thành điểm
- Hãy là người trưởng thành, tôn trọng mọi người và chịu trách nhiệm trước mọi hành động của mình
- Không làm việc riêng trong lớp

- Tôi muốn bạn “thành công”
 - “thành công” có nghĩa là “học hỏi”, không nhất thiết phải “đạt điểm cao” (tùy bạn)
- Đạo đức và chuyên nghiệp:
 - Trung thực trong học tập – đừng đầu hàng trước sự cám dỗ
 - Nội dung bạn gửi phải là 100% công sức bạn bỏ ra - trích dẫn rõ ràng bất kỳ nguồn dữ liệu mà bạn tham khảo, có ảnh hưởng đến nội dung của bạn
 - Hoàn thành công việc - đừng cầu thả và lười biếng
- Tài liệu môn học (slides, readings, assignments, labs...) lưu hành nội bộ

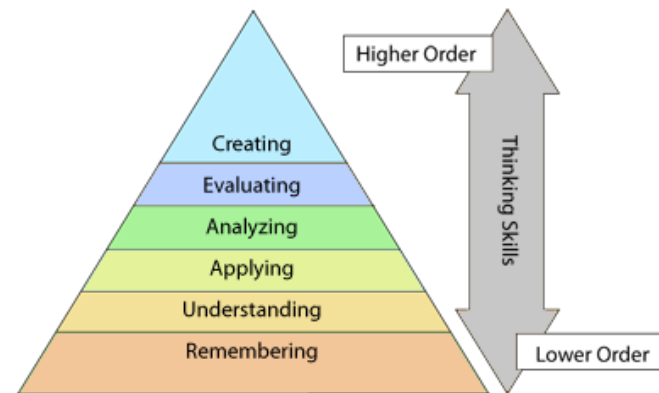
- Khuyến khích gửi mail: khoanh@uit.edu.vn
 - Vui lòng thêm tiêu đề email với “**mã lớp học – tên môn học**”
- Văn phòng: **E8.1** (Phòng thí nghiệm An toàn Thông tin – UIT InSecLab)
- Giờ làm việc: 8h -17h từ Thứ Hai đến Thứ Sáu
 - Vui lòng gửi email trước cho tôi nếu bạn định đến vào giờ hành chính
- *Nếu bạn muốn làm việc với tôi (nghiên cứu, đồ án, luận văn...), hãy tìm hiểu ở website inseclab.uit.edu.vn hoặc đề xuất ý tưởng của bạn*

Giới thiệu môn học

Câu hỏi (nếu có)



- Biết ngôn ngữ lập trình Python
- Biết môi trường hoạt động của ngôn ngữ lập trình
- Áp dụng IDE để lập trình



1. Lịch sử phát triển của Ngôn ngữ Lập trình (NNLT)
2. Ngôn ngữ lập trình Python
3. Tại sao phải dùng Python?
4. Nhu cầu sử dụng Python
5. Giới thiệu, cài đặt IDE để lập trình Python
6. Các vấn đề, hàm cơ bản cần biết trước

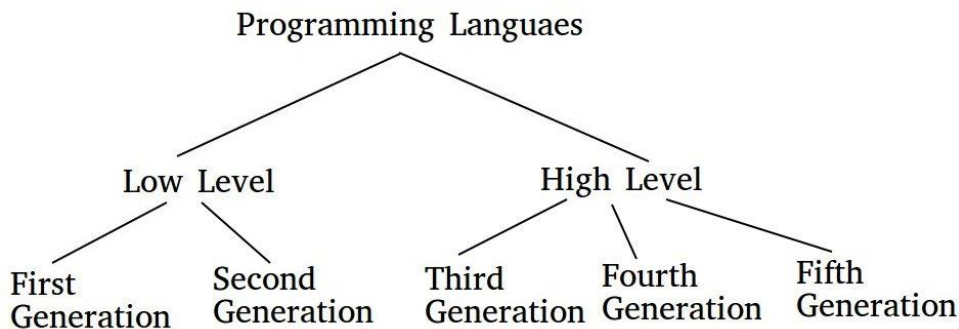
- Ngôn ngữ lập trình (Programming language)?
 - Là ngôn ngữ máy tính dùng để phát triển các chương trình phần mềm theo một hệ thống **tập lệnh** các **quy tắc riêng** để máy tính thực thi.
- Chương trình dịch là gì?
 - Trình biên dịch (compiler)
 - Trình thông dịch (interpreter)

- [illegible]

Lịch sử phát triển của NNLT



- 1GL: first-generation programming languages
- 2GL: second-generation programming languages
- 3GL: third-generation programming languages
- 4GL: fourth-generation programming languages
- 5GL: fifth-generation programming languages



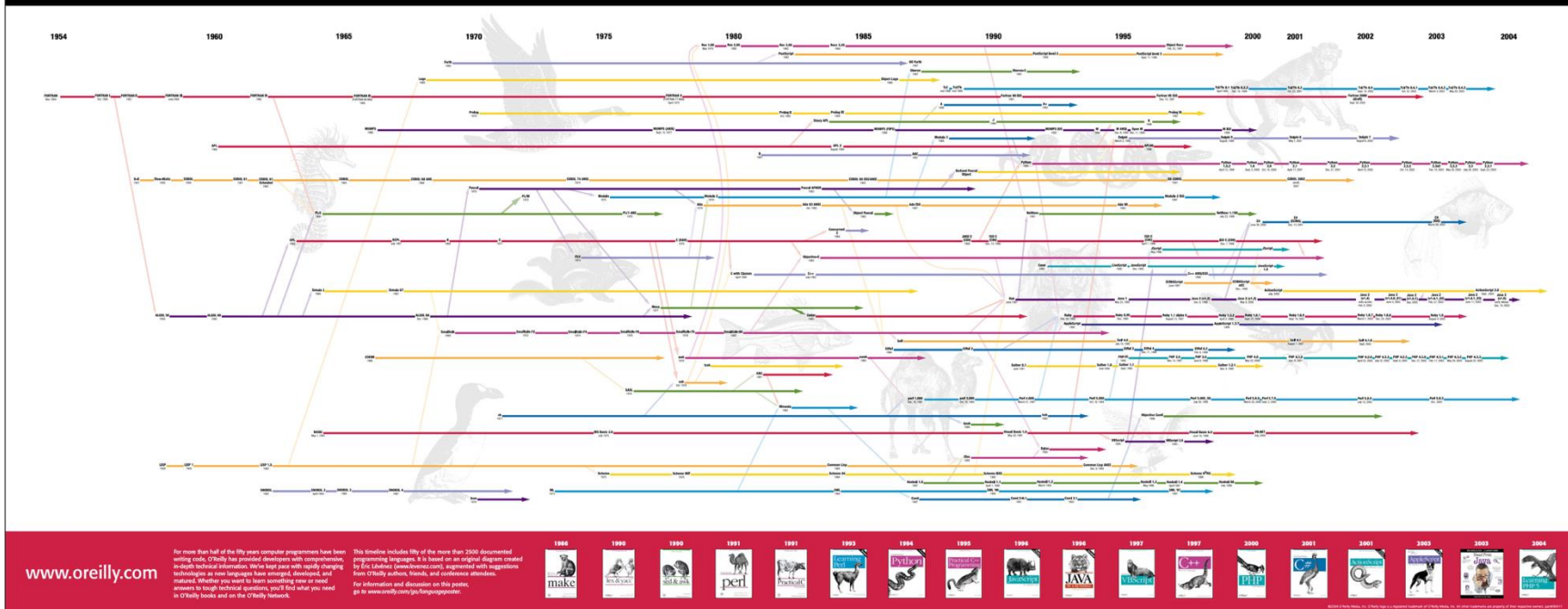
Lịch sử phát triển của NNLT



TRUNG TÂM PHÁT TRIỂN
CÔNG NGHỆ THÔNG TIN

History of Programming Languages

O'REILLY



Chi tiết: http://cdn.oreilystatic.com/news/graphics/prog_lang_poster.pdf



- **1GL:**

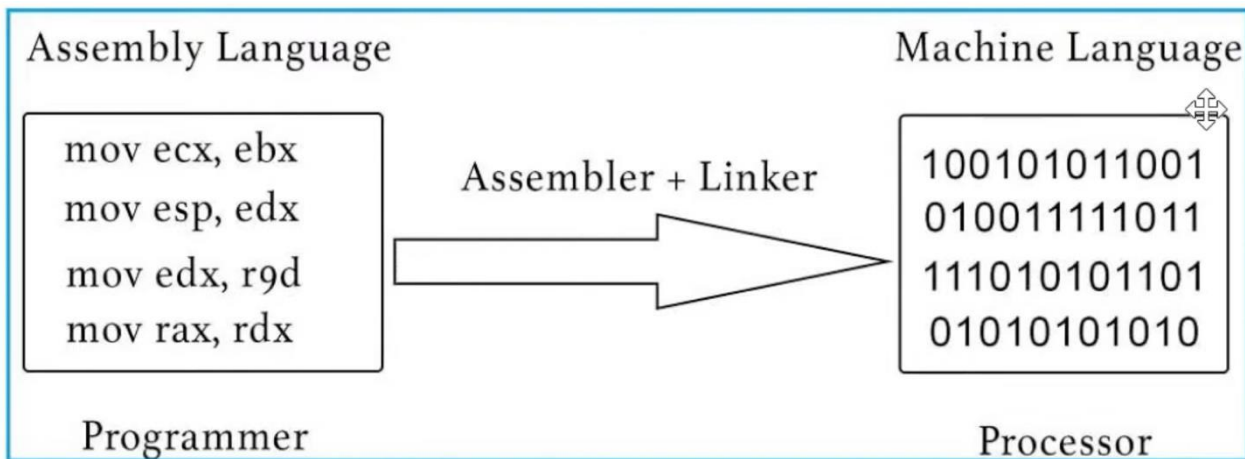
- Giai đoạn NNLT cấp độ mã máy (cấp thấp).
- Chỉ duy nhất hai trạng thái thể hiện, bao gồm nhiều con số “0” và “1”.

First Generation – Machine language

```
11010100 0011 11001101  
01011100 1010 10001111  
11001111 1010 11111110  
10000111 1011 11000001
```

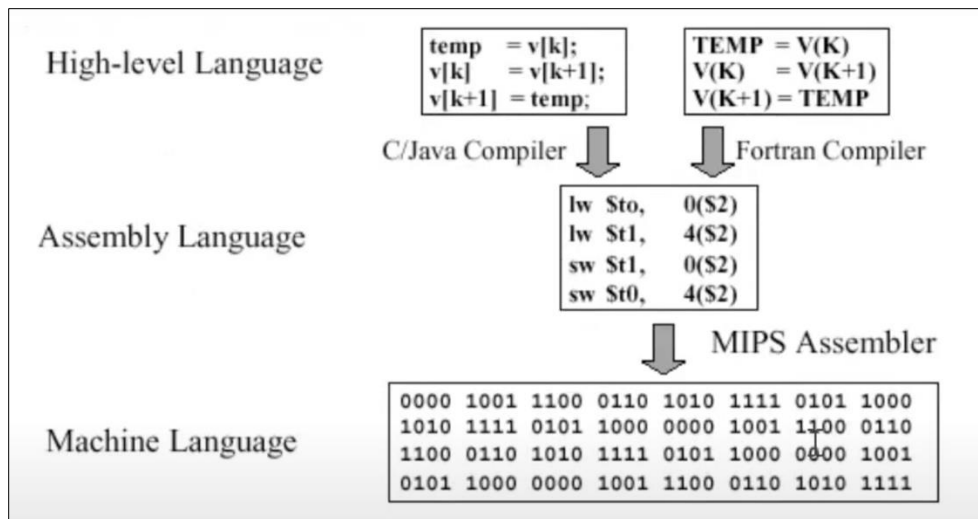

- **2GL:**

- Hợp ngữ (assembly language), cấp thấp (low-level).
- Hiện vẫn còn được sử dụng.



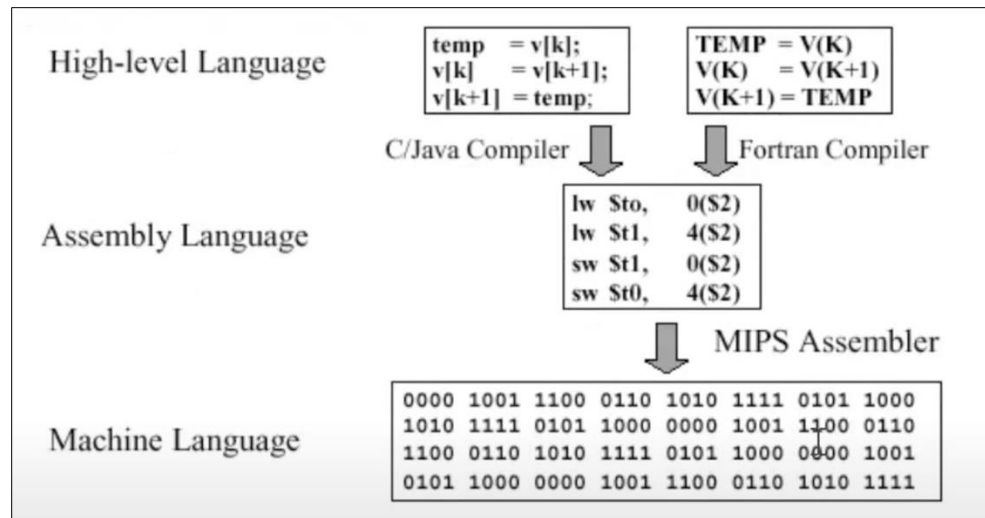
• 3GL:

- Ngôn ngữ lập trình bậc cao (high-level programming languages) như là Pascal, C, Java, Python...
- Câu lệnh của ngôn ngữ thể hệ thứ ba gần gũi với ngôn ngữ tự nhiên (logic).



• 3GL:

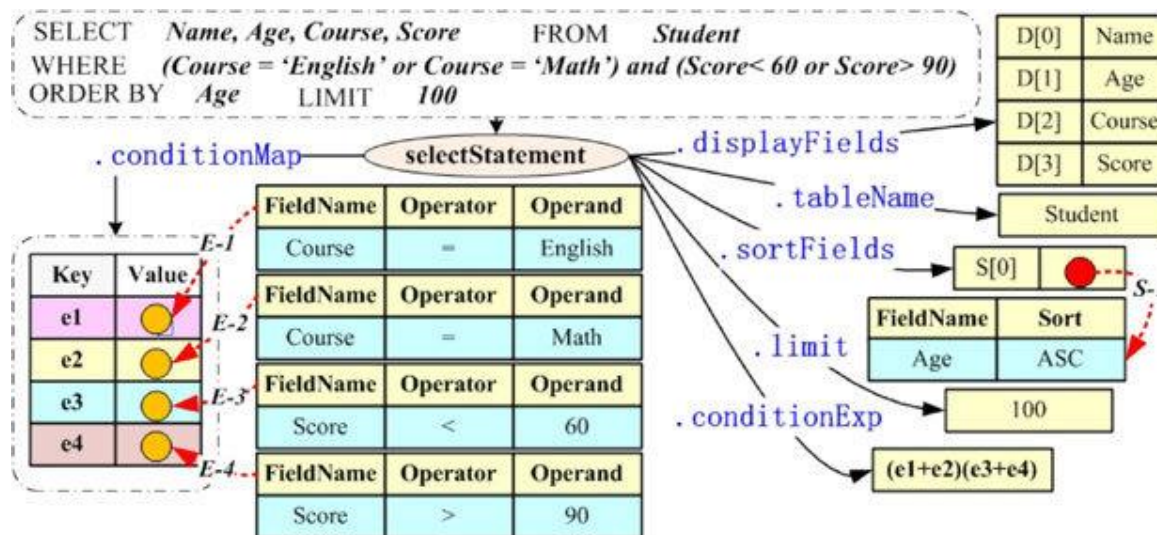
- Dễ đọc, cú pháp rõ ràng.
- Thích hợp phát triển các ứng dụng lớn hoặc ứng dụng trong lĩnh vực học thuật, khoa học.
- Không phụ thuộc vào MT.
- Linh hoạt trên nhiều platform.
- Dễ gỡ lỗi và kiểm soát lỗi.
- Tốc độ thực hiện.



Lịch sử phát triển của NNLT

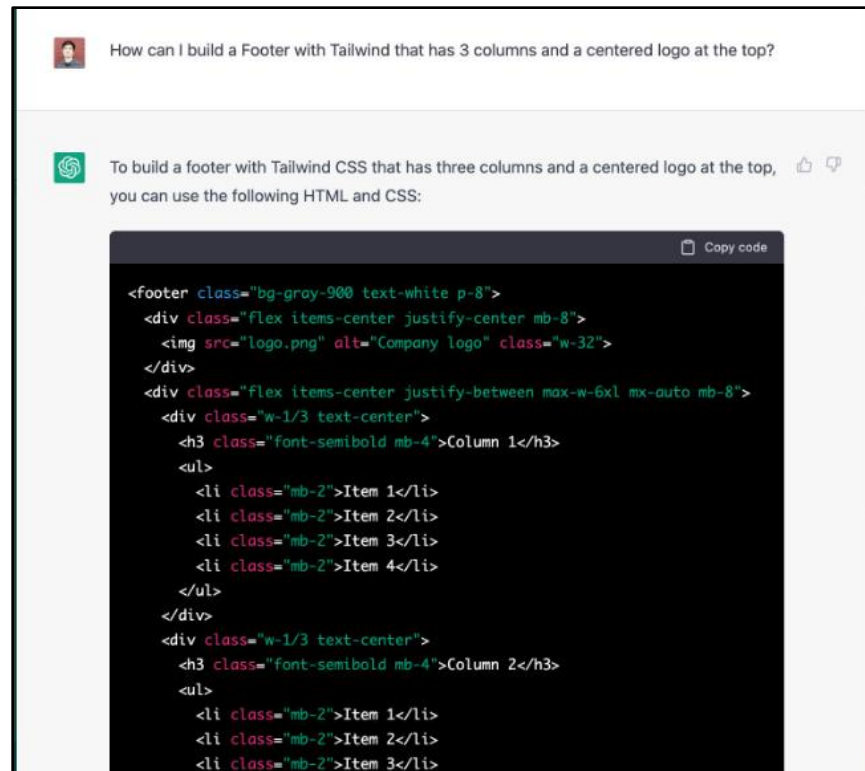
• 4GL:

- Câu lệnh tương tự như ngôn ngữ tự nhiên của con người.
- Thường được dùng trong lập trình cơ sở dữ liệu.



• 5GL:

- Đặt ra bài toán => máy tính tự giải quyết.
- Ngôn ngữ 5GL chủ yếu được sử dụng cho mục đích nghiên cứu, đặc biệt trong lĩnh vực trí tuệ nhân tạo (AI).
- Cộng đồng chuyên gia dự đoán rằng nó sẽ thay thế tất cả các ngôn ngữ khác => tương lai xa.



- Python Software Foundation <https://www.python.org/>



Version history

Python 1.0 - January 1994

Python 1.5 - December 31, 1997

Python 1.6 - September 5, 2000

Python 2.0 - October 16, 2000

Python 2.1 - April 17, 2001

Python 2.2 - December 21, 2001

Python 2.3 - July 29, 2003

Python 2.4 - November 30, 2004

Python 2.5 - September 19, 2006

Python 2.6 - October 1, 2008

Python 2.7 - July 3, 2010

Python 3.0 - December 3, 2008

Python 3.1 - June 27, 2009

Python 3.2 - February 20, 2011

Python 3.3 - September 29, 2012

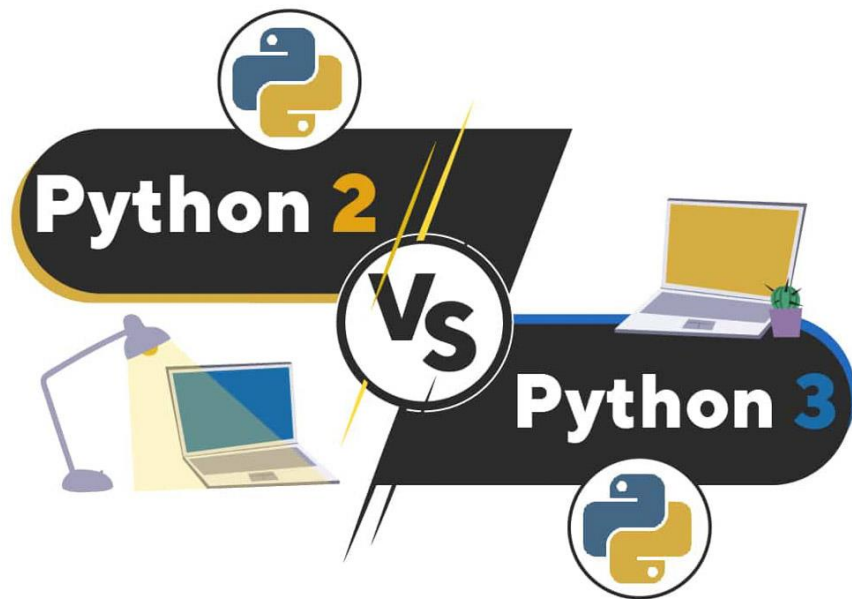
Python 3.4 - March 16, 2014

Python 3.5 - September 13, 2015

Python 3.6 - December 16, 2016

Python 3.7 – June 27, 2018

Python 3.8 – October, 2019



PYTHON 2.X



PYTHON 3.X

← LEGACY

It is still entrenched in the software at certain companies

FUTURE →

It will take over Python 2 by the end of 2019



LIBRARY

Many older libraries built for Python 2 are not forwards compatible

0100
0001

ASCII

Strings are stored as ASCII by default



$7/2=3$

It rounds your calculation down to the nearest whole number



```
print "WELCOME TO  
GEEKSFORGEEKS"
```

It rounds your calculation down to the nearest whole number



LIBRARY

Many of today's developers are creating libraries strictly for use with Python 3

0000
0000
0100
0001

UNICODE

Text Strings are Unicode by default

$7/2=3.5$



This expression will result in the expected result

```
print("WELCOME TO  
GEEKSFORGEEKS")
```



This expression will result in the expected result

Tại sao phải dùng Python

- Cú pháp đơn giản.
- Gần gũi với ngôn ngữ tự nhiên.
- Code ít hơn.



Giải thích tại sao?

Tại sao phải dùng Python

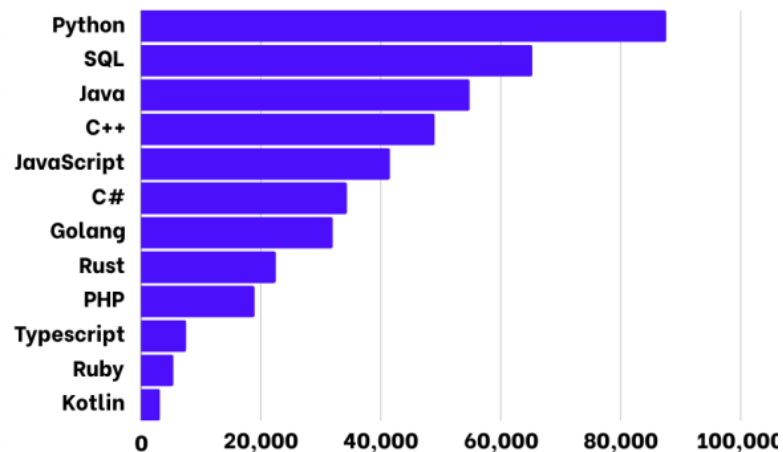
Language

Development

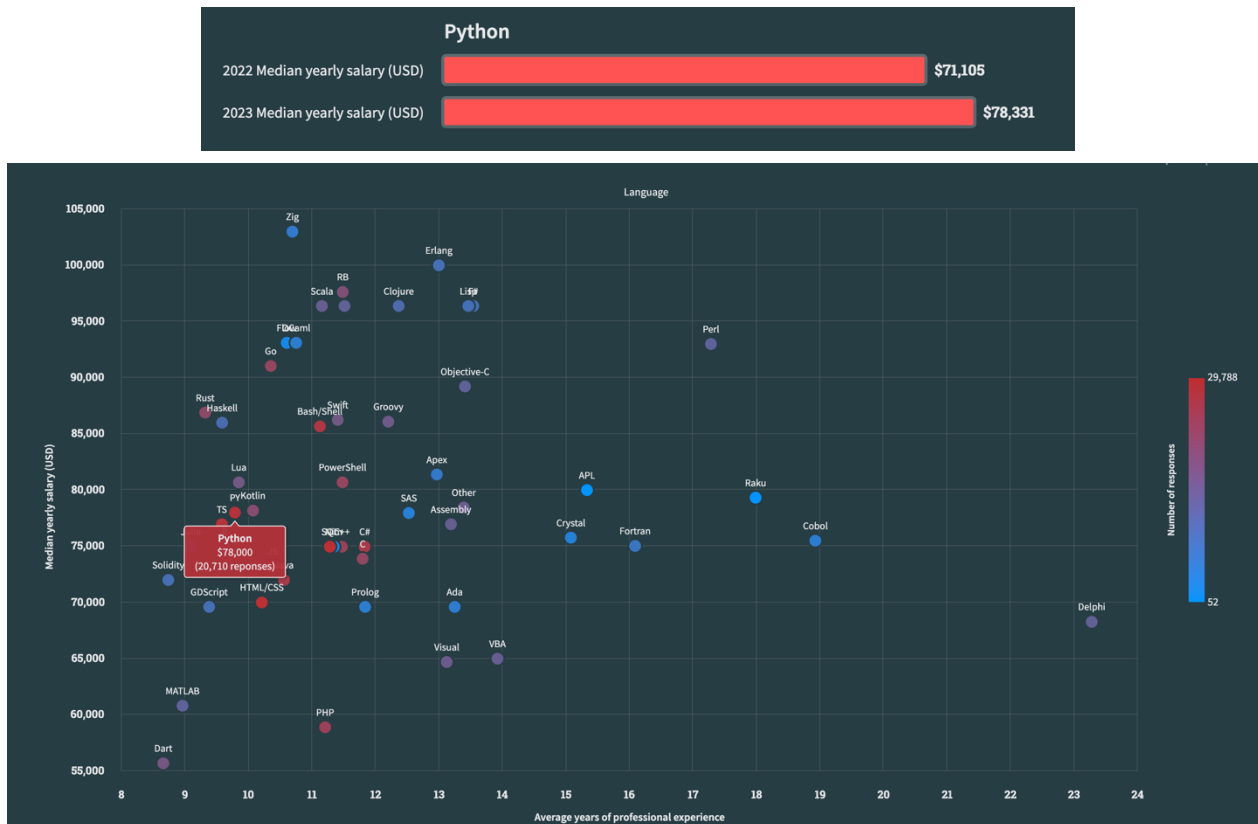
	Front-end web	Back-end web	Mobile	Game	Desktop Application	System Programming
JS JavaScript	✓	✓	✓			
Elm	✓					
TS TypeScript	✓			✓		
Scala		✓			✓	
Python		✓			✓	
Go		✓			✓	✓
Ruby		✓				
Swift			✓			
Java			✓			
C Objective C			✓			
Unity				✓		
Rust						✓

2024 Programming Language Breakdown

of open job postings for each language, in the US only



Tại sao phải dùng Python



Tại sao phải dùng Python

- Data analysis and machine learning:

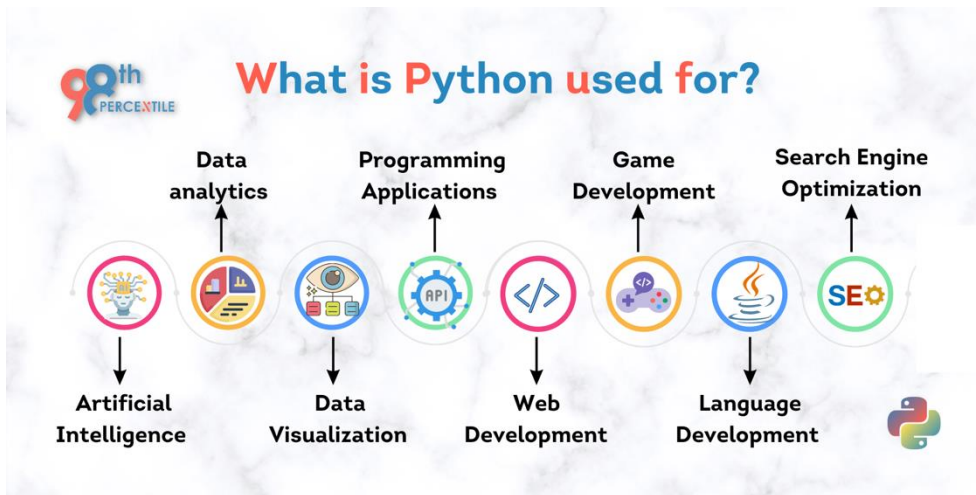


- Web development:



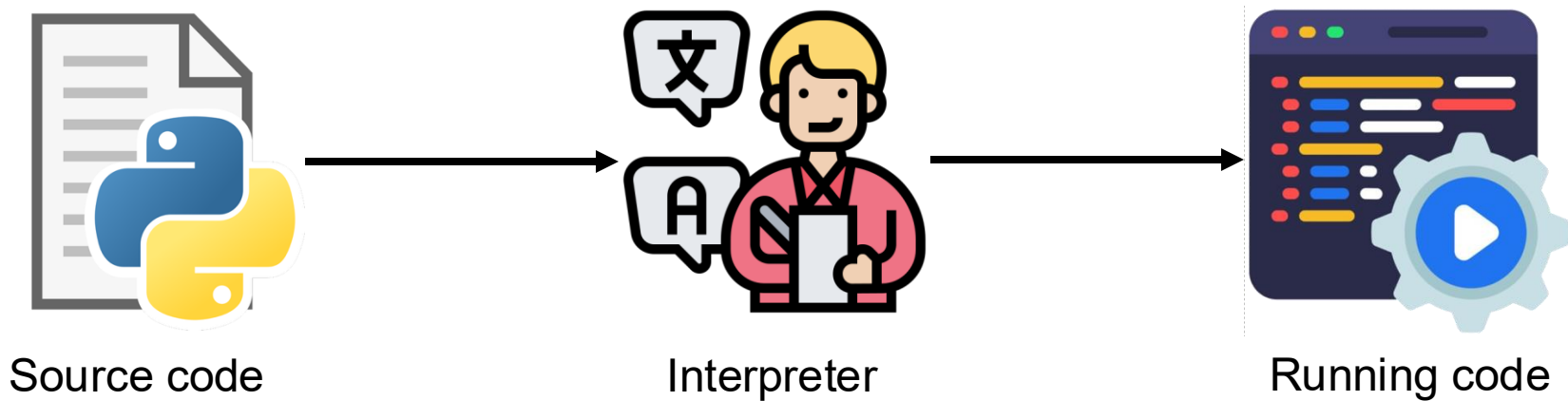
- Đã tạo ra một hệ sinh thái các Library, Module, Package hỗ trợ cho:
 - Artificial Intelligence (AI)
 - Data science
 - Machine learning
 - Deep learning
 - GIS, Remote sensing...

Nhu cầu sử dụng Python



<https://www.python.org/about/apps/>

Giới thiệu và cài đặt IDE



- **Lập trình Python bằng cách nào?**
 - Python Software Foundation <https://www.python.org/>

Download the latest version for macOS

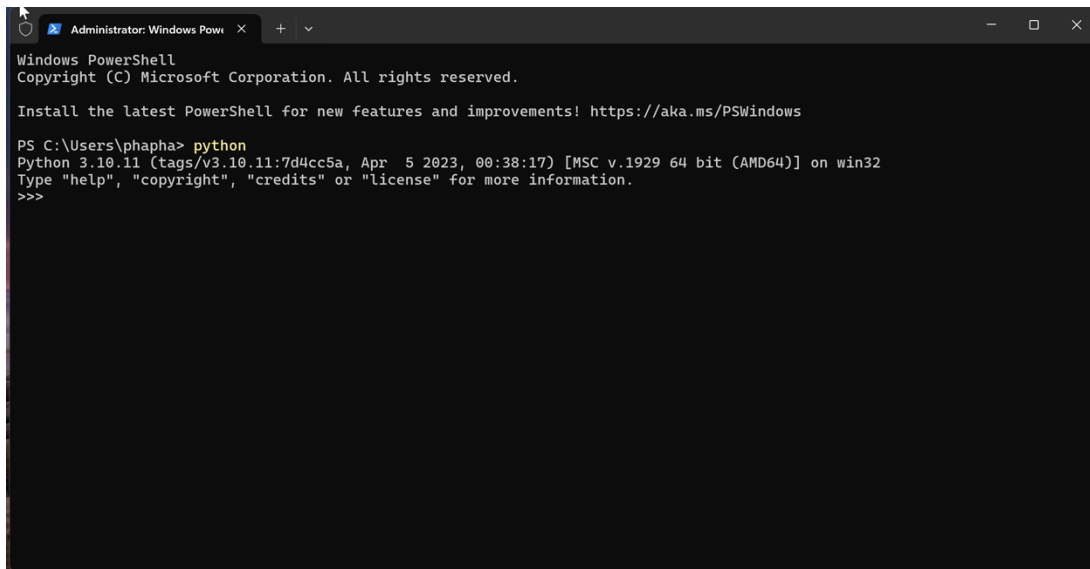
Download Python 3.13.0

Looking for Python with a different OS? Python for [Windows](#),
[Linux/UNIX](#), [macOS](#), [Other](#)

Want to help test development versions of Python 3.14? [Prereleases](#),
[Docker images](#)

Giới thiệu và cài đặt IDE

- Lập trình Python bằng cách nào?
 - Python Shell (cửa sổ dòng lệnh của Python)
 - CMD/Powershell/Terminal => python



```
Administrator: Windows Powe x + - x
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

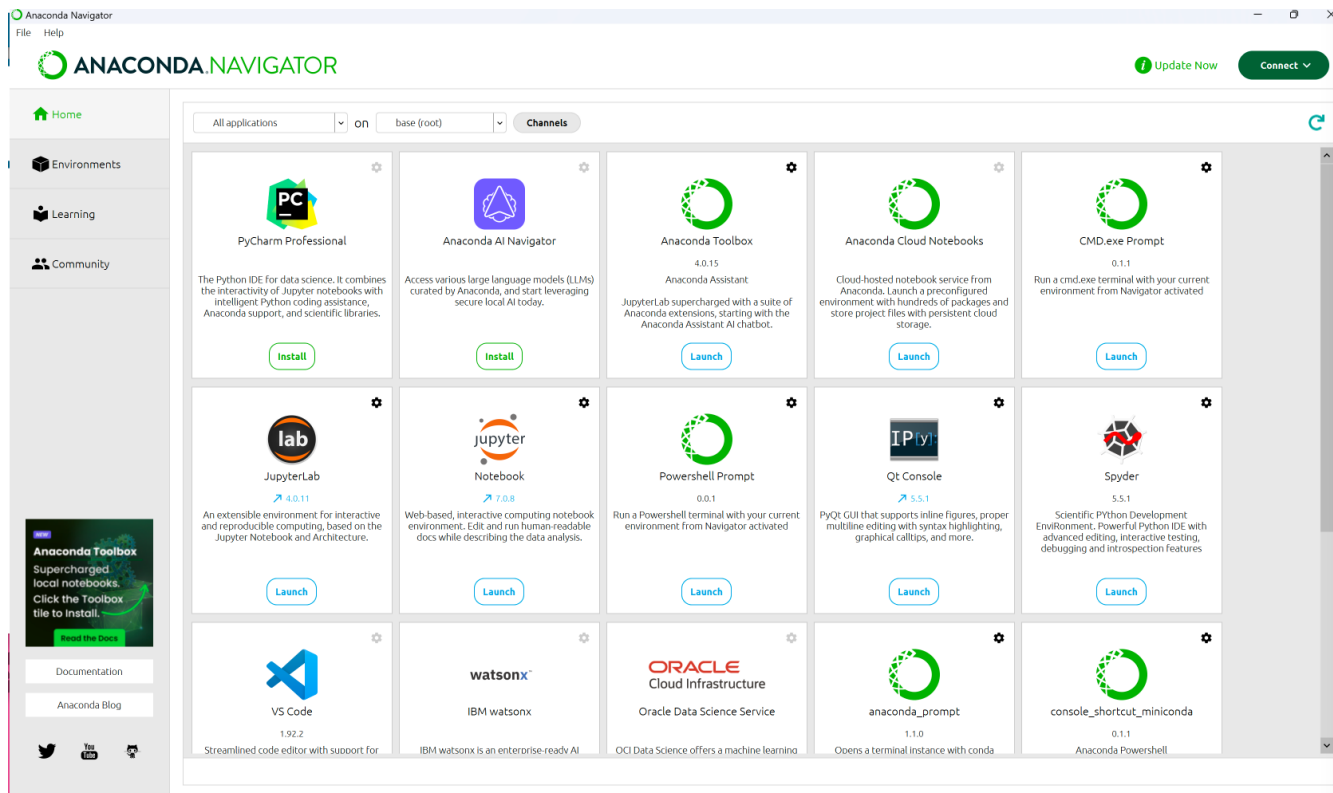
Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\phapha> python
Python 3.10.11 (tags/v3.10.11:7d4cc5a, Apr  5 2023, 00:38:17) [MSC v.1929 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>>
```

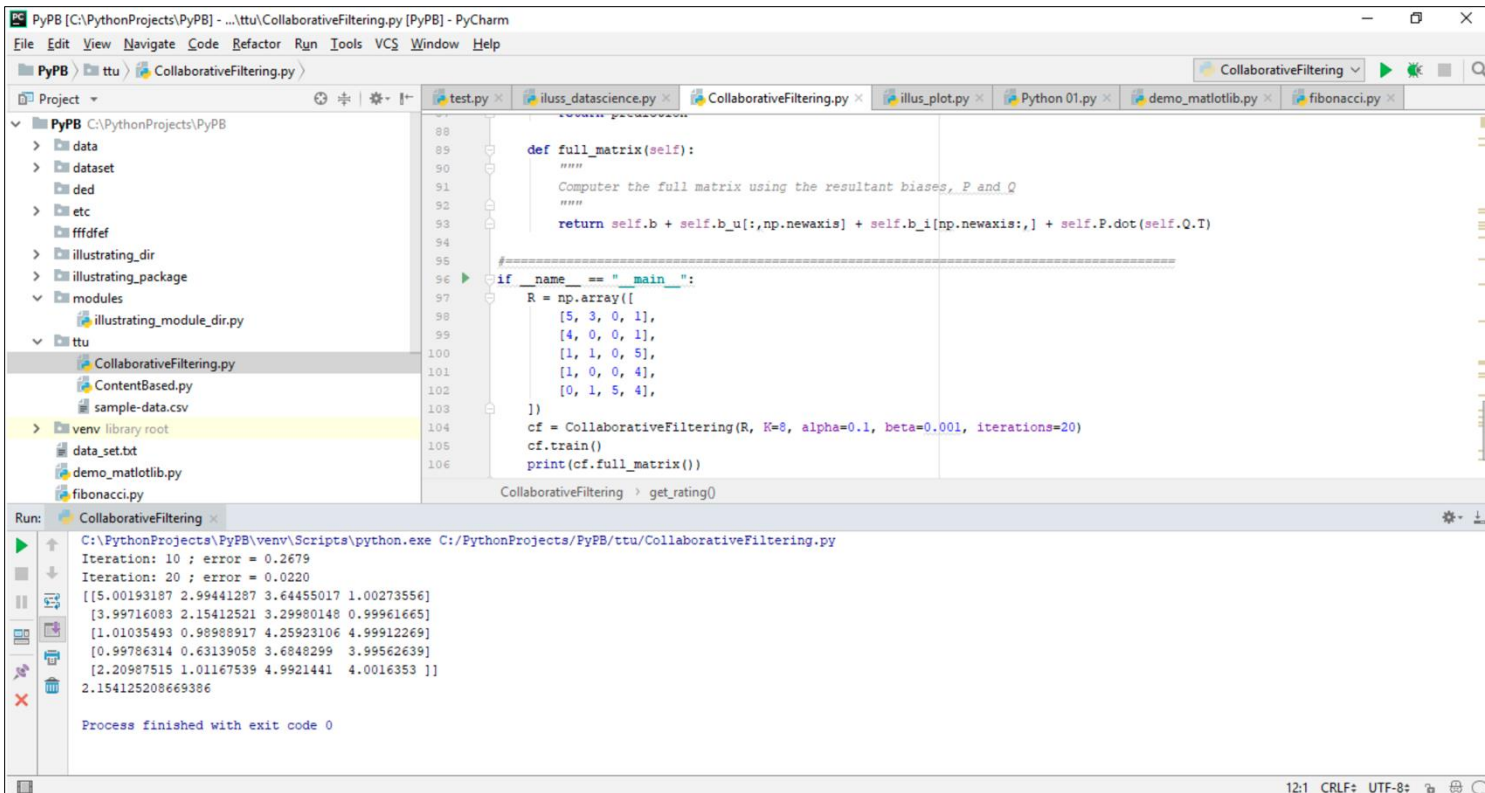
- **Integrated Development Environment (IDE) là gì?**

- Phần mềm được tích hợp nhiều công cụ như viết mã lệnh (code editor), công cụ sửa lỗi (debugger)... để hỗ trợ người sử dụng phát triển phần mềm.
- IDE hỗ trợ rất nhiều cho các NNLT phát triển ở thế hệ thứ ba, tư, và năm.
- **Hoặc, có thể hiểu IDE là một phần mềm, được tích hợp nhiều gói phần mềm khác dùng để phát triển phần mềm.**
- Ví dụ: Netbeans IDE dùng để phát triển các ngôn ngữ lập trình Java, PHP, C, C++ và HTML5 (bao gồm Javascript và CSS). Eclipse dùng cho C, C++, Python, Ruby, PHP, R...

Giới thiệu và cài đặt IDE



Giới thiệu và cài đặt IDE



The screenshot displays the PyCharm IDE interface. The top menu bar includes File, Edit, View, Navigate, Code, Refactor, Run, Tools, VCS, Window, and Help. The project explorer on the left shows the directory structure of the 'PyPB' project, including folders like 'data', 'dataset', 'ded', 'etc', 'fffddef', 'illustrating_dir', 'illustrating_package', 'modules', and 'ttu'. The code editor in the center shows the 'CollaborativeFiltering.py' file, which contains a class definition and a main function. The run console at the bottom shows the output of the program, including iteration counts, error values, and a matrix of ratings.

```
PyPB [C:\PythonProjects\PyPB] - ...\\ttu\\CollaborativeFiltering.py [PyPB] - PyCharm
File Edit View Navigate Code Refactor Run Tools VCS Window Help

PyPB | ttu | CollaborativeFiltering.py | CollaborativeFiltering.py | illuss_datascience.py | illuss_plot.py | Python 01.py | demo_matplotlib.py | fibonacci.py

Project | PyPB | C:\PythonProjects\PyPB
  | data
  | dataset
  | ded
  | etc
  | fffdfef
  | illustrating_dir
  | illustrating_package
  | modules
  | | illustrating_module_dir.py
  | | ttu
  | | CollaborativeFiltering.py
  | | ContentBased.py
  | | sample-data.csv
  | | venv library root
  | | data_set.txt
  | | demo_matplotlib.py
  | | fibonacci.py
  | Run: CollaborativeFiltering |

88
89
90 def full_matrix(self):
91     """
92     Computer the full matrix using the resultant biases, P and Q
93     """
94     return self.b + self.b_u[:,np.newaxis] + self.b_i[np.newaxis:,] + self.P.dot(self.Q.T)
95
96 #=====
97 if __name__ == "__main__":
98     R = np.array([
99         [5, 3, 0, 1],
100         [4, 0, 0, 1],
101         [1, 1, 0, 5],
102         [1, 0, 0, 4],
103         [0, 1, 5, 4],
104     ])
105     cf = CollaborativeFiltering(R, K=8, alpha=0.1, beta=0.001, iterations=20)
106     cf.train()
107     print(cf.full_matrix())
108
109 CollaborativeFiltering > get_rating()

Run: C:\PythonProjects\PyPB\venv\python.exe C:\PythonProjects\PyPB\ttu\CollaborativeFiltering.py
Iteration: 10 ; error = 0.2679
Iteration: 20 ; error = 0.0220
[[5.00193187 2.99441287 3.64455017 1.00273556]
 [3.99716083 2.15412521 3.29980148 0.99961665]
 [1.01035493 0.98988917 4.25923106 4.99912269]
 [0.99786314 0.63139058 3.6848299 3.99562639]
 [2.20987515 1.01167539 4.9921441 4.0016353 ]]
2.154125208669386

Process finished with exit code 0
```

- **IDE dùng để lập trình Python là gì?**
 - Gồm rất nhiều IDE hỗ trợ lập trình Python.
 - Khóa học này tập trung sử dụng:
 - Anaconda Navigator
 - **Jupyter Notebook**
 - Jupyter Lab
 - Spyder
 - JetBrains PyCharm Community Edition
 - **Colab** notebooks - Google Colab

- **IDE dùng để lập trình Python là gì?**

- PyCharm Community Edition hoặc Professional

- <https://www.jetbrains.com/pycharm/download>




- Anaconda => Jupyter Notebook/Lab

- https://repo.anaconda.com/archive/Anaconda3-2024.06-1-Windows-x86_64.exe

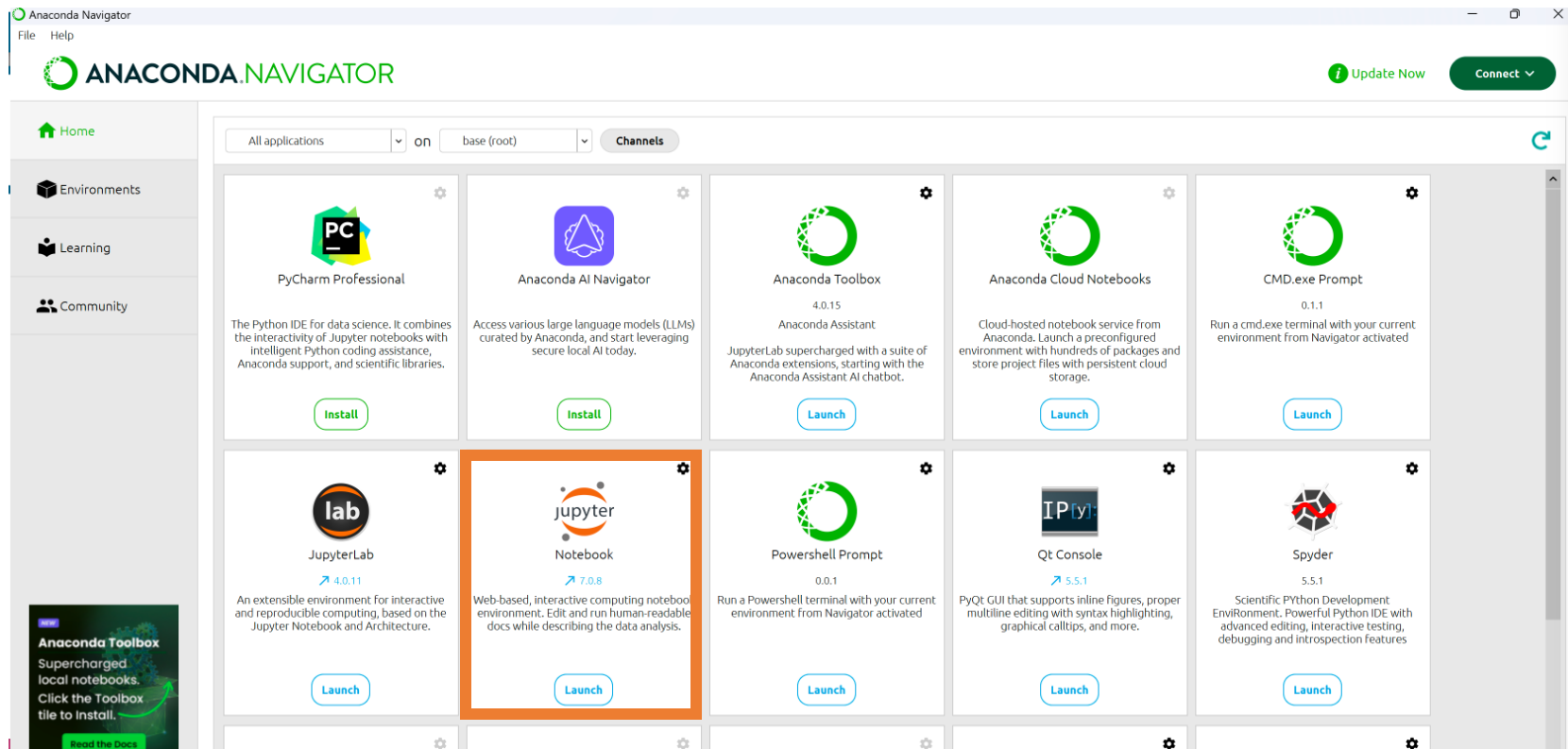
- Online => colab, azure...

- <https://colab.research.google.com/>
- <https://notebooks.azure.com/help/jupyter-notebooks>

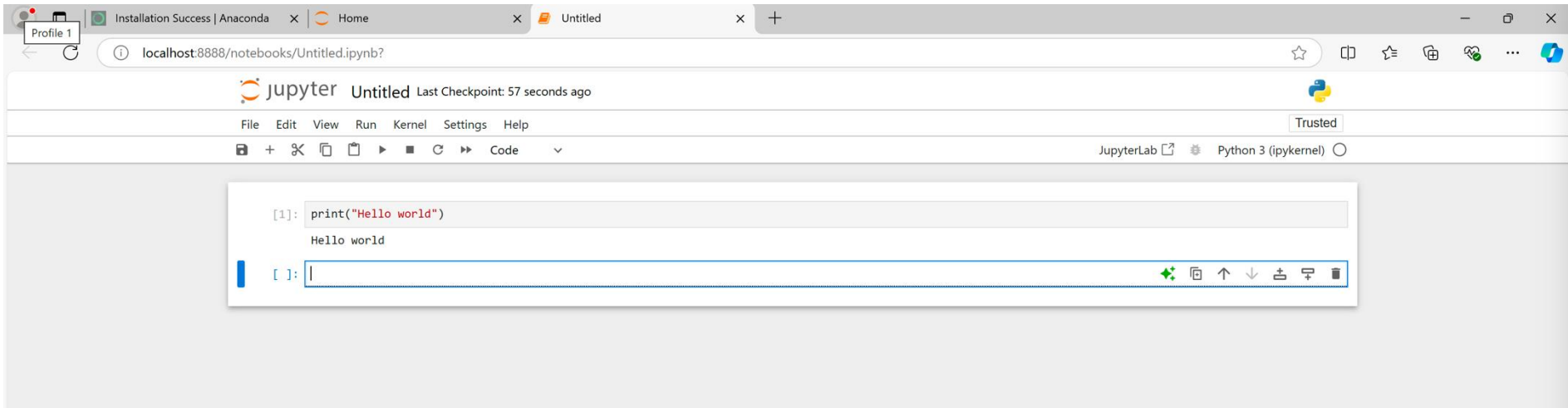
- Cách cài đặt?
 - <https://www.anaconda.com/download/success>

 Windows	 Mac	 Linux
Python 3.12	Python 3.12	Python 3.12
↓ 64-Bit Graphical Installer (912.3M)	↓ 64-Bit (Apple silicon) Graphical Installer (704.7M)	↓ 64-Bit (x86) Installer (1007.9M)
	↓ 64-Bit (Apple silicon) Command Line Installer (707.3M)	↓ 64-Bit (AWS Graviton2 / ARM64) Installer (800.6M)
	↓ 64-Bit (Intel chip) Graphical Installer (734.7M)	↓ 64-bit (Linux on IBM Z & LinuxONE) Installer (425.8M)
	↓ 64-Bit (Intel chip) Command Line Installer (731.2M)	

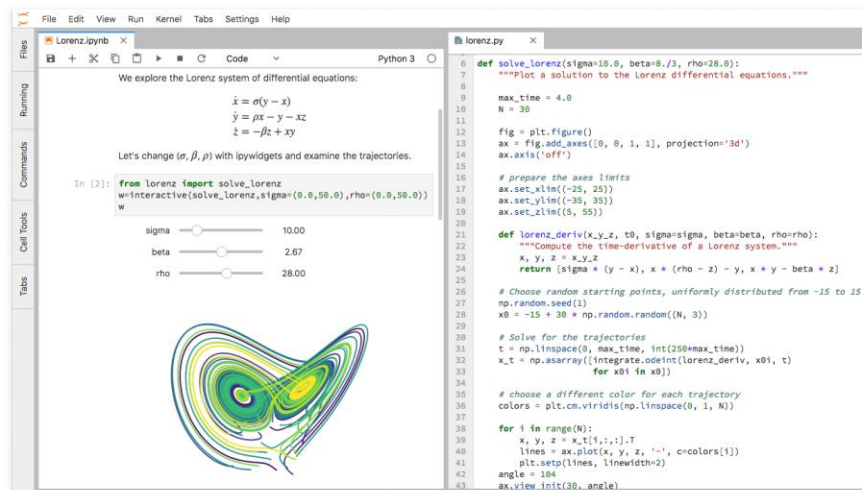
Anaconda Navigator



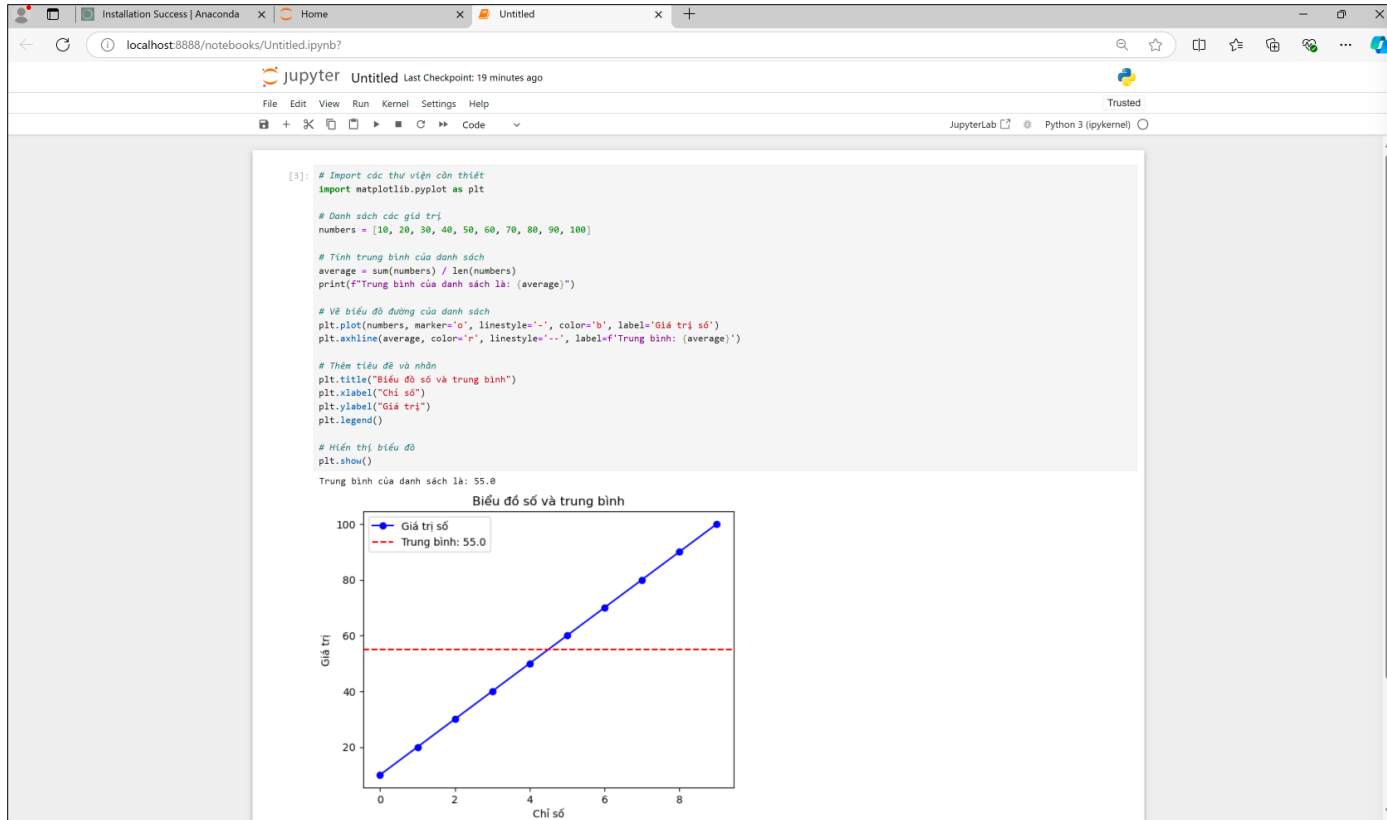
Anaconda Navigator



- Jupyter notebook là gì?
 - Ứng dụng web mã nguồn mở để tạo và chia sẻ các hướng dẫn/minh họa về lập trình. Gồm: câu lệnh lập trình, biểu thức, phương trình, công thức toán, kèm hình ảnh, văn bản.
 - Tích hợp nhiều ngôn ngữ lập trình.
 - Thích hợp blogger viết các instruction



Jupyter Notebook



- **Cách cài đặt KHÁC?**

- Bước 1: cài Python 3 (Ghi nhớ đường dẫn lưu trữ Interpreter)
- Bước 2: vào CMD, lệnh `pip install jupyter notebook`
- Run: `jupyter notebook`

- **Tìm hiểu cách sử dụng các IDE: PyCharm, Anaconda (Jupyter Notebook/Lab):**



30 phút

- Xác định xem trong quá trình lập trình, chúng ta thường xử dụng và lập đi lập lại thao tác nào nhiều nhất từ đó tìm các hotkey liên quan.
 - Các hotkey thường dùng về Run...New...Code...Comment...
- **Phương án:**
 - Cho sinh viên thực hiện cá nhân => ghi chú lại => trình bày

Function	Keyboard Shortcut	Menu Tools
Save notebook	Esc + s	File → Save and Checkpoint
Create new cell	Esc + a (above), Esc + b (below)	Insert → cell above Insert → cell below
Run Cell	Shift + enter	Cell → Run Cell
Copy Cell	c	Copy Key
Paste Cell	v	Paste Key
Interrupt Kernel	Esc + i i	Kernel → Interrupt
Restart Kernel	Esc + 0 0	Kernel → Restart
Find and replace on your code but not the outputs	Esc + f	N/A
merge multiple cells	Shift + M	N/A

**Gợi ý
Jupyter**

• Gọi ý PyCharm

- Alt + Shift + F10
- Alt + Shift + F9
- Shift + F10
- Shift + F9
- Ctrl + /

- VĐ #1: Biến số và phép gán?
- VĐ #2: Hàm cơ bản có sẵn trong Python?
- VĐ #3: Cách comment trong Python?
- VĐ #4: Cách viết khối lệnh trong Python?

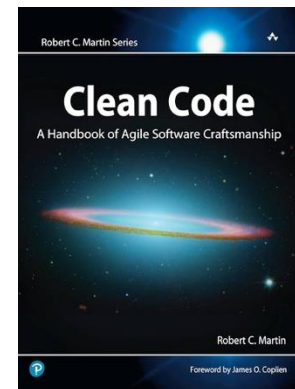
- **Biến số?**

- Là nơi chứa các giá trị.
- Có thể thay đổi trong quá trình thực hiện.
- Cách khai báo/khởi tạo biến: dùng dấu “=”

- **Cách khai báo**

- Biến khai báo rất đơn giản.
- Cú pháp:
 - tên_biến = <giá trị>
- Cú pháp xóa biến:
 - **del** tên_biến

- Cách đặt tên biến?
 - Phân biệt “**CHỮ HOA**” và “**chữ thường**”
 - **KHÔNG** sử dụng ký tự **@, \$ và %** đặt tên biến, hàm...
 - **KHÔNG** khoảng trắng
 - **KHÔNG** bắt đầu bằng số
 - **KHÔNG** đặt trùng với từ khóa
 - Đặt làm sau dễ nhớ.
 - VD: SoLuongNhapKho, so_luong_nhap_kho, x, X...



- **Ví dụ:**

- GV minh họa trong PyCharm và **Jupyter**.
- Sau đó, SV thực hiện theo.

- Hàm **print()**
- Hàm **input()**
- Hàm **type(<object* cần kiểm tra>)**

- **print()**

- Hàm **print(<giá trị hoặc chuỗi cần in>)**

- Kết quả in, xuất, hiển thị giá trị ra màn hình.
- Dùng dấu phẩy “,” hoặc cộng “+” để kết chuỗi.
- Cách xử dụng nâng cao hàm **print()** sẽ được học sau.
- Ví dụ:

- `print("Khai giảng lớp Python S1")`

- `print('Khai giảng lớp Python S1')`

- **input()**

- Hàm **input(<chuỗi thông báo nếu có>)**
- Nhập vào giá trị từ keyboard.
- Kiểu giá trị nhận được là **string**.
- Ví dụ minh họa, SV thực hiện theo

```
# Yêu cầu người dùng nhập tên  
name = input("Nhập tên của bạn: ")
```

```
# Hiển thị lời chào với tên vừa nhập  
print(f"Chào mừng, {name}!")
```

- **type()**

- Hàm **type(object)**
- Trả về kiểu dữ liệu của **object** cần kiểm tra.
- Ví dụ: `type("chuỗi")`
- Kết quả: `<class 'str'>`

- Dùng dấu # để chứa các nội dung comment.
- Chú ý các hotkey để thực hiện cho nhanh. Trong Pycharm: **Ctrl + /**

Đây là chú thích và sẽ không được xem là câu lệnh.

- Cách viết khối lệnh trong Python

```
x = 1
if x == 1:
    print("Khối lệnh 1. Dòng 1.")
    print("Khối lệnh 1. Dòng 2.")
    print("Khối lệnh 1. Dòng 3.")
else:
    print("Khối lệnh 2. Dòng 1.")
    print("Khối lệnh 2. Dòng 2.")
    print("Khối lệnh 2. Dòng 3.")
```

- Cách viết khối lệnh trong Python

```
x = 1
if x == 1:
    print("Khối lệnh 1. Dòng 1.")
    print("Khối lệnh 1. Dòng 2.")
    print("Khối lệnh 1. Dòng 3.")
else:
    print("Khối lệnh 2. Dòng 1.")
    print("Khối lệnh 2. Dòng 2.")
    print("Khối lệnh 2. Dòng 3.")
```

```
Cell In[1], line 3
    print("Khối lệnh 1. Dòng 1.")
    ^
```

SyntaxError: invalid non-printable character U+00A0

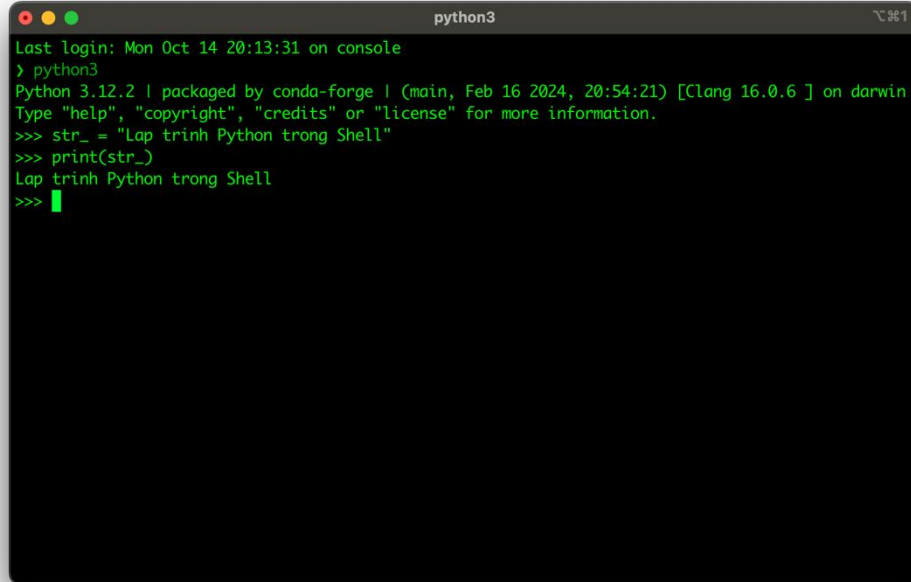
- **Cách viết khối lệnh trong Python**

- Viết nhiều câu lệnh trên cùng một dòng, các câu lệnh ngăn cách nhau bởi dấu “;”
- Ví dụ:

```
var = 8; str_ = "Hello"; dict_ = {}
```

- **Mục tiêu các bài tập này giúp học viên phản xạ, làm quen với: Jupyter, PyCharm, Colab**
 1. Tạo tài khoản sử dụng Jupyter **Notebook/Lab** trong Azure, **Colab** google hoặc trên máy tính VS Code + Jupyter **Notebook**
 2. Viết CT nhập thông tin từ bàn phím, sau đó in ra chuỗi, rồi kiểm tra kiểu dữ liệu nhập vào.
 3. Viết chương trình nhập vào hai string. Sau đó dùng “,” hoặc “+” trong print() để kết hai string đó và xuất ra màn hình.
 4. Viết lệnh nhập thông tin từ bàn phím, rồi xuất ra giá trị nhập, viết lệnh input() lồng trong lệnh print().

- Sử dụng lại các bài tập trên nhưng thực hành trên Python Shell
 - Rút ra kết luận gì khi lập trình Python trên shell?
 - Nên dùng Python shell trong trường hợp nào?



```
python3
Last login: Mon Oct 14 20:13:31 on console
> python3
Python 3.12.2 | packaged by conda-forge | (main, Feb 16 2024, 20:54:21) [Clang 16.0.6 ] on darwin
Type "help", "copyright", "credits" or "license" for more information.
>>> str_ = "Lập trình Python trong Shell"
>>> print(str_)
Lập trình Python trong Shell
>>> █
```

Hôm nay, kết thúc!

- Nghi Hoàng Khoa
- khoanh@uit.edu.vn
- inseclab.uit.edu.vn