



HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG  
Posts and Telecommunications Institute of Technology

# CHƯƠNG 0: GIỚI THIỆU MÔN HỌC LẬP TRÌNH PYTHON

VŨ HOÀI THU  
CNTT1. PTIT

# THÔNG TIN MÔN HỌC

---

- ❑ Tên môn học: Lập trình Python (Python Programming)
- ❑ Số lượng tín chỉ: 3

# YÊU CẦU VỚI MÔN HỌC

---

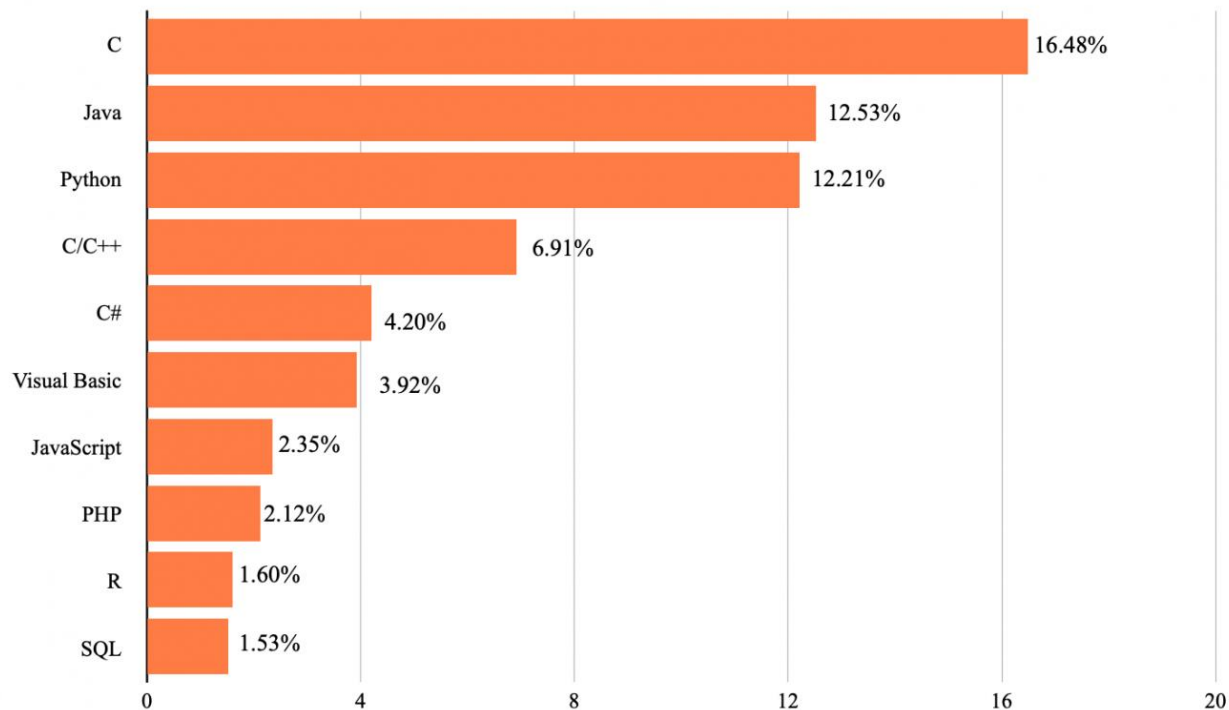
- ❑ Mỗi chương đều có bài tập và yêu cầu sinh viên hoàn thành tất cả các bài tập trước khi sang chương mới.
- ❑ Thiếu một điểm thành phần (bài tập, điểm kiểm tra giữa kỳ), hoặc nghỉ quá 20% tổng số tiết học thì sẽ không được thi hết môn.

# PHƯƠNG PHÁP ĐÁNH GIÁ

Hình thức kiểm tra	Tỷ lệ đánh giá	Đặc điểm đánh giá
<ul style="list-style-type: none"><li>- Đi học đầy đủ (trong lớp không gây ồn ào ảnh hưởng đến người khác, mỗi lần nhắc nhở trừ một điểm; Mỗi buổi nghỉ học không có lý do chính đánh trừ một điểm)</li><li>- Tích cực thảo luận và hăng hái trong các tiết học (không phát biểu buổi nào sẽ được 0 điểm, phát biểu 1 buổi sẽ được 4 điểm, sau đó số buổi học có phát biểu tăng lên 1 thì điểm tăng lên 1)</li></ul>	10%	Cá nhân
Trung bình kiểm tra (trung bình điểm các bài tập trên hệ thống <a href="https://code.ptit.edu.vn">https://code.ptit.edu.vn</a> )	10%	Cá nhân
Điểm bài tập lớn	20%	Nhóm
Kiểm tra cuối kỳ	60%	Cá nhân

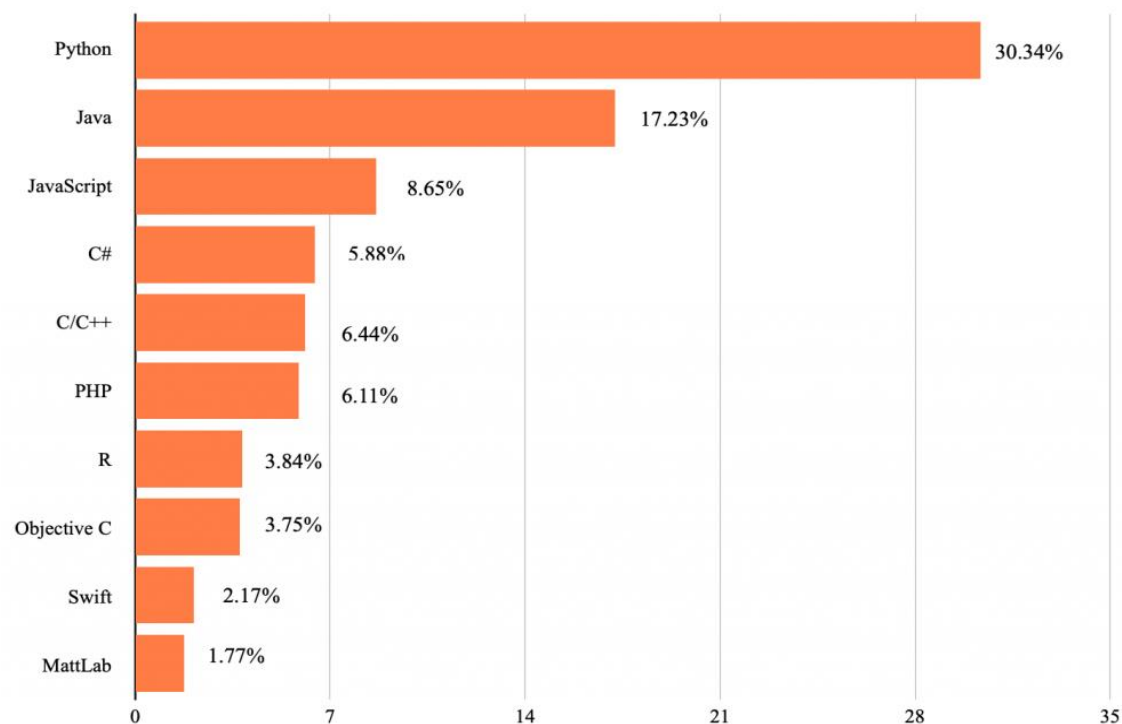
# TẠI SAO CẦN HỌC PYTHON

■ The popularity of Programming Language (TIOBE) Ranking 2020



Chỉ số Cộng đồng lập trình TIOBE

■ The popularity of Programming Language (PYPL) Ranking 2020



Chỉ số đo tần suất xuất hiện của ngôn ngữ lập trình trên công cụ tìm kiếm Google

# TẠI SAO CẦN HỌC PYTHON (1)

---

- ❑ Python là một mã nguồn mở (open source), miễn phí và là một ngôn ngữ lập trình dựa trên kịch bản, dễ học dễ nhớ.
- ❑ Python được sử dụng phát triển các ứng dụng phía máy chủ cũng như để phát triển các ứng dụng độc lập.
- ❑ Các ứng dụng có tích hợp sẵn ngôn ngữ lập trình Python phải kể đến như: Instagram, Youtube, Printest,...
- ❑ Python có cộng đồng nhà phát triển lớn, có thể kể đến như StackOverflow, CodeAcademy.

## TẠI SAO CẦN HỌC PYTHON (2)

---

- ❑ Python có một bộ sưu tập lớn các chức năng sẵn sàng được sử dụng, được gọi là các thư viện chuẩn.
- ❑ Nhiều công cụ phát triển của bên thứ ba cho ngôn ngữ lập trình Python cho phép chúng ta sử dụng ngôn ngữ đó cho các tác vụ khác nhau, chẳng hạn như phát triển trang web, phát triển trò chơi, lập trình tương đương Matlab và các công việc khác.
- ❑ Mã nguồn được viết bằng Python thường không yêu cầu bất kỳ thay đổi nào để thực thi trên một nền tảng máy tính khác. Chuyển từ nền tảng này sang nền tảng khác là chuyển tiếp thẳng

# TẠI SAO CẦN HỌC PYTHON

---

- ❑ Python hỗ trợ rất tốt cho hầu hết các cơ chế lập trình hướng đối tượng phổ biến.
- ❑ Python là một ngôn ngữ lập trình phổ biến và thú vị, nó đã được sử dụng bởi hơn 1 triệu nhà phát triển trên khắp thế giới (2018)
- ❑ Quản lý bộ nhớ trong Python là tự động. Bộ gom rác theo dõi tất cả các việc cấp phát bộ nhớ.
- ❑ Python là một ngôn ngữ lập trình kiểu động, không cần khai báo các biến có kiểu cụ thể và kích thước cụ thể.



# LỊCH SỬ PYTHON



[Guido van Rossum](#), nhà sáng lập ra ngôn ngữ Python.

# LỊCH SỬ PYTHON

---

- ❑ Python lần đầu tiên được giới thiệu vào năm 1989 bởi Guido van Rossum tại CWI như một sự kế thừa cho ngôn ngữ lập trình ABC.
- ❑ Guido Van Rossum đã xuất bản phiên bản đầu tiên của mã Python (phiên bản 0.9.0) tại alt.sources vào tháng 2 năm 1991. Bản phát hành này đã bao gồm xử lý ngoại lệ, các hàm và các kiểu dữ liệu cốt lõi của list, dict, str và các loại khác. Nó cũng hướng đối tượng và có một hệ thống mô-đun.
- ❑ Phiên bản Python 1.0 được phát hành vào tháng 1 năm 1994. Các tính năng mới chính bao gồm trong phiên bản này là các công cụ lập trình chức năng lambda, bản đồ, bộ lọc và giảm thiểu.

# LỊCH SỬ PYTHON

---

- ❑ Python 2.0 được phát hành vào tháng 10 năm 2000. Bản phát hành này đã giới thiệu một bộ thu gom rác và được xây dựng để hỗ trợ bảng mã Unicode.
- ❑ Python 3.0 được phát hành vào tháng 12 năm 2008. Những thay đổi trong phiên bản này lớn đến nỗi nó bao gồm một công cụ giúp chuyển đổi mã Python được viết trong phiên bản trước thành một phiên bản tương thích với Python 3.0.

# CÁC ĐỊNH LUẬT CỦA PYTHON (ZEN of PYTHON)

1. Beautiful is better than ugly.
2. Explicit is better than implicit.
3. Simple is better than complex.
4. Complex is better than complicated.
5. Flat is better than nested.
6. Sparse is better than dense.
7. Readability counts.
8. Special cases aren't special enough to break the rules.
9. Although practicality beats purity.

# CÁC ĐỊNH LUẬT CỦA PYTHON (1)

---

- 10.Errors should never pass silently.
- 11.Unless explicitly silenced.
- 12.In the face of ambiguity, refuse the temptation to guess.
- 13.There should be one -- and preferably only one -- obvious way to do it. 14.Although that way may not be obvious at first unless you're Dutch.
- 15.Now is better than never.
- 16.Although never is often better than *\*right\** now.
- 17.If the implementation is hard to explain, it's a bad idea.
- 18.If the implementation is easy to explain, it may be a good idea.
- 19.Namespaces are one honking great idea -- let's do more of those!

# TÀI LIỆU THAM KHẢO

## ❑ Học liệu bắt buộc

- ✓ Eric Matthes. Python crash course, 2nd Edition: A hands-on, projectbased introduction to programming, No Starch Press, 2019.

## ❑ Học liệu tham khảo

- ✓ Allen B. Downey, Think Python: How to Think Like a Computer Scientist, O'Reilly, 2015
- ✓ Zed A. Shaw, Learn Python 3 the Hard Way, Addison-Wesley, 2016

# TRAO ĐỔI

## ☐ Hình thức liên lạc:

✓ Email: [thuvh@ptit.edu.vn](mailto:thuvh@ptit.edu.vn)

✓ SĐT: 0387011735

## ☐ Chia nhóm

Q & A