



BỘ GIÁO DỤC VÀ ĐÀO TẠO
TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT THÀNH PHỐ HỒ CHÍ MINH
65 NĂM XÂY DỰNG VÀ PHÁT TRIỂN

TRẦN NHẬT QUANG
PHẠM VĂN KHOA

GIÁO TRÌNH **LẬP TRÌNH PYTHON CĂN BẢN**



NHÀ XUẤT BẢN
ĐẠI HỌC QUỐC GIA TP. HỒ CHÍ MINH

TS. TRẦN NHẬT QUANG, TS. PHẠM VĂN KHOA

GIÁO TRÌNH
LẬP TRÌNH
PYTHON CĂN BẢN

NHÀ XUẤT BẢN ĐẠI HỌC QUỐC GIA
THÀNH PHỐ HỒ CHÍ MINH – 2023

LỜI NÓI ĐẦU

Python hiện là ngôn ngữ lập trình phổ biến nhất thế giới¹. Ưu điểm nổi bật của Python là dễ học, dễ viết. Không những thế, Python còn có một cộng đồng người dùng lớn và hệ thống thư viện mã nguồn mở đồ sộ giúp bạn hoàn thành các dự án của mình nhanh chóng và hiệu quả. Cho dù đó là một dự án về phân tích dữ liệu, học máy, xử lý ảnh, game, điều khiển thiết bị, hoặc chỉ đơn giản là tự động hóa các tác vụ trên máy tính của bạn, thì gần như bạn đều có thể tìm thấy các thư viện Python hỗ trợ.

Quyển giáo trình này được biên soạn với mong muốn hỗ trợ hiệu quả cho các bạn sinh viên học lập trình Python. Tiêu chí thứ nhất của nhóm biên soạn là **giúp người học lập trình lần đầu** cũng có thể dễ dàng nắm bắt được các khái niệm và kỹ thuật lập trình với Python. Tiêu chí thứ hai là cố gắng trình bày nội dung một cách **súc tích, nhưng vẫn đầy đủ**, để những bạn đã học qua lập trình có thể nhanh chóng làm quen và sử dụng được Python khi học xong giáo trình.

Rất mong quyển giáo trình này sẽ hữu ích với các bạn! Nhóm biên soạn cũng mong nhận được góp ý của độc giả để cải thiện giáo trình này ngày càng tốt hơn. Xin cảm ơn các bạn!

¹ Theo xếp hạng của TIOBE Index và PYPL Index cập nhật vào tháng 3 năm 2022.

MỤC LỤC

LỜI NÓI ĐẦU	3
MỤC LỤC	5
DANH MỤC HÌNH ẢNH	8
Chương 1 GIỚI THIỆU VÀ CÀI ĐẶT PYTHON	9
1.1 Đôi nét về ngôn ngữ lập trình Python	9
1.2 Lịch sử phát triển Python	10
1.3 Cài đặt Python	11
1.4 Một số lời khuyên hữu ích cho những người mới	16
Chương 2 PHÉP TOÁN CƠ BẢN, BIẾN VÀ NHẬP XUẤT TRONG PYTHON	18
2.1 Sử dụng VS Code như một máy tính cầm tay	18
2.2 Các phép toán	19
2.3 Biến	21
2.4 Nhập xuất cơ bản	23
2.5 Lệnh if	26
Bài tập có lời giải	32
Bài tập thực hành	33
Chương 3 VÒNG LẶP VÀ CẤU TRÚC DỮ LIỆU MẢNG	35
3.1 Vòng lặp while	35
3.2 Cấu trúc dữ liệu mảng	41
3.3 Vòng lặp for	46
3.4 Break, continue và pass	49
Bài tập có lời giải	50
Bài tập thực hành	51
Chương 4 NUMPY	54
4.1 Giới thiệu về Numpy	54
4.2 Cài đặt thư viện numpy	54
4.3 Numpy arrays	55

Bài tập thực hành	58
Chương 5 SETS VÀ DICTIONARIES	60
5.1 Sets.....	60
5.2 Dictionaries	62
Bài tập có lời giải.....	67
Bài tập thực hành	68
Chương 6 STRINGS.....	69
6.1 Khái niệm và khởi tạo strings	69
6.2 Hàm xử lý strings.....	71
Bài tập thực hành	73
Chương 7 HÀM.....	74
7.1 Khái niệm và cú pháp	74
7.2 Một số ví dụ	74
7.3 Biến đoạn code bất kỳ thành hàm	81
Bài tập có lời giải.....	83
Bài tập thực hành	84
Chương 8 LỖI VÀ SỬA LỖI.....	86
8.1 Các dạng lỗi trong lập trình	86
8.2 Xử lý lỗi runtime.....	87
8.3 Xử lý lỗi logic	90
8.4 Các lưu ý khi viết code để hạn chế lỗi	95
Bài tập thực hành	97
Chương 9 VẼ VỚI MATPLOTLIB.....	99
9.1 Cách thức vẽ trên màn hình kỹ thuật số.....	99
9.2 Vẽ với thư viện matplotlib	100
9.3 Vẽ đồ thị trong tọa độ cực.....	106
9.4 Tùy chỉnh hình vẽ	108
9.5 Vẽ trên nhiều phân vùng với subplotlib.....	116
Bài tập thực hành	120

PHỤ LỤC	122
Mã nguồn bài tập có lời giải chương 2	122
Mã nguồn bài tập có lời giải chương 3	127
Mã nguồn bài tập có lời giải chương 5	131
Mã nguồn bài tập có lời giải chương 7	135
Keywords của Python	142
Cài đặt các thuật toán sắp xếp bằng Python	145
Cài đặt các thuật toán tìm kiếm trên mảng bằng Python	151
Cài đặt các thuật toán tìm kiếm trên chuỗi bằng Python	158
INDEX	161
TÀI LIỆU THAM KHẢO	163

DANH MỤC HÌNH ẢNH

Hình 1-1 Logo của Python. Nguồn: Python Software Foundation.....	9
Hình 1-2 Cài đặt Python bằng cách tick vào ô Add Python... to PATH ở cửa sổ cài đặt đầu tiên.....	12
Hình 1-3 Giao diện của Visual Studio Code.	12
Hình 1-4 Giao diện trang OnlineGDB.....	16
Hình 1-5 Ứng dụng Pydroid 3 trên Google Play.	16
Hình 2-1 Lưu đồ của khối lệnh if.	28
Hình 3-1 Lưu đồ hoạt động của vòng lặp while.	35
Hình 3-2 Lưu đồ hoạt động của vòng lặp for.	47

Chương 1

GIỚI THIỆU VÀ CÀI ĐẶT PYTHON

1.1 Đôi nét về ngôn ngữ lập trình Python

Python là một ngôn ngữ lập trình cấp cao và đa dụng (general-purpose) được phát triển bởi Guido van Rossum. Phiên bản đầu tiên của nó được phát hành vào năm 1991. Tên của nó được đặt theo chương trình hài Monty Python¹ của Anh như một cách phản ánh triết lý thiết kế của Python: một ngôn ngữ lập trình thú vị khi sử dụng.



Hình 1-1 Logo của Python.

Nguồn: Python Software Foundation

Triết lý của ngôn ngữ lập trình Python được mô tả bằng những cách ngôn trong tài liệu The Zen of Python (PEP 20) như:

- Simple is better than complex (tạm dịch: Đơn giản tốt hơn phức tạp).
- Complex is better than complicated (tạm dịch: Phức hợp tốt hơn phức tạp).
- Explicit is better than implicit (tạm dịch: Tường minh tốt hơn là ngầm định).
- Readability counts (tạm dịch: Lưu tâm đến sự dễ đọc hiểu).

Với những triết lý đó, Python hướng tới sự đơn giản, ngắn gọn trong mã lệnh của mình. Bạn sẽ cảm nhận được điều này khi bắt đầu lập trình với Python và so sánh nó với các ngôn ngữ như C/C++, Java.

Đến nay, Python đã được phát triển qua nhiều phiên bản. Hai nhóm phiên bản được sử dụng hiện nay là Python 2.x và Python 3.x. Tuy nhiên,

¹ Theo General Python FAQ, docs.python.org

các phiên bản 2.x đã không còn được hỗ trợ đầy đủ từ ngày 1/1/2020¹. Phiên bản mới nhất của Python là 3.10 (phát hành ngày 4/10/2021)².

1.2 Lịch sử phát triển Python³

Python được bắt đầu phát triển vào cuối những năm 1980 bởi Guido van Rossum tại Centrum Wiskunde & Informatica (CWI), Hà Lan như một ngôn ngữ kế thừa của ngôn ngữ lập trình ABC có khả năng xử lý ngoại lệ và giao tiếp với hệ điều hành Amoeba.

Python được bắt đầu phát triển vào tháng 12 năm 1989. Vào thời điểm đó, van Rossum là tác giả duy nhất của dự án, với tư cách là nhà phát triển chính, cho đến ngày 12 tháng 7 năm 2018. Vào tháng 1 năm 2019, các nhà phát triển cốt lõi Python đã bầu ra một Hội đồng chỉ đạo gồm năm thành viên để lãnh đạo dự án.

Python 2.0 được phát hành vào ngày 16 tháng 10 năm 2000, với nhiều tính năng mới. Python 3.0, được phát hành vào ngày 3 tháng 12 năm 2008, với nhiều tính năng chính vẫn hỗ trợ ngược Python 2.6.x và 2.7.x. Các bản phát hành của Python 3 tích hợp tiện ích 2to3 giúp dịch mã tự động từ Python 2 sang Python 3.

Python 2.7 ban đầu được chỉ định sẽ chấm dứt hoạt động vào năm 2015, nhưng sau đó bị hoãn lại đến năm 2020 vì những lo ngại về việc chuyển đổi các code Python 2 hiện có sang Python 3. Từ thời điểm đó, Python 2 không nhận được thêm bất kỳ bản vá bảo mật hoặc cải tiến nào nữa.

Sau khi Python 2 bị ngừng hỗ trợ, chỉ còn Python 3.6.x và các phiên bản mới hơn được hỗ trợ. Một thời gian sau, Python 3.6 cũng bị ngừng hỗ trợ. Đến năm 2021, Python 3.9.2 và 3.8.8 được phát triển vì tất cả các phiên bản Python trước (bao gồm 2.7) đều có vấn đề bảo mật có thể khiến máy tính bị thực thi mã từ xa và nhiễm độc bộ nhớ web cache.

Vào năm 2022, Python 3.10.4 và 3.9.12 được phát triển và các bản cũ hơn bao gồm 3.8.13 và 3.7.13 được cập nhật vì nhiều vấn đề bảo mật.

¹ Theo PEP 373 -- Python 2.7 Release Schedule

² Theo PEP 619 -- Python 3.10 Release Schedule

³ Theo [https://en.wikipedia.org/wiki/Python_\(programming_language\)](https://en.wikipedia.org/wiki/Python_(programming_language))

1.3 Cài đặt Python

Có nhiều cách khác nhau để lập trình với Python. Phần này mô tả ba cách phù hợp với các nhóm người dùng với điều kiện về thiết bị khác nhau.

Cách 1. Lập trình Python trên máy tính sử dụng VS Code

Nếu bạn sở hữu một máy tính cá nhân thì *nên dùng cách này* để tận dụng được đầy đủ chức năng của Python một cách thuận tiện.

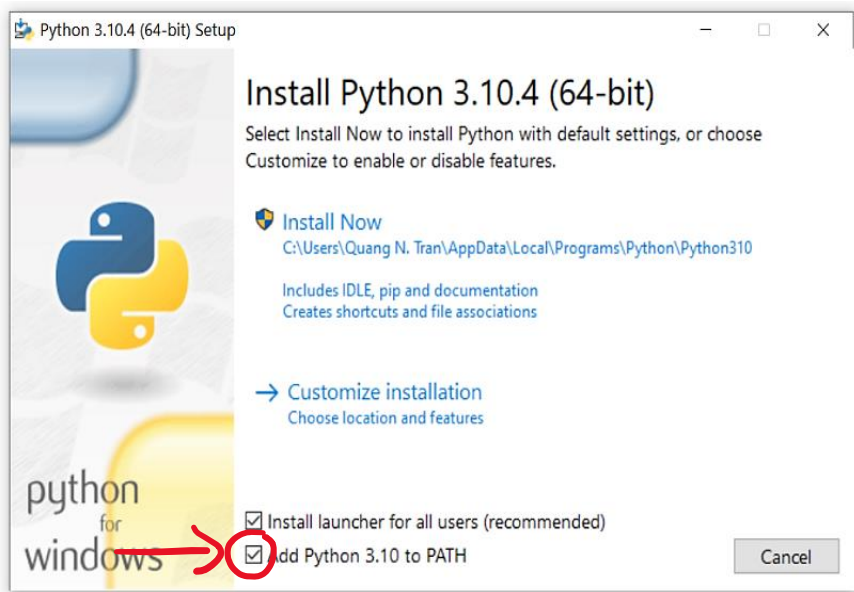
Trước tiên, bạn cần cài đặt **trình biên dịch Python (Python interpreter)**. Nên chọn phiên bản Python 3.7 hoặc mới hơn.¹ Một lưu ý khi cài đặt Python interpreter là bạn nên tick vào ô “Add Python ... to PATH” ở cửa sổ cài đặt đầu tiên (xem Hình 1-2).

Sau khi đã cài đặt xong trình biên dịch Python, bạn nên cài đặt một **editor** (trình soạn thảo) hoặc một **IDE** (Integrated Development Environment) để lập trình Python được dễ dàng hơn. Có nhiều editor, IDE khác nhau hỗ trợ lập trình Python. Trong giáo trình này, chúng tôi khuyến nghị sử dụng Visual **Studio Code**² (VS Code). Editor này có các ưu điểm như:

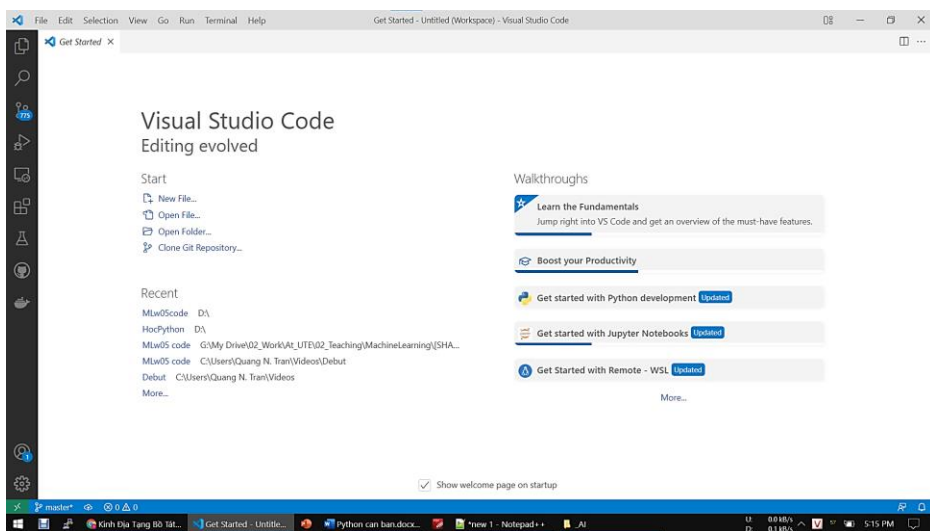
- Miễn phí, mã nguồn mở, hỗ trợ nhiều nền tảng (Windows, Linux, Mac).
- IntelliSense: giúp viết code nhanh chóng hơn bằng cách đưa ra các lựa chọn tự động hoàn thành code cho bạn.
- Hỗ trợ tìm và sửa lỗi (debug) hiệu quả.
- Nhiều extension hữu ích: như kết nối với Git, đọc file Jupyter Notebook (.ipynb), hỗ trợ Docker.

¹ Download Python phiên bản mới nhất tại <https://www.python.org/downloads/>. Với những phiên bản cũ, download tại <https://www.python.org/ftp/python/>.

² Download bản cài đặt VS Code tại <https://code.visualstudio.com/download>.



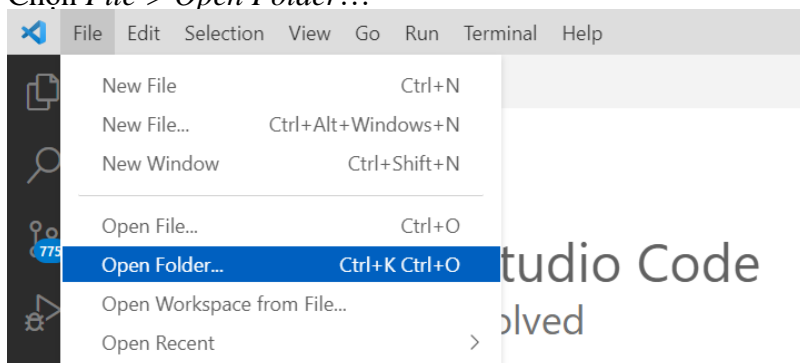
Hình 1-2 Lưu ý: Khi cài đặt Python nên tick vào ô Add Python ... to PATH để thuận tiện chạy các Scripts Python về sau..



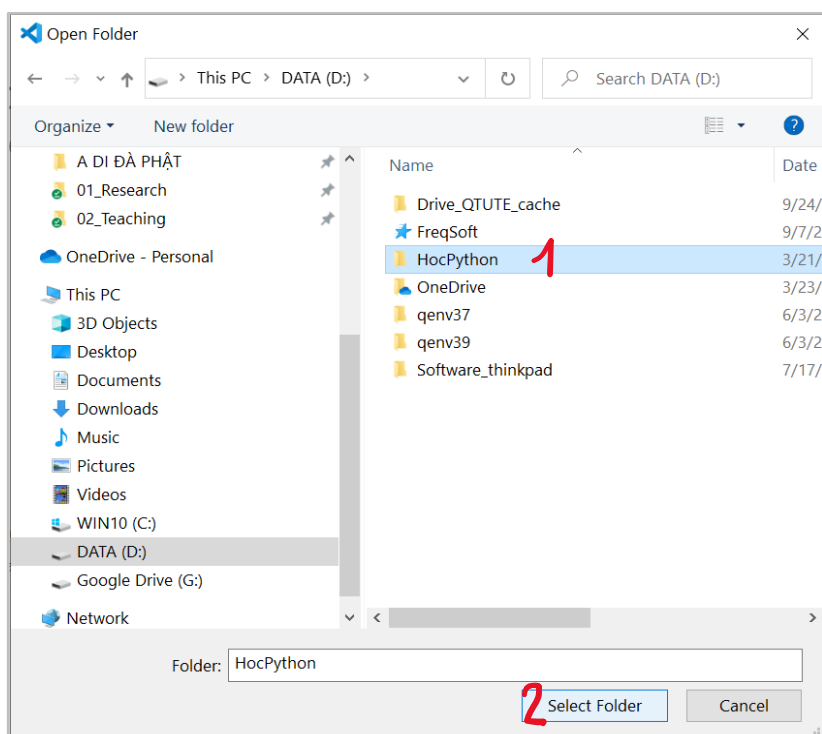
Hình 1-3 Giao diện của Visual Studio Code.


Sau khi cài đặt VS Code xong, bạn nên thực hiện các bước sau để tạo một file code Python đầu tiên và cũng để VS Code hoàn tất cài đặt các extension hỗ trợ lập trình Python.

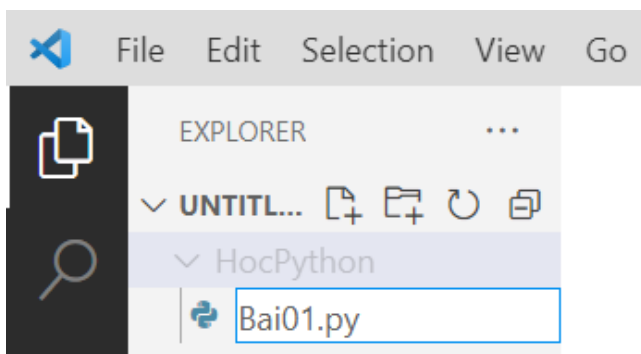
1. Chọn *File > Open Folder...*



2. Tạo mới hoặc chọn một thư mục rồi nhấn *Select Folder*.
Lưu ý: đối với máy tính sử dụng Windows, bạn nên chọn một thư mục trong ổ D: hoặc một ổ đĩa mà bạn có đầy đủ quyền chạy các mã lệnh. Không dùng các thư mục có tên “python” hoặc “code” vì có thể gây lỗi về sau. Tên và đường dẫn thư mục tốt nhất là không chứa khoảng trắng và không chứa dấu tiếng Việt.



3. Nhấn vào biểu tượng  và nhập tên file.
Lưu ý: Tên file phải có phần mở rộng `.py`. Không nên dùng tên file `python.py` hoặc `code.py` để tránh bị lỗi khi chạy. Tên file tốt nhất là không chứa khoảng trắng và không chứa dấu tiếng Việt.



4. Sau khi tạo file xong, VS Code có thể hỏi bạn có muốn cài đặt extension cho Python không. Hãy chọn *Install*.
5. Nhập nội dung sau đây vào file vừa tạo:

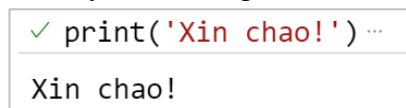
```
#%%
```

```
print("Xin chào!")
```

6. Nhấn **Shift-Enter** để chạy.

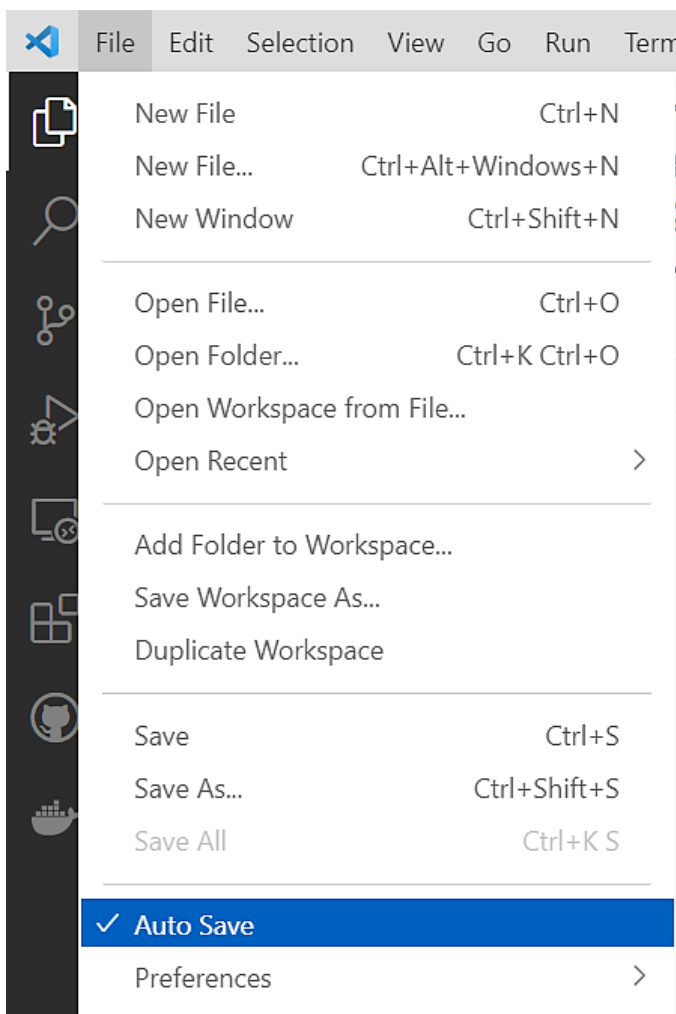
Lưu ý: Lúc này VS Code có thể hỏi bạn muốn cài đặt các extension hỗ trợ không, hãy chọn *Install*.

7. Nếu kết quả hiện ra như hình sau tức là bạn đã hoàn thành cài đặt và chạy thành công đoạn lệnh Python đầu tiên.



8. Chọn *File > Auto Save* để VS Code tự động lưu code.

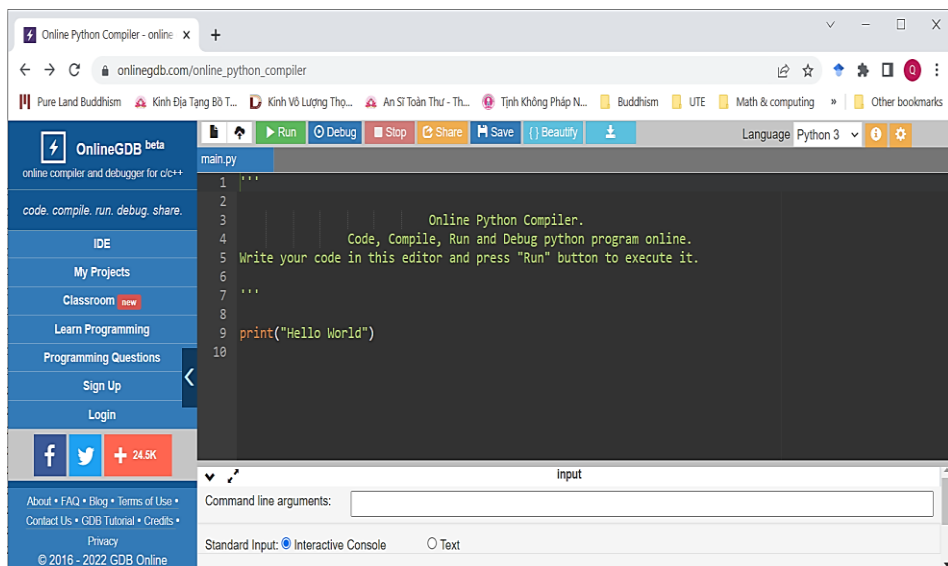
Lưu ý: Nếu không dùng Auto Save thì phải lưu (Save) file code thủ công bằng cách nhấn **Ctrl-S** mỗi khi muốn lưu.



Cách 2. Lập trình Python trên máy tính sử dụng trình duyệt web

Nếu bạn sử dụng máy tính công cộng hoặc một máy tính không thể cài đặt Python thì bạn có thể sử dụng trình duyệt web có kết nối internet và truy cập vào địa chỉ sau để lập trình Python: https://www.onlinegdb.com/online_python_compiler.

Lưu ý: Cách này không đảm bảo hỗ trợ đầy đủ chức năng của Python. Chỉ nên dùng khi không thể cài đặt Python và VS Code như cách 1.



Hình 1-4. Giao diện trang OnlineGDB.

Cách 3. Lập trình Python sử dụng smartphone

Trong trường hợp không có máy tính, bạn vẫn có thể lập trình Python bằng cách sử dụng smartphone và cài đặt ứng dụng **Pydroid 3** (hoặc một ứng dụng tương tự).

Lưu ý: cách này không được khuyến khích sử dụng vì không hỗ trợ đủ tính năng của Python. Chỉ nên sử dụng tạm thời cách này trong một thời gian ngắn khi không thể dùng máy tính.



Pydroid 3 - IDE for Python 3

IIEC Education

★ ★ ★

Everyone

Contains Ads · Offers in-app purchases

✗ This app is not available for any of your devices

Add to Wishlist

Hình 1-5 Ứng dụng Pydroid 3 trên Google Play.

1.4 Một số lời khuyên hữu ích cho những người mới

Lập trình là một kỹ năng. Vì vậy nếu muốn lập trình tốt không có cách nào khác ngoài **thực hành**. Tự mình thực hành càng nhiều càng tốt!

Một số kinh nghiệm khi luyện lập trình:

- Cảm thấy không hiểu rõ khi **lần đầu** học về một khái niệm, kỹ năng lập trình. Điều này là hoàn toàn bình thường! Bạn chỉ cần bỏ thêm chút thời gian xem lại một vài lần, rồi tự mình ngồi code lại nội dung được học thì sẽ dần dần hiểu rõ.

Thậm chí đôi khi bạn đã rất cố gắng mà vẫn cảm thấy không hiểu rõ hết. Không sao cả! Có một hiện tượng là: dường như não của chúng ta tự động tổng hợp kiến thức mà nó từng biết qua. Sau một học kỳ, hoặc vài tháng, thậm chí một năm sau, bạn sẽ ngạc nhiên khi **gặp lại** kiến thức lúc trước bạn thấy bế tắc: lúc này bạn hiểu nó rất rõ ràng! Vì vậy, đừng ngại khi học qua một lần mà chưa hiểu rõ.

Keep learning!

- Những lần đầu lập trình đôi khi giống **học thuộc lòng**, sao chép code. Tức là bạn xem code mẫu rồi gõ lại giống như vậy. Điều này cũng hoàn toàn bình thường! Học thuộc luôn luôn là một phần của việc thu nạp kiến thức.
- Đừng ngại ngồi **gõ lại code mẫu**. Tự mình gõ lại code khác xa với việc ngồi nhìn code và nghĩ rằng mình hiểu. Gõ lại code giúp bạn trải nghiệm lập trình. Khi bạn gõ và chạy code, bạn sẽ tự nhiên ghi nhớ, tư duy và phát hiện các lỗi, các vấn đề trong đoạn lệnh. Bạn sẽ hiểu nó cặn kẽ và từ từ sẽ tự viết được các đoạn code theo ý mình.
- **Internet** có thể rất hữu ích. Nếu gặp những lỗi khó hiểu khi lập trình, hoặc nghĩ mãi chưa ra cách lập trình cho một vấn đề, bạn có thể thử google. Nhiều khả năng bạn sẽ tìm được lời giải từ các diễn đàn, các bài viết về lập trình.

*Tuy nhiên, **đừng lạm dụng**! Lúc nào cũng tra google trước khi tự mình tìm cách giải quyết, hoặc chỉ copy code mà không hiểu, chắc chắn sẽ có hại cho bạn!*

Ngoài ra, bạn cũng nên cẩn thận với vấn đề **đạo văn (plagiarism)**. Nên tìm hiểu về quy định bản quyền của đoạn code mà bạn định sử dụng và nhớ phải ghi nguồn.

- **Thực hành càng nhiều càng tốt!**

Chương 2

PHÉP TOÁN CƠ BẢN, BIẾN VÀ NHẬP XUẤT TRONG PYTHON

Chương này giúp bạn làm quen với các phép toán cơ bản, khái niệm biến (variables) trong lập trình và các hàm cơ bản để nhập xuất dữ liệu trong Python.

2.1 Sử dụng VS Code như một máy tính cầm tay

Phần này demo một số cách thực thi lệnh Python với VS Code. Bạn có thể tận dụng những cách này để biến VS Code với Python thành một công cụ tính toán như máy tính cầm tay (calculator).

Lưu ý: Trước khi có thể chạy code Python trong VS Code, bạn cần Cài đặt Python (xem mục 1.3 Cài đặt Python).

Sau đây giới thiệu ba cách thực thi lệnh Python trong VS Code.

Cách 1. Tạo và thực thi cell

Để tạo một cell, bạn dùng cú pháp `%% Tên gọi nhớ cho cell` như ví dụ sau:

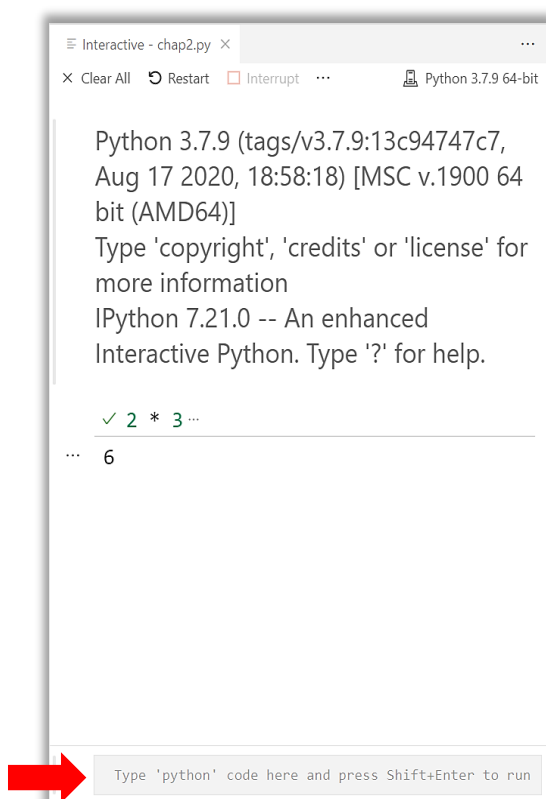
```
Run Cell | Run Below | Debug Cell  
1 %% Example 1  
2 2 * 3
```

Để chạy cell, bạn đặt con nháy vào 1 dòng bất kỳ trong cell (nằm giữa 2 đường kẻ màu xanh như trong hình trên), và nhấn *Run Cell* (nằm ở ngay phía trên dòng `%%`) hoặc sử dụng tổ hợp phím tắt **Shift-Enter**.

Khi bạn chạy cell lần đầu tiên, VS Code sẽ load Python interpreter rồi mới thực thi code nên mất thời gian một chút. Từ lần chạy thứ 2, code sẽ được thực thi ngay vì Python interpreter đã được load vào bộ nhớ rồi.

Cách 2. Thực thi code trong cửa sổ interactive

Để sử dụng cách này trước tiên bạn phải chạy ít nhất một cell theo cách 1 để VS Code load Python interpreter và khởi tạo cửa sổ interactive (như hình dưới).

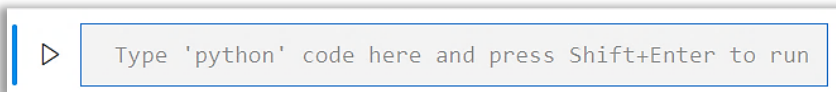


```
Python 3.7.9 (tags/v3.7.9:13c94747c7,
Aug 17 2020, 18:58:18) [MSC v.1900 64
bit (AMD64)]
Type 'copyright', 'credits' or 'license' for
more information
IPython 7.21.0 -- An enhanced
Interactive Python. Type '?' for help.

✓ 2 * 3 ...
... 6

Type 'python' code here and press Shift+Enter to run
```

Trong cửa sổ interactive, bạn có thể nhập code vào dòng lệnh (nằm dưới cùng cửa sổ interactive, tại vị trí có ghi dòng chữ *Type 'python' code here and press Shift+Enter to run* (như hình dưới). Khi nhập xong, nhấn **Shift+Enter** để thực thi.



2.2 Các phép toán

Python hỗ trợ các phép toán cơ bản như mô tả trong bảng sau.

Phép toán	Cú pháp	Code mẫu	Kết quả
Trị tuyệt đối	<code>abs()</code>	<code>abs(-2)</code>	2
Cộng	<code>+</code>	<code>2 + 5</code>	7
Trừ	<code>-</code>	<code>3 - 1</code>	2
Nhân	<code>*</code>	<code>3 * 2</code>	6