**HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG**

**LẬP TRÌNH VỚI PYTHON  
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
  
  
**

**BÀI TẬP LỚN**

|  |  |
| --- | --- |
| **Giảng viên hướng dẫn** | **: Phạm Văn Sự** |
| **Sinh viên** | **: Võ Hữu Tuấn** |
| **Mã sinh viên** | **: B20DCCN622** |

**Hà Nội – 2022**

Lời mở đầu

Python là một [ngôn ngữ lập trình](https://vi.wikipedia.org/wiki/Ng%C3%B4n_ng%E1%BB%AF_l%E1%BA%ADp_tr%C3%ACnh) bậc cao cho các mục đích lập trình đa năng, do [Guido van Rossum](https://vi.wikipedia.org/w/index.php?title=Guido_van_Rossum&action=edit&redlink=1) tạo ra và lần đầu ra mắt vào năm [1991](https://vi.wikipedia.org/wiki/1990). Python được thiết kế với ưu điểm mạnh là dễ đọc, dễ học và dễ nhớ. Python là ngôn ngữ có hình thức rất sáng sủa, cấu trúc rõ ràng, thuận tiện cho người mới học lập trình và là ngôn ngữ lập trình dễ học; được dùng rộng rãi trong phát triển [trí tuệ nhân tạo](https://vi.wikipedia.org/wiki/Tr%C3%AD_tu%E1%BB%87_nh%C3%A2n_t%E1%BA%A1o). Cấu trúc của Python còn cho phép người sử dụng viết mã lệnh với số lần gõ phím tối thiểu.

Để học tập và làm việc với Python, chúng ta có thể sử dụng 1 IDE, ví dụ như: Visual Studio Code, Pycharm, …

Bài báo cáo này sẽ trình bày 1 số ý về Python, 1 số trình IDE nên sử dụng, và cách thiết lập môi trường để hỗ trợ lập trình Python

Bài báo cáo được thực hiện bới Võ Hữu Tuấn – B20DCCN622

Nội dung

[*Lời mở đầu* 2](#_Toc114843583)

[**I.** **Tóm lược về Python** 4](#_Toc114843584)

[**1** **Biến trong Python** 4](#_Toc114843585)

[Quy tắc đặt tên biến: 4](#_Toc114843586)

[Tham chiếu đối tượng: 4](#_Toc114843587)

[Nhận dạng đối tượng: 4](#_Toc114843588)

[**2** **Keyword trong Python** 5](#_Toc114843589)

[**3** **Comment trong Python** 5](#_Toc114843590)

[**4** **Các kiểu dữ liệu số và phép toán trong Python** 6](#_Toc114843591)

[**5** **Boolean trong python** 6](#_Toc114843592)

[**6** **String trong Python** 6](#_Toc114843593)

[a, Giới thiệu 6](#_Toc114843594)

[b, Một số phướng thức trong Python 7](#_Toc114843595)

[c, String Slices 8](#_Toc114843596)

[d, String % 9](#_Toc114843597)

[**7** **Tuple trong Python** 9](#_Toc114843598)

[**8** **List trong Python** 10](#_Toc114843599)

[**9** **Set trong Python** 10](#_Toc114843600)

[**10** **Dictionary trong Python** 11](#_Toc114843601)

[**11** **Vòng lặp trong Python** 11](#_Toc114843602)

[a, vòng lặp while 11](#_Toc114843603)

[b, vòng lặp for in 11](#_Toc114843604)

[**12** **Funtion** 12](#_Toc114843605)

[a, hàm không đầu vào 12](#_Toc114843606)

[b, hàm có đầu vào 12](#_Toc114843607)

[c, hàm lamda 12](#_Toc114843608)

[**II.** **Đánh giá so sánh 1 số loại IDE hỗ trợ lập trình Python** 13](#_Toc114843609)

[*Giới thiệu* 13](#_Toc114843610)

[1. PyCharm 13](#_Toc114843611)

[2. Spyder 14](#_Toc114843612)

[3. Microsoft Visual Studio 14](#_Toc114843613)

[**III.** **IDE đang sử dụng(Pycharm)** 14](#_Toc114843614)

[**1. Hướng dẫn tải** 14](#_Toc114843615)

[**2. Hướng dẫn sử dụng** 15](#_Toc114843616)

[**IV.** **Tổng kết** 20](#_Toc114843617)

[Liên kết kham khảo 21](#_Toc114843618)

1. **Tóm lược về Python**
2. **Biến trong Python**

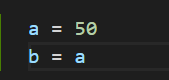
Quy tắc đặt tên biến:

* Ký tự đầu tiên của biến phải là một bảng chữ cái hoặc dấu gạch dưới (\_).
* Tất cả các ký tự ngoại trừ ký tự đầu tiên có thể là bảng chữ cái viết thường (a-z), viết hoa (A-Z), gạch dưới hoặc chữ số (0-9).
* Tên số nhận dạng không được chứa bất kỳ khoảng trắng hoặc ký tự đặc biệt nào (!, @, #,%, ^, &, \*). Tên định danh không được giống với bất kỳ từ khóa nào được xác định trong ngôn ngữ.
* Tên định danh có phân biệt chữ hoa chữ thường; ví dụ: tên của tôi và MyName không giống nhau.
* Ví dụ về số nhận dạng hợp lệ: a123, \_n, n\_9, v.v.
* Ví dụ về số nhận dạng không hợp lệ: 1a, n%4, n 9, v.v.

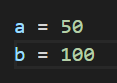
Tham chiếu đối tượng:



* Trong hình trên, biến a tham chiếu đến một đối tượng số nguyên.
* Giả sử chúng ta gán giá trị số nguyên 50 cho một biến mới b.



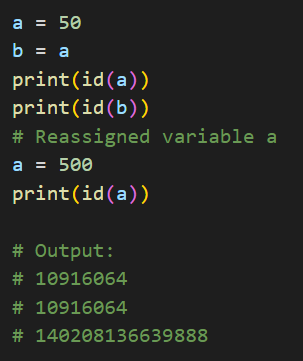
* Biến b tham chiếu đến cùng một đối tượng mà a trỏ đến vì Python không tạo một đối tượng khác.



* Gán giá trị mới cho b. Bây giờ cả hai biến sẽ tham chiếu đến các đối tượng khác nhau.

Nhận dạng đối tượng:

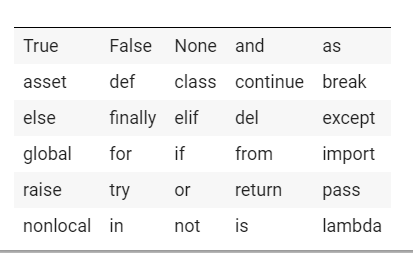
* Trong Python, mọi đối tượng được tạo đều nhận dạng duy nhất trong Python. Python cung cấp sự đảm bảo rằng không có hai đối tượng nào có cùng một mã định danh. Hàm id () tích hợp sẵn, được sử dụng để xác định mã định danh đối tượng. Hãy xem xét ví dụ sau.



* Chúng ta đã gán b = a, a và b đều trỏ đến cùng một đối tượng. Khi chúng tôi kiểm tra bằng hàm id (), nó trả về cùng một số. Chúng tôi chỉ định lại a thành 500; sau đó nó tham chiếu đến định danh đối tượng mới.

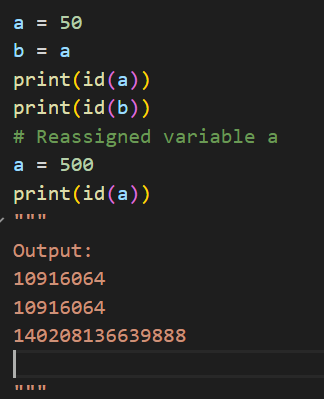
1. **Keyword trong Python**

* Từ khóa Python là những từ dành riêng đặc biệt truyền tải một ý nghĩa đặc biệt cho trình biên dịch/trình thông dịch. Mỗi từ khóa có một ý nghĩa đặc biệt và một hoạt động cụ thể. Các từ khóa này không thể được sử dụng như một biến. Sau đây là danh sách các từ khóa Python.



1. **Comment trong Python**

* Để áp dụng nhận xét trong mã, chúng tôi sử dụng dấu thăng (#) ở đầu câu lệnh hoặc mã.
* Chúng tôi cũng có thể sử dụng dấu ngoặc kép ('' '' '') cho nhận xét nhiều dòng. Dấu ngoặc kép cũng được sử dụng để định dạng chuỗi. Hãy xem xét ví dụ sau.



1. **Các kiểu dữ liệu số và phép toán trong Python**

Integet: số nguyên

Float: số thực

Phép cộng: +

Phép trừ: -

Phép nhân: \*

Phép chia: /

Phép chia nguyên: //

Phép chia dư: %

Phép lũy thừa \*\*

Phép gán

1. **Boolean trong python**

* True - Nó đại diện cho Boolean true, nếu điều kiện đã cho là true, thì nó trả về "True". Các giá trị khác 0 được coi là đúng.
* Sai - Nó đại diện cho sai Boolean; nếu điều kiện đã cho là sai, thì nó trả về "Sai". Giá trị 0 được coi là sai
* Không có - Nó biểu thị giá trị null hoặc void. Danh sách trống hoặc Số không không thể được coi là Không có.

1. **String trong Python**

a, Giới thiệu

* Python có một lớp chuỗi tích hợp tên là "str" ​​với nhiều tính năng tiện dụng (có một mô-đun cũ hơn có tên "string" mà bạn không nên sử dụng). Các ký tự chuỗi có thể được bao bởi dấu ngoặc kép hoặc dấu nháy đơn, mặc dù dấu ngoặc kép thường được sử dụng nhiều hơn. Các thoát dấu gạch chéo ngược hoạt động theo cách thông thường trong cả ký tự được trích dẫn đơn và kép - ví dụ: \ n \ '\ ". Một chuỗi ký tự được trích dẫn kép có thể chứa các dấu ngoặc kép đơn mà không gây phiền phức (ví dụ:" Tôi đã không làm điều đó ") và tương tự như vậy được trích dẫn đơn chuỗi có thể chứa dấu ngoặc kép. Một ký tự chuỗi có thể kéo dài nhiều dòng, nhưng phải có dấu gạch chéo ngược \ ở cuối mỗi dòng để thoát khỏi dòng mới. Chuỗi ký tự bên trong dấu ngoặc kép, "" "hoặc '' ', có thể kéo dài nhiều dòng của văn bản.
* Các chuỗi Python là "bất biến" có nghĩa là chúng không thể thay đổi sau khi được tạo (các chuỗi Java cũng sử dụng kiểu bất biến này). Vì không thể thay đổi chuỗi, chúng tôi tạo chuỗi \* new \* khi chúng tôi biểu diễn các giá trị đã tính. Vì vậy, ví dụ biểu thức ('hello' + 'there') nhận vào 2 chuỗi 'hello' và 'there' và xây dựng một chuỗi mới 'hellothere'.
* Các ký tự trong một chuỗi có thể được truy cập bằng cú pháp [] tiêu chuẩn và giống như Java và C ++, Python sử dụng lập chỉ mục dựa trên 0, vì vậy nếu s là 'hello' thì [1] là 'e'. Nếu chỉ mục nằm ngoài giới hạn cho chuỗi, Python sẽ gây ra lỗi. Phong cách Python (không giống như Perl) là tạm dừng nếu nó không thể biết phải làm gì, thay vì chỉ tạo một giá trị mặc định. Cú pháp "lát cắt" tiện dụng (bên dưới) cũng hoạt động để trích xuất bất kỳ chuỗi con nào từ một chuỗi. Hàm len (string) trả về độ dài của một chuỗi. Cú pháp [] và hàm len () thực sự hoạt động trên bất kỳ kiểu trình tự nào - chuỗi, danh sách, v.v. Python cố gắng làm cho các hoạt động của nó hoạt động nhất quán trên các kiểu khác nhau. Thành viên mới Python gotcha: không sử dụng "len" làm tên biến để tránh chặn hàm len (). Toán tử '+' có thể nối hai chuỗi.

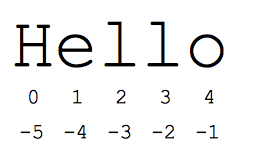
b, Một số phướng thức trong Python

Dưới đây là một số phương thức chuỗi phổ biến nhất. Một phương thức giống như một hàm, nhưng nó chạy "trên" một đối tượng. Nếu biến s là một chuỗi, thì mã s.lower () chạy phương thức low () trên đối tượng chuỗi đó và trả về kết quả (ý tưởng này về phương thức chạy trên một đối tượng là một trong những ý tưởng cơ bản tạo nên đối tượng. Lập trình định hướng, OOP). Dưới đây là một số phương thức chuỗi phổ biến nhất:

* s.lower (), s.upper () - trả về phiên bản viết thường hoặc viết hoa của chuỗi
* s.strip () - trả về một chuỗi có khoảng trắng bị xóa khỏi đầu và cuối
* s.isalpha () / s.isdigit () / s.isspace () ... - kiểm tra nếu tất cả các ký tự chuỗi nằm trong các lớp ký tự khác nhau
* s.startswith ('other'), s.endswith ('other') - kiểm tra nếu chuỗi bắt đầu hoặc kết thúc bằng chuỗi khác đã cho
* s.find ('other') - tìm kiếm chuỗi khác đã cho (không phải biểu thức chính quy) trong s và trả về chỉ mục đầu tiên nơi nó bắt đầu hoặc -1 nếu không tìm thấy
* s.replace ('old', 'new') - trả về một chuỗi trong đó tất cả các lần xuất hiện của 'old' đã được thay thế bằng 'new'
* s.split ('delim') - trả về danh sách các chuỗi con được phân tách bằng dấu phân cách đã cho. Dấu phân cách không phải là một biểu thức chính quy, nó chỉ là văn bản. 'aaa, bbb, ccc'.split (', ') -> [' aaa ',' bbb ',' ccc ']. Là một trường hợp đặc biệt thuận tiện, s.split () (không có đối số) phân tách trên tất cả các ký tự khoảng trắng.
* s.join (list) - đối lập với split (), nối các phần tử trong danh sách đã cho với nhau bằng cách sử dụng chuỗi làm dấu phân cách. ví dụ: '---'. tham gia (['aaa', 'bbb', 'ccc']) -> aaa --- bbb --- ccc

c, String Slices

Cú pháp "lát cắt" là một cách tiện dụng để tham chiếu đến các phần con của chuỗi - thường là chuỗi và danh sách. Slice s [start: end] là các phần tử bắt đầu từ đầu và kéo dài đến nhưng không bao gồm kết thúc. Giả sử chúng ta có s = "Xin chào"



* s [1: 4] là 'ell' - các ký tự bắt đầu từ chỉ mục 1 và kéo dài đến nhưng không bao gồm chỉ mục 4
* s [1:] là 'ello' - bỏ qua một trong hai chỉ mục mặc định ở đầu hoặc cuối chuỗi
* s [:] là 'Xin chào' - việc bỏ qua cả hai luôn cung cấp cho chúng ta một bản sao của toàn bộ (đây là cách hay ho để sao chép một chuỗi như một chuỗi hoặc danh sách)
* s [1: 100] là 'ello' - một chỉ mục quá lớn sẽ bị cắt bớt xuống độ dài chuỗi

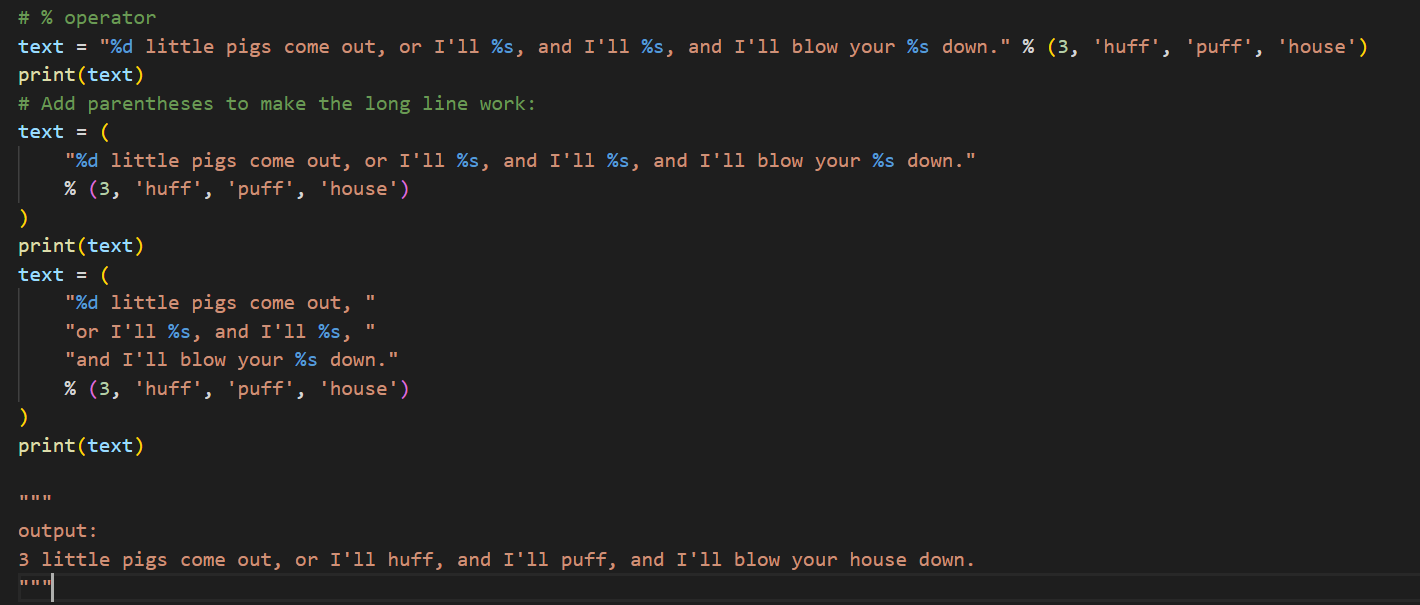
Các số chỉ mục dựa trên số 0 tiêu chuẩn cho phép dễ dàng truy cập vào các ký tự gần đầu chuỗi. Thay vào đó, Python sử dụng số âm để dễ dàng truy cập vào các ký tự ở cuối chuỗi: s [-1] là ký tự cuối cùng 'o', s [-2] là 'l' là ký tự tiếp theo char, và như vậy. Số chỉ mục âm được đếm ngược từ cuối chuỗi:

* s [-1] là 'o' - ký tự cuối cùng (đầu tiên từ cuối)
* s [-4] là 'e' - thứ 4 từ cuối
* s [: - 3] là 'Anh ấy' - đi lên nhưng không bao gồm 3 ký tự cuối cùng.
* s [-3:] là 'llo' - bắt đầu bằng ký tự thứ 3 từ cuối và kéo dài đến cuối chuỗi.

Đó là một sự thật rõ ràng về các lát cắt mà đối với bất kỳ chỉ số n nào s[:n] + s[n:] == s,. Điều này hoạt động ngay cả đối với n âm hoặc ngoài giới hạn. Hay nói một cách khác s [: n] và s [n:] luôn phân chia chuỗi thành hai phần chuỗi, bảo toàn tất cả các ký tự. Như chúng ta sẽ thấy trong phần danh sách sau này, các lát cắt cũng hoạt động với danh sách.

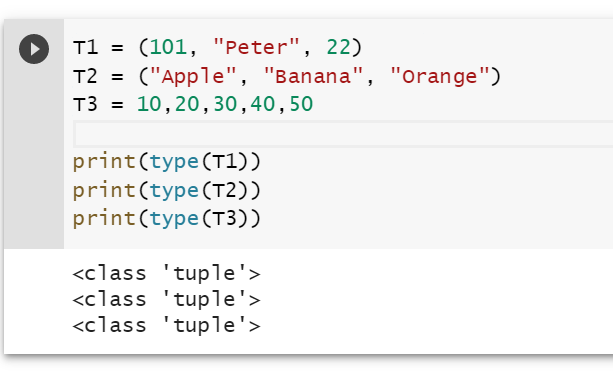
d, String %

Python có một cơ sở giống như printf () để ghép một chuỗi lại với nhau. Toán tử% lấy một chuỗi định dạng kiểu printf ở bên trái (% d int,% s string,% f /% g dấu phẩy động) và các giá trị phù hợp trong một bộ ở bên phải (một bộ được tạo bởi các giá trị được phân tách bằng dấu phẩy, thường được nhóm bên trong dấu ngoặc đơn):



1. **Tuple trong Python**

Python Tuple được sử dụng để lưu trữ chuỗi các đối tượng Python bất biến. Tuple tương tự như danh sách vì giá trị của các mục được lưu trữ trong danh sách có thể thay đổi, trong khi tuple là bất biến và không thể thay đổi giá trị của các mục được lưu trữ trong tuple.



1. **List trong Python**

Một danh sách trong Python được sử dụng để lưu trữ chuỗi các loại dữ liệu khác nhau. Danh sách Python là loại có thể thay đổi nghĩa là chúng ta có thể sửa đổi phần tử của nó sau khi nó được tạo. Tuy nhiên, Python bao gồm sáu kiểu dữ liệu có khả năng lưu trữ các chuỗi, nhưng kiểu phổ biến và đáng tin cậy nhất là danh sách.

Một danh sách có thể được định nghĩa là một tập hợp các giá trị hoặc các mục thuộc các loại khác nhau. Các mục trong danh sách được phân tách bằng dấu phẩy (,) và kèm theo dấu ngoặc vuông [].



1. **Set trong Python**

Set Python là Set các mục không có thứ tự. Mỗi phần tử trong Set phải là duy nhất, không thay đổi và các Set loại bỏ các phần tử trùng lặp. Các Set có thể thay đổi, có nghĩa là chúng ta có thể sửa đổi nó sau khi tạo.

Không giống như các Colections khác trong Python, không có chỉ mục nào được gắn vào các phần tử của set, tức là chúng ta không thể truy cập trực tiếp vào bất kỳ phần tử nào của tập hợp bởi chỉ mục. Tuy nhiên, chúng ta có thể in tất cả chúng lại với nhau, hoặc chúng ta có thể lấy danh sách các phần tử bằng cách lặp qua set.

1. **Dictionary trong Python**

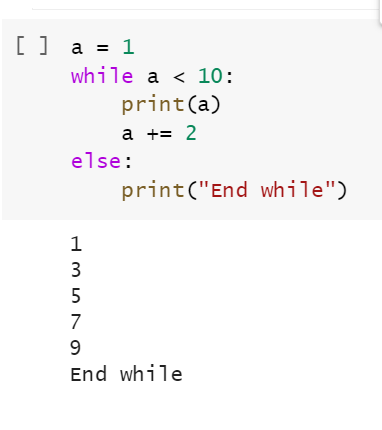
Dictionary Python được sử dụng để lưu trữ dữ liệu ở định dạng cặp khóa-giá trị. Dictionary là kiểu dữ liệu trong Python, có thể mô phỏng cách sắp xếp dữ liệu trong cuộc sống thực, trong đó một số giá trị cụ thể tồn tại cho một số khóa cụ thể. Nó là cấu trúc dữ liệu có thể thay đổi. Dictionary được định nghĩa thành các Khóa và giá trị phần tử.

* Các phím phải là một phần tử duy nhất
* Giá trị có thể là bất kỳ loại nào chẳng hạn như danh sách, tuple, số nguyên, v.v.

Nói cách khác, chúng ta có thể nói rằng Dictionary là tập hợp các cặp khóa-giá trị trong đó giá trị có thể là bất kỳ đối tượng Python nào. Ngược lại, các khóa là đối tượng Python bất biến, tức là Numbers, string hoặc tuple.

1. **Vòng lặp trong Python**

a, vòng lặp while

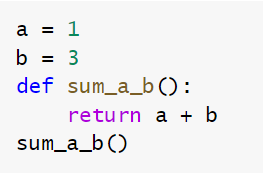


b, vòng lặp for in



1. **Funtion**

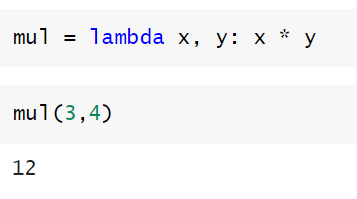
a, hàm không đầu vào



b, hàm có đầu vào



c, hàm lamda



1. **Đánh giá so sánh 1 số loại IDE hỗ trợ lập trình Python**

*Giới thiệu*

**IDE** là viết tắt của Integrated Development Environment (môi trường phát triển tích hợp) được định nghĩa là một công cụ mã hóa giúp tự động hóa quá trình chỉnh sửa, biên dịch, kiểm thử mã nguồn và nó giúp nhà phát triển dễ dàng chạy, viết và debug code.

Nó được thiết kế đặc biệt để phát triển phần mềm bao gồm một số công cụ được sử dụng để phát triển và kiểm thử phần mềm.

*Giới thiệu IDE phổ biến trong lập trình Python như sau:*

* PyCharm
* Spyder
* PyDev
* Atom
* Wing
* Jupyter Notebook
* Thonny
* Rodeo
* Microsoft Visual Studio
* Eric

1. PyCharm

* PyCharm được phát triển bởi Jet Brains và đây là môi trường phát triển tích hợp đa nền tảng (IDE) được thiết kế đặc biệt cho Python. Đây là IDE được sử dụng rộng rãi nhất và có sẵn ở cả phiên bản trả phí và nguồn mở miễn phí.
* PyCharm là một IDE Python hoàn hảo với một các tính năng phong phú như tự đồng hoàn thiện code, điều hướng project nhanh, test và debug nhanh, hỗ trợ phát triển từ xa, khả năng truy cập cơ sở dữ liệu, v.v.

**Tính năng, đặc điểm:**

* Điều hướng mã thông minh
* Đánh dấu lỗi
* Trình gỡ lỗi (debug) mạnh mẽ
* Hỗ trợ các framework phát triển web Python, ví dụ, Angular JS, Javascript

1. Spyder

Spyder là một công cụ mã nguồn mở có sự công nhận cao trong thị trường IDE và phù hợp nhất với khoa học dữ liệu. Tên đầy đủ của Spyder là môi trường phát triển Python khoa học. Nó hỗ trợ tất cả các nền tảng quan trọng Linux, Windows và MacOS X.

Nó cung cấp một tập hợp các tính năng như trình soạn thảo mã cục bộ, trình xem tài liệu, trình thám hiểm biến, bảng điều khiển tích hợp, v.v. và hỗ trợ các mô-đun khoa học như NumPy, SciPy, v.v.

**Tính năng, đặc điểm:**

* Làm nổi bật cú pháp đúng và hoàn thành mã tự động
* Tích hợp mạnh mẽ với Python console
* Hoạt động tốt trong chế độ chỉnh sửa đa ngôn ngữ và chế độ hoàn thành mã tự động

1. Microsoft Visual Studio

Microsoft Visual Studio là một trình soạn thảo mã nguồn mở phù hợp nhất để phát triển và gỡ lỗi các dự án web và đám mây mới nhất. Nó có thị trường riêng cho các phần mở rộng.

**Tính năng, đặc điểm:**

* Hỗ trợ mã hóa Python trong Visual studio
* Có sẵn ở cả phiên bản trả phí và miễn phí

***Ngoài ra còn rất nhiều IDE khác***

1. **IDE đang sử dụng(Pycharm)**

**1. Hướng dẫn tải**

**Bước 1:** Đầu tiên, bạn vào trang chủ của [**Pycharm**](https://www.jetbrains.com/pycharm/)

Sau đó **nhấn chọn Tải xuống**

Lúc này, giao diện sẽ hiển thị ra 2 danh mục mới cho bạn lựa chọn tải về

* **Chuyên nghiệp**: Để phát triển cả Khoa học và Web Python. Với hỗ trợ HTML, JS và SQL. Bạn sẽ được dùng thử miễn phí, nếu muốn dùng luôn bạn sẽ phải trả phí bản quyền
* **Cộng đồng**: Để phát triển Python thuần túy. Được sử dụng miễn phí.

**Bước 2:** Quá trình tải xuống mất khoảng 3 phút

Hoàn tất tải xuống, bạn nhấn vào file vừa được tải về, sau đó **nhấn vào mục Run**

Đợi phần mềm load đủ 100%

**Bước 3:** Sau đó, bạn mở file lên, nhấn **chọn Next**

**Bước 4:** **Tiếp tục nhấn chọn Next**

**Bước 5:** Tại thanh công cụ, tùy chỉnh sang mục **JetBrains**, sau đó **nhấn chọn Install**

**Bước 6:** Đợi quá trình Installing diễn ra

**Bước 7:** Nhấn **chọn Finish**

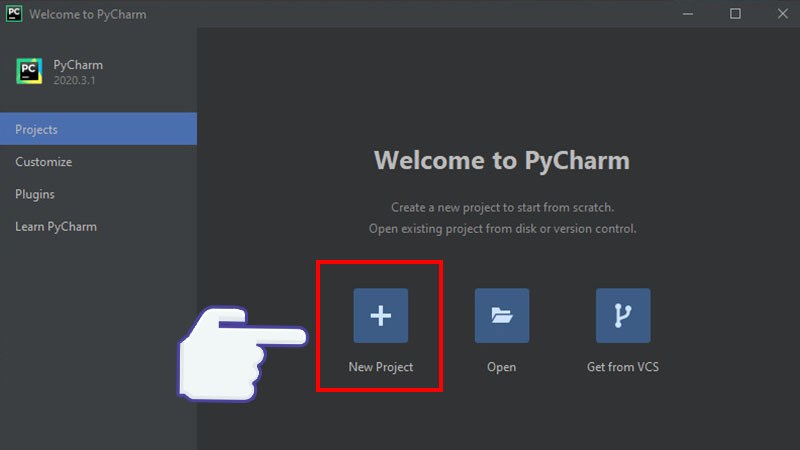
**Bước 8:** Tại giao diện tiếp theo,**bạn tick vào 3 mục: 64-bit launcher, .py và Add launchers dir to the PATH nếu máy bạn chưa cài đặt Java**. Sau đó **nhấn Next**

**Bước 9:** **Tick vào Reboot now**, sau đó **nhấn chọn Finish để hoàn thành cài đặt**

**2. Hướng dẫn sử dụng**

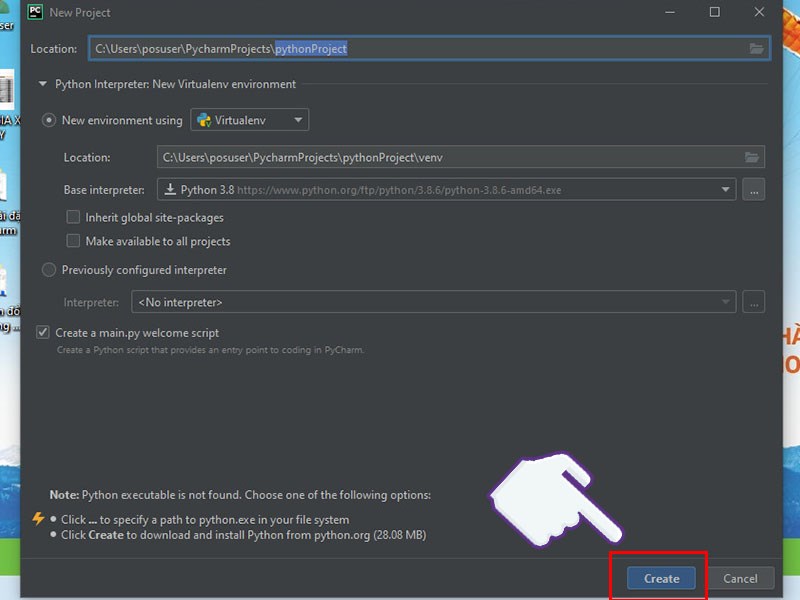
**Bước 1:** Nhấp đúp để mở Pycharm lên.

**Bước 2:** Vào**New** **Project** để tạo một Project mới.



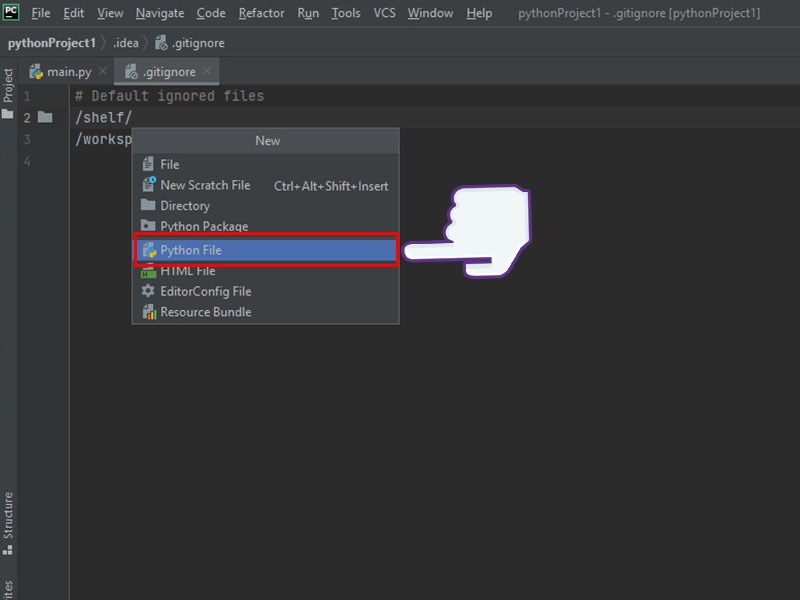
Vào New Project để tạo một Project mới

**Bước 3:** Sau đó bạn chọn thư mục chứa Project mới được tạo, rồi nhấn vào **Create.**



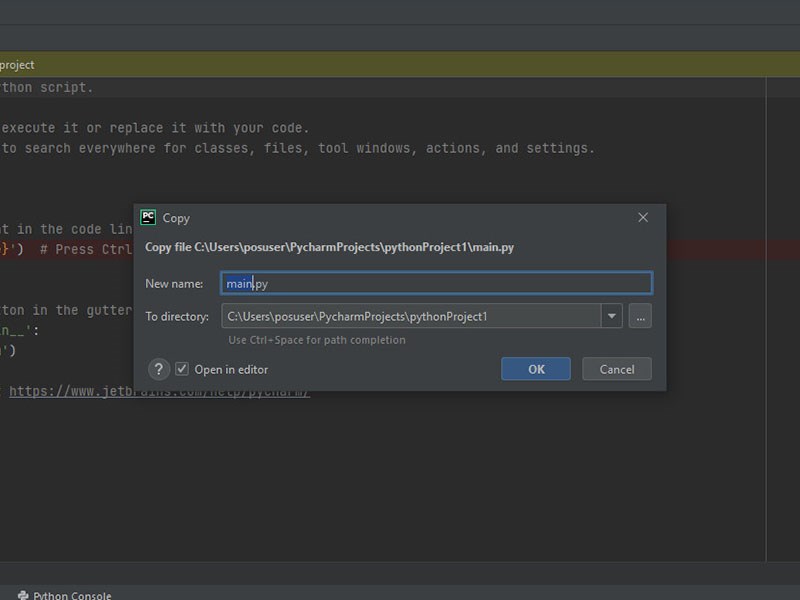
Nhấn vào Create

**Bước 4:**Vào**Python File** để tạo một file sử dụng mã nguồn Python.



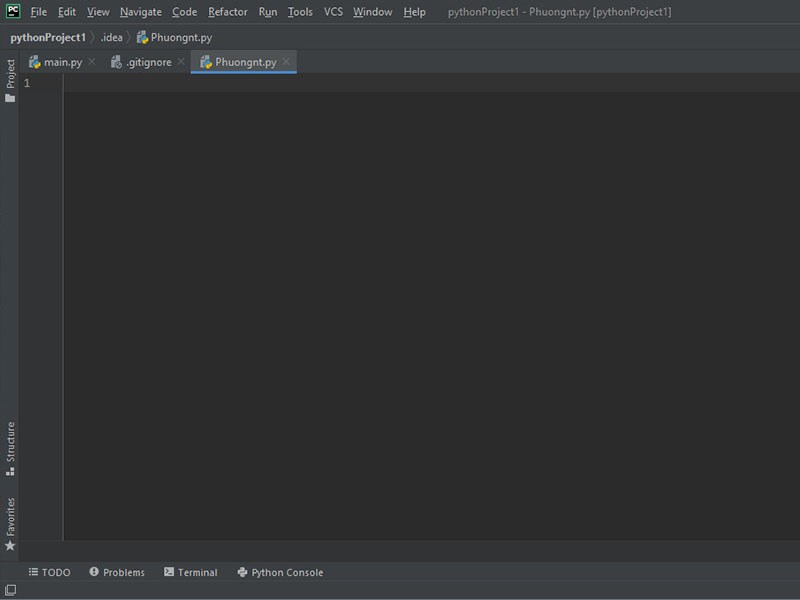
Vào Python File để tạo một file sử dụng mã nguồn Python

**Bước 5:**Đặt tên cho file.



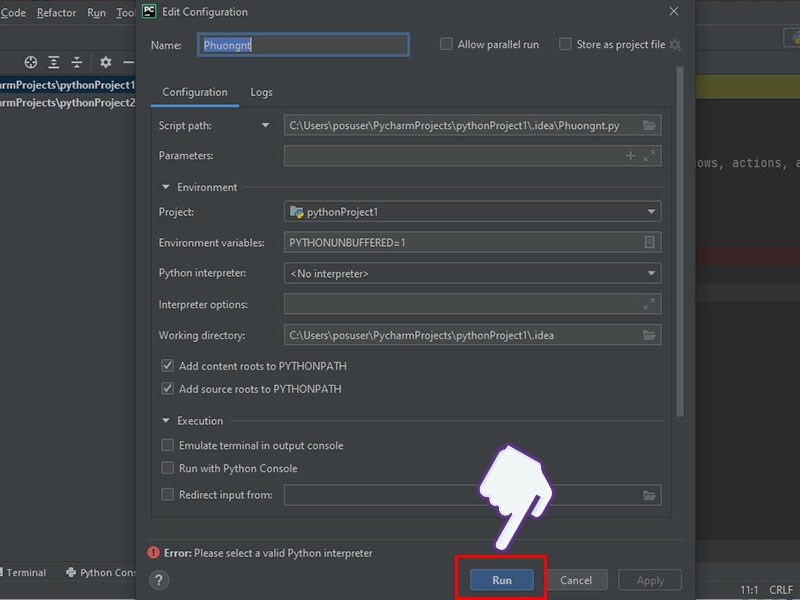
Đặt tên cho file

**Bước 6:** Viết mã nguồn Python.



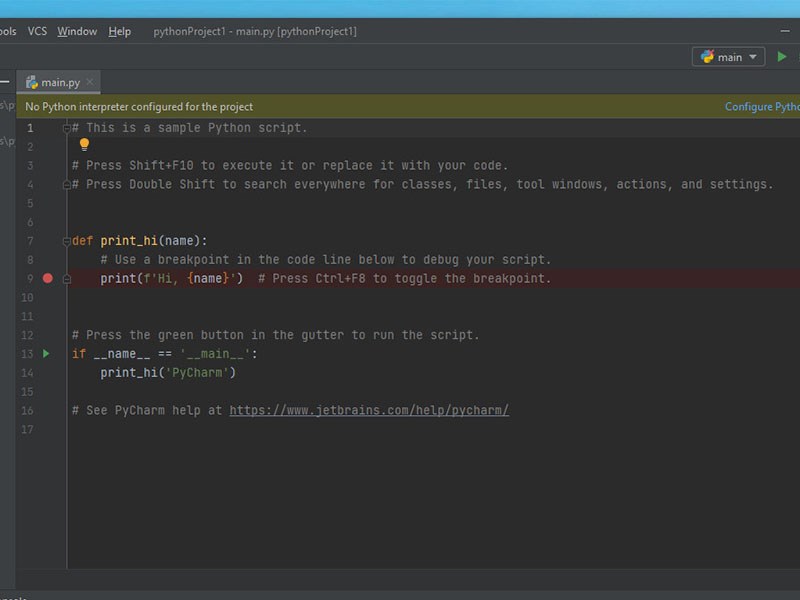
Viết mã nguồn Python

**Bước 7:** Viết xong mã nguồn, bạn nhấn lưu lại, sau đó **nhấn chọn mục Run để thực thi chạy file Python.**



Nhấn chọn mục Run để thực thi file

**Bước 8:** Sau đó, hệ thống bắt đầu chạy thử nghiệm File Python bằng PyCharm.



1. **Tổng kết**

Python là 1 ngôn ngữ bậc cao dễ dùng, có nhiều IDE hỗ trợ

Liên kết kham khảo

Tài liệu Python: <https://developers.google.com/edu/python/>

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