



# Data Preprocessing with Python for Absolute Beginners

---

**Step-by-Step Guide  
with Hands-on Projects  
and Exercises**

Our Books are designed  
to teach beginners  
Data Science and AI

© Bản quyền 2020 của AI Publishing

Mọi quyền được bảo lưu.

Lần in đầu tiên, 2020

Biên tập bởi AI Publishing

Ebook được chuyển đổi và bìa bởi Gazler Studio

Xuất bản bởi AI Publishing LLC

Mã số ISBN-13: 978-1-7347901-0-8

Nội dung của cuốn sách này không được phép sao chép, nhân bản hoặc truyền tải mà không có sự cho phép trực tiếp bằng văn bản của tác giả.

Trong mọi trường hợp, nhà xuất bản sẽ không phải chịu bất kỳ trách nhiệm pháp lý hoặc đổ lỗi nào cho bất kỳ khoản bồi thường, thiệt hại hoặc mất mát tiền tệ nào do thông tin trong đây gây ra, dù trực tiếp hay gián tiếp.

Thông báo pháp lý:

Bạn không được sửa đổi, phân phối, bán, sử dụng, trích dẫn hoặc diễn giải bất kỳ phần nào trong nội dung của cuốn sách này mà không có sự đồng ý cụ thể của tác giả.

Thông báo miễn trừ trách nhiệm:

Xin lưu ý thông tin có trong tài liệu này là dành cho

chỉ nhằm mục đích giáo dục và giải trí. Không có bất kỳ bảo đảm nào được nêu rõ hoặc ngụ ý. Người đọc thừa nhận rằng tác giả không tham gia vào việc cung cấp tư vấn pháp lý, tài chính, y tế hoặc chuyên môn. Vui lòng tham khảo ý kiến của chuyên gia được cấp phép trước khi thử bất kỳ kỹ thuật nào được nêu trong cuốn sách này.

Khi đọc tài liệu này, người đọc đồng ý rằng trong mọi trường hợp, tác giả không chịu trách nhiệm về bất kỳ tổn thất nào, trực tiếp hay gián tiếp, phát sinh do sử dụng thông tin có trong tài liệu này, bao gồm nhưng không giới hạn ở lỗi, thiếu sót hoặc không chính xác.

## Làm thế nào để liên lạc với chúng tôi

Nếu bạn có bất kỳ phản hồi nào, vui lòng cho chúng tôi biết bằng cách gửi một email tới [contact@aipublishing.io](mailto:contact@aipublishing.io).

Phản hồi của bạn được đánh giá rất cao và chúng tôi mong muốn để nghe từ bạn. Nó sẽ có lợi cho chúng tôi để cải thiện chất lượng sách của chúng tôi.

Để lấy mã Python và tài liệu được sử dụng trong cuốn sách này, vui lòng nhấp vào liên kết bên dưới:

[www.aipublishing.io/book-preprocessing-python](http://www.aipublishing.io/book-preprocessing-python)

Số đơn hàng là bắt buộc.

# Về Nhà xuất bản

Tại AI Publishing Company, chúng tôi đã thành lập một công ty quốc tế nền tảng học tập dành riêng cho học sinh trẻ, người mới bắt đầu, các doanh nghiệp nhỏ, các công ty khởi nghiệp và các nhà quản lý mới vào nghề khoa học dữ liệu và trí tuệ nhân tạo.

Thông qua các cuốn sách và khóa học tương tác, mạch lạc và thực tế, chúng tôi giúp người mới bắt đầu học các kỹ năng quan trọng để phát triển các dự án khoa học dữ liệu và AI.

Các khóa học và sách của chúng tôi bao gồm từ các khóa học giới thiệu cơ bản về lập trình ngôn ngữ và khoa học dữ liệu đến các khóa học nâng cao về học máy, học sâu, thị giác máy tính, dữ liệu lớn và nhiều hơn nữa, sử dụng các ngôn ngữ lập trình như Python, R và một số phần mềm khoa học dữ liệu và AI.

Trọng tâm cốt lõi của AI Publishing là giúp người học có thể tạo ra và thử nghiệm các giải pháp chủ động cho các vấn đề kỹ thuật số bằng cách tận dụng tối đa sức mạnh của AI và khoa học dữ liệu.

Hơn nữa, chúng tôi cung cấp hỗ trợ chuyên biệt dưới hình thức của chúng tôi nội dung trực tuyến miễn phí và sách điện tử, cung cấp thông tin cập nhật và hiểu biết hữu ích về các hoạt động AI và các chủ đề khoa học dữ liệu, cùng với việc xóa bỏ những nghi ngờ và quan niệm sai lầm về AI và lập trình.

Các chuyên gia của chúng tôi đã thận trọng phát triển các khóa học trực tuyến và giữ cho chúng súc tích, ngắn gọn và toàn diện để bạn có thể hiểu mọi thứ một cách rõ ràng và hiệu quả, cũng như có thể bắt đầu thực hành các ứng dụng ngay lập tức.

Chúng tôi cũng cung cấp dịch vụ tư vấn và đào tạo doanh nghiệp về AI và khoa học dữ liệu để nhân viên có thể điều hướng quy trình làm việc hiệu quả.

Với AI Publishing, bạn luôn có thể gần gũi hơn với sự đổi mới thế giới AI và khoa học dữ liệu.

Nếu bạn háo hức muốn tìm hiểu từ A đến Z về AI và khoa học dữ liệu nhưng không biết bắt đầu từ đâu, AI Publishing là nơi tốt nhất để đi.

Vui lòng liên hệ với chúng tôi qua email: [contact@aipublishing.io](mailto:contact@aipublishing.io).

# AI Publishing đang tìm kiếm những tác giả như bạn

Bạn có hứng thú trở thành tác giả cho AI Publishing không? Vui lòng  
liên hệ với chúng tôi tại [author@aipublishing.io](mailto:author@aipublishing.io).

Chúng tôi đang làm việc với các nhà phát triển và chuyên gia công nghệ AI giống  
như bạn, để giúp họ chia sẻ hiểu biết của mình với những người yêu thích AI và  
Khoa học dữ liệu toàn cầu. Bạn có thể chia sẻ mọi kiến thức của mình về các chủ  
đề nóng trong AI và Khoa học dữ liệu.

# Tải xuống Hình ảnh màu

Chúng tôi cũng cung cấp tệp PDF có hình ảnh màu của ảnh chụp màn hình/sơ đồ được sử dụng trong cuốn sách này.

Bạn có thể tải xuống ở đây:

[www.aipublishing.io/book-preprocessing-python](http://www.aipublishing.io/book-preprocessing-python)

# Liên hệ với chúng tôi

Chúng tôi luôn hoan nghênh phản hồi từ độc giả.

Để nhận phản hồi chung, vui lòng gửi email cho chúng  
tôi theo địa chỉ [contact@aipublishing.io](mailto:contact@aipublishing.io)  
và đề cập đến tựa sách trong tiêu đề tin nhắn của bạn.

Mặc dù chúng tôi đã hết sức cẩn thận để đảm bảo tính chính xác của nội  
dung, nhưng vẫn có thể xảy ra lỗi. Nếu bạn tìm thấy lỗi trong  
cuốn sách này, chúng tôi sẽ rất biết ơn nếu bạn có thể báo cáo lỗi  
này cho chúng tôi sớm nhất có thể.

Nếu bạn quan tâm đến việc trở thành tác giả của AI Publishing và nếu  
có chủ đề nào đó mà bạn có chuyên môn và bạn muốn viết  
hoặc đóng góp vào một cuốn sách, vui lòng gửi email cho chúng tôi  
theo địa chỉ [author@aipublishing.io](mailto:author@aipublishing.io).

---



# Mục lục

Cách liên hệ với chúng tôi .....	iii
Về Nhà xuất bản.....	iv
AI Publishing đang tìm kiếm những tác giả như bạn .....	vi
Tái xuống hình ảnh màu.....	vii
Liên hệ với chúng tôi.....	viii
Lời nói đầu .....	
Về tác giả.	
Chương 1: Giới thiệu.	
1.1. Tiền xử lý dữ liệu là gì? .....	7
1.2. Thiết lập môi trường .....	8
Windows .....	8
Thiết lập Mac.....	13
Thiết lập Linux.....	19
1.3. Khóa học Python Crash Course.....	22
1.3.1. Viết chương trình đầu tiên của bạn.....	22
1.3.2. Biến và kiểu dữ liệu Python.....	26
1.3.3. Toán tử Python.....	28
1.3.4. Câu lệnh điều kiện.....	35
1.3.5. Các câu lệnh lặp lại.....	37
1.3.6. Chức năng .....	39
1.3.7. Đối tượng và Lớp.....	41
1.4. Các thư viện khác nhau để xử lý dữ liệu trước .....	43
1.4.1. NumPy .....	43

1.4.2. Học Scikit .....	43
1.4.3. Matplotlib.....	44
1.4.4. Sinh vật biển .....	44
1.4.5. Gấu trúc .....	44
Bài tập 1.1.....	45
Bài tập 1.2 .....	46

Chương 2: Tìm hiểu về các kiểu dữ liệu.

2.1. Giới thiệu .....	47
2.1.1. Biến là gì? .....	47
2.1.2. Kiểu dữ liệu.....	48
2.2. Dữ liệu số.....	48
2.2.1. Dữ liệu rời rạc.....	50
2.2.2. Dữ liệu liên tục .....	51
2.2.3. Dữ liệu nhị phân .....	52
liệu phân loại .....	53
2.3.1. Dữ liệu thứ tự.....	54
2.3.2. Dữ liệu danh nghĩa.....	55
2.4. Dữ liệu ngày và giờ .....	57
2.5. Kiểu dữ liệu hỗn hợp.....	57
Giá trị bị thiếu.....	58
2.6.1. Nguyên nhân gây ra dữ liệu bị thiếu.....	58
2.6.2. Nhược điểm của dữ liệu bị thiếu.....	59
2.6.3. Cơ chế đằng sau giá trị bị thiếu.....	59
2.7. Tính số lượng trong dữ liệu danh mục.....	60
2.8. Phân phối xác suất .....	61
2.9. Giá trị ngoại lệ .....	64
Bài tập 2.1.....	66

Chương 3: Xử lý dữ liệu bị thiếu...

3.1. Giới thiệu .....	69
3.2. Phân tích trường hợp hoàn chỉnh.....	69
3.3. Xử lý dữ liệu số bị thiếu.....	71
3.3.1. Tính toán trung bình hoặc trung vị.....	71
3.3.2. Tính toán cuối phần phối.....	78

3.3.3. Quy kết giá trị tùy ý.....	84
3.4. Xử lý dữ liệu danh mục bị thiếu.....	88
Quy kết danh mục thường xuyên.....	88
3.4.2. Thiếu sự quy kết danh mục.....	94
Bài tập 3.1.....	96
Bài tập 3.2 .....	97

## Chương 4: Mã hóa dữ liệu phân loại...

4.1. Giới thiệu .....	99
4.2. Một mã hóa nóng.....	99
4.3. Mã hóa nhân.....	106
4.4. Mã hóa tần số.....	108
hóa thứ tự.....	111
bình.....	113
Bài tập 4.1 .....	115
Bài tập 4.2.....	116

## Chương 5: Phân rã dữ liệu....

5.1. Giới thiệu .....	117
5.2. Phân biệt độ rộng bằng nhau .....	118
biệt tần số bằng nhau .....	124
5.4. Phân biệt K-Means.....	129
5.5. Phân rã cây quyết định.....	133
5.6. Phân biệt tùy chỉnh.....	136
Bài tập 5.1.....	139
Bài tập 5.2 .....	140

## Chương 6: Xử lý giá trị ngoại lai...

6.1. Giới thiệu .....	141
6.2. Cắt bỏ giá trị ngoại lệ .....	142
Giới hạn giá trị ngoại lệ bằng IQR.....	146
trị ngoại lệ bằng Mean và Std.....	150
Quantiles .....	153
chỉnh.....	157
Bài tập 6.1 .....	159

Bài tập 6.2 .....	160
-------------------	-----

## Chương 7: Mở rộng tính năng...

7.1. Giới thiệu .....	161
7.2. Chuẩn hóa .....	161
7.3. Tỷ lệ Min/Max.....	165
7.4. Chuẩn hóa trung bình.....	167
7.5. Tỷ lệ tuyệt đối tối đa .....	169
7.6. Tỷ lệ trung vị và tỷ lệ phân vị.....	171
7.7. Tỷ lệ độ dài đơn vị vectơ .....	173
Bài tập 7.1.....	175
Bài tập 7.2.....	176

## Chương 8: Xử lý biến hỗn hợp và biến DateTime....

8.1. Giới thiệu .....	177
8.2. Xử lý các giá trị hỗn hợp.....	177
8.3. Xử lý kiểu dữ liệu ngày tháng.....	182
8.4. Xử lý kiểu dữ liệu thời gian .....	186
Bài tập 8.1 .....	190
Bài tập 8.2 .....	191

## Chương 9: Xử lý tập dữ liệu không cân bằng....

9.1. Giới thiệu .....	193
9.2. Ví dụ về tập dữ liệu mất cân bằng .....	193
9.3. Lấy mẫu xuống .....	197
9.4. Lấy mẫu lên.....	199
9.5. Lấy mẫu SMOTE Up .....	200
Bài tập 9.1 .....	202

## Dự án cuối cùng - Một quy trình xử lý dữ liệu hoàn chỉnh...

1.1. Giới thiệu .....	203
1.2. Tiền xử lý dữ liệu.....	204
1.3. Dự án phân loại.....	205
1.4. Dự án hồi quy.....	211

Bài tập 2.1.....	219
Bài tập 3.1.....	221
Bài tập 3.2 .....	223
Bài tập 4.1 .....	224
Bài tập 4.2.....	226
Bài tập 5.1.....	227
Bài tập 5.2 .....	228
Bài tập 6.1 .....	229
Bài tập 6.2 .....	230
Bài tập 7.1.....	232
Bài tập 7.2.....	233
Bài tập 8.1 .....	235
Bài tập 8.2 .....	236
Bài tập 9.1 .....	237



# Lời nói đầu

## S Phương pháp tiếp cận sách

Cuốn sách theo một cách tiếp cận rất đơn giản. Nó được chia thành chín chương.

Chương 1 giới thiệu khái niệm cơ bản về xử lý dữ liệu trước, cùng với các bước cài đặt phần mềm mà chúng ta sẽ cần để thực hiện xử lý dữ liệu trước trong cuốn sách này.

Chương 1 cũng chứa một khóa học cấp tốc về Python.

Chương 2 cung cấp tổng quan ngắn gọn về các kiểu dữ liệu khác nhau.

Chương 3 giải thích cách xử lý các giá trị bị thiếu trong dữ liệu, trong khi mã hóa theo danh mục dữ liệu số được giải thích trong Chương 4. Phân loại dữ liệu được trình bày trong Chương 5.

Chương 6 giải thích quá trình xử lý ngoại lệ đường nét thủ công, trong khi Chương 7 giải thích cách chia tỷ lệ các tính năng trong tập dữ liệu.

Việc xử lý dữ liệu hỗn hợp và dữ liệu ngày giờ được giải thích trong Chương 8, trong khi việc cân bằng dữ liệu và lấy mẫu lại được giải thích trong Chương 9. Một dự án cuối cùng về tiền xử lý dữ liệu đầy đủ cũng có sẵn ở cuối sách.

Trong mỗi chương, các loại kỹ thuật xử lý dữ liệu khác nhau đã được giải thích về mặt lý thuyết, sau đó là các ví dụ thực tế. Mỗi chương cũng có một bài tập mà sinh viên có thể sử dụng để đánh giá sự hiểu biết của mình về các khái niệm được giải thích trong chương. Sổ tay Python dành cho

## 2 | Lời nói đầu

---

mỗi chương được cung cấp trong các tài nguyên. Bạn nên tự viết mã thay vì sao chép mã và trong trường hợp có lỗi, bạn hãy đối chiếu mã của mình với sổ ghi chép Python tương ứng, tìm và sau đó sửa lỗi. Các tập dữ liệu

được sử dụng trong cuốn sách này được tải xuống khi chạy hoặc là có sẵn trong thư mục Tài nguyên/Bộ dữ liệu .

### § Khoa học dữ liệu và xử lý dữ liệu trước

Khoa học dữ liệu và tiền xử lý dữ liệu là hai khái niệm khác nhau nhưng có liên quan với nhau. Khoa học dữ liệu đề cập đến khoa học về việc trích xuất và khám phá dữ liệu để tìm ra các mẫu có thể được sử dụng để ra quyết định ở các cấp độ khác nhau. Tiền xử lý dữ liệu là quá trình tiền xử lý dữ liệu theo cách có thể truyền tải thông tin hữu ích có thể được sử dụng làm đầu vào cho nhiều quy trình khoa học dữ liệu khác nhau như trực quan hóa dữ liệu, học máy, học sâu, v.v.

Cuốn sách này dành riêng cho việc xử lý trước dữ liệu và giải thích cách thực hiện các kỹ thuật xử lý trước dữ liệu khác nhau trên nhiều tập dữ liệu khác nhau bằng cách sử dụng nhiều thư viện xử lý trước dữ liệu được viết bằng ngôn ngữ lập trình Python. Bạn nên sử dụng cuốn sách này chỉ cho mục đích xử lý trước dữ liệu chứ không phải cho khoa học dữ liệu hoặc học máy. Để ứng dụng xử lý trước dữ liệu trong khoa học dữ liệu và học máy, hãy đọc cuốn sách này cùng với các cuốn sách chuyên đề về học máy và khoa học dữ liệu.

### § Cuốn sách này dành cho ai?

Cuốn sách này giải thích quá trình xử lý dữ liệu trước bằng cách sử dụng nhiều thư viện khác nhau từ đầu. Do đó, cuốn sách này hướng đến

lý tưởng cho người mới bắt đầu xử lý dữ liệu trước. Mặc dù có nền tảng về ngôn ngữ lập trình Python và tính năng



kỹ thuật có thể giúp tăng tốc quá trình học, cuốn sách có khóa học cấp tốc về ngôn ngữ lập trình Python trong chương đầu tiên. Do đó, điều kiện tiên quyết duy nhất để sử dụng hiệu quả cuốn sách này là máy tính có kết nối internet. Tất cả các mã và tập dữ liệu đã được cung cấp. Tuy nhiên, để tải xuống các thư viện tiền xử lý dữ liệu, bạn sẽ cần có internet.

Ngoài những người mới bắt đầu xử lý dữ liệu bằng Python, cuốn sách này cũng có thể được sử dụng như một tài liệu tham khảo cho các lập trình viên trung cấp và có kinh nghiệm vì nó chứa các mẫu mã xử lý dữ liệu bằng cách sử dụng nhiều hình ảnh hóa dữ liệu thư viện.

### § Sử dụng cuốn sách này như thế nào?

Như tôi đã nói trước đó, các kỹ thuật và khái niệm tiền xử lý dữ liệu được dạy trong cuốn sách này đã được chia thành nhiều chương. Để tận dụng tối đa cuốn sách này, tôi khuyên bạn trước tiên hãy làm quen với ngôn ngữ lập trình Python, đặc biệt là các khái niệm lập trình hướng đối tượng. Để làm như vậy, bạn có thể tham gia khóa học cấp tốc về Python trong chương 1 của cuốn sách này. Ngoài ra, hãy cố gắng đọc các chương của cuốn sách này theo thứ tự vì các khái niệm được dạy trong các chương sau dựa trên các chương trước. Trong mỗi chương, trước tiên hãy cố gắng hiểu các khái niệm lý thuyết đằng sau các loại kỹ thuật tiền xử lý dữ liệu khác nhau và sau đó thử thực thi mã ví dụ. Tôi xin nhấn mạnh lại rằng thay vì sao chép và dán mã, hãy thử tự viết mã và trong trường hợp có bất kỳ lỗi nào, bạn có thể đối chiếu mã của mình với mã nguồn được cung cấp trong sách cũng như trong sổ tay Python trong tài liệu tham khảo. Cuối cùng, hãy cố gắng trả lời các câu hỏi trong các bài tập ở cuối mỗi chương. Các giải pháp cho các bài tập đã được đưa ra ở cuối sách.

Để tạo điều kiện thuận lợi cho quá trình đọc, thỉnh thoảng, cuốn sách trình bày ba loại thẻ hộp có màu sắc khác nhau: **Yêu cầu**, **Đọc thêm** và **Thời gian thực hành**. Ví dụ về các loại này

Các hộp được hiển thị bên dưới.

#### Yêu cầu

Hộp này liệt kê tất cả các yêu cầu cần thực hiện trước khi chuyển sang chủ đề tiếp theo. Nói chung, nó hoạt động như một danh sách kiểm tra để xem mọi thứ đã sẵn sàng trước khi hướng dẫn hay chưa.

#### Đọc thêm

Tại đây, bạn sẽ được chỉ đến một số tài liệu tham khảo hoặc nguồn bên ngoài đóng vai trò là nội dung bổ sung về Chủ đề **cụ thể** đang được nghiên cứu. Nhìn chung, nó bao gồm các gói, tài liệu và bảng hướng dẫn.

#### Thời gian thực hành

Tại đây, bạn sẽ được chuyển đến một tệp bên ngoài để đào tạo và kiểm tra mọi kiến thức thu được về Công cụ đã được nghiên cứu. Nhìn chung, các tệp này là sổ ghi chép Jupyter (.ipynb), tệp Python (.py) hoặc tài liệu (.pdf).

**Yêu cầu** của box-tag liệt kê các bước mà người đọc cần thực hiện sau khi đọc một hoặc nhiều chủ đề. **Đọc thêm** cung cấp các tài liệu tham khảo có liên quan cho các chủ đề cụ thể để tìm hiểu thêm nội dung của các chủ đề. **Thời gian thực hành** chỉ ra các công cụ thực tế để bắt đầu làm việc trên các chủ đề đã chỉ định.

Thực hiện theo các hướng dẫn trong hộp nhãn để hiểu rõ hơn về các chủ đề được trình bày trong cuốn sách này.

## Về tác giả



M. Usman Malik có bằng Tiến sĩ Khoa học Máy tính từ Đại học Normandy, Pháp, với Trí tuệ nhân tạo và Học máy là các lĩnh vực nghiên cứu chính. Usman Malik có hơn năm năm kinh nghiệm trong ngành Khoa học dữ liệu và đã làm việc với cả các tổ chức khu vực tư nhân và công cộng. Trong thời gian rảnh rỗi, anh ấy thích nghe nhạc và chơi bi-da.

# Cảnh báo

Trong Python, thực lè rất quan trọng. Thực lè Python là một cách để nói với trình thông dịch Python rằng nhóm các câu lệnh thuộc về một khối mã cụ thể. Sau mỗi vòng lặp hoặc nếu có điều kiện, hãy chắc chắn chú ý kỹ đến ý định.

## Ví dụ

```
# Python program showing  
# indentation  
  
site = 'aisciences'  
  
if site == 'aisciences':  
    print('Logging to www.aisciences.io...')  
else:  
    print('retype the URL.')  
print('All set !')
```

Để tránh các vấn đề trong quá trình thực hiện, chúng tôi khuyên bạn nên tải xuống các mã có sẵn trên Github bằng cách yêu cầu quyền truy cập từ liên kết bên dưới. Vui lòng chuẩn bị sẵn số đơn hàng của bạn  
truy cập:

[www.aipublishing.io/book-preprocessing-python](http://www.aipublishing.io/book-preprocessing-python)

# 1

## Giới thiệu

### 1.1 Tiền xử lý dữ liệu là gì

Với lượng dữ liệu khổng lồ có sẵn, ngày càng nhiều nhà nghiên cứu và chuyên gia trong ngành tìm ra cách sử dụng dữ liệu này cho mục đích nghiên cứu và lợi ích thương mại. Tuy nhiên, hầu hết dữ liệu có sẵn theo mặc định đều quá thô. Điều quan trọng là phải xử lý dữ liệu trước khi có thể sử dụng để xác định các mẫu quan trọng hoặc có thể sử dụng để đào tạo các mô hình thống kê có thể được sử dụng để đưa ra dự đoán.

Tiền xử lý dữ liệu là quá trình làm sạch và thiết kế dữ liệu theo cách có thể được sử dụng làm đầu vào cho một số tác vụ khoa học dữ liệu quan trọng như trực quan hóa dữ liệu, học máy, học sâu và phân tích dữ liệu.

Một số tác vụ tiền xử lý dữ liệu phổ biến nhất bao gồm kỹ thuật tính năng, thay đổi tỷ lệ tính năng, phát hiện giá trị ngoại lai, xử lý giá trị bị thiếu, mã hóa biến phân loại, phân biệt dữ liệu, v.v. Bạn sẽ thấy tất cả các khái niệm này một cách chi tiết trong chương này, với sự trợ giúp của nhiều ví dụ được triển khai trong ngôn ngữ lập trình Python.

Trong chương này, bạn sẽ thấy cách thiết lập Python môi trường cần thiết để chạy nhiều thư viện trực quan hóa dữ liệu.

Chương này cũng có một khóa học Python cấp tốc dành cho người mới bắt đầu học Python. Cuối cùng, các thư viện tiền xử lý dữ liệu khác nhau mà chúng ta sẽ nghiên cứu trong cuốn sách này đã được thảo luận. Chương này kết thúc bằng một bài tập đơn giản.

## 1.2 Bộ môi trường

### 1.2.1. Cài đặt Windows

Đã đến lúc cài đặt Python trên Windows bằng IDE. Trên thực tế, chúng ta sẽ sử dụng Anaconda trong suốt cuốn sách này ngay từ khi cài đặt Python cho đến khi viết mã đa luồng trong các bài giảng sắp tới. Bây giờ, chúng ta hãy bắt đầu cài đặt.

Phần này giải thích cách bạn có thể tải xuống và cài đặt Anaconda trên Windows.

Để tải xuống và cài đặt Anaconda, hãy làm theo các bước sau.

1. Mở URL sau trong trình duyệt của bạn.

<https://www.anaconda.com/distribution/>

2. Trình duyệt sẽ đưa bạn tới trang web sau.

Chọn phiên bản Python mới nhất (3.8 tại thời điểm viết cuốn sách này). Bây giờ, hãy nhấp vào nút Tải xuống để tải xuống tệp thực thi. Tùy thuộc vào tốc độ internet của bạn, tệp sẽ tải xuống trong vòng 2-3 phút.

 Windows |  macOS |  Linux

## Anaconda 2019.07 for Windows Installer

### Python 3.7 version

[Download](#)

64-Bit Graphical Installer (486 MB)  
32-Bit Graphical Installer (418 MB)

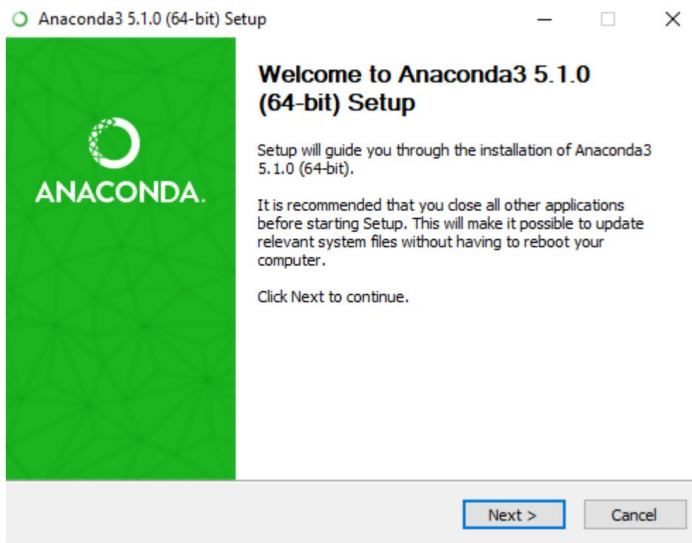
### Python 2.7 version

[Download](#)

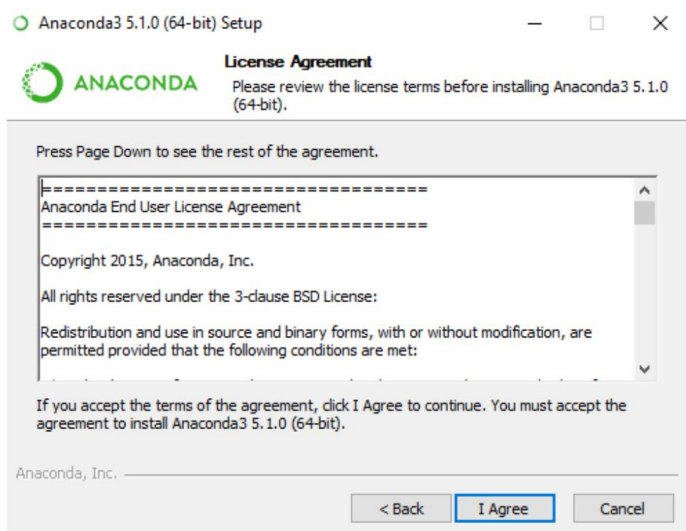
64-Bit Graphical Installer (427 MB)  
32-Bit Graphical Installer (361 MB)

3. Chạy tệp thực thi sau khi tải xuống hoàn tất.

Bạn có nhiều khả năng tìm thấy tệp đã tải xuống trong thư mục tải xuống. Tên của tệp phải tương tự đến "Anaconda3-5.1.0-Windows-x86\_64." Trình hướng dẫn cài đặt sẽ mở ra khi bạn chạy tệp, như thể hiện trong hình sau. Nhấp vào nút Tiếp theo .

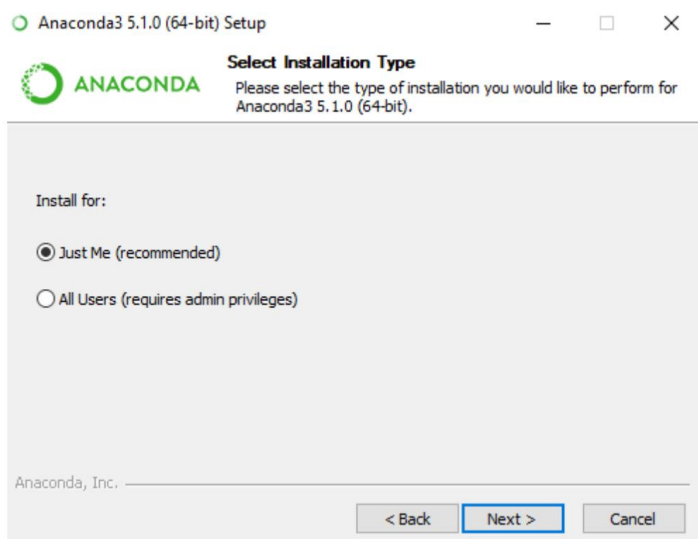


4. Bây giờ hãy nhấp vào Tôi đồng ý trên hộp thoại Thỏa thuận cấp phép , như được hiển thị trong ảnh chụp màn hình sau.



#### 5. Kiểm tra nút radio Chỉ tôi từ Chọn

Hộp thoại Loại cài đặt . Nhấp vào nút Tiếp theo để  
Tiếp tục.

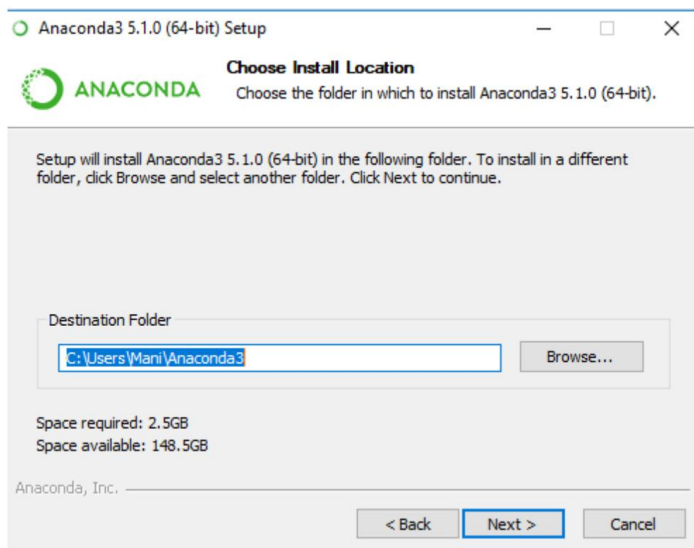


#### 6. Bây giờ, hộp thoại Chọn vị trí cài đặt sẽ được hiển thị.

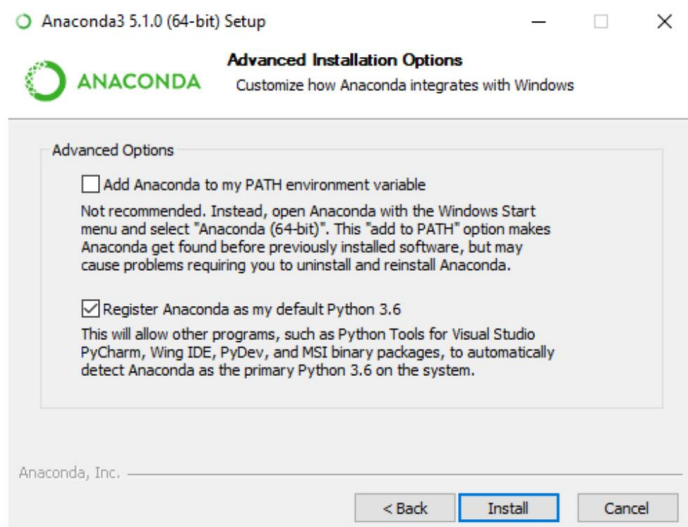
Thay đổi thư mục nếu bạn muốn, nhưng mặc định là tốt nhất. Thư mục cài đặt  
phải có ít nhất 3



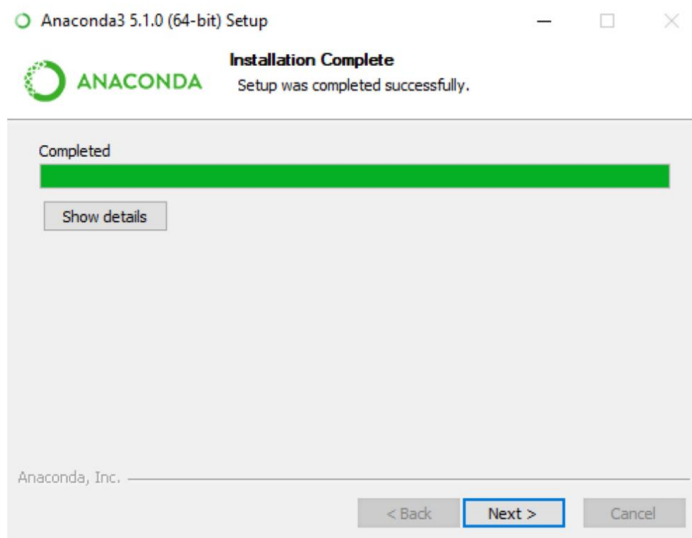
GB dung lượng trống cho Anaconda. Nhấp vào nút Tiếp theo .



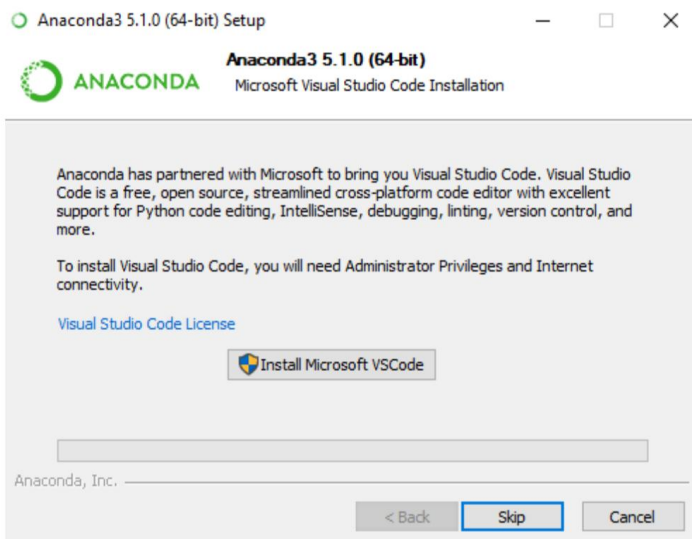
7. Chọn tùy chọn thứ hai, Đăng ký Anaconda làm Python 3.8 mặc định của tôi trong Tùy chọn cài đặt nâng cao  
hộp thoại. Nhấp vào nút Cài đặt để bắt đầu cài đặt, có thể mất một thời gian để hoàn tất.



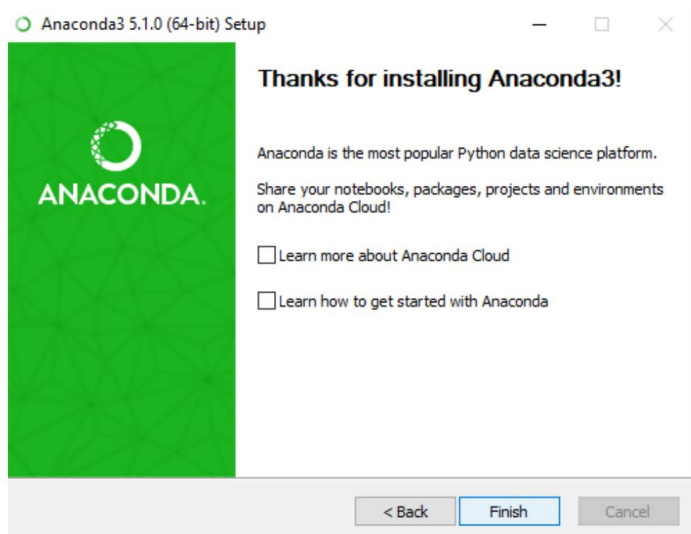
8. Nhấp vào Tiếp theo sau khi quá trình cài đặt hoàn tất.



9. Nhấp vào Bỏ qua trên hộp thoại Cài đặt Microsoft Visual Studio Code .



10. Bạn đã cài đặt thành công Anaconda trên Windows của mình. Tuyệt vời. Bước tiếp theo là bỏ chọn cả hai hộp kiểm trên hộp thoại. Bây giờ, hãy nhấp vào Nút kết thúc .



### 1.2.2. Cài đặt Mac

Quá trình cài đặt Anaconda gần như giống hệt với Mac.

Có thể khác nhau về mặt đồ họa, nhưng bạn sẽ làm theo các bước tương tự như bạn được theo dõi cho Windows. Sự khác biệt duy nhất là bạn phải tải xuống tệp thực thi tương thích với Hệ điều hành Mac.

Phần này giải thích cách bạn có thể tải xuống và cài đặt Anaconda trên máy Mac.

Để tải xuống và cài đặt Anaconda, hãy làm theo các bước sau.

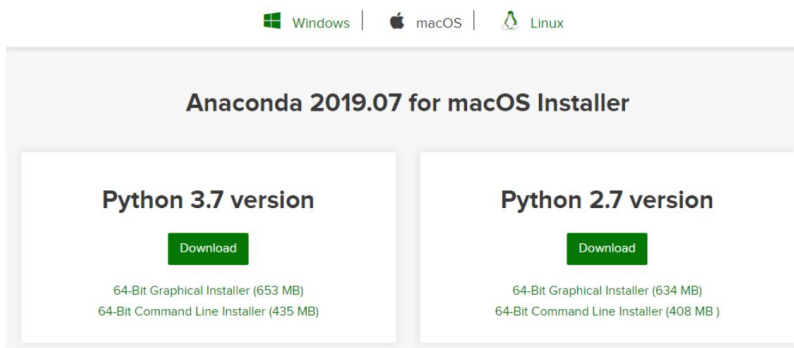
1. Mở URL sau trong trình duyệt của bạn.

<https://www.anaconda.com/distribution/>

2. Trình duyệt sẽ đưa bạn tới trang web sau.

Chọn phiên bản mới nhất của Python cho Mac (3.8 tại thời điểm viết cuốn sách này). Bây giờ, hãy nhấp vào Tải xuống nút để tải xuống tệp thực thi. Tùy thuộc vào

tùy thuộc vào tốc độ internet của bạn, tập tin sẽ được tải xuống trong vòng 2-3 phút.

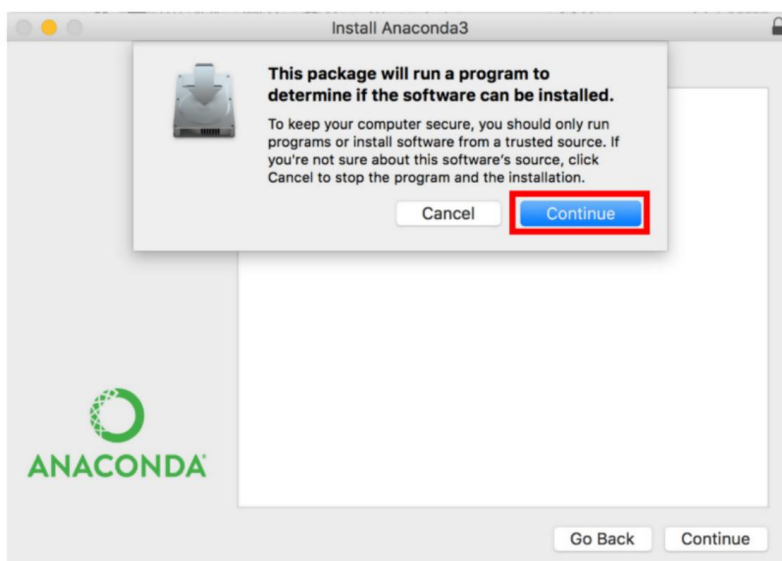


3. Chạy tệp thực thi sau khi tải xuống hoàn tất.

Bạn có nhiều khả năng tìm thấy tệp đã tải xuống trong thư mục tải xuống. Tên của tệp phải tương tự

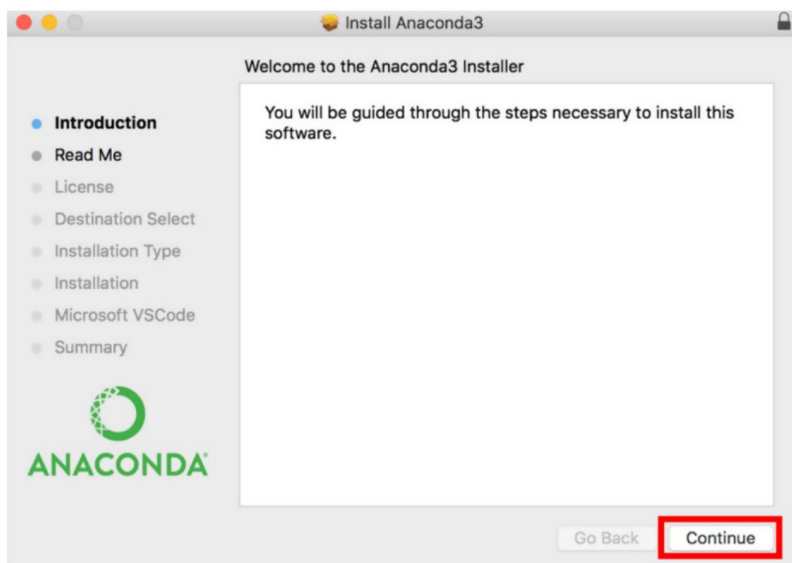
đến "Anaconda3-5.1.0-Windows-x86\_64." Cài đặt

trình hướng dẫn sẽ mở khi bạn chạy tệp, như được hiển thị trong hình sau. Nhấp vào nút Tiếp tục .



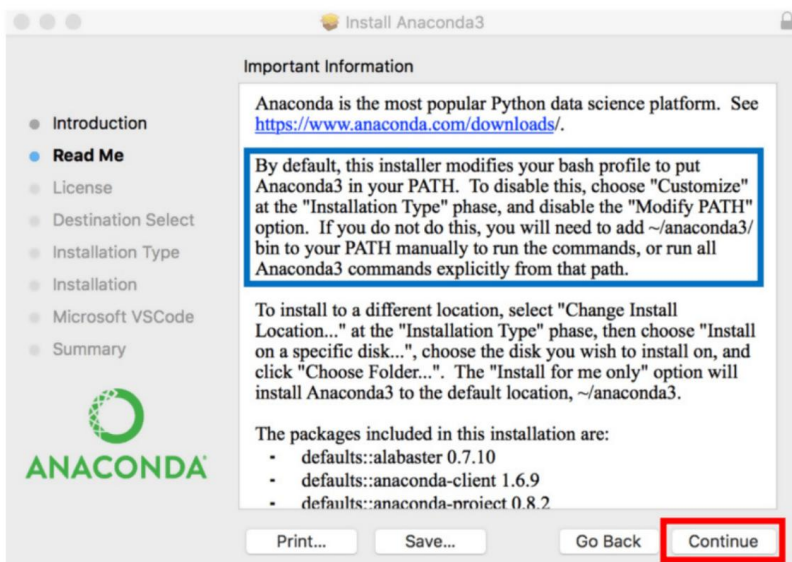
4. Bây giờ hãy nhấp vào Tiếp tục trên Welcome to Anaconda 3

Cửa sổ cài đặt như hiển thị trong ảnh chụp màn hình sau.



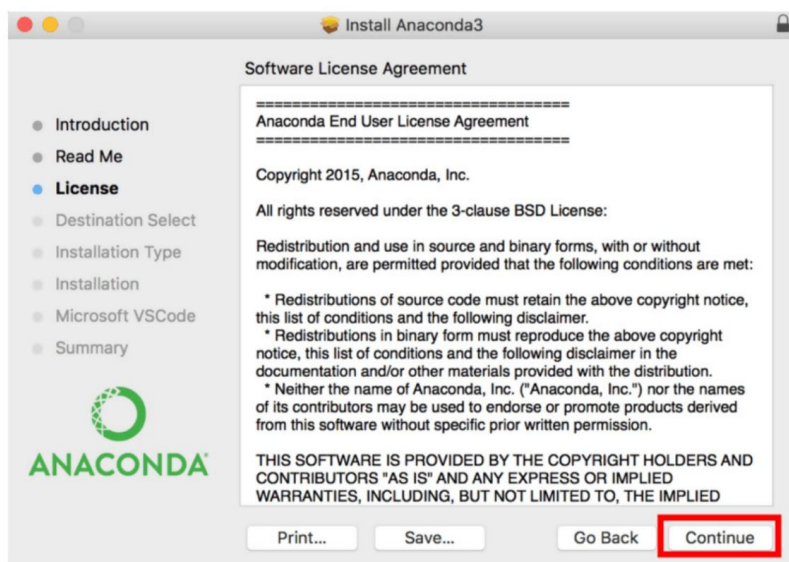
5. Hộp thoại Thông tin quan trọng sẽ bật lên. Chỉ cần nhấp vào Tiếp tục để sử dụng phiên bản mặc định

Trần Nam Mỹ 3.

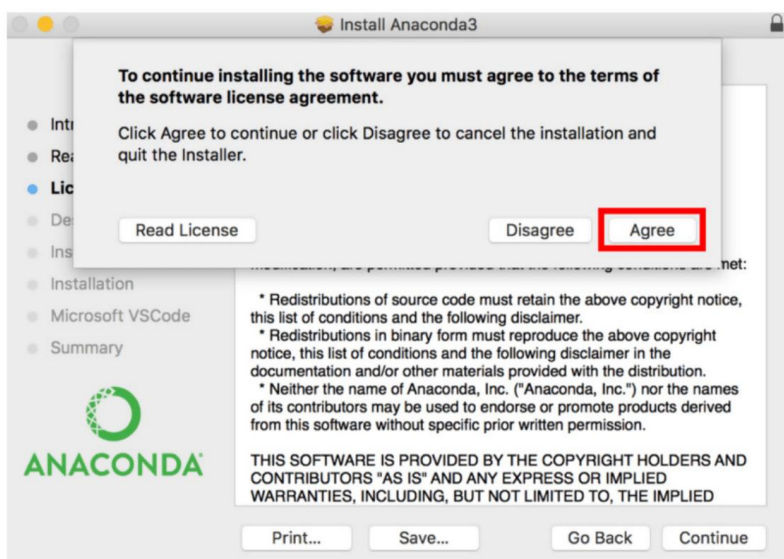


## 6. Nhấp vào Tiếp tục trên Thỏa thuận cấp phép phần mềm

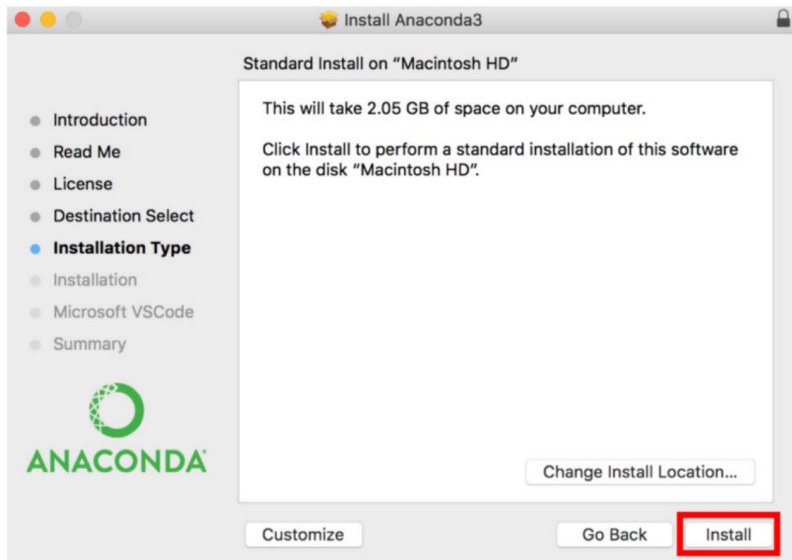
Đối thoại.



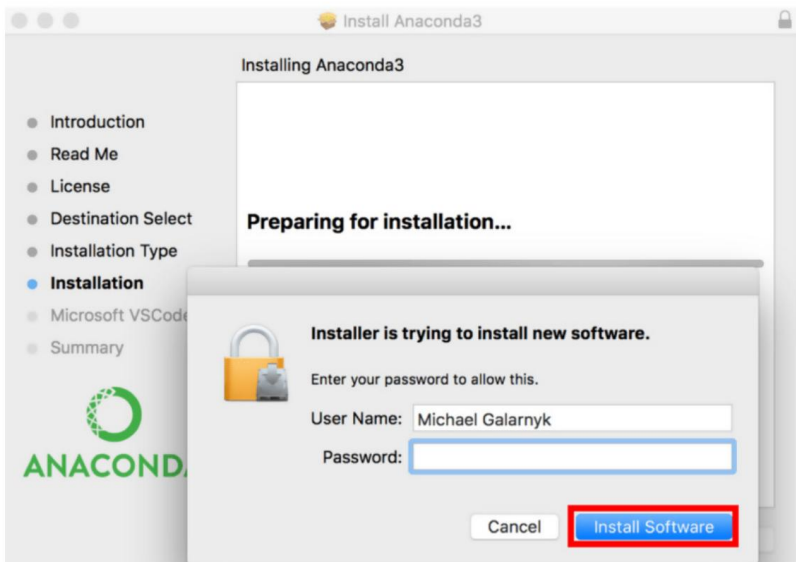
## 7. Bạn phải đọc thỏa thuận cấp phép và nhấp vào nút Đồng ý trước khi bạn có thể nhấp vào Tiếp tục nút một lần nữa.



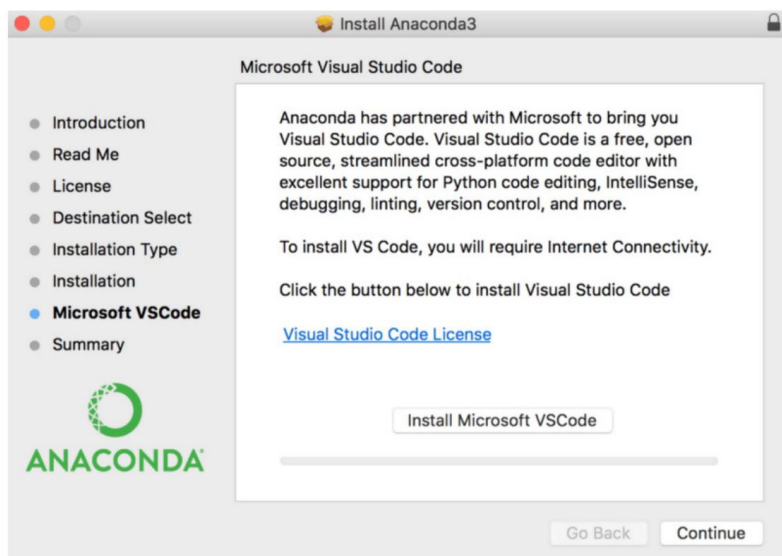
8. Chỉ cần nhấp vào Cài đặt trên cửa sổ tiếp theo xuất hiện.



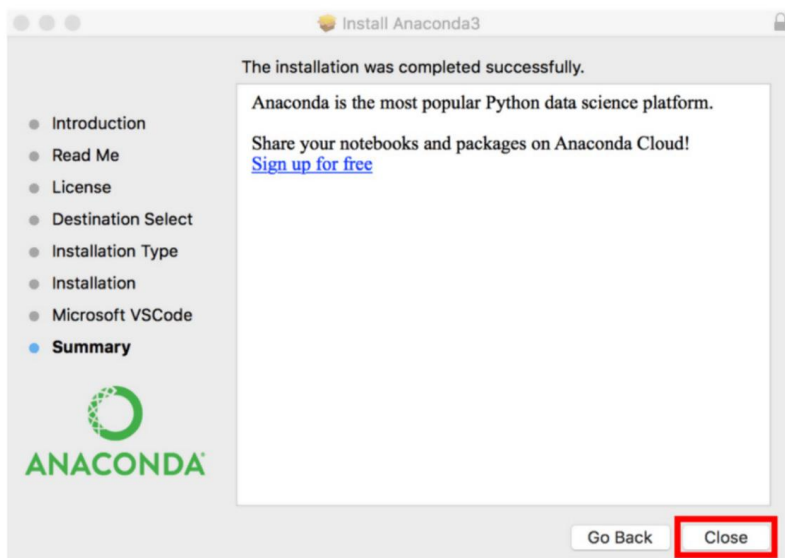
Hệ thống sẽ nhắc bạn cung cấp mật khẩu. Sử dụng cùng mật khẩu bạn dùng để đăng nhập vào máy tính Mac. Bây giờ, nhấp vào Cài đặt phần mềm.



9. Nhấp vào Tiếp tục trên cửa sổ tiếp theo. Bạn cũng có  
tùy chọn cài đặt Microsoft VSCode tại thời điểm này.



Màn hình tiếp theo sẽ hiển thị thông báo cài đặt đã hoàn tất thành công. Nhấp vào nút Đóng để đóng  
trình cài đặt.





Vậy là xong. Bạn đã cài đặt thành công Anaconda trên máy tính Mac của mình. Bây giờ, bạn có thể viết mã Python trong Jupyter và Spyder theo cùng cách bạn đã viết trong Windows.

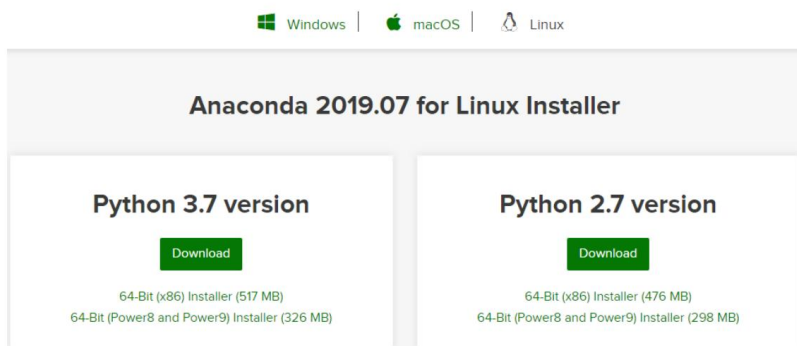
### 1.2.3. Cài đặt Linux

Chúng tôi đã sử dụng trình cài đặt đồ họa của Python để cài đặt trên Windows và Mac. Tuy nhiên, chúng tôi sẽ sử dụng dòng lệnh để cài đặt Python trên Ubuntu hoặc Linux. Linux cũng thân thiện với tài nguyên hơn và việc cài đặt phần mềm này cũng đặc biệt dễ dàng.

Thực hiện theo các bước sau để cài đặt Anaconda trên Linux (bản phân phối Ubuntu).

1. Truy cập vào liên kết sau để sao chép tập lệnh bash cài đặt từ phiên bản mới nhất hiện có.

<https://www.anaconda.com/distribution/>



2. Bước thứ hai là tải xuống tập lệnh bash cài đặt.

Đăng nhập vào máy tính Linux của bạn và mở terminal.

Bây giờ, hãy vào thư mục /temp và tải xuống bash mà bạn đã tải xuống từ trang chủ của Anaconda bằng curl.

```
$ cd/tmp

$ curl -o https://repo.anaconda.com.archive/
Anaconda3-5.2.0-Linux-x86_64.sh
```

3. Bạn cũng nên sử dụng xác minh băm mật mã thông qua tổng kiểm tra SHA-256 để xác minh tính toàn vẹn của trình cài đặt.

```
$ sha256sum Anaconda3-5.2.0-Linux-x86_64.sh
```

Bạn sẽ nhận được kết quả sau.

```
09f53738b0cd3bb96f5b1bac488e5528df9906be2480fe61df40e0e0d19e3d48
Anaconda3-5.2.0-Linux-x86_64.sh
```

4. Bước thứ tư là chạy Anaconda Script, như thể hiện trong hình sau.

```
$ bash Anaconda3-5.2.0-Linux-x86_64.sh
```

Dòng lệnh sẽ tạo ra kết quả sau. Bạn sẽ được yêu cầu xem lại thỏa thuận cấp phép. Tiếp tục nhấn Enter cho đến khi bạn đến cuối.

```
Đầu ra

Chào mừng đến với Anaconda3 5.2.0

Để tiếp tục quá trình cài đặt, vui lòng xem lại thỏa thuận cấp phép.

Vui lòng nhấn Enter để tiếp tục
>>>
...
Bạn có chấp thuận các điều khoản cấp phép không? [Có|Không]
```

Nhập Có khi bạn xem đến cuối Thỏa thuận cấp phép.

5. Trình cài đặt sẽ yêu cầu bạn chọn cài đặt vị trí sau khi bạn đồng ý với thỏa thuận cấp phép.

## Cơ bản về học sâu dành cho người mới bắt đầu |

Chỉ cần nhấn Enter để chọn vị trí mặc định. Bạn cũng có thể chỉ định một vị trí khác nếu muốn.

Đầu ra

Anaconda3 sẽ được cài đặt tại vị trí này:

/trang chủ/tola/anaconda3

- Nhấn ENTER để xác nhận vị trí
- Nhấn CTRL-C để hủy cài đặt
- Hoặc chỉ định một vị trí khác bên dưới

[/home/tola/anaconda3] >>>

Quá trình cài đặt sẽ tiếp tục khi bạn nhấn Enter. Một lần nữa, bạn phải kiên nhẫn vì quá trình cài đặt mất một thời gian để hoàn tất.

6. Bạn sẽ nhận được kết quả sau khi cài đặt

đã hoàn tất. Nếu bạn muốn sử dụng lệnh conda, hãy nhập Đúng.

Đầu ra

...

Quá trình cài đặt đã hoàn tất.

Bạn có muốn trình cài đặt thêm vị trí cài đặt Anaconda3 vào đường dẫn trong /home/tola/.bashrc của bạn không? [có|không]

[không]>>>

Tại thời điểm này, bạn cũng sẽ có tùy chọn tải xuống

Visual Studio Code. Nhập có hoặc không để cài đặt hoặc từ chối,

tương ứng.

7. Sử dụng lệnh sau để kích hoạt sản phẩm mới của bạn

cài đặt Anaconda3.

\$ nguồn ~/.bashrc

8. Bạn cũng có thể kiểm tra cài đặt bằng cách sử dụng conda yêu cầu.

\$ danh sách conda

Xin chúc mừng! Bạn đã cài đặt thành công Anaconda trên hệ thống Linux của mình.

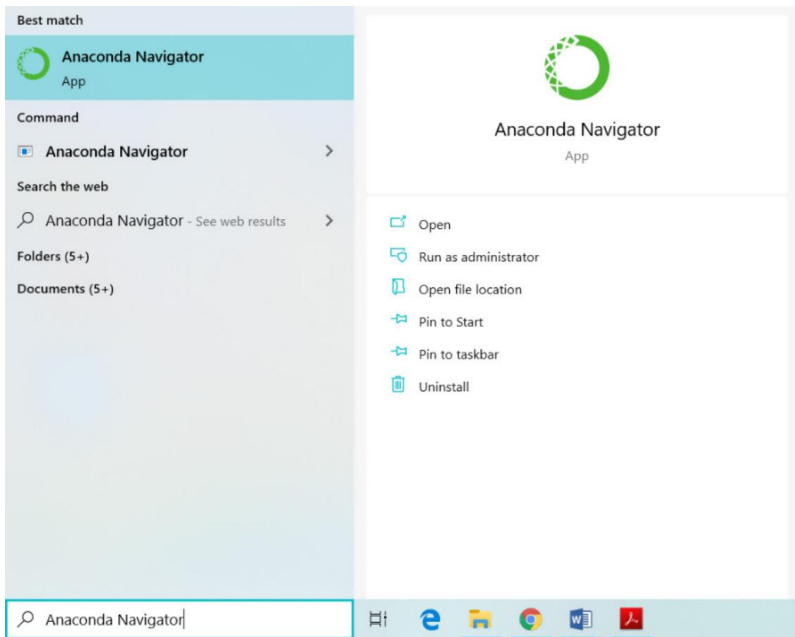
## 1.3 Khóa học Python Crash

Nếu bạn quen thuộc với các khái niệm cơ bản của Python ngôn ngữ lập trình, bạn có thể bỏ qua phần này. Đối với những là người mới bắt đầu học Python, phần này cung cấp tóm tắt ngắn gọn về một số khái niệm cơ bản nhất của Python. Python là một ngôn ngữ lập trình rộng lớn và phần này được viết bởi không có nghĩa là thay thế được một cuốn sách Python hoàn chỉnh. Tuy nhiên, nếu bạn muốn xem các hoạt động và lệnh khác nhau như thế nào được thực hiện bằng Python, bạn có thể theo dõi phần còn lại của phần này.

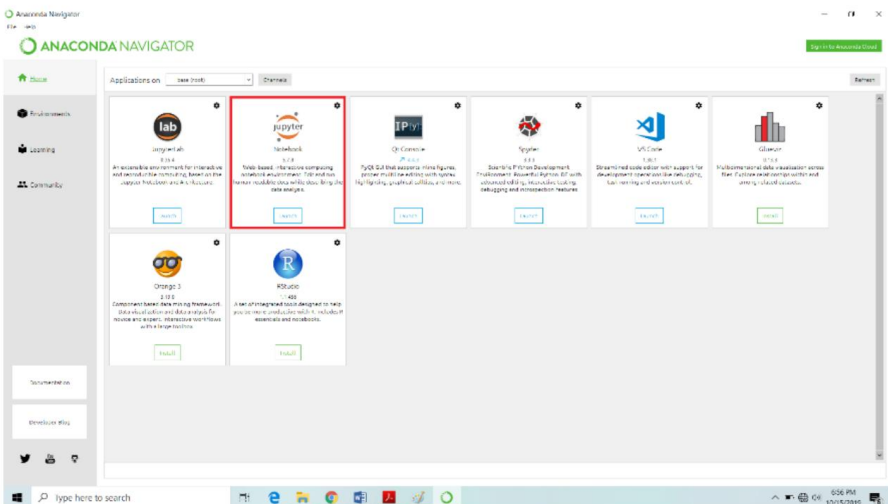
### 1.3.1. Viết chương trình đầu tiên của bạn

Bạn đã cài đặt Python trên máy tính của bạn và đã tạo ra một môi trường độc đáo dưới dạng Anaconda. Bây giờ, đã đến lúc viết chương trình đầu tiên của bạn, đó là Hello Thế giới!

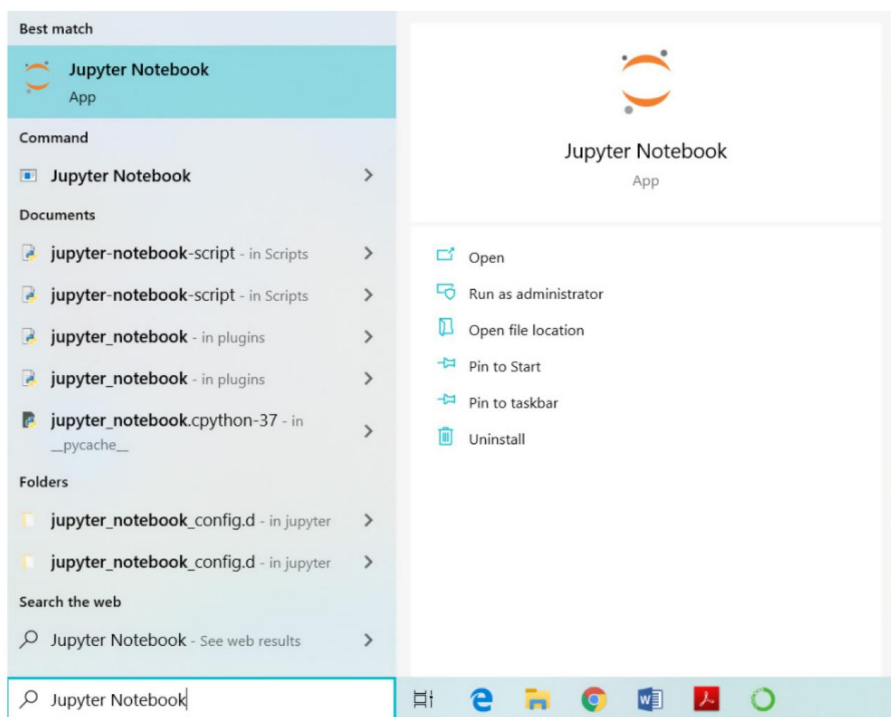
Để viết chương trình trong Anaconda, bạn phải khởi chạy Anaconda Navigator. Tìm kiếm Anaconda Navigator trong Hộp tìm kiếm Windows của bạn. Bây giờ, hãy nhấp vào biểu tượng ứng dụng Anaconda Navigator, như thể hiện trong hình sau.



Khi bạn nhấp vào ứng dụng, bảng điều khiển Anaconda sẽ mở ra. Bảng điều khiển cung cấp cho bạn vô số công cụ để viết mã. Chúng tôi sẽ sử dụng Jupyter Notebook, công cụ phổ biến nhất trong số các công cụ này, để viết và giải thích mã trong suốt bài viết này sách.



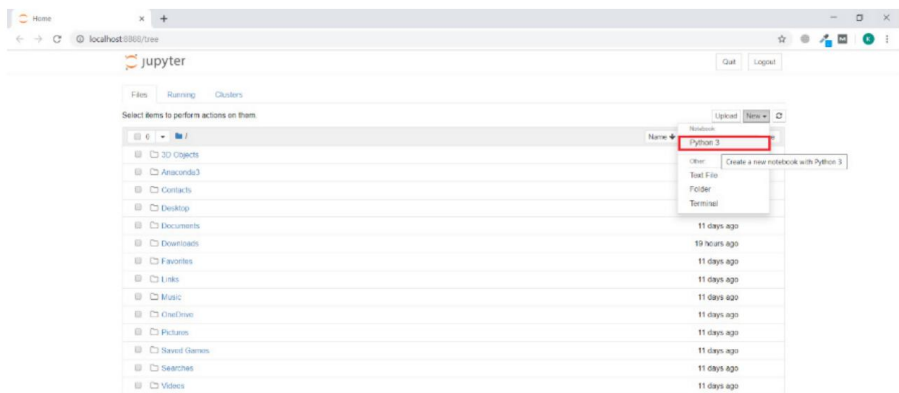
Jupyter Notebook có sẵn ở vị trí thứ hai từ trên cùng của bảng điều khiển. Bạn có thể sử dụng Jupyter Notebook ngay cả khi bạn không có quyền truy cập internet vì nó chạy ngay trong trình duyệt mặc định của bạn. Một phương pháp khác để mở Jupyter Notebook là nhập Jupyter Notebook vào thanh tìm kiếm của Windows. Sau đó, nhấp vào ứng dụng Jupyter Notebook. Ứng dụng sẽ mở trong một tab mới trên trình duyệt của bạn.



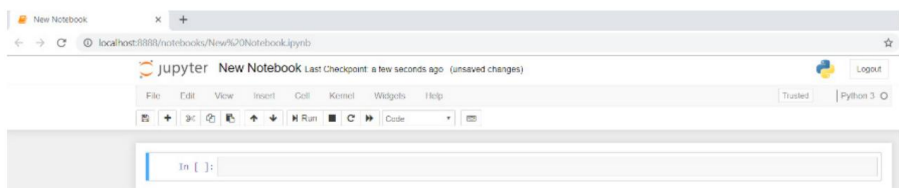
Góc trên bên phải của bảng điều khiển Jupyter Notebook có nút New , bạn phải nhấp vào nút này để mở một tài liệu mới. Một danh sách thả xuống chứa một số tùy chọn sẽ xuất hiện.

Nhấp vào Python 3.

## Cơ bản về học sâu dành cho người mới bắt đầu |



Một sổ tay Python mới sẽ xuất hiện để bạn viết chương trình. Nó trông như sau.



Jupyter Notebook bao gồm các ô, như thể hiện rõ trong hình ảnh trên, làm cho bố cục của nó đơn giản và dễ hiểu. Bạn sẽ viết mã của mình bên trong các ô này. Hãy cùng viết chương trình Python đầu tiên của chúng ta trong Jupyter Notebook.

### 1.3.1. Writing Your First Program

```
print("Welcome to Data Preprocessing with Python")
```

Welcome to Data Preprocessing with Python

Về cơ bản, đoạn mã trên in một giá trị chuỗi trong đầu ra bằng phương thức `print()`. Phương thức `print()` được sử dụng để in

trên bảng điều khiển bất kỳ chuỗi nào được truyền vào. Nếu bạn thấy kết quả sau, bạn đã chạy thành công chương trình Python đầu tiên của mình.

Đầu ra:

```
Chào mừng đến với Trực quan hóa dữ liệu với Python
```

Bây giờ chúng ta hãy cùng khám phá một số khái niệm quan trọng khác của Python bắt đầu với Biến và Kiểu dữ liệu.

#### Yêu cầu – Anaconda, Jupyter và Matplotlib

- Tất cả các tập lệnh trong cuốn sách này đều được thực hiện thông qua sổ tay Jupyter. Do đó, bạn nên cài đặt sổ tay Jupyter.
- Các thư viện Numpy, Pandas và Matplotlib cũng cần được cài đặt trước chương này.

#### Thời gian thực hành – Mã nguồn

Tất cả sổ tay IPython cho mã nguồn của tất cả các tập lệnh trong chương này có thể được tìm thấy trong Mã/Chương 1. tệp ipynb từ tài nguyên sách Tôi đề nghị bạn tự viết toàn bộ mã trong chương này và xem liệu bạn có nhận được kết quả đầu ra giống như đã đề cập trong chương này hay không.

### 1.3.2. Biến và Kiểu dữ liệu Python

Các kiểu dữ liệu trong ngôn ngữ lập trình đề cập đến loại dữ liệu mà ngôn ngữ có khả năng xử lý. Sau đây là những kiểu dữ liệu chính được Python hỗ trợ.

- a. Dây đàn
- b. Số nguyên
- c. Số dấu phẩy động
- d. Boolean
- e. Danh sách



f. Bộ

g. Từ điển

Một biến là một bí danh cho địa chỉ bộ nhớ nơi thực tế dữ liệu được lưu trữ. Dữ liệu hoặc các giá trị được lưu trữ tại bộ nhớ địa chỉ có thể được truy cập và cập nhật thông qua tên biến.

Không giống như các ngôn ngữ lập trình khác như C++, Java và C#, Python được gõ một cách lỏng lẻo, điều đó có nghĩa là bạn không cần phải nêu kiểu dữ liệu trong khi tạo một biến. Thay vào đó, kiểu dữ liệu được đánh giá khi chạy.

Ví dụ sau đây minh họa cách tạo các kiểu dữ liệu khác nhau và cách lưu trữ chúng trong các biến tương ứng. Tập lệnh cũng in kiểu của các biến thông qua hàm `type()`.

Kịch bản 1:

```
# Một biến chuỗi
first_name = "Joseph"
in(kiểu(first_name))

# Một biến số nguyên
tuổi = 20
in(loại(tuổi))

# Một biến dấu phẩy động
trọng lượng = 70,35
in(kiểu(trọng lượng))

# Một biến boolean
đã kết hôn = Sai
in(loại(đã kết hôn))

# Danh sách
xe hơi = ["Honda", "Toyota", "Suzuki"]
in(loại(xe hơi))
```

```
#Tuples
ngày = ("Chủ Nhật", "Thứ Hai", "Thứ Ba", "Thứ Tư",
        "Thứ năm", "Thứ sáu", "Thứ bảy") in(loại(ngày))

#Tử điển
days2 = {1:"Chủ Nhật", 2:"Thứ Hai", 3:"Thứ Ba", 4:"Thứ Tư",
          5:"Thứ năm", 6:"Thứ sáu", 7:"Thứ bảy"} print(type(days2))
```

Đầu ra:

```
<lớp 'str'>
<lớp 'int'>
<lớp 'float'>
<lớp 'bool'>
<lớp 'danh sách'>
<lớp 'tuple'> <lớp
'dict'>
```

### 1.3.3. Toán tử Python

Ngôn ngữ lập trình Python chứa các loại toán tử sau:

- a. Toán tử số học
- b. Các toán tử logic
- c. Các toán tử so sánh
- d. Toán tử gán
- e. Người điều hành thành viên

Chúng ta hãy cùng xem xét sơ lược từng loại toán tử này.

Toán tử số học

Toán tử số học được sử dụng để thực hiện các phép toán số học trong Python. Bảng sau đây tóm tắt các toán tử số học được Python hỗ trợ. Giả sử  $X = 20$  và  $Y = 10$ .

Người điều hành Tên	Biểu tượng	Chức năng	Ví dụ
Phép cộng	+	Thêm các toán hạng ở cả hai bên	$X + Y = 30$
Phép trừ	-	Trừ các toán hạng ở cả hai bên	$X - Y = 10$
Phép nhân	*	Nhân các toán hạng ở cả hai bên	$X * Y = 200$
Phân công	/	Chia toán hạng bên trái cho toán hạng bên phải	$X/Y = 2.0$
Mô đun	%	Chia toán hạng bên trái cho toán hạng bên phải và trả về phần dư	$X \% Y = 0$
Số mũ	**	Lấy số mũ của toán hạng bên trái theo lũy thừa bên phải	$** Y = X^{1024 \times e10}$

Sau đây là một ví dụ về toán tử số học có đầu ra:

Kịch bản 2:

```
X = 20
Y = 10
in(X + Y)
in(X - Y)
in(X * Y)
in(X / Y)
in(X ** Y)
```

Đầu ra:

```
30
10
200
2.0
10240000000000
```

Toán tử logic

Các toán tử logic được sử dụng để thực hiện các phép toán logic AND, OR và NOT trong Python. Bảng sau tóm tắt các toán tử logic. Ở đây, X là True và Y là False.

Biểu tượng toán tử		Chức năng	Ví dụ
Hợp lý VÀ	và	Điều kiện trở thành đúng nếu cả hai toán hạng đều đúng.	(X và Y) = SAI
Hợp lý HOẶC	hoặc	Điều kiện trở thành đúng nếu bất kỳ toán hạng nào trong hai toán hạng là đúng.	(X hoặc Y) = ĐÚNG VẬY
Hợp lý KHÔNG	không	Được sử dụng để đảo ngược trạng thái logic của toán hạng.	không(X và Y) =Đúng

Sau đây là một ví dụ giải thích cách sử dụng toán tử logic của Python.

Kịch bản 3:

```
X = Đúng
Y = Sai
in(X và Y) in(X
hoặc Y)
in(không phải(X và Y))
```

Đầu ra:

```
SAI
ĐÚNG VẬY
ĐÚNG VẬY
```

Toán tử so sánh

Toán tử so sánh, như tên gọi của nó, được sử dụng để so sánh hai hoặc nhiều hơn hai toán hạng. Tùy thuộc vào mối quan hệ giữa các toán hạng, toán tử so sánh trả về các giá trị Boolean. Bảng sau tóm tắt các toán tử so sánh trong Python. Ở đây, X là 20 và Y là 35.

Biểu tượng toán tử		Sự miêu tả	Ví dụ
Sự bình đẳng	==	Nếu giá trị của cả hai toán hạng bằng nhau thì điều kiện trả về giá trị true.	(X == Y) = sai
Bất bình đẳng	!=	Nếu giá trị của cả hai toán hạng không bằng nhau thì điều kiện trả về ĐÚNG VẬY.	(X = Y) = đúng
Lớn hơn	>	Nếu giá trị của toán hạng bên trái lớn hơn toán hạng bên phải thì điều kiện trả về giá trị đúng.	(X> Y) = SAI
Nhỏ hơn	<	Trả về true nếu giá trị của toán hạng bên trái nhỏ hơn toán hạng bên phải.	(X<Y) = ĐÚNG VẬY
Lớn hơn hoặc bằng	>=	Nếu giá trị của toán hạng bên trái lớn hơn hoặc bằng toán hạng bên phải thì điều kiện trả về giá trị đúng.	(X > =Y) = Sai
Nhỏ hơn hoặc bằng	<=	Trả về true nếu giá trị của toán hạng bên trái nhỏ hơn hoặc bằng toán hạng bên phải	(X<= Y) = Đúng

Các toán tử so sánh đã được chứng minh hoạt động trong ví dụ sau:

Kịch bản 4:

```
X = 20
Y = 35

in(X == Y)
in(X != Y)
in(X > Y)
in(X < Y)
in(X >= Y)
in(X <= Y)
```

Đầu ra:

```
SAI
ĐÚNG VẬY
SAI
ĐÚNG VẬY
SAI
ĐÚNG VẬY
```

Toán tử gán

Toán tử gán được sử dụng để gán giá trị cho các biến.

Bảng sau tóm tắt các toán tử gán.

Ở đây, X là 20 và Y bằng 10.

Biểu tượng toán tử	Sự miêu tả	Ví dụ
Phân công	=	Được sử dụng để gán giá trị của toán hạng bên phải cho bên phải.  R = X+ Y gán 30 cho R
Thêm và chỉ định	+= Thêm	các toán hạng ở cả hai bên và gán kết quả cho toán hạng bên trái  X += Y gán 30 cho X
Trừ và gán	=	Trừ các toán hạng ở cả hai bên và gán kết quả cho toán hạng bên trái  X -= Y gán 10 cho X

Nhân và Giao phó	<code>*</code>	Nhân các toán hạng ở cả hai bên và gán kết quả cho toán hạng bên trái	<code>X *= Y</code> gán 200 cho X
Chia và Giao phó	<code>/</code>	Chia các toán hạng bên trái cho toán hạng bên phải và gán kết quả cho toán hạng bên trái	<code>X /= Y</code> gán 2 cho X
Lấy môđun và gán	<code>%</code>	Chia các toán hạng bên trái cho toán hạng bên phải và gán phần còn lại cho toán hạng bên trái	<code>X %= Y</code> gán 0 cho X
Lấy số mũ và gán	<code>**</code>	Lấy số mũ của toán hạng bên trái theo lũy thừa phải và gán phần dư cho toán hạng bên trái	<code>X **= Y</code> gán 1024 x e10 đến X

Hãy xem tập lệnh 6 để xem các toán tử gán Python trong hoạt động.

Kịch bản 5:

```
X = 20; Y = 10
R = X + Y
in(R)

X = 20;
Y = 10
X += Y
in(X)

X = 20;
Y = 10
X -= Y
in(X)
```

```
X = 20;
```

```
Y = 10
```

```
X *= Y
```

```
in(X)
```

```
X = 20;
```

```
Y = 10
```

```
X /=
```

```
Y in(X)
```

```
X = 20;
```

```
Y = 10
```

```
X% = Y
```

```
in(X)
```

```
X = 20;
```

```
Y = 10
```

```
X **= Y
```

```
in(X)
```

Đầu ra:

```
30
```

```
30
```

```
10
```

```
200
```

```
2.0
```

```
0
```

```
10240000000000
```

Nhà điều hành thành viên

Toán tử thành viên được sử dụng để tìm xem một mục có phải là thành viên của một tập hợp các mục hay không. Có hai loại toán tử thành viên. Chúng là toán tử in và toán tử not in.

Đoạn mã sau đây cho thấy toán tử in đang hoạt động.



Kịch bản 6:

```
ngày = ("Chủ Nhật", "Thứ Hai", "Thứ Ba", "Thứ Tư",  
        "Thứ năm", "Thứ sáu", "Thứ bảy")  
print('Chủ Nhật' tính theo ngày)
```

Đầu ra:

ĐÚNG VẬY

Và đây là một ví dụ về toán tử not in.

Kịch bản 7:

```
ngày = ("Chủ Nhật", "Thứ Hai", "Thứ Ba", "Thứ Tư",  
        "Thứ năm", "Thứ sáu", "Thứ bảy")  
print('Xunday' không tính bằng ngày)
```

Đầu ra:

ĐÚNG VẬY

#### 1.3.4. Câu lệnh có điều kiện

Các câu lệnh có điều kiện được sử dụng để triển khai logic có điều kiện trong Python. Các câu lệnh có điều kiện giúp bạn quyết định có nên thực thi một khối mã nào đó hay không. Có

Ba loại câu lệnh điều kiện chính trong Python:

- a. Câu lệnh If
- b. Câu lệnh If-else
- c. Câu lệnh If-elif

Câu lệnh IF

Nếu bạn phải kiểm tra một tình trạng duy nhất và bạn không quan tâm đến điều kiện thay thế, bạn có thể sử dụng if câu lệnh. Ví dụ, nếu bạn muốn kiểm tra xem 10 có lớn hơn không hơn 5, và dựa trên đó bạn muốn in một báo cáo, bạn có thể sử dụng câu lệnh if. Điều kiện được đánh giá bởi if

câu lệnh trả về giá trị Boolean. Nếu điều kiện được đánh giá

bởi câu lệnh if là đúng, khối mã theo sau câu lệnh if

câu lệnh thực thi. Điều quan trọng cần đề cập là trong Python,

khối mã mới bắt đầu ở một dòng mới với tab thụt lề

từ bên trái khi so sánh với khối bên ngoài.

Ở đây, trong ví dụ sau, điều kiện  $10 > 5$  được đánh giá, trả về giá trị true. Do đó,

khối mã theo sau if

câu lệnh được thực thi và một thông báo được in ra trên bảng điều khiển.

Kịch bản 8:

```
# Câu lệnh if

nếu 10 > 5:
    print("Mười lớn hơn n 10")
```

Đầu ra:

```
Mười lớn hơn n 10
```

Câu lệnh IF-Else

Câu lệnh If-else hữu ích khi bạn muốn thực thi một đoạn mã thay thế trong trường hợp điều kiện cho câu lệnh if trả về false. Ví dụ, trong ví dụ sau, điều kiện  $5 < 10$  sẽ trả về false. Do đó, đoạn mã

khối theo sau câu lệnh else sẽ được thực thi.

Kịch bản 9:

```
# câu lệnh if-else

nếu 5 > 10:
    print("5 lớn hơn n 10")
khác:
    print("10 lớn hơn n 5")
```