

**BỘ GIÁO DỤC VÀ ĐÀO TẠO  
TRƯỜNG ĐẠI HỌC QUẢN LÝ VÀ CÔNG NGHỆ HẢI PHÒNG**



ISO 9001:2015

# **ĐỒ ÁN TỐT NGHIỆP**

**NGÀNH: CÔNG NGHỆ THÔNG TIN**

**Sinh viên : Lê Gia Tiên**

**Giảng viên hướng dẫn: Ths. Nguyễn Thị Xuân Hương**

**HẢI PHÒNG – 2020**

**BỘ GIÁO DỤC VÀ ĐÀO TẠO**  
**TRƯỜNG ĐẠI HỌC QUẢN LÝ VÀ CÔNG NGHỆ HẢI PHÒNG**

---

**TÌM HIỂU NGÔN NGỮ LẬP TRÌNH PYTHON**  
**VIẾT CHƯƠNG TRÌNH THỬ NGHIỆM**  
**DỰ BÁO SỰ BIẾN ĐỘNG CỦA GIÁ CHỨNG KHOÁN**

**ĐỒ ÁN TỐT NGHIỆP ĐẠI HỌC HỆ CHÍNH QUY**  
**NGÀNH: CÔNG NGHỆ THÔNG TIN**

**Sinh viên : LÊ GIA TIẾN**

**Giảng viên hướng dẫn : Ths. NGUYỄN THỊ XUÂN HƯƠNG**

**HẢI PHÒNG – 2020**

BỘ GIÁO DỤC VÀ ĐÀO TẠO  
TRƯỜNG ĐẠI HỌC QUẢN LÝ VÀ CÔNG NGHỆ HẢI PHÒNG

-----

## **NHIỆM VỤ ĐỀ TÀI TỐT NGHIỆP**

**Sinh viên:** LÊ GIA TIẾN

**Mã SV:** 1512111031

**Lớp** : CT1901C

**Ngành** : CÔNG NGHỆ THÔNG TIN

**Tên đề tài:** Tìm hiểu ngôn ngữ lập trình python, viết chương trình thử nghiệm dự báo sự biến động của giá chứng khoán.

# NHIỆM VỤ ĐỀ TÀI

## **1. Nội dung và các yêu cầu cần giải quyết trong nhiệm vụ đề tài tốt nghiệp**

- Tìm hiểu về ngôn ngữ lập trình python.
- Tìm hiểu về bài toán dự báo trong python.
- Viết chương trình thử nghiệm dự báo sự biến động của giá chứng khoán bằng ngôn ngữ python.

## **2. Các tài liệu, số liệu cần thiết**

- Số liệu: Số liệu về giá cổ phiếu của các công ty niêm yết trên thị trường chứng khoán Việt Nam.

## **3. Địa điểm thực tập tốt nghiệp**

- Trường Đại Quản lý và Công nghệ Hải Phòng

## CÁN BỘ HƯỚNG DẪN ĐỀ TÀI TỐT NGHIỆP

**Họ và tên** : Nguyễn Thị Xuân Hương

**Học hàm, học vị** : Thạc sỹ

**Cơ quan công tác** : Trường Đại học Quản lý và Công nghệ Hải Phòng

**Nội dung hướng dẫn:**

- Tìm hiểu về ngôn ngữ lập trình python.
- Tìm hiểu về bài toán dự báo.
- Tìm hiểu về bài toán dự báo giá chứng khoán
- Viết chương trình dự báo sự biến động của giá chứng khoán bằng ngôn ngữ python.

Đề tài tốt nghiệp được giao ngày 03 tháng 08 năm 2020

Yêu cầu phải hoàn thành xong trước ngày 17 tháng 10 năm 2020

Đã nhận nhiệm vụ ĐTTN

*Sinh viên*

Đã giao nhiệm vụ ĐTTN

*Giảng viên hướng dẫn*

*Hải Phòng, ngày tháng năm 2020*

**TRƯỞNG KHOA**

**CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM**  
**Độc lập - Tự do - Hạnh phúc**

**PHIẾU NHẬN XÉT CỦA GIÁNG VIÊN HƯỚNG DẪN TỐT NGHIỆP**

Họ và tên giảng viên: Nguyễn Thị Xuân Hương

Đơn vị công tác: Khoa Công nghệ Thông tin – Trường Đại học quản lý và Công nghệ Hải Phòng.

Họ và tên sinh viên: ..... Ngành: .....

Nội dung hướng dẫn:

- Tìm hiểu về ngôn ngữ lập trình python.
- Tìm hiểu về bài toán dự báo.
- Tìm hiểu về bài toán dự báo sự biến động của giá chứng khoán
- Viết chương trình dự báo sự biến động của giá chứng khoán bằng ngôn ngữ python.

**Tinh thần thái độ của sinh viên trong quá trình làm đề tài tốt nghiệp**

- Nghiên túc, chịu khó và có tinh thần học hỏi, tự nghiên cứu.
- Hoàn thành các yêu cầu do giáo viên hướng dẫn đề ra.

**1. Đánh giá chất lượng của đề án/khóa luận (so với nội dung yêu cầu đã đề ra trong nhiệm vụ Đ. T. T. N trên các mặt lý luận, thực tiễn, tính toán số liệu...)**

- Về mặt lý thuyết, sinh viên Lê Gia Tiến đã tìm hiểu và trình bày một cách logic để lập trình trên ngôn ngữ Python. Sinh viên cũng đã tìm hiểu về bài toán dự báo và bài toán dự báo sự biến động của giá chứng khoán.
- Về thực nghiệm, sinh viên đã thực hiện viết chương trình thử nghiệm một số mô hình dự báo thống kê trên ngôn ngữ Python.
- Đề án đạt được các mục tiêu đề ra, đề nghị cho sinh viên được bảo vệ trước hội đồng chấm tốt nghiệp ngành Công nghệ Thông tin.

**3. Ý kiến của giảng viên hướng dẫn tốt nghiệp**

Đạt  Không đạt  Điểm:.....

..

*Hải Phòng, ngày... .. tháng .... năm 2020*

**Giảng viên hướng dẫn**

*(Ký và ghi rõ họ tên)*

**CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM**

**Độc lập - Tự do - Hạnh phúc**

---

**PHIẾU NHẬN XÉT CỦA GIẢNG VIÊN CHẤM PHẢN BIỆN**

Họ và tên giảng viên: .....

Đơn vị công tác: .....

Họ và tên sinh viên: ..... Ngành: .....

Đề tài tốt nghiệp: Tìm hiểu ngôn ngữ lập trình python, viết chương trình thử nghiệm dự báo sự biến động của giá chứng khoán.

**1. Phân nhận xét của giảng viên chấm phản biện**

.....

.....

..

.....

.....

..

.....

.....

..

.....

.....

..

.....

.....

..

.....

.....

..

.....  
.....  
..

## 2. Những mặt còn hạn chế

.....  
.....  
..

.....  
.....  
..

.....  
.....  
..

.....  
.....  
..

.....  
.....  
..

.....  
.....  
..

.....  
.....  
..

.....  
.....  
..

## 3. Ý kiến của giảng viên chấm phản biện

Được bảo vệ  Không được bảo vệ  Điểm.....

Hải Phòng, ngày ..... tháng .... năm 2020

**Giảng viên chấm phản biện**

*(Ký và ghi rõ họ tên)*



## LỜI CẢM ƠN

Lời đầu tiên em xin chân thành cảm ơn các Thầy, Cô trong khoa Công nghệ Thông tin cùng toàn thể các Thầy, Cô trường Đại học Dân lập Hải Phòng nay là trường Đại học Quản lý và Công nghệ Hải Phòng đã giảng dạy, truyền đạt cho em những kiến thức quý báu và tạo điều kiện thuận lợi cho em trong suốt quá trình học tập tại trường cũng như trong thời gian thực hiện đồ án tốt nghiệp.

Đặc biệt, em muốn gửi lời cảm ơn tới Thạc Sĩ Nguyễn Thị Xuân Hương giảng viên trực tiếp hướng dẫn tận tình chỉ bảo giúp em khắc phục những khó khăn, thiếu sót để có thể hoàn thành các phần trong đồ án tốt nghiệp từ tìm hiểu lý thuyết cho tới thực hành sử dụng công cụ.

Xin cảm ơn các bạn bè, người thân đã luôn đồng hành cùng tôi trong suốt thời gian qua và cho tôi chỗ dựa vững chắc để tôi đạt được những kết quả như ngày hôm nay.

Với hiểu biết tìm tòi của bản thân và sự chỉ bảo hướng dẫn tận tình của giảng viên em đã cố gắng hoàn thành đồ án một cách tốt nhất có thể nhưng cũng không thể tránh được thiếu sót. Kính mong nhận được sự đóng góp ý kiến từ thầy cô để em có thể nâng cao cũng như bổ sung thêm kiến thức cho bản thân, hoàn thiện đồ án với một kết quả tốt và hoàn chỉnh hơn.

Em xin chân thành cảm ơn!

Hải Phòng, ngày tháng năm 2020.

Sinh viên thực hiện

Lê Gia Tiến

## MỤC LỤC

MỤC LỤC .....	8
MỞ ĐẦU .....	14
Chương 1 NGÔN NGỮ LẬP TRÌNH PYTHON .....	16
1.1 Giới thiệu .....	16
Tại Sao Nên Học Lập Trình Python?.....	17
1.2 Cài đặt môi trường làm việc “cài đặt PyCharm” .....	17
1.3 Cài đặt python .....	19
1.4 Tạo file và viết mã Python trên PyCharm .....	20
1.5 Các Khái Niệm Cơ Bản Trong Lập Trình Python.....	21
1.5.1 Comments trong python.....	21
1.6 Cấu trúc dữ liệu là gì, các kiểu cấu trúc dữ liệu trong python. ....	22
1.7 Cấu trúc điều khiển trong python .....	23
1.7.1 Lệnh IF .....	23
1.7.2 Lệnh FOR.....	24
1.7.3 Lệnh While.....	26
1.8 Cài đặt các gói thư viện cơ bản trong python .....	27
1.8.1 Cài đặt pip .....	27
1.8.2 một số gói thư viện thông dụng cho Python .....	28
1.9 Sử Dụng Hàm Trong Python .....	29
1.9.1 Hàm (Function) .....	29
1.9.2 Các thông số của Hàm (Function Parameters).....	30
1.9.3 Câu lệnh return trong Python .....	31

1. 9. 4	DocStrings trong Python .....	33
1. 9. 5	Ví dụ về hàm DocStrings trong Python .....	33
1. 10	Đọc file CSV trong python.....	34
1. 10. 1	Tệp CSV là gì ? .....	34
1. 10. 2	Các hàm của module CSV trong Python .....	34
1. 10. 3	Cách đọc tệp CSV .....	35
1. 10. 4	Cách viết tệp CSV .....	36
1. 11	Sử dụng thư viện pandas trong Python.....	37
1. 11. 1	Thư viện pandas python là gì? .....	37
1. 11. 2	Tại sao lại dùng thư viện pandas? .....	37
1. 11. 3	Đọc file csv sử dụng thư viện pandas .....	38
1. 11. 4	Thao tác với dataframe trong pandas .....	39
1. 12	Vẽ biểu đồ trong python .....	42
1. 12. 1	Matplotlib .....	42
1. 12. 2	Khái niệm chung .....	42
1. 12. 3	Ví dụ tạo một biểu đồ đơn giản.....	43
Chương 2	BÀI TOÁN DỰ BÁO .....	44
2. 1	Bài toán dự báo .....	44
2. 1. 1	Dự báo hỗ trợ quá trình ra quyết định trong các tình huống.....	44
2. 1. 2	Chuỗi thời gian (Time Series). .....	45
2. 2	Tiến trình của một bài toán dự báo .....	49
2. 2. 1	Tiêu chuẩn dự báo. ....	50
2. 2. 2	Các đặc tính định dạng:.....	50
Chương 3	BÀI TOÁN DỰ BÁO GIÁ CỔ PHIẾU VÀ THỰC NGHIỆM.....	51

3.1	Giới thiệu bài toán dự báo giá cổ phiếu.....	51
3.2	Phát biểu bài toán.....	53
3.3	Một số mô hình dự báo thống kê .....	54
3.3.1	Mô hình trượt đơn Moving Average (MA) để phân tích dự báo giá chứng khoán. ....	54
3.3.2	Mô hình làm mịn theo cấp số nhân (exponential smoothing). ....	55
3.3.3	Mô hình làm mịn hàm mũ kép “Double exponential smoothing”. ..	56
	Chương trình thực nghiệm .....	57
3.3.4	Kết quả mô hình trượt đơn Moving Average (MA) .....	58
3.3.5	Kết quả thực nghiệm mô hình làm mịn theo cấp số nhân (exponential smoothing). ....	60
3.3.6	Kết quả thực nghiệm mô hình làm mịn hàm mũ kép “Double exponential smoothing”. ....	61
3.3.7	Tổng kết.....	62
	KẾT LUẬN .....	63
	DANH MỤC TÀI LIỆU THAM KHẢO .....	64

## DANH MỤC HÌNH ẢNH TRONG BÁO CÁO

Hình 1. 2-1 dowload PyCharm.....	18
Hình 1. 2-2 Cài đặt PyCharm .....	18
Hình 1. 2-3 Tạo project .....	19
Hình 1. 3-1 Download python.....	19
Hình 1. 3-2 Cài đặt python .....	20
Hình 1. 4-1 Tạo file python.....	20
Hình 1. 4-2 Ví dụ Hello world .....	20
Hình 1. 5-1 Comment trong python .....	21
Hình 1. 6-1 So sánh list và tuple .....	23
Hình 1. 7-1 Ví dụ lệnh if trong bài toán.....	24
Hình 1. 7-2 kết quả ví dụ lệnh if .....	24
Hình 1. 7-3 ví dụ lệnh for.....	25
Hình 1. 7-4 kết quả ví dụ lệnh for .....	25
Hình 1. 7-5 Ví dụ lệnh while.....	26
Hình 1. 7-6 kết quả ví dụ lệnh while.....	26
Hình 1. 8-1 Minh họa cài đặt pip .....	27
Hình 1. 8-2 Ví dụ cài đặt thư viện matplotlib .....	27
Hình 1. 9-1 Ví dụ về hàm.....	30
Hình 1. 9-2 kết quả.....	30
Hình 1. 9-3 Ví dụ hàm Function parameters.....	31
Hình 1. 9-4 kết quả ví dụ hàm Function parameters.....	31
Hình 1. 9-5 lệnh return .....	32

Hình 1. 9-6 ví dụ lệnh return .....	32
Hình 1. 9-7 kết quả ví dụ lệnh return .....	32
Hình 1. 9-8 ví dụ hàm DocString .....	33
Hình 1. 9-9 Kết quả ví dụ hàm DocString .....	34
Hình 1. 10-1 ví dụ cách đọc tệp CSV.....	35
Hình 1. 10-2 kết quả ví dụ đọc tệp CSV .....	36
Hình 1. 10-3 Cách viết tệp CSV.....	36
Hình 1. 10-4 kết quả trong tệp CSV.....	36
Hình 1. 11-1 Đọc file CSV sử dụng pandas.....	38
Hình 1. 11-2 Ví dụ.....	39
Hình 1. 11-3.....	40
Hình 1. 11-4 kết quả theo 1 cột .....	40
Hình 1. 11-5 kết quả theo nhiều cột .....	41
Hình 1. 11-6 Lấy bản ghi theo chỉ số .....	41
Hình 1. 11-7 kết quả numpy arrays trả về.....	42
Hình 1. 11-8 thêm cột từ dữ liệu .....	<b>Error! Bookmark not defined.</b>
Hình 1. 11-9 thêm cột vào dữ liệu có sẵn .....	<b>Error! Bookmark not defined.</b>
Hình 1. 11-10 kết quả data frame.....	<b>Error! Bookmark not defined.</b>
Hình 1. 11-11 kết quả thực hiện.....	<b>Error! Bookmark not defined.</b>
Hình 1. 11-12 kết quả.....	<b>Error! Bookmark not defined.</b>
Hình 1. 11-13 kết quả trên từng cột .....	<b>Error! Bookmark not defined.</b>
Hình 1. 11-14 kết quả tạo mới.....	<b>Error! Bookmark not defined.</b>
Hình 1. 11-15 kết quả từ python list.....	<b>Error! Bookmark not defined.</b>
Hình 1. 11-16 kết quả ví dụ.....	<b>Error! Bookmark not defined.</b>

Hình 1. 11-17 kết quả được lưu.....	<b>Error! Bookmark not defined.</b>
Hình 1. 12-1 ví dụ tạo biểu đồ.....	43
Hình 2. 2-1 Đồ thị của $x_t$ theo $t$ .....	46
Hình 2. 2-3 Đồ thị của $x_t/x_{t-1} * 100$ theo $t$ .....	47
Hình 2. 2-4 Đồ thị của $x_t - x_{t-1}$ theo $t$ .....	47
Hình 2. 2-5 Một số định dạng dữ liệu .....	48
Hình 2. 3-1 Tiến trình chung của một bài toán dự báo .....	49
Hình 3. 3-1 Ví dụ về mô hình làm mịn theo cấp số nhân .....	56
Hình 3. 3-2 Minh họa bảng dữ liệu .....	57
Hình 3. 3-3 Kết quả thực nghiệm mô hình Moving Average kích thước cửa sổ trượt là 60 .....	58
Hình 3. 3-4 Kết quả thực nghiệm mô hình Moving Average kích thước cửa sổ trượt là 30 .....	58
Hình 3. 3-5 Kết quả thực nghiệm mô hình Moving Average kích thước cửa sổ trượt là 15 .....	59
Hình 3. 3-6 Kết quả thực nghiệm mô hình làm mịn theo cấp số nhân .....	60
Hình 3. 3-7 Kết quả thực nghiệm mô hình làm mịn hàm mũ kép .....	61

## MỞ ĐẦU

Trong cuộc sống hiện đại ngày nay, khi nền kinh tế ngày càng phát triển ngành công nghệ thông tin là một trợ thủ đắc lực cho hầu hết các ngành nghề. Sự góp mặt của các chương trình ứng dụng đã mang lại nhiều lợi ích thiết thực và làm cho cách thức vận hành và hoạt động của các liên ngành khác trở lên linh hoạt và mang lại hiệu quả cao. Đặc biệt, trong thời đại công nghệ 4.0 chúng ta gần như không thể thiếu được những khái niệm liên quan đến công nghệ thông tin, từ các phương tiện truyền thông, xã hội, văn hóa, giải trí, khoa học kỹ thuật cho đến lĩnh vực kinh doanh, quản lý, y tế, giáo dục v.v. Đâu đâu cũng cần những phần mềm cũng như sản phẩm của công nghệ thông tin để hỗ trợ giúp thúc đẩy sự phát triển vững mạnh hơn. Ví dụ trong lĩnh vực kinh doanh buôn bán hàng hóa theo cách truyền thống vẫn còn tồn tại nhiều nhược điểm như thống kê chi tiết chưa chính xác, quy trình thanh toán chưa được chặt chẽ và nhanh chóng, nhưng nhờ có các phần mềm quản lý, thanh toán điện tử mà các nhà quản lý có thể theo dõi các kết quả thực hiện chính xác của hệ thống.

Để tạo ra các sản phẩm phần mềm hữu ích, cùng với sự phát triển của nhiều ngôn ngữ lập trình đáp ứng cho công nghiệp 4.0 thì Python là một cái tên đáng chú ý. Hiện nay ngôn ngữ Python được xếp hạng 1 trong Top 10 các ngôn ngữ lập trình phổ biến nhất đang được thế giới sử dụng. Python là một ngôn ngữ có hình thái rất sáng sủa, cấu trúc rõ ràng, thuận tiện cho người mới bắt đầu học lập trình. Cấu trúc của Python còn cho phép người sử dụng viết mã lệnh với số lần gõ phím tối thiểu nhất. Python là một ngôn ngữ lập trình đơn giản nhưng lại rất hiệu quả. Bên cạnh đó, Python là một ngôn ngữ có tính hướng đối tượng cao, và được sử dụng rất nhiều cho các bài toán xử lý với dữ liệu lớn. Với ngôn ngữ lập trình python là một ngôn ngữ lập trình đa năng với nhiều ưu điểm vượt trội, và đang đứng ở vị trí số một trong top các ngôn ngữ lập trình hiện nay. Chính vì những ưu điểm này nên lem đã chọn đề tài tìm hiểu ngôn ngữ lập trình Python và viết chương trình thực nghiệm dự báo sự biến động của giá chứng khoán làm đề tài tốt nghiệp của mình.



Đồ án gồm các nội dung sau:

**Chương I.** Tìm hiểu về ngôn ngữ lập trình python, trong đó trình bày về lịch sử phát triển, cài đặt phần mềm, các cấu trúc điều khiển, các cấu trúc dữ liệu và một số công cụ trong ngôn ngữ lập trình Python.

**Chương II** Tìm hiểu về bài toán dự báo, trong đó trình bày về bài toán dự báo, tiến trình của bài toán dự báo và tiêu chuẩn của bài toán dự báo.

**Chương III** Bài toán ứng dụng dự báo giá cổ phiếu và thực nghiệm, trong đó trình bày về bài toán dự báo giá cổ phiếu, một số phương pháp thống kê để dự báo giá cổ phiếu, chương trình thử nghiệm và kết quả dự báo với dữ liệu chứng khoán của công ty chứng khoán được niêm yết trên thị trường chứng khoán Việt Nam.

Cuối cùng là phần kết luận trong đó trình bày các nội dung và kết quả thực hiện của đề tài, những hạn chế và hướng nghiên cứu tiếp theo.

# Chương 1 NGÔN NGỮ LẬP TRÌNH PYTHON

## 1.1 Giới thiệu

### Lịch sử hình thành

Python đã được hình thành vào cuối những năm 1980 và được bắt đầu thực hiện vào tháng 12/1989 bởi Guido van Rossum tại CWI tại Hà Lan như là người kế thừa của ngôn ngữ ABC (tự lấy cảm hứng từ SETL) có khả năng xử lý ngoại lệ và giao tiếp với hệ điều hành Amoeba. Van Rossum là tác giả chính của Python, và vai trò trung tâm của ông tiếp tục trong việc quyết định hướng phát triển của Python được phản ánh trong tiêu đề mà cộng đồng Python dành cho ông “Độc tài nhân từ cho cuộc sống” (benevolent dictator for life)(BDFL).

Python 2.0 được phát hành vào ngày 16/10/2000, với nhiều tính năng chính mới bao gồm một bộ dọn rác đầy đủ và hỗ trợ Unicode. Với phiên bản này, quá trình phát triển đã được thay đổi và trở thành minh bạch hơn và được cộng đồng ủng hộ.

Python 3.0 (còn được gọi là Python 3000 hoặc Py3k), một bản phát hành lớn, không tương thích ngược, được phát hành vào ngày 03/12/2008 sau một thời gian dài thử nghiệm. Nhiều trong số các tính năng chính của nó đã được điều chỉnh để tương thích ngược với Python 2.6 và 2.7. Các tính năng và triết lý phát triển Python là 1 ngôn ngữ lập trình đa hình: lập trình hướng đối tượng và hướng cấu trúc được hỗ trợ đầy đủ, và có 1 số tính năng của ngôn ngữ hỗ trợ lập trình theo chức năng và lập trình hướng khía cạnh (Aspect-oriented programming). Nhiều mô hình khác được hỗ trợ bằng việc sử dụng các phần mở rộng, bao gồm thiết kế theo hợp đồng (design by contract) và lập trình luận lý.

Các trang như Mozilla, Reddit, Instagram và PBS đều được viết bằng Python.

Ngôn ngữ lập trình Python được dùng vào các mục đích:

- Phát triển web (trên máy chủ)
- Phát triển phần mềm

- Tính toán một cách khoa học
- Lên kịch bản cho hệ thống

### Tại Sao Nên Học Lập Trình Python?

- Python hỗ trợ nhiều nền tảng khác nhau (Windows, Mac, Linux, Raspberry Pi, etc).
- Python có cú pháp đơn giản, dễ đọc hiểu và rất gần gũi với tiếng Anh.
- Cú pháp của Python giúp lập trình viên sử dụng ít dòng code để lập trình cùng một thuật toán hơn so với các ngôn ngữ lập trình khác.
- Python sử dụng trình thông dịch để thực thi các dòng code. Do đó, những dòng code có thể được thực thi ngay lập tức mà không cần biên dịch toàn bộ chương trình. Như vậy giúp chúng ta kiểm tra code nhanh hơn.

Python cũng hỗ trợ hàm, thủ tục, hay kể cả lập trình hướng đối tượng.

Để viết mã nguồn Python, ta có thể sử dụng bất kỳ một trình soạn thảo nào, kể cả những trình soạn thảo đơn giản nhất như NotePad. Tuy nhiên, để phát triển các ứng dụng một cách hiệu quả hơn, ta nên sử dụng một IDE, để có thể tiết kiệm thời gian và công sức viết code.

Trong phần tìm hiểu này, đề án sử dụng một trong những IDE thông dụng nhất để lập trình ứng dụng Python, đó là PyCharm IDE.

### **1.2 Cài đặt môi trường làm việc “cài đặt PyCharm”**

Để download Pycharm, ta truy cập vào:

<https://www.jetbrains.com/pycharm/download/#section=windows> và tải về



Version: 2019.3.1  
Build: 193.5662.61  
19 December 2019

[System requirements](#)

[Installation Instructions](#)

## Download PyCharm

[Windows](#) [Mac](#) [Linux](#)

### Professional

For both Scientific and Web Python development. With HTML, JS, and SQL support.

Download

Free trial

### Community

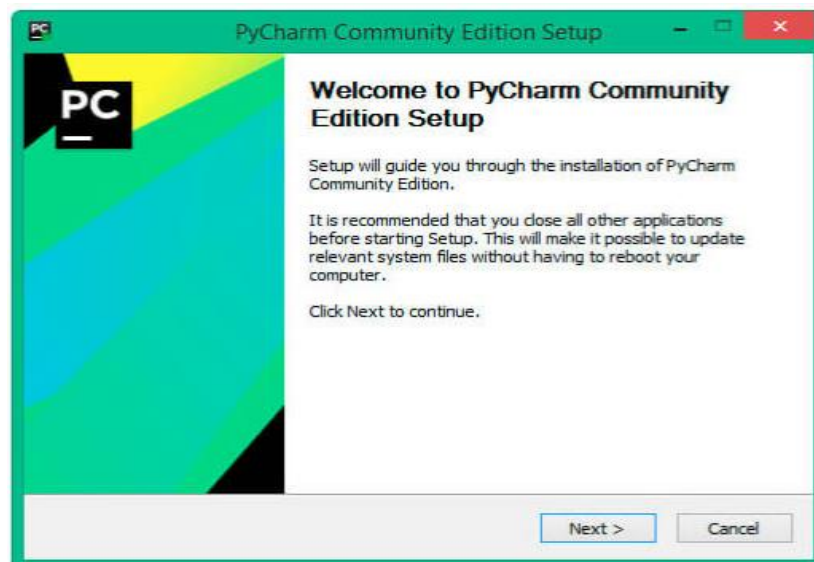
For pure Python development

Download

Free, open-source

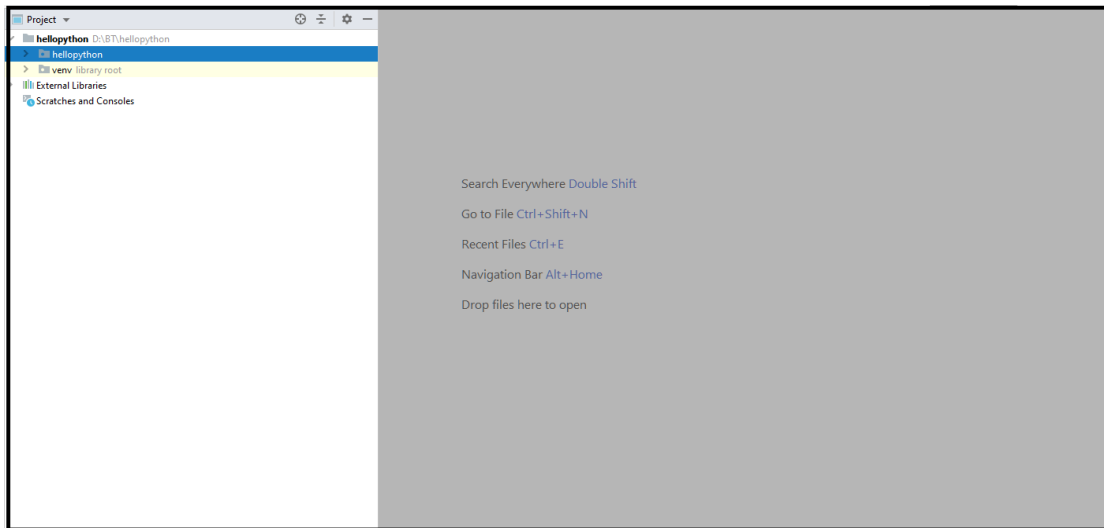
*Hình 1.2-1 download PyCharm*

Sau khi đã tải phiên bản pycahrm về máy ta tiến hành cài đặt pycharm theo các bước trong hướng dẫn và những tùy chọn cài đặt.



*Hình 1.2-2 Cài đặt PyCharm*

Sau khi quá trình cài đặt được hoàn tất, giao diện của PyCharm sẽ như hình

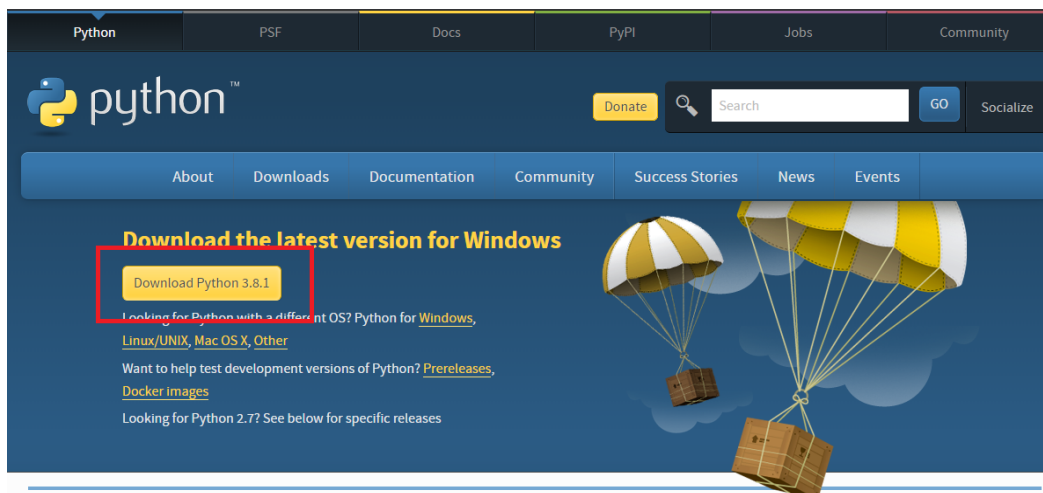


Hình 1.2-3 Tạo project

### 1.3 Cài đặt python

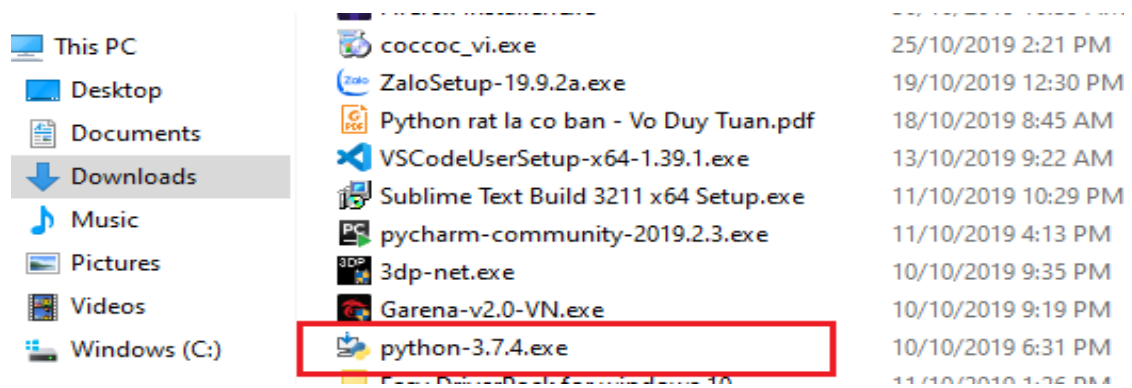
Để download Python, ta truy cập địa chỉ trang chủ của python qua địa chỉ

<https://www.python.org/downloads/> và tải python bản mới nhất.



Hình 1.3-1 Download python

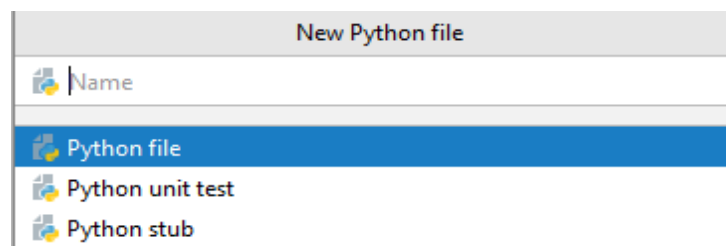
Sau khi download xong nhấn vào file. exe vừa download và kích hoạt để cài đặt. Lúc này chỉ cần thực hiện lần lượt các bước theo hướng dẫn như chọn các thành phần cài đặt, xác định đường dẫn cần thiết và các thông số khác.



Hình 1.3-2 Cài đặt python

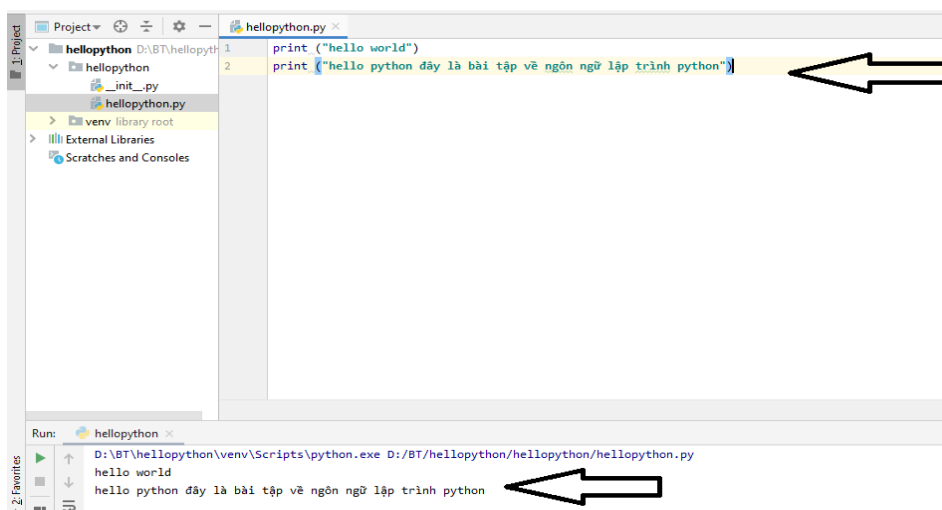
## 1.4 Tạo file và viết mã Python trên PyCharm

Sau khi đã tạo xong Project, ta click phải chuột lên Project, rồi tạo mới một Python File, để tạo một file mã nguồn Python. Và đặt tên cho file đó, file được viết bằng ngôn ngữ lập trình python có đuôi “py”



Hình 1.4-1 Tạo file python

Thử viết một ví dụ in ra “Hello world” trên pycharm bằng ngôn ngữ lập trình python.

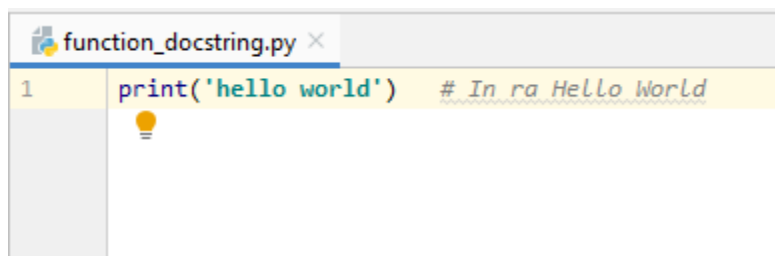


Hình 1.4-2 Ví dụ Hello world

## 1.5 Các Khái Niệm Cơ Bản Trong Lập Trình Python

### 1.5.1 Comments trong python

Trong Python bất kỳ văn bản nào ở bên phải biểu tượng # thì sẽ được trình biên dịch hiểu là một comment và không biên dịch phần đó.



Hình 1.5-1 Comment trong python

Sử dụng càng nhiều comment hữu ích trong chương trình của bạn sẽ làm cho công việc lập trình của bạn dễ dàng hơn:

Giải thích các giả định.

Giải thích các quyết định quan trọng.

Giải thích chi tiết quan trọng.

Giải thích vấn đề bạn đang cố gắng giải quyết.

Giải thích các vấn đề đang cố gắng khắc phục trong chương trình của mình, v.v. .

**Hằng số** (Literal Constants): Ví dụ về một hằng số theo nghĩa đen là một số như 5, 1. 23 hoặc một chuỗi như 'python' hay "It's a string!". Nó được gọi là nghĩa đen bởi vì sử dụng giá trị của nó theo nghĩa đen. Số 2 luôn luôn đại diện cho chính nó và không có gì khác và nó là một hằng số vì giá trị của nó không thể thay đổi. Do đó, tất cả những giá trị này được gọi là hằng số.

**Số** (Numbers): Số chủ yếu có hai loại – số nguyên (integer) và số thực (float).

**Chuỗi** (String): Một chuỗi là một dãy các ký tự. Chuỗi về cơ bản chỉ là một loạt các từ.

**Biến:** Biến chính xác như tên gọi của nó, tức là giá trị của nó có thể thay đổi. Các biến có thể giúp lưu trữ bất cứ cái gì nếu có thể định nghĩa được nó. Các biến chỉ là một phần của bộ nhớ máy tính nơi lưu trữ một số thông tin.

**Đối tượng (Object):** Python đề cập đến bất cứ điều gì được sử dụng trong một chương trình như là một đối tượng.

## 1.6 Cấu trúc dữ liệu là gì, các kiểu cấu trúc dữ liệu trong python.

Việc tổ chức, quản lý và lưu trữ dữ liệu rất quan trọng vì nó cho phép truy cập dễ dàng hơn và sửa đổi hiệu quả. Cấu trúc dữ liệu (Data Structure) cho phép bạn sắp xếp dữ liệu của mình theo cách cho phép bạn lưu trữ các bộ dữ liệu được thu thập, liên quan đến chúng và theo đó mà thực hiện các thao tác trên chúng.

Python có hỗ trợ ngầm cho Cấu trúc dữ liệu cho phép lưu trữ và truy cập dữ liệu. Các cấu trúc này được gọi là List, Dictionary, Tuple và Set.

Python cho phép người dùng tạo Cấu trúc dữ liệu của riêng họ, cho phép toàn quyền kiểm soát chức năng. Các cấu trúc dữ liệu nổi bật nhất là Stack, Queue, Tree, Linked List, v.v. đồng thời cũng có sẵn trong các ngôn ngữ lập trình khác.

### Cấu trúc dữ liệu tích hợp (Built-in Data Structures)

Về cấu trúc dữ liệu trong Python, các Cấu trúc dữ liệu này được tích hợp sẵn với Python giúp lập trình dễ dàng hơn và giúp các lập trình viên sử dụng chúng để có được các giải pháp nhanh hơn. Và có các kiểu cấu trúc dữ liệu là:

**List:** Được sử dụng để lưu trữ dữ liệu của các loại dữ liệu khác nhau một cách tuần tự. Có các địa chỉ được gán cho mọi thành phần của danh sách, được gọi là Index. Giá trị chỉ mục bắt đầu từ 0 và tiếp tục cho đến khi phần tử cuối cùng được gọi là chỉ số dương.

**Dictionary:** Được sử dụng để lưu trữ các cặp key-value. Để hiểu rõ hơn, hãy nghĩ đến một thư mục điện thoại nơi hàng trăm và hàng ngàn tên và số tương ứng của chúng đã được thêm vào. Bây giờ các giá trị không đổi ở đây là Tên và Số điện thoại được gọi là các phím. Và các tên và số điện thoại khác nhau là các giá trị đã



được đưa vào các phím. Nếu truy cập các giá trị của các phím, sẽ nhận được tất cả tên và số điện thoại. Vì vậy, đó là những gì một cặp key-value. Và trong Python, cấu trúc này được lưu trữ bằng Dictionary

**Set:** Là một tập hợp các yếu tố không có thứ tự là duy nhất. Có nghĩa là ngay cả khi dữ liệu được lặp lại nhiều lần, nó sẽ chỉ được nhập vào tập hợp một lần.

**Tuple** (các bộ dữ liệu): Tuples giống như các list với ngoại lệ là dữ liệu một khi được nhập vào bộ dữ liệu không thể thay đổi bất kể điều gì. Ngoại lệ duy nhất là khi dữ liệu bên trong Tuple có thể thay đổi, chỉ sau đó dữ liệu Tuple có thể được thay đổi.

LIST	TUPLE
Được sử dụng cho các loại dữ liệu đồng nhất	Thường được sử dụng cho các loại dữ liệu không đồng nhất
Có thể thay đổi trong môi trường	Bất biến trong môi trường giúp lặp lại nhanh hơn
Không có yếu tố bất biến	Các yếu tố bất biến có thể được sử dụng làm key cho từ điển
Không đảm bảo rằng dữ liệu được bảo vệ chống ghi	Việc thực hiện một bộ dữ liệu không thay đổi đảm bảo rằng nó được bảo vệ chống ghi

Hình 1.6-1 So sánh list và tuple

## 1.7 Cấu trúc điều khiển trong python

Python luôn chạy một loạt các câu lệnh theo thứ tự từ trên xuống một cách chính xác. Câu lệnh điều khiển là loại câu lệnh được dùng để điều khiển luồng chạy của các câu lệnh khác trong chương trình.

Điều này đạt được bằng cách sử dụng các câu lệnh điều khiển. Có 3 lệnh kiểm soát cấu trúc điều khiển của Python là: **if**, **for** và **while**.

### 1.7.1 Lệnh IF

Được sử dụng để kiểm tra một điều kiện: *nếu* điều kiện là đúng sẽ chạy một khối các câu lệnh (được gọi là *if-block*), *nếu sai* chương trình sẽ xử lý một khối các câu lệnh khác (được gọi là *else-block* ).

Ví dụ:

Cho một số nguyên cho trước, nhập vào một số nguyên khác và báo về các kết quả. Nếu thỏa mãn các điều kiện thì in ra các kết quả khác nhau

```
if.py x
1   number = 22
2   guess = int(input('Mời nhập vào số nguyên: '))
3   if guess == number:
4       # Bat dau Block moi o day
5       print('Chính xác!')
6       # Ket thuc Block
7   elif guess < number:
8       # Bat dau Block khac
9       print('Không, số đã nhập bé hơn!')
10      # Ban co the them vao bat cu gi vao day...
11  else:
12      print('Không, số đã nhập lớn hơn')
13
14  print('Kết thúc!')
```

Hình 1.7-1 Ví dụ lệnh if trong bài toán

Kết quả cho thấy khi chúng ta nhập một số nguyên bất kỳ. Nếu thỏa mãn điều kiện bằng với số cho trước thì in ra “chính xác”. Và kết quả khác khi điền vào số lớn hoặc nhỏ hơn số cho trước.

Kết quả:

```
D:\BT\hellopython\venv\Scripts\python.exe C
Nhap vao mot so nguyen: 25
Khong, So da nhap l hon
Ket thuc!

Process finished with exit code 0
```

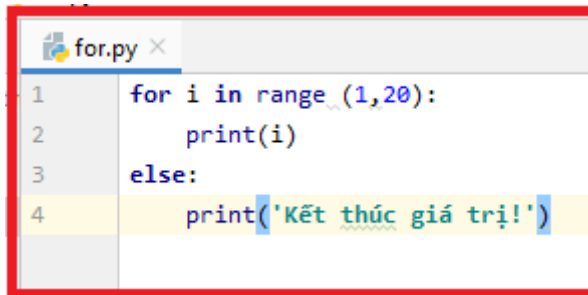
Hình 1.7-2 kết quả ví dụ lệnh if

## 1.7.2 Lệnh FOR

Câu lệnh `for` là một câu lệnh lặp khác, nó lặp đi **lặp lại** qua một chuỗi (sequences) các đối tượng tức là đi qua từng mục trong một chuỗi. Chúng ta sẽ tìm

hiểu nhiều hơn về sequences trong bài về cấu trúc dữ liệu trong Python sau. Những gì bạn cần biết ngay bây giờ là một chuỗi trình từ chỉ là một tập hợp các items.

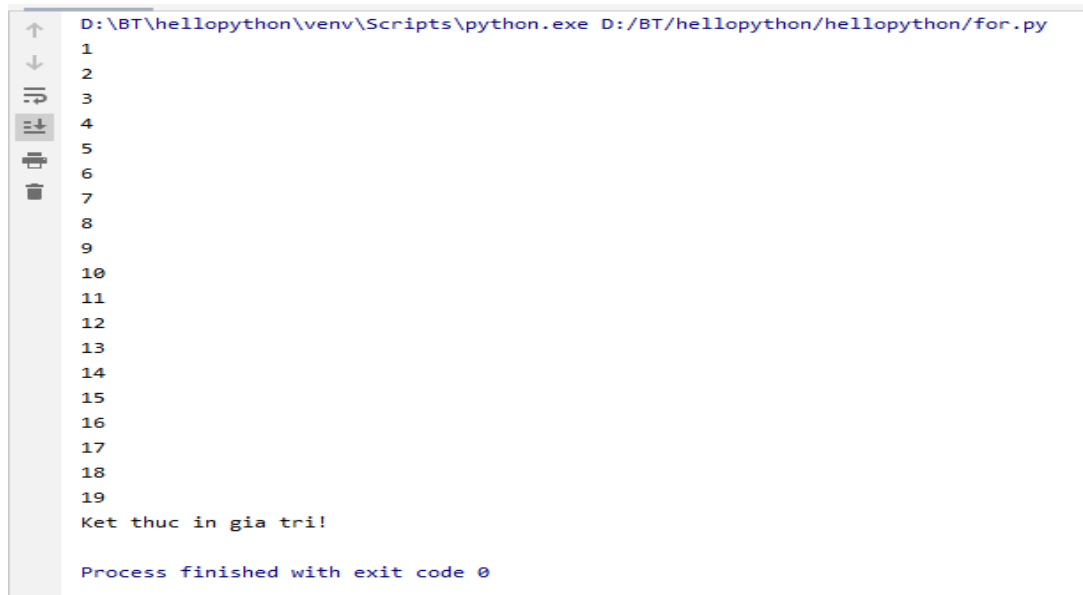
Ví dụ



```
for.py x
1 for i in range(1,20):
2     print(i)
3 else:
4     print('Kết thúc giá trị!')
```

Hình 1.7-3 ví dụ lệnh for

Kết quả Vòng lặp for trong Python hoàn toàn khác với vòng lặp trong C / C ++. vòng lặp for trong Python tương tự như vòng lặp foreach trong C#. Trong C / C ++, nếu bạn muốn viết `for(int i = 0; i <5; i ++)`, thì trong Python bạn chỉ cần viết `for i in range(0, 5)`. Vòng lặp for đơn giản hơn, biểu cảm hơn và ít bị lỗi hơn trong Python.



```
D:\BT\hellopython\venv\Scripts\python.exe D:/BT/hellopython/hellopython/for.py
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
Ket thuc in gia tri!
Process finished with exit code 0
```

Hình 1.7-4 kết quả ví dụ lệnh for

### 1.7.3 Lệnh While

Trong Python: Câu lệnh while cho phép liên tục thực thi một khối các câu lệnh miễn là điều kiện là đúng. Một câu lệnh while là một ví dụ về cái được gọi là câu lệnh lặp. Một câu lệnh while có thể có một mệnh đề khác tùy chọn.

Ví dụ nhập vào một số nguyên sao cho thỏa mãn các điều kiện cho trước, sau khi thỏa mãn một khối các lệnh thì in ra kết quả “Hoàn thành”.

```
while.py x
1   number = 23
2   running = True
3
4   while running:
5       guess = int(input('Nhập vào một số nguyên: '))
6       if guess == number:
7           print('Chính xác ')
8           running = False
9       elif guess < number:
10          print('Số đã nhập nhỏ hơn số đã chọn.')
11      else:
12          print('Số đã nhập lớn hơn số đã chọn')
13
14  print('Hoàn thành')
```

Hình 1.7-5 Ví dụ lệnh while

Kết quả:

```
D:\BT\hellopython\venv\Scripts\python.exe t
Nhập vào một số nguyên: 20
Số đã nhập nhỏ hơn số đã chọn.
Nhập vào một số nguyên: 24
Số đã nhập lớn hơn số đã chọn
Nhập vào một số nguyên: 23
Chính xác
Hoàn thành
```

Hình 1.7-6 kết quả ví dụ lệnh while

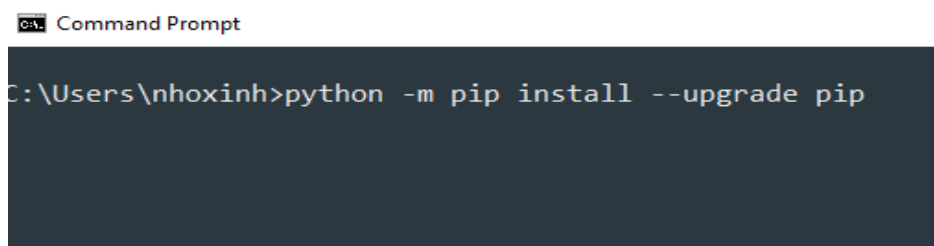
## 1.8 Cài đặt các gói thư viện cơ bản trong python

### 1.8.1 Cài đặt pip

Pip là công cụ quản lý các gói thư viện python (Python package manager). Với việc sử dụng pip, ta sẽ dễ dàng cài đặt các gói thư viện cần thiết trong quá trình làm việc với Python.

Để cập nhật pip hiện thời lên phiên bản mới nhất, ta mở cmd và dung câu lệnh

```
python -m pip install --upgrade pip
```

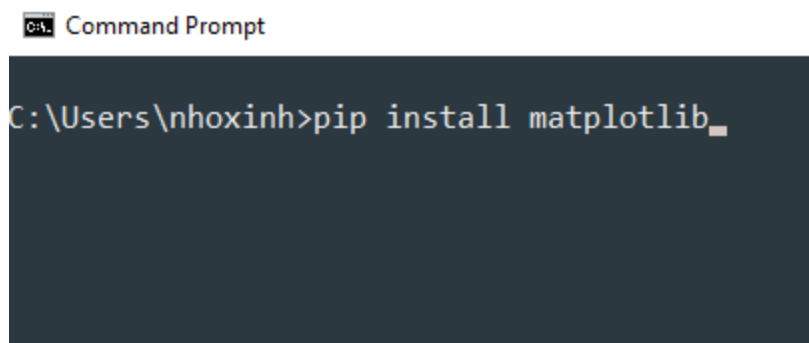


```
Command Prompt
C:\Users\nhoxinh>python -m pip install --upgrade pip
```

*Hình 1.8-1 Minh họa cài đặt pip*

Để cài đặt một thư viện python bất kì ta mở cmd trên windows và gõ pip install “tên thư viện”.

Ví dụ:



```
Command Prompt
C:\Users\nhoxinh>pip install matplotlib_
```

*Hình 1.8-2 Ví dụ cài đặt thư viện matplotlib*

thư viện **Matplotlib** là một thư viện xây dựng các khối cần thiết để tạo các biểu đồ chất lượng từ mảng và trực quan và tương tác với chúng.

## 1.8.2 Một số gói thư viện thông dụng cho Python

### 1.8.2.1 NumPy

Được tạo bởi Travis Oliphant, **NumPy** là một “ngựa kéo” phân tích thực sự của Python. Nó cung cấp cho người dùng cách làm việc với các mảng nhiều chiều, cùng một số lượng lớn các hàm để xử lý trên các toán tử toán học nhiều chiều trên các mảng đó. Mảng là các khối dữ liệu được sắp xếp theo nhiều chiều dựa trên các véc tơ và ma trận trong toán học. Mảng thường hữu ích không chỉ trong việc lưu dữ liệu mà cả việc tính toán nhanh các ma trận, điều không thể thiếu khi giải quyết các vấn đề liên quan đến khoa học dữ liệu.

- Cài đặt: `pip install numpy`
- Trang chủ: <http://www.numpy.org>

### 1.8.2.2 SciPy

Là một dự án gốc bởi Travis Oliphant, Pearu Peterson, and Eric Jones, SciPy hoàn thiện các tính năng của NumPy, nhằm cung cấp các thuật toán cho đại số tuyến tính, không gian ma trận, xử lý tín hiệu và xử lý ảnh, tối ưu, biến đổi Fourier, ...

- Cài đặt: `pip install scipy`
- Trang chủ: <http://www.scipy.org>

### 1.8.2.3 Pandas

**Pandas** là thư viện thực hiện mọi thứ mà **NumPy** và **SciPy** không thể làm. Nó làm việc với các đối tượng cấu trúc dữ liệu, DataFrames và Chuỗi (Series). pandas cho phép bạn có thể xử lý các bảng dữ liệu phức tạp của nhiều loại khác nhau (điều mà các mảng của NumPy thông thường không thể làm được) và chuỗi thời gian. Bạn sẽ dễ dàng tải dữ liệu từ nhiều nguồn khác nhau, sau đó slice, dice, xử lý các thành phần còn thiếu, thêm, đổi tên, tổng hợp (aggregate), reshape và cuối cùng là trực quan dữ liệu theo ý của bạn.

- Cài đặt: `pip install pandas`
- Trang chủ: <http://pandas.pydata.org>

#### 1.8.2.4 Scikit-learn

Bắt đầu như một phần của SciKits, Scikit-learn là lõi hoạt động của khoa học dữ liệu trên Python. Nó cung cấp tất cả những gì bạn cần để tiền xử lý dữ liệu, học giám sát và không giám sát, lựa chọn mô hình, validate và error metrics.

- Cài đặt: `pip install scikit-learn`
- Trang chủ: <http://scikit-learn.org/stable>

#### 1.8.2.5 IPython

Một cách tiếp cận khoa học yêu cầu thử nghiệm nhanh các giả thuyết khác nhau trong một khoảng thời gian. IPython được tạo bởi Fernando Perez để giải quyết việc cần thiết một lệnh Shell Python (dựa trên shell, trình duyệt web, và giao diện ứng dụng) với đồ họa tích hợp, các lệnh có thể tùy chỉnh, lịch sử phong phú (dưới định dạng JSON) và khả năng tính toán song song để cải thiện hiệu năng tính toán.

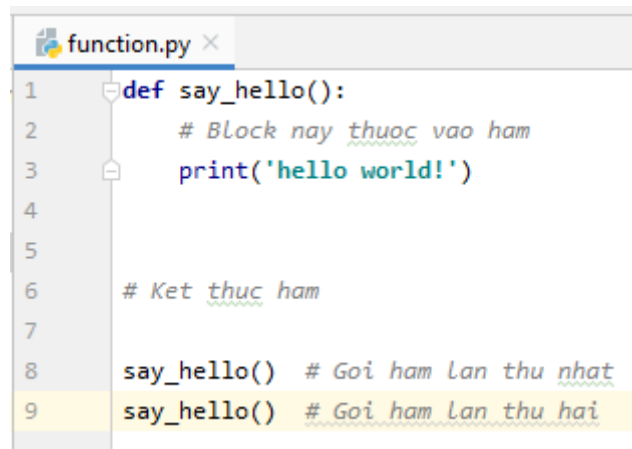
- Cài đặt: `pip install "ipython[ notebook] "`
- Trang chủ: <http://ipython.org>

## 1.9 Sử Dụng Hàm Trong Python

### 1.9.1 Hàm (Function)

Là những phần tái sử dụng của chương trình. Chúng cho phép đặt tên cho một khối các câu lệnh, cho phép chạy khối đó bằng cách sử dụng tên được chỉ định ở bất kỳ đâu trong chương trình với số lần không hạn chế. Điều này được gọi là gọi hàm. Các Hàm được xác định bằng cách sử dụng từ khóa “def”. Sau khi từ khóa này xuất hiện một tên định danh cho hàm, theo sau là một cặp dấu ngoặc đơn có thể kèm theo một số tên của các biến và bởi dấu hai chấm cuối cùng kết thúc dòng. Tiếp theo sau là khối các câu lệnh của Hàm.

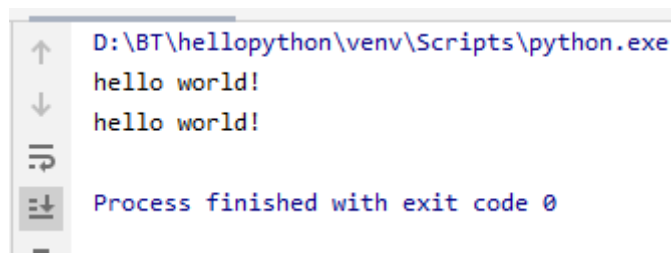
Một ví dụ về hàm đơn giản:



```
function.py x
1 def say_hello():
2     # Block này thuộc vào hàm
3     print('hello world!')
4
5
6     # Kết thúc hàm
7
8 say_hello() # Gọi hàm lần thứ nhất
9 say_hello() # Gọi hàm lần thứ hai
```

Hình 1.9-1 Ví dụ về hàm

Kết quả:



```
D:\BT\hellopython\venv\Scripts\python.exe
hello world!
hello world!
Process finished with exit code 0
```

Hình 1.9-2 kết quả

### 1.9.2 Các thông số của Hàm (Function Parameters)

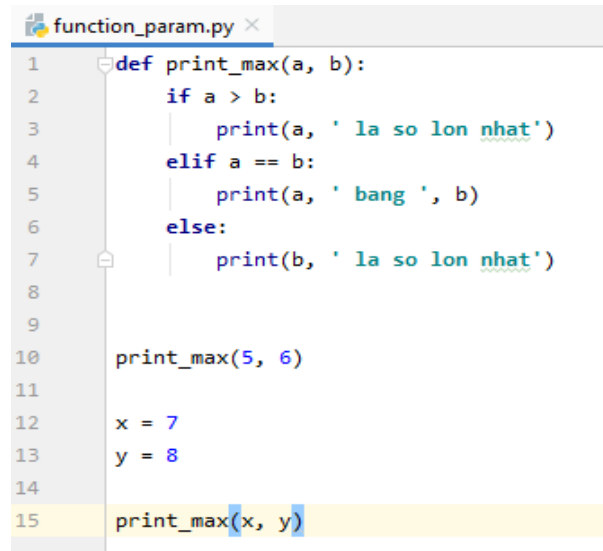
Một hàm có thể lấy tham số, là các giá trị bạn cung cấp cho hàm để hàm có thể sử dụng các giá trị đó cho một mục đích cụ thể. Các tham số này giống như các biến ngoại, trừ các giá trị của các biến này được xác định khi chúng ta gọi hàm và đã được gán các giá trị khi hàm chạy.

Các tham số được chỉ định trong cặp dấu ngoặc trong khai báo hàm, được phân tách bằng dấu phẩy. Khi chúng ta gọi hàm, chúng ta cung cấp các giá trị theo cùng một cách. Lưu ý thuật ngữ được sử dụng – các tên được đưa ra trong định nghĩa



hàm được gọi là **tham số** trong khi các giá trị bạn cung cấp trong lệnh gọi hàm được gọi là **đối số**.

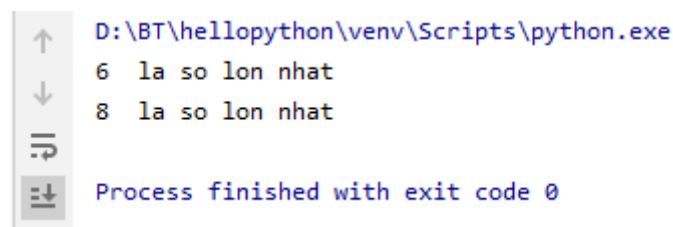
Ví dụ:



```
function_param.py x
1 def print_max(a, b):
2     if a > b:
3         print(a, ' la so lon nhat')
4     elif a == b:
5         print(a, ' bang ', b)
6     else:
7         print(b, ' la so lon nhat')
8
9
10 print_max(5, 6)
11
12 x = 7
13 y = 8
14
15 print_max(x, y)
```

Hình 1.9-3 Ví dụ hàm Function parameters

Kết quả:



```
D:\BT\hellopython\venv\Scripts\python.exe
6 la so lon nhat
8 la so lon nhat
Process finished with exit code 0
```

Hình 1.9-4 kết quả ví dụ hàm Function parameters

### 1.9.3 Câu lệnh return trong Python

Lệnh return thường được dùng để thoát hàm và trở về nơi mà tại đó hàm được gọi. Lệnh này có thể chứa biểu thức được tính toán và giá trị trả về. Nếu không có biểu thức nào trong câu lệnh hoặc không có lệnh return trong hàm thì hàm sẽ trả về None. Lệnh return dùng để trả về một giá trị (hoặc một biểu thức), hoặc đơn giản

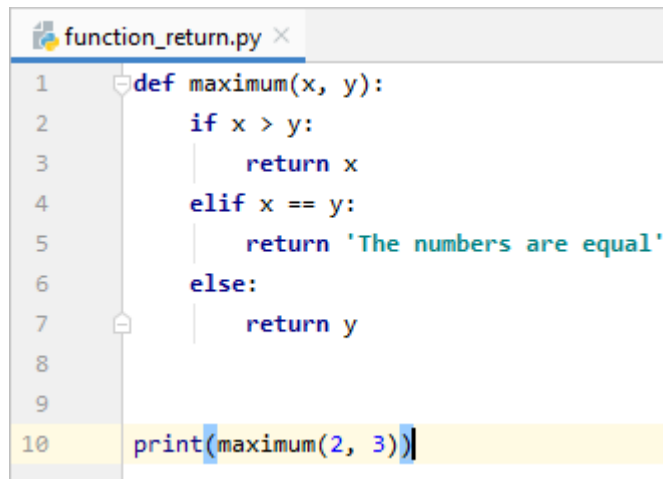
là trả về "không gì cả". Khi lệnh return được thực thi, hàm sẽ kết thúc. return là lệnh không bắt buộc phải có trong thân hàm.

Ví dụ	Mô tả
return 3	Hàm trả về một giá trị, và kết thúc
return	Hàm trả về không gì cả, và kết thúc

Hình 1.9-5 lệnh return

Cú pháp: return [danh\_sach\_bieu\_thuc]

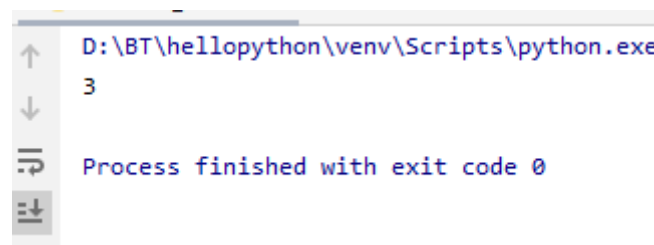
Ví dụ về lệnh return so sánh hai số x, y và tìm số lớn nhất. nếu x=y thì trả về kết quả hai số bằng nhau. Nếu  $x > y$  thì in ra x và ngược lại.



```
function_return.py x
1 def maximum(x, y):
2     if x > y:
3         return x
4     elif x == y:
5         return 'The numbers are equal'
6     else:
7         return y
8
9
10 print(maximum(2, 3))
```

Hình 1.9-6 ví dụ lệnh return

Kết quả đầu ra: in ra giá trị lớn nhất trong hai số cho trước



```
D:\BT\hellopython\venv\Scripts\python.exe
3
Process finished with exit code 0
```

Hình 1.9-7 kết quả ví dụ lệnh return

### 1.9.4 DocStrings trong Python

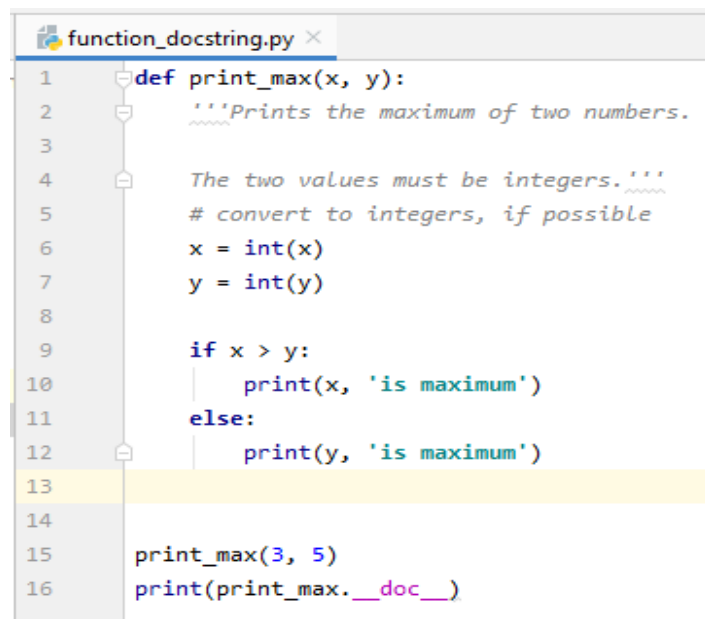
Chuỗi đầu tiên ngay sau tiêu đề hàm được gọi là *docstring* (documentation string), nó được dùng để giải thích chức năng cho hàm. Mặc dù docstring là không bắt buộc, nhưng việc giải thích ngắn gọn về chức năng của hàm sẽ giúp người dùng sau khi gọi hàm có thể hiểu ngay hàm sẽ làm gì mà không cần phải tìm lại định nghĩa hàm để xem xét.

Việc thêm tài liệu cho code giúp sau khi quay trở lại có thể nhớ được chi tiết, rõ ràng đoạn code đã viết trước đó mà không có sai sót gì.

Python có một tính năng tiện lợi gọi là *chuỗi tài liệu*, thường được gọi bằng tên ngắn hơn của nó *docstrings*. *DocStrings* là một công cụ quan trọng mà bạn nên sử dụng vì nó giúp ghi lại chương trình tốt hơn và dễ hiểu hơn. Chúng ta thậm chí có thể in hoặc sử dụng các chuỗi này trong các hàm.

### 1.9.5 Ví dụ về hàm DocStrings trong Python

Cho hai số nguyên x, y tìm số lớn hơn. nếu: x lớn hơn y thì in ra “x is maximum”. và ngược lại nếu y lớn hơn y thì in ra “y is maximum” và kết thúc.



```
function_docstring.py x
1 def print_max(x, y):
2     '''Prints the maximum of two numbers.
3
4     The two values must be integers.'''
5     # convert to integers, if possible
6     x = int(x)
7     y = int(y)
8
9     if x > y:
10        print(x, 'is maximum')
11    else:
12        print(y, 'is maximum')
13
14
15    print_max(3, 5)
16    print(print_max.__doc__)
```

Hình 1.9-8 ví dụ hàm DocString

Kết quả: in ra giá trị lớn nhất (y=5, y >3) suy ra y is maximum

```
D:\BT\hellopython\venv\Scripts\python.exe
5 is maximum
Prints the maximum of two numbers.

    The two values must be integers.

Process finished with exit code 0
```

Hình 1.9-9 Kết quả ví dụ hàm DocString

## 1.10 Đọc file CSV trong python

### 1.10.1 Tập CSV là gì ?

Một tập csv là viết tắt của (comma separated values) được định nghĩa là định dạng tập đơn giản sử dụng cấu trúc cụ thể để sắp xếp dữ liệu dạng bảng.

Tập CSV là một loại tập văn bản thuần túy sử dụng cấu trúc cụ thể để sắp xếp dữ liệu dạng bảng. CSV là một định dạng phổ biến để trao đổi dữ liệu vì sự nhỏ gọn, đơn giản và tính khái quát. Nhiều dịch vụ trực tuyến cho phép người dùng xuất dữ liệu dạng bảng từ trang web thành tập CSV. Các tập của CSV sẽ mở bằng Excel và gần như tất cả các cơ sở dữ liệu đều có công cụ cho phép nhập từ tập CSV. Định dạng chuẩn được xác định bởi dữ liệu hàng và cột. Hơn nữa, mỗi hàng được kết thúc bởi một dòng mới để bắt đầu hàng tiếp theo. Cũng trong hàng, mỗi cột được phân tách bằng dấu phẩy.

### 1.10.2 Các hàm của module CSV trong Python

Công việc mô-đun CSV được sử dụng để xử lý các tập CSV để đọc ghi và nhận dữ liệu từ các cột được chỉ định. Có nhiều loại hàm CSV khác nhau, như sau:

*csv.field\_size\_limit* - Nó trả về kích thước max của trường hiện tại.

*csv.get\_dialect* - Trả về dialect liên kết đến tên.

*csv.list\_dialects* - Trả về tên của tất cả các dialect đã đăng ký.

*csv.reader* - Đọc dữ liệu từ tập csv.

*csv. register\_dialect* - Nó liên kết dialect với một tên. Tên phải là một chuỗi hoặc một đối tượng Unicode.

*csv. writer* - Ghi dữ liệu vào tệp csv.

*csv. unregister\_dialect* - Nó xóa dialect được liên kết với tên từ sổ đăng ký dialect. Nếu tên không phải là tên dialect đã đăng ký, thì lỗi sẽ được đưa ra.

*csv. QUOTE\_ALL* - Nó hướng dẫn đối tượng writer trích dẫn tất cả các trường.

*csv. QUOTE\_MINIMAL* - Nó hướng dẫn đối tượng writer chỉ trích dẫn những trường có chứa các ký tự đặc biệt như dấu ngoặc kép, dấu phân cách, v.v.

*csv. QUOTE\_NONNUMERIC* - Nó hướng dẫn đối tượng writer trích dẫn tất cả các trường không phải là số.

*csv. QUOTE\_NONE* - Nó hướng dẫn đối tượng writer không bao giờ trích dẫn các trường.

### 1.10.3 Cách đọc tệp CSV

Để đọc dữ liệu từ tệp CSV, cần phải sử dụng hàm đọc để tạo đối tượng đọc.

Hàm đọc được viết để lấy từng hàng của tệp và tạo danh sách tất cả các cột. Sau đó, chọn cột bạn muốn trích xuất dữ liệu.

Vi dụ đọc và hiển thị file “1. csv”

```
docfile.py
1
2 import pandas
3
4 df = pandas.read_csv("C:\\Users\\nhoxinh\\Desktop\\1.csv")
5
6 print(df)
```

Hình 1.10-1 ví dụ cách đọc tệp CSV

**Kết quả:** trong file “1. csv” có 2526 hàng và 14 cột.

```

<Ticker> <DTYYYYMMDD> <OpenFixed> ... <VolumeDeal> <VolumeFB> <VolumeFS>
0 AAA 20200907 12.6000 ... 482000 121650 10000
1 AAA 20200904 12.3500 ... 0 1000 20750
2 AAA 20200903 12.6000 ... 0 14900 95030
3 AAA 20200901 12.5000 ... 0 3000 560
4 AAA 20200831 12.7000 ... 0 8360 6860
... ..
2521 AAA 20100721 12.2413 ... 20000 0 0
2522 AAA 20100720 12.9996 ... 0 0 0
2523 AAA 20100719 13.4058 ... 40000 0 0
2524 AAA 20100716 13.4059 ... 0 0 0
2525 AAA 20100715 13.8121 ... 0 0 0
[2526 rows x 14 columns]
[Finished in 1.8s]

```

Hình 1.10-2 kết quả ví dụ đọc tệp CSV

### 1.10.4 Cách viết tệp CSV

Khi muốn lưu trữ một bộ dữ liệu trong tệp CSV, ta sử dụng hàm `writer()`. Để ghi lại dữ liệu qua các hàng (dòng), sử dụng hàm `writerow()`.

Ví dụ ghi dữ liệu vào một tệp "writeData.csv" trong đó dấu phân cách là dấu nhảy đơn.

```

#import necessary modules
import csv

with open('X:\writeData.csv', mode='w') as file:
    writer = csv.writer(file, delimiter=',', quotechar='"', quoting=csv.QUOTE_MINIMAL)

    #way to write to csv file
    writer.writerow(['Programming language', 'Designed by', 'Appeared', 'Extension'])
    writer.writerow(['Python', 'Guido van Rossum', '1991', '.py'])
    writer.writerow(['Java', 'James Gosling', '1995', '.java'])
    writer.writerow(['C++', 'Bjarne Stroustrup', '1985', '.cpp'])

```

Hình 1.10-3 Cách viết tệp CSV

Kết quả trong tệp csv là:

```

Programming language, Designed by, Appeared, Extension
Python, Guido van Rossum, 1991, .py
Java, James Gosling, 1995, .java
C++, Bjarne Stroustrup, 1983, .cpp

```

Hình 1.10-4 kết quả trong tệp CSV

## 1.11 Sử dụng thư viện pandas trong Python

### 1.11.1 Thư viện pandas python là gì?

Thư viện pandas trong python là một thư viện mã nguồn mở, hỗ trợ đắc lực trong thao tác dữ liệu. Đây cũng là bộ công cụ phân tích và xử lý dữ liệu mạnh mẽ của ngôn ngữ lập trình python. Thư viện này được sử dụng rộng rãi trong cả nghiên cứu lẫn phát triển các ứng dụng về khoa học dữ liệu. Thư viện này sử dụng một cấu trúc dữ liệu riêng là Dataframe. Pandas cung cấp rất nhiều chức năng xử lý và làm việc trên cấu trúc dữ liệu này. Chính sự linh hoạt và hiệu quả đã khiến cho pandas được sử dụng rộng rãi.

### 1.11.2 Tại sao lại dùng thư viện pandas?

- DataFrame đem lại sự linh hoạt và hiệu quả trong thao tác dữ liệu và lập chỉ mục;
- Là một công cụ cho phép đọc/ ghi dữ liệu giữa bộ nhớ và nhiều định dạng file: csv, text, excel, sql database, hdf5;
- Liên kết dữ liệu thông minh, xử lý được trường hợp dữ liệu bị thiếu. Tự động đưa dữ liệu lộn xộn về dạng có cấu trúc;
- Dễ dàng thay đổi bố cục của dữ liệu;
- Tích hợp cơ chế trượt, lập chỉ mục, lấy ra tập con từ tập dữ liệu lớn.
- Có thể thêm, xóa các cột dữ liệu;
- Tập hợp hoặc thay đổi dữ liệu với *group by* cho phép bạn thực hiện các toán tử trên tập dữ liệu;
- Hiệu quả cao trong trộn và kết hợp các tập dữ liệu;
- Lập chỉ mục theo các chiều của dữ liệu giúp thao tác giữa dữ liệu cao chiều và dữ liệu thấp chiều;
- Tối ưu về hiệu năng;
- Pandas được sử dụng rộng rãi trong cả học thuật và thương mại. Bao gồm thống kê, thương mại, phân tích, quảng cáo, v.v.

### 1.11.3 Đọc file csv sử dụng thư viện pandas

#### Đọc csv file vào dataframe

Bạn có thể dễ dàng đọc vào một file. csv bằng cách sử dụng hàm `read_csv` và được trả về 1 dataframe. Mặc định, hàm này sẽ phân biệt các trường của file csv theo dấu phẩy. Cách đọc như sau:

```
peoples_df = pd.read_csv('./people.csv')
```

Để in ra n bản ghi đầu tiên của dataframe sử dụng hàm `head`. Ngược lại của hàm `head` là hàm `tail`:

```
peoples_df.head(5)
```

Kết quả in ra có dạng như sau:

	person_ID	name	first	last	middle	email	phone	fax	title
0	3130	Burks, Rosella	Rosella	Burks	NaN	BurksR@univ.edu	963.555.1253	963.777.4065	Professor
1	3297	Avila, Damien	Damien	Avila	NaN	AvilaD@univ.edu	963.555.1352	963.777.7914	Professor
2	3547	Olsen, Robin	Robin	Olsen	NaN	OlsenR@univ.edu	963.555.1378	963.777.9262	Assistant Professor
3	1538	Moises, Edgar Estes	Edgar	Moises	Estes	MoisesE@univ.edu	963.555.2731x3565	963.777.8264	Professor
4	2941	Brian, Heath Pruitt	Heath	Brian	Pruitt	BrianH@univ.edu	963.555.2800	963.777.7249	Associate Curator

Hình 1.11-1 Đọc file CSV sử dụng pandas

Một số tham số của hàm `read_csv`:

- *encoding*: chỉ định encoding của file đọc vào. Mặc định là *utf-8*.
- *sep*: thay đổi dấu ngăn cách giữa các cột. Mặc định là dấu phẩy (',' )
- *header*: chỉ định file đọc vào có header(tiêu đề của các cột) hay không. Mặc định là *infer*.
- *index\_col*: chỉ định chỉ số cột nào là cột chỉ số(số thứ tự). Mặc định là *None*.
- *nrows*: chỉ định số bản ghi sẽ đọc vào. Mặc định là *None* – đọc toàn bộ.

- `import pandas as pd`
- `import matplotlib.pyplot as plt`
- `import random`



- `peoples_df = pd. read_csv('. /people. csv', encoding='utf-8', header=None, sep=',')`
- `peoples_df. head(5)`

Khi chỉ định không có header, dòng header của chúng ta đã biến thành 1 bản ghi dữ liệu:

	0	1	2	3	4	5	6	7	8
0	person_ID	name	first	last	middle	email	phone	fax	title
1	3130	Burks, Rosella	Rosella	Burks	NaN	BurksR@univ.edu	963.555.1253	963.777.4065	Professor
2	3297	Avila, Damien	Damien	Avila	NaN	AvilaD@univ.edu	963.555.1352	963.777.7914	Professor
3	3547	Olsen, Robin	Robin	Olsen	NaN	OlsenR@univ.edu	963.555.1378	963.777.9262	Assistant Professor
4	1538	Moises, Edgar Estes	Edgar	Moises	Estes	MoisesE@univ.edu	963.555.2731x3565	963.777.8264	Professor

*Hình 1.11-2 Ví dụ*

#### 1.11.4 Thao tác với dataframe trong pandas

Khi đọc file csv như phía trên, `peoples_df` sẽ là một đối tượng có hàng, cột gọi là DataFrame. Vậy nếu bạn muốn lấy giá trị theo hàng hoặc cột cụ thể nào đó thì làm như nào?

Thư viện pandas python cung cấp rất nhiều hàm hữu ích cho phép bạn thao tác với đối tượng dữ liệu dataframe này.

##### 1.11.4.1 Xem thông tin của dataframe

Bạn có thể xem thông tin của dataframe vừa đọc vào bằng cách sử dụng hàm `info()` hoặc xem kích thước của dataframe này với thuộc tính `shape`. Cụ thể như sau:

```
# Xem chiều dài của df, tương đương shape[0]
print('Len:', len(peoples_df))
# Xem thông tin dataframe vừa đọc được
peoples_df. info()
```

```
# Xem kích thước của dataframe
print('Shape:', peoples_df. shape)
```

Kết quả thực hiện:

```
0
1 Len: 40
2 <class 'pandas.core.frame.DataFrame'>
3 RangeIndex: 40 entries, 0 to 39
4 Data columns (total 11 columns):
5 person_ID    40 non-null int64
6 name         40 non-null object
7 first        40 non-null object
8 last         40 non-null object
9 middle       15 non-null object
10 email        40 non-null object
11 phone        40 non-null object
12 fax          40 non-null object
13 title        40 non-null object
14 age          40 non-null int64
15 is_young     40 non-null bool
16 dtypes: bool(1), int64(2), object(8)
17 memory usage: 3.2+ KB
18 Shape: (40, 11)
19
```

Hình 1.11-3

### 1.11.4.2 Truy xuất dữ liệu trên dataframe

❖ Lấy 1 cột theo tên cột

Để chỉ định cột muốn lấy, bạn chỉ cần truyền vào tên cột như sau:

```
peoples_df['name']
```

Kết quả thực hiện:

```
In [18]: peoples_df['name']
Out[18]: 0      Burks, Rosella
1      Avila, Damien
2      Olsen, Robin
3      Moises, Edgar Estes
4      Brian, Heath Pruitt
5      Claude, Elvin Haney
6      Mosley, Edmund
7      Derek, Antoine Mccoy
8      Hawkins, Callie
9      Pate, Andrea
10     Austin, Liz
11     Kendrick, Reba Alford
12     Sims, Angelina
13     Mullins, Kimberly
14     Chuck, Lloyd Haney
15     Payne, Ladonna
16     Baxter, Johnathan Browning
17     Weiss, Gilbert
```

lấy 1 cột của dataframe dùng thư viện pandas python

Hình 1.11-4 kết quả theo 1 cột

### ❖ Lấy theo nhiều cột

Thay vì truyền vào 1 string thì hãy truyền vào 1 list các tên cột. Mình thêm.

`head(5)` để chỉ in ra 5 bản ghi đầu tiên cho ngắn, mặc định sẽ lấy hết.

```
peoples_df[['name', 'age']]. head(5)
```

Kết quả:

	name	age
0	Burks, Rosella	25
1	Avila, Damien	23
2	Olsen, Robin	21
3	Moises, Edgar Estes	18
4	Brian, Heath Pruitt	30

*lấy nhiều cột trong dataframe*

Hình 1.11-5 kết quả theo nhiều cột

### ❖ Lấy bản ghi theo chỉ số

Để lấy một hoặc nhiều bản ghi liên tiếp trong dataframe, sử dụng cơ chế trượt theo chỉ số giống như trên list trong python. Lấy 5 bản ghi đầu tiên:

```
peoples_df[0:5]
```

Trong trường hợp này kết quả giống như hàm head phía trên. Đều là lấy 5 bản ghi đầu tiên.

	0	1	2	3	4	5	6	7	8
0	person_ID	name	first	last	middle	email	phone	fax	title
1	3130	Burks, Rosella	Rosella	Burks	NaN	BurksR@univ.edu	963.555.1253	963.777.4065	Professor
2	3297	Avila, Damien	Damien	Avila	NaN	AvilaD@univ.edu	963.555.1352	963.777.7914	Professor
3	3547	Olsen, Robin	Robin	Olsen	NaN	OlsenR@univ.edu	963.555.1378	963.777.9262	Assistant Professor
4	1538	Moises, Edgar Estes	Edgar	Moises	Estes	MoisesE@univ.edu	963.555.2731x3565	963.777.8264	Professor

Hình 1.11-6 Lấy bản ghi theo chỉ số

### ❖ Lấy giá trị trả về numpy arrays

Để lấy giá trị của một cột trả về dưới dạng numpy array trong thư viện pandas python, bạn chỉ cần thêm `values` vào sau, ví dụ:

```
peoples_df['name']. values
```

Kết quả thu được như sau:

```
array(['Burks, Rosella ', 'Avila, Damien ', 'Olsen, Robin ',  
      'Moises, Edgar Estes', 'Brian, Heath Pruitt',  
      'Claude, Elvin Haney', 'Mosley, Edmund ', 'Derek, Antoine Mccoy',  
      'Hawkins, Callie ', 'Pate, Andrea ', 'Austin, Liz ',  
      'Kendrick, Reba Alford', 'Sims, Angelina ', 'Mullins, Kimberly ',  
      'Chuck, Lloyd Haney', 'Payne, Ladonna ',  
      'Baxter, Johnathan Browning', 'Weiss, Gilbert ',  
      'Deirdre, Florence Barrera', 'Fernando, Toby Calderon',  
      'Garrison, Patrica ', 'Effie, Leila Vinson', 'Buckley, Rose ',  
      'Stanton, Kathie ', 'Banks, Shannon ', 'Barnes, Cleo ',  
      'Brady, Nellie ', 'Katheryn, Ruben Holt', 'Michael, Dianne ',  
      'Grant, Adam ', 'Head, Kurtis ', 'Berger, Jami ',  
      'Earline, Jaime Fitzgerald', 'Evelyn, Summer Frost',  
      'Quentin, Sam Hyde', 'Dunlap, Ann ', 'Shields, Rich Pena',  
      'Page, Winnie ', 'Sparks, Ezra ', 'Kaufman, Elba '], dtype=object)
```

Hình 1.11-7 kết quả numpy arrays trả về

## 1.12 Vẽ biểu đồ trong python

### 1.12.1 Matplotlib

Để thực hiện các suy luận thống kê cần thiết, cần phải trực quan hóa dữ liệu của nó. Nó là một thư viện vẽ đồ thị rất mạnh mẽ hữu ích cho những người làm việc với Python và NumPy. Module được sử dụng nhiều nhất của Matplotlib là Pyplot cung cấp giao diện như MATLAB nhưng thay vào đó, nó sử dụng Python và nó là nguồn mở.

Để cài đặt Matplotlib nếu bạn có Anaconda chỉ cần gõ *conda install matplotlib* hoặc sử dụng tools pip *pip install matplotlib*.

### 1.12.2 Khái niệm chung

Một Matplotlib figure có thể được phân loại thành nhiều phần như dưới đây:

Figure: Như một cái cửa sổ chứa tất cả những gì bạn sẽ vẽ trên đó.

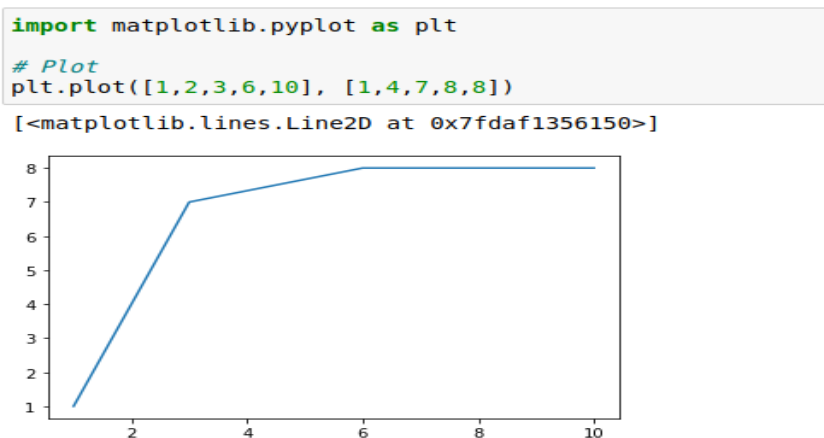
Axes: Thành phần chính của một figure là các axes (những khung nhỏ hơn để vẽ hình lên đó). Một figure có thể chứa một hoặc nhiều axes. Nói cách khác, figure chỉ là khung chứa, chính các axes mới thật sự là nơi các hình vẽ được vẽ lên.

Axis: Chúng là dòng số giống như các đối tượng và đảm nhiệm việc tạo các giới hạn biểu đồ.

Artist: Mọi thứ mà bạn có thể nhìn thấy trên figure là một artist như Text objects, Line2D objects, collection objects. Hầu hết các Artists được gắn với Axes.

### 1.12.3 Ví dụ tạo một biểu đồ đơn giản

Sau đây ta vẽ một biểu đồ đơn giản với phương thức plot(), mặc định nó sẽ vẽ biểu đồ đường. Với các tham số tối thiểu là list các tọa độ x, list tọa độ y (lưu ý 2 list này phải có độ dài bằng nhau).



Hình 1.12-1 ví dụ tạo biểu đồ

Để thay đổi một chút, ta có thể thêm tham số về định dạng vào

Chẳng hạn như: tham số màu: r - red, g -green, b - blue, y - yellow, ... định dạng đường: - (đường thẳng), -- (đường nét đứt), nếu không truyền tham số không vẽ đường.

## Chương 2 BÀI TOÁN DỰ BÁO

**Dự báo** là quá trình tạo ra các nhận định về các hiện tượng mà thông thường các đầu ra của chúng còn chưa quan sát được.

### 2.1 Bài toán dự báo

Dự báo là một trong những yếu tố quan trọng nhất trong việc ra các quyết định quản lý bởi vì ảnh hưởng sau cùng của một quyết định thường phụ thuộc vào sự tác động của các nhân tố không thể nhìn thấy tại thời điểm ra quyết định. Vai trò của dự báo là nhạy cảm trong các lĩnh vực như tài chính, nghiên cứu thị trường, lập kế hoạch sản xuất, hành chính công, điều khiển quá trình sản xuất hay nghiên cứu, . . . Trong giới doanh nhân, các câu hỏi thường xuyên được đưa ra là:

Lượng hàng sẽ bán trong tháng tới là bao nhiêu?

Tháng này nên đặt mua bao nhiêu hàng?

Nên giữ bao nhiêu cổ phiếu?

Nên mua bao nhiêu nguyên liệu?

Mục tiêu bán hàng sắp tới là gì?

Có nên tăng nhân công không?

#### 2.1.1 Dự báo hỗ trợ quá trình ra quyết định trong các tình huống

Điều tiết nguồn tài nguyên sẵn có: Dự báo nhu cầu cho sản phẩm, nguyên liệu, nhân công, tài chính hay dịch vụ như là một đầu vào thiết yếu để điều tiết kế hoạch sản xuất, vận tải, tiền vốn và nhân lực.

Yêu cầu thêm tài nguyên: Dự báo giúp xác định tài nguyên cần có trong tương lai (như nhân lực, máy móc thiết bị, vốn. . .)

Thiết kế, lập quy hoạch: Dự báo các hiện tượng thiên nhiên như lũ lụt, hạn hán để thiết kế các công trình như đê, đập, hồ chứa và quy hoạch vùng sản xuất. Nhược điểm của dự báo là không thể tránh khỏi sai số. Trên quan điểm thực tiễn, cần hiểu

rõ cả mặt mạnh lẫn mặt hạn chế của các phương pháp dự báo và tính đến chúng trong khi sử dụng dự báo.

### 2.1.2 Chuỗi thời gian (Time Series).

Chuỗi thời gian là một dãy dữ liệu được quan sát ở các thời điểm kế tiếp nhau với cùng một đơn vị đo mẫu.

Trong chuỗi thời gian, trình tự thời gian đóng một vai trò thực sự quan trọng, vì vậy các tính toán thống kê thông thường như trung bình mẫu, độ lệch quân phương mẫu, khoảng tin cậy, kiểm định các giả thuyết, v.v. không còn thích hợp

Một chuỗi thời gian thường bao gồm những thành phần sau đây

#### **Thành phần xu thế(xu hướng).**

Xu hướng thể hiện sự tăng trưởng hoặc giảm sút của một biến số theo thời gian với khoảng thời gian đủ dài. Một số biến số kinh tế có xu hướng tăng giảm dài hạn như:

Tốc độ tăng dân số của Việt Nam có xu hướng giảm.

Tỷ trọng nông nghiệp trong GDP của Việt Nam có xu hướng giảm.

Mức giá có xu hướng tăng.

#### **Thành phần mùa (thời vụ).**

Biến động thời vụ của biến số kinh tế là sự thay đổi lặp đi lặp lại từ năm này sang năm khác theo mùa vụ. Biến động thời vụ xảy ra do khí hậu, ngày lễ, phong tục tập quán...Biến động thời vụ có tính ngắn hạn với chu kỳ lặp lại thường là 1 năm.

#### **Thành phần ngẫu nhiên.**

Những dao động không thuộc ba loại trên được xếp vào dao động ngẫu nhiên. Các nguyên nhân gây ra biến động ngẫu nhiên có thể là thời tiết bất thường, chiến tranh, khủng hoảng năng lượng, biến động chính trị...

### Thành phần chu kì (dài hạn).

Các số liệu kinh tế vĩ mô thường có sự tăng giảm có quy luật theo chu kỳ kinh tế. Sau một thời kỳ suy thoái kinh tế sẽ là thời kỳ phục hồi và bùng nổ kinh tế, kế tiếp tăng trưởng kinh tế sẽ chững lại và khởi đầu cho một cuộc suy thoái mới. Tùy theo nền kinh tế mà chu kỳ kinh tế có thời hạn là 5 năm, 7 năm hay 10 năm.

#### 2.1.2.1 Các phương pháp hiển thị chuỗi thời gian.

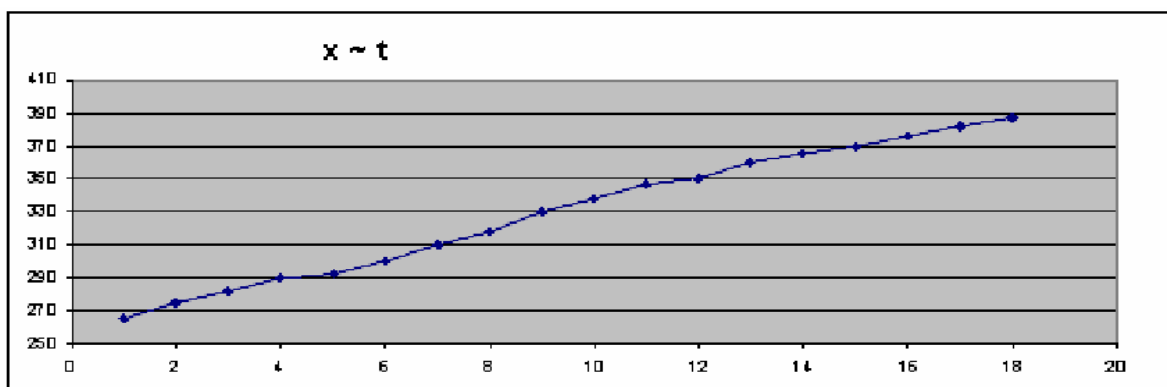
Phân tích chuỗi thời gian bao gồm việc nghiên cứu dạng dữ liệu trong quá khứ và giải thích các đặc điểm chính của nó. Một trong các phương pháp đơn giản và hiệu quả nhất là hiển thị trực quan chuỗi đó. Các đặc điểm không dễ thấy trong bảng dữ liệu thường nổi lên qua các minh họa đồ thị.

Ví dụ cho bảng dữ liệu sau:

t	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
$x_t$	265	275	282	290	292	300	310	318	330	338	347	350	360	365	370	376	382	387
$x_t/x_{t-1}$		104	103	103	101	103	103	103	104	102	103	101	103	101	101	102	102	101
$x_t - x_{t-1}$		10	7	8	2	8	10	8	12	8	9	3	10	5	5	6	6	5

❖ Ba loại đồ thị minh họa chuỗi thời gian tương ứng với bản dữ liệu trên là:

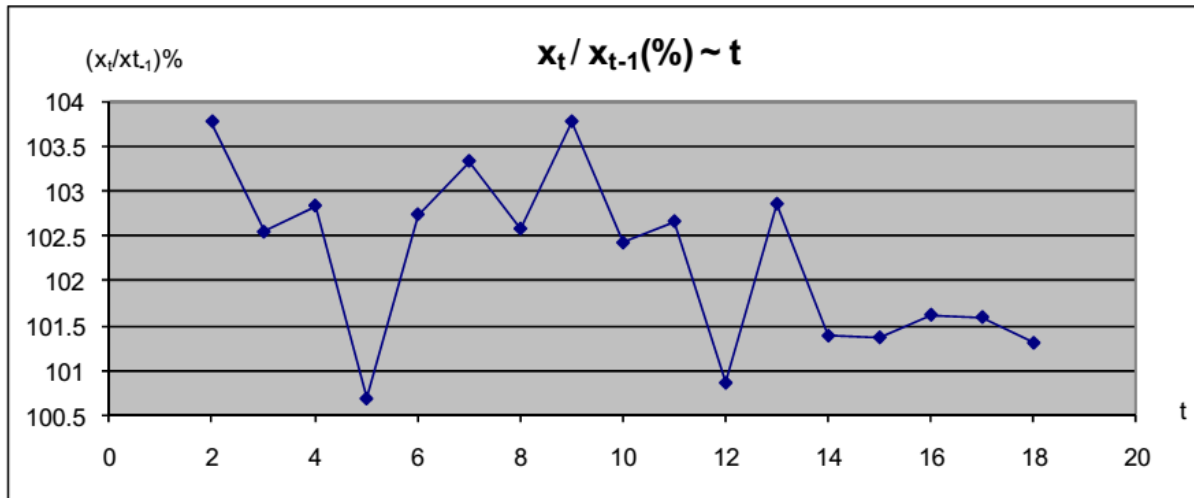
#### 2.1.2.2 Đồ thị của $x_t$ theo t: cung cấp lịch sử dữ liệu gốc chưa bị chuyển đổi qua bất cứ phép biến đổi nào, giúp cho việc nghiên cứu xu thế và nhận dạng.



Hình 2.1-1 Đồ thị của  $x_t$  theo t

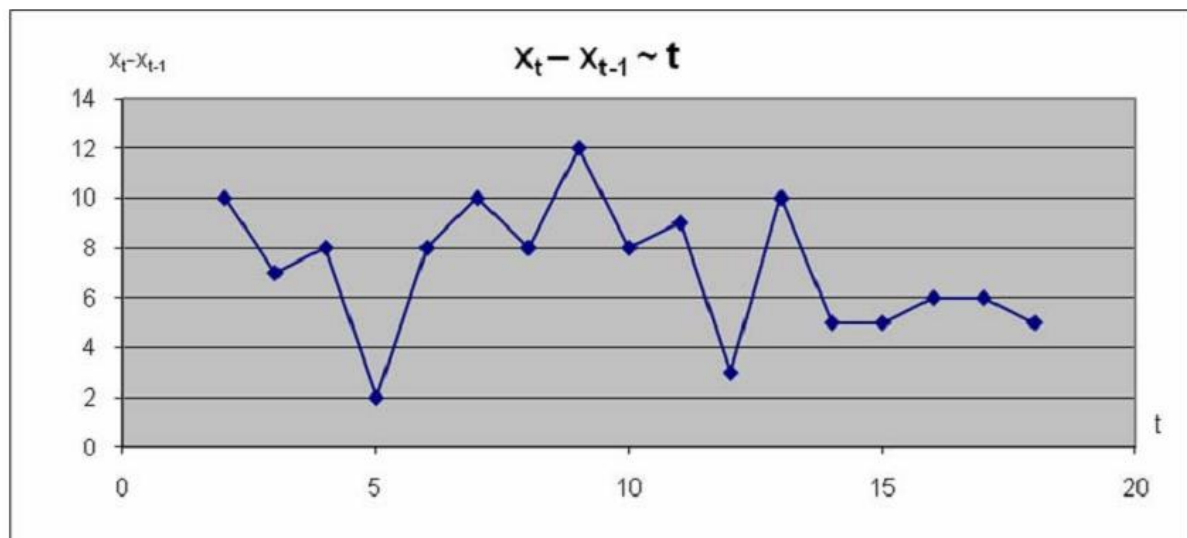


2.1.2.3 Đồ thị của  $x_t/x_{t-1} \cdot 100$  theo t: mỗi điểm trên đồ thị này cho biết giá trị hiện thời của chuỗi tăng hay giảm so với giá trị trước đó.



Hình 2.1-2 Đồ thị của  $x_t/x_{t-1} \cdot 100$  theo t

2.1.2.4 . Đồ thị của  $x_t - x_{t-1}$  theo t: Đồ thị này biểu diễn sự thay đổi giữa các bước thời gian kế tiếp nhau. Nhìn vào đồ thị ta thấy được khoảng các giá trị biến đổi giữa các bước kế nhau.



Hình 2.1-3 Đồ thị của  $x_t - x_{t-1}$  theo t

### 2.1.2.5 Các định dạng dữ liệu

Trước khi áp dụng bất cứ một phương pháp dự báo khoa học cho một tình huống nào, cần phải ghép nối các thông tin (dữ liệu có liên quan) về tình huống đó càng nhiều càng tốt. Những dữ liệu đó được phân thành 2 loại:

Các dữ liệu bên trong, ví dụ số liệu sản phẩm bán ra trong quá khứ, v.v.

Các dữ liệu bên ngoài, ví dụ như các thống kê của ngân hàng về tình hình tài chính của công ty (phản ánh thông tin bên trong).

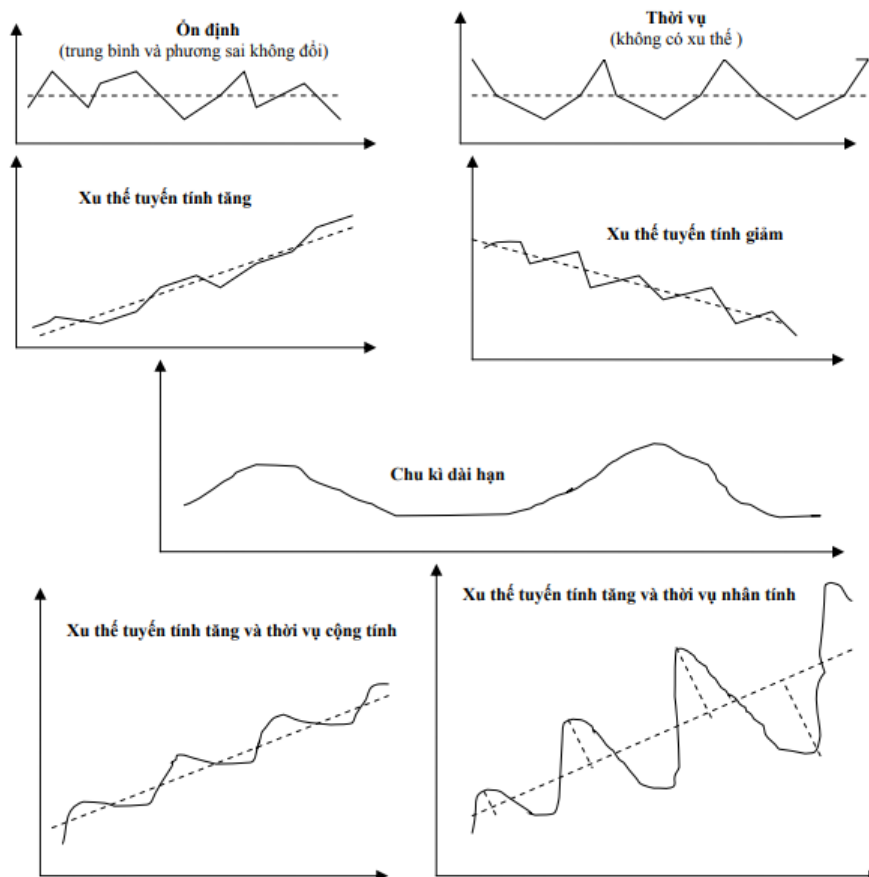
Từ các thông tin này, người làm dự báo phải chọn ra thông tin liên quan nhiều nhất đến tình huống cần dự báo. Chẳng hạn, trong dự báo bán hàng, báo cáo hàng bán được trong quá khứ của công ty sẽ cung cấp những thông tin tối thiểu cho việc dự báo.

Thông tin tối thiểu cần thỏa mãn các yêu cầu về:

Tính liên quan: Nó có phải là thông tin liên quan trực tiếp nhất không?

Độ tin cậy: Dữ liệu được thu thập như thế nào? Có đáng tin cậy không?

Tính thời sự: Liệu các thông tin mới nhất đã được cập nhật chưa? Chúng có sẵn khi cần không?

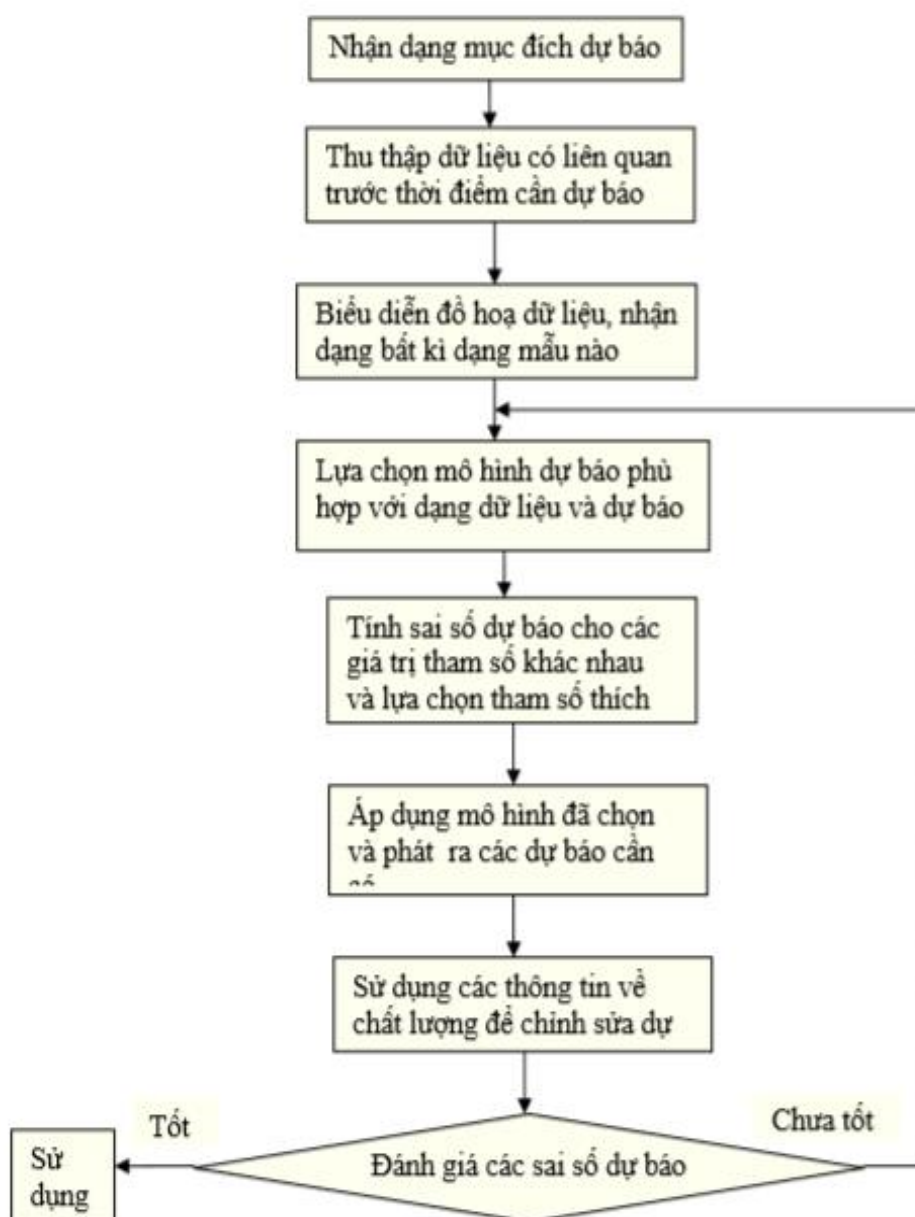


Hình 2.1-4 Một số định dạng dữ liệu

Khi đã có những thông tin tối thiểu cần thiết, ta cần phải nghiên cứu đặc điểm của nó bằng cách minh họa đồ thị. Dạng dữ liệu quá khứ là rất quan trọng vì nó quyết định việc lựa chọn mô hình dự báo. Mô hình dự báo được chọn phải tương thích với dạng dữ liệu mẫu trong quá khứ.

## 2.2 Tiến trình của một bài toán dự báo

Sơ đồ tiến trình chung của một bài toán dự báo (Nguyễn Thị Vinh, Đại học Thủy Lợi).



Hình 2.2-1 Tiến trình chung của một bài toán dự báo

### 2.3 Tiêu chuẩn dự báo.

Các tiêu chuẩn chung đánh giá sự thành công của một mô hình dự báo khi áp dụng vào một tập dữ liệu là:

- Trùng càng nhiều với các thay đổi ngẫu nhiên trong dữ liệu càng tốt.
- Không vượt quá xa bất kì một đặc tính nào của dữ liệu

Xét về mặt sai số, hai loại đặc tính cần quan tâm khi thử nghiệm một công thức dự báo trên dữ liệu là:

Các đặc tính thống kê: Một phương pháp dự báo tốt thường cho sai số trung bình nhỏ. Trong các mô hình dự báo, người ta thường sử dụng các loại sai số như

$$\text{MAE} = \frac{1}{n} \sum |e_i| \quad (\text{Mean Absolute Error})$$

$$\text{MSE} = \frac{1}{n} \sum e_i^2 \quad (\text{Mean Square Error})$$

$$\text{RMSE} = \sqrt{\text{MSE}} \quad (\text{squareRoot Mean Square Error})$$

ở đây sai số  $e_i = x_i - f_i$  với  $f_i$  là dự báo của  $x_i$

Các đặc tính định dạng: Trong các mô hình dự báo, sự có mặt của các dạng sai số (như tính lệch, tính chu kì, tính kiên định, v.v.) đều bị xem là dấu hiệu không tốt. Sự xuất hiện của bất cứ xu thế nào trong sai số cũng nên khử càng nhanh càng tốt. Có thể sai phân hóa chuỗi các giá trị ban đầu để đối phó với các tác động này.

Như vậy có hai tiêu chuẩn dự báo về định lượng và định tính là: sai số nhỏ và không tuân theo một định dạng nào.

## **Chương 3 BÀI TOÁN DỰ BÁO GIÁ CỔ PHIẾU VÀ THỰC NGHIỆM**

### **3.1 Giới thiệu bài toán dự báo giá cổ phiếu**

Đầu tư tài chính là một trong những cách thức kiếm tiền khá phổ biến hiện nay của người dùng. Bằng việc phân tích các yếu tố cơ bản hay yếu tố định lượng để dự báo sự tăng giảm của giá cổ phiếu, các nhà đầu tư sẽ ra quyết định mua hay bán cổ phiếu nhằm tìm kiếm lợi nhuận cho mình. Tuy nhiên, thị trường chứng khoán luôn có những yếu tố bất ngờ không theo ý chủ quan của người dùng, do đó dự đoán xu thế của thị trường chứng khoán là bài toán quan trọng đã và đang thu hút được sự quan tâm của các nhà đầu tư, các chuyên gia kinh tế học và các nhà khoa học. Bên cạnh đó, sự phát triển mạnh mẽ của công nghệ thông tin giúp cho việc trao đổi và lưu trữ dữ liệu về chứng khoán trở lên dễ dàng hơn bao giờ hết. Người dùng có thể dễ dàng tra cứu lịch sử tăng giảm của các mã cổ phiếu mà họ quan tâm. Mặt khác nhiều công cụ và ứng dụng được phát triển nhằm khai thác và xử lý dữ liệu tài chính khổng lồ cho các mô hình dự báo đã làm cho công việc này trở lên dễ dàng, nhanh và chính xác hơn.

Dự báo giá cổ phiếu luôn là bài toán thách thức vì tính không thể đoán trước của nó. Giả thuyết thị trường hàng ngày là không thể dự báo giá cổ phiếu và giá cổ phiếu là ngẫu nhiên. Tuy nhiên, bằng các phân tích kỹ thuật cho thấy hầu hết các giá cổ phiếu được phản ánh trong các dữ liệu trước đó, do đó xu hướng vận động giá là rất quan trọng để dự đoán giá hiệu quả. Hơn nữa, các nhóm cổ phiếu và thị trường chứng khoán bị ảnh hưởng bởi một số yếu tố kinh tế như sự kiện chính trị, điều kiện kinh tế chung, chỉ số giá hàng hóa, kỳ vọng của nhà đầu tư, sự vận động của thị trường chứng khoán khác, tâm lý của nhà đầu tư, v. vv.v. Giá trị của các nhóm chứng khoán được tính toán với vốn hóa thị trường cao. Có các thông số kỹ thuật khác nhau để xác định được dữ liệu thống kê từ giá trị của giá cổ phiếu. Thông thường, các chỉ số chứng khoán đạt được từ giá của các cổ phiếu có đầu tư thị trường cao và từ đó đưa ra đánh giá về tình trạng nền kinh tế ở mỗi quốc gia.

Bản chất của sự vận động giá cổ phiếu là mơ hồ và làm cho các khoản đầu tư có tính rủi ro cao. Trên thực tế, giá chứng khoán nói chung là động, không tham số và phi tuyến tính; do đó chúng thường làm giảm hiệu năng của các mô hình thống kê và không thể dự đoán giá và sự vận động chính xác của nó.

Học máy là công cụ mạnh nhất bao gồm các thuật toán khác nhau để tăng hiệu năng trong một số trường hợp nhất định. Học máy có khả năng xác định thông tin và phát hiện các mẫu từ bộ dữ liệu. Các mô hình học nhóm dựa trên máy học như học tăng cường (boosting) học bao đóng (Bagging) cho kết quả vượt trội so với từng phương pháp khi dự đoán chuỗi thời gian. Gần đây của các mô hình dựa trên cây như thuật toán tăng cường độ dốc (gradient boosting) và XGBoost được các nhà khoa học dữ liệu hàng đầu sử dụng trong các cuộc thi. Thêm nữa, một xu hướng hiện đại trong học máy là học sâu (Deep Learning) có khả năng tuyệt vời để trích xuất các thông tin từ chuỗi thời gian tài chính. Mạng nơ ron hồi quy (RNN) đã đạt được thành công đáng kể trong lĩnh vực phân tích tài chính với hiệu năng tuyệt vời của nó. Có thể thấy, quá trình dự đoán của thị trường chứng khoán không chỉ liên quan đến thông tin hiện tại mà dữ liệu trước đó đóng vai trò quan trọng. Việc huấn luyện mô hình sẽ không đủ nếu chỉ sử dụng dữ liệu vào thời điểm gần nhất. RNN có thể sử dụng mạng để duy trì bộ nhớ của các sự kiện gần đây và xây dựng các kết nối giữa mỗi đơn vị của mạng, do đó, nó hoàn toàn phù hợp với các dự báo kinh tế. Mô hình bộ nhớ dài ngắn (LSTM) là một cải tiến của phương pháp RNN được sử dụng trong lĩnh vực học sâu. LSTM có ba đầu ra khác nhau để loại bỏ các vấn đề trong các nơ ron RNN và cũng có thể xử lý các điểm dữ liệu đơn lẻ hoặc toàn bộ chuỗi dữ liệu. nhiều nghiên cứu sử dụng các phương pháp này đã được sử dụng trong dự báo thị trường.

Đã có nhiều phương pháp dự báo đã và đang được sử dụng cho dự báo sự biến động của giá chứng khoán. Các tiếp cận thống kê như: làm mịn theo cấp số nhân (exponential smoothing), trung bình động (moving average) và ARIMA là các mô hình tuyến tính phổ biến trong kinh tế để dự đoán xu thế vận động của giá trong tương lai. Bên cạnh đó, một số nghiên cứu cho bài toán dự báo như Mạng nơ ron

nhân tạo (ANN), giải thuật di truyền (GA), logic mờ, một số phương pháp học máy thông kê hiện đại như RNN, LSTM, v.v. đã được nghiên cứu trong thời gian gần đây có khả năng xử lý các biến động của thị trường chứng khoán với độ chính xác dự đoán cao.

### **3.2 Phát biểu bài toán**

Thị trường chứng khoán Việt Nam đã có một chặng đường dài phát triển với hơn 20 năm hoạt động. Trong thời gian đó, tổng sản phẩm quốc nội (GDP) bình quân đầu người tăng khoảng 7 lần và Việt Nam từ một nước nghèo đã trở thành một quốc gia có thu nhập trung bình. Đất nước hình chữ S đã tham gia một cách sâu rộng vào nền kinh tế thế giới với việc bình thường hóa quan hệ thương mại với Mỹ, gia nhập tổ chức thương mại thế giới WTO và ký hàng loạt các hiệp định thương mại tự do với hầu hết các cường quốc trên thế giới, trong đó có cả các hiệp định thế hệ mới như CPTPP, EVFTA, EVIPA, v.v. Do đó, có thể thấy những kết quả đạt được của thị trường chứng khoán khi thể hiện rõ vai trò là kênh dẫn vốn quan trọng của nền kinh tế, theo nhiều góc độ về định lượng như quy mô vốn hóa, cơ sở nhà đầu tư, số lượng sản phẩm, quy mô giao dịch hàng ngày, v.v. hay định tính như tính chuyên nghiệp của các thành viên, v.v.

Sau 20 năm vận hành, quy mô thị trường và đặc biệt các công ty chứng khoán (CTCK) đã có nhiều thay đổi cả về số lượng, quy mô lẫn chất lượng, sản phẩm cung cấp cho nhà đầu tư. Số lượng CTCK tăng vọt, quy mô vốn tăng hàng trăm lần so với thời điểm mới thành lập cùng với sự xuất hiện của nhiều CTCK ngoại. Phí giao dịch ngày càng thấp, nhiều CTCK miễn phí giao dịch. Nhiều sản phẩm mới ra đời, áp dụng công nghệ vào hoạt động hỗ trợ nhà đầu tư giao dịch thay vì viết phiếu lệnh bằng cách thực hiện Online. Các báo cáo phân tích được thực hiện chuyên nghiệp, nhiều công ty thực hiện báo cáo bằng tiếng Anh để hỗ trợ cho NĐT nước ngoài. Các CTCK không chỉ cung cấp thuần dịch vụ môi giới mà còn có nhiều nghiệp vụ khác như Ngân hàng đầu tư (IB), tư vấn niêm yết, bảo lãnh phát hành trái phiếu, cung cấp các sản phẩm phái sinh (HĐTL chỉ số VN30, HĐTL trái

phiếu chính phủ), v.v. Không những vậy, nhiều CTCK cũng đã áp dụng công nghệ vào hoạt động môi giới cũng như tư vấn đầu tư. Việc có thể dễ dàng tham gia thị trường và tìm kiếm thông tin đã tạo ra sức hút rất lớn đối với nhà đầu tư. Nhiều trang web cho phép người dùng có thể xem và tải các dữ liệu lịch sử về giá cũng như các báo cáo hoạt động của các công ty niêm yết trên thị trường một cách dễ dàng như: cafef.vn, cophieu68.vn, investing.com, tradingview.com, v.v. Giúp các nhà đầu tư có dữ liệu để thực hiện các phân tích kỹ thuật cũng như phân tích định lượng nhằm xác định xu hướng giá cổ phiếu mà mình quan tâm trong các hoạt động đầu tư của mình.

Trong nghiên cứu này, em sẽ thực hiện phân tích dự báo sự tăng giảm của giá cổ phiếu trong thị trường chứng khoán Việt Nam dựa trên phân tích chuỗi thời gian.

Xây dựng mô hình thực hiện yêu cầu sau:

**Input:** Cho giá của mã chứng khoán ở ngày  $t$

**Output:** Giá của mã chứng khoán đó trong ngày  $t+1$

### 3.3 Một số mô hình dự báo thống kê

#### 3.3.1 Mô hình trượt đơn Moving Average (MA) để phân tích dự báo giá chứng khoán.

Mô hình trung bình trượt đơn (Moving average) là cách tiếp cận ngây thơ nhất đối với mô hình chuỗi thời gian.

Mô hình này thực hiện quan sát tiếp theo là giá trị trung bình của tất cả các quan sát trong quá khứ.

Mô hình trượt đơn có thể sử dụng để xác định các xu hướng quan tâm trong dữ liệu. Ta có thể định nghĩa một cửa sổ trượt để áp dụng mô hình trượt để làm trơn cho các chuỗi thời gian và đánh dấu các xu hướng khác nhau.



$$\begin{aligned}
f_{t+1} &= (x_t + x_{t-1} + \dots + x_{t-N+1}) / N \\
&= (x_{t-1} + x_{t-2} + \dots + x_{t-N}) / N + x_t / N - x_{t-N} / N \\
\text{hay } f_{t+1} &= f_t + x_t / N - x_{t-N} / N
\end{aligned}$$

Dự báo ở thời điểm  $t+1$  chỉ là điều chỉnh của dự báo ở thời điểm  $t$  trước đó.

Khi  $N$  tăng đủ lớn thì lượng điều chỉnh  $x_t / N - x_{t-N} / N \rightarrow 0$  và trung bình trượt trở thành trung bình mẫu như phương pháp ngây thơ.

Chỉ nên áp dụng phương pháp này khi số giá trị quan sát được là ít và tập dữ liệu có tính ổn định theo thời gian.

Mặt khác, mô hình trượt đơn có thể sử dụng để xác định các xu hướng quan tâm trong dữ liệu. Ta có thể định nghĩa một cửa sổ trượt để áp dụng mô hình trượt để làm trơn cho các chuỗi thời gian và đánh dấu các xu hướng khác nhau.

### 3.3.2 Mô hình làm mịn theo cấp số nhân (exponential smoothing).

Mô hình làm mịn theo cấp số nhân “exponential smoothing” là một phương pháp dự báo chuỗi thời gian cho dữ liệu đơn biến.

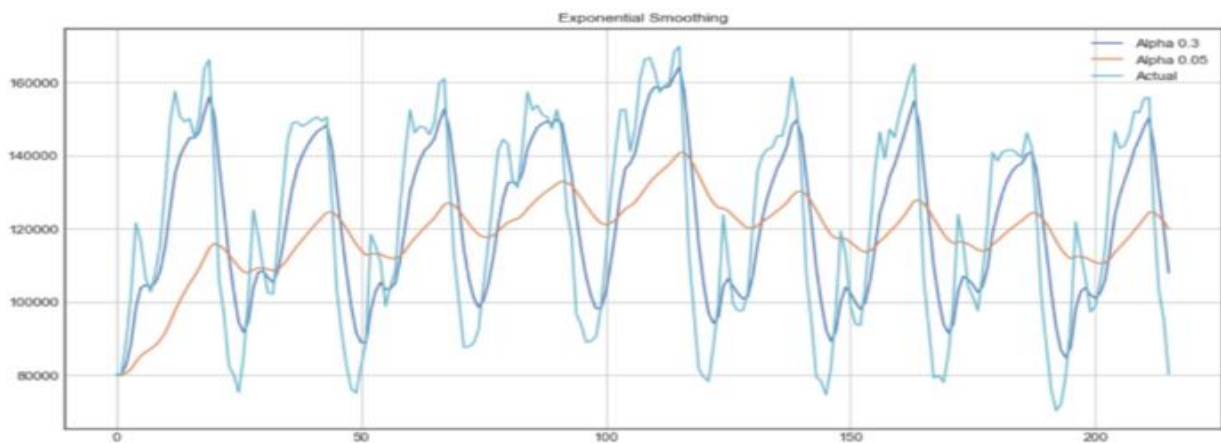
Mô hình làm mịn theo cấp số nhân thực hiện tương tự như mô hình trung bình động, nhưng trong đó sử dụng trọng số giảm dần khác nhau được gán cho mỗi quan sát. Nói cách khác, các quan sát lâu hơn (về quá khứ) hơn xa với hiện tại được đánh giá là ít quan trọng hơn.

Khi đó, công thức làm trơn theo cấp số nhân được thực hiện như sau:

$$y = \alpha x_t + (1 - \alpha)y_{t-1}, t > 0$$

Trong đó,  $\alpha$  là hệ số làm mịn nhận các giá trị từ 0 đến 1. Nó quyết định trọng số giảm nhanh như thế nào đối với các quan sát trước đó.

**Alpha** ( $\alpha$ ): Hệ số làm mịn cho cấp độ.



Hình 3.3-1 Ví dụ về mô hình làm mịn theo cấp số nhân

Trong ví dụ trên, đường màu xanh lam đậm biểu thị độ làm mịn theo cấp số nhân của chuỗi thời gian sử dụng hệ số làm mịn 0,3, trong khi đường màu cam sử dụng hệ số làm mịn 0,05.

Ta có thể thấy, hệ số làm mịn càng nhỏ thì chuỗi thời gian sẽ càng mượt. Điều này có nghĩa là khi hệ số làm mịn tiến tới 0, mô hình này sẽ tiệm cận mô hình trung bình động.

Các phương pháp làm trơn hàm mũ có thể được coi là các phương pháp ngang hàng và là phương pháp thay thế cho loại phương pháp phổ biến của Box-Jenkins ARIMA để dự báo chuỗi thời gian.

### 3.3.3 Mô hình làm mịn hàm mũ kép “Double exponential smoothing”.

Làm mịn theo cấp số nhân kép là một phần mở rộng của làm mịn theo cấp số nhân (exponential smoothing), bổ sung hỗ trợ một cách rõ ràng cho các xu hướng trong chuỗi thời gian đơn biến. Phương pháp này đơn giản là cách sử dụng đệ quy làm trơn hai lần theo cấp số nhân.

Ngoài tham số alpha ( $\alpha$ ) để kiểm soát hệ số làm mịn cho cấp độ, một hệ số làm mịn bổ sung được thêm vào để kiểm soát sự suy giảm ảnh hưởng của sự thay đổi trong xu hướng được gọi là beta ( $\beta$ ).

Công thức thực hiện như sau:

$$y = \alpha x_t + (1 - \alpha)(y_{t-1} + b_{t-1})$$

$$b_t = \beta(y_t - y_{t-1}) + (1 - \beta)b_{t-1}$$

Alpha ( $\alpha$ ): Hệ số làm mịn cho cấp độ.

Beta ( $\beta$ ): Yếu tố làm mịn xu hướng.

Ngoài tham số  $\alpha$  là hệ số làm mịn nhận các giá trị trong khoảng từ 0 đến 1 (tương tự như mô hình làm mịn theo cấp số nhân), mô hình này sử dụng thêm tham số  $\beta$  là tham số làm mượt xu hướng, nó nhận các giá trị trong khoảng từ 0 đến 1.

Chương trình thực nghiệm

### Dữ liệu:

Dữ liệu là giá của mã chứng khoán VNM của công ty cổ phần Vinamilk từ năm 2014 đến năm 2018 được niêm yết trên thị trường chứng khoán Việt Nam.

Dữ liệu được tải về từ trang [cophieu68.vn](http://cophieu68.vn) dưới dạng file CSV.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
1	TICKER	FIGI	DATE	TYPE	FREQUENCY	OPEN	HIGH	LOW	CLOSE	VOLUME	ADJ_OPEN	ADJ_HIGH	ADJ_LOW	ADJ_CLOSE	ADJ_VOLL	ADJ_FACTOR	EX_DIVIDEN	SPLIT_RATIO	
2	GF	BBG000C3	6/5/2018	Intraday	daily	135.88	139.69	134.72	137.44	879160	175	179.9	173.5	177	205000			0	1
3	GF	BBG000C3	6/4/2018	EOD	daily	130.88	134.88	130.11	134.72	965230	170	175.2	169	175	0			0	1
4	GF	BBG000C3	6/1/2018	EOD	daily	127.8	130.26	127.41	129.49	1051610	166	169.2	165.5	168.2	90000			0	1
5	GF	BBG000C3	5/31/2018	EOD	daily	127.41	127.79	125.87	126.41	2231710	165.5	166	163.5	164.2	0			0	1
6	GF	BBG000C3	5/30/2018	EOD	daily	129.34	129.42	127.11	127.41	491920	168	168.1	165.1	165.5	100000			0	1
7	GF	BBG000C3	5/29/2018	EOD	daily	125.88	130.49	124.72	129.8	703120	163.5	169.5	162	168.6	0			0	1
8	GF	BBG000C3	5/25/2018	EOD	daily	127.03	127.42	124.72	127.03	464830	165	165.5	162	165	2329980			0	1
9	GF	BBG000C3	5/24/2018	EOD	daily	128.56	129.33	127.79	127.8	512860	167	168	166	166	50000			0	1
10	GF	BBG000C3	5/23/2018	EOD	daily	130.11	130.42	128.42	129.33	807290	169	169.4	166.8	168	0			0	1
11	GF	BBG000C3	5/22/2018	EOD	daily	123.26	130.49	123.26	128.8	971280	160.1	169.5	160.1	167.3	6340530			0	1
12	GF	BBG000C3	5/21/2018	EOD	daily	127.03	127.8	123.18	124.72	1185350	165	166	160	162	0			0	1
13	GF	BBG000C3	5/18/2018	EOD	daily	132.41	132.41	127.1	127.8	1042490	172	172	165.1	166	0	0.964035	0.7148	1	1
14	GF	BBG000C3	5/17/2018	EOD	daily	127.8	131.73	127.49	131.72	1153810	166	171.1	165.6	171.1	160000			0	1
15	GF	BBG000C3	5/16/2018	EOD	daily	130.1	130.49	127.02	127.03	1028400	169	169.5	165	165.5	0			0	1
16	GF	BBG000C3	5/15/2018	EOD	daily	135.49	136.34	130.18	130.18	1644010	176	177.1	169.1	169.1	0			0	1
17	GF	BBG000C3	5/14/2018	EOD	daily	138.57	138.95	135.49	135.49	1240740	180	180.5	176	176	0			0	1
18	GF	BBG000C3	5/11/2018	EOD	daily	137.8	138.57	135.49	137.03	575000	179	180	176	178	0			0	1
19	GF	BBG000C3	5/10/2018	EOD	daily	138.57	139.34	136.26	137.03	1006620	180	181	177	178	52000			0	1
20	GF	BBG000C3	5/9/2018	EOD	daily	143.19	143.19	137.8	138.57	451010	186	186	179	180	146000			0	1
21	GF	BBG000C3	5/8/2018	EOD	daily	142.42	143.19	141.65	142.42	555260	185	186	184	185	46030			0	1
22	GF	BBG000C3	5/7/2018	EOD	daily	146.65	146.65	142.5	142.5	514230	190.5	190.5	185.1	185.1	100000			0	1
23	GF	BBG000C3	5/4/2018	EOD	daily	142.42	146.65	142.03	146.27	771810	185	190.5	184.5	190	0			0	1
24	GF	BBG000C3	5/3/2018	EOD	daily	142.27	142.42	141.42	141.65	735730	184.8	185	183.7	184	60000			0	1
25	GF	BBG000C3	5/2/2018	EOD	daily	143.96	143.96	141.66	142.42	878320	187	187	184	185	220360			0	1
26	GF	BBG000C3	5/1/2018	EOD	daily	143.19	143.19	142.04	142.04	619230	186	186	184.5	184.5	632930			0	1
27	GF	BBG000C3	4/30/2018	EOD	daily	140.42	145.5	140.11	142.42	997890	182.4	189	182	185	183750			0	1
28	GF	BBG000C3	4/27/2018	EOD	daily	140.03	140.03	138.57	139.34	959520	181.9	181.9	180	181	589050			0	1
29	GF	BBG000C3	4/26/2018	EOD	daily	137.8	139.34	135.57	138.57	1469750	179	181	176.1	180	400000			0	1
30	GF	BBG000C3	4/25/2018	EOD	daily	142.42	143.11	140.11	140.42	636940	185	185.9	182	182.4	595450			0	1
31	GF	BBG000C3	4/24/2018	EOD	daily	143.97	143.97	142.43	143.19	739700	187	187	185	186	109550			0	1
32	GF	BBG000C3	4/23/2018	EOD	daily	144.35	144.97	143.5	144.35	845170	187.5	188.3	186.4	187.5	159110			0	1
33	GF	BBG000C3	4/20/2018	EOD	daily	142.42	144.35	141.89	144.35	756500	185	187.5	184.3	187.5	96910			0	1

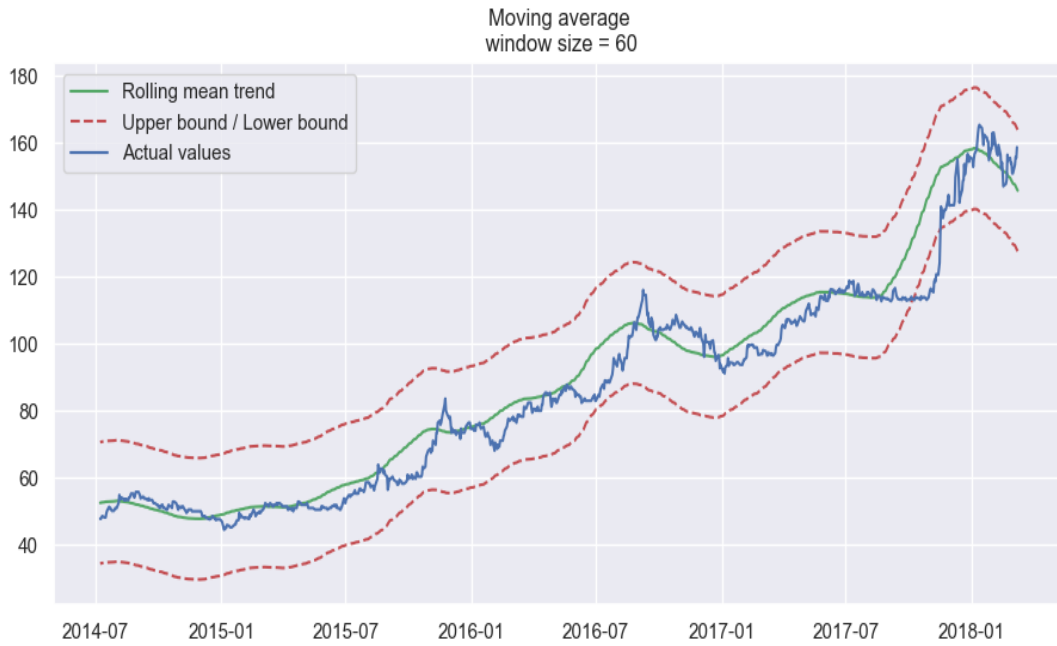
Hình 3.3-2 Minh họa bảng dữ liệu

Công cụ: Ngôn ngữ lập trình python 3. 8. 5

Môi trường thực nghiệm: windows 10 version(1909)

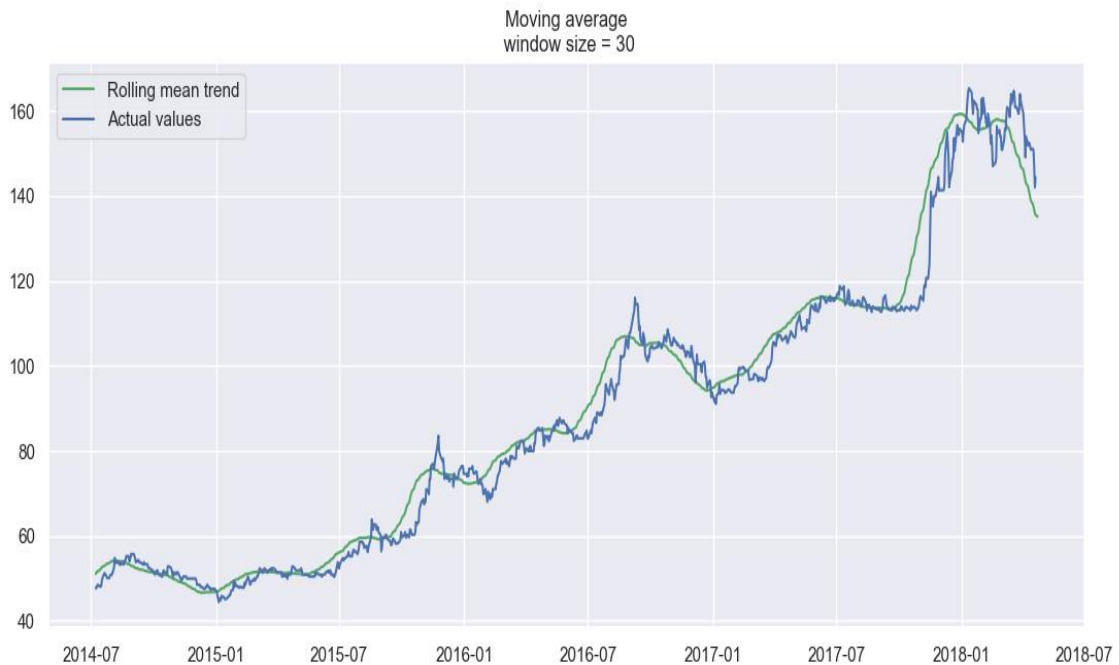
### 3.3.4 Kết quả mô hình trượt đơn Moving Average (MA)

**Kết quả được thực nghiệm với kích thước cửa sổ trượt =60**



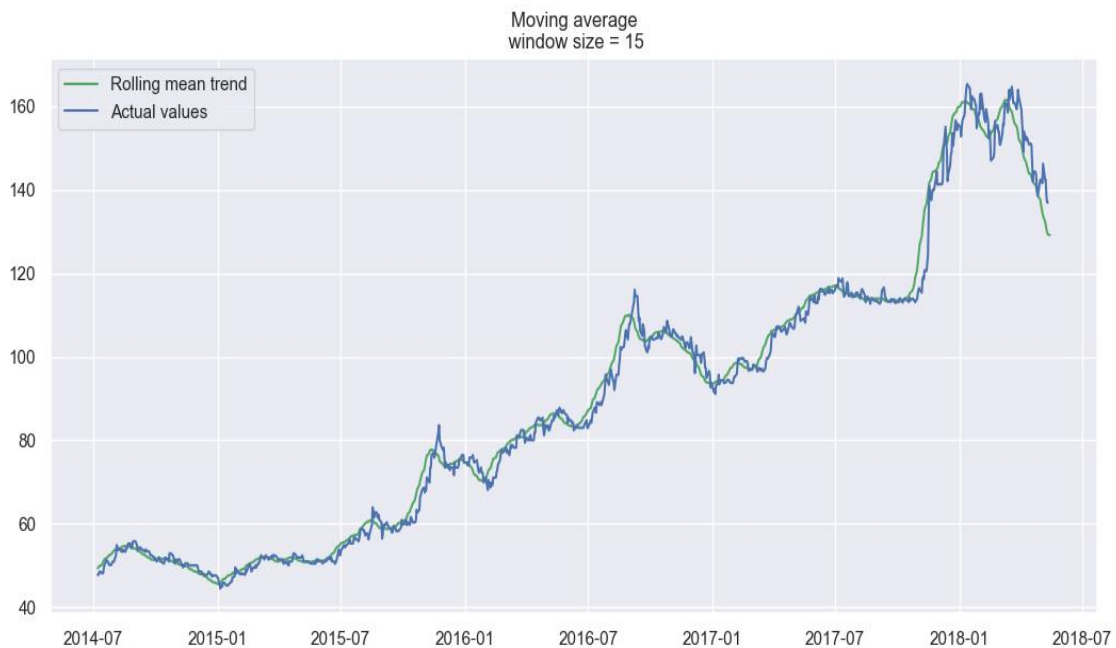
Hình 3.3-3 Kết quả thực nghiệm mô hình Moving Average kích thước cửa sổ trượt là 60

**Kết quả được thực nghiệm với kích thước cửa sổ trượt =30**



Hình 3.3-4 Kết quả thực nghiệm mô hình Moving Average kích thước cửa sổ trượt là 30

## Kết quả được thực nghiệm với kích thước cửa sổ trượt =15



Hình 3.3-5 Kết quả thực nghiệm mô hình Moving Average kích thước cửa sổ trượt là 15

### Chú thích đồ thị:

Rolling mean trend: Xu hướng trung bình(giá trị giá dự đoán)

Actual values: Giá trị giá thực tế

Upper bound/Lower bound: Giá trị giá cao nhất/giá trị giá thấp nhất

Windows size: Cửa sổ trượt (giá trị trung bình giá của các ngày trước đó)

### Nhận xét:

Qua 3 đồ thị áp dụng mô hình Moving Average với 3 kích thước cửa sổ trượt lần lượt là 60, 30, 15 đã cho thấy sự thay đổi rõ ràng về dự báo giá chứng khoán khi kích thước cửa sổ trượt thay đổi.

Khi áp dụng mô hình Moving Average thì kích thước cửa sổ trượt (Windows size) càng lớn thì độ chênh lệch với thực tế càng lớn, và ngược lại.

### 3.3.5 Kết quả thực nghiệm mô hình làm mịn theo cấp số nhân (exponential smoothing).



Hình 3.3-6 Kết quả thực nghiệm mô hình làm mịn theo cấp số nhân

#### Chú thích đồ thị:

Alpha 0. 05: Giá trị  $\alpha = 0. 05$

Alpha 0. 3: Giá trị  $\alpha = 0. 3$

Actual : Giá trị thực tế

#### Nhận xét:

Khi áp dụng mô hình (exponential smoothing) làm mịn theo cấp số nhân vào bài toán qua đồ thị trên có thể thấy khi giá trị  $\alpha = 0. 3$  thì giá trị dự đoán gần với giá trị thực tế hơn khi giá trị  $\alpha = 0. 05$ .

### 3.3.6 Kết quả thực nghiệm mô hình làm mịn hàm mũ kép “Double exponential smoothing”.



Hình 3.3-7 Kết quả thực nghiệm mô hình làm mịn hàm mũ kép

#### Chú thích đồ thị:

Alpha 0.9 beta 0.9: Giá trị  $\alpha = 0.9, \beta = 0.9$

Alpha 0.9 beta 0.02: Giá trị  $\alpha = 0.9, \beta = 0.02$

Alpha 0.02 beta 0.9: Giá trị  $\alpha = 0.02, \beta = 0.9$

Alpha 0.9 beta 0.02: Giá trị  $\alpha = 0.9, \beta = 0.02$

Actual: Giá trị thực tế

#### Nhận xét:

Sự thay đổi rõ ràng nhất khi áp dụng mô hình (Double exponential smoothing) là khi hai tham số Alpha ( $\alpha = 0.02$ ) và beta ( $\beta = 0.02$ ) đạt cực tiểu (nhỏ nhất có thể) thì giá trị đã được làm mịn tối đa (giảm xu hướng).



Và khi Alpha ( $\alpha = 0.9$ ) và beta ( $\beta = 0.02$ ) đã cho ra giá trị độ sai lệch ít nhất (gần nhất) với giá trị thực tế.

### 3.3.7 Tổng kết

Qua 3 mô hình và 5 đồ thị ở trên mỗi một mô hình đều có ưu nhược điểm riêng.

**Moving Average (MA)** mô hình trượt đơn: kết quả của mô hình này phụ thuộc rất nhiều vào dữ liệu đầu vào trước đó. Chỉ nên áp dụng phương pháp này khi số giá trị quan sát được là ít và tập dữ liệu có tính ổn định theo thời gian.

**Exponential smoothing** mô hình làm mịn theo cấp số nhân : kết quả của mô hình này cho thấy giá trị alpha ( $\alpha$ ) càng lớn thì độ sai lệch càng ít, giá trị dự báo càng gần so với giá trị thực tế hơn và Alpha thường được đặt thành giá trị từ 0 đến 1. Giá trị lớn thì mô hình chủ yếu chú ý đến những quan sát gần đây nhất trong quá khứ, trong khi giá trị nhỏ hơn thì lịch sử sẽ được tính đến nhiều hơn khi đưa ra dự đoán.

**Double exponential smoothing** mô hình làm mịn hàm mũ kép: Phương pháp này cho thấy ưu thế hơn so với phương pháp Moving Average (MA) mô hình trượt đơn đối với những bài toán dữ liệu lớn, vì khi dữ liệu đủ lớn nó có thể đi theo một xu hướng (tăng hoặc giảm), khi đó tham số beta  $\beta$  có thể làm giảm xu hướng theo thời gian.

**Điểm chung:** Cả 3 mô hình đều có điểm chung là giá trị dự báo phụ thuộc vào giá trị quan sát được gần hay xa trong quá khứ, nếu càng gần thì độ chính xác dự báo càng cao và ngược lại.



## KẾT LUẬN

Sau thời gian tham khảo tìm tòi và dưới sự chỉ bảo của cô hướng dẫn về bài tập đồ án Tìm hiểu ngôn ngữ lập trình python, viết chương trình thử nghiệm dự báo sự biến động của giá chứng khoán trong khoảng thời gian nhất định dành cho việc thực hiện đề tài, nên một số vấn đề vẫn chưa được hoàn chỉnh. Tuy nhiên, đồ án đã đạt được một số kết quả.

Những kết quả đạt được trong đồ án:

- Tìm hiểu về ngôn ngữ python.
- Tìm hiểu môi trường lập trình PyCharm.
- Tìm hiểu về bài toán dự báo.
- Tìm hiểu về bài toán ứng dụng dự báo giá cổ phiếu.
- Viết chương trình thử nghiệm dự báo bài toán giá cổ phiếu.

Tuy nhiên trong quá trình thực hiện vẫn còn một số điều cần bổ sung như:

- Chưa thao tác thành thạo về lập trình python.
- Chưa hiểu sâu về bài toán dự báo.
- Chương trình thử nghiệm chưa được tối ưu hóa giao diện.

Trong thời gian tiếp theo, em sẽ cố gắng để ngày càng hoàn thiện, trau dồi kỹ năng lập trình trở nên tốt hơn, tìm hiểu về các phương pháp học máy hiện đại như mạng nơ ron, mạng nơ ron học sâu để thử nghiệm với các phương pháp dự báo cho kết quả thực hiện tốt hơn.

Em xin chân thành cảm ơn!

## DANH MỤC TÀI LIỆU THAM KHẢO

1. Sách tự học lập trình Python căn bản-NXB Đại Học Quốc Gia HCM
2. Phân tích chuỗi thời gian và dự báo “Nguyễn Thị Vinh – Đại học thủy lợi”
3. <https://www.w3schools.com/>
4. <https://youtube.com/howkteam>
5. Cophieu68.vn