

## Phụ lục

Phần phụ lục này tập hợp các bài viết độc lập hoặc nối tiếp các ngày trong nội dung chính của eBook. Đây là phần chia sẻ thêm để giúp bạn đọc có nhu cầu thực hành, khám phá thêm các góc nhìn mới về Phân tích dữ liệu nói riêng, AI nói chung.

Các bài viết có thể ở dạng ý tưởng và đang trong giai đoạn khám phá. Vì vậy nếu bạn đọc thấy chưa hoàn thiện thì đó xem như bài tập của các bạn nhé!

Thân ái,

## Chỉ số VN INDEX biến động như thế nào từ thứ Hai đến thứ Sáu

Tôi có câu hỏi là nên giao dịch chứng khoán vào thứ mấy trong tuần?

Để trả lời câu hỏi này thì chúng ta cần giải quyết một số vấn đề sau:

- Xem dữ liệu VNIndex giá trị High, Low các ngày trong tuần

Bài viết này minh họa bằng code R.

### Cách lấy dữ liệu VN INDEX

Vào trang web “<https://www.cophieu68.vn/export.php>”, đăng ký tài khoản (miễn phí) bấm vào mục “DỊCH VỤ DỮ LIỆU”, “Tải dữ liệu” để lấy file csv về máy.

KHO DỮ LIỆU - METASTOCK - EXCEL				
	DL lịch sử Metastock & AmiBroker	DL lịch sử Excel (Full)	DL Báo cáo tài chính	DL
ALL DATA (HOSE & HNX)	Download			
VNINDEX	Download	Download		
HNX	Download	Download		
000001.SS	Download	Download	Download	
A32	Download	Download	Download	
AAA	Download	Download	Download	
AAM	Download	Download	Download	

Dùng phần mềm R (<https://cran.r-project.org/>) để xem tổng quan dữ liệu.

Các dòng bắt đầu bằng dấu thăng (#) là chú thích của tôi để giải thích ý nghĩa của lệnh R.

```
# Dùng lệnh file.choose() để mở hộp thoại
# chọn file "excel_vnindex.csv" sau khi download
f = file.choose()
data = read.csv(f)
dim(data)
```

```
[1] 4506 14
```

# Kết quả lệnh dim cho thấy có 4506 dòng dữ liệu

# 14 cột dữ liệu (Xem thêm lệnh head bên dưới)

```
head(data)
```

```
  X.Ticker. X.DYYYYMMDD. X.OpenFixed. X.HighFixed. X.LowFixed. X.CloseFixed.
1 ^VNINDEX      20190426       972.86       979.64       970.73       979.64
2 ^VNINDEX      20190425       976.46       976.46       971.92       974.13
3 ^VNINDEX      20190424       969.66       978.71       969.66       976.92
4 ^VNINDEX      20190423       964.84       970.98       964.35       968.00
```

5	AVNINDEX	20190422	963.78	966.69	959.33	965.86		
6	AVNINDEX	20190419	968.27	971.73	965.46	966.21		
	X.Volume	X.Open	X.High	X.Low	X.Close	X.VolumeDeal	X.VolumeFB	X.VolumeFS
1	118539754	972.86	979.64	970.73	979.64	0	6527232	6355132
2	149114784	976.46	976.46	971.92	974.13	0	15370270	16830400
3	134071764	969.66	978.71	969.66	976.92	0	12740050	7683390
4	154887616	964.84	970.98	964.35	968.00	0	11690700	18355300
5	216344286	963.78	966.69	959.33	965.86	0	6673120	6090200
6	106976093	968.27	971.73	965.46	966.21	0	3204780	4536850

Chú ý: Cột ngày “X.DTYYYYMMDD.” được trình bày theo dạng yyyyMMdd – không có dấu cách giữa năm, tháng ngày nên R hiểu đây là số nguyên. Dùng lệnh class để xem kiểu dữ liệu:

```
class(data$X.DTYYYYMMDD.)
```

```
[1] "integer"
```

Chúng ta cần chuyển đổi dữ liệu thời gian một chút thông qua 2 bước:

**Bước 1:** Thêm cột strDate bằng cách lấy dữ liệu cột “X.DTYYYYMMDD.” chuyển thành kiểu kí tự (chuỗi).

```
data$strDate = as.character(data$X.DTYYYYMMDD.)
class(data$strDate)
```

```
[1] "character"
```

**Bước 2:** Thêm cột data bằng cách lấy dữ liệu cột “strDate” vừa thêm chuyển thành kiểu ngày bằng hàm as.Date(strDate, “%Y%m%d”)

```
data$date = as.Date(data$strDate, format = '%Y%m%d')
class(data$date)
```

```
[1] "Date"
```

Chú ý: trong định dạng %Y%d%m thì chữ **d** và **m** là chữ thường.

Kiểm tra lại vài dòng dữ liệu

```
head(data)
```

	X.Ticker.	X.DTYYYYMMDD.	X.OpenFixed.	X.HighFixed.	X.LowFixed.	X.CloseFixed.	X.Volume.
	X.Open.	X.High.	X.Low.	X.Close.	X.VolumeDeal.		
1	AVNINDEX	20190426	972.86	979.64	970.73	979.64	118539754
	972.86	979.64	970.73	979.64	0		
2	AVNINDEX	20190425	976.46	976.46	971.92	974.13	149114784
	976.46	976.46	971.92	974.13	0		
3	AVNINDEX	20190424	969.66	978.71	969.66	976.92	134071764
	969.66	978.71	969.66	976.92	0		
4	AVNINDEX	20190423	964.84	970.98	964.35	968.00	154887616
	964.84	970.98	964.35	968.00	0		
5	AVNINDEX	20190422	963.78	966.69	959.33	965.86	216344286
	963.78	966.69	959.33	965.86	0		
6	AVNINDEX	20190419	968.27	971.73	965.46	966.21	106976093
	968.27	971.73	965.46	966.21	0		
	X.VolumeFB.	X.VolumeFS.	strDate	date			
1	6527232	6355132	20190426	2019-04-26			
2	15370270	16830400	20190425	2019-04-25			
3	12740050	7683390	20190424	2019-04-24			
4	11690700	18355300	20190423	2019-04-23			

```
5 6673120 6090200 20190422 2019-04-22
6 3204780 4536850 20190419 2019-04-19
```

Lúc này dữ liệu cột date được hiển thị có dấu gạch giữa năm tháng và ngày.

Xem tổng quan dữ liệu bằng lệnh summary:

```
summary(data)
```

X.Ticker.	X.DTYYYYMMDD.	X.OpenFixed.	X.HighFixed.	X.LowFixed.	
X.CloseFixed.					
AVNINDEX:4506	Min. :20000728	Min. : 100.0	Min. : 100.0	Min. : 100.0	M
in. : 100.0	1st Qu.:20050926	1st Qu.: 296.5	1st Qu.: 298.9	1st Qu.: 296.3	1
st Qu.: 297.4	Median :20100412	Median : 487.6	Median : 490.6	Median : 484.0	M
edian : 486.9	Mean :20098088	Mean : 507.6	Mean : 510.1	Mean : 505.0	M
ean : 507.5	3rd Qu.:20141019	3rd Qu.: 610.7	3rd Qu.: 614.4	3rd Qu.: 607.8	3
rd Qu.: 610.6	Max. :20190426	Max. :1207.6	Max. :1211.3	Max. :1197.4	M
ax. :1204.3					
X.Volume.	X.Open.	X.High.	X.Low.	X.Close.	
X.VolumeDeal.					
Min. : 174	Min. : 100.0	Min. : 100.0	Min. : 100.0	Min. : 100.0	
Min. :0	1st Qu.: 296.5	1st Qu.: 298.9	1st Qu.: 296.3	1st Qu.: 297.3	
1st Qu.: 1487382	Median : 487.6	Median : 490.6	Median : 484.0	Median : 486.8	
Median : 26702355	Mean : 507.6	Mean : 510.1	Mean : 505.0	Mean : 507.5	
Median :0	3rd Qu.: 610.7	3rd Qu.: 614.4	3rd Qu.: 607.8	3rd Qu.: 610.7	
Mean : 53381363	Max. :1207.6	Max. :1211.3	Max. :1197.4	Max. :1204.3	
Mean :0					
3rd Qu.: 93457508					
3rd Qu.:0					
Max. :445940510					
Max. :0					
				NA's :1	
X.VolumeFB.	X.VolumeFS.	date			
Min. :0.000e+00	Min. :0.000e+00	Min. :2000-07-28			
1st Qu.:0.000e+00	1st Qu.:0.000e+00	1st Qu.:2005-09-26			
Median :2.540e+06	Median :2.140e+06	Median :2010-04-12			
Mean :1.614e+07	Mean :1.625e+07	Mean :2010-04-02			
3rd Qu.:6.608e+06	3rd Qu.:5.979e+06	3rd Qu.:2014-10-19			
Max. :2.147e+09	Max. :1.847e+09	Max. :2019-04-26			

Ghi chú:

- Trong cột ‘date’ cho biết dữ liệu từ ngày 28/7/2000 (dòng **Min.**) đến 26/4/2019 (Dòng **Max.**)

Thêm cột “dayOfWeek” để thể hiện Thứ trong tuần.

```
data$dayOfWeek = weekdays(date)
```

Chuyển dayOfWeek thành Factor để phục vụ cho việc phân tích

```
data$dayOfWeek = as.factor(data$dayOfWeek)
```

Xem lại tổng quan dữ liệu của dayOfWeek bằng lệnh summary bạn sẽ thấy số lượng dữ liệu theo thứ.

```
summary(data$dayOfWeek)
```

Friday 938	Monday 911	Thursday 862	Tuesday 854	wednesday 941
---------------	---------------	-----------------	----------------	------------------

## Vẽ biểu đồ với thư viện zoo

Zoo hỗ trợ phân tích dữ liệu theo thời gian.

Cài đặt thư viện zoo:

```
install.packages('zoo')
install.packages('ggfortify')
library(zoo)
library(ggfortify)
```

Tạo dữ liệu x theo thời gian của giá trị thấp nhất và cao nhất của VNIndex

```
z = zoo(x = cbind(X.Low., X.High.), order.by = date)
```

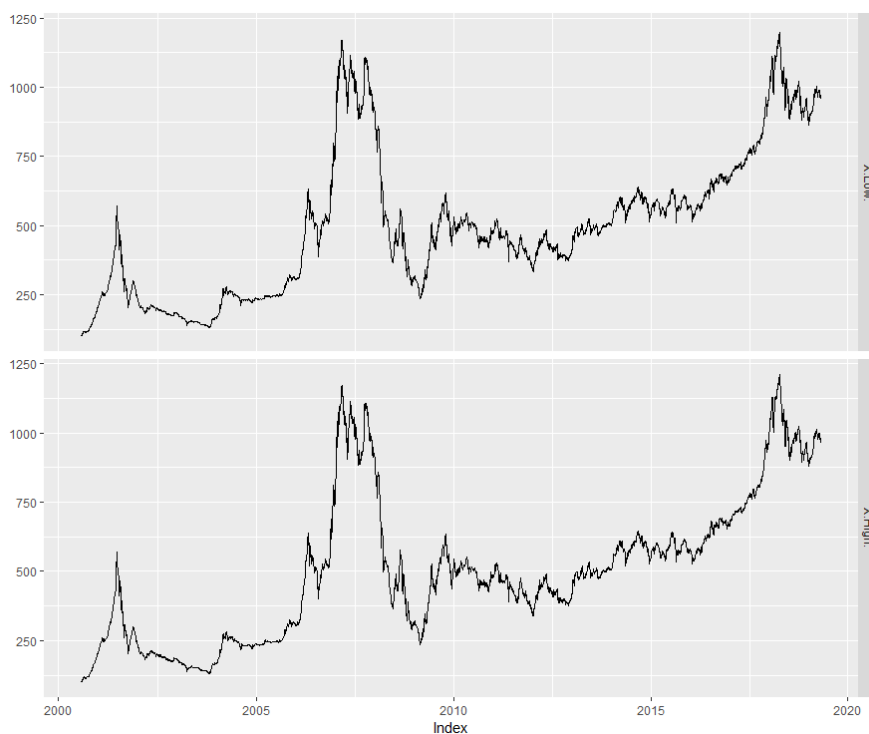
Nhìn qua dữ liệu của biến z theo thời gian

```
head(z)
```

	X.Low.	X.High.
2000-07-28	100.00	100.00
2000-07-31	101.55	101.55
2000-08-02	103.38	103.38
2000-08-04	105.20	105.20
2000-08-07	106.92	106.92
2000-08-09	108.64	108.64

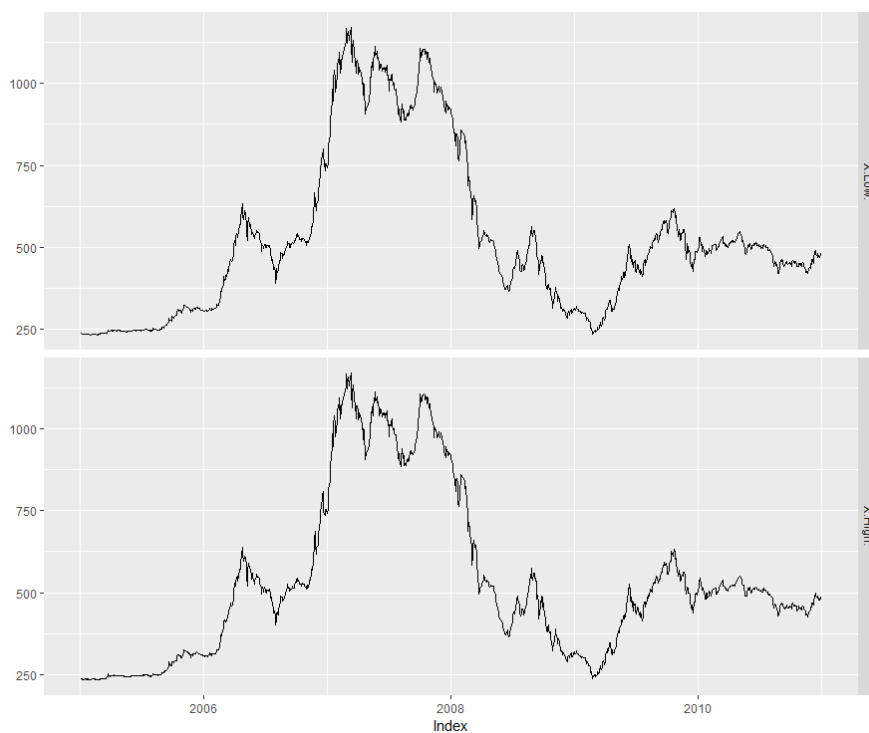
Vẽ biểu đồ VNIndex thấp nhất và cao nhất theo ngày

```
autoplot(z)
```



Nhìn vào dữ liệu giữa năm 2005 và 2010 thì có núi bất thường? Để xem chi tiết dữ liệu từ năm 2005 đến 2010 thì dùng lệnh window và vẽ biểu đồ:

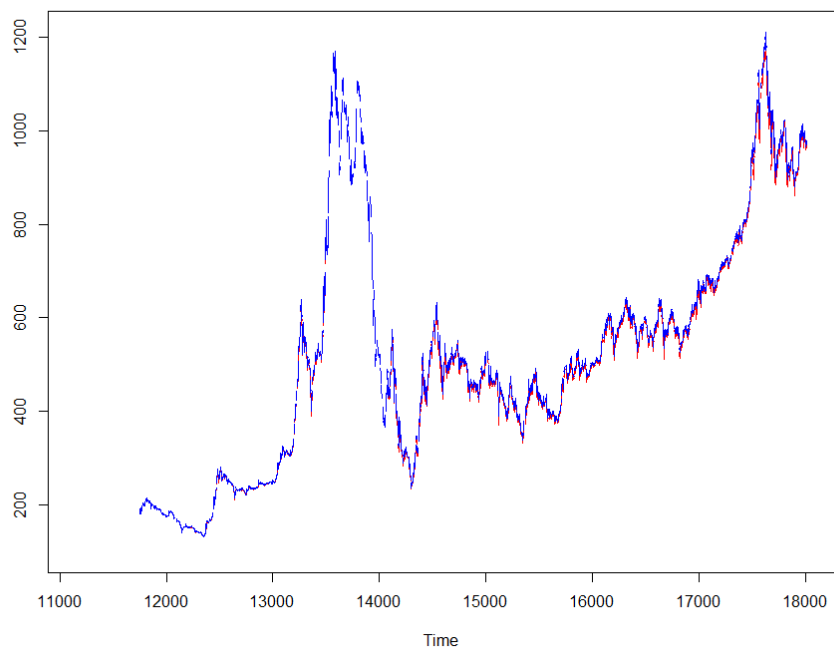
```
z1 = window(z, start = as.Date('2005/1/1'), end =
as.Date('2010/12/31'))
autoplot(z1)
```



Năm 2008 là năm khủng hoảng kinh tế nên chứng khoán lao dốc.

Xem giá trị Cao nhất và Thấp nhất trong cùng 1 biểu đồ

```
ts.plot(z, col = c("red", "blue"))
```



## Sử dụng PerformanceAnalytics

```
install.packages('PerformanceAnalytics')  
library('PerformanceAnalytics')  
PerformanceAnalytics::chart.TimeSeries(z)
```



### Gom dữ liệu theo tuần

Chúng ta cần bảng số liệu sau VNIndex High như sau:

	Mon	Tue	Wed	Thu	Fri
W1					
W2					
W3					

### Sử dụng R để vẽ biểu đồ

Để xem được số liệu và biểu đồ liên quan đến chỉ số VN Index trước và sau lễ 30/4 thì cần lọc dữ liệu trước và sau ngày 30 tháng 4. Dữ liệu trong Bảng 1 ở trên có thể download tại:

“[https://drive.google.com/open?id=1S3sf6YRT3Jt6a7U0n\\_I4mCCzKq6dqGkw](https://drive.google.com/open?id=1S3sf6YRT3Jt6a7U0n_I4mCCzKq6dqGkw)”

Trong bài này dùng thư viện “ggplot2” để vẽ biểu đồ.

Trong R, dùng lệnh `install.packages(...)` để cài thư viện.

```
install.packages('ggplot2')
```

Các lệnh R sau đây sẽ xử lý một chút dữ liệu từ file csv và vẽ một số biểu đồ:

```
# Chọn file csv sau khi download
f = file.choose()
```



```
data = read.csv(f)

data$strDate = as.character(data$DTYYYYMMDD)
data$date = as.Date(data$strDate, format = '%Y%m%d')
# Xóa cột strDate
data$strDate = NULL
data$year = as.numeric(format(data$date, "%Y"))
data$month = as.factor(format(data$date, "%m"))
data$yyyymm = as.factor(format(data$date, "%Y-%m"))
data$m = as.numeric(format(data$date, "%m"))

library(ggplot2)
attach(data)
p = ggplot(data, aes(x = yyyymm, y = Close, fill = month))

p1 = p + geom_bar(stat="identity") + xlab("Ngày giao dịch trước và sau  
Lễ 30/4") + ylab("Giá đóng cửa")
p1 = p1 + theme(axis.text.x = element_text(angle = 90))
p1 = p1 + ggtitle("Giá đóng cửa chỉ số VN Index trước và sau lễ 30/4  
trong 10 năm")
p1 = p1 + labs(fill = "Tháng")

plot(p1)

# Giá mở cửa
p = ggplot(data, aes(x = yyyymm, y = Open, fill = month))

p1 = p + geom_bar(stat="identity") + xlab("Ngày giao dịch trước và sau  
Lễ 30/4") + ylab("Giá mở cửa")
p1 = p1 + theme(axis.text.x = element_text(angle = 90))
p1 = p1 + ggtitle("Giá mở cửa chỉ số VN Index trước và sau lễ 30/4  
trong 10 năm")
p1 = p1 + labs(fill = "Tháng")

plot(p1)

# Giá cho nhất
p = ggplot(data, aes(x = yyyymm, y = High, fill = month))
```

```
p1 = p + geom_bar(stat="identity") + xlab("Ngày giao dịch trước và sau  
Lễ 30/4") + ylab("Giá cao nhất")  
p1 = p1 + theme(axis.text.x = element_text(angle = 90))  
p1 = p1 + ggtitle("Giá cao nhất chỉ số VN Index trước và sau lễ 30/4  
trong 10 năm")  
p1 = p1 + labs(fill = "Tháng")  
  
plot(p1)
```

**Tham khảo**

<https://cran.r-project.org/web/packages/timeSeries/vignettes/timeSeriesPlot.pdf>

Tp.HCM, ngày 1/5/2019

## Quan sát giao dịch cổ phiếu VNM (Vinamilk)

Bài viết này minh họa bằng code Python.

### Đọc dữ liệu

Để lấy dữ liệu thì tôi tự viết phần mềm để sưu tầm giao dịch theo lô của cổ phiếu VNM từ trang [cafef.vn](http://cafef.vn) và lưu trên link:

[https://thachln.github.io/datasets/VNM\\_20200710.zip](https://thachln.github.io/datasets/VNM_20200710.zip).

Dữ liệu minh họa trong bài viết này được tập hợp từ ngày 10/27/2014 đến 10/7/2020.

```
import pandas as pd
df =
pd.read_csv('https://thachln.github.io/datasets/VNM_20200710.zip')
```

### Hiểu một chút về dữ liệu

#### Dùng hàm `info()`

```
df.info()
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 674488 entries, 0 to 674487
Data columns (total 4 columns):
#   Column  Non-Null Count  Dtype
---  ---
0   symbol  674488 non-null   object
1   time    674488 non-null   object
2   price   674488 non-null   float64
3   volume  674488 non-null   int64
dtypes: float64(1), int64(1), object(2)
```

Kết quả `info` cho thấy cột `time` có kiểu dữ liệu `object` chứ không phải là thời gian (`datetime`). Để chuyển kiểu cột `time` cho đúng kiểu thời gian thì sử dụng tiếp lệnh sau:

```
df['time'] = pd.to_datetime(df['time'])
```

Chạy lại lệnh `info` ở trên sẽ cho ra kết quả như sau:

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 674488 entries, 0 to 674487
Data columns (total 4 columns):
#   Column  Non-Null Count  Dtype
---  ---
0   symbol  674488 non-null   object
1   time    674488 non-null   datetime64[ns]
2   price   674488 non-null   float64
3   volume  674488 non-null   int64
dtypes: datetime64[ns](1), float64(1), int64(1), object(1)
```

Hàm `info()` cho thấy dataframe gồm 4 cột dữ liệu:

Cột	Ý nghĩa
-----	---------

<b>symbol</b>	là mã cổ phiếu. Trong dữ liệu này chỉ có cổ phiếu VNM
<b>time</b>	thời gian giao dịch. Kiểu dữ liệu là <code>datetime64[ns]</code>
<b>price</b>	giá giao dịch
<b>volume</b>	số lượng cổ phiếu được giao dịch

### Dùng hàm `describe()` và thuộc tính `shape`

Xem vài thông tin thống kê về dữ liệu bằng hàm `describe()`:

```
df.describe()
```

	price	volume
count	674488.000000	6.744880e+05
mean	137.262721	1.565956e+03
std	28.859136	2.549325e+04
min	83.700000	1.000000e+00
25%	117.800000	8.000000e+01
50%	132.000000	3.200000e+02
75%	151.400000	1.040000e+03
max	215.000000	1.887654e+07

Xem thêm thuộc tính `shape`:

```
df.shape
```

```
(674488, 4)
```

Như vậy có thể tóm tắt vài chỉ số thống kê như sau:

- Dữ liệu có 674488 dòng, mỗi dòng là mỗi giao dịch.
  - Giá trị mean (trung bình) cho thấy giá trung bình của cổ phiếu VNM là 138 nghìn đồng. Trung bình mỗi giao dịch  $1.62 \times 10^3 = 1620$  cổ phiếu.
  - Độ lệch chuẩn của giá cổ phiếu là 30. Chú ý giá cổ phiếu ở đây tính bằng đơn vị là **Nghìn** đồng. Điều này nói lên điều gì? Nó phản ánh sự khác biệt về giá trong các lần giao dịch. Các mức giá giao dịch có sự khác biệt nhau tầm 30 nghìn đồng xung quanh giá trung bình.
- Ở đây phải chú ý là chúng ta không có dữ liệu về các lần chia tách cổ phiếu. Mỗi lần chia tách thì giá cổ phiếu được điều chỉnh lại. Tạm thời bỏ qua yếu tố này để cho “bài tập thể dục” đơn giản.
- Tương tự, bạn có thể nhìn qua các chỉ số min; max; bách phân vị 25%, 50%, 75% của giá.

### Xem vài dòng dữ liệu

```
df.head()
```

symbol	time	price	volume
--------	------	-------	--------

0	VNM	2020-07-10	14:47:03	115.3	16660
1	VNM	2020-07-10	14:30:03	115.4	1000
2	VNM	2020-07-10	14:30:01	115.4	450
3	VNM	2020-07-10	14:29:45	115.4	660
4	VNM	2020-07-10	14:29:35	115.4	200

### Thêm cột ngày

Hiện tại cột time chứa thời gian giao dịch đến mức giây. Các phân tích tiếp theo của chúng ta là tính theo ngày nên cần phải thêm cột date để chứa ngày tháng năm.

```
df['date'] = df['time'].dt.date
df.head()
```

	symbol		time	price	volume	date
0	VNM	2020-07-10	14:47:03	115.3	16660	2020-07-10
1	VNM	2020-07-10	14:30:03	115.4	1000	2020-07-10
2	VNM	2020-07-10	14:30:01	115.4	450	2020-07-10
3	VNM	2020-07-10	14:29:45	115.4	660	2020-07-10
4	VNM	2020-07-10	14:29:35	115.4	200	2020-07-10

### Tính tổng giá trị giao dịch

```
df['trade value'] = df['price'] * df['volume']
df.head()
```

	symbol		time	price	volume	date	trade_value
0	VNM	2020-07-10	14:47:03	115.3	16660	2020-07-10	1920898.0
1	VNM	2020-07-10	14:30:03	115.4	1000	2020-07-10	115400.0
2	VNM	2020-07-10	14:30:01	115.4	450	2020-07-10	51930.0
3	VNM	2020-07-10	14:29:45	115.4	660	2020-07-10	76164.0
4	VNM	2020-07-10	14:29:35	115.4	200	2020-07-10	23080.0

### Tính giá trị trung bình của cổ phiếu

Giá trị trung bình trong ngày bằng cách tính tổng các giá trị giao dịch trong ngày. Sau đó chia cho tổng lượng giao dịch trong ngày. Kết quả lưu trong dataframe mới df\_avg\_price.

```
df_avg_price = df.groupby(['date'])['volume', 'trade_value'].sum()
df_avg_price['avg_price'] = df_avg_price['trade_value'] /
df_avg_price['volume']

df_avg_price.head()
```

	date	volume	trade_value	avg_price
0	2014-10-27	68880	7223620.0	104.872532
1	2014-10-29	27810	2892260.0	104.000719
2	2014-10-30	49530	5197670.0	104.939834
3	2014-11-03	20410	2155160.0	105.593337
4	2014-11-05	125160	13045320.0	104.229147

Xem thông tin của dataframe df\_avg\_price:

```
df_avg_price.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 1318 entries, 2014-10-27 to 2020-07-10
Data columns (total 3 columns):
#   Column          Non-Null Count  Dtype
---  ---
0   volume          1318 non-null   int64
1   trade_value     1318 non-null   float64
2   avg_price       1318 non-null   float64
dtypes: float64(2), int64(1)
```

Bạn để ý thì thấy dataframe `df_avg_price` không có cột `date`.

### *Thêm cột gom nhóm vào dataframe*

Để thêm cột làm tiêu chí gom nhóm (cột `date`) vào dataframe thì dùng hàm `reset_index()`:

```
df_avg_price = df_avg_price.reset_index()
```

Xem lại thông tin của dataframe `df_avg_price`:

```
df_avg_price.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1318 entries, 0 to 1317
Data columns (total 4 columns):
#   Column          Non-Null Count  Dtype
---  ---
0   date            1318 non-null   object
1   volume          1318 non-null   int64
2   trade_value     1318 non-null   float64
3   avg_price       1318 non-null   float64
dtypes: float64(2), int64(1), object(1)
```

Một điểm chú ý là cột `date` có kiểu dữ liệu là `object`.

### *Chuyển kiểu object sang dạng date*

Để chuyển cột `date` đang là `object` sang kiểu thời gian thì dùng hàm `pd.to_datetime(...)`:

```
df_avg_price['date'] = pd.to_datetime(df_avg_price['date'])
```

Xem lại thông tin:

```
df_avg_price.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1318 entries, 0 to 1317
Data columns (total 4 columns):
#   Column          Non-Null Count  Dtype
---  ---
0   date            1318 non-null   datetime64[ns]
1   volume          1318 non-null   int64
2   trade_value     1318 non-null   float64
3   avg_price       1318 non-null   float64
dtypes: datetime64[ns](1), float64(2), int64(1)
```

Xem vài dòng dữ liệu:

```
df_avg_price.head()
```

	date	volume	trade_value	avg_price
0	2014-10-27	68880	7223620.0	104.872532
1	2014-10-29	27810	2892260.0	104.000719
2	2014-10-30	49530	5197670.0	104.939834
3	2014-11-03	20410	2155160.0	105.593337
4	2014-11-05	125160	13045320.0	104.229147

### Sắp xếp lại dataframe theo ngày giảm dần

```
df_avg_price = df_avg_price.sort_values(by = ['date'],
ascending=False)
df_avg_price.head()
```

	date	volume	trade_value	avg_price
1317	2020-07-10	677180	78415689.0	115.797408
1316	2020-07-09	1560410	181141529.0	116.085855
1315	2020-07-08	551310	63805697.0	115.734699
1314	2020-07-07	1024960	119346196.0	116.439857
1313	2020-07-06	1319980	152240966.0	115.335813

### Xem vài thông tin mô tả dataframe mới

```
df_avg_price.describe()
```

	volume	trade_value	avg_price
count	1.318000e+03	1.318000e+03	1318.000000
mean	8.013798e+05	1.112912e+08	136.770054
std	8.625206e+05	1.238176e+08	26.715244
min	2.913000e+03	3.612120e+05	84.764400
25%	3.382375e+05	4.410212e+07	119.588663
50%	6.499100e+05	8.893423e+07	133.040233
75%	1.060290e+06	1.503052e+08	148.547710
max	1.907396e+07	2.733192e+09	214.305513

Hãy quan sát vài chỉ số của cột bên trái đối với giá trung bình của cổ phiếu (cột avg\_price)!

### Tính chênh lệch giá giữa 2 ngày liền kề

```
df_avg_price['delta'] = df_avg_price['avg_price'].diff(periods = -1)
df_avg_price.head()
```

date	volume	trade_value	avg_price	delta
2020-07-10	677180	78415689.0	115.797408	-0.288447
2020-07-09	1560410	181141529.0	116.085855	0.351156
2020-07-08	551310	63805697.0	115.734699	-0.705158
2020-07-07	1024960	119346196.0	116.439857	1.104045
2020-07-06	1319980	152240966.0	115.335813	1.370475

### Thêm cột kí hiệu giá tăng hay giảm

Thêm cột pn (Positive or Negative) để ghi chú giá trung bình của cổ phiếu là tăng (1) hay giảm (-1) hay bằng (0) so với ngày hôm trước.

Giả định nếu giá tăng chưa tới 1 đồng thì xem như không tăng. Code bên dưới sẽ thêm cột pn với giá trị là 0 (xem như giá cổ phiếu không tăng so với ngày hôm trước).

Sau đó thiết lập lại giá trị nếu tăng trên 0.009 nghìn thì thiết lập cột pn là 1. Ngược lại nếu giảm hơn 0.009 nghìn thì thiết lập cột pn là -1.

```
df_avg_price['pn'] = 0
df_avg_price.loc[df_avg_price['delta'] > 0.009, 'pn'] = 1
df_avg_price.loc[df_avg_price['delta'] < -0.009, 'pn'] = -1
```

Xem thử kết quả

```
df_avg_price.head()
```

date	volume	trade_value	avg_price	delta	pn
2020-07-10	677180	78415689.0	115.797408	-0.288447	-1
2020-07-09	1560410	181141529.0	116.085855	0.351156	1
2020-07-08	551310	63805697.0	115.734699	-0.705158	-1
2020-07-07	1024960	119346196.0	116.439857	1.104045	1
2020-07-06	1319980	152240966.0	115.335813	1.370475	1

Đến đây thì trong tay của bạn đã có dữ liệu giá trung bình cổ phiếu VNM mỗi ngày và cột delta, pn cho biết sự chênh lệch giá giữa hai ngày liên tiếp, cụ thể là tăng so với ngày hôm trước (cột pn có giá trị 1) hoặc giảm so với ngày hôm trước (cột pn có giá trị -1).

Xem lại các chỉ số thống kê của dataframe df\_avg\_price:

```
df_avg_price.describe()
```

	volume	trade_value	avg_price	delta	pn
count	1.318000e+03	1.318000e+03	1318.000000	1317.000000	1318.000000
mean	8.013798e+05	1.112912e+08	136.770054	0.008295	-0.015175
std	8.625206e+05	1.238176e+08	26.715244	2.653074	0.997223
min	2.913000e+03	3.612120e+05	84.764400	-29.834159	-1.000000
25%	3.382375e+05	4.410212e+07	119.588663	-0.871813	-1.000000
50%	6.499100e+05	8.893423e+07	133.040233	-0.020519	-1.000000
75%	1.060290e+06	1.503052e+08	148.547710	1.052080	1.000000
max	1.907396e+07	2.733192e+09	214.305513	21.134308	1.000000

### Thống kê thử số ngày tăng, số ngày giảm

Thống kê thử số ngày tăng, số ngày giảm bằng hàm crosstab (...):

```
pn_count = pd.crosstab(index=df_avg_price['pn'], columns='count')
print(pn_count)
```

col_0	count
pn	
-1	665
0	8
1	645

Số ngày tăng (645) không khác biệt lắm so với số ngày giảm (665).

### Thêm cột ngày trong tuần

Thêm cột day cho 2 dataframe df và df\_avg\_price:



```
df['day'] = df['time'].dt.dayofweek
df_avg_price['day'] = df_avg_price['date'].dt.dayofweek
```

Xem dữ liệu của dataframe df:

```
df.head()
```

	symbol	time	price	volume	date	day	trade_value
0	VNM	2020-07-10 14:47:03	115.3	16660	2020-07-10	4	1920898.0
1	VNM	2020-07-10 14:30:03	115.4	1000	2020-07-10	4	115400.0
2	VNM	2020-07-10 14:30:01	115.4	450	2020-07-10	4	51930.0
3	VNM	2020-07-10 14:29:45	115.4	660	2020-07-10	4	76164.0
4	VNM	2020-07-10 14:29:35	115.4	200	2020-07-10	4	23080.0

Xem dữ liệu của dataframe df\_avg\_price:

```
df_avg_price.head()
```

	date	volume	trade_value	avg_price	delta	pn	day
1317	2020-07-10	677180	78415689.0	115.797408	-0.288447	-1	4
1316	2020-07-09	1560410	181141529.0	116.085855	0.351156	1	3
1315	2020-07-08	551310	63805697.0	115.734699	-0.705158	-1	2
1314	2020-07-07	1024960	119346196.0	116.439857	1.104045	1	1
1313	2020-07-06	1319980	152240966.0	115.335813	1.370475	1	0

### Đếm số lượng các thứ trong tuần

Sử dụng hàm `crosstab()`:

```
pd.crosstab(index=df['day'], columns='count')
```

col_0	count
day	
0	134510
1	138445
2	135880
3	128702
4	136951

Tra lịch ngày 10/7/2020 là thứ Sáu, thuộc tính `dayofweek` của cột `date` cho giá trị là 4 (cột `day`)

July 2020						
Su	Mo	Tu	We	Th	Fr	Sa
28	29	30	1	2	3	4
5	6	7	8	9	10	11

Hàm `crosstab(...)` ở trên cho thấy thứ trong tuần được đánh số từ 0 tới 4 tương ứng với Thứ Hai đến Thứ Sáu.

### Thống kê số ngày tăng/giảm/không đổi theo thứ trong tuần

```
df_day_pn = pd.crosstab(df_avg_price['day'], df_avg_price['pn'])
```

df\_day\_pn

pn	-1	0	1
day			
0	138	2	118
1	137	3	120
2	121	1	143
3	126	1	140
4	143	1	124

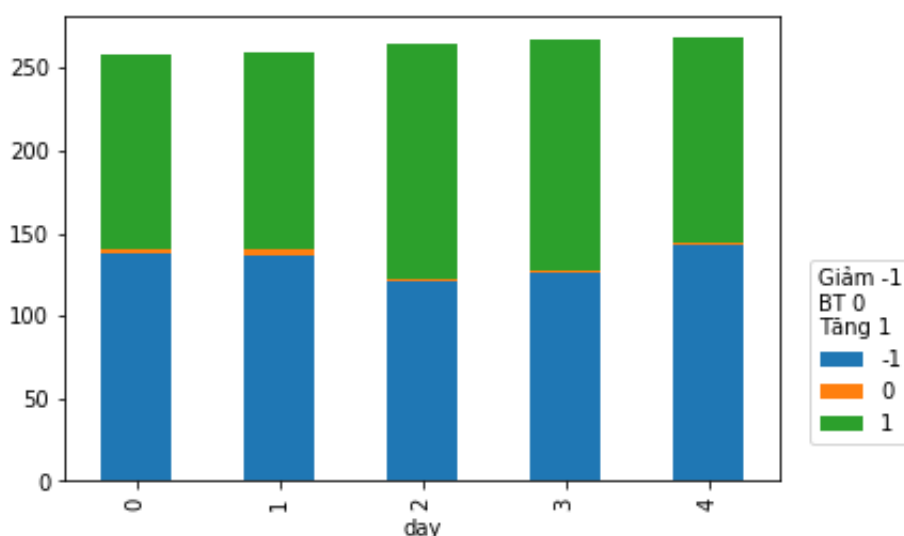
Kết quả cho thấy vài thông tin:

- Trong Thứ 2, số ngày giảm là 138, số ngày tăng là 118. Chú ý khái niệm Tăng/Giảm ở đây là so với ngày giao dịch trước đó (ở đây là Thứ Sáu tuần trước).
- Trong Thứ 3, số ngày giảm là 137, số ngày tăng là 120.
- Trong Thứ 4, số ngày giảm là 121, **số ngày tăng là 143**.
- Trong Thứ 5, số ngày giảm là 126, số ngày tăng là 140.
- Trong Thứ 6, **số ngày giảm là 143**, số ngày tăng là 124.

Theo số liệu thì Thứ 4 và Thứ 6 có vẻ ngược nhau. Thứ 4 thì tăng nhiều hơn so với giảm. Thứ 6 thì giảm nhiều hơn là tăng.

### Vẽ biểu đồ

```
import matplotlib.pyplot as plt
df_day_pn.plot.bar(stacked=True)
plt.legend(title='Giảm -1\nBT 0\nTăng 1', bbox_to_anchor=(1.2, 0.5))
plt.show()
```



Câu hỏi đặt ra ở đây là số ngày tăng hoặc giảm trong một thứ nào đó có ý nghĩa thống kê hay không?

Ví dụ nhìn biểu đồ thì có vẻ ngày thứ 4 là tỉ số Tăng/Giảm nhiều nhất.

Xem lại dữ liệu ngày Thứ 4 (cột day = 2)

df\_day\_pn

pn	-1	0	1
day			
0	138	2	118
1	137	3	120
2	<b>121</b>	<b>1</b>	<b>143</b>
3	126	1	140
4	143	1	124

Trong bộ dữ liệu có  $121 + 1 + 143 = 265$  ngày thứ Tư. Trong đó có 121 ngày giảm (chiếm  $121/265 = 45.66\%$ ) và 143 ngày tăng, chiếm  $143/265 = 53.96\%$

Thử tính chỉ số ztest và p-value (Xem lại kiến thức [Ngày 3, Bài 14: So sánh 2 tỉ lệ](#))

```
import numpy as np
from statsmodels.stats.proportion import proportions_ztest

# Count là số ngày tăng và số ngày giảm trong thứ 4
count = np.array([143, 121])
nobs = np.array([265, 265])
zstat, pval = proportions_ztest(count, nobs)
print('Tỉ số ztest:', zstat)
print('Trị số p:', pval)
```

Tỉ số ztest: 1.9112514762620285

Trị số p: 0.05597227155191926

Chỉ số ztest là 1.9, **gần 2 lần**. Chúng ta có thể có sự khác biệt giữa số ngày tăng và số ngày giảm của cổ phiếu VNM trong Thứ 4. Đồng thời trị số p = 0.056, hơi lớn hơn 0.05. Về lý thuyết diễn giải theo trị số p là không có ý nghĩa thống kê.

Tuy nhiên, giá trị ztest và trị số p đang rất sát ngưỡng “có ý nghĩa thống kê”. Kết quả này rất đáng xem xét giả thuyết: **Vào ngày thứ 4 thì cổ phiếu VNM thường là tăng**. Nhìn vào biểu đồ sẽ suy đoán là ngày Thứ 5 đa số sẽ tăng. Sau đó đến ngày Thứ 6 thì đa số sẽ giảm.

Phần kiểm chứng giả thuyết trên thì nhường lại cho các chuyên gia về cổ phiếu nhé!

### *Nhìn bảng số liệu theo %*

Thống kê theo % bằng cách sử dụng hàm `pd.crosstab(...)` với tham số `normalize:`

```
df_day_pn_percentage = pd.crosstab(df_avg_price['day'],  
df_avg_price['pn'], normalize='index').round(4)*100  
print('%Tăng/Giảm theo thứ:\n', df_day_pn_percentage)
```

```
%Tăng/Giảm theo thứ:  
pn      -1      0      1  
day  
0      53.49  0.78  45.74  
1      52.69  1.15  46.15  
2      45.66  0.38  53.96  
3      47.19  0.37  52.43  
4      53.36  0.37  46.27
```

## Đọc và vẽ tín hiệu âm thanh

Bài viết này minh họa bằng code Python.

Tài liệu tham khảo chính:

eBook “Python Machine Learning Cookbook”, (Packt Publishing, 2019) của Prateek Joshi.

Tải file <https://thachln.github.io/datasets/good-morning.wav> về đường dẫn “D:/ai2020/data/good-morning.wav” và thực thi đoạn code sau:

```
import numpy as np
import matplotlib.pyplot as plt
from scipy.io import wavfile

# Read the input file
wav_file_path = 'D:/ai2020/data/good-morning.wav'
sampling_freq, audio = wavfile.read(wav_file_path)

print('\nSampling frequency:', sampling_freq)

# Print the params
print('\nShape:', audio.shape)
print('Datatype:', audio.dtype)
print('Duration:', round(audio.shape[0] / float(sampling_freq), 3),
      'seconds')

# Normalize the values
audio = audio / (2.*15)

# Extract first 100 values for plotting
audio = audio[0:100]

# Build the time axis
x_values = np.arange(0, len(audio), 1) / float(sampling_freq)

# Convert to seconds
x_values *= 1000
```

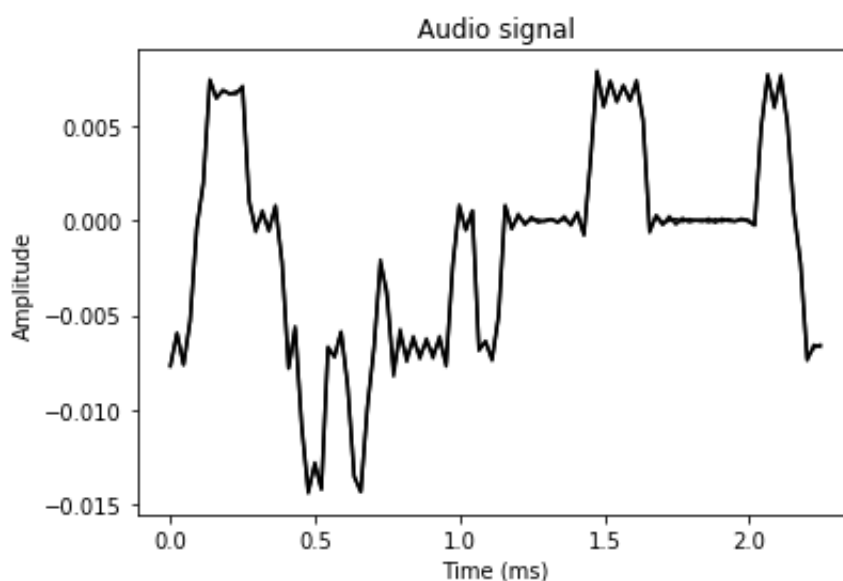
```
# Plotting the chopped audio signal
plt.plot(x_values, audio, color='black')
plt.xlabel('Time (ms)')
plt.ylabel('Amplitude')
plt.title('Audio signal')
plt.show()
```

Sampling frequency: 44100

Shape: (33897, 2)

Datatype: int16

Duration: 0.769 seconds



Một chút phân tích:

### Bước 1: Import thư viện

Import các gói thư viện trong 3 dòng đầu tiên:

```
import numpy as np
import matplotlib.pyplot as plt
from scipy.io import wavfile
```

### Bước 2: Đọc file âm thanh và xem tần số mẫu

Đọc file âm thanh từ thư mục và hiển thị tần số mẫu.

```
# Read the input file
wav file path = 'D:/ai2020/data/good-morning.wav'
sampling_freq, audio = wavfile.read(wav file path)

print('\nSampling frequency:', sampling_freq)
```

Sampling frequency: 44100

Ghi chú:

File âm thanh (trong ví dụ này là file good-morning.wav là do tôi tự thu âm bằng phần mềm Audacity) là một file trên máy tính được lưu trữ dạng số (digitized version) của tín hiệu âm thanh thật (actual audio signals, trong trường hợp này là giọng nói “good morning” của tôi). Trong khi thu âm thì phần mềm Audacity mặc định cấu hình tần số mẫu (sample) là 44100 Hz. Tức là mỗi một giây phần mềm Audacity (kèm micro của máy tính) thu được 44100 phần tín hiệu âm thanh (audio parts). Nói cách khác mỗi phần tín hiệu này được lưu trong 1/44100 giây. Khi tần số lấy mẫu (sampling rate) cao thì chúng ta sẽ cảm giác tín hiệu âm thanh liên tục khi nghe lại bằng các thiết bị phát âm thanh (audio players)

### Bước 3: Xem tham số của âm thanh

Ba dòng tiếp theo hiển thị thêm các tham số của âm thanh

```
print('\nShape:', audio.shape)
print('Datatype:', audio.dtype)
print('Duration:', round(audio.shape[0] / float(sampling_freq), 3),
      'seconds')
```

Shape: (33897, 2)  
Datatype: int16  
Duration: 0.769 seconds

Kết quả cho thấy có 2 luồng âm thanh, mỗi luồng có 33897 tín hiệu. Mỗi tín hiệu (audio signal) được lưu trong số nguyên có độ dài 16bit.

Tính độ dài của đoạn âm thanh bằng cách: lấy số tín hiệu nhân với 1/sampling\_freq:  $33897 * 1/44100 \approx 0.769$  giây.

### Bước 4: Chuẩn hóa tín hiệu

Do tín hiệu âm thanh (audio signal) được lưu trong số nguyên có dấu với độ dài 16bit, nên chuẩn hóa bằng cách lấy độ lớn của tín hiệu chia cho  $2^{15}$

```
audio = audio / (2.**15)
```

### Bước 5: Chọn tín hiệu để vẽ biểu đồ

Chọn 100 giá trị của tín hiệu đầu tiên:

```
audio = audio[0:100]
```

## Tải sách nói “Từ tốt đến vĩ đại”

Mã nguồn Python sau đây sẽ giúp bạn tải các file audio sách nói “Từ Tốt Đến Vĩ Đại” từ trang web <https://phatphapungdung.com/sach-noi/tu-tot-den-vi-dai-171248.html> về thư mục 'D:/Temp/Tu-tot-den-vi-dai/'.

Hãy tự khám phá nội dung mã nguồn và sửa lại thư mục theo ý bạn nhé.

Gợi ý: Nên copy & paste mã nguồn vào phần mềm Spyder (xem lại Bài 4), chọn các dòng code và nhấn F9 để chạy các dòng đã chọn)

```
import bs4
from bs4 import BeautifulSoup
import json
import requests
import pandas as pd
import os
from pathlib import Path

# =====
# Download audio from <url> to <outFolder? with filename <audioName>
# =====
def downloadAudio(url, outFolder, audioName='null'):
    if (audioName == 'null'):
        audioName = Path(url).name
    fileOut = Path(os.path.join(outFolder, audioName))
    response = requests.get(url)

    fileOut.write_bytes(response.content)

    return

# =====
# Get list of audio link and title from website
# @return dataframe(title, url)
# =====
def parseData():
    titles = []
    urls = []
    url = 'https://phatphapungdung.com/sach-noi/tu-tot-den-vi-dai-171248.html'
```



```
response = requests.get(url)

html_soup = BeautifulSoup(response.text, 'html.parser')

html_data = html_soup.find('div', {'class': 'fp-playlist-external'})

# article_containers = html_soup.find_all("div", class_="fp-playlist-external is-audio")

# content = article_containers.text

# print(html_data)

for e in html_data:
    if isinstance(e, bs4.element.Tag):
        print(type(e))
        print('Element:', e)
        data_item_json = e['data-item']
        print('Type of data item', type(data_item_json))
        json_data = json.loads(data_item_json)
        print('Sources:', json_data['sources'])
        print('Title', json_data['fv_title'])

        title = json_data['fv_title']
        print("Type of json_data[sources]:",
              type(json_data['sources']))

        print("Type of json_data[sources][0]:",
              type(json_data['sources'][0]))

        dict_data = json_data['sources'][0]
        print(dict_data['src'])

        url = dict_data['src']

        print('Parsed data: (Title, Url)=', title, url)
        titles.append(title)
        urls.append(url)
```

```

        else:
            print('No parse:', type(e))
    return pd.DataFrame({'title': titles, 'url': urls})

# Change your folder to contains books
outFolder = 'D:/Temp/Tu-tot-den-vi-dai/'
# Create root folder
if (not os.path.exists(outFolder)):
    os.mkdir(outFolder)

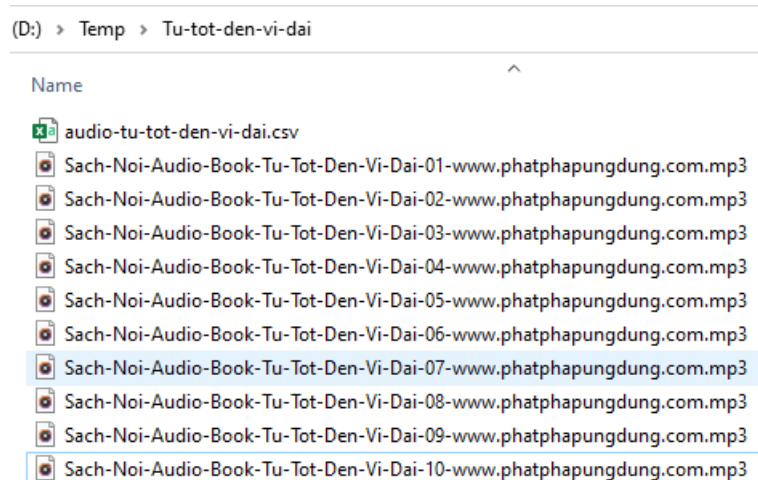
df = parseData()

# Write dataframe into CSV
df.to_csv(os.path.join(outFolder, 'audio-tu-tot-den-vi-dai.csv'),
          encoding='utf-8')

# Scan all data frame of free books
for index, row in df.iterrows():
    url = row['url']
    title = row['title']
    print('...')
    downloadAudio(url, outFolder)

```

Kết quả được thư mục như sau:



## Vẽ bản đồ Việt Nam

### Cài thư viện

```
install.packages('raster')
```

### Lấy dữ liệu bản đồ Việt Nam

```
# Lấy dữ liệu cho VN ở cấp tỉnh:
library(raster)

vietnam = getData('GADM', country = 'Vietnam', level = 1)
head(vietnam)
```

	GID_0	NAME_0	GID_1	NAME_1	VARNAME_1	NL_NAME_1	TYPE_1
1	VNM	Vietnam	VNM.1_1	An Giang	An Giang	<NA>	T<U+1EC9>nh
12	VNM	Vietnam	VNM.2_1	B<U+1EA1>c Liêu	Bac Lieu	<NA>	T<U+1EC9>nh
23	VNM	Vietnam	VNM.3_1	B<U+1EAF>c Giang	Bac Giang	<NA>	T<U+1EC9>nh
34	VNM	Vietnam	VNM.4_1	B<U+1EAF>c K<U+1EA1>n	Bac Kan	<NA>	T<U+1EC9>nh
45	VNM	Vietnam	VNM.5_1	B<U+1EAF>c Ninh	Bac Ninh	<NA>	T<U+1EC9>nh
56	VNM	Vietnam	VNM.6_1	B<U+1EBF>n Tre	Ben Tre	<NA>	T<U+1EC9>nh

	ENGTYPE_1	CC_1	HASC_1
1	Province	<NA>	VN.AG
12	Province	<NA>	VN.BL
23	Province	<NA>	VN.BG
34	Province	<NA>	VN.BK
45	Province	<NA>	VN.BN
56	Province	<NA>	VN.BR

### Xem các cột dữ liệu

```
names(vietnam)
```

```
[1] "GID_0"      "NAME_0"      "GID_1"      "NAME_1"      "VARNAME_1"  "NL_NAME_1"  "TY
PE_1"
[8] "ENGTYPE_1" "CC_1"       "HASC_1"
```

### Xem kiểu dữ liệu của biến vietnam

```
class(vietnam)
```

```
[1] "SpatialPolygonsDataFrame"
attr(,"package")
[1] "sp"
```

### Plot bản đồ

```
plot(vietnam)
```



Ghi chú: Có vẻ dữ liệu về Quần đảo Hoàng Sa và Trường Sa của Việt Nam không rõ ràng.

Code đầy đủ

# Lấy dữ liệu cho VN ở cấp tỉnh:

```
library(raster)
```

```
vietnam = getData('GADM', country = 'Vietnam', level = 1)
```

```
plot(vietnam)
```

## Đọc ảnh y khoa DiCOM

### Cài đặt thư viện

```
pip install pydicom
```

### Code Python

Đoạn code code sẽ đọc ảnh DiCOM từ thư mục, hiển thị vài thông tin cơ bản và hiển thị ảnh:

Tham khảo code:

[https://pydicom.github.io/pydicom/stable/auto\\_examples/input\\_output/plot\\_read\\_dicom.html](https://pydicom.github.io/pydicom/stable/auto_examples/input_output/plot_read_dicom.html)

```
import matplotlib.pyplot as plt
import pydicom
filePath = 'D:/ai2020/data/mri/ThachLN.dcm'
dataset = pydicom.dcmread(filePath)

print("Storage type.....:", dataset.SOPClassUID)

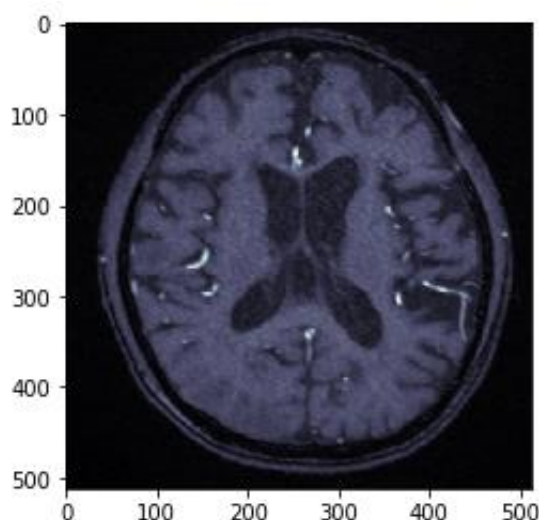
pat_name = dataset.PatientName
display_name = pat_name.family_name + ", " + pat_name.given_name
print("Patient's name....:", display_name)
print("Patient id.....:", dataset.PatientID)
print("Modality.....:", dataset.Modality)
print("Study Date.....:", dataset.StudyDate)

if 'PixelData' in dataset:
    rows = int(dataset.Rows)
    cols = int(dataset.Columns)
    print("Image size.....: {rows:d} x {cols:d}, {size:d} bytes".format(
        rows=rows, cols=cols, size=len(dataset.PixelData)))
    if 'PixelSpacing' in dataset:
        print("Pixel spacing.....:", dataset.PixelSpacing)

# use .get() if not sure the item exists, and want a default value if missing
print("Slice location....:", dataset.get('SliceLocation', "(missing)"))
```

```
# plot the image using matplotlib
plt.imshow(dataset.pixel_array, cmap=plt.cm.bone)
plt.show()
```

```
Storage type.....: 1.2.840.10008.5.1.4.1.1.4
Patient's name....: Le Ngoc Thach,
Patient id.....: ThachLN.github.io
Modality.....: MR
Study Date.....: 20200608
Image size.....: 512 x 512, 524288 bytes
Pixel spacing....: [0.3515625, 0.3515625]
Slice location...: 25.713560265779
```



### Xem toàn bộ dữ liệu của Dataset

Trong ví dụ trên bạn thấy có một đối tượng là Dataset chứa toàn bộ thông tin về ảnh DiCOM. Đoạn code sau hiển thị toàn bộ thông tin của file ảnh (bạn thay đường dẫn filePath trở tới ảnh trong máy bạn) bằng cách duyệt từng phần tử (element) trong dataset)

```
import pydicom
filePath = 'D:/Thach/DICOM1'
dataset = pydicom.dcmread(filePath)

for elem in dataset:
    print(elem)
```

Kết quả có dạng như sau:

```
(0008, 0008) Image Type CS: ['ORIGINAL', 'PRIMARY', 'M
', 'ND', 'NORM']
(0008, 0012) Instance Creation Date DA: '20200511'
(0008, 0013) Instance Creation Time TM: '135512.445000'
(0008, 0016) SOP Class UID UI: MR Image Storage
(0008, 0018) SOP Instance UID UI: 1.3.6..366.0
(0008, 0020) Study Date DA: '20200511'
(0008, 0021) Series Date DA: '20200511'
```

(0008, 0022) Acquisition Date	DA: '20200511'
(0008, 0023) Content Date	DA: '20200511'
(0008, 0030) Study Time	TM: '135300.695000'
(0008, 0031) Series Time	TM: '135512.442000'
(0008, 0032) Acquisition Time	TM: '135444.742500'
(0008, 0033) Content Time	TM: '135512.445000'
(0008, 0050) Accession Number	SH: ''
(0008, 0060) Modality	CS: 'MR'
(0008, 0070) Manufacturer	LO: 'MYWORKSPACE.VN'
(0008, 0080) Institution Name	LO: 'BV LÊ NGỌC THẠCH'
(0008, 0081) Institution Address	ST: '123 THÀNH PHỐ THỦ ĐỨC'
...	

Dấu ... có nghĩa là còn nhiều thông tin nữa mà tôi không đưa vào eBook được. Trong đó 3 dòng cuối cùng là tôi sửa lại cho vui.

Một câu hỏi đặt ra là bạn muốn chia sẻ ảnh này cho đồng nghiệp nhưng không muốn giữ lại các thông tin mang tính nhạy cảm. Tức là bạn muốn thay thế các dữ liệu riêng tư của bệnh nhân, bệnh viện bằng các thông tin chung chung (gọi là anonymous data) thì làm sao? Phần tiếp theo sẽ có lời giải.

### Xóa dữ liệu riêng tư trong ảnh DICOM

Đoạn code Python sau dựa vào địa chỉ của vùng dữ liệu trong dataset để thay đổi giá trị của dataset. Sau đó lưu thông tin đã chỉnh sửa thành file DICOM mới:

```
import pydicom

filePath = 'dicom file'
dataset = pydicom.dcmread(filePath)

# Patient's Name
dataset[0x0010, 0x0010].value = 'NO Manufacturer'
dataset[0x0008, 0x0070].value = 'NO Manufacturer'
dataset[0x0008, 0x0080].value = 'NO NAME'
dataset[0x0008, 0x0081].value = 'NO ADDRESS'

# Write to new file
dataset.save_as('new_file')
```

## Áp dụng biến đổi Fourier cho ảnh

Biến đổi Fourier (Fourier Transformation) là kỹ thuật giúp lọc và làm giảm thông tin nhiễu của thông tin nói chung. Trong bài này chúng ta áp dụng thử nghiệm Code Python để ứng dụng kỹ thuật Fourier Transformation để lọc thông tin nhiễu trong ảnh.

### Đọc ảnh từ Internet

Đoạn code sau đọc một hình ảnh từ Internet và hiển thị bằng hàm `matplotlib.pyplot.imshow(...)`.

```
import cv2
import numpy as np
import urllib
import matplotlib.pyplot as plt

url = 'https://thachln.github.io/datasets/ThachLN.png'
resp = urllib.request.urlopen(url)

img = np.asarray(bytearray(resp.read()), dtype="uint8")
img = cv2.imdecode(img, cv2.IMREAD_COLOR)
```

### Lưu ảnh vào thư mục

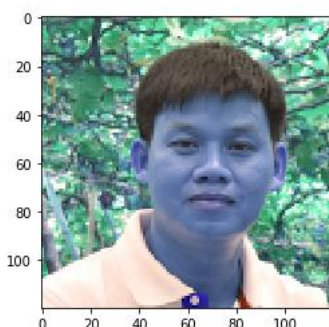
Để đảm bảo biến `img` chứa được ảnh đúng chính xác thì hãy thử lưu lại và dùng phần mềm xem ảnh để kiểm tra lại.

```
cv2.imwrite('D:/Temp/ThachLN.png', img)
```

### Hiển thị ảnh

Dùng lệnh `imshow(...)` trong thư viện `matplotlib.pyplot`

```
plt.imshow(img)
```



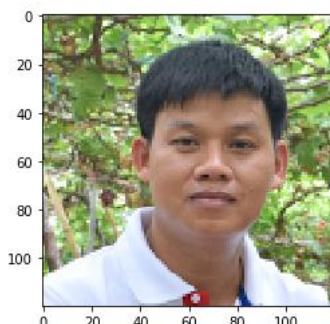
Có vẻ màu sắc hiển thị không được đúng?



Hãy thêm dòng code sau rồi gọi lại lệnh `plot.imshow(...)`:

```
img = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
```

bạn sẽ thấy hình được hiển thị đúng màu như hình gốc:



Lý giải cho điều này như sau: thư viện OpenCV đọc nội dung file ảnh vào lưu dạng mảng nhiều chiều (multi-dimensional) NumPy với thứ tự màu sắc là BGR (Blue Green Red). Trong khi đó việc hiển thị hình ảnh lên màn hình máy tính theo thứ tự màu sắc là RGB. Vì vậy cần có lệnh để hoán chuyển thứ tự màu sắc như trên. Trong hằng số `COLOR_BGR2RGB` của OpenCV, số 2 viết là two, đọc cũng là to, ý là **BGR to RGB**.

### Đọc ảnh vào chế độ xám

Code Python được viết lại đầy đủ như sau:

```
import cv2
import numpy as np
import urllib
import matplotlib.pyplot as plt

url = 'https://thachln.github.io/datasets/ThachLN.png'
resp = urllib.request.urlopen(url)

img = np.asarray(bytearray(resp.read()), dtype="uint8")
img = cv2.imdecode(img, cv2.IMREAD_GRAYSCALE)

img = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
plt.axis('off')
plt.imshow(img)
```



## Sử dụng Fast Transform Fourier

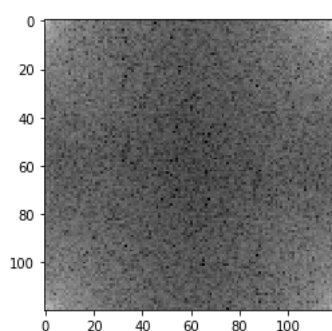
Đoạn code sau dùng OpenCV đọc ảnh với tham số `cv2.COLOR_BGR2BGRA`  
Sử dụng hàm `fft2` trong thư viện NumPy.fft

```
import cv2
import numpy as np
import urllib
import matplotlib.pyplot as plt

url = 'https://thachln.github.io/datasets/ThachLN.png'
resp = urllib.request.urlopen(url)

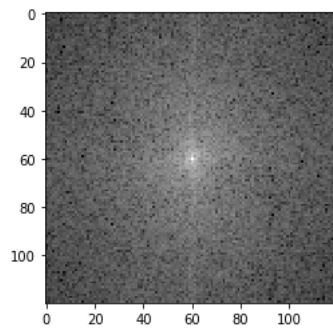
img = np.asarray(bytearray(resp.read()), dtype="uint8")
img = cv2.imdecode(img, cv2.COLOR_BGR2BGRA)

img_spectrum = np.fft.fft2(img)
plt.imshow(np.log(1+np.abs(img_spectrum)), 'gray')
```

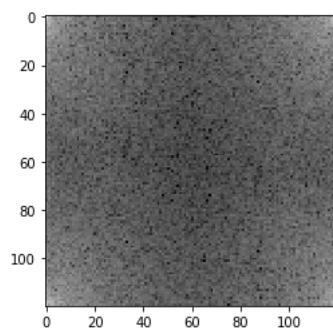


Tiếp tục xử lý bằng hàm `np.fft.fftshift`:

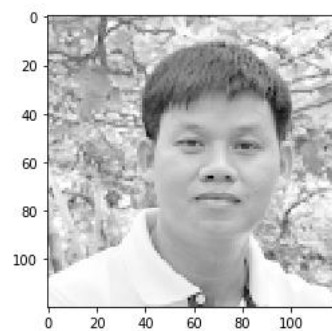
```
img_centered_spectrum = np.fft.fftshift(img_spectrum)
plt.imshow(np.log(1+np.abs(img_centered_spectrum)), 'gray')
```



```
img_decentralized = np.fft.ifftshift(img_centered_spectrum)
plt.imshow(np.log(1+np.abs(img_decentralized)), 'gray')
```



```
img_processed = np.fft.ifft2(img_decentralized)
plt.imshow(np.log(1+np.abs(img_processed)), 'gray')
```



## Sử dụng Git

### Giới thiệu

Trong thực tế khi có nhiều người làm việc chung với nhau trong một dự án mà sản phẩm số dưới dạng file trên máy tính thì cần có các công cụ hỗ trợ. Cụ thể là cần công cụ giải quyết các tình huống sau:

- Làm sao để có một nơi để lưu trữ file chung?
- Trường hợp hai hoặc nhiều người cùng chỉnh sửa một file thì làm sao? Đặc biệt là các file mã nguồn của phần mềm máy tính, trong đó có nhiều đoạn code có thể được nhiều lập trình viên chỉnh sửa cùng một lúc rồi nộp lên server?
- Làm sao lưu lại toàn bộ lịch sử và dữ liệu mà mọi người đã nộp lên server?
- Làm sao chốt các phiên bản phần mềm để phát hành (release).

Để giải quyết nhu cầu này thì các công ty sẽ triển khai một hệ thống gọi là Version Control System (Hệ thống Quản lý Phiên bản). Có nhiều giải pháp hoặc cách thức quản lý phiên bản, trong đó có 2 dạng phổ biến là SVN và GIT. Mỗi giải pháp thì có nhiều phần mềm cụ thể của nhiều hãng cung cấp.

Ví dụ hãng GitHub cung cấp cho cả thế giới trang web <https://github.com> để giúp người dùng cũng như doanh nghiệp có thể tạo và quản lý dự án phần mềm. Đối với doanh nghiệp muốn có một hệ thống tương tự như trang **github.com** thì có thể triển khai phần mềm GitLab (xem trang <https://gitlab.com/>). Hãng Microsoft cũng cung cấp hệ thống Azure cho phép người dùng quản lý dự án phần mềm trên trang <http://dev.azure.com/>. Cái hay của các trang github.com, gitlab.com, dev.azure.com này là vừa cho người dùng tạo tài khoản để **dùng miễn phí**, vừa có chế độ tính tiền (tùy theo nhu cầu và tính năng sử dụng). Vì các trang web này dùng chung giải pháp gọi là GIT nên tôi gọi chung là **Git Server**.

Trong bài viết này tôi giải thích các nội dung sau:

- ① Cách khai thác các dự án có sẵn trên Git Server. Cụ thể là minh họa cách lấy dự án từ các trang github.com, gitlab.com, dev.asuzre.com.
- ② Cách “sao chép” dự án có sẵn trên github.com để chỉnh sửa lại theo ý của mình.
- ③ Cách tạo dự án cho mình và cho nhóm để cùng làm việc trên Git Server. Cụ thể là minh họa trên các trang github.com, gitlab.com, dev.asuzre.com.

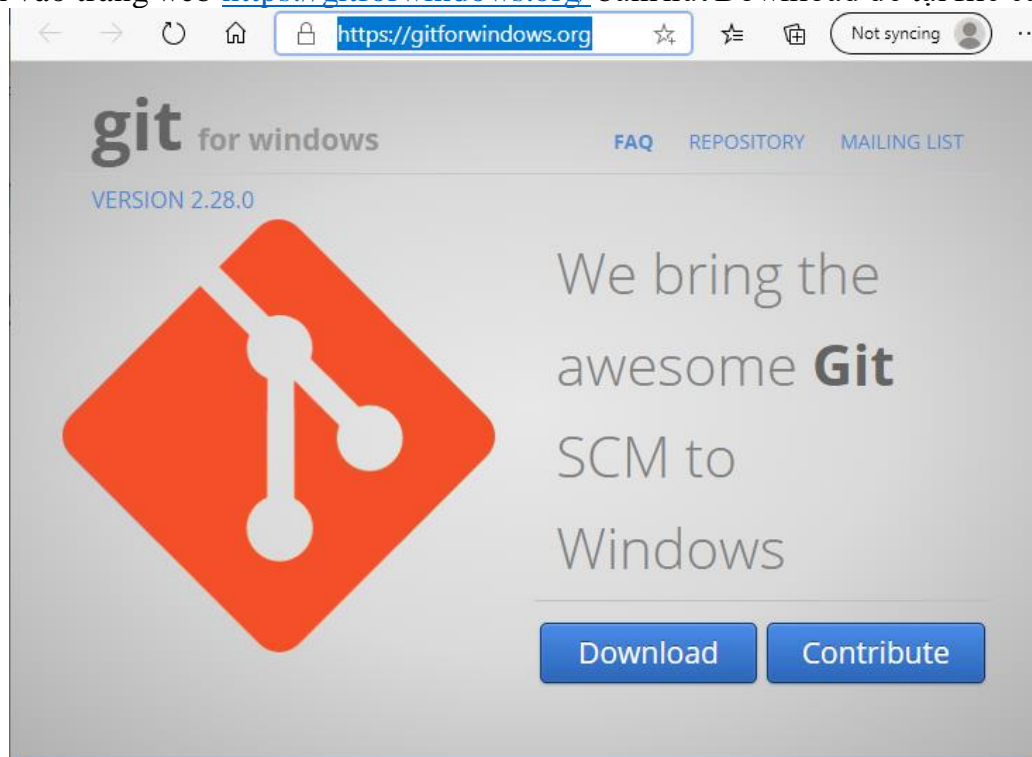
## Cài đặt phần mềm Git cho Windows

Để khai thác được GitServer thì trên máy tính dùng Windows cần cài hai phần mềm phổ biến gồm:

- ① git for windows
- ② TortoiseGit

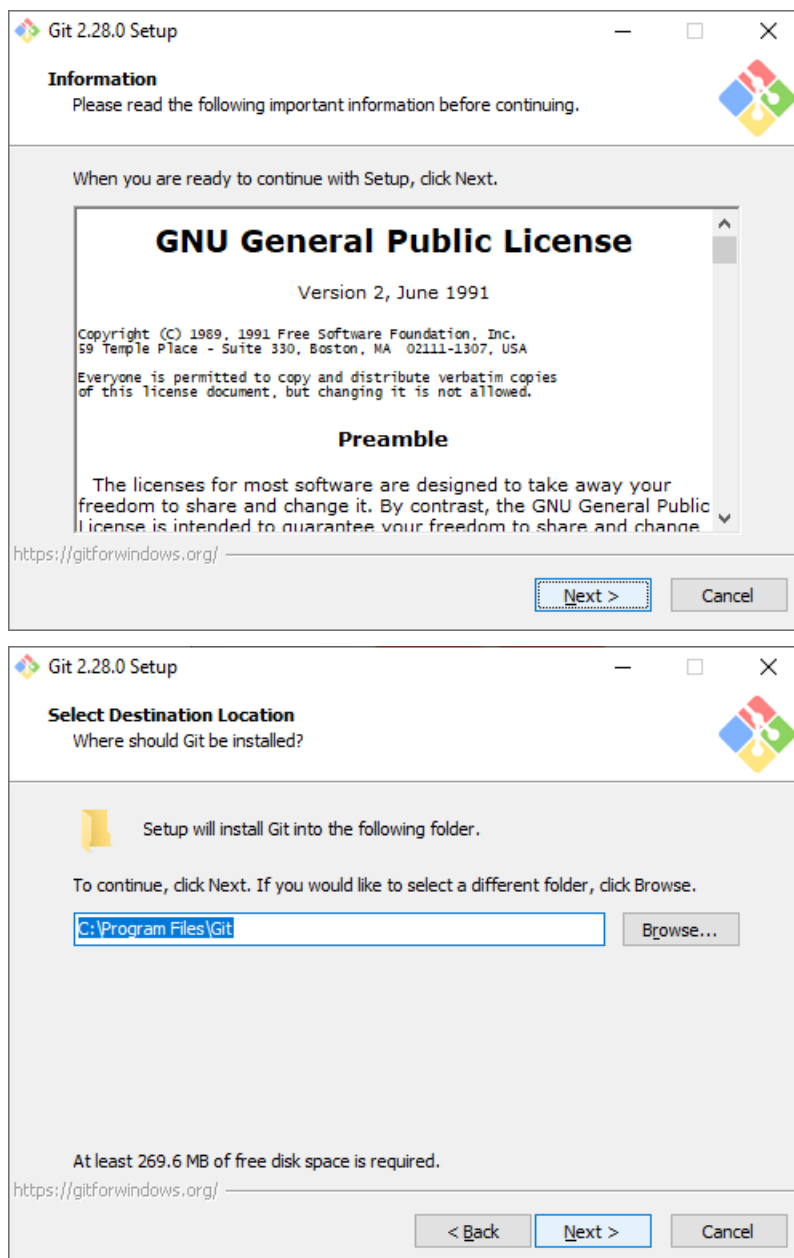
### *Tải và cài đặt gitforwindows*

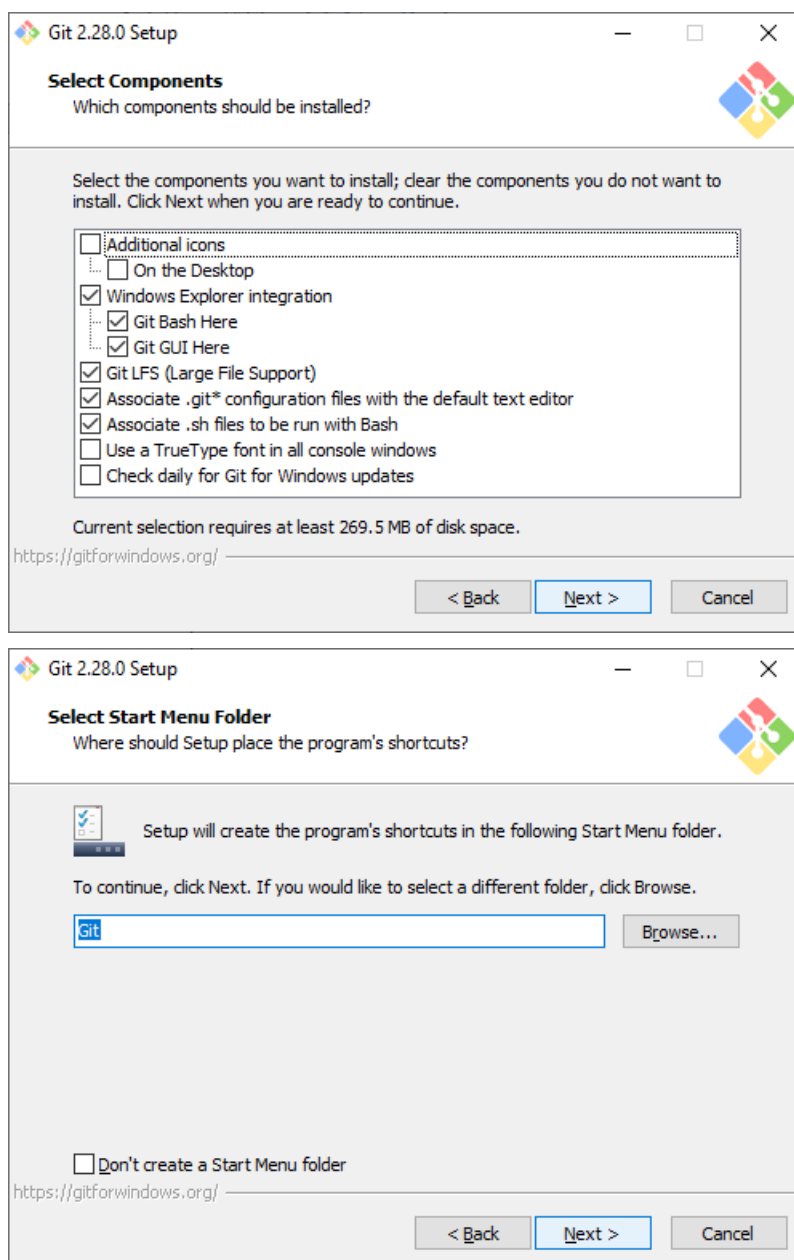
Bạn vào trang web <https://gitforwindows.org/> bấm nút Download để tải file cài đặt.

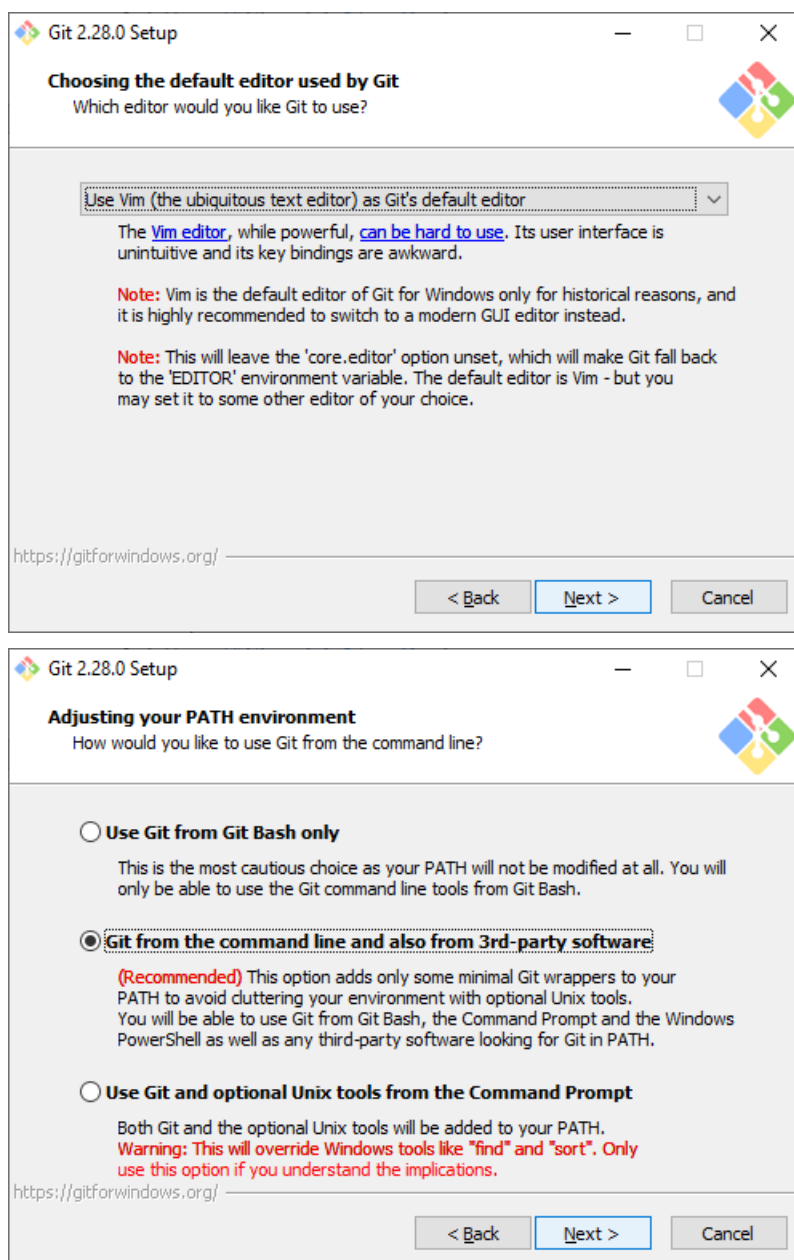


Máy của tôi là Windows 64 bit nên tải file có phiên bản hiện tại 2.28.0 như sau: **Git-2.28.0-64-bit.exe**.

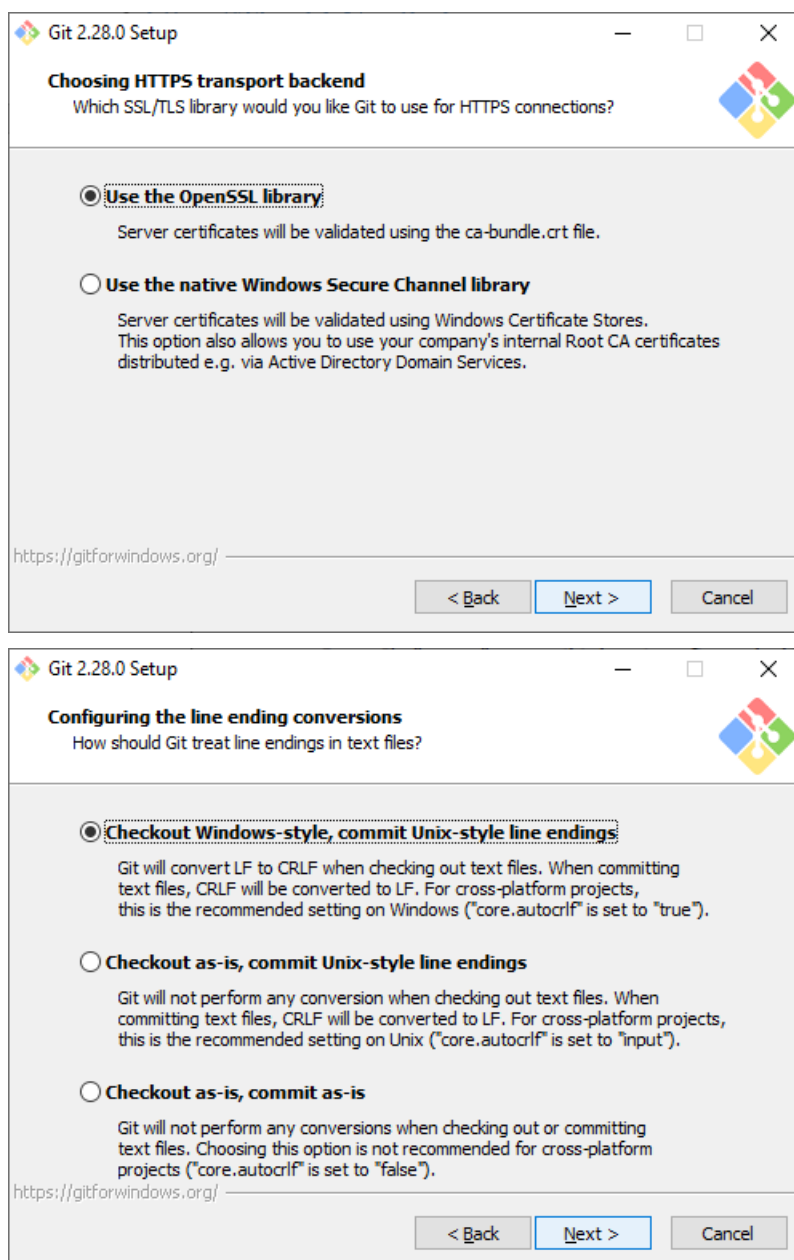
Sau khi tải về máy thì double-click (nhấn chuột hai lần liên tục) vào tên file để cài đặt. Hãy đọc qua các màn hình cài đặt, hiểu được chút nào thì hiểu và nhấn nút Next cho đến Finish.

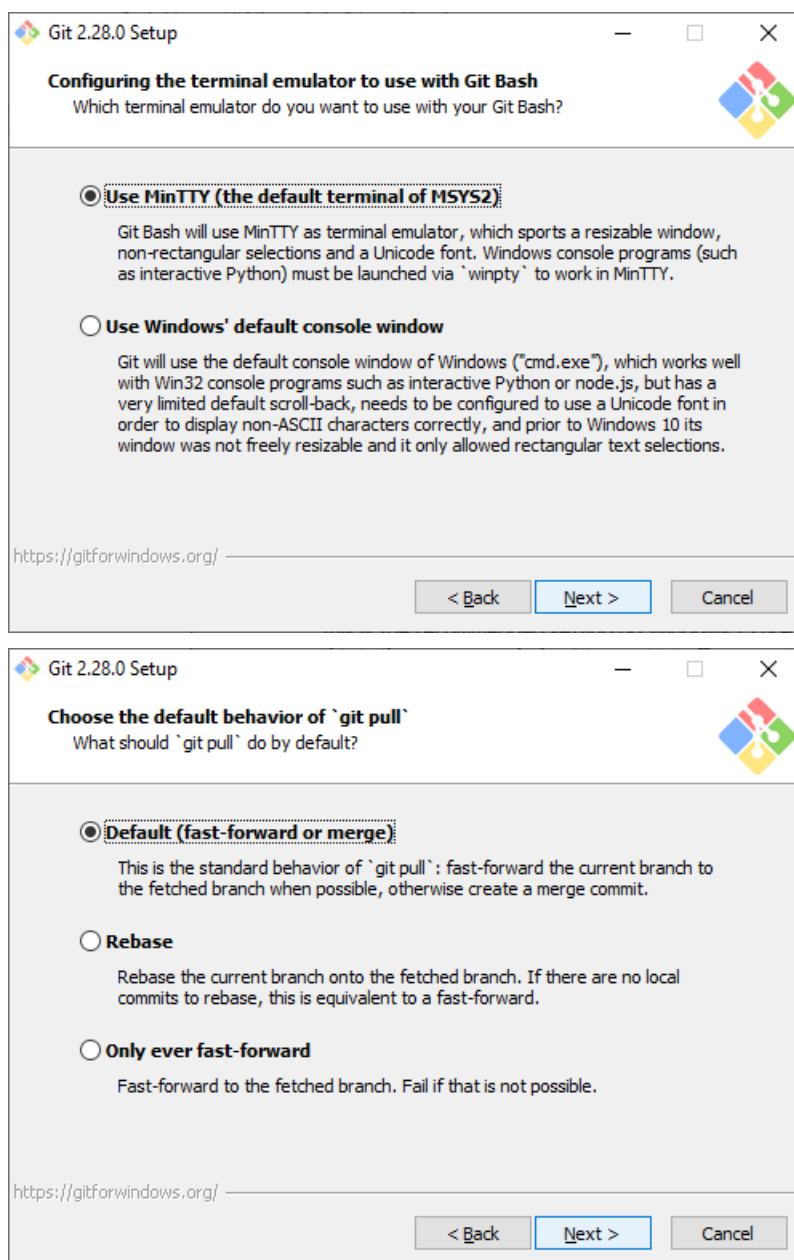


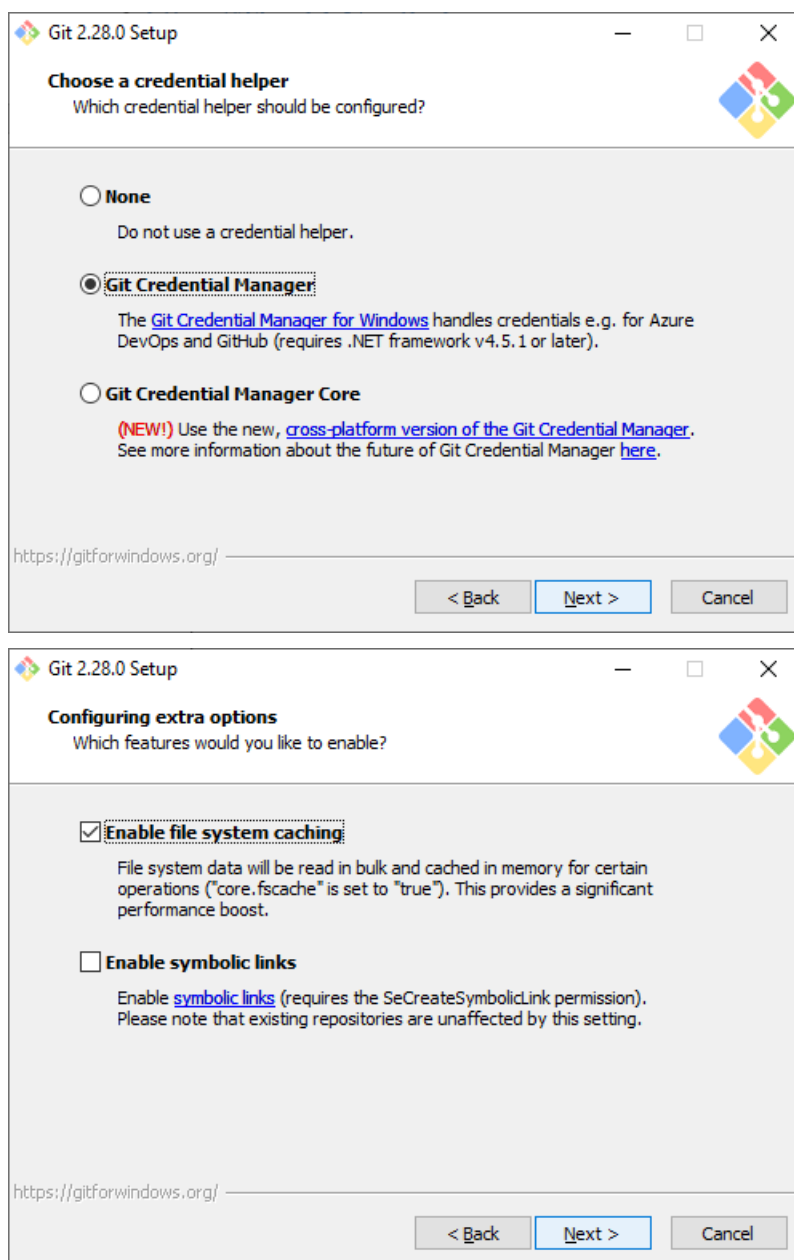


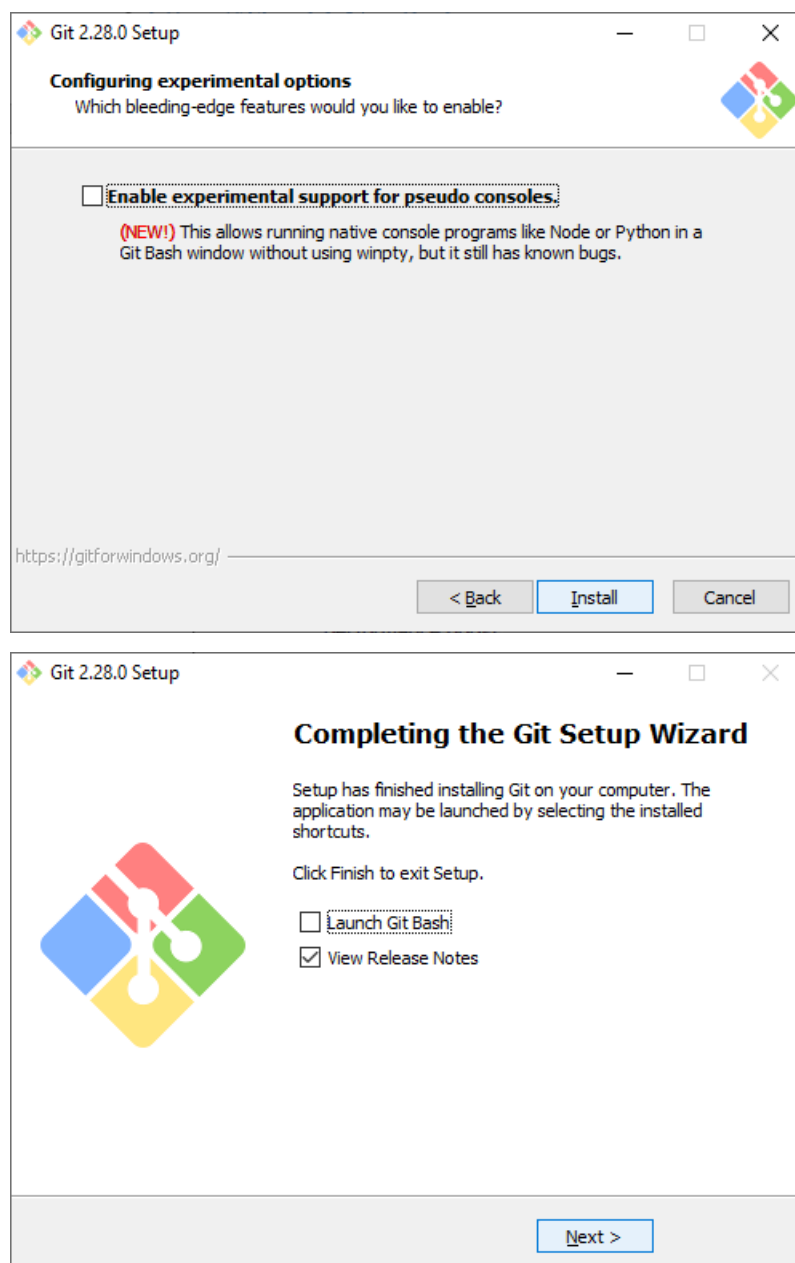








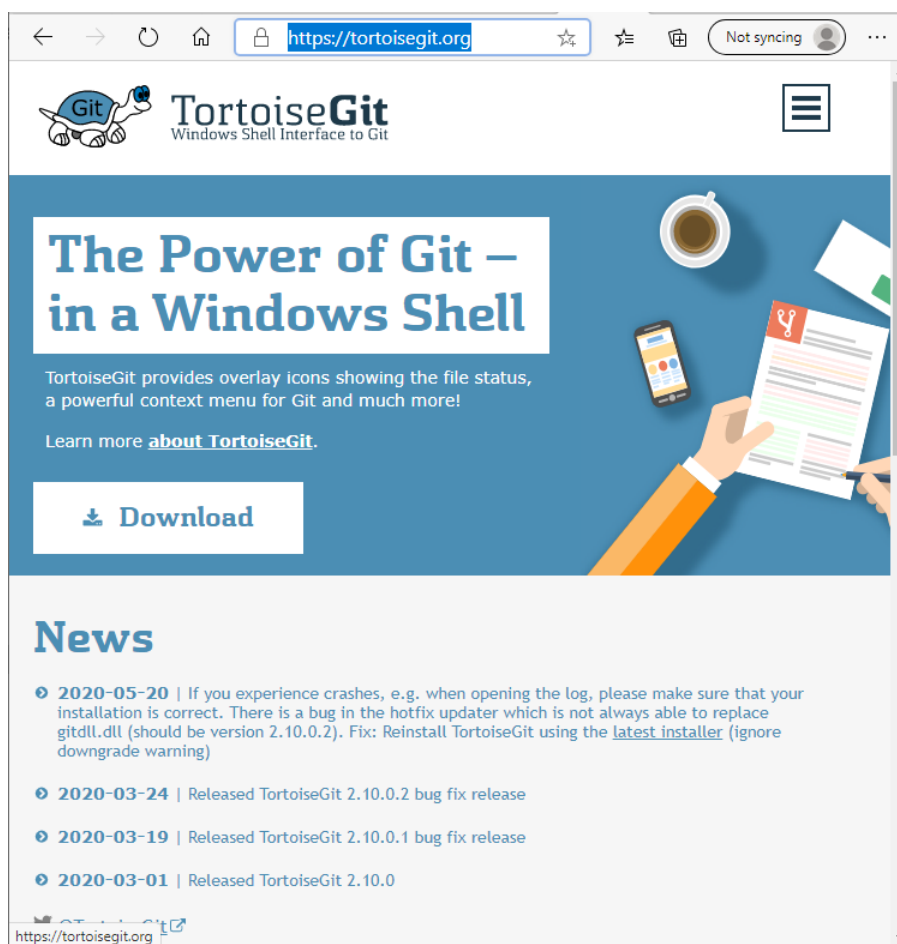




Đối với các bạn quen gõ lệnh thì gitforwindows này là đủ cho các bạn dùng. Tuy nhiên không phải ai cũng rành máy tính và nhớ được lệnh nên tôi khuyến nghị là nên dùng thêm phần mềm có giao diện tương đối dễ dùng là TortoiseGit.

### *Cài đặt TortoiseGit*

Bạn vào trang web <https://tortoisegit.org/> để tải phiên bản mới nhất.



[TortoiseGit.org](https://tortoisegit.org) » Download

## Download

The current stable version is: **2.10.0**

For detailed info on what's new, read the [release notes](#).

[FAQ: System prerequisites and installation](#) - This version doesn't run on Windows Vista and below, use [2.4.0](#) instead.

**Donate**

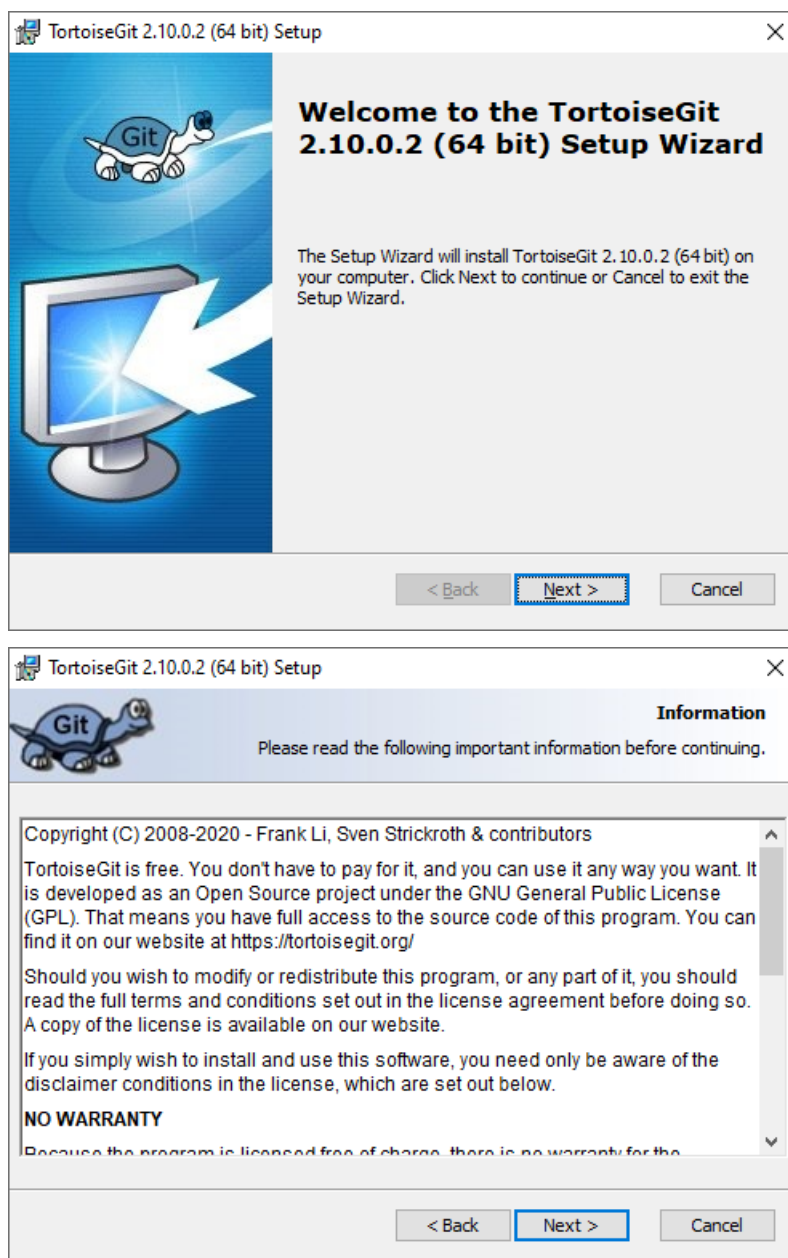
Please make sure that you choose the right installer for your PC, otherwise the setup will fail.

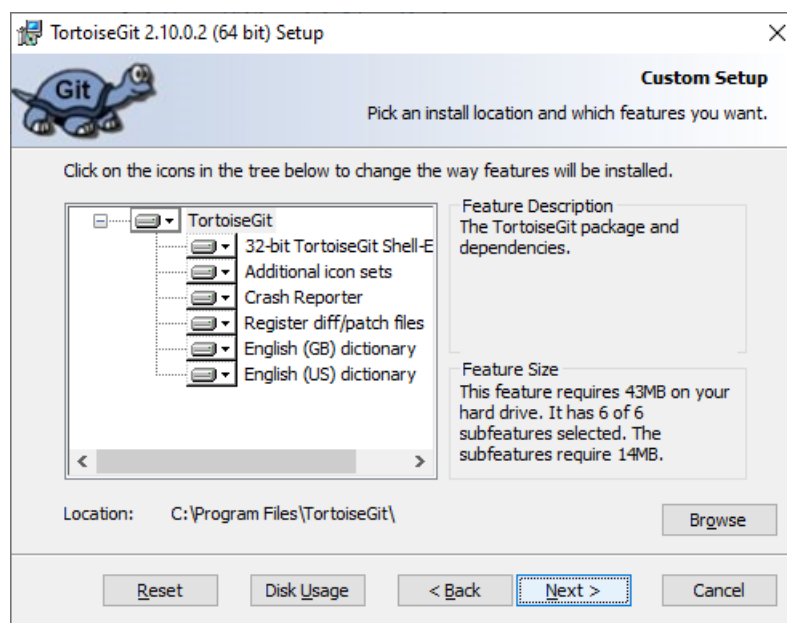
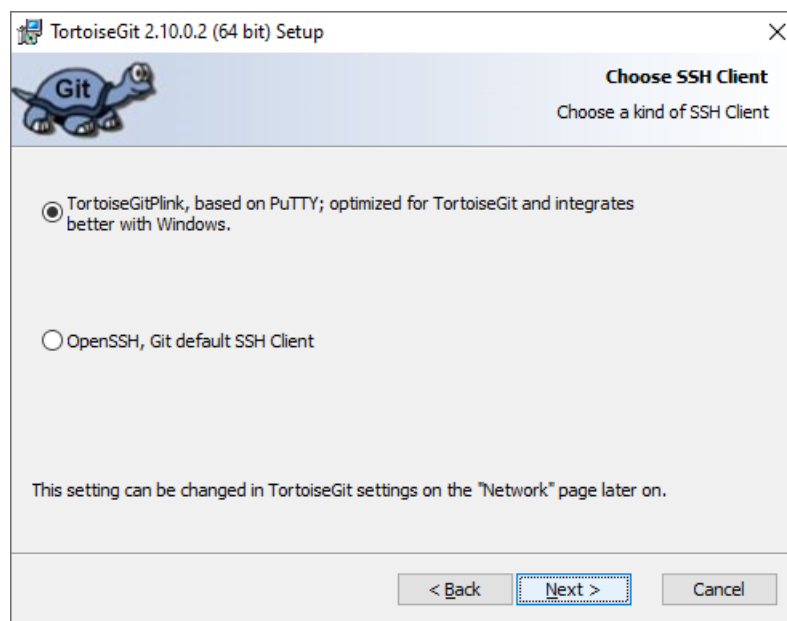
for 32-bit Windows	for 64-bit Windows
<a href="#">Download TortoiseGit 2.10.0.2 - 32-bit</a> (~16.8 MiB)	<a href="#">Download TortoiseGit 2.10.0.2 - 64-bit</a> (~19.0 MiB)

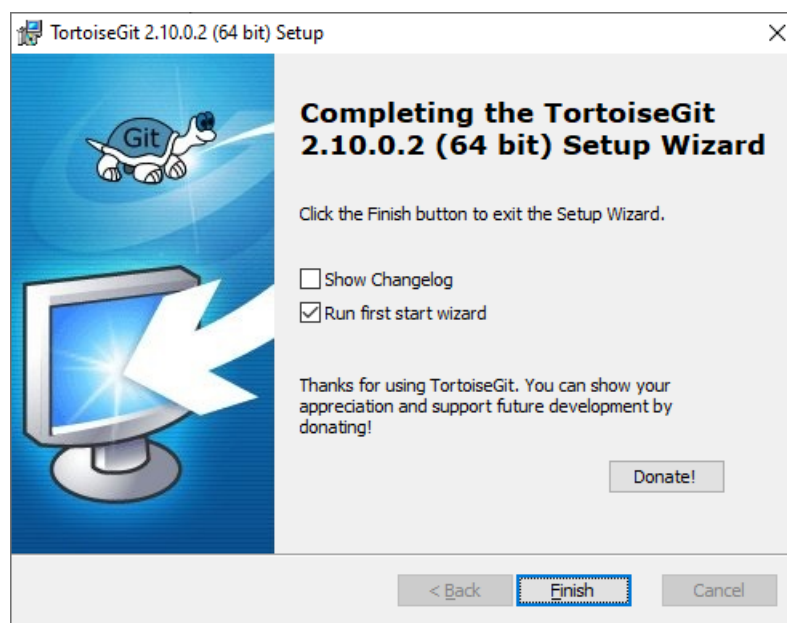
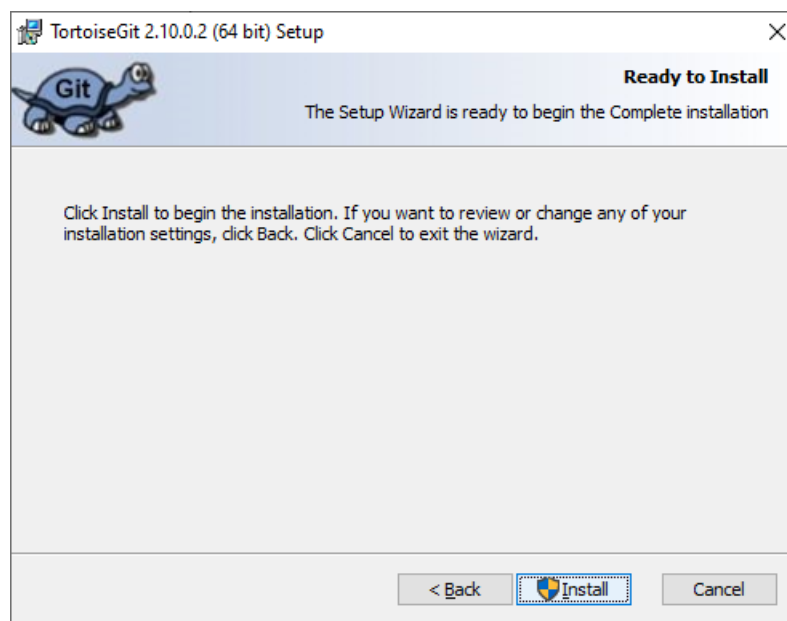
Tại thời điểm tài liệu này được viết thì TortoiseGit có phiên bản mới nhất là 2.10.0.2. Tôi dùng Windows 64 bit nên tải file TortoiseGit-2.10.0.2-64bit.msi.

Sau khi tải file về máy thì thực thi bằng cách double-click lên nó.

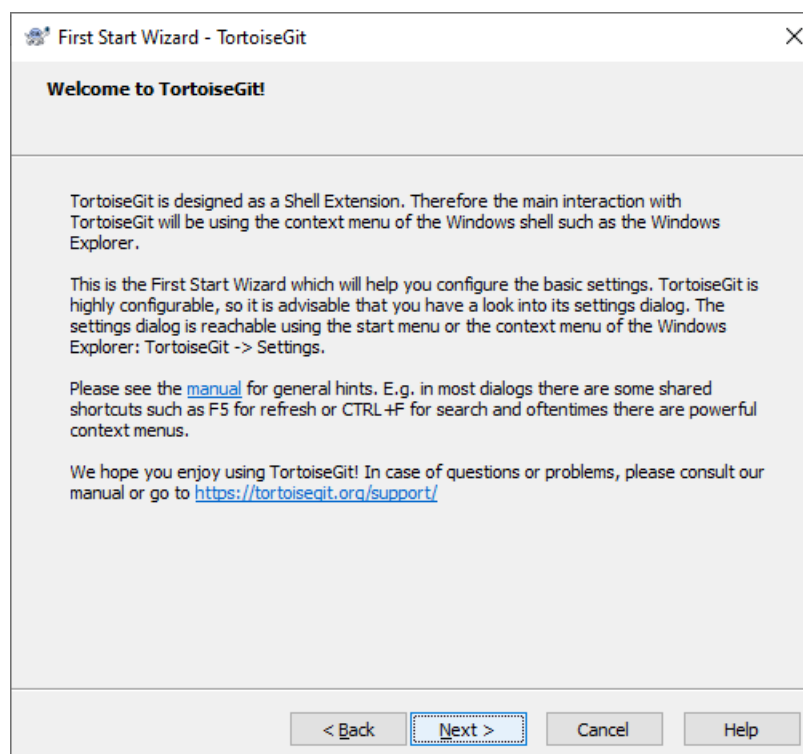
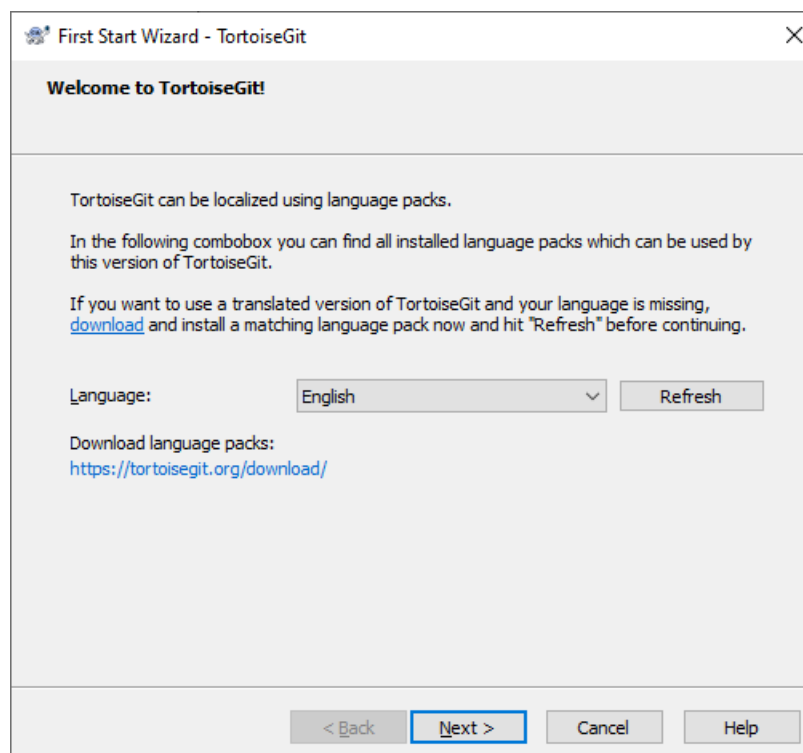
Quá trình cài đặt cũng tương tự như cài gitforwindow. Chủ yếu là nhấp nút Next.

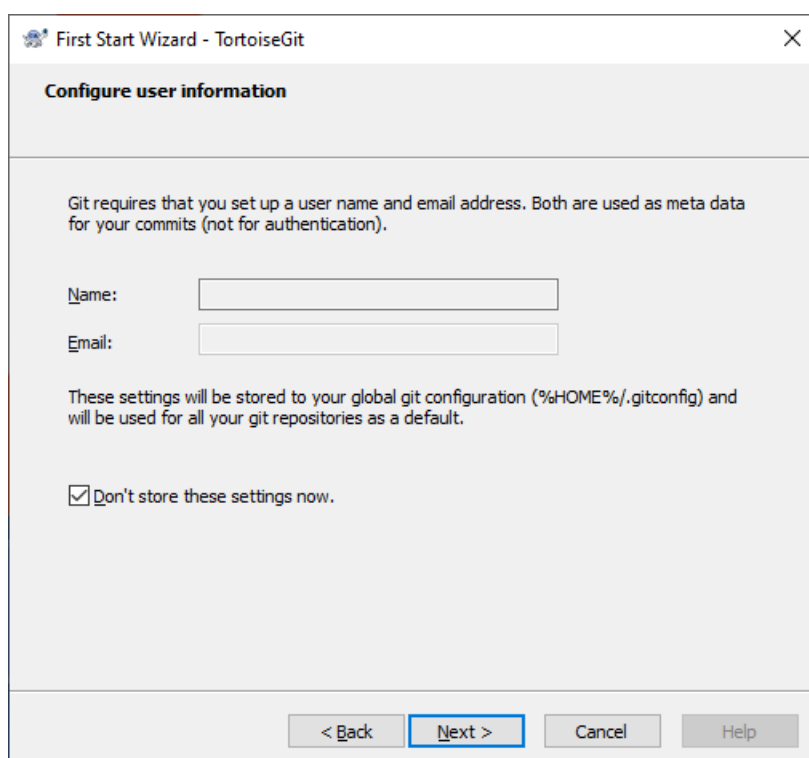
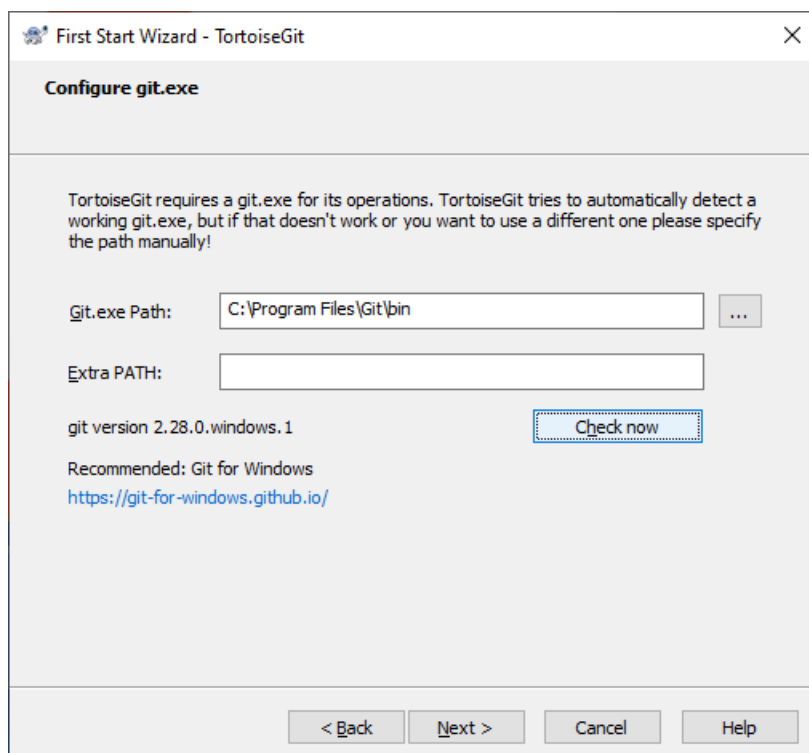


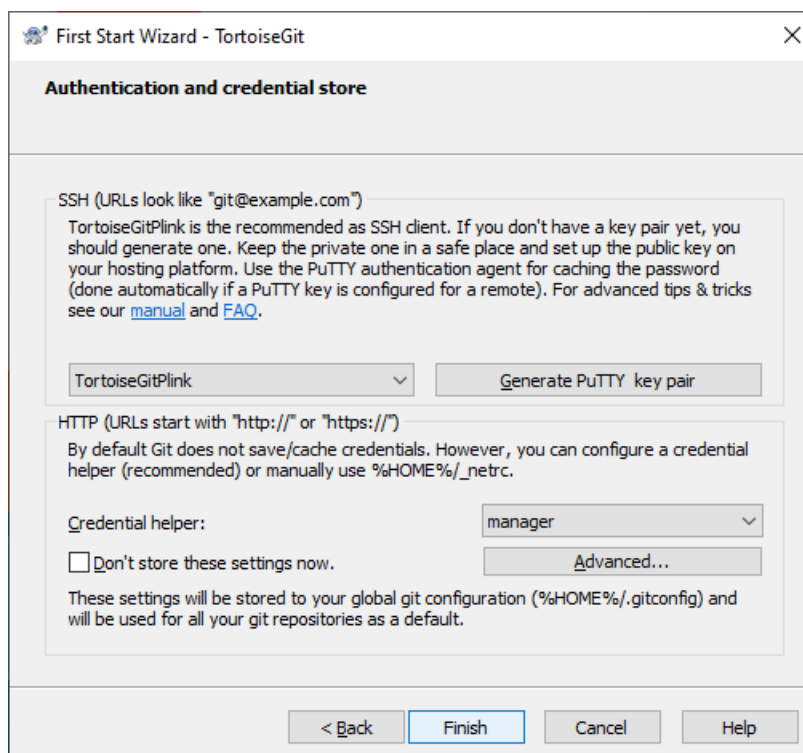














## Clone dự án có sẵn từ GitServer

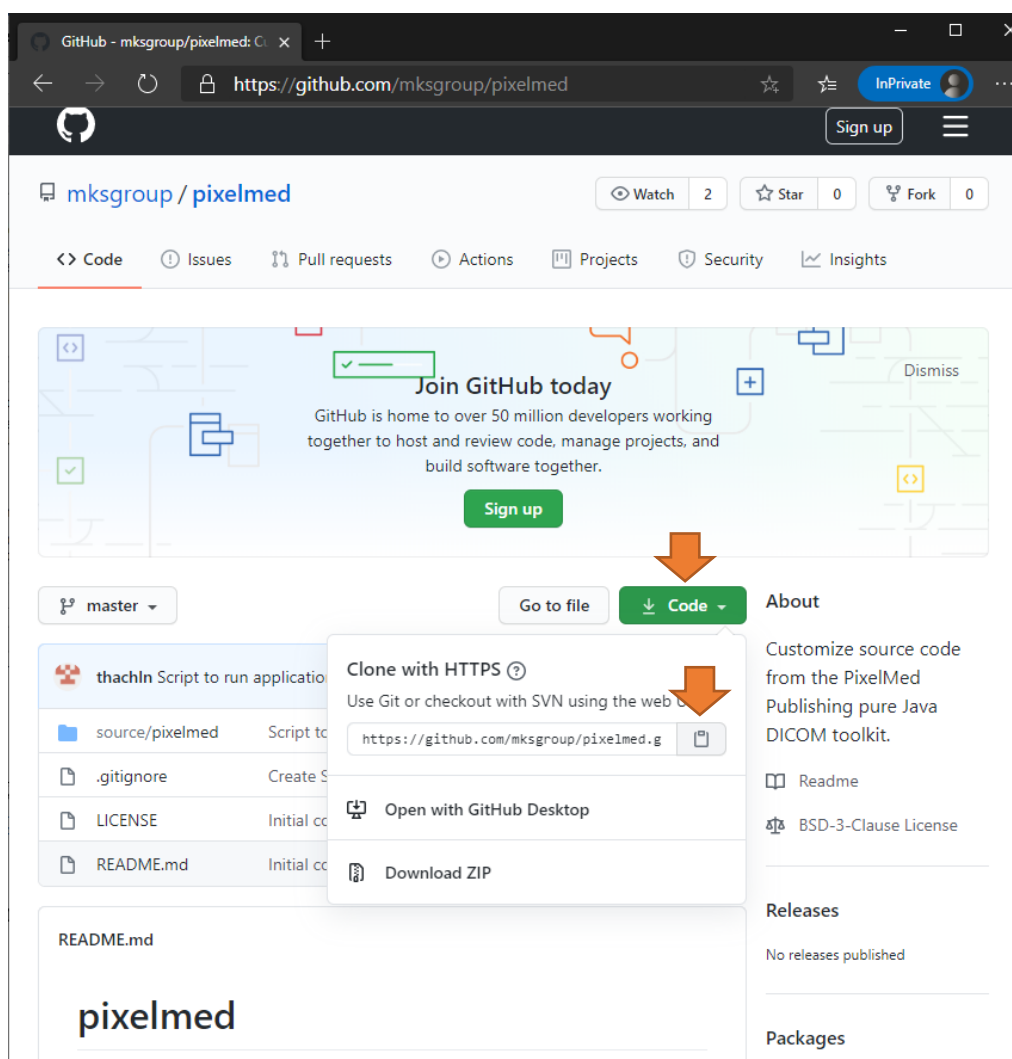
Clone là quá trình tạo bản sao toàn bộ dữ liệu của dự án từ trên GitServer về máy tính của người dùng. Sau khi Clone thì trên máy tính người dùng có thư mục dữ liệu của dự án để sẵn sàng làm việc (thêm/xóa/chỉnh sửa file).

### *Clone Dự án từ github.com*

#### **Bước 1:** Copy đường dẫn git của dự án

Ví dụ bạn được một đồng nghiệp chia sẻ là có một dự án cung cấp mã nguồn xử lý ảnh DiCOM tại link: <https://github.com/mksgroup/pixelmed>

Bạn có thể mở website lên xem và nhớ copy địa chỉ bằng cách quét chọn (select) địa chỉ và nhấn phím Ctrl + C. Một cách khác bấm vào nút  **Code** rồi sau đó bấm vào biểu tượng  bên phải ô địa chỉ như hình bên dưới:



## Bước 2: Tạo thư mục để chuẩn bị chứa dự án

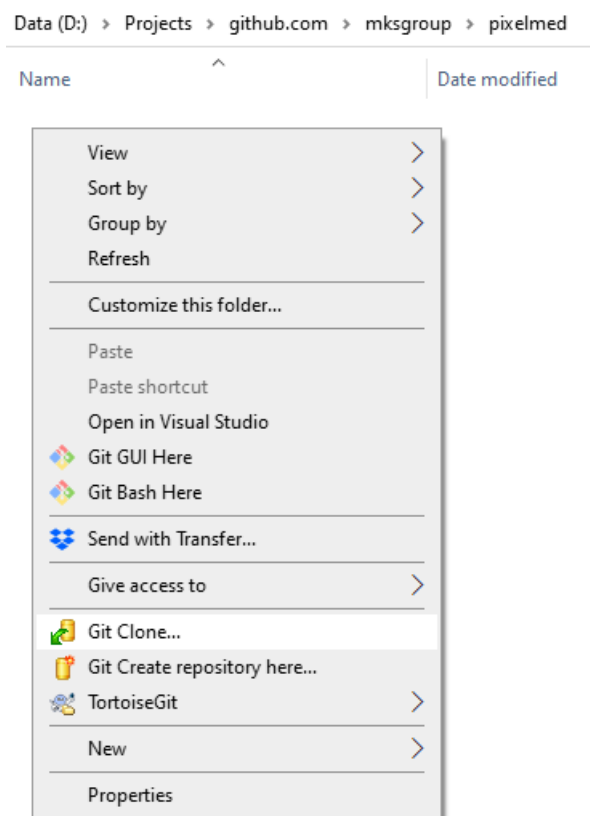
Để tổ chức thư mục rõ ràng thì tôi khuyến nghị bạn tạo thư mục như sau để chuẩn bị clone dự án “pixelmed” trên đường link <https://github.com/mksgroup>:

D:\Projects\github.com\mksgroup\pixelmed

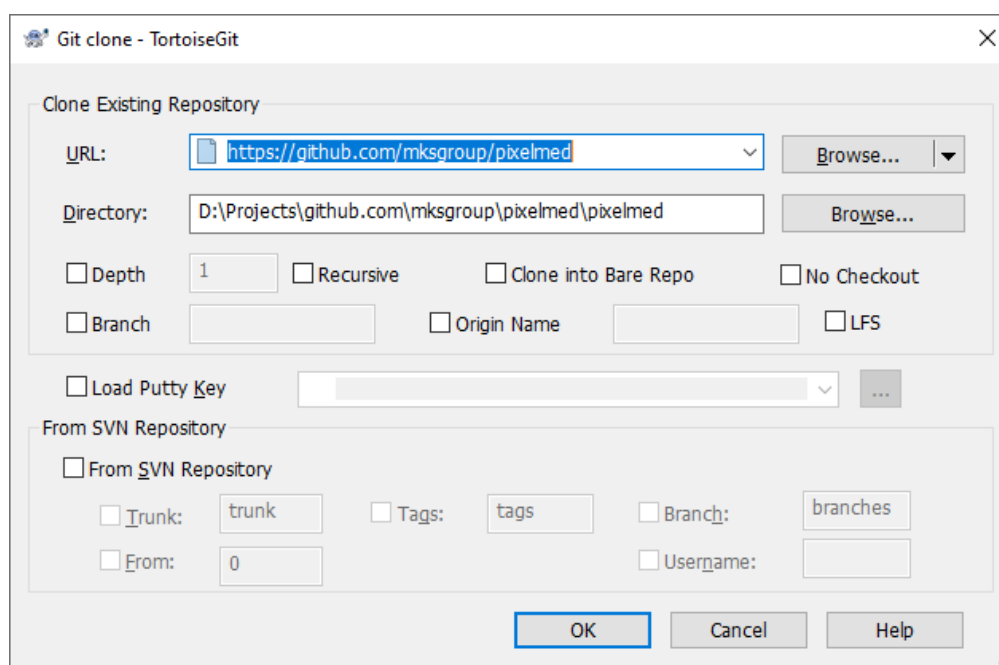
Khi bạn clone dự án khác từ github.com thì tạo thư mục tương tự, thay đổi phần in đậm.

## Bước 3: Clone dự án từ đường dẫn Internet về thư mục

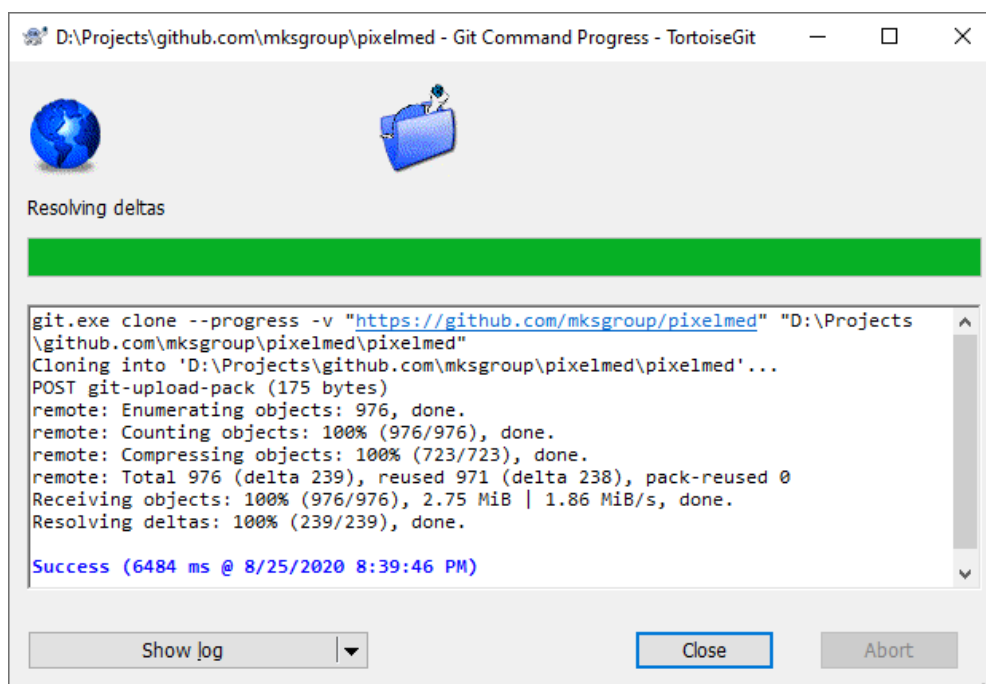
Bên trong thư mục dự án đang mở bằng chương trình File Explorer, nhấp phải chuột, chọn menu “**Git clone...**”.



Hộp thoại **Git clone** sẽ hiện với đường dẫn URL chính là đường dẫn đã copy trong bước 1. Trường hợp URL không đúng ý bạn thì có thể nhập lại.



Nhấn OK để thực hiện. Nếu may mắn thì kết quả không có lỗi như bên dưới.



Xem kết quả trong thư mục: D:\Projects\github.com\mksgroup\pixelmed\pixelmed

Trong đó sẽ có thư mục “.git”. Đây là thư mục ẩn (hidden) để chứa các dữ liệu đặc biệt để phần mềm git xử lý dữ liệu. Bạn không nên đụng tới thư mục này.

Bài tập:

Nếu bạn quen với Java thì hãy thử clone dự án Sakai từ địa chỉ sau:

<https://github.com/sakaiproject/sakai>

### Tóm tắt

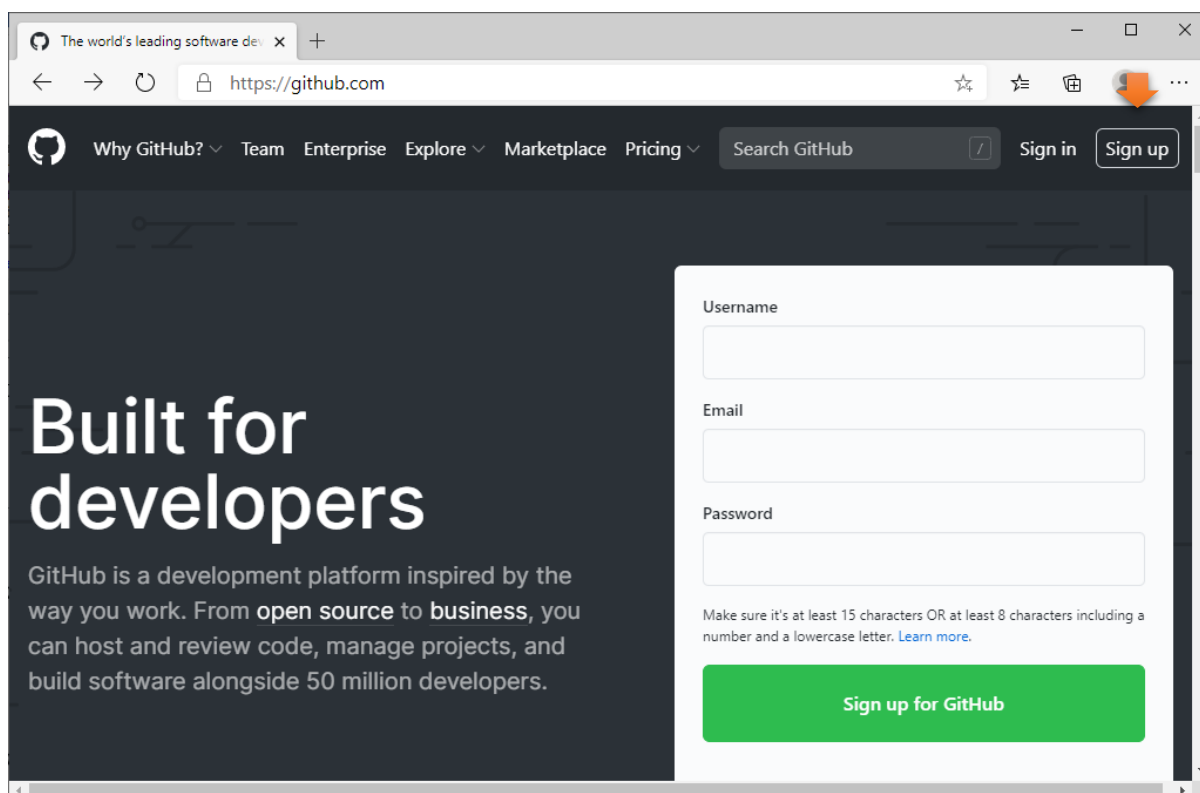
Như vậy bạn đã biết cách cài đặt phần mềm GIT trên máy tính chạy Windows và cũng biết cách clone dự án về máy của mình nếu biết được link của dự án trên github.com. Đối với gitlab.com, dev.azure.com hay các GIT server khác trên Internet hoặc trong công ty bạn thì cách thức clone cũng sẽ tương tự.

## Đăng ký tài khoản với GitServer

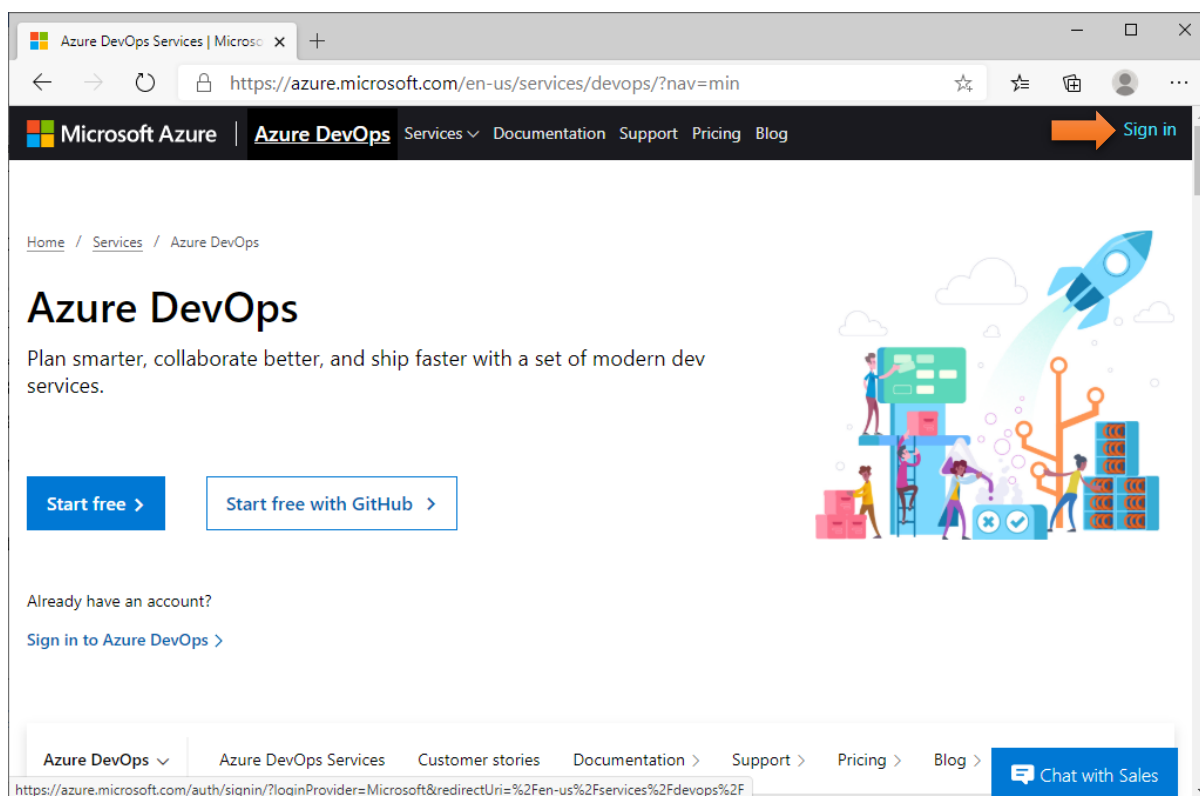
Thông thường nếu trong công ty bạn có hệ thống Git Server (thường là các anh chị IT sẽ dùng GitLab để triển khai) và bạn sẽ được đội IT thông báo tài khoản và có hướng dẫn sử dụng. Trong trường hợp bạn muốn tham gia các dự án trên github.com, gitlab.com, dev.azure.com mà các dự án này yêu cầu phải đăng nhập (thường là các dự án Private) thì bạn phải có tài khoản.

Thông thường các web site có hướng dẫn khá rõ ràng cho người dùng chưa có tài khoản.

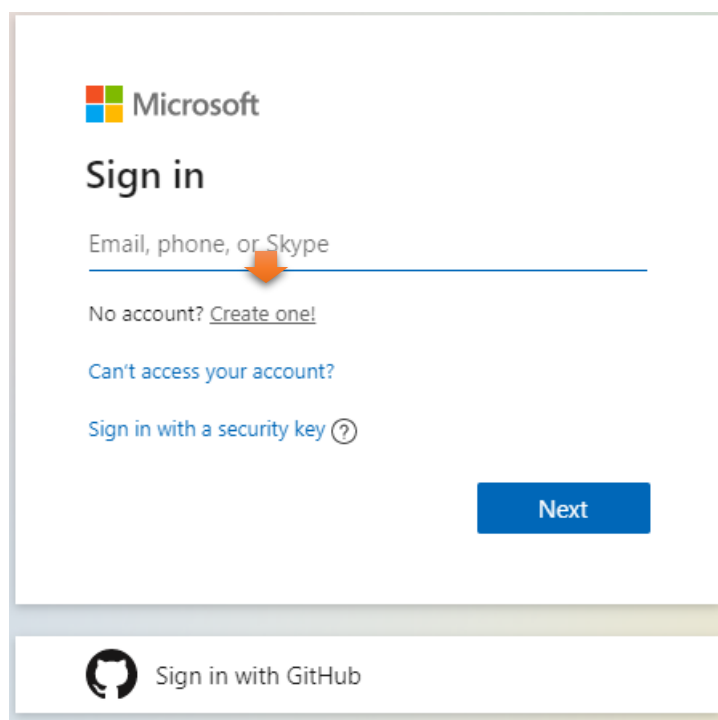
Ví dụ trong trên trang github.com thì có nút Sign up ở góc phải trên cho bạn tạo tài khoản mới.



Với dev.azure.com thì bấm vào nút Sign in ở góc phải trên.

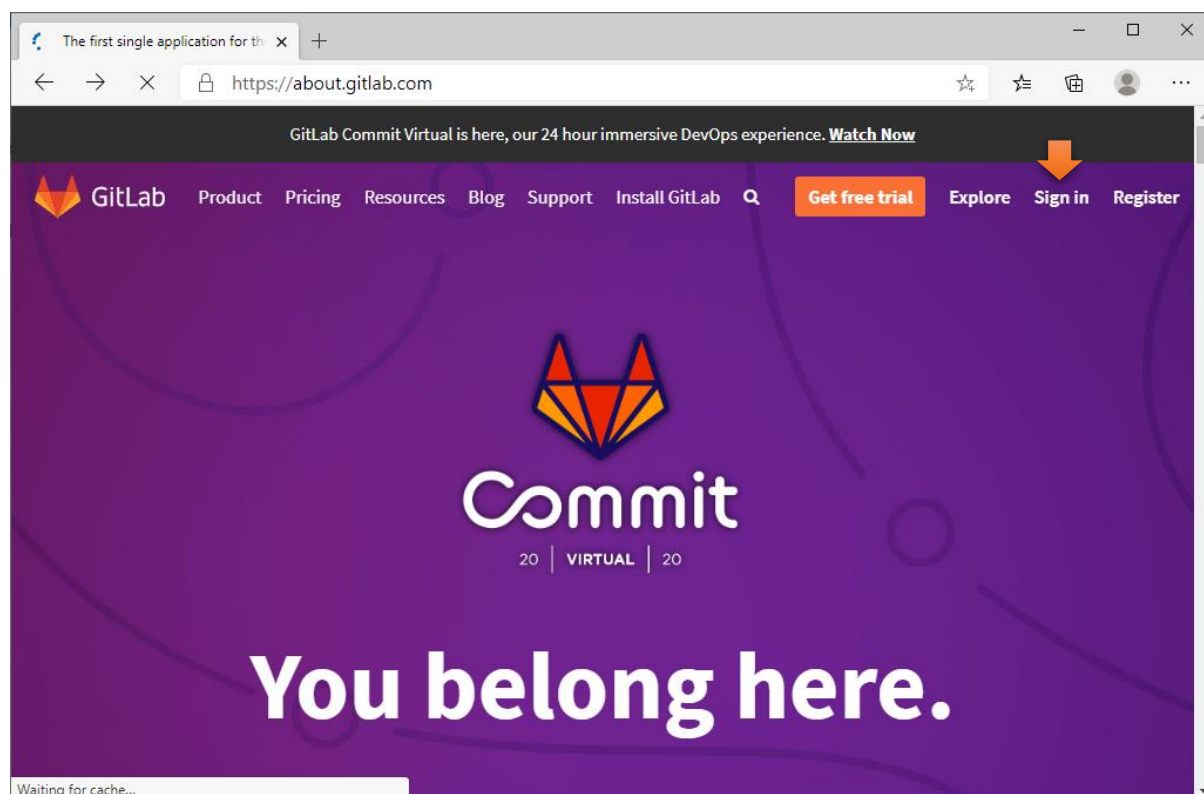


Trong màn hình đăng nhập (Sign in) thì có link để tạo account (link [Create one!](#)).

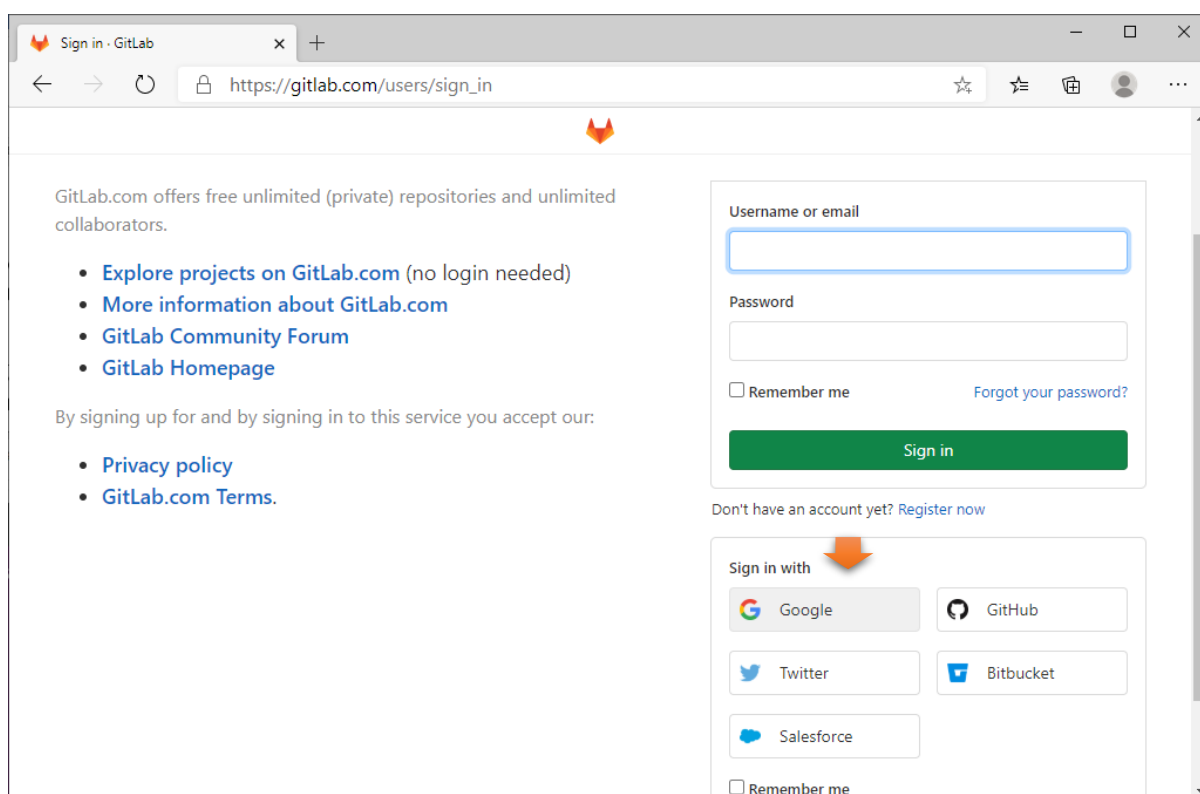


Điểm hay của dev.azure.com là cho phép dùng tài khoản của GitHub để đăng nhập (không phải tạo tài khoản mới): bấm vào link ở cuối vùng đăng nhập “Sign in with GitHub”.

Đối với gitlab.com cũng khá hay, có thể Sign in với tài khoản của Google (được hiểu là tài khoản Gmail).







Bạn cũng có thể khám phá từ màn hình đăng nhập của gitlab.com ở trên: gitlab.com cho phép đăng nhập bằng rất nhiều tài khoản sẵn có của bạn như: Google, GitHub, v.v...

Sau khi bạn đã có tài khoản của mình thì thông báo với người quản lý dự án để cho phép bạn tham gia vào dự án.

### Tóm tắt

Với vài gợi ý như trên thì tôi tin là bạn hoàn toàn có thể tự đăng ký tài khoản github.com hoặc dev.azure.com cho mình, hoặc dùng tài khoản Gmail để đăng nhập vào gitlab.com.

### Tham gia đóng góp cùng dự án

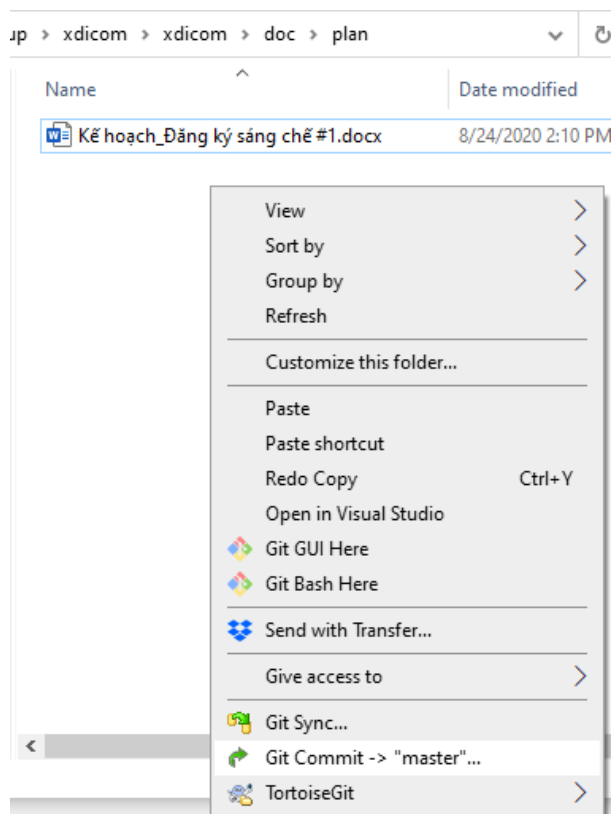
Đóng góp ở trong mục này đơn giản là bạn tạo ra file mới hoặc chỉnh sửa file có sẵn và nộp lên Git Server để cho đồng đội của mình thấy và chỉnh sửa tiếp (nếu cần).

Tình huống đơn giản nhất mà tôi sẽ minh họa cho các bạn là người quản lý dự án đã cấu hình cho phép bạn có quyền “đóng góp” trực tiếp vào thư mục dự án với hai thao tác Commit và Push. Các khái niệm và minh họa trong phần này đều áp dụng cho các Git Server mà tôi đã đề cập ở trên.

### *Commit file mới hoặc file chỉnh sửa vào git*

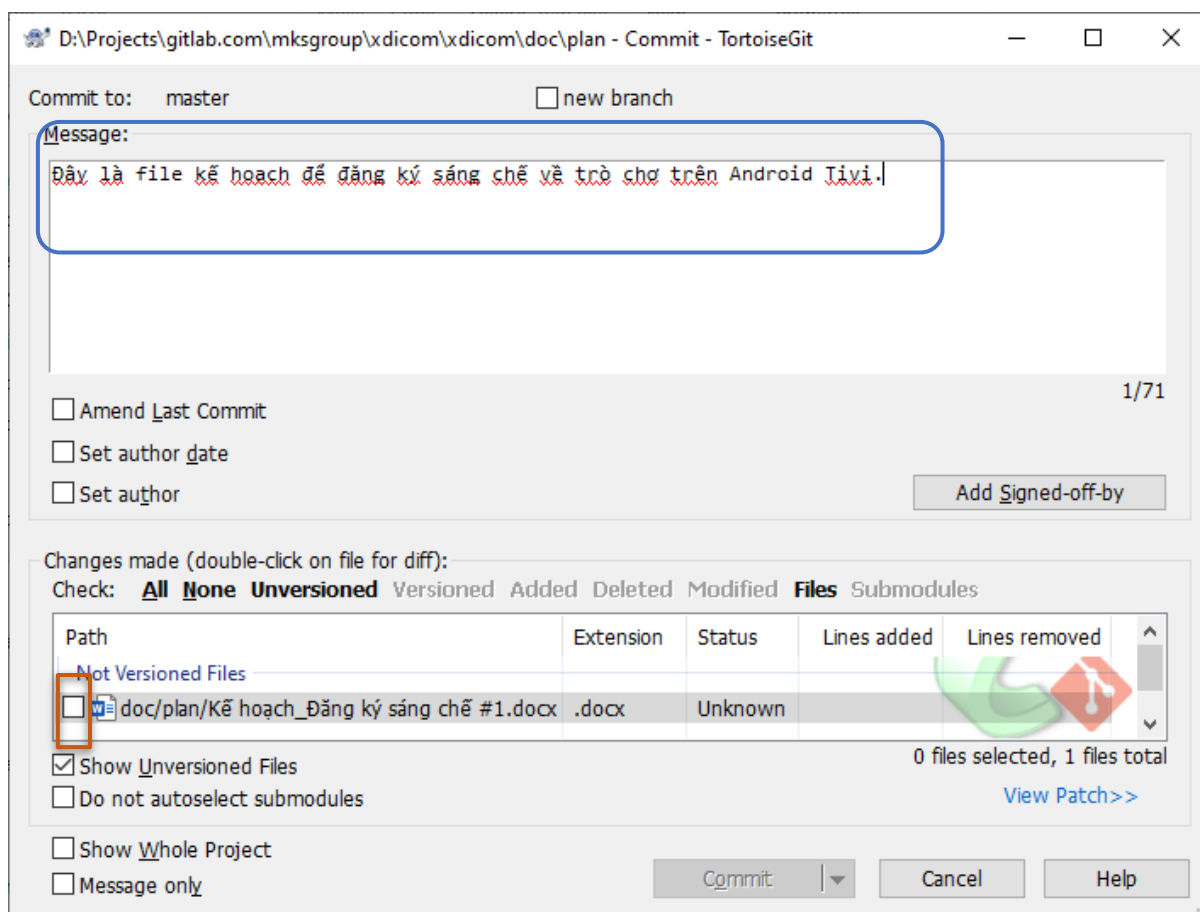
Sau khi bạn đã clone dự án về máy mà tôi đã hướng dẫn ở trên (trong trường hợp dự án cần bạn đăng nhập để clone thì hãy trao đổi với người quản lý dự án để cấp quyền cho bạn) thì bạn có thể làm việc trong thư mục của dự án trên máy của bạn. Ví dụ bạn tạo thêm thư mục và tài liệu của bạn thì hãy tạo nội dung và lưu vào thư mục như bình thường. Sau đó **nhấp phải chuột vào vùng trống** trong cửa sổ File Explorer (đang nói trên máy tính chạy Windows nhé).

Chọn menu: Git commit -> “master”...



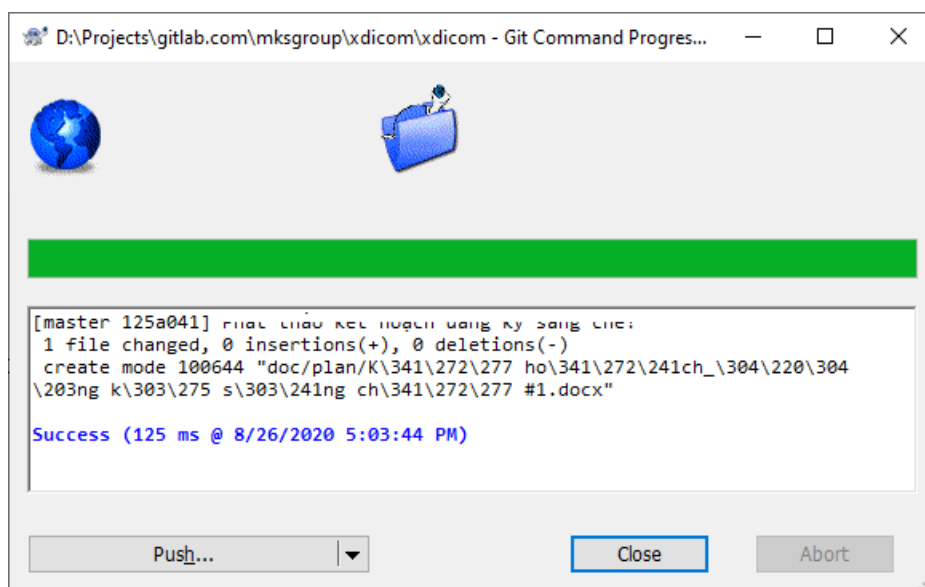
Sau đó cửa sổ Commit sẽ hiển thị như bên dưới. Có hai điểm chú ý bạn cần làm quen:

- ① Hãy chọn các file mà bạn muốn đóng góp cho nhóm bằng cách stick vào ô vuông bên trái của file (xem chỗ tôi vẽ hình chữ nhật).
- ② Trong nội dung Message hãy gõ giải thích vào để đồng đội của mình hiểu là file bạn đưa thêm vào có mục đích gì.

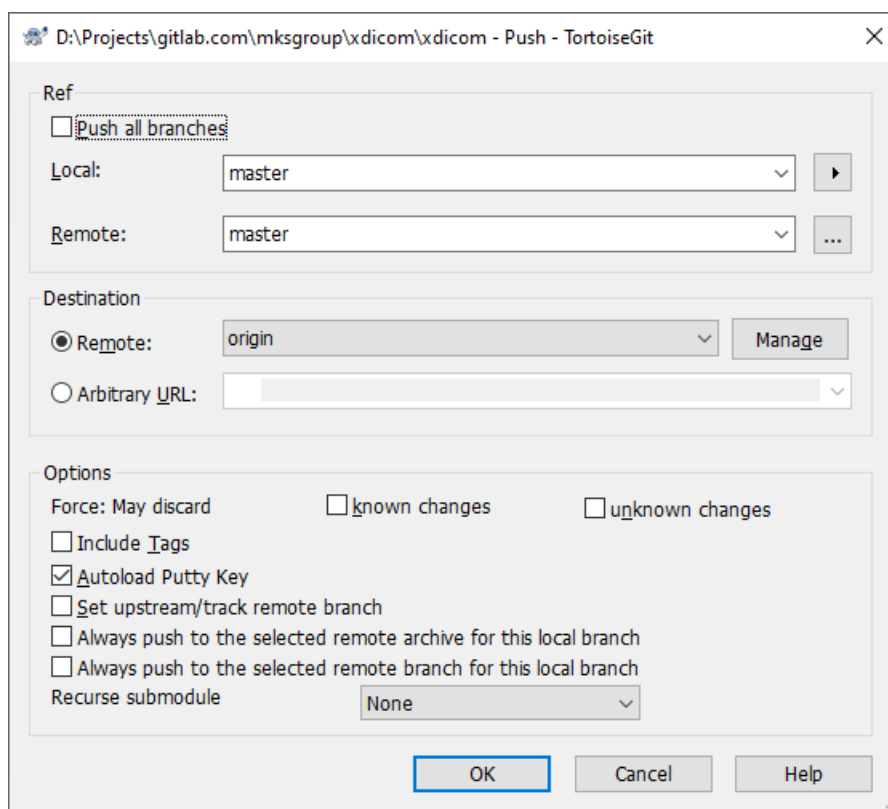


Sau khi chọn file và điền Message thì nút Commit sẽ nổi lên. Hãy bấm Commit để đồng ý đưa file nào vào trong dự án. Chú ý lúc nào file được chọn chưa có trên Git Server (hãy nhờ đồng đội vào website của Git Server xem cho chắc). Sau thao tác Commit này thì các file bạn chọn sẽ được đưa vào thư mục ẩn “.git” mà tôi có đề cập trong phần Clone với ghi chú là “Đừng có đụng vào”. Tức là phần mềm gitforwindow và TortoiseGit sẽ dùng thư mục ẩn “.git” để quản lý lịch sử các file bạn thêm/sửa/xóa,...trong thư mục dự án trên máy của bạn.

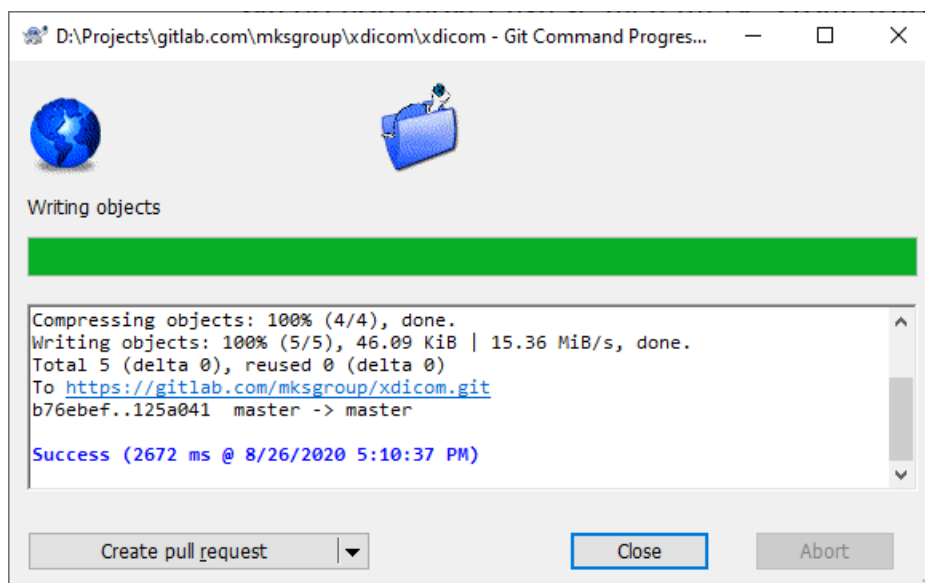
Sau khi quá trình Commit thành công thì hộp thoại sau sẽ hiển thị ra. Trong đó có nút Push. Để đưa các file đã Commit lên server thì bạn nhấn nút Push.



Sau đó hộp thoại Push sẽ hiển thị ra. Trong trường hợp đơn giản nhất mà người quản trị dự án đã tạo ra thì bạn chỉ cần nhấn OK.



Nếu bạn may mắn thì quá trình Push thành công như sau:

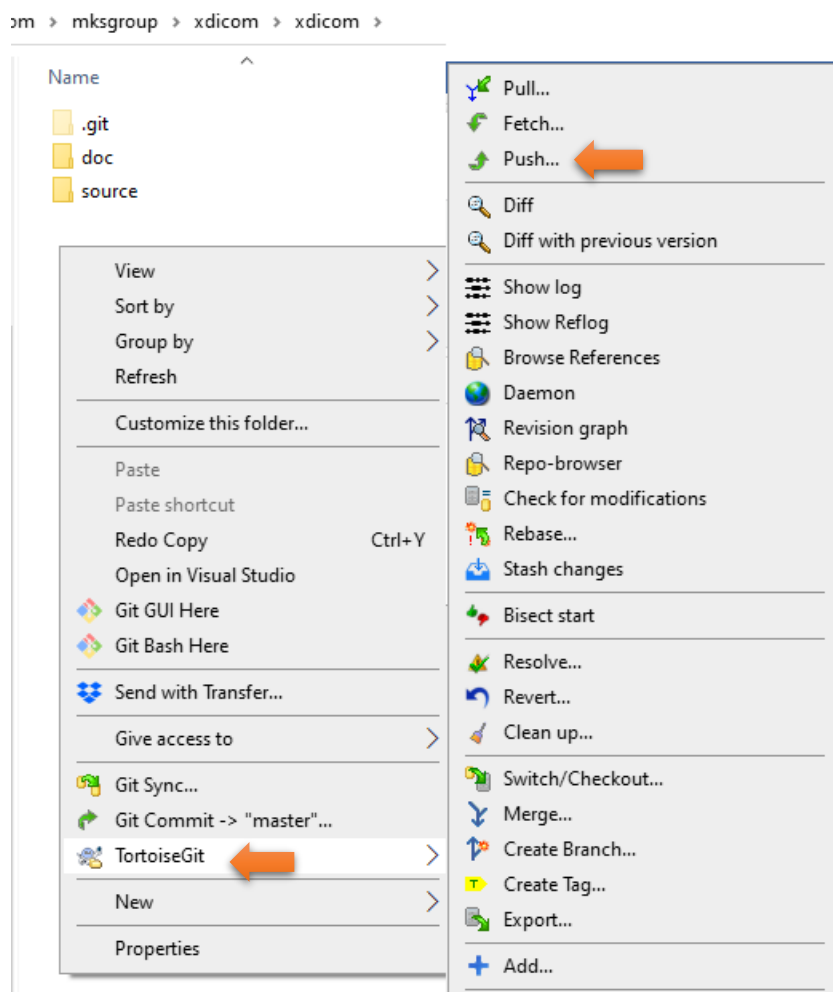


Cách tốt nhất để đảm bảo là bạn đã đóng góp file vào dự án là nhờ đồng đội của mình kiểm tra lại bằng cách vào website hoặc dùng chức năng Pull (tôi sẽ giải thích tiếp ở phần sau).

### *Sử dụng chức năng Push*

Ở phần trước bạn đã biết cách sử dụng chức năng Push ngay sau khi Commit. Trong trường hợp bạn chỉ Commit file (nhắc lại Commit là quá trình xác nhận bạn lưu các thay đổi vào trong thư mục ẩn “.git” trên máy bạn thôi chứ trên Git Server chưa có thay đổi gì) thì bạn có thể không cần Push mà vẫn làm việc và Commit tiếp nhiều file khác. Để đưa toàn bộ những gì bạn đã thay đổi (tức là đã Commit) lên Git Server thì trong cửa sổ File Explore, hãy nhấn phải chuột vào vùng trống rồi chọn menu:

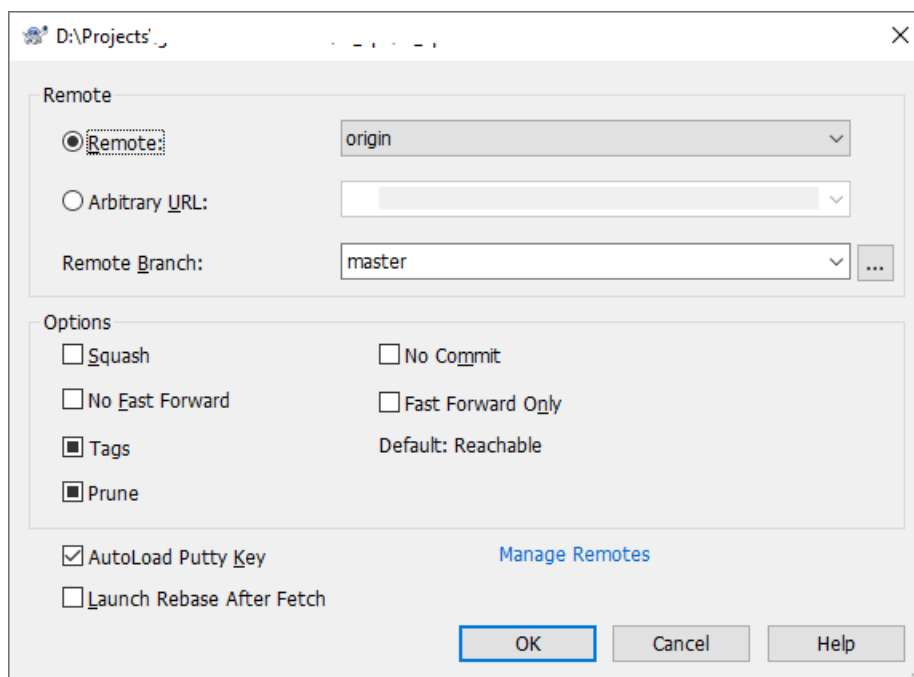
TortoiseGit > Push...



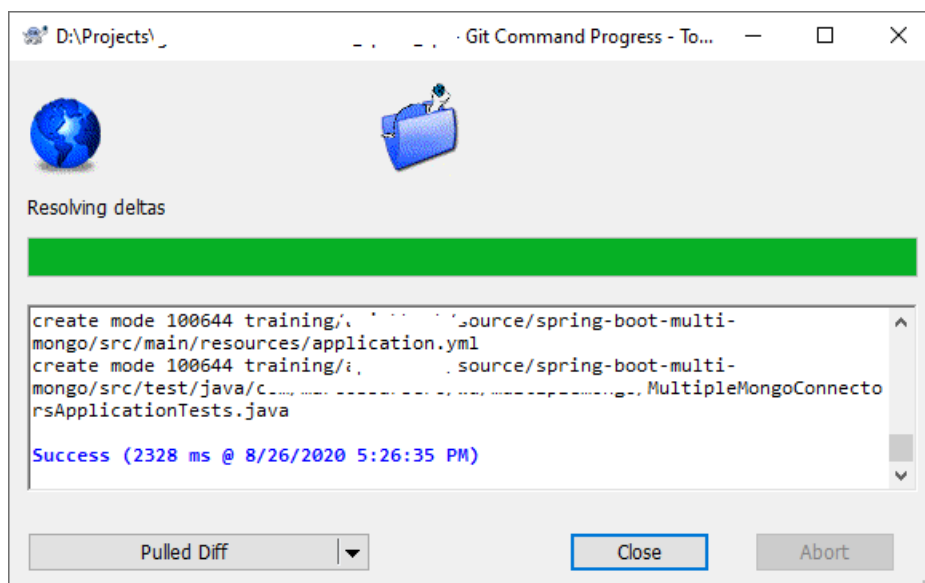
Sau đó nhấn nút OK và Close là xong (Mọi thứ người quản trị dự án sẽ chịu trách nhiệm. Nếu có sai sót gì đó thì người quản trị sẽ chịu trách nhiệm bởi vì họ không hướng dẫn cho bạn đủ chi tiết 😊).

### *Sử dụng chức năng Pull*

Để biết trong dự án có ai đóng góp gì đó mới không thì bạn dùng chức năng Pull. Các thao tác tương tự cách Push ở trên: Bấm phải chuột trong vùng trống của một thư mục nào đó trong dự án đang mở bằng File Explorer, chọn menu TortoiseGit > Pull.



Bấm nút OK để thực hiện Pull. Nếu may mắn thì bạn sẽ không gặp lỗi gì. Màn hình kết quả tựa như sau:



Muốn biết lần Pull này có khác biệt gì giữa các file mới trên Git Server và các file trên máy bạn lúc trước khi Pull thì nhấn nút “Pulled Diff”.

### Tóm tắt

Tới thời điểm này thì bạn đã biết cách Clone một dự án từ Git Server về nếu bạn biết đường dẫn URL git của dự án. Trường hợp cần đăng nhập trong lúc Clone thì bạn cần có tài khoản và người quản trị dự án đã cấp quyền cho bạn. Sau khi clone được dự án về máy thì bạn có thể tạo file mới hoặc chỉnh sửa file đang có, hoặc xóa file không cần thiết. Sau đó có thể dùng chức năng Commit và Push để

cập nhật các thay đổi lên Git Server. Và bạn cũng biết cách Pull dự án để cập nhật các thay đổi trên Git Server về máy của mình.



## Khảo sát ảnh và ma trận

```
import matplotlib.pyplot as plt
# read the image
path = 'D:/ai2020/data/image_5x5.png'
im = plt.imread(path)

# print matrix
print(im)

# show the image
plt.imshow(im)
plt.show()
```

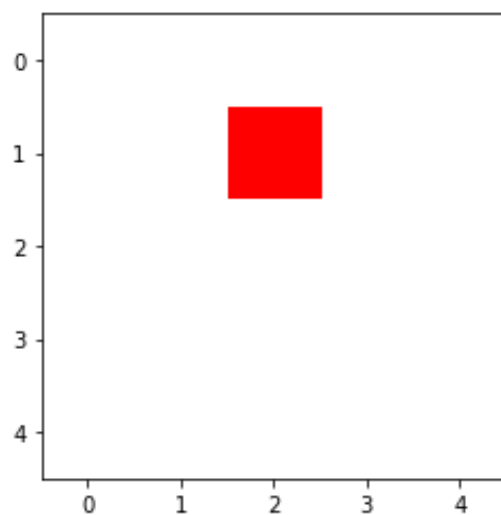
```
[[[1. 1. 1.]
  [1. 1. 1.]
  [1. 1. 1.]
  [1. 1. 1.]
  [1. 1. 1.]]

 [[1. 1. 1.]
  [1. 1. 1.]
  [1. 0. 0.]
  [1. 1. 1.]
  [1. 1. 1.]]

 [[1. 1. 1.]
  [1. 1. 1.]
  [1. 1. 1.]
  [1. 1. 1.]
  [1. 1. 1.]]

 [[1. 1. 1.]
  [1. 1. 1.]
  [1. 1. 1.]
  [1. 1. 1.]
  [1. 1. 1.]]

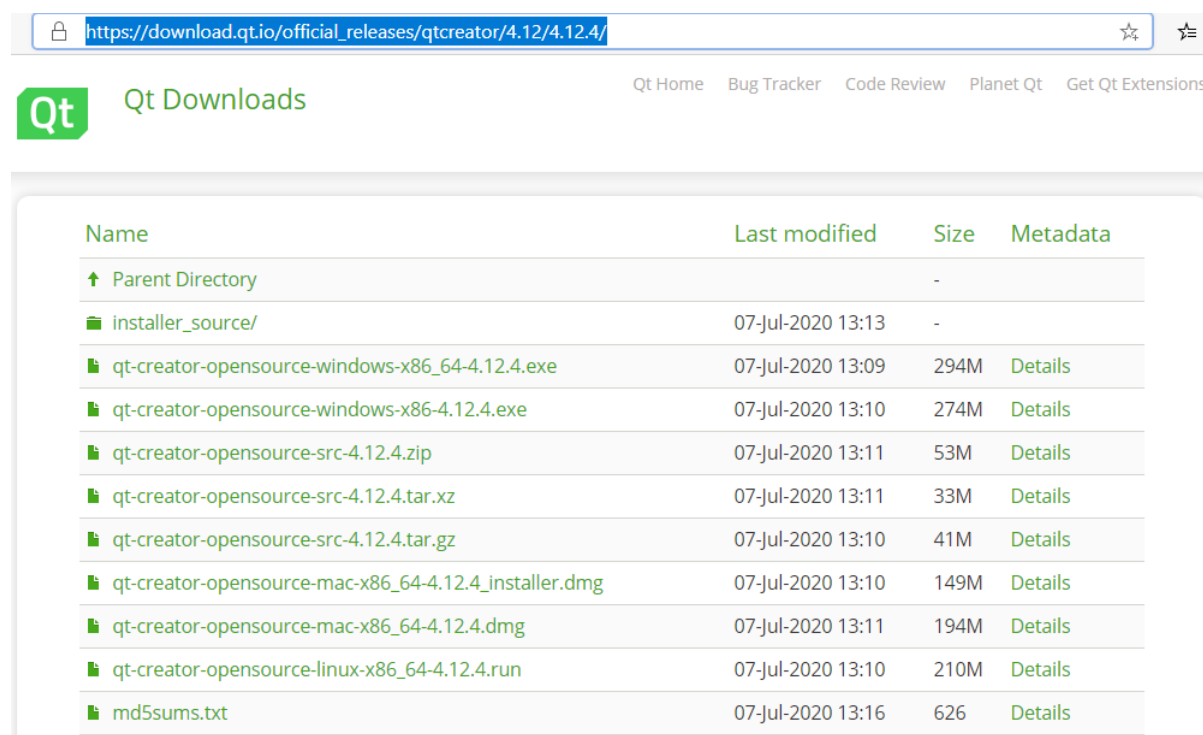
 [[1. 1. 1.]
  [1. 1. 1.]
  [1. 1. 1.]
  [1. 1. 1.]
  [1. 1. 1.]]]
```



## Phát triển ứng dụng với Python

### Tải phần mềm

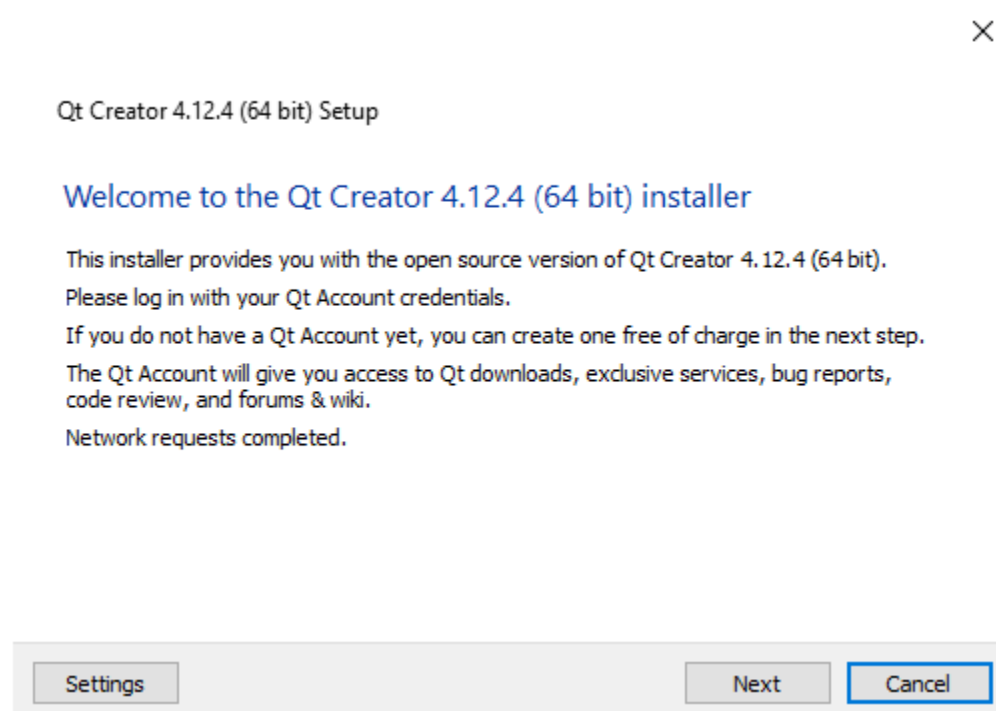
[https://download.qt.io/official\\_releases/qtcreator/4.12/4.12.4/](https://download.qt.io/official_releases/qtcreator/4.12/4.12.4/)



Tôi sử dụng phiên bản cho Windows 64 bit:

[qt-creator-opensource-windows-x86\\_64-4.12.4.exe](#)

### Cài đặt





Qt Creator 4.12.4 (64 bit) Setup

## Qt Account – Your unified login to everything Qt

Please log in to Qt Account

Login [Forgot password?](#)

Need a Qt Account?

Sign-up



Qt Creator 4.12.4 (64 bit) Setup

## Qt Open Source Usage Obligations

Qt Open Source version is available under GPLv 2, GPLv3 or LGPL v3.  
Please read and accept the Open Source Usage Obligations below. Reading the link below helps you choosing the right license for your project.

[Choosing the right license for your projects](#)

[Buy Qt](#)

### GPL v2, GPLv3 and LGPL v3 obligations

- You must not combine code developed with a commercial Qt license with code developed with an open source license of Qt in one project or product
- Provide a re-linking mechanism for Qt libraries
- Provide a license copy & explicitly acknowledge Qt use
- Make a Qt source code copy available for customers
- Accept that Qt source code modifications are non-proprietary
- Make "open" consumer devices
- Accept Digital Rights Management terms, please see the [GPL FAQ](#)
- Take special consideration when attempting to enforce software patents [FAQ](#)

☒ I have read and approve the obligations of using Open Source Qt

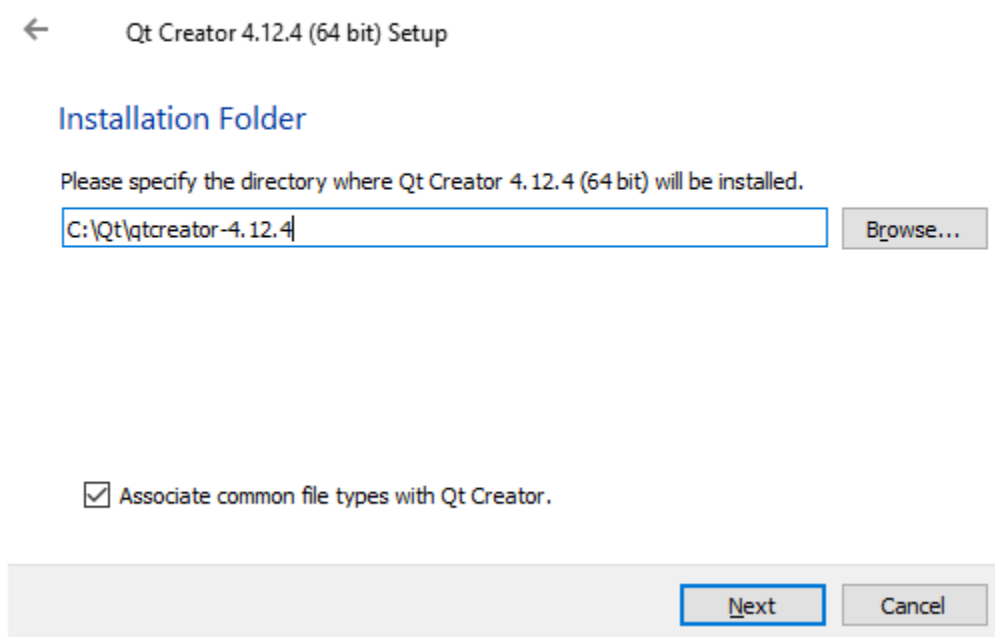
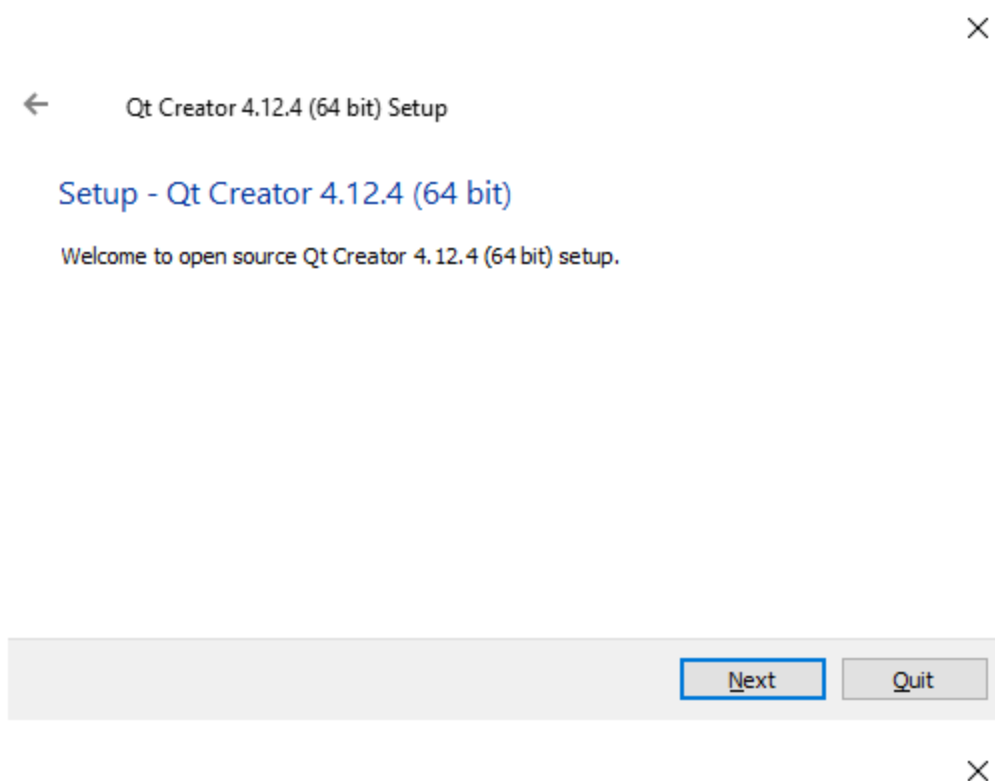
Please enter your company/business name

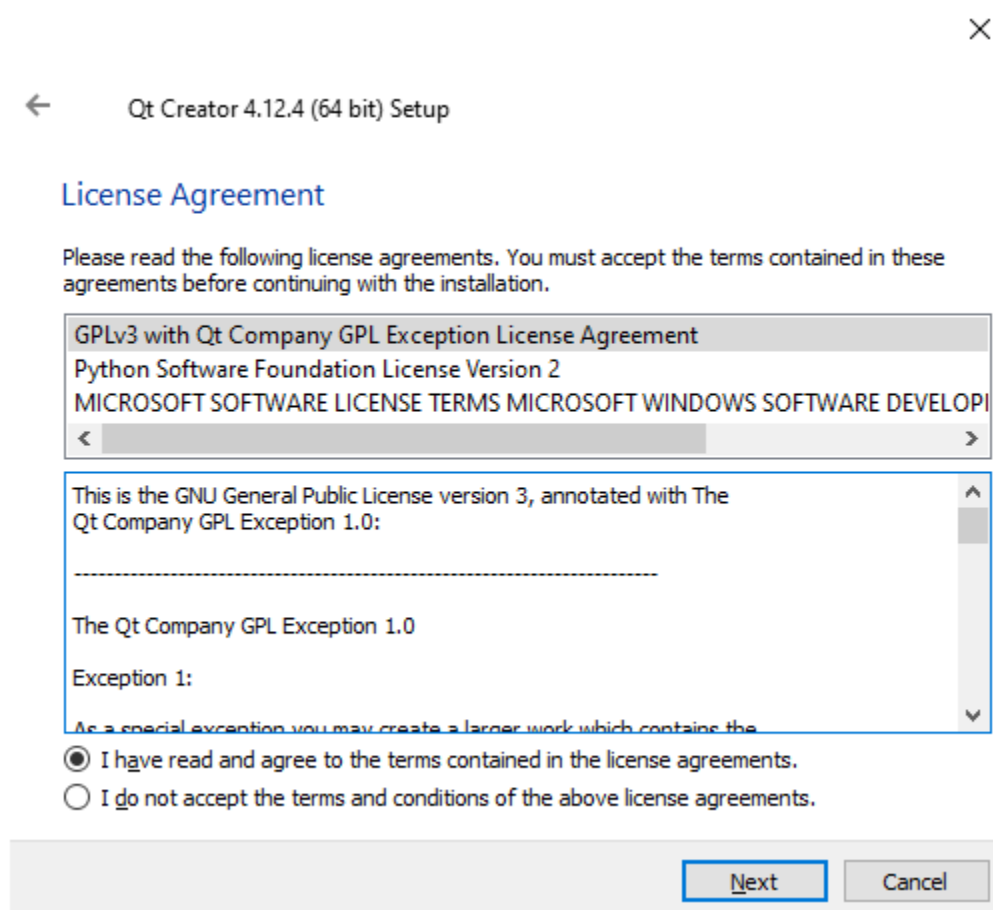
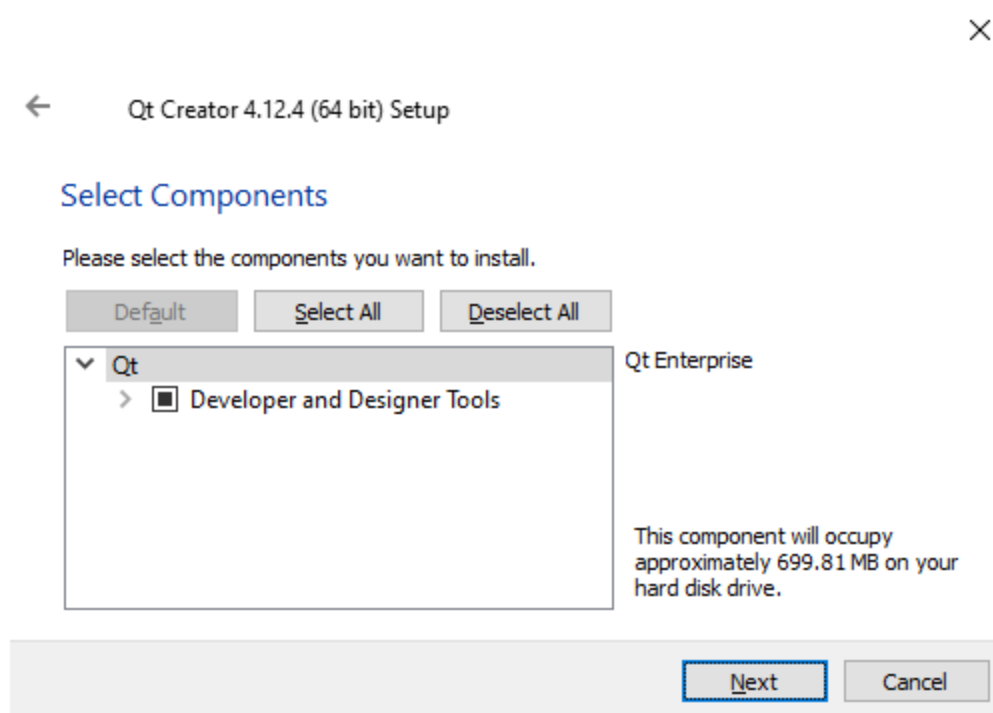
☒ I am an individual person not using Qt for any company

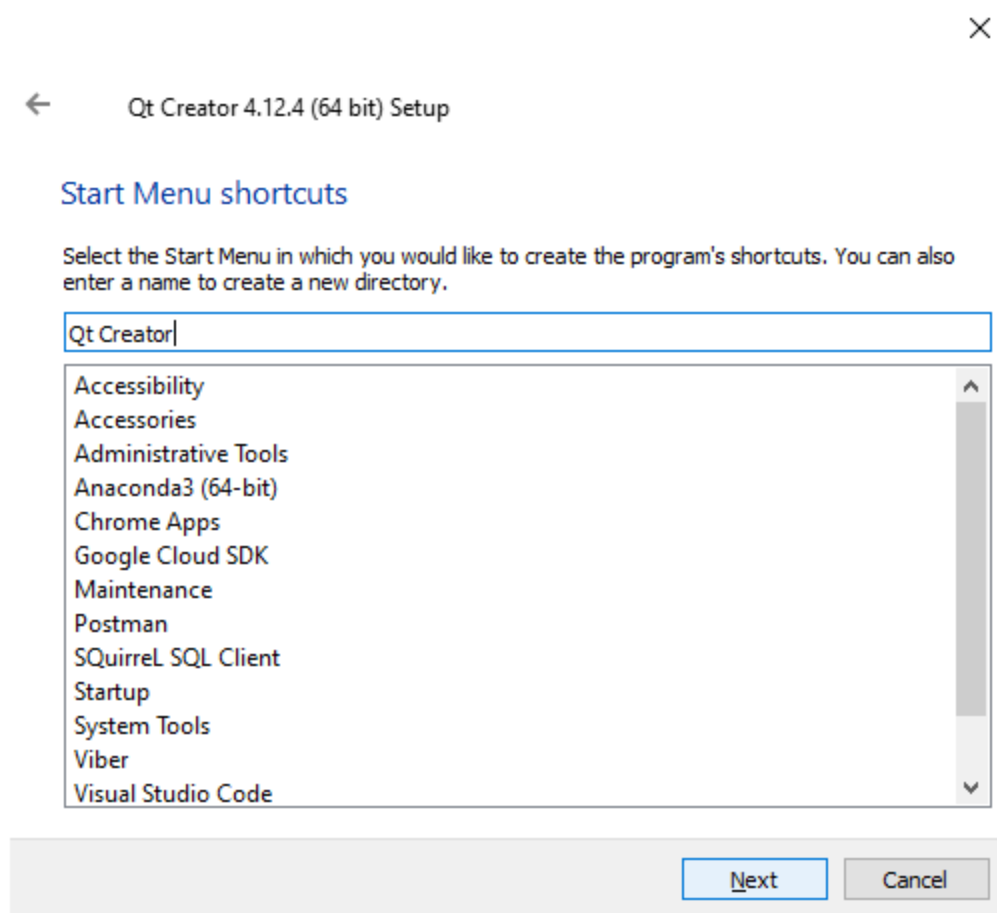
Settings

Next

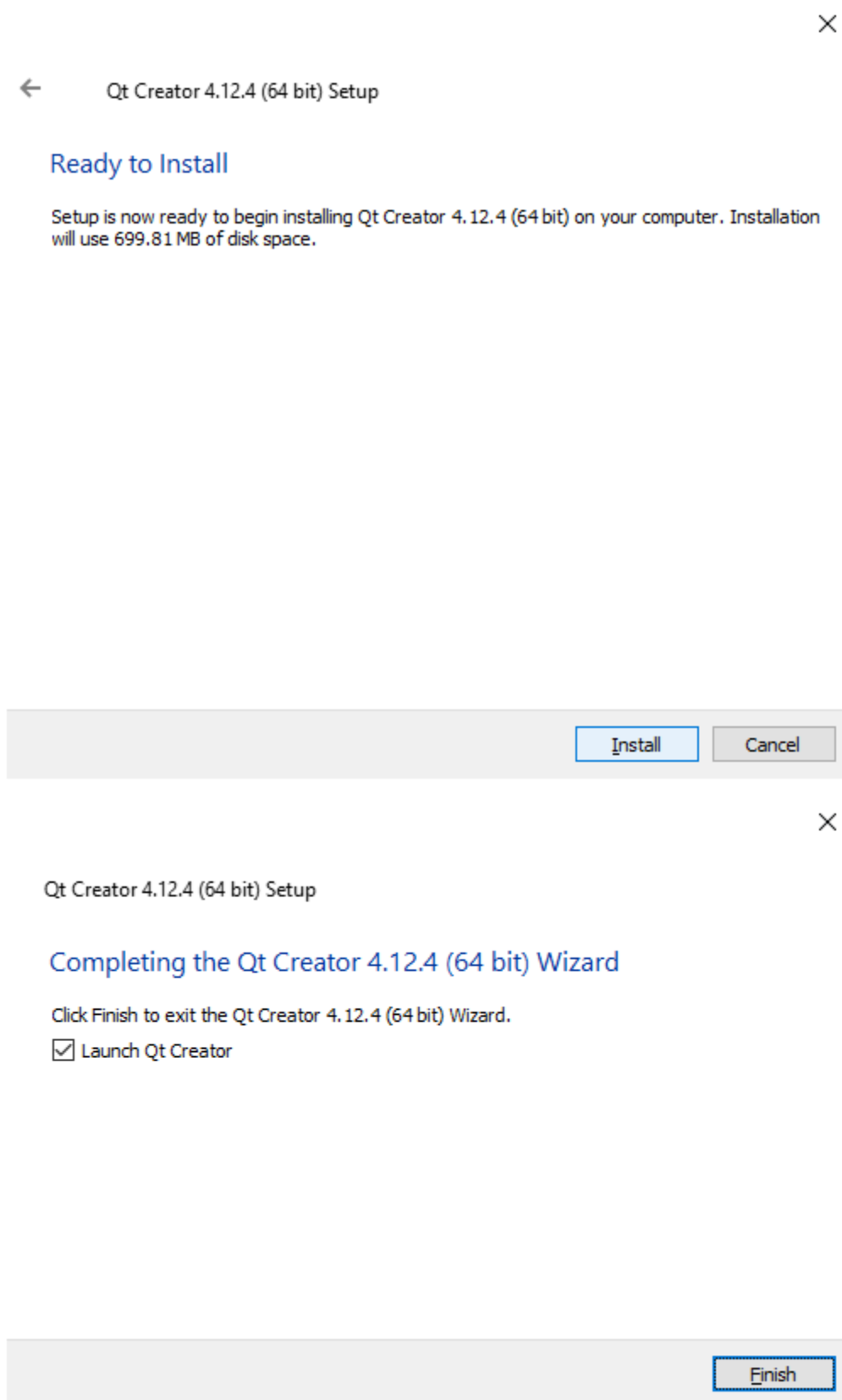
Cancel











## Trải nghiệm

## Sử dụng OpenCV để phát hiện đối tượng trong video

Bài này mở rộng Ngày 11 – Chủ đề Computer Vision. Trong ngày 11 bạn đã làm quen với code Python để nhận diện khuôn mặt, cũng như làm quen với mô hình CNN.

Bài viết này bổ sung thêm vài kỹ thuật để nhận diện đối tượng trong video.

### Chuẩn bị Python và công cụ lập trình PyCharm

- Python 3.9

Cài Python 3.9.5 tại thời điểm bài này được viết tại:

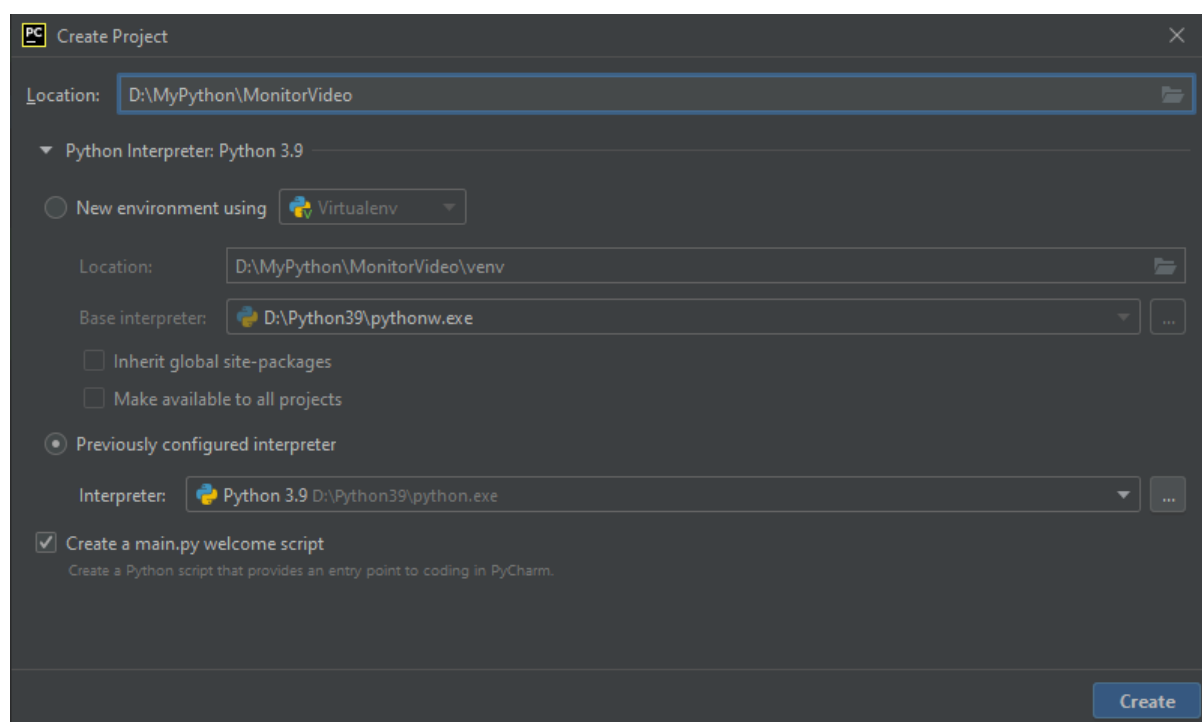
<https://www.python.org/downloads/release/python-395>

- PyCharm

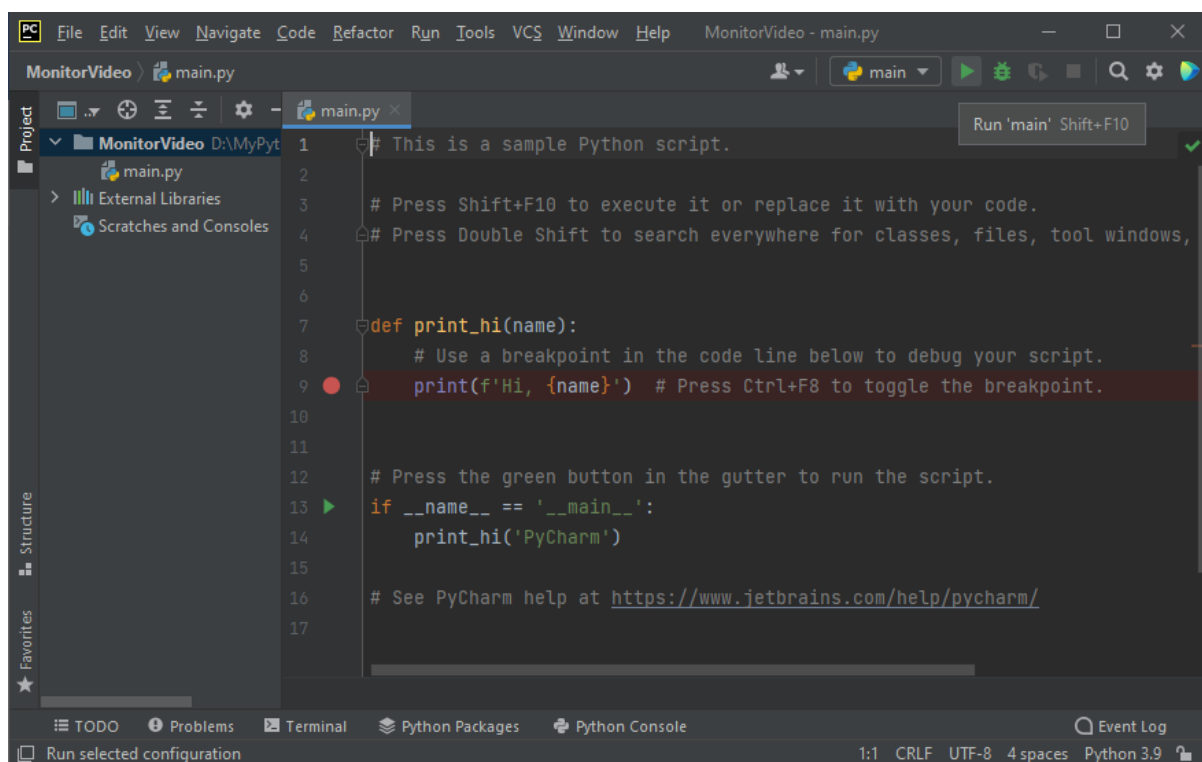
Tải bản Community 2021.2 tại:

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

### Tạo dự án

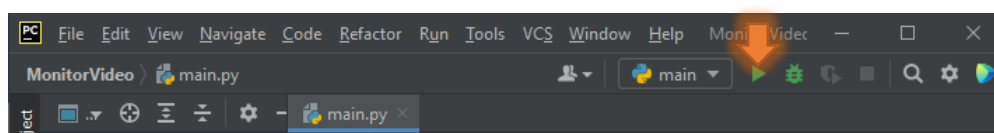


Khi Pycharm tạo dự án thì tạo luôn file main.py cho bạn:

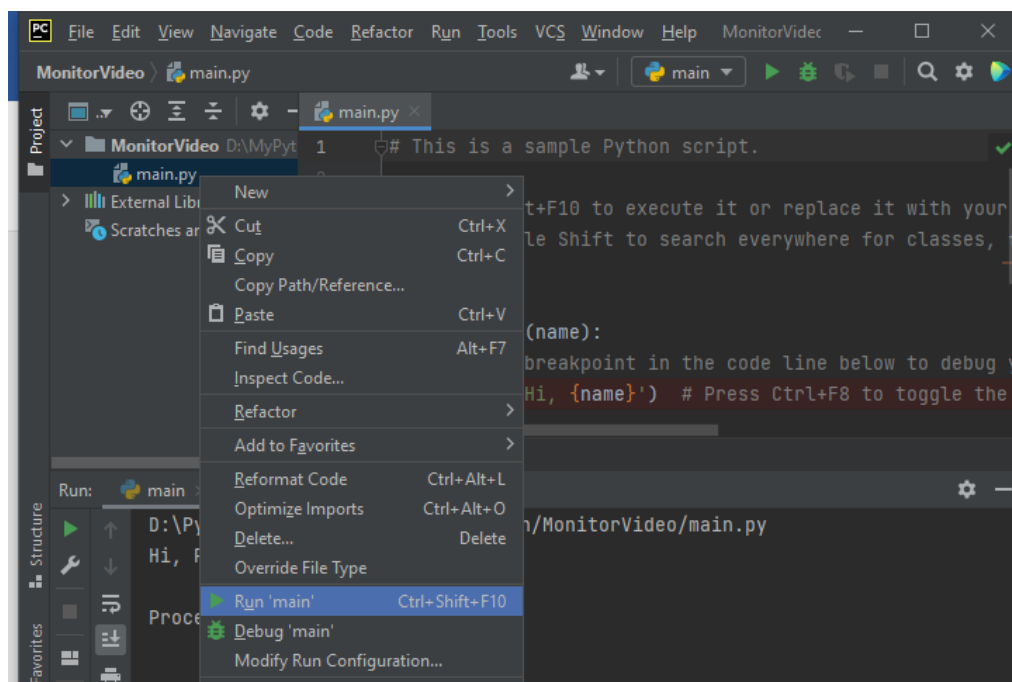


Để chạy chương trình thì có thể thực hiện một trong các cách sau:

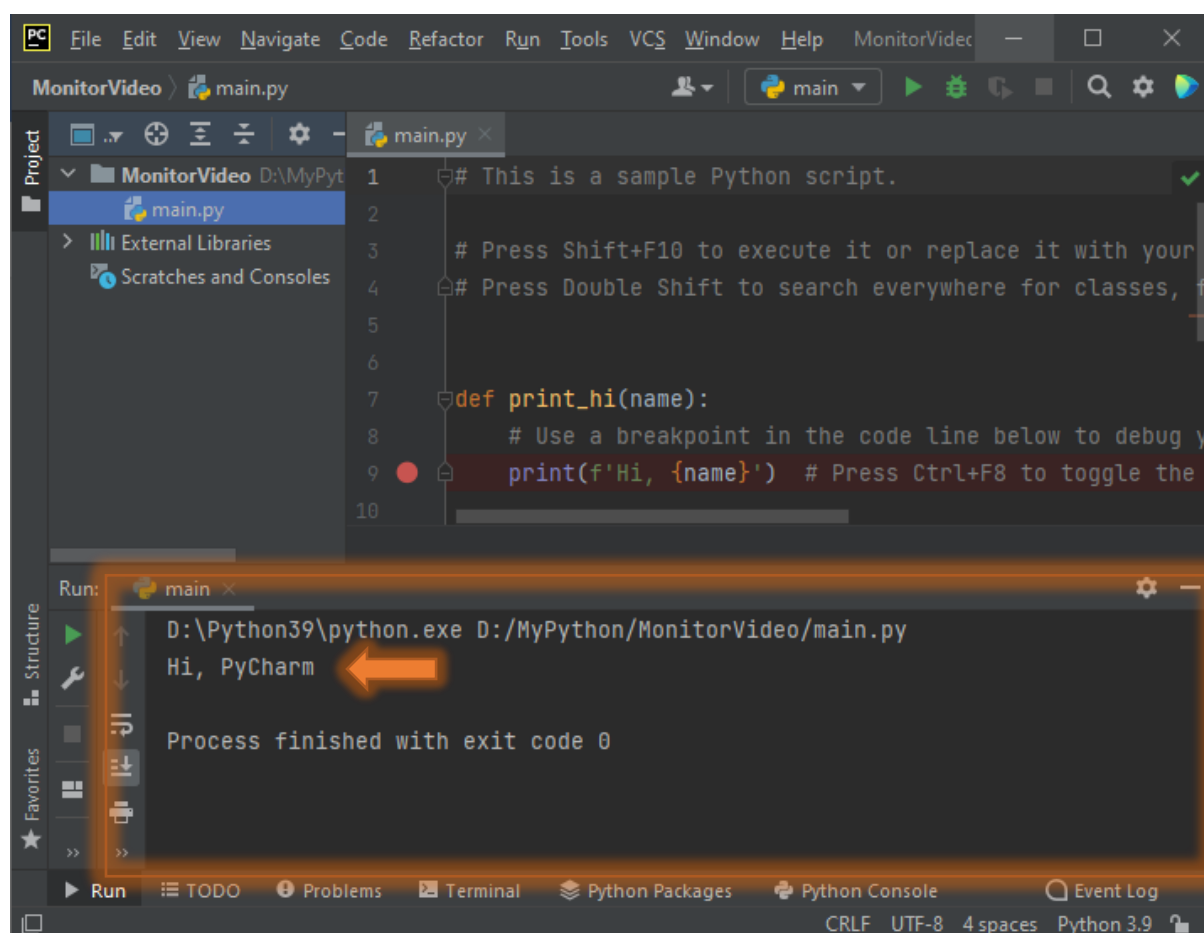
- Nhấn phím Shift + F10
- Click vào biểu  trên thanh công cụ:



- Nhấp phải chuột lên file main.py, chọn menu “Run main”



Kết quả chương trình sẽ hiển thị ra câu thông báo ở khung bên dưới:



### Nâng cấp code giai đoạn 1

Bạn sửa lại code file main.py như sau để làm quen với kỹ thuật:

- Lấy thông tin từ tham số dòng lệnh
- Lấy khung hình từ video, camera hoặc webcam

```
import argparse
import cv2

parser = argparse.ArgumentParser()
parser.add_argument("-i", "--input",
                    help="Video path, stream URI, or camera ID ",
                    default="https://thachln.github.io/video/rat.mp4")
args = parser.parse_args()

if args.input.isdigit():
    args.input = int(args.input)

cap = cv2.VideoCapture(args.input)

for res, frame in iter(cap.read, (False, None)):
    cv2.imshow("out", frame)
    if cv2.waitKey(1) == 27:
        cv2.waitKey(0)
        exit()
```

### Phân tích tham số dòng lệnh

- Sử dụng thư viện `argparse` để giúp bạn khi chạy chương trình có tham số một cách dễ dàng. Ví dụ khi chạy lệnh sau:

```
python main.py -help
```

thì sẽ ra kết quả hướng dẫn như:

```
usage: main.py [-h] [-i INPUT]

optional arguments:
  -h, --help            show this help message and exit
  -i INPUT, --input INPUT
                        Video path, stream URI, or
                        camera ID
```

Để làm được việc này thì đầu tiên cần khởi tạo đối tượng `parser` bằng lệnh sau:

```
parser = argparse.ArgumentParser()
```

Tiếp theo gọi hàm `add_argument` với 4 tham số như sau:

```
parser.add_argument("-i", "--input", help="...", default="...")
```

Tham số đầu tiên và thứ hai là kí hiệu cần phải gõ trên dòng lệnh. Gõ nhanh thì `-i ...`, gõ đầy đủ thì `--input ...`. Dấu `...` là giá trị của tham số.

Quay lại ý nghĩa hướng dẫn của chương trình này. Nó cần một tham số dòng lệnh `[-i INPUT]`. Tham số này có tên gọi là `input`.

- Dấu ngoặc vuông [] ý nói tham số input này là không bắt buộc. Nếu không truyền trên dòng lệnh thì input sẽ có giá trị mặc định trong chỗ “default=...”.

- Trong đoạn code ở trên thì Giá trị

`default=https://thachln.github.io/video/detect_rat.mp4`

Như vậy khi bạn chạy chương trình trong Pycharm (bằng cách nhấn phím Shift+F10) hoặc bằng lệnh “python main.py” thì chương trình sẽ xử lý input với giá trị:

[https://thachln.github.io/video/detect\\_rat.mp4](https://thachln.github.io/video/detect_rat.mp4).

Cách xử lý như thế nào sẽ bàn sau.

Để hiểu hơn 1 chút về ý nghĩa hàm `parser.add_argument` thì hãy thử sửa lại dòng lệnh đó như sau:

```
parser.add_argument("-i", "--input",
                    help="Đường dẫn file video, địa chỉ URI của stream video, hoặc camera ID ",
                    default="https://thachln.github.io/video/detect_rat.mp4")
```

Rồi chạy lại lệnh:

```
python main.py -help
```

```
usage: main.py [-h] [-i INPUT]

optional arguments:
  -h, --help            show this help message and exit
  -i INPUT, --input INPUT
                        Đường dẫn file video, địa chỉ
                        URI của stream video, hoặc camera ID
```

- Trong hướng dẫn có nói input là Đường dẫn video, hoặc đường dẫn URI của video đang phát (stream video), hoặc định danh của Camera (ID) trên máy bạn. Hãy thử các lệnh sau xem sao:

```
python main.py -i 0
```

Tham số -i 0 có nghĩa là chương trình sẽ đọc camera đầu tiên trong máy (thứ tự camera được đánh số từ 0)

Chương trình phân tích được số thứ tự sau tham số dòng lệnh -i là do dòng lệnh này:

```
if args.input.isdigit():
    args.input = int(args.input)
```

Ý nghĩa: nếu `input` là con số thì chương trình hiểu là bạn muốn truy cập camera. Dòng lệnh trong điều kiện `if` ở trên chuyển tham số `input` dạng chuỗi thành số nguyên.

### *Phân tích code lấy hình ảnh từ camera*

Xem đoạn code lấy và hiển thị khung hình từ `input` như sau:

```
cap = cv2.VideoCapture(args.input)

for res, frame in iter(cap.read, (False, None)):
    cv2.imshow("out", frame)
    if cv2.waitKey(1) == 27:
        cv2.waitKey(0)
        exit()
```

- Đầu tiên dùng hàm `VideoCapture` của OpenCV (thư viện `cv2` đã được import) với tham số là `input`. Nhắc lại `input` có thể là string chỉ đến file video hoặc là đường dẫn web chỉ tới file video; hoặc là một số nguyên chỉ số thứ tự của webcam hoặc camera trên máy bạn (gọi chung là camera)
- Vòng lặp `for` lấy ra từng khung hình của file video hoặc từ khung hình của camera.
- Lệnh `cv2.imshow("out", frame)` sẽ hiển thị khung hình lên cửa sổ có tên là `out`.
- Đoạn lệnh `if cv2.waitKey(1) == 27:` được chạy nếu bạn nhấn phím `Esc` thì chương trình sẽ dừng. Sau đó nhấn phím bất kỳ để kết thúc chương trình.

### *Tóm tắt giai đoạn 1.*

Như vậy bạn đã biết cách dùng thư viện `argparse` để nhận tham số từ dòng lệnh khi code chương trình Python. Bạn cũng biết cách dùng thư viện `opencv` với code Python để lấy từng khung hình trong một file video có sẵn, hoặc từ một link web, hoặc từ camera sau đó hiển thị từng khung hình lên màn hình máy tính.

Phần tiếp theo bên sẽ phân tích vật thể trong video.

## **Nâng cấp code giai đoạn 2**

### *Chuẩn bị dữ liệu*

Tải mô hình tại:

[http://download.tensorflow.org/models/object\\_detection/ssd\\_mobilenet\\_v2\\_coco\\_2018\\_03\\_29.tar.gz](http://download.tensorflow.org/models/object_detection/ssd_mobilenet_v2_coco_2018_03_29.tar.gz)

Lấy file “`frozen_inference_graph.pb`” lưu vào thư mục dự án.

Tải file:

[https://raw.githubusercontent.com/Qengineering/MobileNet\\_SSD\\_OpenCV\\_TensorFlow/master/ssd\\_mobilenet\\_v2\\_coco\\_2018\\_03\\_29.pbtxt](https://raw.githubusercontent.com/Qengineering/MobileNet_SSD_OpenCV_TensorFlow/master/ssd_mobilenet_v2_coco_2018_03_29.pbtxt)

lưu vào thư mục dự án.

### *Nâng cấp mã nguồn*

Lấy mã nguồn tại đây:

<https://drive.google.com/file/d/1A56nPdOhMP6bNWKvmiJgWdKHYJNzNCg2/view?usp=sharing>

Mã nguồn tham khảo mã nguồn gốc tại đây:

<https://github.com/PacktPublishing/OpenCV-4-with-Python-Blueprints-Second-Edition/tree/master/chapter10>



## Xử lý file pdf

### Đọc file pdf có encrypted

#### *Cài đặt thư viện pikepdf*

```
pip install pikepdf
```

```
import pikepdf
file_path = 'encrypted.pdf'

# Elegant, Pythonic API
with pikepdf.open(file_path) as pdf:
    num pages = len(pdf.pages)
    del pdf.pages[-1]
    pdf.save(decrypted.pdf')
```

#### *Trích xuất text*

```
import PyPDF2
import json
from fpdf import FPDF

file_path = 'descripted.pdf'

read_pdf = PyPDF2.PdfFileReader(file_path, strict=False)

print (read_pdf)

# get the read object's meta info
pdf_meta = read_pdf.getDocumentInfo()

# get the page numbers
num = read_pdf.getNumPages()
print ("PDF pages:", num)

# get the page numbers
num = read_pdf.getNumPages()
print ("PDF pages:", num)
```

```
# create a dictionary object for page data
all_pages = {}

# put meta data into a dict key
all_pages["meta"] = {}

# Use 'iteritems()' instead of 'items()' for Python 2
for meta, value in pdf_meta.items():
    print (meta, value)
    all_pages["meta"][meta] = value

# iterate the page numbers
for page in range(num):
    data = read pdf.getPage(page)
    #page mode = read pdf.getPageMode()

    # extract the page's text
    page_text = data.extractText()

    # put the text data into the dict
    all_pages[page] = page_text

    print(page_text)
```

### Xóa watermark khỏi file pdf

Vì nhu cầu nào đó mà bạn nhận được file pdf có ảnh nền mờ xuất hiện trong tất cả các trang của file pdf. Thông thường ảnh nền mờ được người tạo file pdf đưa vào dưới dạng watermark để có ý bảo vệ nội dung. Phần này không khuyến khích mọi người thay đổi nội dung file mà mình không phải là tác giả. Tuy nhiên vì nhu cầu nào đó mà bạn cần loại bỏ watermark ra khỏi file pdf thì đây là một trải nghiệm nên thử.

#### *Cài đặt thư viện*

Cài đặt thư viện pdf2image vào môi trường Python.

```
pip install pdf2image
```

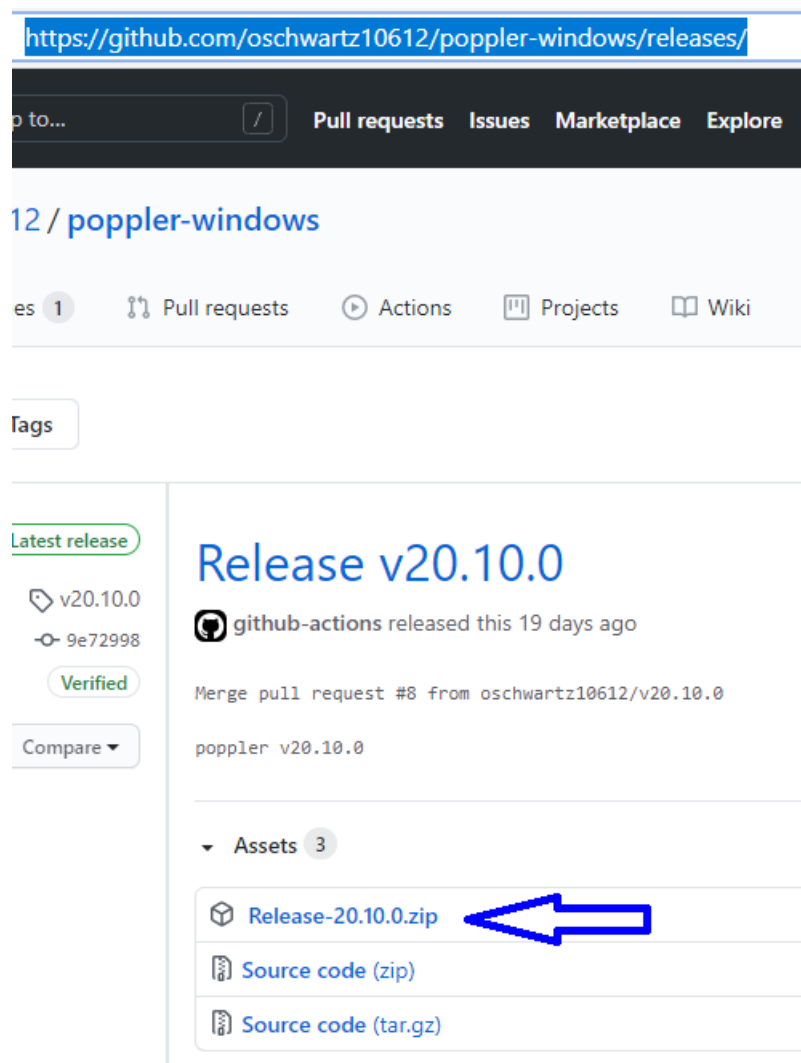
Thư viện này cần phần mềm poppler. Bạn có thể tải poppler tại

<https://github.com/oschwartz10612/poppler-windows/releases>

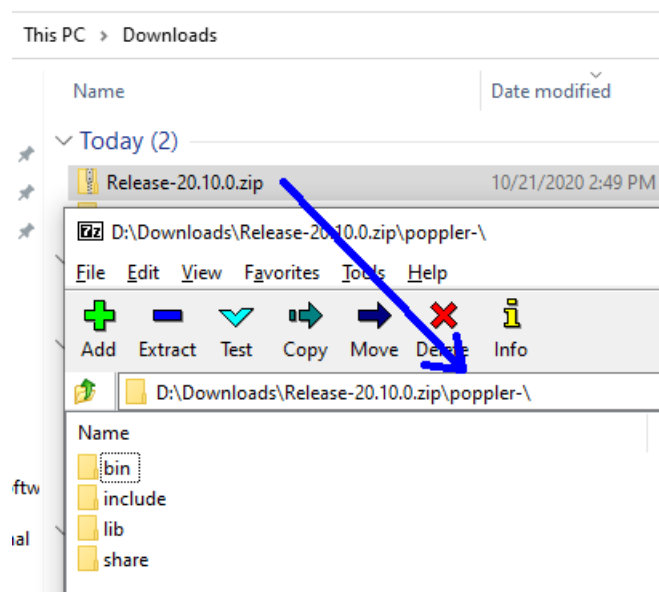
Sau khi tải về, giải nén ra thư mục bất kỳ rồi copy thư mục chứa thư mục bin vào thư mục D:\RunNow\poppler.

Cụ thể:

Sau khi tải phiên bản 20.10.0 (xem mục chỗ mũi tên trong hình bên dưới)



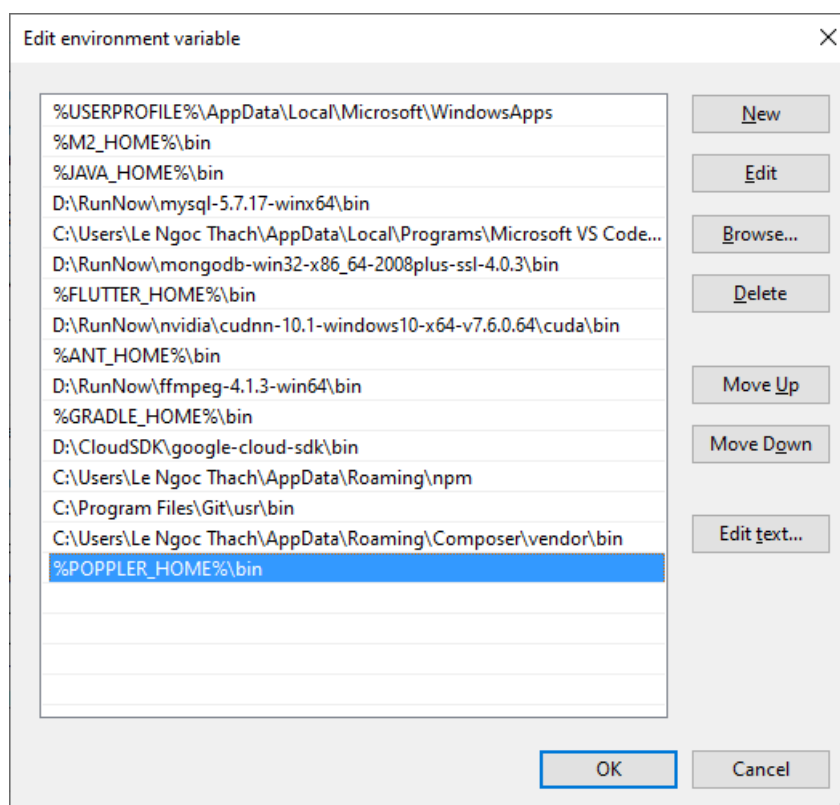
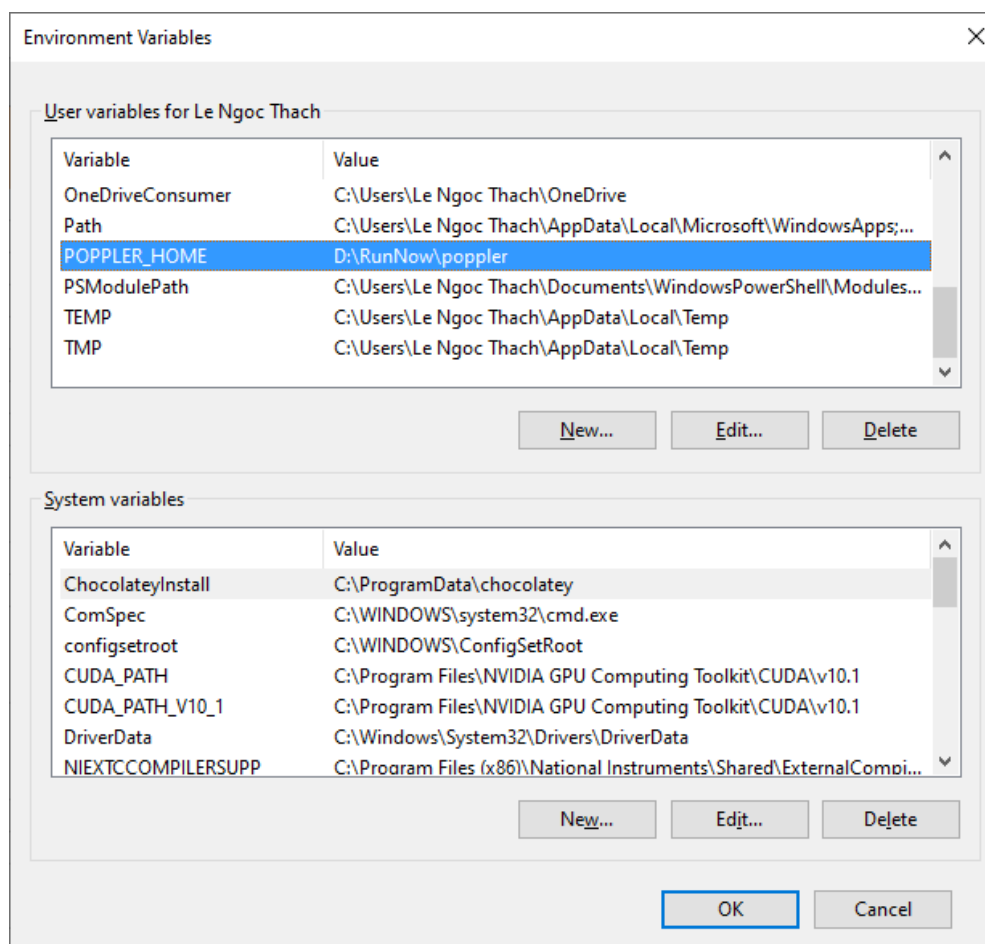
và dùng phần mềm 7-zip ở file ra thì bạn sẽ thấy trong file zip có thư mục "propler-" (có dấu trừ ở cuối tên thư mục).



Bạn copy thư mục "poppler-" này vào thư mục D:\RunNow và đổi tên lại là **"poppler"**.

Bước tiếp theo là cấu hình biến môi trường để máy tính (cụ thể là Windows) hiểu được đường dẫn chứa các lệnh trong phần mềm poppler này.

Hai hộp thoại bên dưới tương ứng là để cấu hình biến môi trường POPPLER\_HOME chỉ tới thư mục "D:\RunNow\poppler" và biến môi trường Path có thêm dòng % POPPLER\_HOME%/bin



Nếu bạn đang chạy Spyder thì cần tắt và mở lại để Spyder hiểu được các biến môi trường đã chỉnh ở trên.

### *Code Python import thư viện*

```
from skimage import io
from pdf2image import convert_from_path
import numpy as np
```

### *Đọc file pdf thành các ảnh*

Lệnh sau sẽ đọc file input.pdf (bạn cần thay đổi thành đường dẫn đầy đủ của file) thành mảng các ảnh, mỗi trang là một ảnh:

```
images = convert_from_path('input.pdf')
```

Dùng hàm `type(variable)` để xem kiểu dữ liệu của biến `images`

```
type(images)
```

```
list
```

Dùng hàm `len(array)` để xem số trang của file pdf, cũng là số phần tử của danh sách `images`.

```
print('Số ảnh, cũng là số trang của file pdf:', len(images))
```

Thử lấy ra ảnh đầu tiên và xem kích thước của ảnh:

```
image0 = np.array(images[0])
image0.shape
```

```
(2200, 1700, 3)
```

Kết quả của file pdf mà tôi thử thì trang đầu tiên được lưu dạng ảnh thì thuộc tính shape là (2200, 1700, 3) có nghĩa là: chiều cao 2200 điểm ảnh, chiều ngang là 1700 điểm ảnh và mỗi điểm ảnh có 3 màu (xem như là chiều sâu) Green-Red-Blue.

Tổng hợp đoạn code sau để thực hiện các việc:

- Đọc file pdf của đề thi TOEIC mẫu có ảnh nền (gọi là watermark)
- Tách mỗi trang pdf thành một image
- Xử lý từng image để loại bỏ điểm ảnh của watermark.
- Lưu từng image vào thư mục

```
from skimage import io
```

```
from pdf2image import convert_from_path
import numpy as np

filePath = 'D:/Temp/ETS_2020_RC.pdf'
images = convert_from_path(filePath)

print('Số ảnh, cũng là số trang của file pdf:', len(images))

def select_pixel2(r,g,b):
    if r > 175 and r < 250 and g > 175 and g < 250 and b > 175 and b < 250:
        return True
    else:
        return False

def handle(imgs):
    for i in range(imgs.shape[0]):
        for j in range(imgs.shape[1]):

            if select_pixel2(imgs[i][j][0],imgs[i][j][1],imgs[i][j][2]):
                imgs[i][j][0] = imgs[i][j][1] = imgs[i][j][2] = 255

    return imgs

index = 0
for img in images:
    index += 1
    img = np.array(img)
    img = handle(img)
    io.imsave('D:/Temp/ETS_2020_RC/' + str(index) + '.jpg', img)
```

**Kết quả:**

## READING TEST

In the Reading test, you will read a variety of texts and answer several different types of reading comprehension questions. The entire Reading test will last 75 minutes. There are three parts, and directions are given for each part. You are encouraged to answer as many questions as possible within the time allowed.

You must mark your answers on the separate answer sheet. Do not write your answers in your test book.

## PART 5

**Directions:** A word or phrase is missing in each of the sentences below. Four answer choices are given below each sentence. Select the best answer to complete the sentence. Then mark the letter (A), (B), (C), or (D) on your answer sheet.

101. Departmental restructuring will be discussed at the ----- monthly meeting.  
(A) next  
(B) always  
(C) soon  
(D) like
102. To keep ----- park beautiful, please place your nonrecyclables in the available trash cans.  
(A) our  
(B) we  
(C) us  
(D) ours
103. Mr. Hardin ----- additional images of the office building he is interested in leasing.  
(A) informed  
(B) asked  
(C) advised  
(D) requested
104. A team of agricultural experts will be brought ----- to try to improve crop harvests.  
(A) because  
(B) either  
(C) between  
(D) together
105. The board of Galaxipharm ----- Mr. Kwon's successor at yesterday's meeting.  
(A) named  
(B) granted  
(C) founded  
(D) proved
106. If your parking permit is damaged, bring it to the entrance station for a -----.  
(A) replacement  
(B) replacing  
(C) replace  
(D) replaces
107. Mr. Ahmad decided to reserve a private room for the awards dinner ----- the restaurant was noisy.  
(A) rather than  
(B) in case  
(C) such as  
(D) unless
108. Ms. Jones has provided a ----- estimate of the costs of expanding distribution statewide.  
(A) conserve  
(B) conservee  
(C) conservative  
(D) conservatively

20

## READING TEST

In the Reading test, you will read a variety of texts and answer several different types of reading comprehension questions. The entire Reading test will last 75 minutes. There are three parts, and directions are given for each part. You are encouraged to answer as many questions as possible within the time allowed.

You must mark your answers on the separate answer sheet. Do not write your answers in your test book.

## PART 5

**Directions:** A word or phrase is missing in each of the sentences below. Four answer choices are given below each sentence. Select the best answer to complete the sentence. Then mark the letter (A), (B), (C), or (D) on your answer sheet.

101. Departmental restructuring will be discussed at the ----- monthly meeting.  
(A) next  
(B) always  
(C) soon  
(D) like
102. To keep ----- park beautiful, please place your nonrecyclables in the available trash cans.  
(A) our  
(B) we  
(C) us  
(D) ours
103. Mr. Hardin ----- additional images of the office building he is interested in leasing.  
(A) informed  
(B) asked  
(C) advised  
(D) requested
104. A team of agricultural experts will be brought ----- to try to improve crop harvests.  
(A) because  
(B) either  
(C) between  
(D) together
105. The board of Galaxipharm ----- Mr. Kwon's successor at yesterday's meeting.  
(A) named  
(B) granted  
(C) founded  
(D) proved
106. If your parking permit is damaged, bring it to the entrance station for a -----.  
(A) replacement  
(B) replacing  
(C) replace  
(D) replaces
107. Mr. Ahmad decided to reserve a private room for the awards dinner ----- the restaurant was noisy.  
(A) rather than  
(B) in case  
(C) such as  
(D) unless
108. Ms. Jones has provided a ----- estimate of the costs of expanding distribution statewide.  
(A) conserve  
(B) conservee  
(C) conservative  
(D) conservatively



## Khảo sát file âm thanh

Tôi chuẩn bị một file âm thanh mẫu bằng cách lấy câu đầu tiên “What is machine learning” trong bài giới thiệu của Adrew Ng tên Coursera<sup>21</sup>. File này được lưu tại:

[https://thachln.github.io/datasets/What\\_is\\_machine\\_learning.wav](https://thachln.github.io/datasets/What_is_machine_learning.wav)

Việc đầu tiên là bạn cần tải file này về máy. Ví dụ tôi lưu tại:

D:/ai2020/data/What\_is\_machine\_learning.wav

### Cài thư viện

```
pip install scipy
```

### Đọc file âm thanh

```
import scipy.io.wavfile as wav

audio_file = 'D:/ai2020/data/What is machine learning.wav'
(samplerate, signal) = wav.read(audio_file)

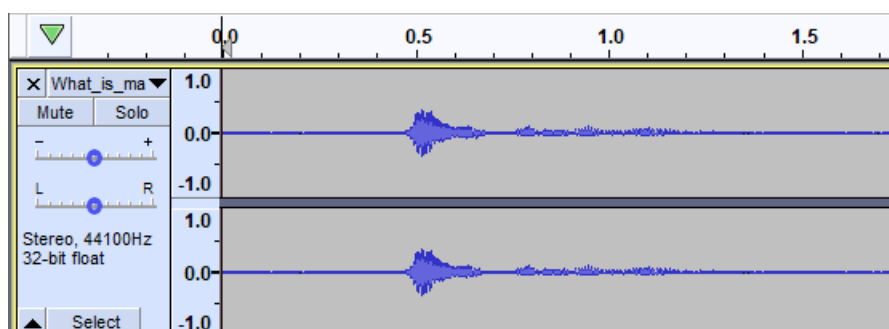
print('samplerate:', samplerate)
print('signal:', signal)
```

```
samplerate: 44100
signal: [[ 0  0]
 [ 0 -1]
 [ 0  2]

...
[57 56]
[56 54]
[58 61]]
```

Số 44100 của samplerate cho biết file wav này tần số lấy mẫu là 44100. Tức là trong một giây thì có 44100 mẫu tín hiệu.

Giá trị signal cho thấy đây là matrix gồm có 2 cột và rất nhiều dòng. Mỗi cột tương ứng cho 1 kênh âm thanh. Nếu bạn dùng phần mềm Audacity để xem file wav này thì thấy có 2 sóng như bên dưới:



<sup>21</sup> <https://www.coursera.org/learn/machine-learning>

## Lấy âm thanh từng kênh

```
channel1 = signal[:, 0]
channel2 = signal[:, 1]
print('channel1:', channel1)
print('channel2:', channel2)
```

```
channel1: [ 0  0  0 ... 57 56 58]
channel2: [ 0 -1  2 ... 56 54 61]
```

Quan sát kết quả thì tính hiệu âm thanh mỗi kênh (channel) là một mảng các số nguyên.

## Lưu một kênh âm thanh ra file

Để kiểm tra lại nội dung của kênh âm thanh số 1 thì dùng lệnh write như bên dưới. Sao đó dùng chương trình nghe nhạc để nghe lại cho chắc. Trong Windows thì chỉ cần double-click vào file để nghe.

```
wav.write('D:/Temp/c1.wav', samplerate, channel1)
```

## Vẽ sóng âm thanh

Để hiển thị được sóng âm thanh thì cần một phép biến đổi FFT (Fast Fourier Transform) đến chuyển mảng các số nguyên ở trên (dạng dữ liệu rời rạc, ý nói tín hiệu rời rạc) thành dạng tín hiệu theo kiểu tần số<sup>22</sup>.

Trong thư viện numpy có sẵn module hỗ trợ các thuật toán FFT.

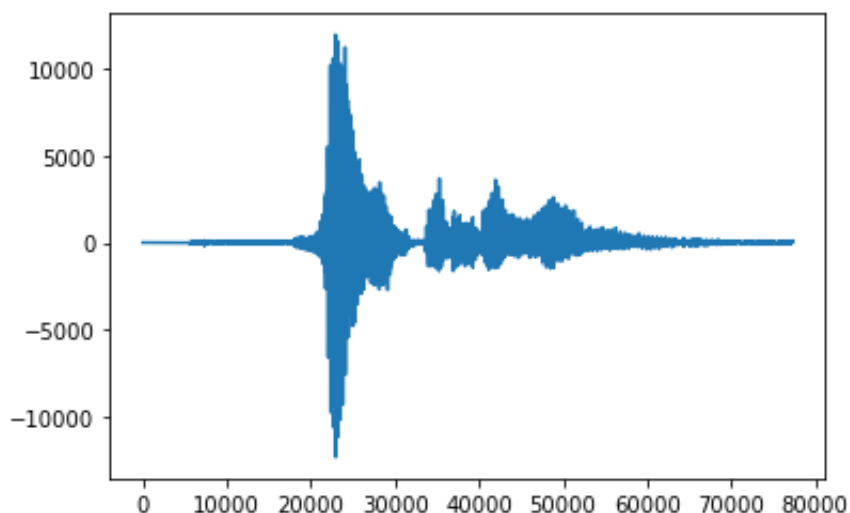
Kết hợp thư viện vẽ biểu đồ matplotlib và numpy như sau:

```
import matplotlib.pyplot as plt
from numpy.fft import fft

freq_channel1 = fft(channel1)
plt.plot(channel1)
plt.show()
```

---

<sup>22</sup> Xem thêm wiki: [https://vi.wikipedia.org/wiki/Biến\\_đổi\\_Fourier\\_nhanh](https://vi.wikipedia.org/wiki/Biến_đổi_Fourier_nhanh)



### Nghe âm thanh từ mảng numpy

Bạn cần cài đặt thư viện `sounddevice` bằng lệnh `pip` (nhớ lạy chạy trong dấu nhắc lệnh của môi trường có lệnh `pip`, khuyến nghị nên chạy trong Anaconda Prompt)

```
pip install sounddevice
```

Code Python để nghe kênh 1 đã được trích xuất như sau:

```
import sounddevice
sounddevice.play(channel1, samplerate)
```

## Phân tích âm thanh với thư viện mutagen

### Cài đặt thư viện

```
pip install mutagen
```

### Tính thời gian của file mp3 với thư viện mutagen

```
from mutagen.mp3 import MP3

audio = MP3('audio.mp3')

audio_info = audio.info
length_in_secs = int(audio_info.length)

print('Duration of audio', length_in_secs)
```

### Tính thời gian của file mp3 với thư viện pydub

Hàm len trong đoạn code sau sẽ trả lại thời lượng của file .mp3 với đơn vị là mili giây (1 giây = 1000 mili giây):

```
from pydub import AudioSegment

audio = AudioSegment.from_mp3('audio.mp3')
print('Duration of audio (milliseconds):', len(audio))
```

### Duyệt thư mục

```
PATH = r'D:\Projects\ -2019-01'
for dirpath, dirnames, filenames in os.walk(PATH):
    for filename in [f for f in filenames if
f.endswith('.mp3')]:
        print(os.path.join(dirpath, filename))
```

## Khám phá Python trong WSL2

### Sử dụng thư viện ocrmypdf

Nguồn tài liệu:

<https://pypi.org/project/ocrmypdf/>

#### *Sử dụng trên Ubuntu (bao gồm Ubuntu trong Windows)*

```
sudo apt-get install ocrmypdf
```

Xem các gói ngôn ngữ

```
apt-cache search tesseract-ocr
```

```
gimagereader - Graphical GTK+ front-end to tesseract-ocr
python3-pyocr - Python wrapper for OCR engines (Python 3)
python3-tesseract - Python wrapper for the tesseract-ocr API (Python3 version)
tesseract-ocr - Tesseract command line OCR tool
...
tesseract-ocr-eng - tesseract-ocr language files for English
...
tesseract-ocr-vie - tesseract-ocr language files for Vietnamese
```

#### *Chuyển file pdf được scan dưới dạng hình ảnh thành dạng text có thể copy được*

```
ocrmypdf input.pdf output.pdf
```

Sau lệnh trên, file output.pdf có thể bôi văn bản và copy rồi dán vào tài liệu được.

Lệnh sau đây sẽ trích xuất text trong file input.pdf ra file output.txt.

```
ocrmypdf --sidecar output.txt ./input.pdf output.pdf
```

#### *Sử dụng trong Windows*

Cài đặt thư viện bằng cách chạy lệnh sau trong môi trường Python:

```
pip install ocrmypdf
```

Tải và cài đặt phần mềm Ghostscript tại:

<https://www.ghostscript.com/download/gsdnld.html>

Tham khảo:

<https://ocrmypdf.readthedocs.io/en/latest/cookbook.html#basic-examples>

Trích xuất hình trong file pdf

```
pip install PyMuPDF
```

```
import fitz
```

```
doc = fitz.open('D:/Temp/ETS_2020_LC_o.pdf')
n_pages = len(doc)
for i in range(n_pages):
    for img in doc.getPageImageList(i):
        xref = img[0]
        pix = fitz.Pixmap(doc, xref)
        if pix.n < 5:
            pix.writePNG('p%s-%s.png' % (i, xref))
        else:
            pix1 = fitz.Pixmap(fitz.csRGB, pix)
            pix1.writePNG('p%s-%s.png' % (i, xref))
            pix1 = None
        pix = None
```

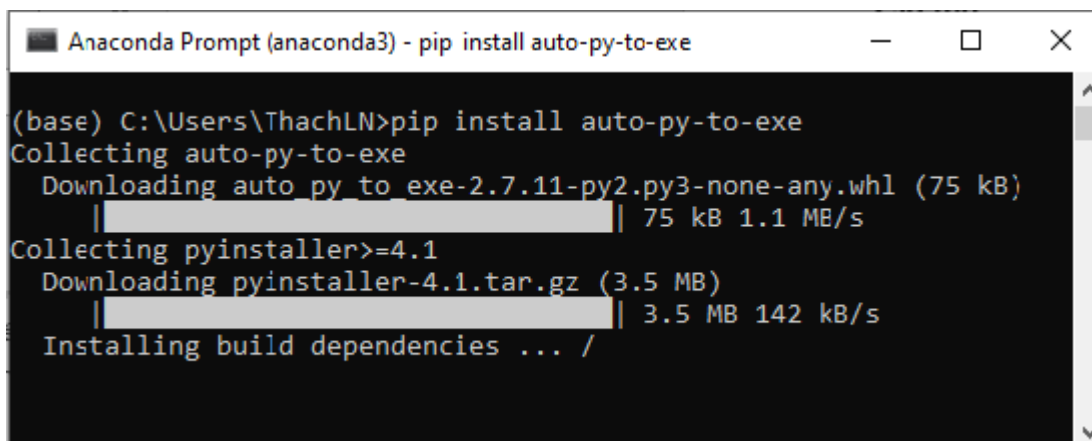
Kết quả: ảnh không được trích xuất đầy đủ.

## Đóng gói chương trình Python

Khi bạn có nhu cầu đóng gói chương trình bạn viết bằng Python thành file thực thi (Ví dụ như file .exe trên Windows) thì có thể sử dụng gói **auto-py-to-exe** như tôi hướng dẫn dưới đây. Bài viết áp dụng cho chương trình đơn giản chỉ có 1 file mã nguồn Python.

### Cài đặt

```
pip install auto-py-to-exe
```



```
Anaconda Prompt (anaconda3) - pip install auto-py-to-exe
(base) C:\Users\ThachLN>pip install auto-py-to-exe
Collecting auto-py-to-exe
  Downloading auto_py_to_exe-2.7.11-py2.py3-none-any.whl (75 kB)
    |-----| 75 kB 1.1 MB/s
Collecting pyinstaller>=4.1
  Downloading pyinstaller-4.1.tar.gz (3.5 MB)
    |-----| 3.5 MB 142 kB/s
Installing build dependencies ... /
```

Tham khảo trang chủ: <https://pypi.org/project/auto-py-to-exe>

### Đóng gói một chương trình đơn giản

Trong cửa sổ lệnh của Anaconda thực hiện lệnh sau:

```
auto-py-to-exe
```

Trình duyệt sẽ mở ứng dụng lên. Bạn bấm nút Browse để chọn file mã nguồn Python của bạn.

Trong ví dụ dưới đây thì tôi chỉnh lại mã nguồn Python trong Ngày 5, Bài 21, Code Python minh họa Hồi qui tuyến tính. Bạn lưu vào file D:\MyPython\regression.py để thực hành.

```
import pandas as pd
from sklearn.linear_model import LinearRegression

file url = 'https://thachln.github.io/datasets/TuyenVN 2019.csv'
print('Download sample data from {}'.format(file url))

df = pd.read_csv(file url)
print(df.columns)
```

```
model = LinearRegression()
model.fit(df[['Weight']], df[['Height']])

LinearRegression(copy_X = True, fit_intercept = True, n_jobs = None,
normalize = False)

intercept = model.intercept_
print('intercept = {}'.format(intercept))

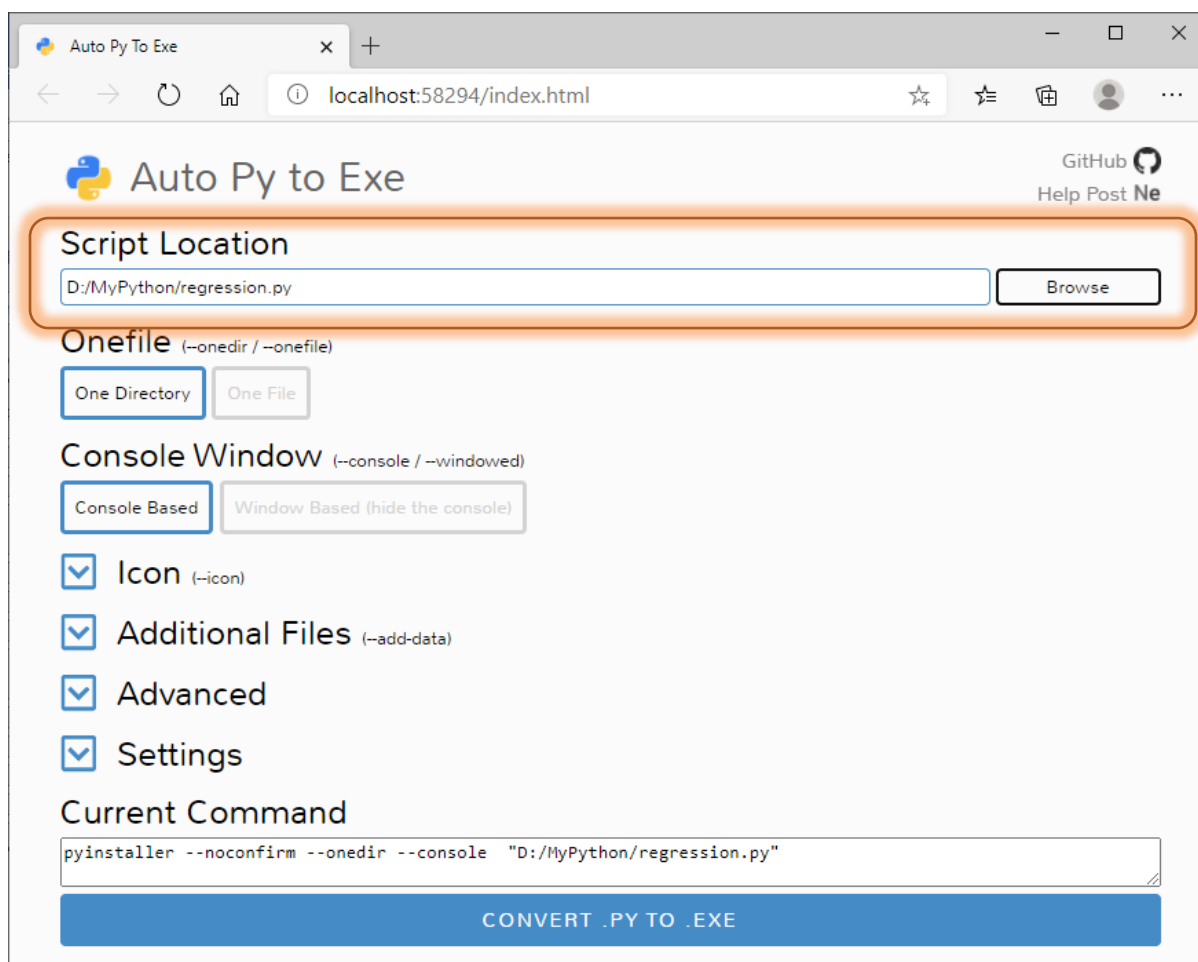
coefficent = model.coef
print('coefficient = {}'.format(coefficent))

input('Press Enter to finish...')
```

Tôi bổ sung thêm dòng `input('...')` để chương trình (sau khi đóng gói) chạy xong thì dừng lại đợi bạn nhấn một phím gì đó trước khi nó thoát (cửa sổ biến mất).

Sau khi có file `D:\MyPython\regression.py` thì hãy nên chạy thử đảm bảo không có lỗi rồi bấm nút Browse trong cửa sổ của trang web “Auto Py to Exe” để chọn file như bên dưới.



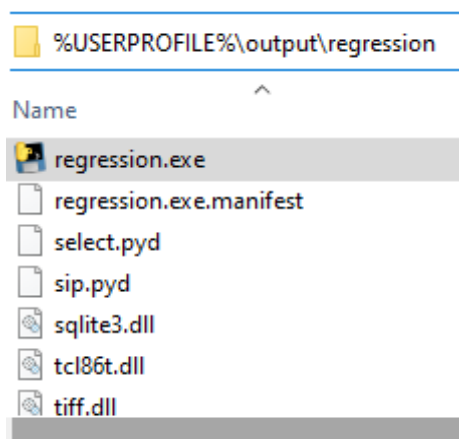


Sau đó bấm nút “CONVERT .PY TO EXE” ở dưới đáy màn hình. Có thể bạn phải ngồi uống hết ly café tùy theo cấu hình máy của bạn.

Kết quả ra màn hình bên dưới.



Bấm vào nút “OPEN OUTPUT FOLDER” để xem thành quả. Module “Auto Py to EXE” sẽ đóng gói các thư viện cần thiết vào thư mục. Trong đó có file .exe tương ứng với tên file .py của bạn. Trong ví dụ của tôi là file regression.exe.



Bạn có thể nén thư mục regression này lại và gửi tặng đồng nghiệp.