



macOS



Linux

**If windows**

## Anaconda 2019.07 for Windows Installer

### Python 3.7 version

**click**

Download

64-Bit Graphical Installer (486 MB)

32-Bit Graphical Installer (418 MB)

### Python 2.7 version

Download

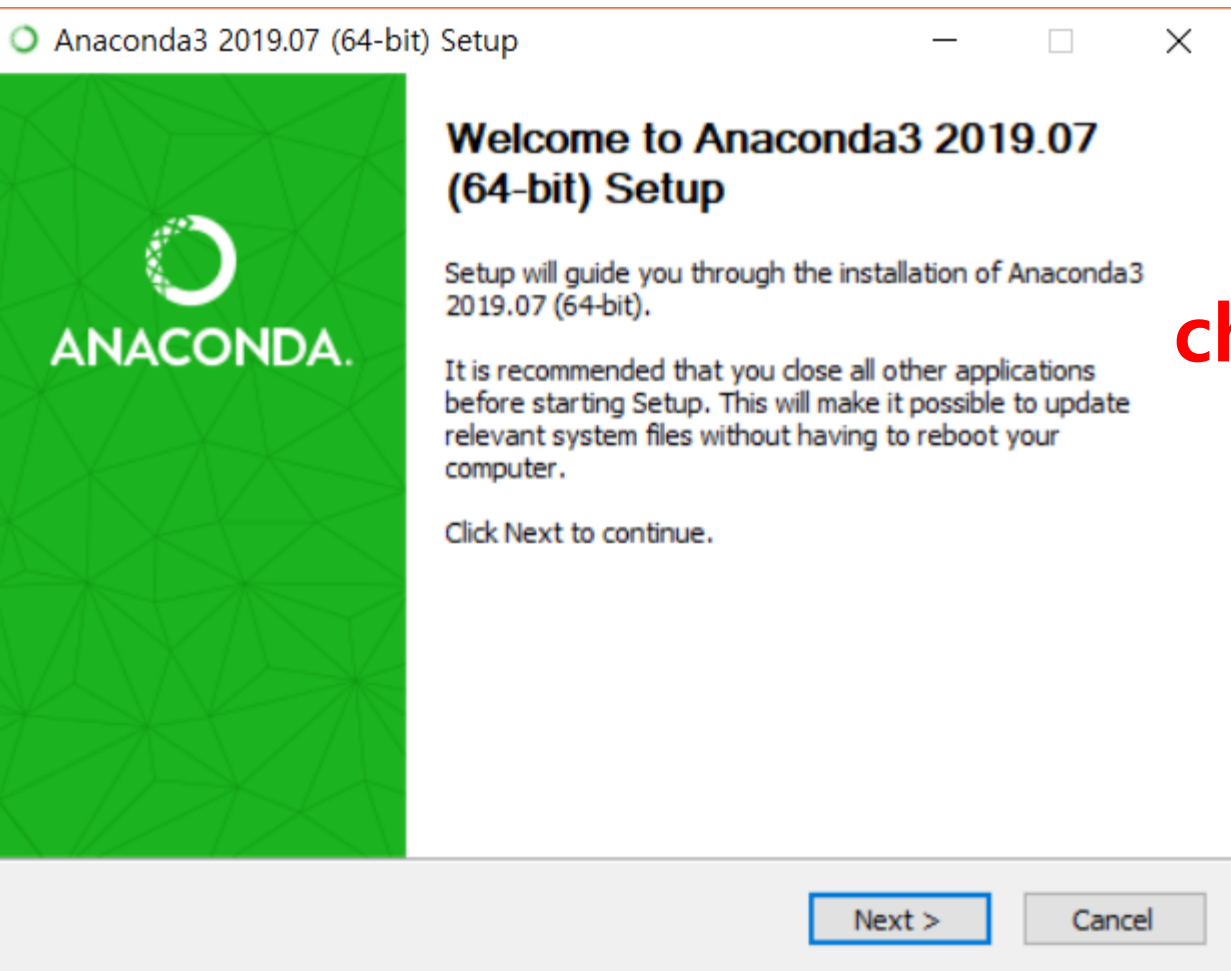
64-Bit Graphical Installer (427 MB)

32-Bit Graphical Installer (361 MB)

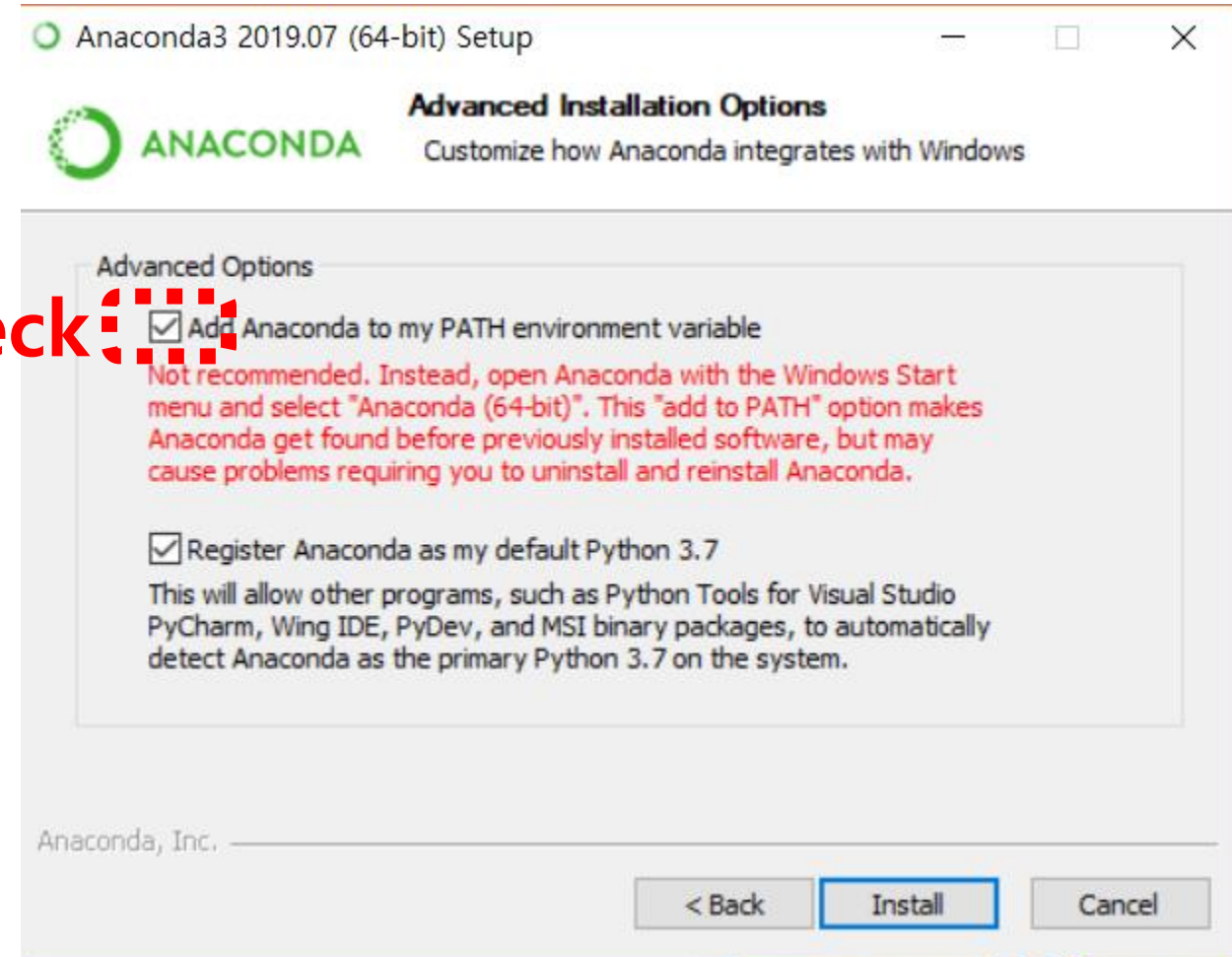
**If linux, use linux download  
64bit or 32bit depends on computer**

# Installing Anaconda

## Check the boxes!



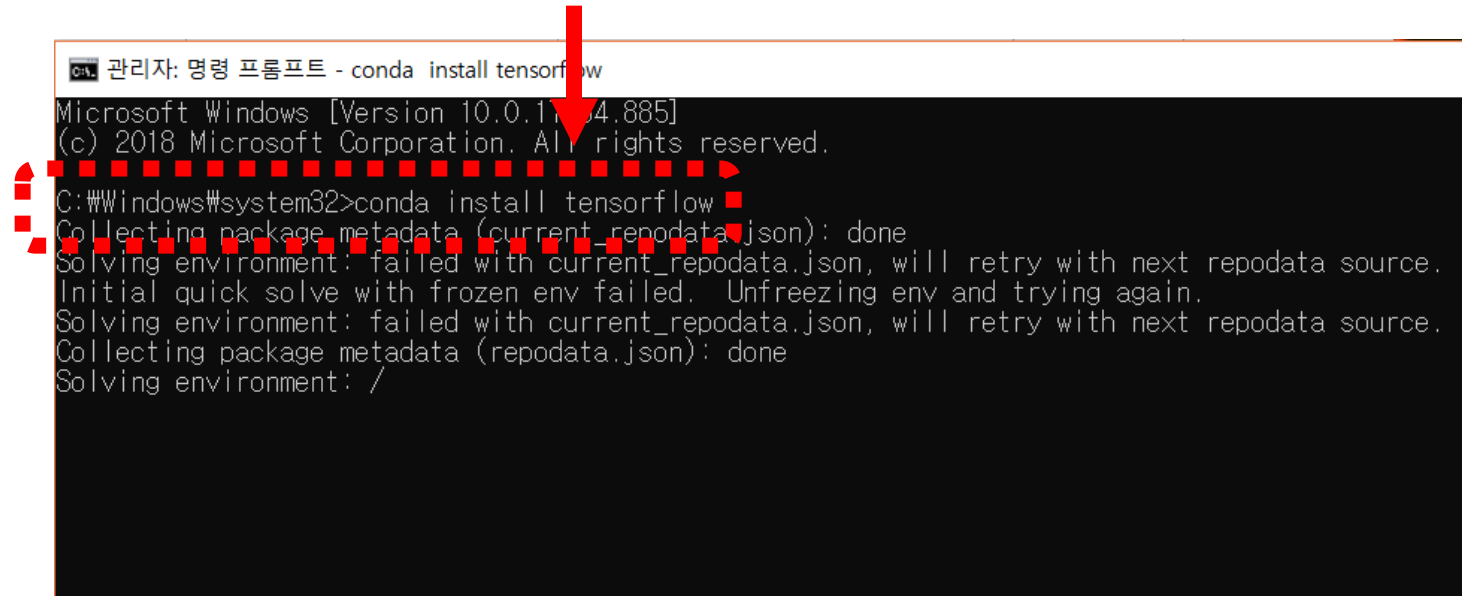
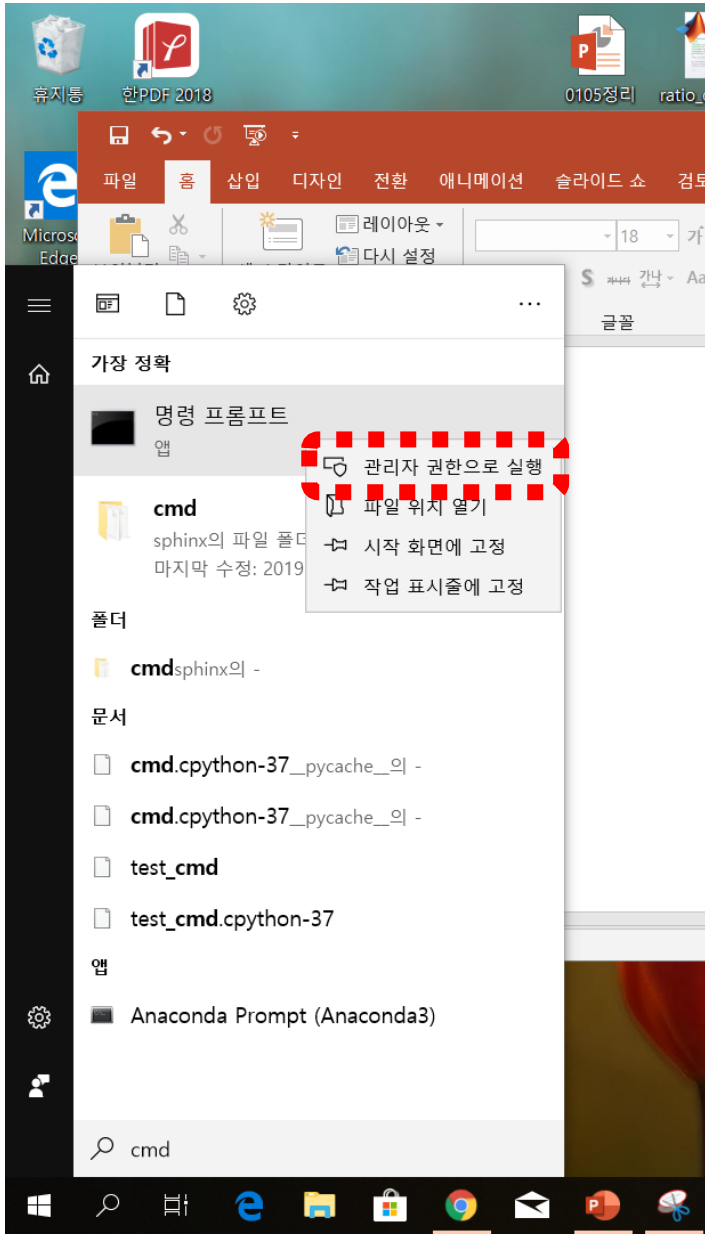
Click next until..... ->



Click next until installation completes

# Installing Tensorflow (Windows)

1. Windows 검색
  2. "cmd" 검색
  3. 명령 프롬프트 오른쪽 클릭
  4. 관리자 권한으로 실행 클릭 -> 예 클릭
  5. "conda install tensorflow" -> 엔터
- (GPU 버전: "conda install tensorflow-gpu")



# Installing Tensorflow (Windows)

C:\ 관리자: 명령 프롬프트 - conda install tensorflow

protobuf-3.8.0	py37h33f27b4_0	527 KB
tensorboard-1.13.1	py37h33f27b4_0	3.1 MB
tensorflow-1.13.1	mkl_py37h9463c59_0	4 KB
tensorflow-base-1.13.1	mkl_py37hcaf7020_0	32.4 MB
tensorflow-estimator-1.13.0	py_0	205 KB
termcolor-1.1.0	py37_1	8 KB
-----		
Total:		60.9 MB

The following NEW packages will be INSTALLED:

_tfflow_select	pkgs/main/win-64::_tfflow_select-2.3.0-mkl
absl-py	pkgs/main/win-64::absl-py-0.7.1-py37_0
astor	pkgs/main/win-64::astor-0.8.0-py37_0
gast	pkgs/main/win-64::gast-0.2.2-py37_0
grpcio	pkgs/main/win-64::grpcio-1.16.1-py37h351948d_1
keras-applications	pkgs/main/noarch::keras-applications-1.0.8-py_0
keras-preprocessing	pkgs/main/noarch::keras-preprocessing-1.1.0-py_1
libmklml	pkgs/main/win-64::libmklml-2019.0.5-0
libprotobuf	pkgs/main/win-64::libprotobuf-3.8.0-h7bd577a_0
markdown	pkgs/main/win-64::markdown-3.1.1-py37_0
protobuf	pkgs/main/win-64::protobuf-3.8.0-py37h33f27b4_0
tensorboard	pkgs/main/win-64::tensorboard-1.13.1-py37h33f27b4_0
tensorflow	pkgs/main/win-64::tensorflow-1.13.1-mkl_py37h9463c59_0
tensorflow-base	pkgs/main/win-64::tensorflow-base-1.13.1-mkl_py37hcaf7020_0
tensorflow-estimator	pkgs/main/noarch::tensorflow-estimator-1.13.0-py_0
termcolor	pkgs/main/win-64::termcolor-1.1.0-py37_1

Proceed ([y]/n)?

6. "y" -> 엔터

7. 다 끝나고 "conda list" -> 엔터

8. 리스트에  
tensorflow,  
tensorflow-base,  
tensorflow-  
estimator 확인하  
면 정상 설치



Version: 2019.2

Build: 192.5728.105

Released: July 23, 2019 GMT

[System requirements](#)

[Installation Instructions](#)

[Other versions](#)

# Download PyCharm

Windows

macOS

Linux

## Professional

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

DOWNLOAD

Free trial

## Community

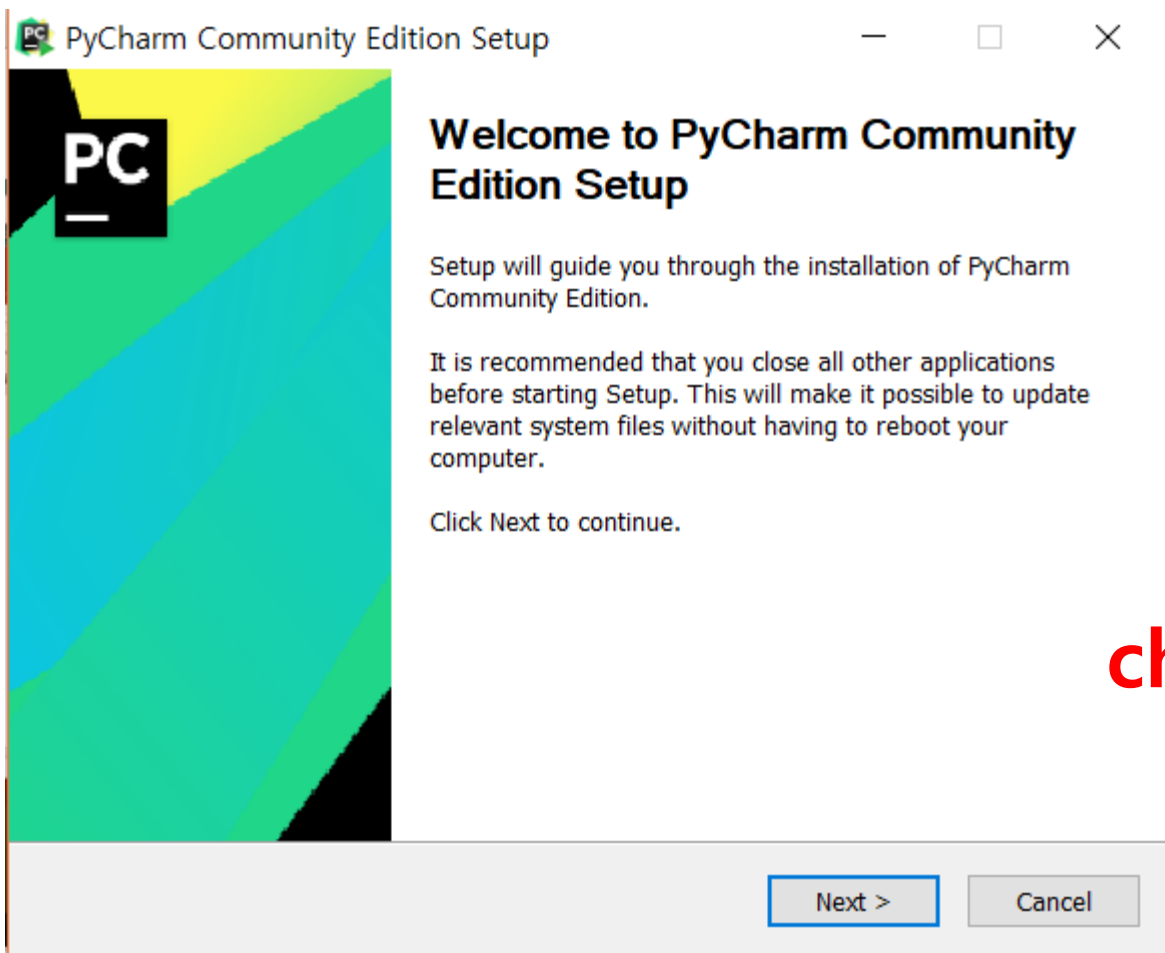
For pure Python development

DOWNLOAD

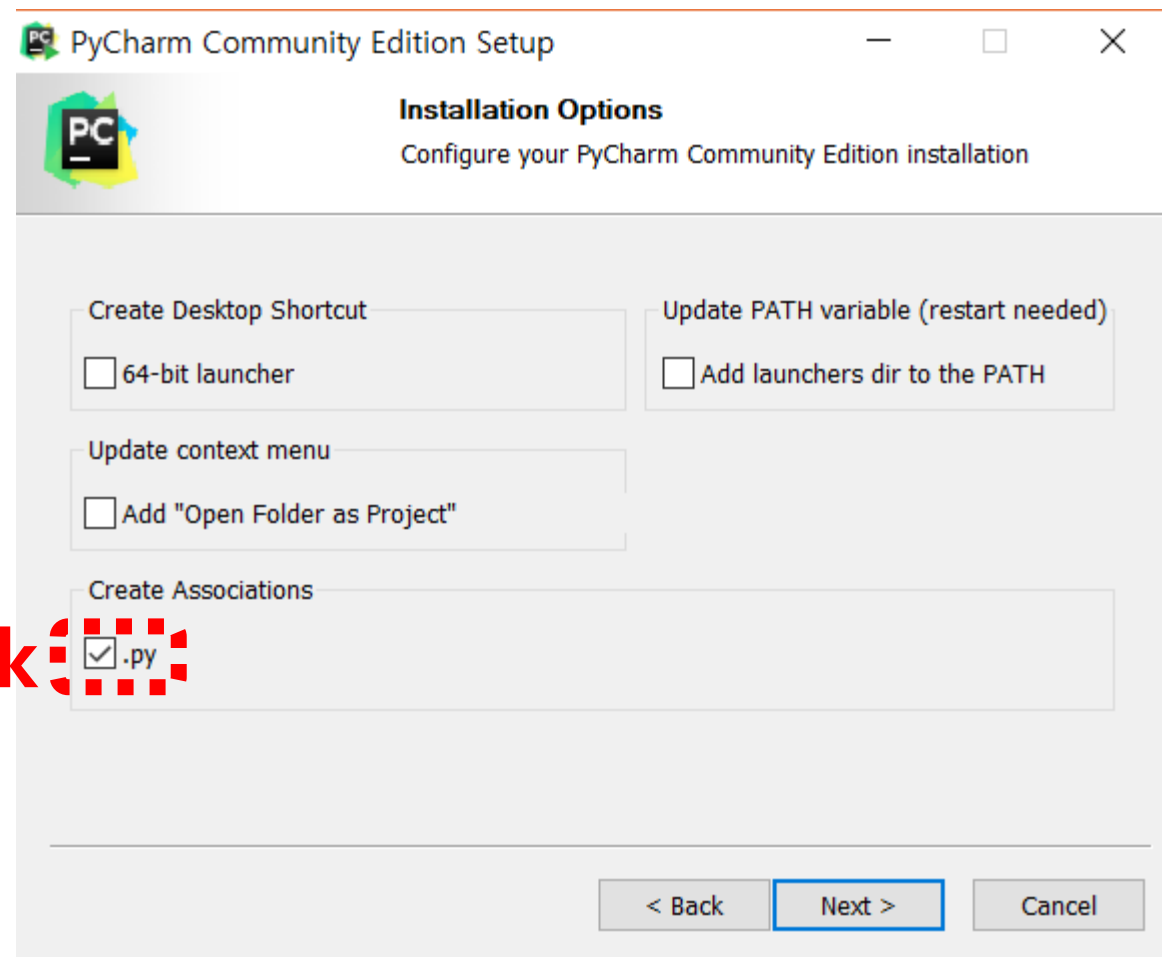
Free, open-source

# Installing Python IDE – pycharm

Check the boxes!



Click next until..... ->

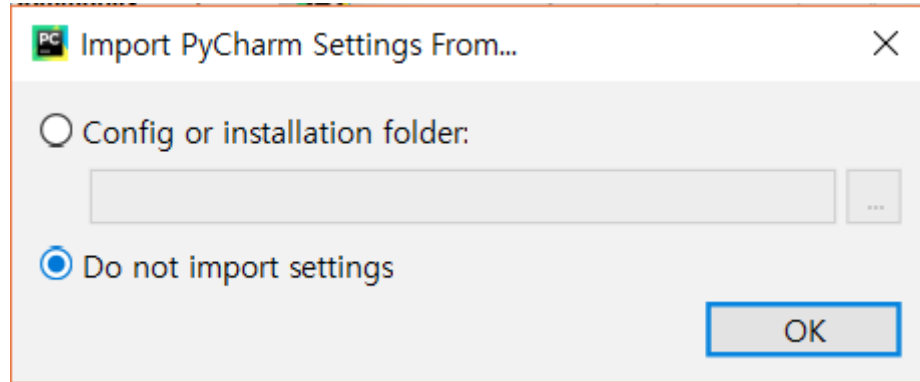


Click next until installation completes

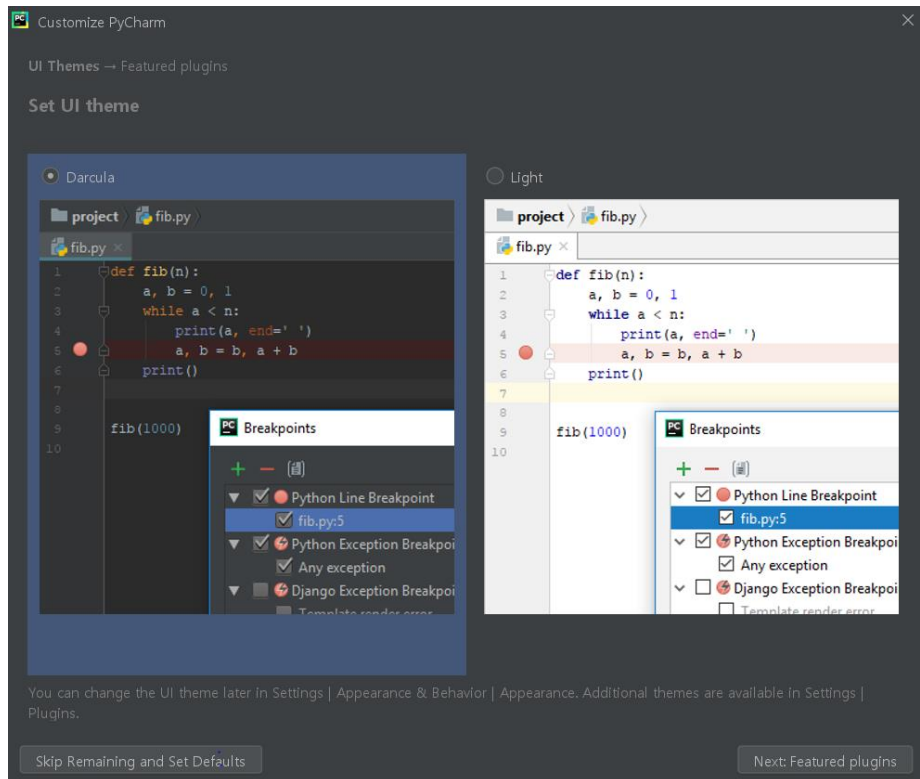
다 끝난 후 컴퓨터 다시시작, 그 후 pycharm 실행

# Setting up pycharm

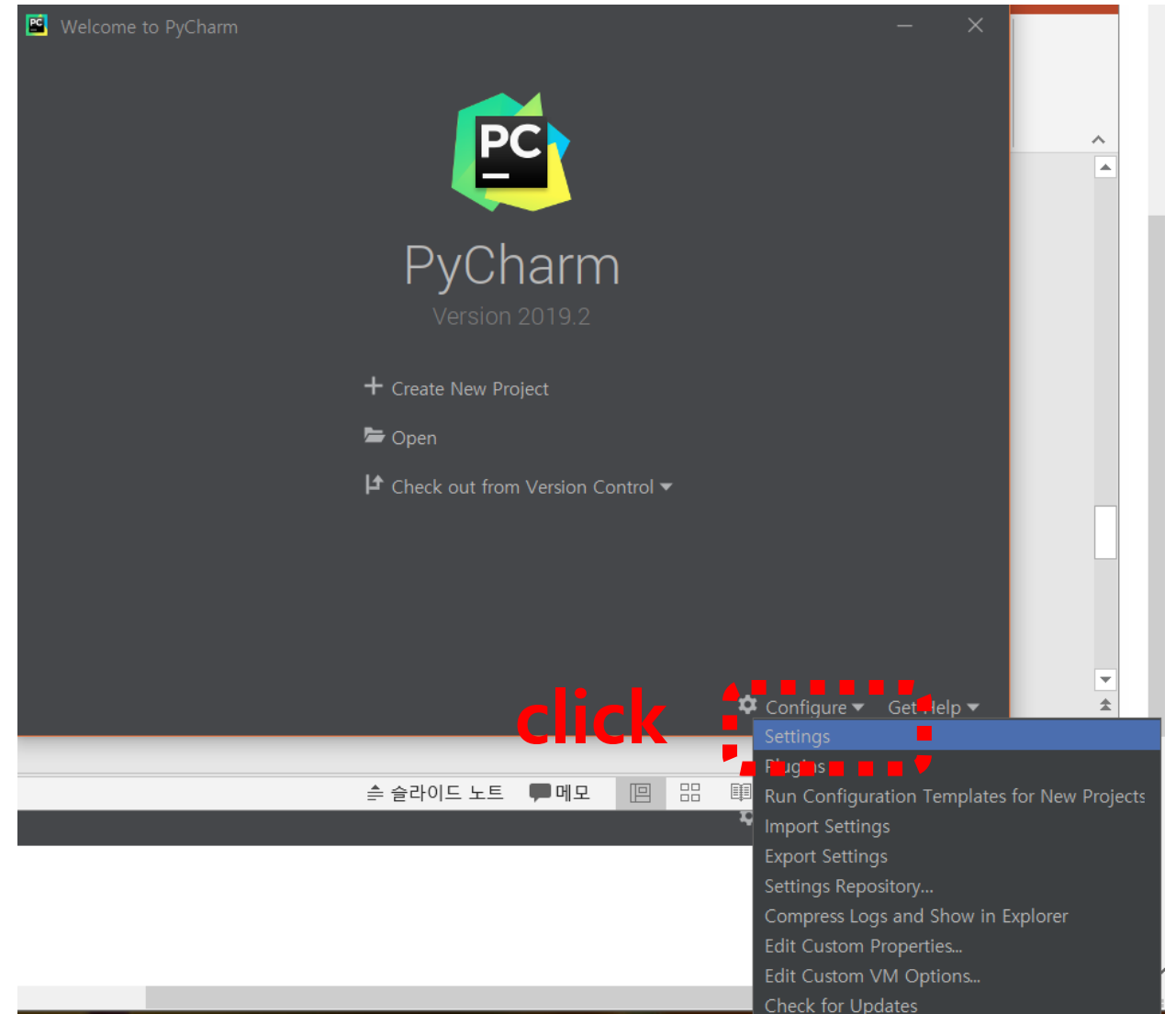
Click next until.....



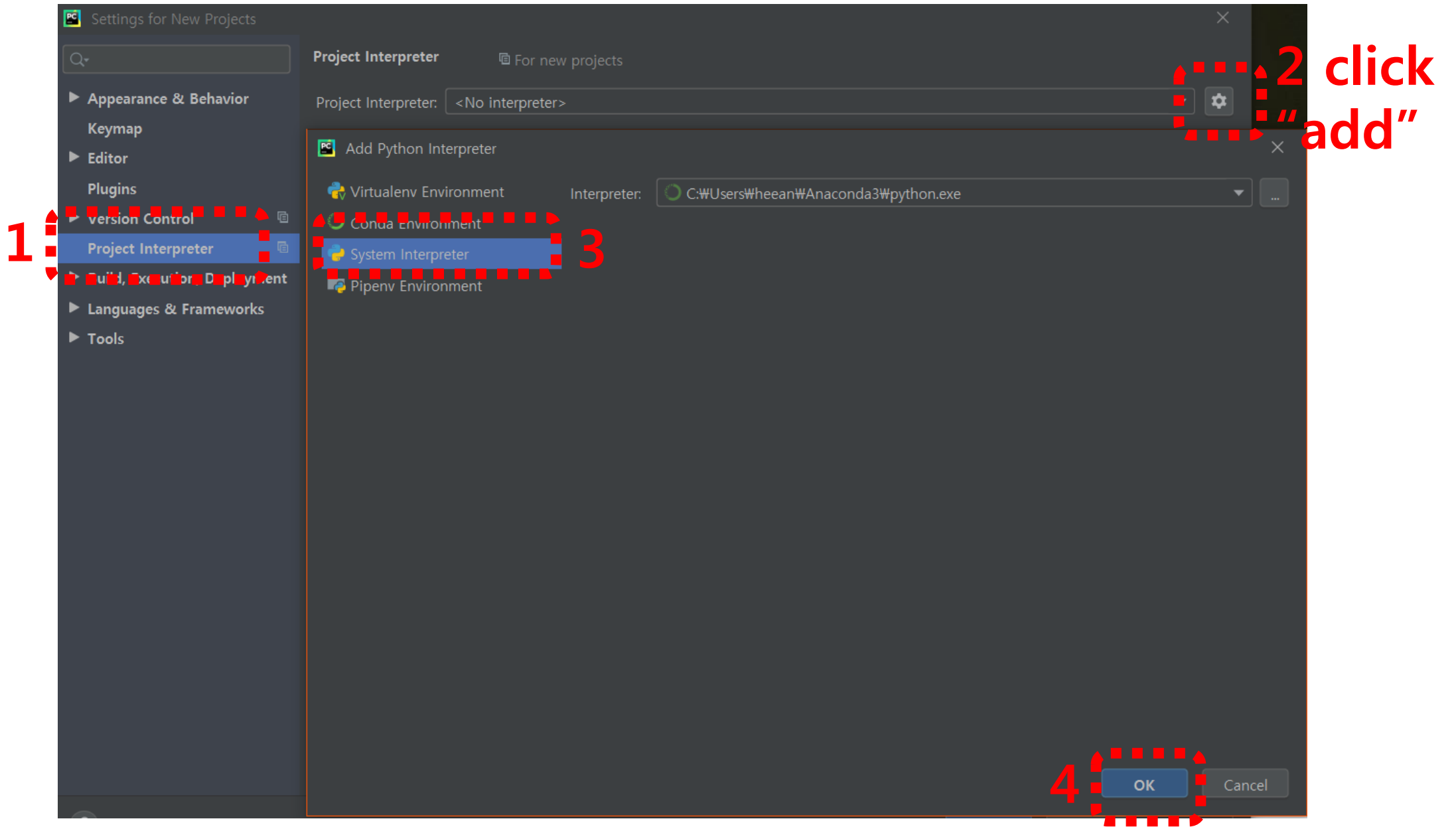
Press skip



Click 'configure' -> 'settings'



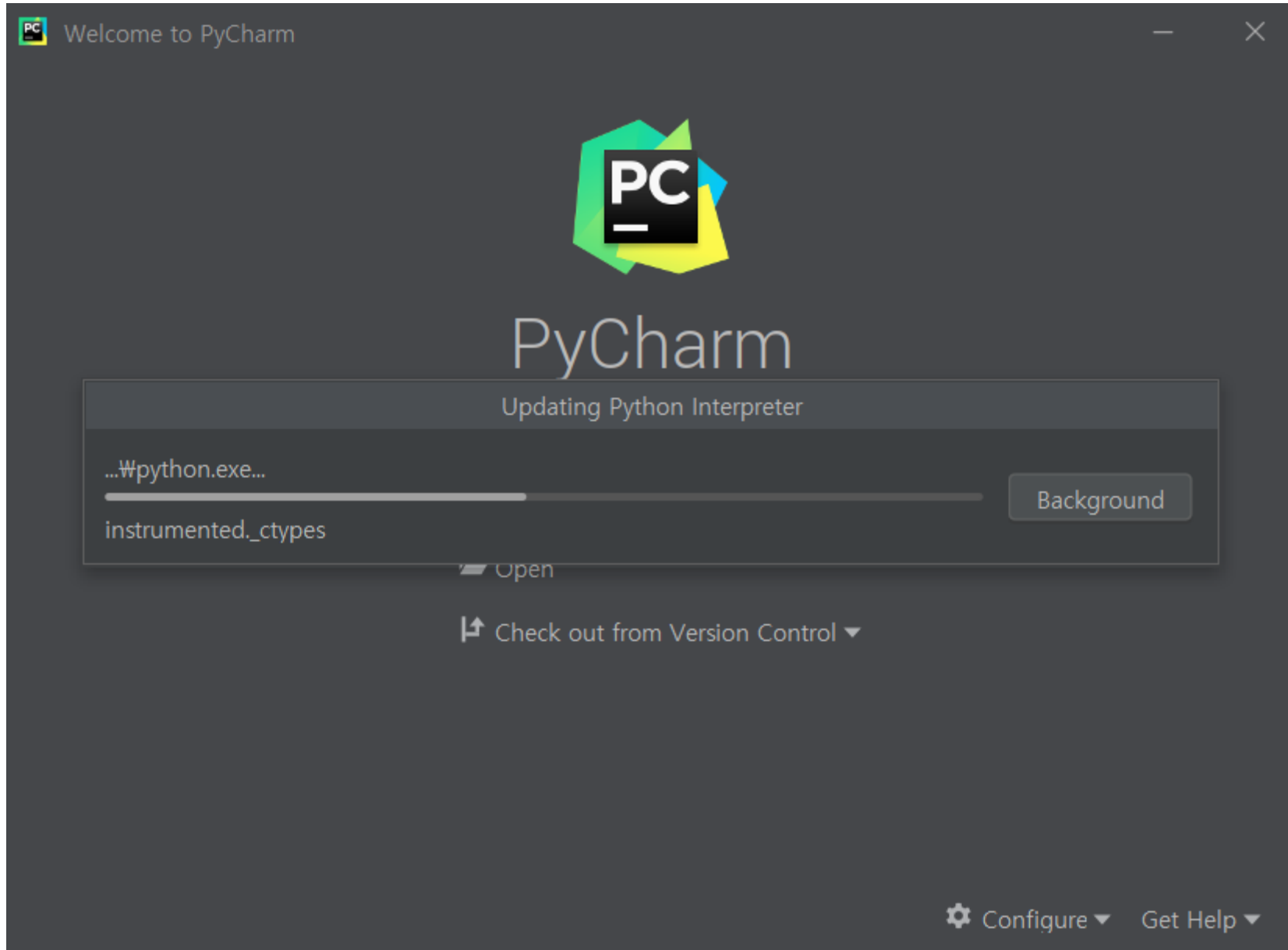
# Setting up pycharm



Ok 누르면 원본 윈도우 ("settings for new projects")에서 "loading" 뜸 -> click "apply" -> click "ok"

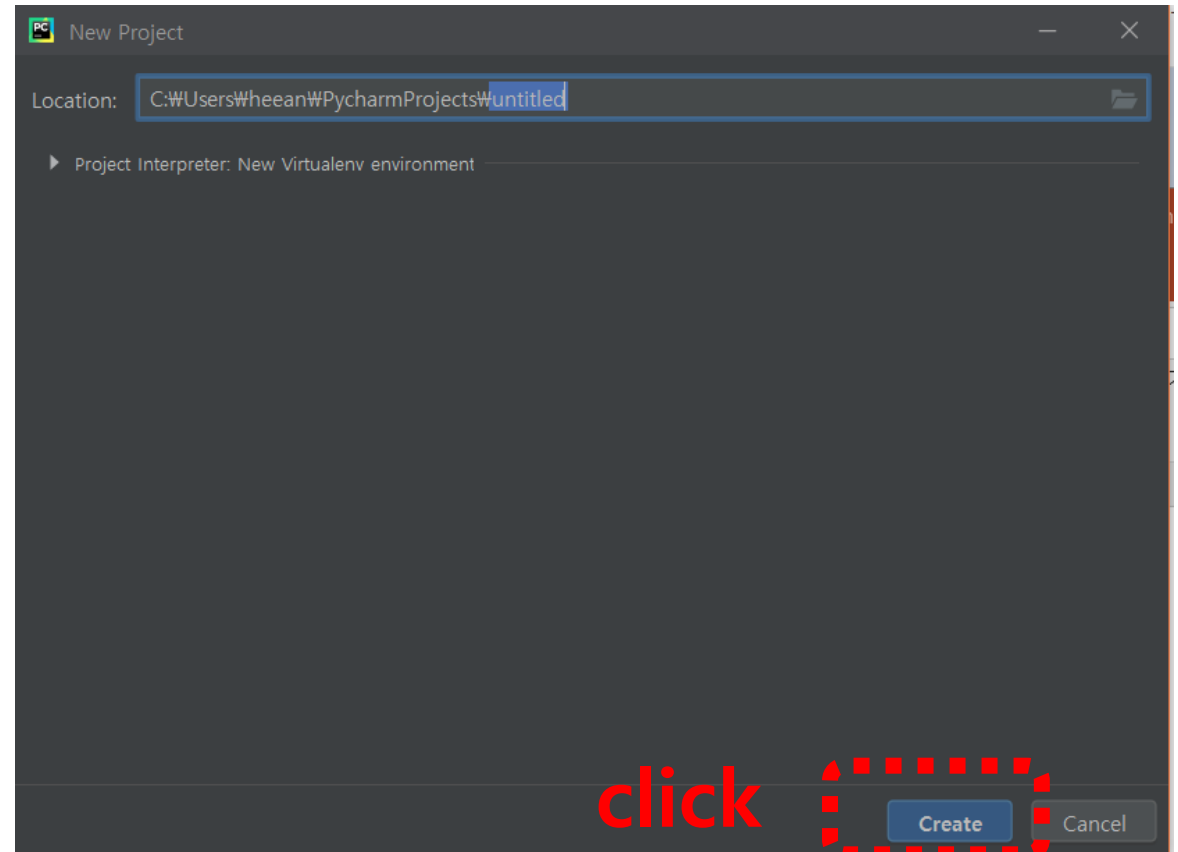
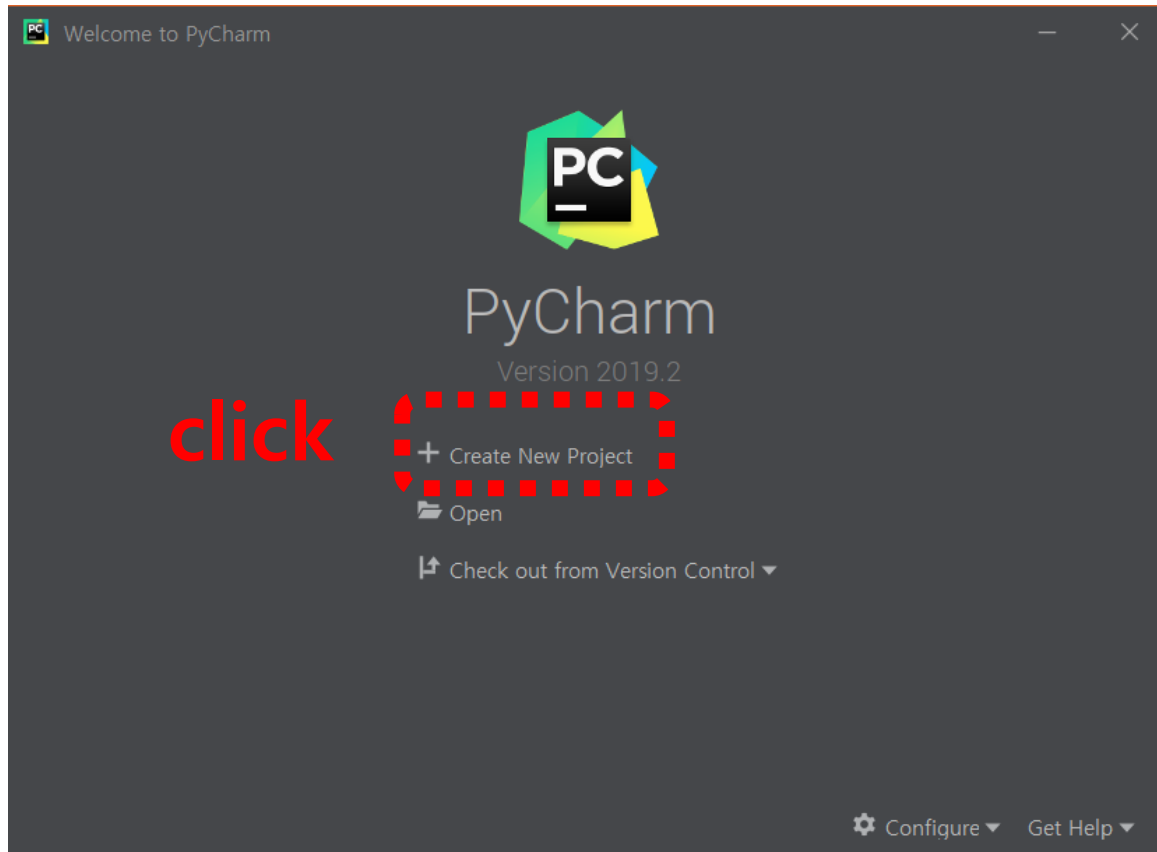


## Setting up pycharm

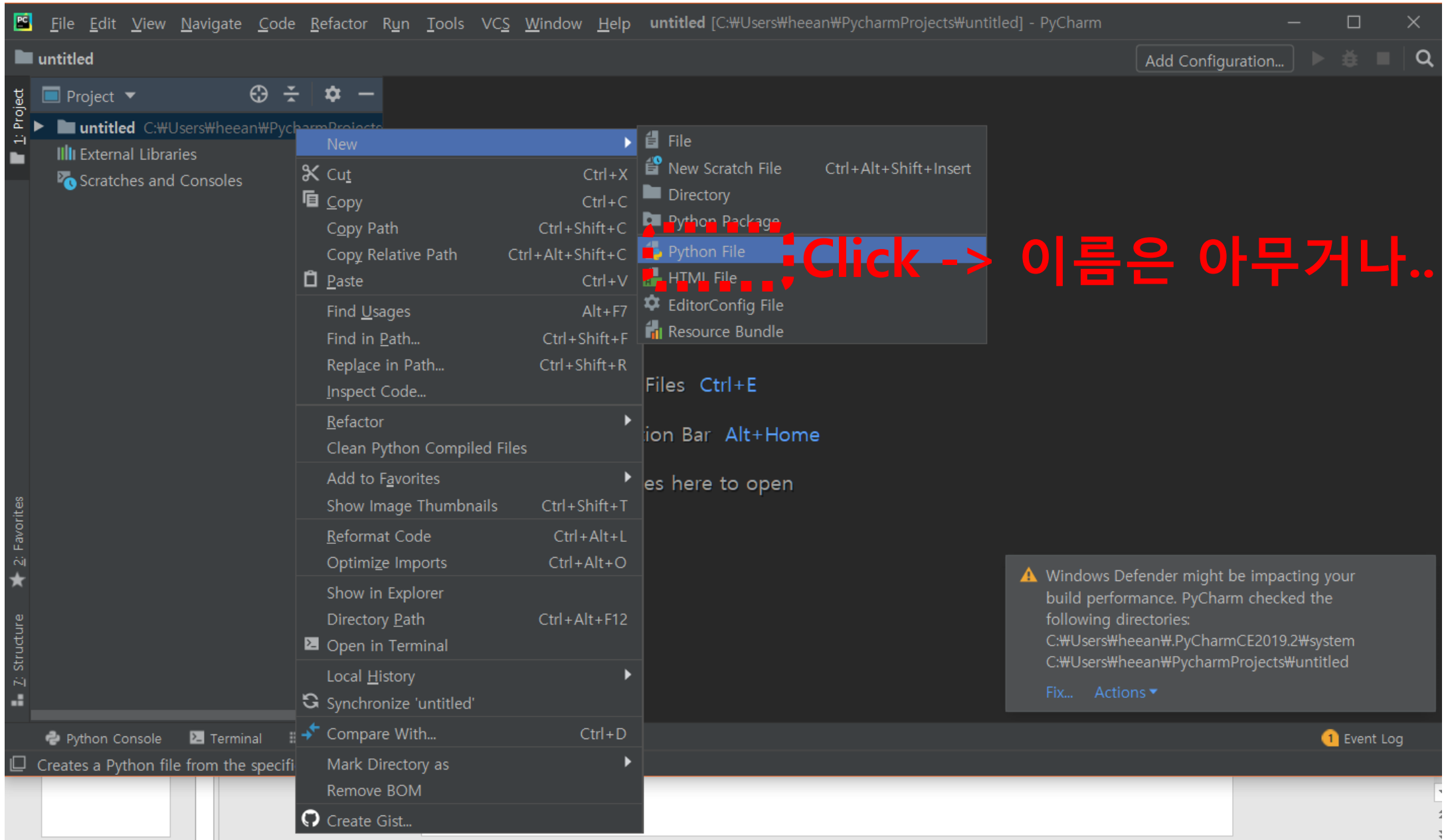


다음 화면에서 위 같은 작업이 진행되는 것을 확인할 수 있음 (컴퓨터 사양에 따라 소요시간 다름)

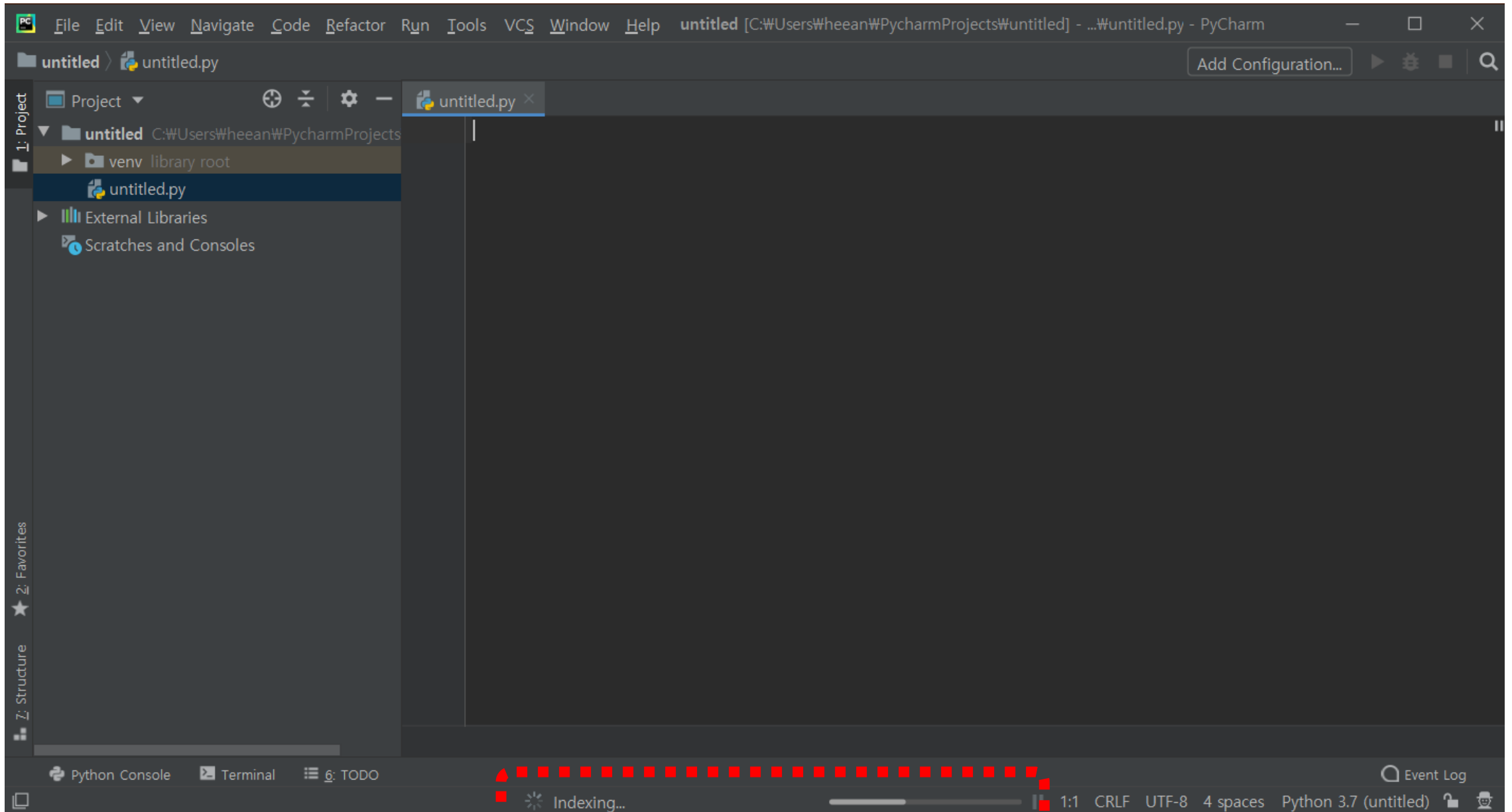
## 설치 확인



# Python script 생성

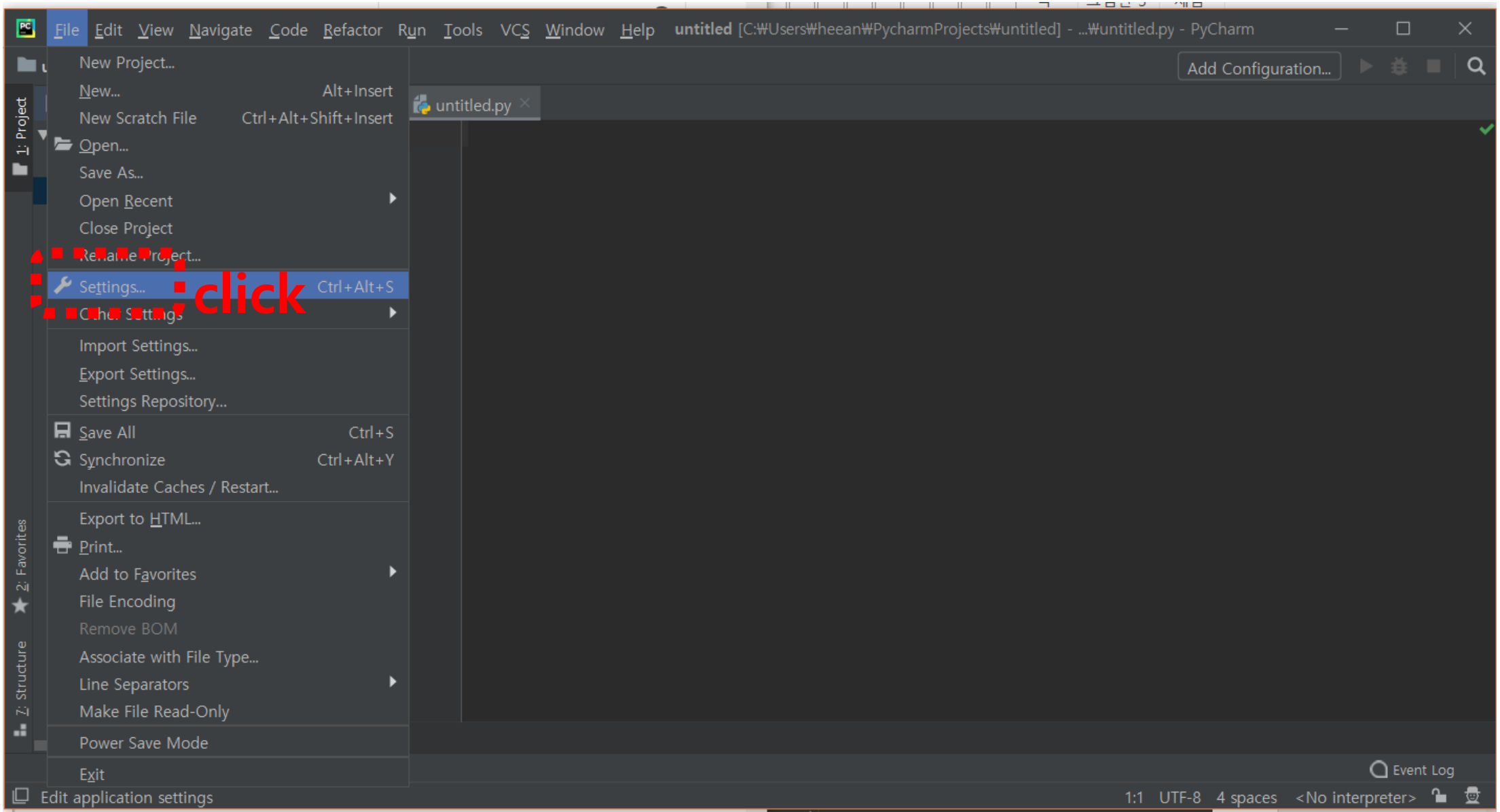


# Python script 생성



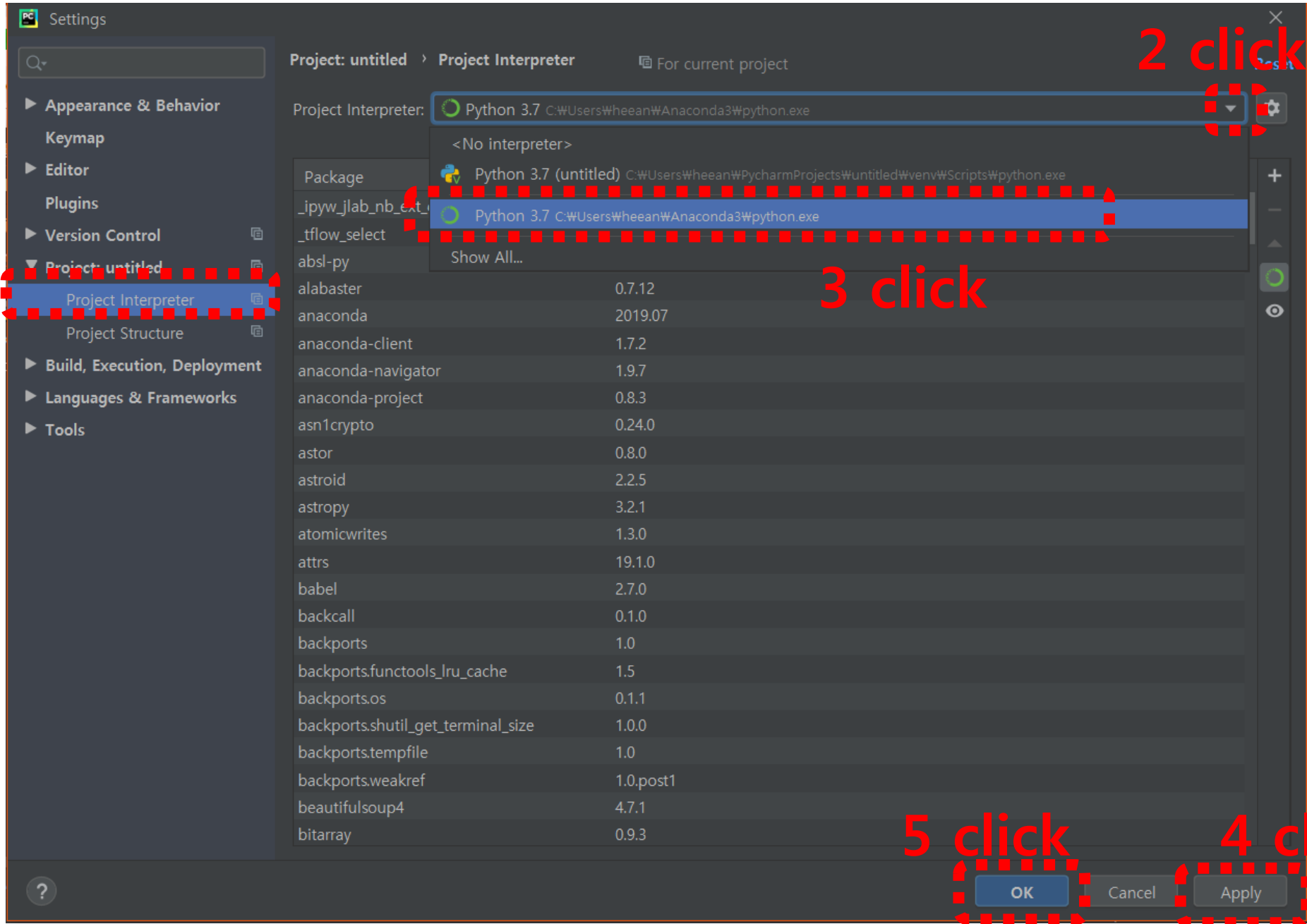
열심히 세팅중...

# Project interpreter 세팅 (매 프로젝트당 해야함)



## Project interpreter 세팅 (매 프로젝트당 해야함)

1 click



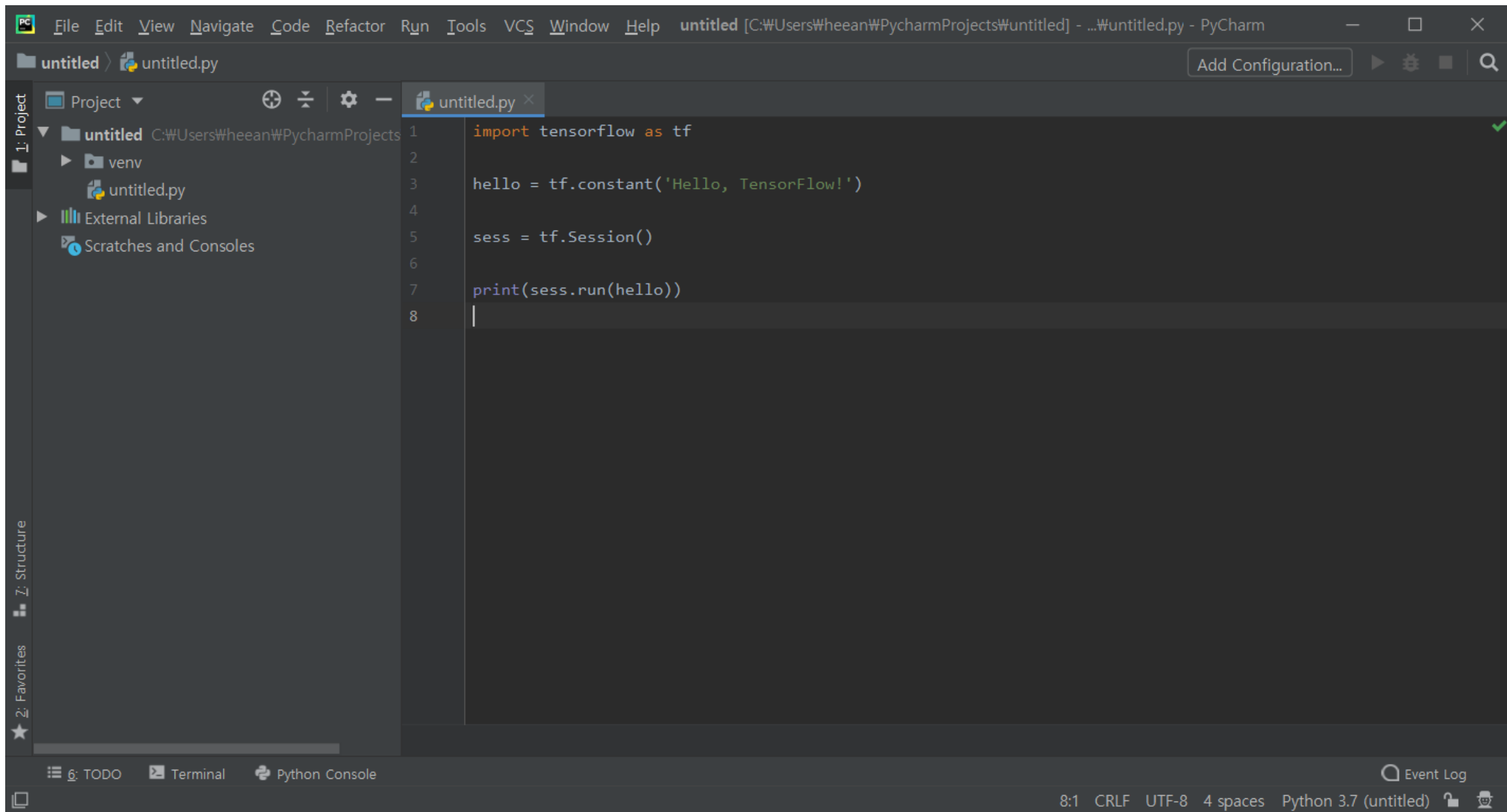
2 click

3 click

5 click

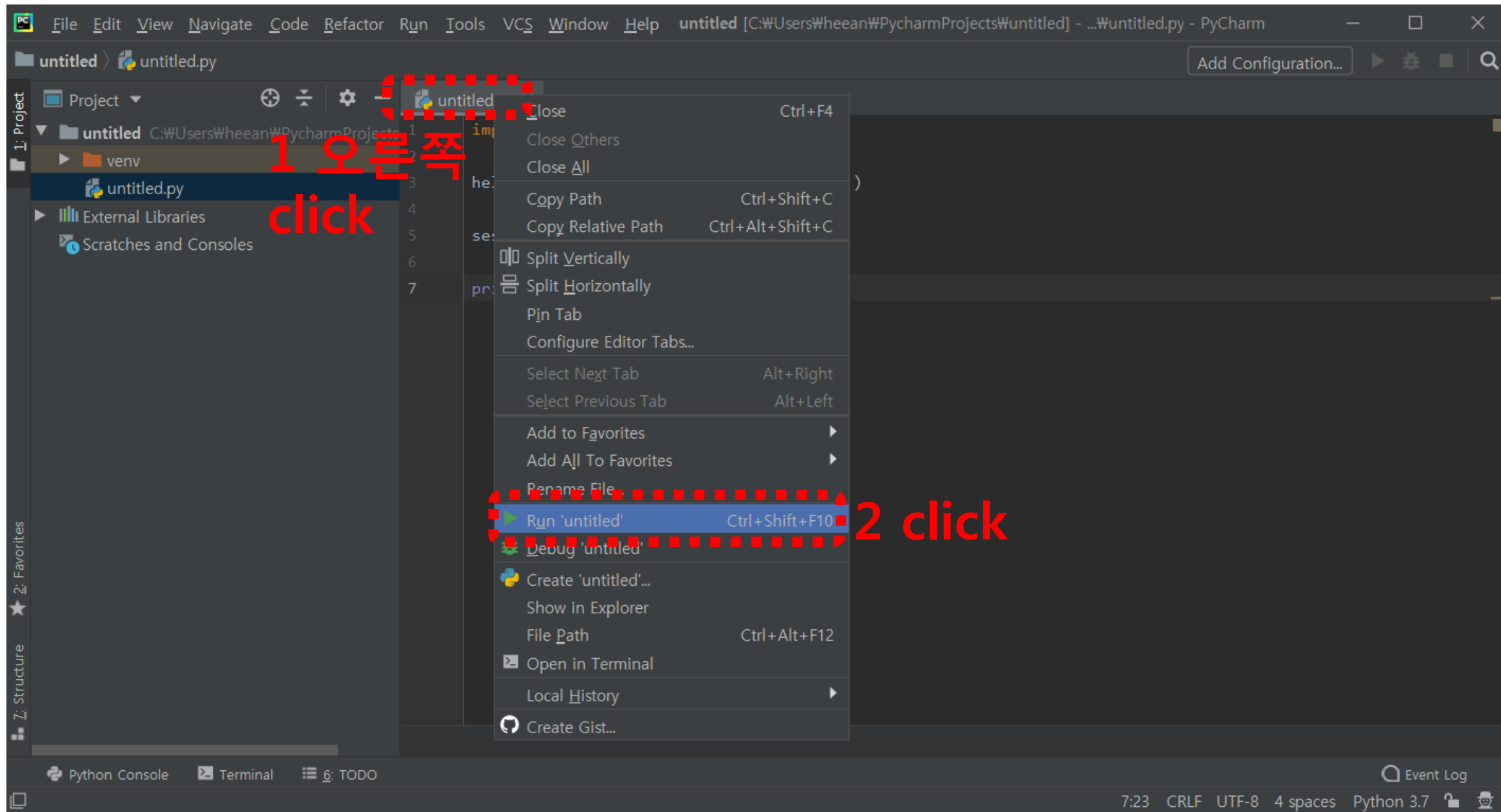
4 click

## 설치 확인 (“hello world” 실행)



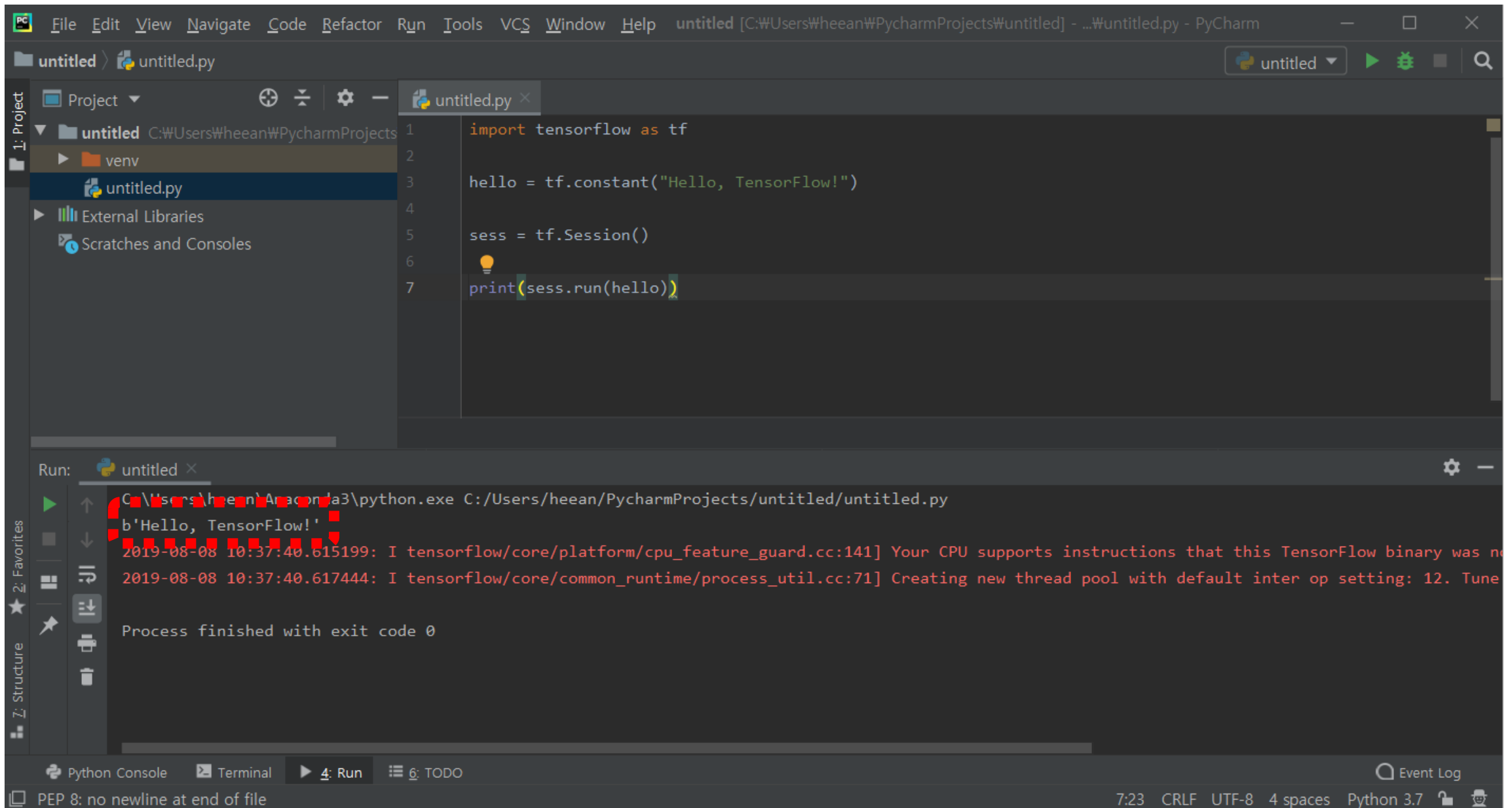
위 코드를 똑같이 타입

## 설치 확인 ("hello world" 실행)





## 설치 확인 ("hello world" 실행)



**결과가 비슷하게 나오면 성공!!**

## 그 후... 실습전 조금 더 배우고 싶은분들

- Python syntax 수업:  
<https://www.codecademy.com/learn/learn-python>
  - 사실 필요한 부분은 일부이지만 좋은 introduction to python
- Tensorflow 기본 실습:  
<https://github.com/aymericdamien/TensorFlow-Examples>
  - Linear regression:  
[https://github.com/aymericdamien/TensorFlow-Examples/blob/master/examples/2\\_BasicModels/linear\\_regression.py](https://github.com/aymericdamien/TensorFlow-Examples/blob/master/examples/2_BasicModels/linear_regression.py)

# Tensorflow linear regression 예제

```
1  '''
2  A linear regression learning algorithm example using TensorFlow library.
3
4  Author: Aymeric Damien
5  Project: https://github.com/aymericdamien/TensorFlow-Examples/
6  '''
7
8  from __future__ import print_function
9
10 import tensorflow as tf
11 import numpy
12 import matplotlib.pyplot as plt
13 rng = numpy.random
14
15 # Parameters
16 learning_rate = 0.01
17 training_epochs = 1000
18 display_step = 50
19
20 # Training Data
21 train_X = numpy.asarray([3.3,4.4,5.5,6.71,6.93,4.168,9.779,6.182,7.59,2.167,
22                          7.042,10.791,5.313,7.997,5.654,9.27,3.1])
23 train_Y = numpy.asarray([1.7,2.76,2.09,3.19,1.694,1.573,3.366,2.596,2.53,1.221,
24                          2.827,3.465,1.65,2.904,2.42,2.94,1.3])
25 n_samples = train_X.shape[0]
26
27 # tf Graph Input
28 X = tf.placeholder("float")
29 Y = tf.placeholder("float")
30
31 # Set model weights
32 W = tf.Variable(rng.randn(), name="weight")
33 b = tf.Variable(rng.randn(), name="bias")
34
35 # Construct a linear model
36 pred = tf.add(tf.multiply(X, W), b)
37
38 # Mean squared error
39 cost = tf.reduce_sum(tf.pow(pred-Y, 2))/(2*n_samples)
40 # Gradient descent
41 # Note, minimize() knows to modify W and b because Variable objects are trainable=True by default
42 optimizer = tf.train.GradientDescentOptimizer(learning_rate).minimize(cost)
43
44 # Initialize the variables (i.e. assign their default value)
45 init = tf.global_variables_initializer()
```

1-45까지 세팅 (강의에서 디테일하게 커버 예정)

8-12 코드를 돌리기 위한 라이브러리 설정

15-18 학습의 파라미터 설정

21-25까지 데이터 로딩

28-29까지 데이터가 들어갈 placeholder 세팅

32-33까지 학습하면서 업데이트 될 weights 세팅

36 weights와 데이터 (x)를 이용한 prediction 설정

39 계산된 prediction과 실제 값의 차이를 나타내는 cost function 설정

42 cost function을 최소화하면서 weights를 업데이트할 optimization 설정

여기까지가 tensorflow graph 설정입니다...

# Tensorflow linear regression 예제

```
47 # Start training
48 with tf.Session() as sess:
49
50     # Run the initializer
51     sess.run(init)
52
53     # Fit all training data
54     for epoch in range(training_epochs):
55         for (x, y) in zip(train_X, train_Y):
56             sess.run(optimizer, feed_dict={X: x, Y: y})
57
58         # Display logs per epoch step
59         if (epoch+1) % display_step == 0:
60             c = sess.run(cost, feed_dict={X: train_X, Y: train_Y})
61             print("Epoch:", '%04d' % (epoch+1), "cost=", "{:.9f}".format(c), \
62                   "W=", sess.run(W), "b=", sess.run(b))
63
64     print("Optimization Finished!")
65     training_cost = sess.run(cost, feed_dict={X: train_X, Y: train_Y})
66     print("Training cost=", training_cost, "W=", sess.run(W), "b=", sess.run(b), '\n')
67
68     # Graphic display
69     plt.plot(train_X, train_Y, 'ro', label='Original data')
70     plt.plot(train_X, sess.run(W) * train_X + sess.run(b), label='Fitted line')
71     plt.legend()
72     plt.show()
73
74     # Testing example, as requested (Issue #2)
75     test_X = numpy.asarray([6.83, 4.668, 8.9, 7.91, 5.7, 8.7, 3.1, 2.1])
76     test_Y = numpy.asarray([1.84, 2.273, 3.2, 2.831, 2.92, 3.24, 1.35, 1.03])
77
78     print("Testing... (Mean square loss Comparison)")
79     testing_cost = sess.run(
80         tf.reduce_sum(tf.pow(pred - Y, 2)) / (2 * test_X.shape[0]),
81         feed_dict={X: test_X, Y: test_Y}) # same function as cost above
82     print("Testing cost=", testing_cost)
83     print("Absolute mean square loss difference:", abs(
84         training_cost - testing_cost))
85
86     plt.plot(test_X, test_Y, 'bo', label='Testing data')
87     plt.plot(train_X, sess.run(W) * train_X + sess.run(b), label='Fitted line')
88     plt.legend()
89     plt.show()
```

텐서플로우는 graph 설정 후, sessio을 실행하여 placeholder에 데이터를 넣어주면서 graph에서 설정한 machine learning algorithm (여기는 선형회기식  $w*x+b=y$ , line 36)을 최적화하는 방식으로 진행됩니다.

54-62 몇번을 돌아가면서 정의한 선형회기식을 최적화함 session.run(optimizer...)

64-72 학습의 결과를 보여주고 그래프로 나타냄

75-89 테스트 데이터 (학습에 없던 데이터)를 제공하여 학습된 선형회기식에 적용... 그 후 그 결과를 보여줌