

<https://wikidocs.net/2825>

1. 1) 아나콘다 설치 파일 다운로드

아나콘다는 Anaconda(이전: Continuum Analytics)라는 곳에서 만든 파이썬 배포판으로, 수백 개의 파이썬 패키지를 포함하고 있습니다. 또한 회사 내에서도 상업용으로 무료로 사용할 수 있다는 장점이 있습니다.

먼저 아나콘다 배포판을 설치하기 위해 다음 URL 로 이동합니다. 파이썬은 현재 2.7 버전과 3.7 버전이 있는데, 이 책에서는 파이썬 3.7 을 사용합니다.

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

파이썬 3.7 버전에도 다시 윈도우 64 비트용 설치 파일과 32 비트용 설치 파일이 있습니다. 보통은 사용 중인 PC 에 64 비트 윈도우가 설치돼 있으면 64 비트용 설치 파일을 내려받고, 32 비트 윈도우가 설치돼 있다면 32 비트용 설치 파일을 내려받으면 됩니다.

그러나 이 책의 후반부에서 다룰 국내 증권사 API 를 정상적으로 사용하려면 반드시 그림 1.3 과 같이 Windows 를 먼저 선택한 후 Python 3.7 version 에서 32 비트용 설치 파일(32-Bit Graphical Installer)을 클릭해서 내려받아야 합니다. 특별한 경우가 아니라면 파이썬 3.7 32 비트용 설치 파일을 내려받기 바랍니다. 참고로 내려받은 파일의 이름에 x64 가 포함되어 있다면 64 비트용 설치 파일을 받으신 겁니다.

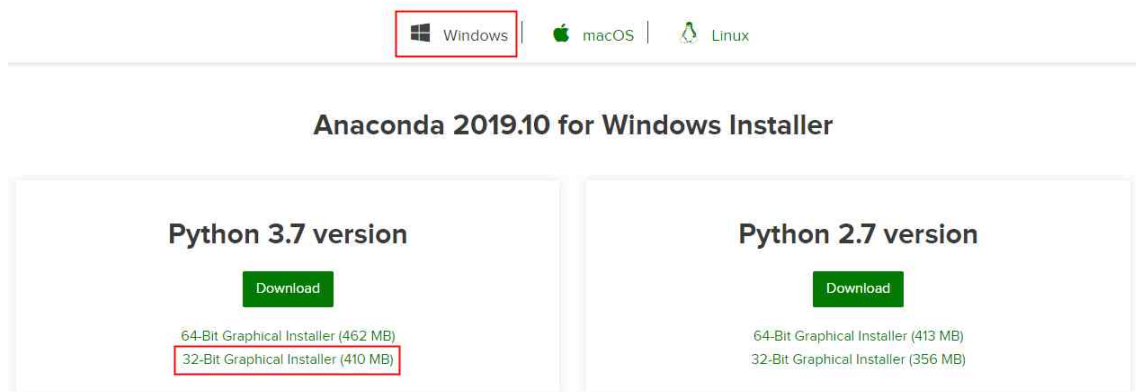


그림 1.3 아나콘다 32 비트 배포판 설치 파일 선택

<https://wikidocs.net/2826>

2. 2) 아나콘다 설치

최신 버전 설치는 다음 페이지를 참고하세요.

<https://wikidocs.net/77491>

아나콘다 32 비트 배포판 설치 파일을 정상적으로 내려받았다면 해당 파일(Anaconda3-2019.10-Windows-x86.exe)에 마우스 오른쪽 버튼을 클릭한 후 [관리자 권한으로 실행] 메뉴를 선택해서 설치를 시작합니다. 그림 1.4 와 같이 첫 번째 설치 화면이 나오면 [Next] 버튼을 클릭해 다음 단계로 진행합니다.

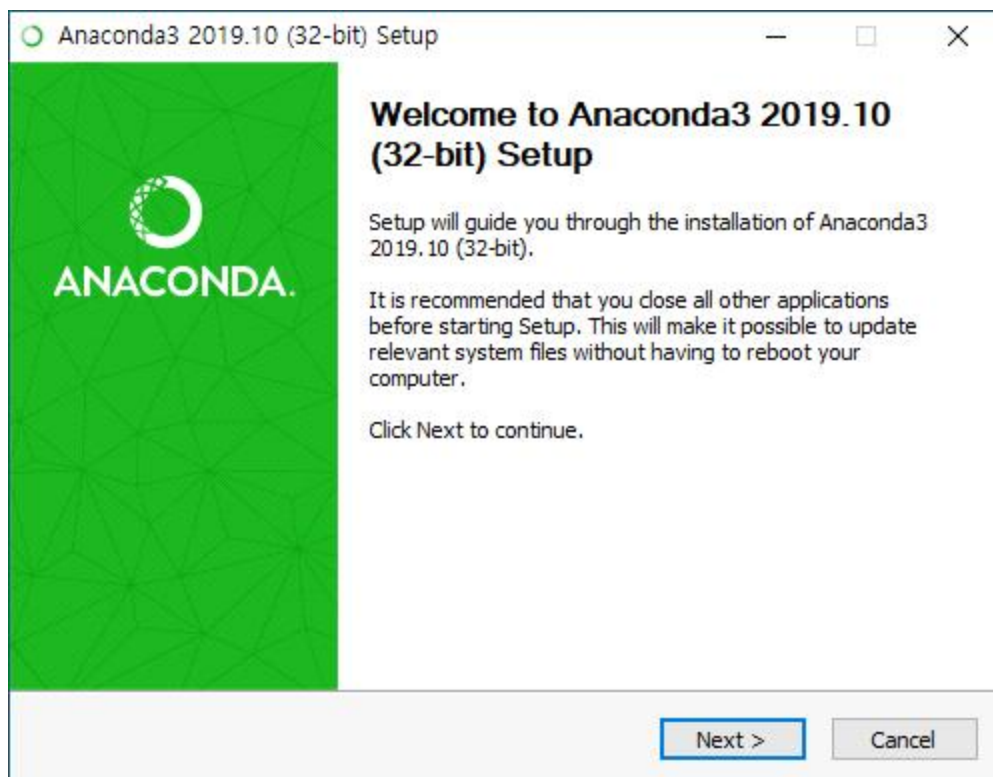


그림 1.4 아나콘다 설치(1/8)

그림 1.5 는 라이선스 동의에 관한 내용입니다. [I Agree] 버튼을 클릭해 다음 단계로 이동합니다.

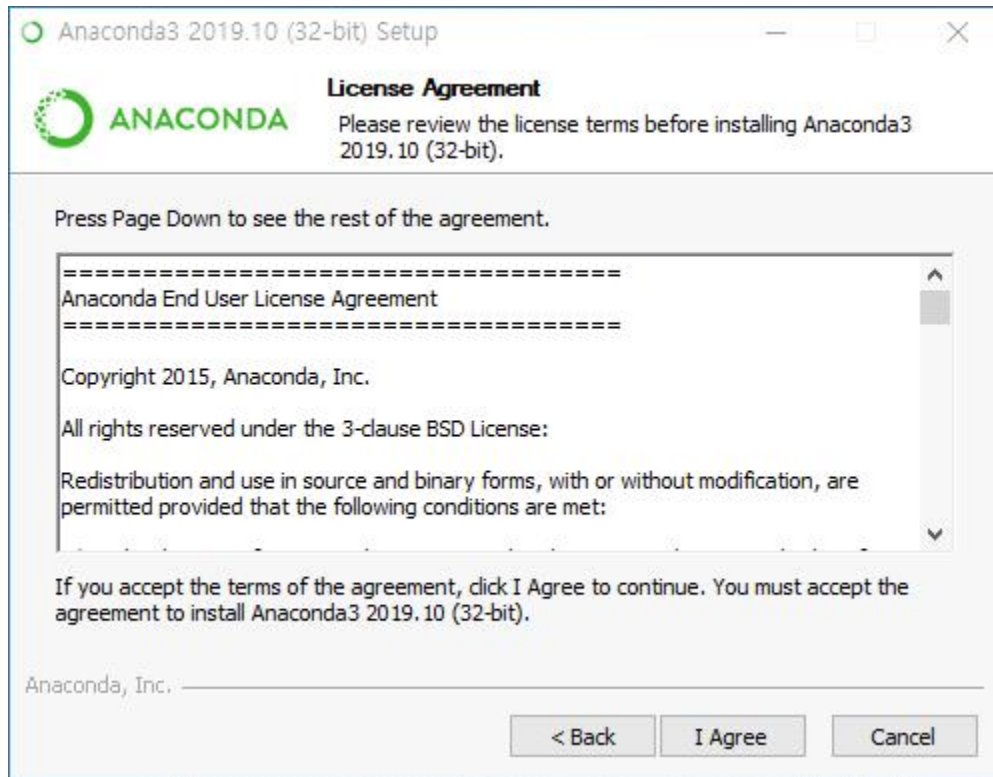


그림 1.5 아나콘다 설치(2/8)

설치 유형을 선택하는 단계에서는 [All Users]를 선택한 후 [Next] 버튼을 클릭합니다.

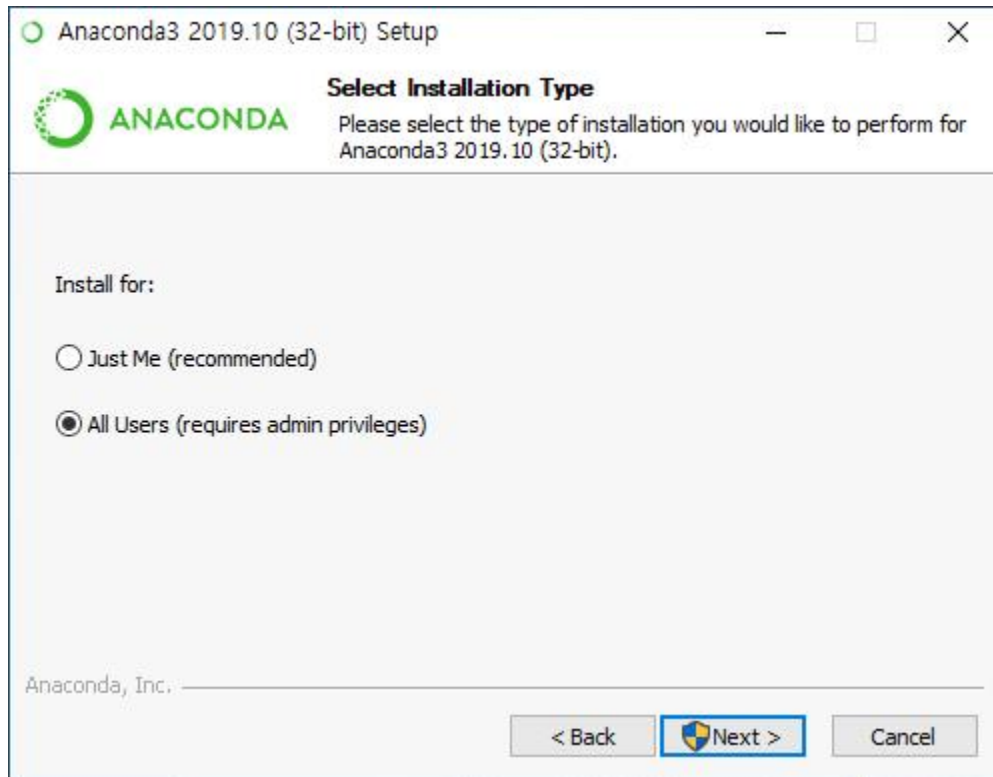


그림 1.6 아나콘다 설치(3/8)

아나콘다 배포판이 설치될 디렉터리를 선택하는 화면에서는 [Destination Folder]를 'C:\Anaconda3'으로 변경한 후 [Next] 버튼을 클릭합니다.

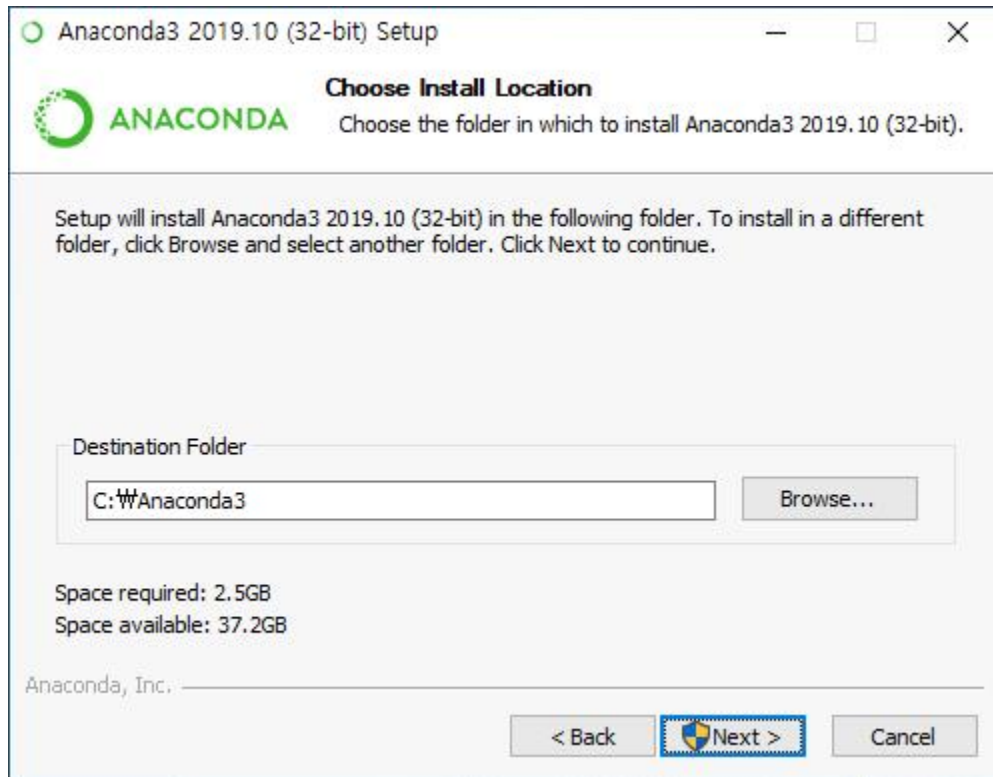


그림 1.7 아나콘다 설치(4/8)

추가옵션선택화면에서 그림 1.8 과 같이 두 가지 옵션을 모두 선택한 후 [Install] 버튼을 클릭해 다음 단계로 이동합니다.

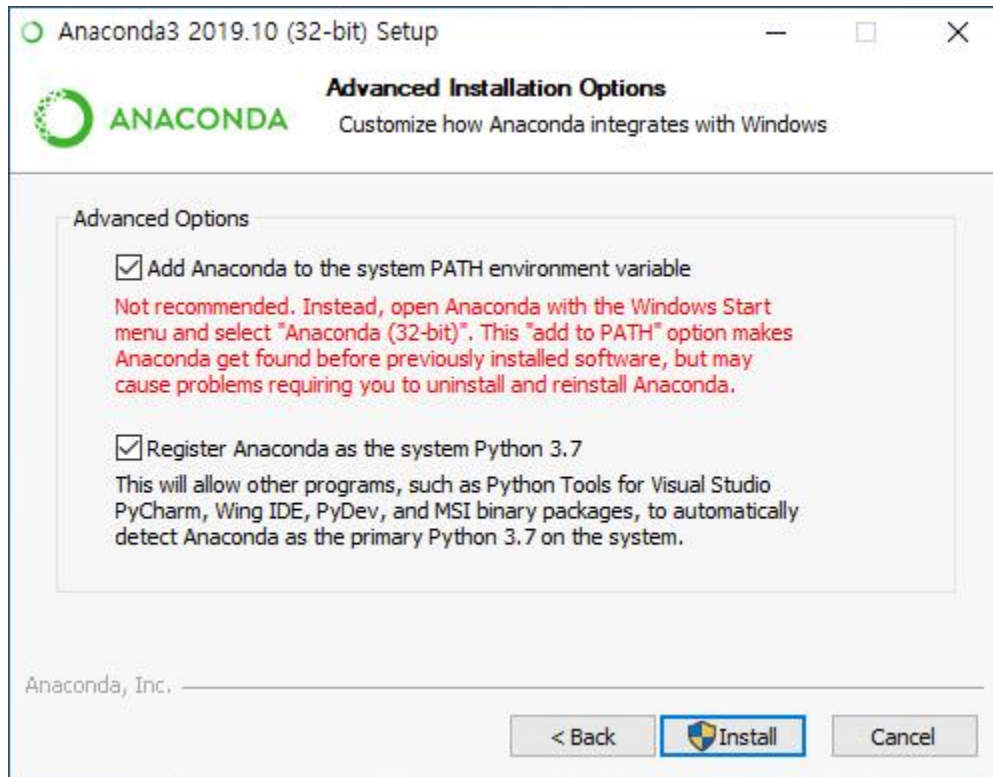


그림 1.8 아나콘다 설치(5/8)

설치가 완료되면 화면에 'Completed'라는 메시지가 출력됩니다. 이때 [Next] 버튼을 눌러 다음 단계로 이동합니다.

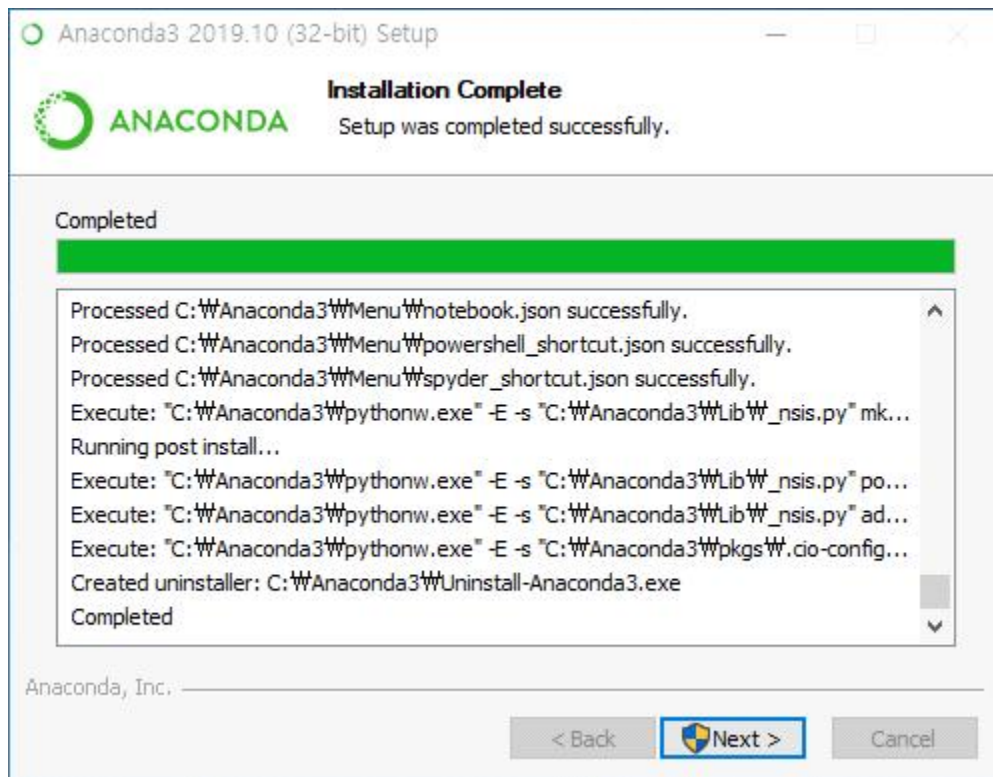


그림 1.9 아나콘다 설치(6/8)

그림 1.10 에서 [Skip] 버튼을 클릭해 다음 단계로 넘어갑니다.

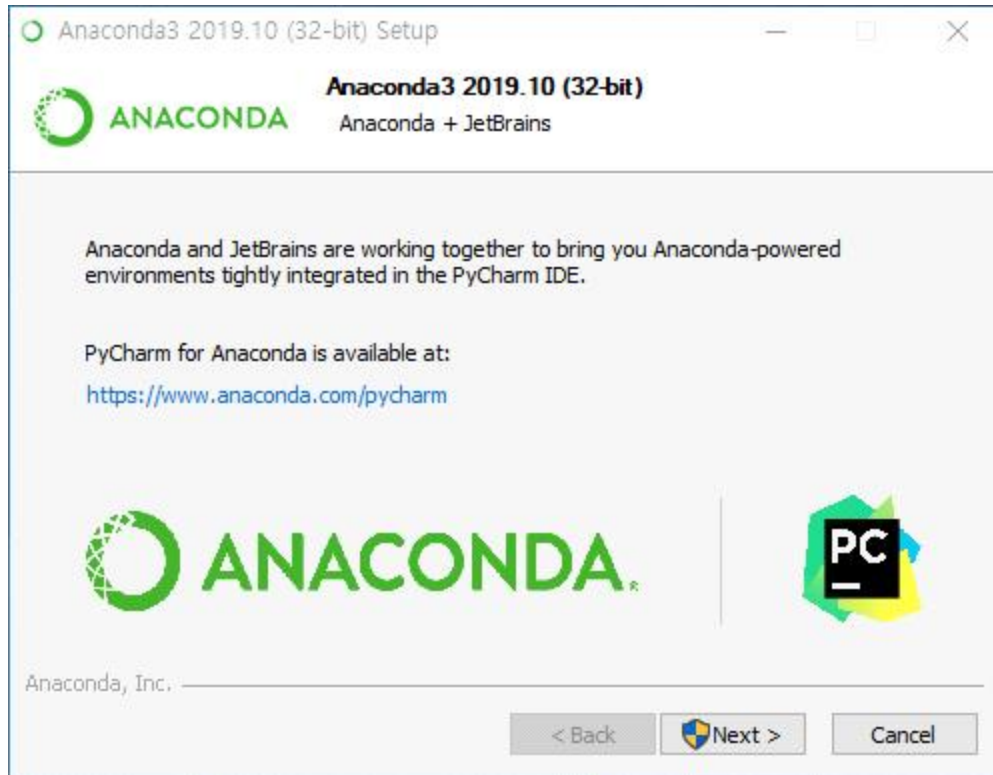


그림 1.10 아나콘다 설치 (7/8)

이로써 아나콘다 배포판의 모든 설치 과정이 완료되었습니다. 그림 1.11 과 같이 [Finish] 버튼을 클릭해 설치 과정을 종료합니다.

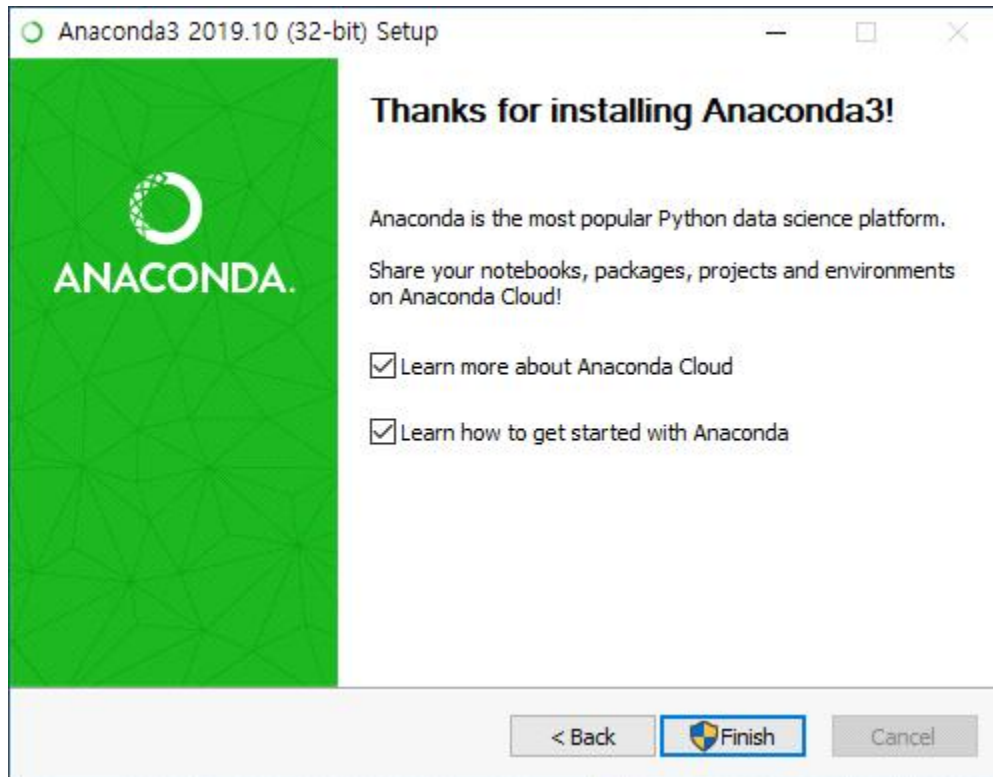


그림 1.11 아나콘다 설치 (8/8)

파이썬 인터프리터 실행

1.2절을 통해 PC에 정상적으로 '아나콘다'라는 이름의 파이썬 배포판을 설치했습니다. 아마도 프로그래밍이란 것을 한 번도 해보지 않는 분들은 배우기도 전에 뭔가 복잡한 설치 과정이 있어 조금 당황스러우셨을 겁니다. 그렇지만 옛말에 '시작이 반'이라고 했습니다. 조금만 참고 따라오시면 재미있는 프로그래밍의 세계로 입문하실 수 있을 것입니다.

앞에서 아나콘다 배포판을 'C:\Anaconda3'에 설치했습니다. 먼저 윈도우의 탐색기 등을 사용해 해당 경로로 이동해 봅시다. 그림 1.12와 같이 해당 경로에 여러 파일과 디렉터리가 설치돼 있음을 확인할 수 있습니다. 이 경로에서 Lib 디렉터리로 이동한 후 다시 idllib 디렉터리로 이동합니다. 최종 경로는 다음과 같습니다.

C:\Anaconda3\Lib\idlelib

위의 경로에도 여러 디렉터리와 파일이 있는데 그림 1.13과 같이 idle.bat 파일에 마우스 오른쪽 버튼을 클릭한 후 [보내기(N)] → [바탕 화면에 바로 가기 만들기]를 선택합니다. 파이썬으로 프로그래밍 하려면 파이썬 IDLE(Integrated DeveLopment Environment)가 필요한데, 해당 파일을 더 쉽게 실행하기 위해 바탕 화면에 바로 가기 아이콘을 생성한 것입니다. 이 과정을 모두 잘 따라 했다면 그림 1.14와 같이 바탕 화면에 아이콘이 생성된 것을 확인할 수 있습니다.



PyCharm을 이용한 개발 환경 구축

이번 장에서는 지금까지 배운 파이썬 기초 문법을 활용해 간단한 프로젝트를 진행해보겠습니다. 이를 통해 지금까지 배운 파이썬 문법이 실제 프로그램 개발에 어떻게 사용되는지 감을 잡을 수 있을 것입니다. 그리고 더 나아가 프로그램 개발에 대한 자신감도 얻을 수 있을 것입니다.

프로그래밍을 처음 배우면 언어의 문법은 어느 정도 알 것 같은데, 막상 프로그램을 작성하려고 하면 어디서부터 어떻게 시작해야 할지 감이 오지 않는 경우가 많습니다. 이러한 단계에서 더 발전하려면 꼭 코드를 직접 입력해 보면서 이번 장에서 소개하는 프로젝트를 완성해 보기 바랍니다.

지금까지 Python IDLE를 이용해 파이썬 코드를 작성했습니다. Python IDLE는 사용하기 편하고 아나콘다 같은 파이썬 배포판이나 파이썬 공식 설치 프로그램만을 설치해도 바로 사용할 수 있어서 초보자에게 적합한 개발 환경입니다. 하지만 조금 규모가 있는 프로그램을 작성할 때는 Python IDLE만으로는 부족한 점이 있습니다.

전문 개발자는 보통 통합 개발 환경이라고 하는 IDE(Integrated Development Environment)를 이용해 프로그램을 개발합니다. 음악을 업으로 하는 사람들이 좋은 악기를 사용해서 연주하는 것을 생각해보면 더 좋은 개발 환경을 사용하는 상황이 쉽게 이해될 겁니다.

IDE는 변수나 함수 이름을 자동으로 완성하는 기능을 제공하고 있어 개발자가 더욱 빠르게 코드를 작성할 수 있게 합니다. 또한 코드의 실행과 결과 확인을 IDE를 통해 한 번에 처리할 수 있어 편리합니다. 그뿐만 아니라 작성한 프로그램이 정상적으로 동작하지 않을 때는 디버거(debugger)를 이용해 코드를 쉽게 분석할 수 있는 기능도 제공합니다. 최근에는 SVN이나 Git과 같은 버전 관리 시스템도 지원하고 있어 프로그램 개발의 전반적인 과정을 IDE를 통해 모두 처리할 수 있습니다. 이제 여러분도 IDE를 사용해 파이썬 프로그램을 개발해봅시다.

파이썬용 IDE에는 비주얼 스튜디오 기반의 PTVS(Python Tools Visual Studio), 이클립스 기반의 PyDev, 그리고 최근 인기가 높은 JetBrains의 PyCharm이 있습니다. 이 책에서는 이 가운데 최근 인기가 높고 무료로 사용할 수 있는 PyCharm을 사용하겠습니다.

- PTVS: <http://microsoft.github.io/PTVS/>
- PyDev: <http://www.pydev.org/>
- PyCharm: <https://www.jetbrains.com/pycharm/download/>
- PyCharm: <https://www.jetbrains.com/pycharm/download/>

3. PyCharm 설치

다운로드한 Community Edition 설치 파일을 실행하면 그림 8.2 와 같은 설치 화면이 나타납니다. [Next] 버튼을 클릭해 다음 화면으로 이동합니다.

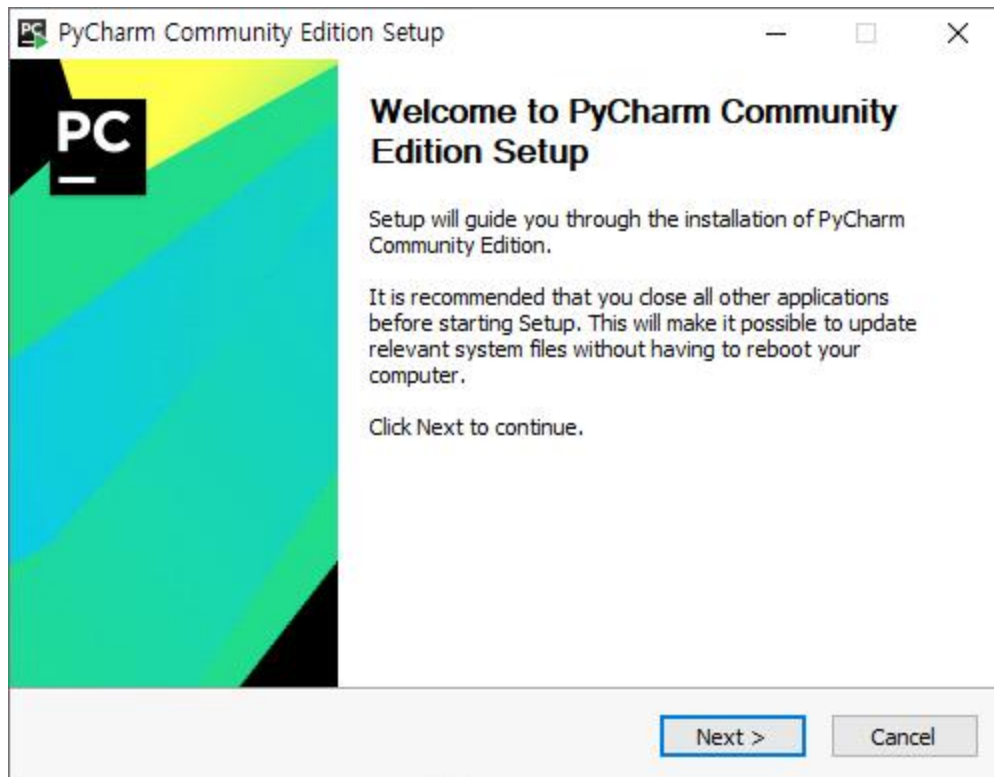


그림 8.2 PyCharm 설치 과정 1

그림 8.3 은 설치 위치를 변경할 수 있는 화면입니다. 기본 설정을 그대로 사용하는 것으로 하고 [Next] 버튼을 클릭해 다음 설치 화면으로 이동합니다.

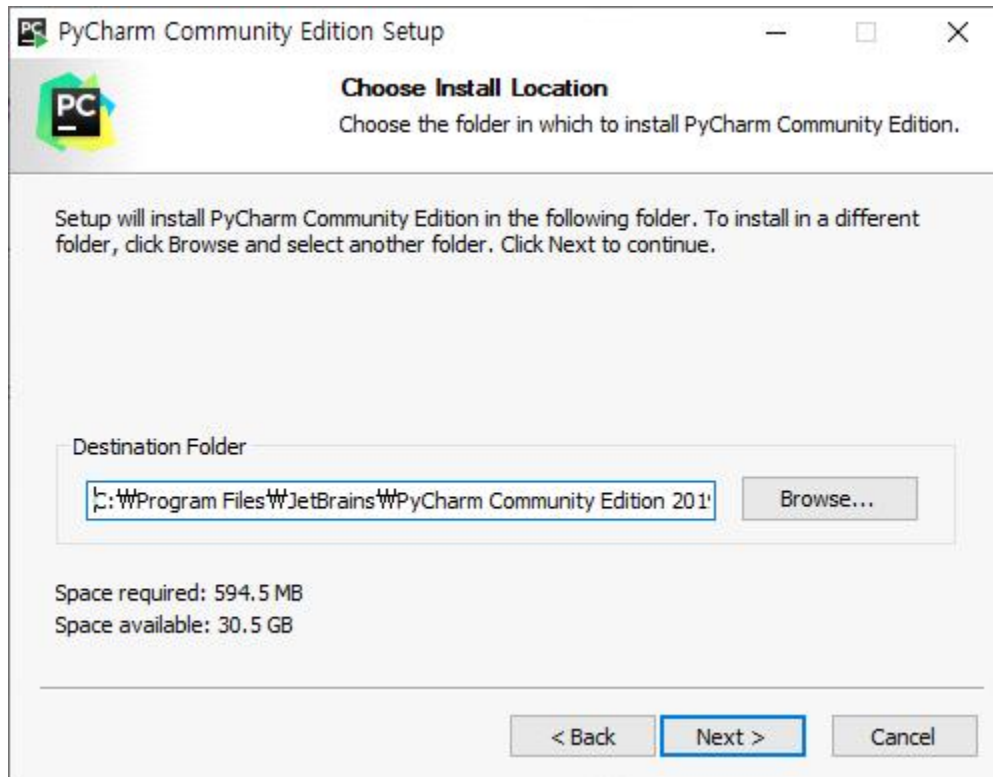


그림 8.3 PyCharm 설치 과정 2

그림 8.4 와 같이 필요한 항목에 체크한 후 [Next] 버튼을 클릭합니다. 'Create Desktop Shortcut'은 바탕화면에 PyCharm 아이콘을 생성해주는 옵션입니다. 'Update context menu'는 마우스 오른쪽 버튼을 누를 때 PyCharm 관련 메뉴가 나타나게 해줍니다. Create Associations 는 확장자가 py 인 파일에 대해서 PyCharm 으로 실행되도록 해줍니다.

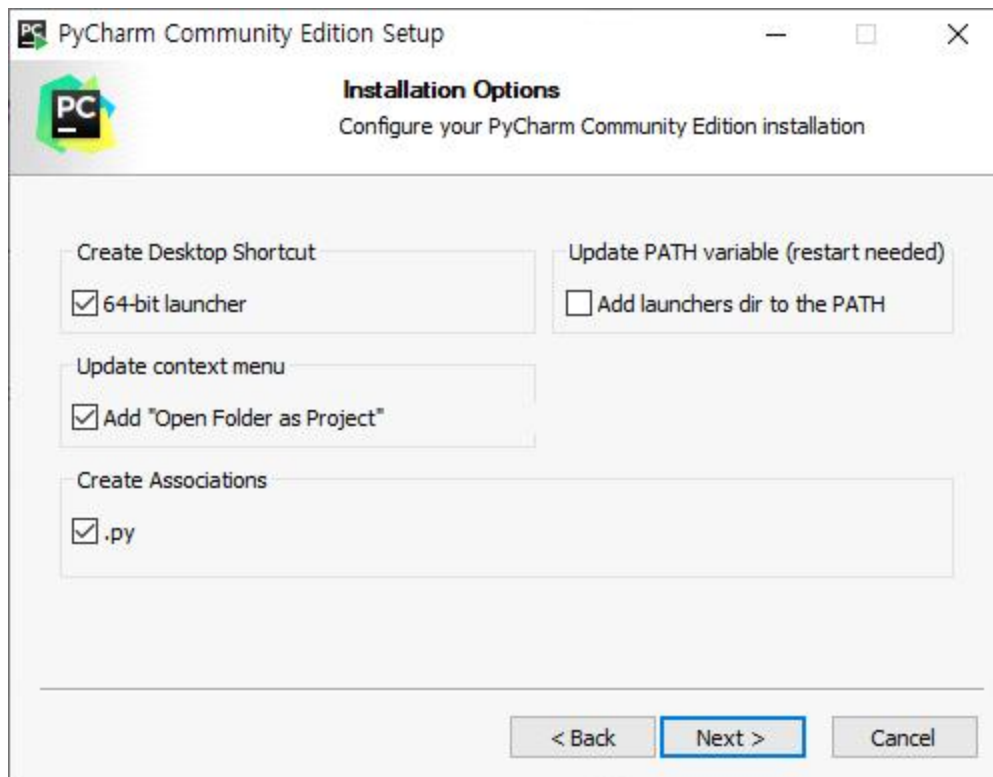


그림 8.4 PyCharm 설치 과정 3

그림 8.5 는 윈도우 시작 메뉴 폴더에 등록되는 이름을 변경할 수 있는 화면입니다. 기본값을 그대로 두고 [Install] 버튼을 클릭해 설치를 진행합니다.

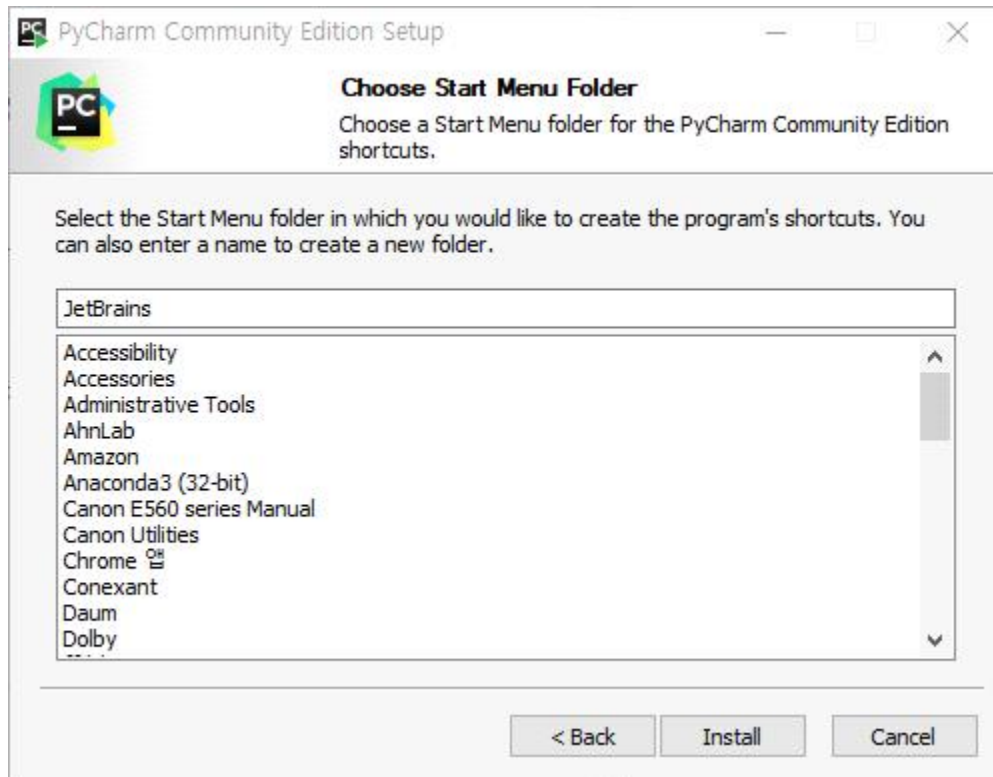


그림 8.5 PyCharm 설치 과정 4

그림 8.6은 PyCharm의 설치가 끝났음을 알리는 화면입니다. [Finish] 버튼을 클릭해 설치를 마칩니다.

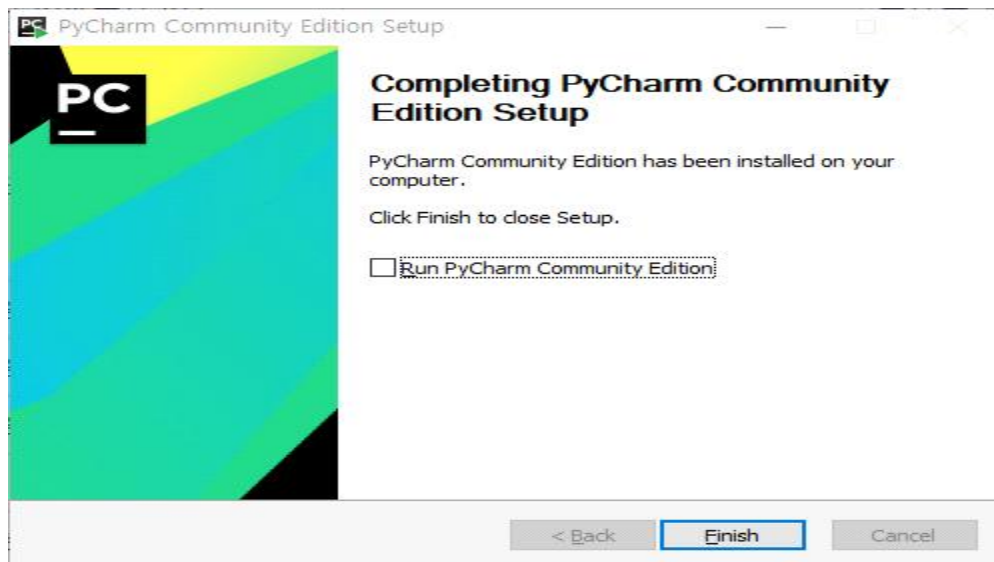


그림 8.6 PyCharm 설치 과정

4. PyCharm 초기 설정

PyCharm 설치가 완료되면 윈도우 바탕 화면에 PyCharm 아이콘이 만들어집니다. 해당 아이콘을 클릭해 PyCharm을 실행해 봅시다. PyCharm을 설치한 후 처음 실행하면 그림 8.7과 같은 화면이 나타납니다. 설정값을 그대로 둔 채 [OK] 버튼을 클릭합니다. Data Sharing Options 창에서는 [Cancel] 버튼을 클릭합니다.

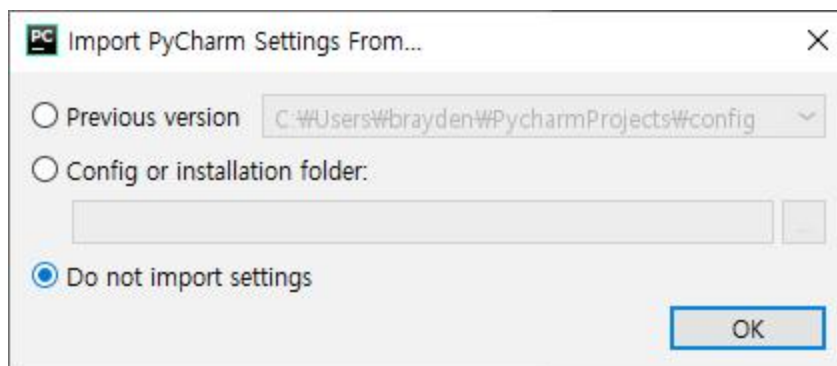
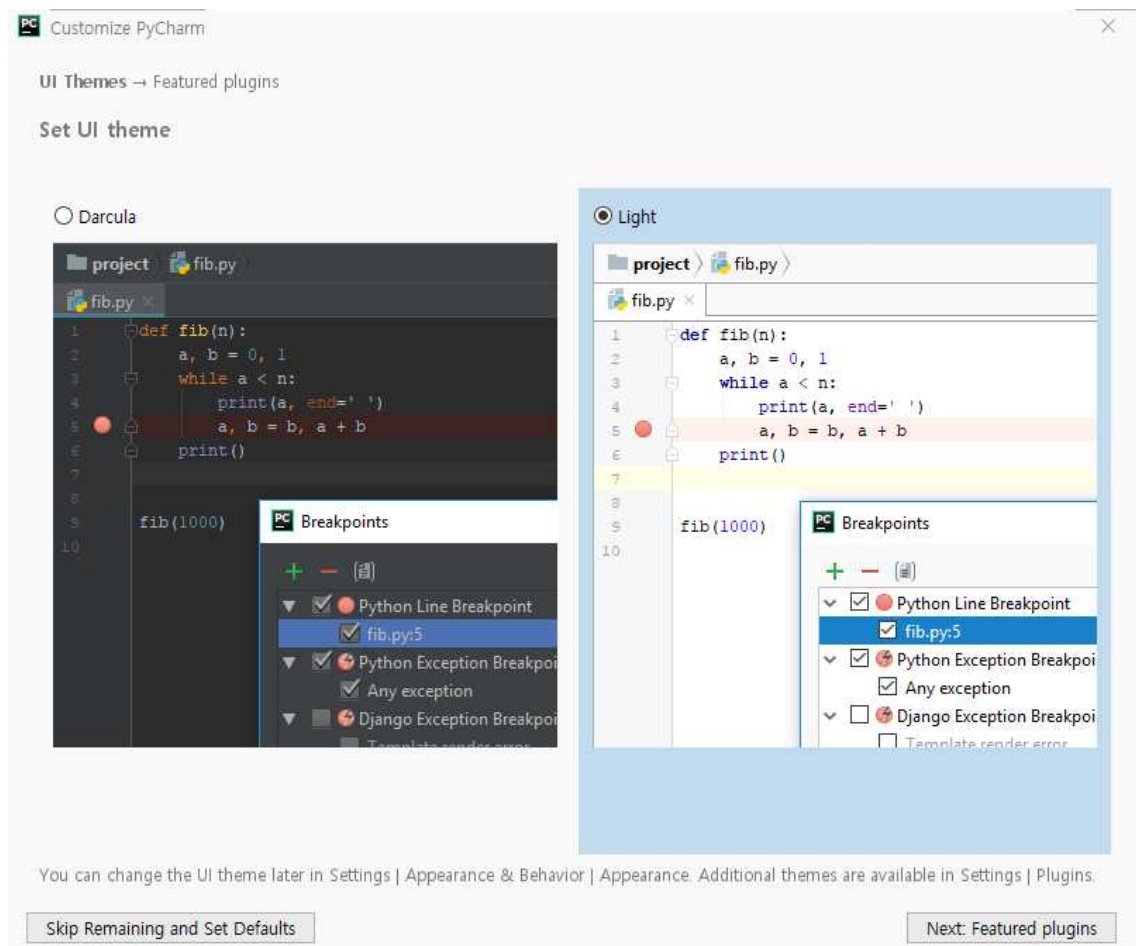


그림 8.8은 PyCharm 키보드 단축키와 IDE 테마 등을 선택하는 화면입니다. 이 부분은 각자 취향에 맞춰 설정하는 부분인데 저자는 그림 8.8과 같이 Light 테마를 선택했고 [Skip Remaining and Set Defaults] 버튼을 눌러 나머지는 기본 값으로 설정했습니다.



5. 프로젝트 생성

PyCharm은 다른 IDE와 마찬가지로 프로젝트 단위로 소스 파일을 관리합니다. 지금까지는 Python IDLE를 통해 프로그래밍했기 때문에 따로 작성된 코드를 파일로 저장하지는 않았습니다. 그러나 앞으로 큰 규모의 프로그램을 개발하기 위해 작성된 코드를 컴퓨터에 저장하겠습니다.

먼저 PyCharm을 실행한 후 그림 8.9와 같이 [Create New Project]를 클릭해 프로젝트 생성 화면으로 이동합니다.

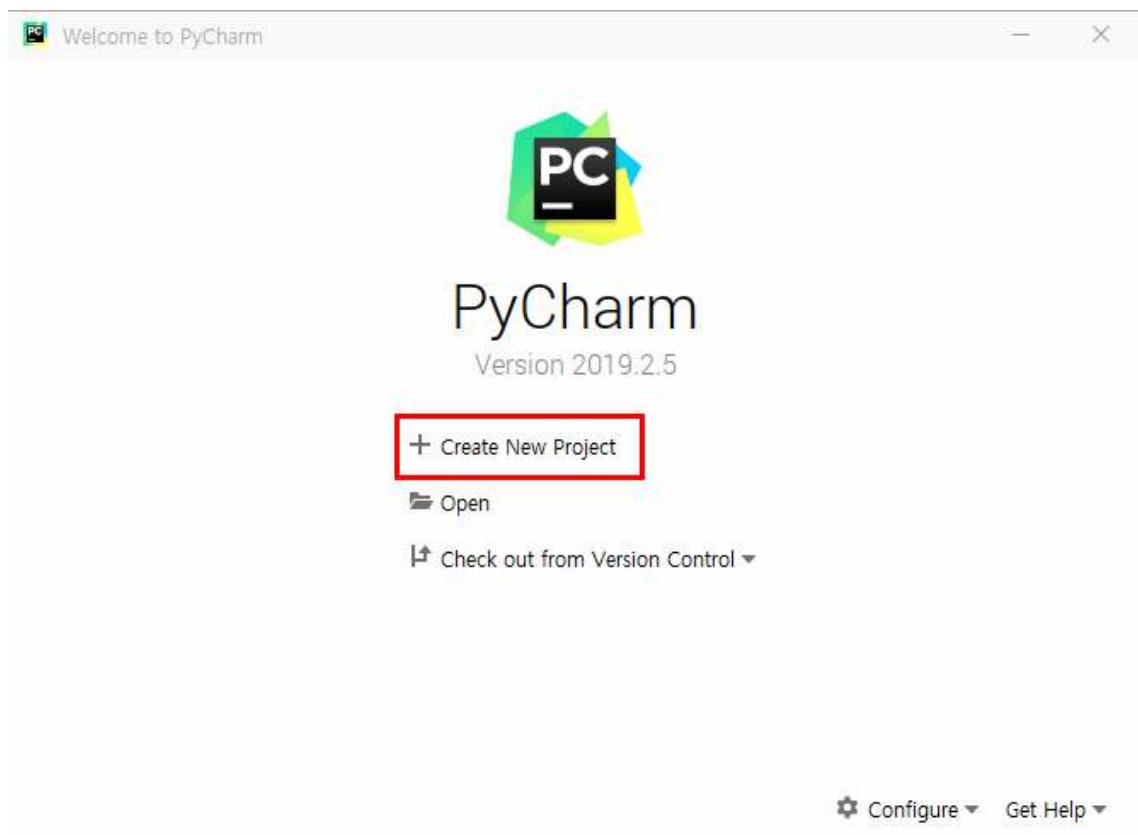


그림 8.10은 프로젝트 저장 위치와 파이썬 인터프리터를 설정하는 화면입니다. 그림 8.10과 같 [Location]의 마지막 부분을 'HelloWorld'라는 이름으로 변경한 후 Project Interpreter 항목을 클릭합니다. Existing interpreter 항목을 선택한 후 [...] 버튼을 클릭합니다.

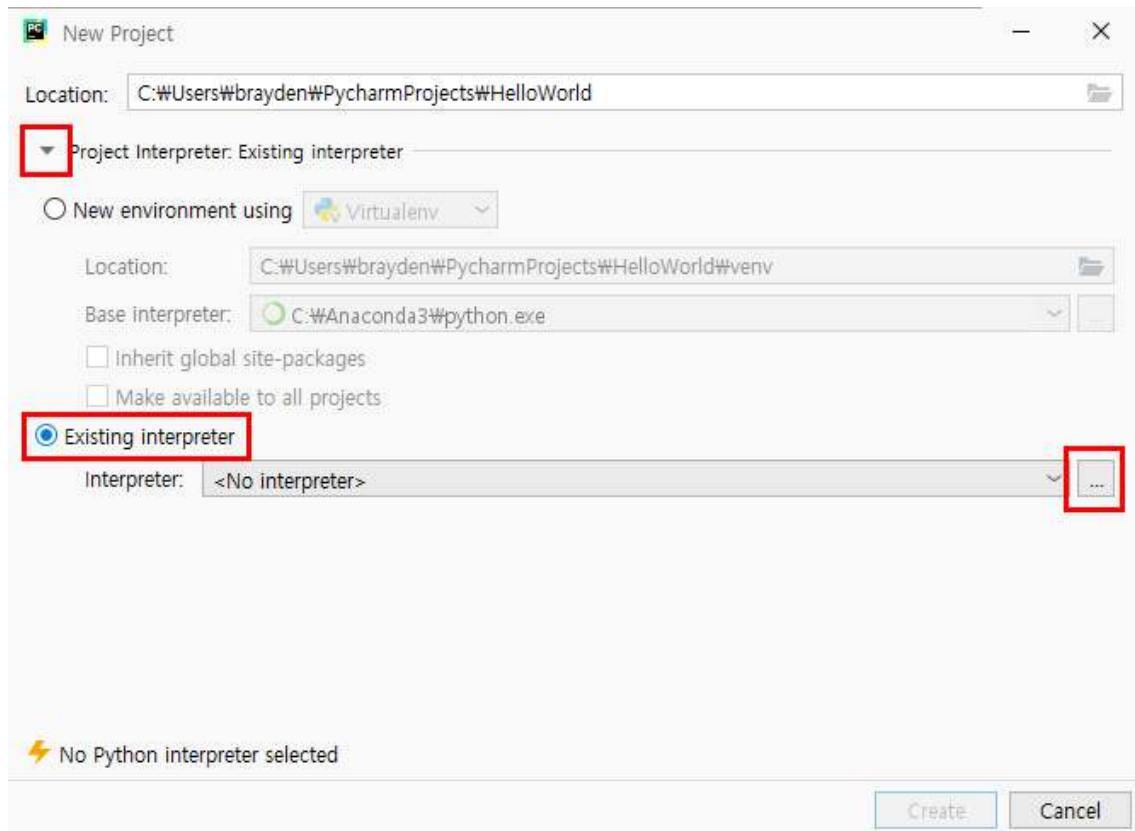


그림 8.11과 같이 System Interpreter 항목을 선택한 후 [...] 버튼을 누릅니다. 새 창이 나타나면 C:\Anaconda3\python.exe 파일을 선택한 후 [OK] 버튼을 누릅니다.

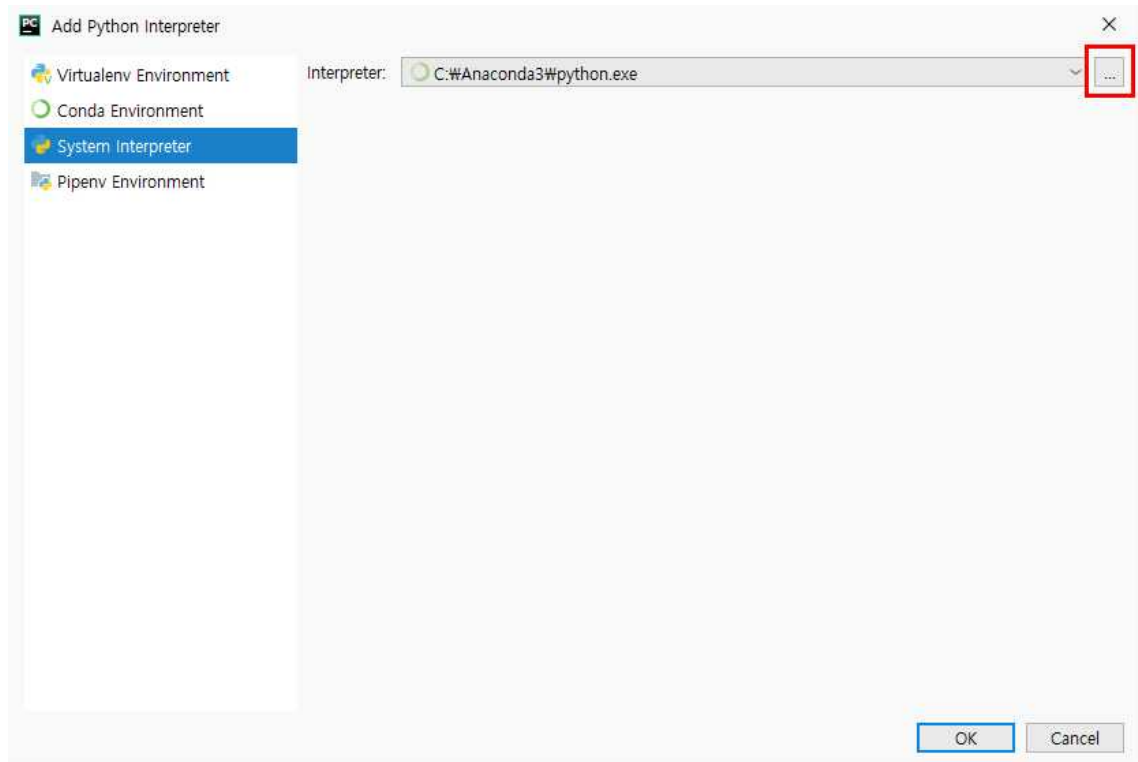


그림 8.12와 같이 설정했다면 [Create] 버튼을 눌러 프로젝트를 생성합니다.

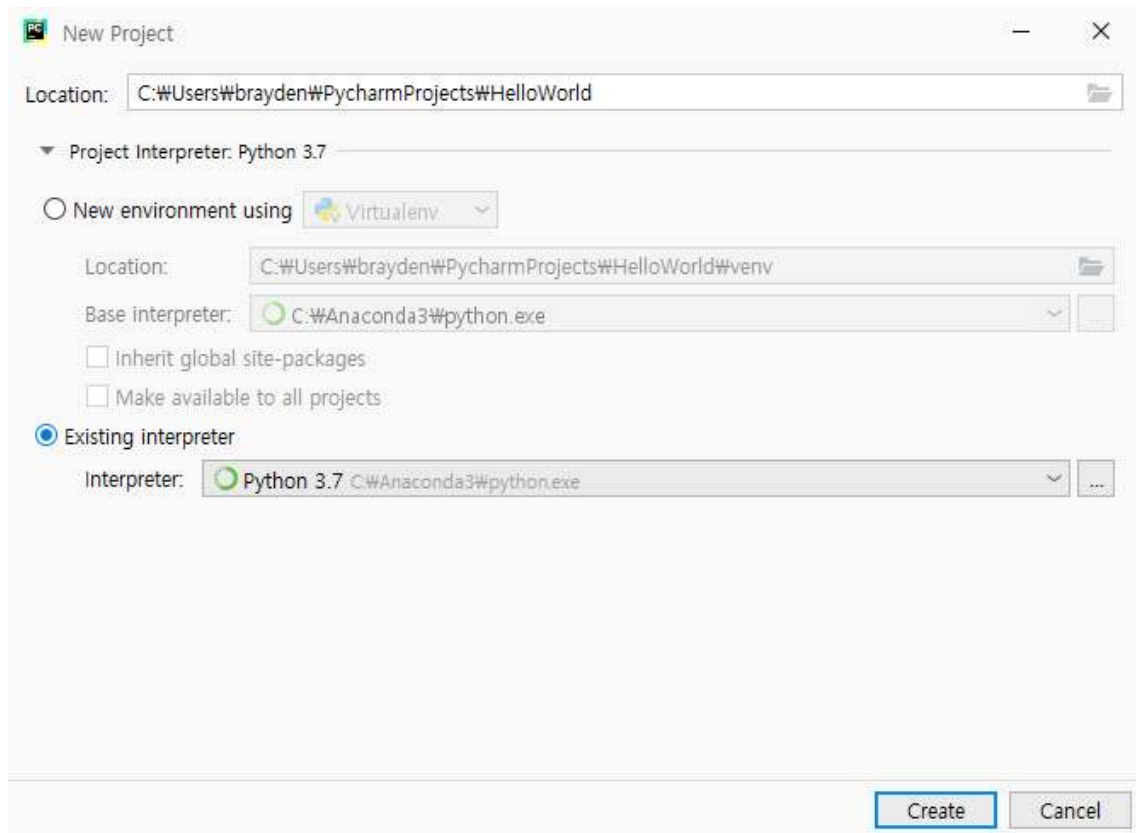
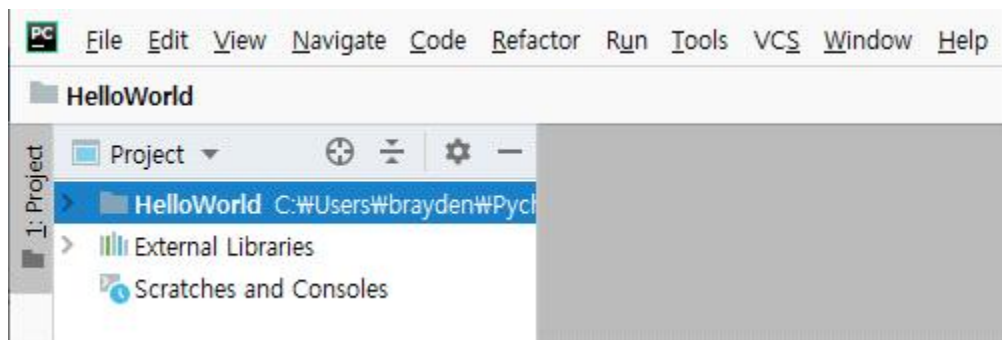


그림 8.13은 실행된 PyCharm의 화면 구성입니다. 프로젝트 부분에 HelloWorld가 보입니다. 이는 그림 8.10의 Location 부분에서 HelloWorld를 사용했기 때문입니다.



생성된 HelloWorld라는 프로젝트는 현재 비어있는 상태입니다. 프로젝트에 파이썬 파일을 추가하려면 그림 8.14와 같이 HelloWorld라는 프로젝트명에 마우스 오른쪽 버튼을 클릭한 후 [New]와 [Python File]을 순서대로 클릭합니다.

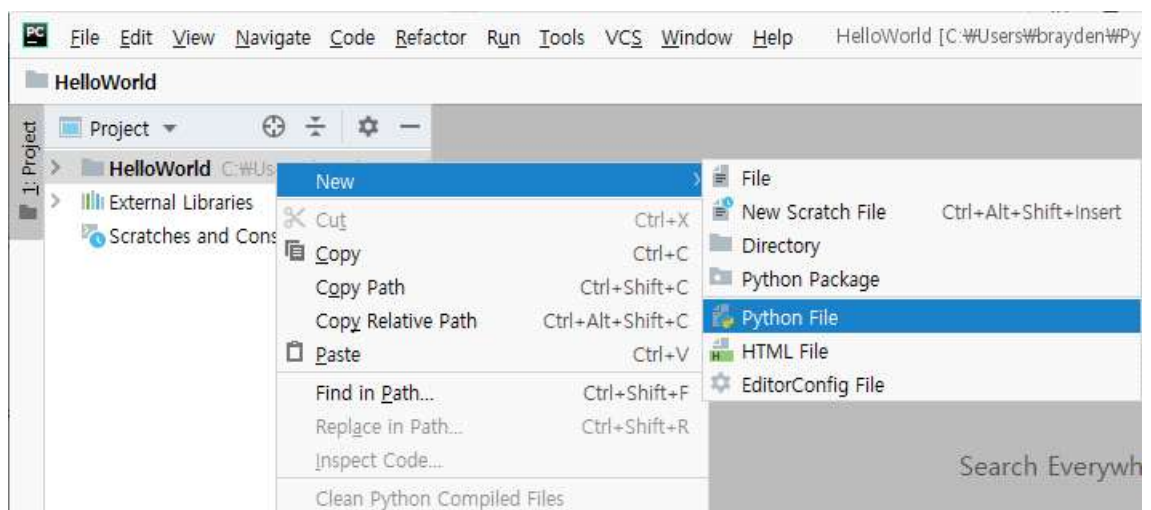
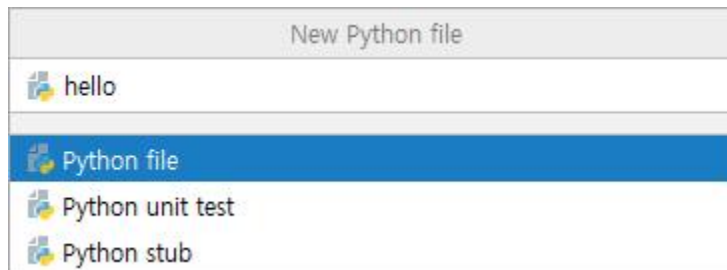
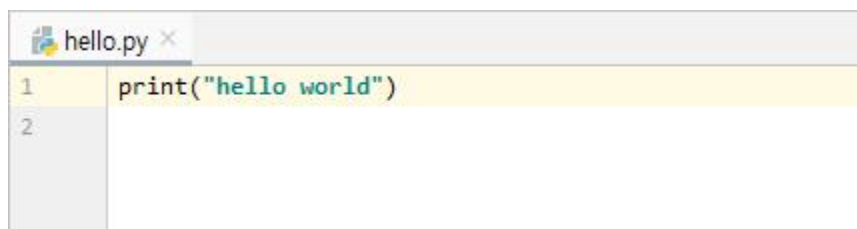


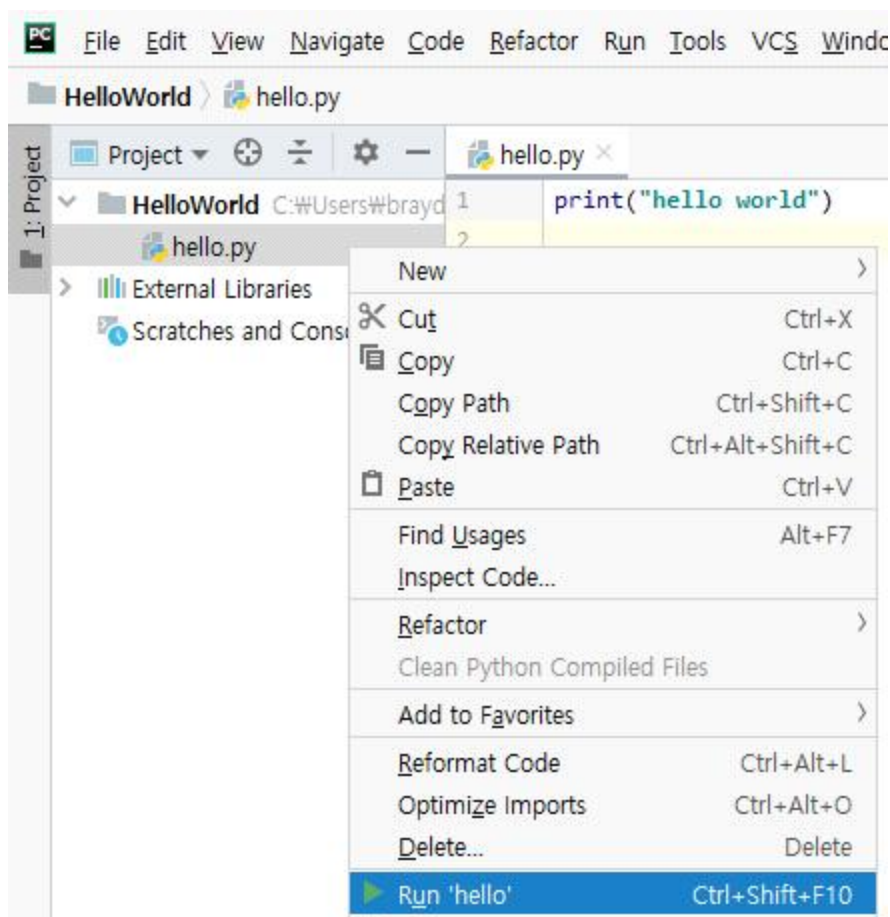
그림 8.15와 같이 새 파일을 생성하는 창에서 'hello'를 입력한 후 엔터키를 치면 hello.py라는 파일이 프로젝트에 추가됩니다.



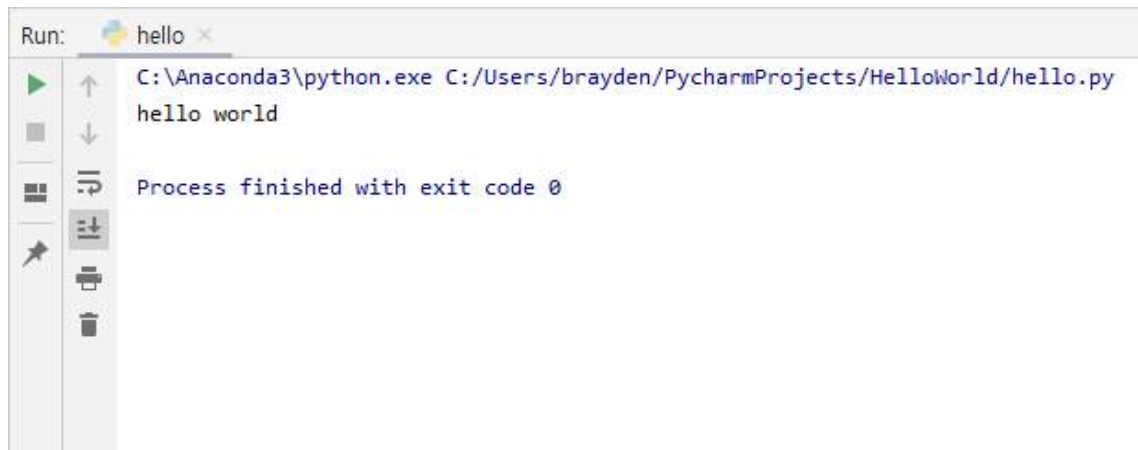
hello.py 파일에 그림 8.16과 같이 `print("hello world")`를 입력해 봅시다. 지금까지는 Python IDLE를 통해 명령어를 입력했지만 앞으로는 그림 8.16과 같이 파이썬 코드 파일에 입력하면 됩니다.



Python IDLE는 인터프리터라서 명령어를 입력할 때마다 해당 명령어를 바로 실행했습니다. 이와 달리 PyCharm은 코드를 작성한 후 실행 명령을 내리는 시점에 프로그램을 실행합니다. PyCharm에서 소스코드를 실행하려면 그림 8.17과 같이 소스코드에 마우스 오른쪽 버튼을 눌러 [Run] 메뉴를 클릭하거나 단축키인 Ctrl + Shift + F10을 누르면 됩니다. 참고로 PyCharm을 설치한 후 처음 실행하면 실행을 준비하는 데 시간이 소요됩니다. [Run] 메뉴가 활성화되지 않았다면 조금 기다렸다가 시도해보시기 바랍니다.



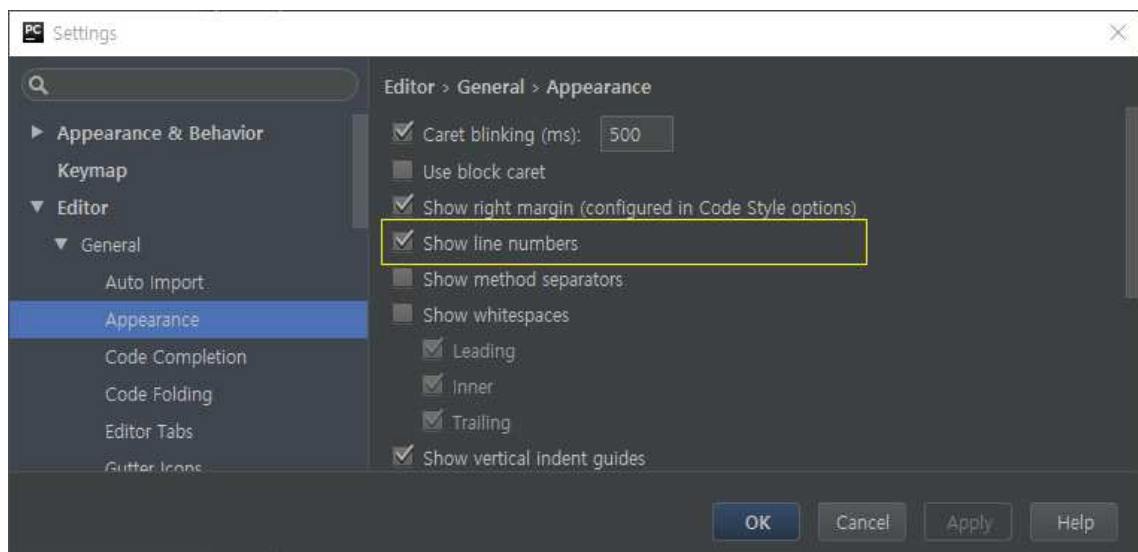
단축키 또는 [Run] 메뉴를 선택해 파이썬 코드를 실행하면 그림 8.18과 같이 PyCharm 하단부의 실행 창에 'hello world'라는 문자열이 출력됩니다.



6. PyCharm 기타 설정

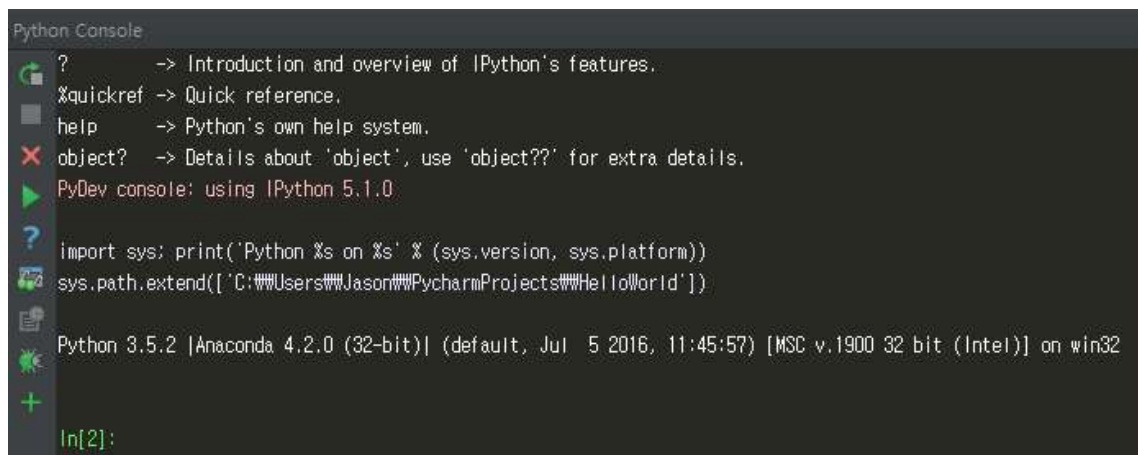
이번 절에서는 PyCharm에서 프로그램을 작성할 때 유용하게 사용할 수 있는 몇 가지 설정을 소개하겠습니다.

첫 번째로 소개할 내용은 파이썬 에디터 창에서 소스코드의 라인 넘버를 보이게 하는 설정입니다. 가끔 다른 누군가에게 코드의 특정 부분을 설명할 때 '몇 번째 줄에 있는 코드를 보라'처럼 표현하곤 합니다. PyCharm의 기본 설정은 에디터에 라인 넘버가 보이지 않습니다. PyCharm의 코드 에디터에서 라인 넘버를 보이게 하려면 [File] → [Settings] → [Editor] → [General] → [Appearance] 메뉴를 차례로 선택한 후 그림 8.19와 같이 [Show line numbers] 항목을 체크하면 됩니다.



두 번째로 소개할 내용은 PyCharm에서 파이썬 인터프리터를 사용하는 기능입니다. PyCharm으로 프로그램을 작성하다 보면 간단한 코드를 파이썬 인터프리터로 확인해보고 싶을 때가 있습니다. [Tools] → [Python Console] 메뉴를 차례로 선택하면 그림 8.20과 같이 PyCharm의 하단부에 Python IDLE와 같은 파이썬 콘솔(Python Console)이 나타납니다.

이처럼 Python IDLE과 비슷한 기능을 수행하는 파이썬 콘솔을 동시에 사용함으로써 더욱 편리하게 개발할 수 있습니다.



```
Python Console
?      -> Introduction and overview of IPython's features.
%quickref -> Quick reference.
help   -> Python's own help system.
object? -> Details about 'object', use 'object??' for extra details.
PyDev console: using IPython 5.1.0
import sys; print('Python %s on %s' % (sys.version, sys.platform))
sys.path.extend(['C:\\Users\\Jason\\PycharmProjects\\HelloWorld'])
Python 3.5.2 |Anaconda 4.2.0 (32-bit)| (default, Jul 5 2016, 11:45:57) [MSC v.1900 32 bit (Intel)] on win32
In[2]:
```

7. G Suite smtpplib.SMTPAuthenticationError: (535, b'5.7.8 Username and Password not accepted.) send email through python

gsuite setting 에서 유저별로 apps manage 하게 해주기
smtpplib.SMTPAuthenticationError: (535, b'5.7.8 Username and Password not
accepted.)

email sender test code 만드는 중 보내는 사람 개인 gmail account 에서
회사메일 gsuite domain 계정으로 보내려고 하니 Error 발생.

에러 메시지로 구글에서 검색하면 아래의 내용이 뜬다.

[https://stackoverflow.com/questions/17759860/python-2-smtpserverdisconnecte
d-connection-unexpectedly-closed](https://stackoverflow.com/questions/17759860/python-2-smtpserverdisconnecte-d-connection-unexpectedly-closed)

아무래도 나는 여기에 해당이 안되는 것 같고..

뭐라고 검색해야 이 문제를 해결할 수 있을까?

먼저 숙지 해야할 사항에는 아래와 같다.

<https://support.google.com/accounts/answer/3466521>

-> gmail account 에 app pwd 를 설정해서 내 구글 account 가 아닌
gmail 에만 접근이 가능하단 사실을 알아야한다.

설정해야하는 이슈는

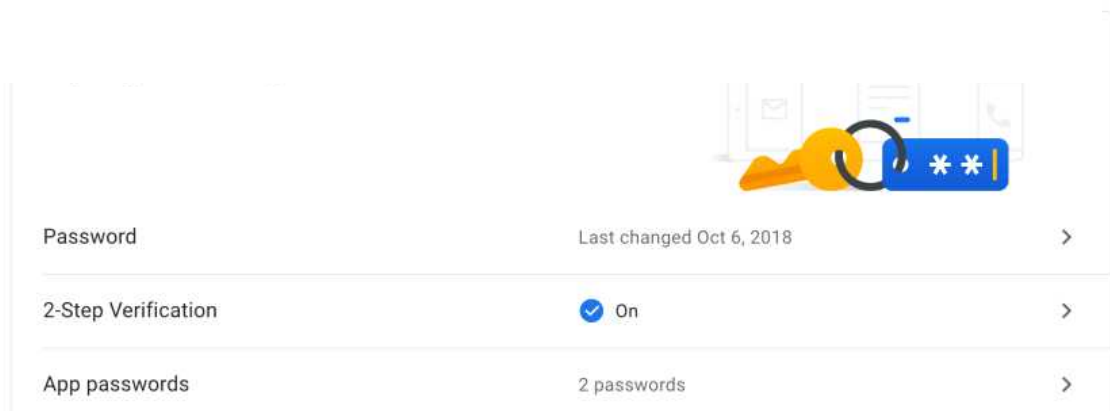
1. gmail 에 접속 가능한 app pwd 설정
2. Gsuite 관리자 설정에서 1 번 허용해주기

우선 1 번부터

<https://myaccount.google.com/u/0/security?pagelid=none>

에서 2 step verification 을 On 시키고, app password 를 추가한다. 이 때 생성된 비번은 16 자리 코드로 아래의 코드 중 pwd 에 위치한다.

smtp.login(email, pwd)



그 다음 2 번

admin.google.com 관리 콘솔에 접속 후

보안>기본설정으로 사용자 수준에서 아래 그림처럼 app 접근을 허용하는
걸로 수정하고

 Enforce access to less secure apps for all users (Not Recommended)

오른쪽 하단에 submit

더 자세한 것은 아래의 링크를 확인하면 된다.

<https://support.google.com/a/answer/6260879?hl=ko>

이렇게 하고 소스코드 실행하면 성공!

=====

성공 !!!

새 암호 ---> fcwwkgkglatjqwhx

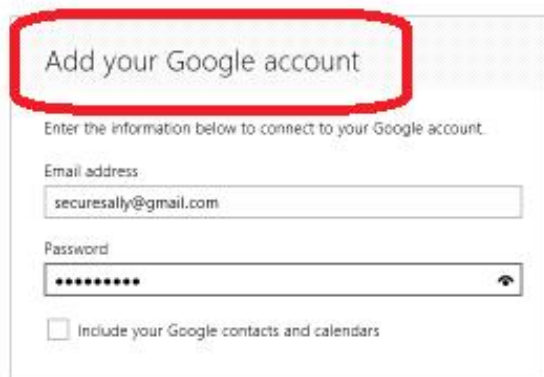
1. 구글에서 앱비밀번호를 생성시킨뒤 기억하고선
2. 메일에 가면 앱 비밀번호를 지정하내고 물어 예스
3. 파이참의 메일 비밀번호에서 생성된 비밀번호를 넣고 작동시키면 성공

생성된 앱 비밀번호

Windows 컴퓨터용 앱 비밀번호

fcww kgkg latj qwhx

메일에서 저장하겠다고 확인하면 끝



사용 방법

1. '메일' 앱을 엽니다.
2. '설정' 메뉴를 엽니다.
3. '계정'을 선택한 뒤 내 Google 계정을 선택합니다.
4. 비밀번호를 위에 표시된 16자리 비밀번호로 교체합니다.

일반적인 비밀번호와 마찬가지로 이 앱 비밀번호는 Google 계정에 대한 완전한 액세스 권한을 부여합니다. 비밀번호를 기억하지 않아도 되므로 적어 놓거나 다른 사용자와 공유하지 마세요.

[자세히 알아보기](#)

ModuleNotFoundError: No module named 'requests'



피티아 · 2021. 8. 5. 22:35

URL 복사 이웃추가

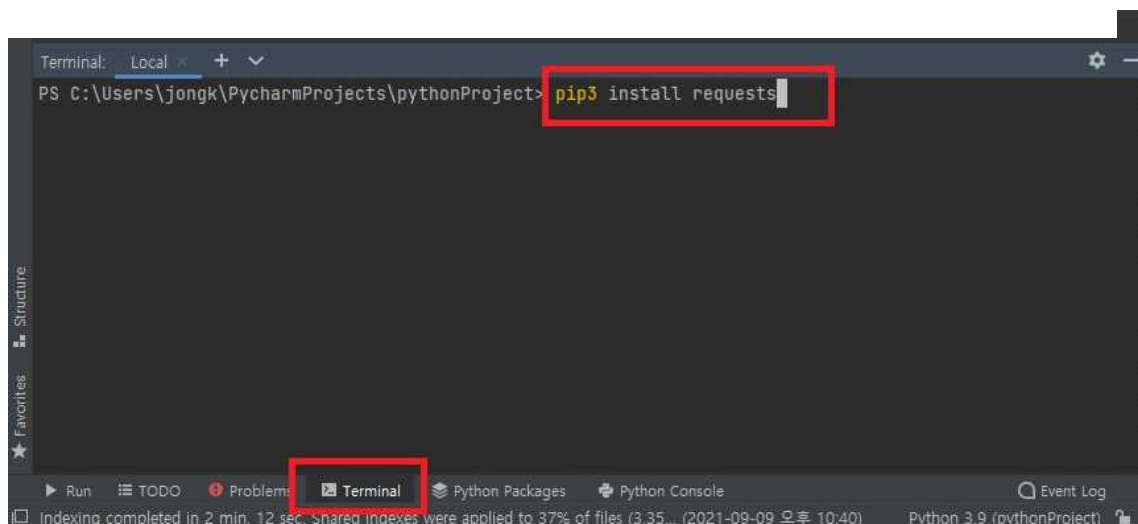
ModuleNotFoundError: No module named 'requests'

위 에러의 경우 해당 모듈이 설치되지 않은거고 본 에러문구에서는 그 해당모듈이 **requests** 인것이다
이건 터미널창에서 바로 설치하면 된다. 모듈명이 **requests** 가 아니면 해당 부분을 원하는 모듈명으로 바꿔주면 된다

`pip3 install requests`

터미널창에 저렇게만 쳐주면 된다. 다른 모듈이 필요한분은 에러메세지 ' ' 안의 모듈명을 **requests** 대신 써서 설치하자

파이참의 경우라면



화면아래부분의 **terminal** 을 눌러서 해당 명령어를 입력하면된다

[출처] [python] ModuleNotFoundError: No module named 'requests'|작성자 피티아

Python으로 Gmail발송시 에러(smtplib.SMTPAuthenticationError: (535, '5.7.8 Username and Password not accepted. Learn more at\n5.7.8)) 해결방법
by 지오준 2021. 8. 15.

기존에 정상적으로 발송되던 Gmail의 메일이 아래와 같은 에러메세지가 호출되면서 메일이 발송되지 않았을때의 대응방법입니다.

smtplib.SMTPAuthenticationError: (535, '5.7.8 Username and Password not accepted. Learn more at\n5.7.8)

1.원인 : 기존의 Gmail비밀번호가 변경됐을 경우에는 자동적으로 앱비밀번호는 초기화되므로 재설정이 필요합니다.

2.해결방법 : 앱비밀번호를 재설정하고 Python소스코드의 기존 앱비밀번호를 재설정한 비밀번호로 변경후 빌드해주시면 해결됩니다.

출처: <https://itinformation.tistory.com/entry/Python으로-Gmail발송시-에러-smtplibSMTPAuthenticationError-535-'578-Username-and-Password-not-accepted-Learn-more-atn578-해결방법> [IT정보공유]
=====

smtplib.SMTPAuthenticationError: (535, b'5.7.8 Username and Password not accepted. Learn more



천동이 • 2019. 11. 12. 18:10

URL 복사 이웃추가

Traceback (most recent call last):

```
File ".\sendemail.py", line 36, in <module>
    s.login( sendEmail , password )
File "C:\Program
Files\WindowsApps\PythonSoftwareFoundation.Python.3.7_3.7.1520.0_x64__qbz5n2k
fra8p0\lib\smtplib.py", line 730, in login
    raise last_exception
File "C:\Program
Files\WindowsApps\PythonSoftwareFoundation.Python.3.7_3.7.1520.0_x64__qbz5n2k
fra8p0\lib\smtplib.py", line 721, in login
    initial_response_ok=initial_response_ok)
File "C:\Program
Files\WindowsApps\PythonSoftwareFoundation.Python.3.7_3.7.1520.0_x64__qbz5n2k
fra8p0\lib\smtplib.py", line 642, in auth
    raise SMTPAuthenticationError(code, resp)
smtplib.SMTPAuthenticationError: (535, b'5.7.8 Username and Password not
accepted. Learn more at\n5.7.8
https://support.google.com/mail/?p=BadCredentials vpgr.37 - gsmtp')
```

```
s.login( sendEmail , password )
File "C:\Program Files\WindowsApps\PythonSoftwareFoundation.Python.3.7_3.7.1520.0_x64__qbz5n2kfra8p0\lib\smtplib.py", line 730, in login
    raise last_exception
File "C:\Program Files\WindowsApps\PythonSoftwareFoundation.Python.3.7_3.7.1520.0_x64__qbz5n2kfra8p0\lib\smtplib.py", line 721, in login
    initial_response_ok=initial_response_ok)
File "C:\Program Files\WindowsApps\PythonSoftwareFoundation.Python.3.7_3.7.1520.0_x64__qbz5n2kfra8p0\lib\smtplib.py", line 642, in auth
    raise SMTPAuthenticationError(code, resp)
smtplib.SMTPAuthenticationError: (535, b'5.7.8 Username and Password not accepted. Learn more at\n5.7.8 https://support.google.com/mail/?p=BadCredentials
') - gsmtp)
```

파이썬에서 구글 gmail 을 이용해 메일을 전송하려는 데 위와 같은 에러가
발생하였다.

해당 문제의 원인은 구글이 보안 수준을 높이면서 발생하는 것 같다.

해당 부분을 해결하려면 아래의 링크로 접속해

"보안 수준이 낮은 앱 허용"을 "사용안함" 에서 "사용함"으로 변경해주면 된다.

<https://myaccount.google.com/lesssecureapps>



로그인 - Google 계정

하나의 계정으로 모든 Google 서비스를 Google 계정으로 로그인 내 계정 찾기 계정 만들기
Google 계정 하나면 OK

myaccount.google.com

← 보안 수준이 낮은 앱의 액세스

일부 앱이나 기기에서 보안 수준이 낮은 로그인 기술을 사용하여 계정 보안이 더 취약해질 수 있습니다. Google은 이러한 앱에 액세스를 허용하지 않을 것을 권장하지만, 위험을 감수하고 사용하려면 액세스 권한을 사용 설정할 수 있습니다. 액세스를 사용하고 있지 않으시다면 자동으로 설정을 사용 중지하도록 하겠습니다. [자세히 알아보기](#)

보안 수준이 낮은 앱 허용: 사용 안함



← 보안 수준이 낮은 앱의 액세스

일부 앱이나 기기에서 보안 수준이 낮은 로그인 기술을 사용하여 계정 보안이 더 취약해질 수 있습니다. Google은 이러한 앱에 액세스를 허용하지 않을 것을 권장하지만, 위험을 감수하고 사용하려면 액세스 권한을 사용 설정할 수 있습니다. 액세스를 사용하고 있지 않으시다면 자동으로 설정을 사용 중지하도록 하겠습니다. [자세히 알아보기](#)

보안 수준이 낮은 앱 허용: 사용



---> 도전 : 로그인을 다시 해야 보안이 낮은 앱 작업을 할 수 있음 <----

보안 수준이 낮은 앱 및 Google 계정

앱 또는 사이트가 Google의 보안 표준을 충족하지 않는 경우 Google은 해당 앱에서 사용자의 계정에 로그인하려는 시도를 차단할 수 있습니다. 보안 수준이 낮은 앱에서는 해커가 사용자의 계정에 쉽게 액세스할 수 있으므로 이러한 앱에서 로그인하지 못하게 차단하면 계정을 안전하게 보호할 수 있습니다.

계정에서 '보안 수준이 낮은 앱의 액세스'가 사용 설정된 경우

보안 수준이 낮은 앱은 계정을 취약하게 만들 수 있으므로 Google에서는 사용자가 설정하지 않은 경우 자동으로 이 설정을 사용 중지합니다.

계정에 '보안 수준이 낮은 앱의 액세스'가 사용 설정되어 있다면 즉시 사용 중지하고 보안 수준이 높은 앱으로 전환하는 것이 좋습니다.

'보안 수준이 낮은 앱의 액세스' 사용 중지

계정을 안전하게 보호하려면 이 설정을 사용 중지하고 보안 수준이 높은 앱을 사용하는 것이 좋습니다.

1. Google 계정의 보안 수준이 낮은 앱의 액세스  섹션으로 이동합니다. 로그인해야 할 수도 있습니다.
2. 보안 수준이 낮은 앱 허용을 사용 중지합니다.

계정에서 '보안 수준이 낮은 앱의 액세스'가 사용 중지된 경우

계정에서 '보안 수준이 낮은 앱의 액세스'가 사용 중지된 경우 다시 사용 설정할 수 있습니다. 대신 보안 수준이 높은 앱으로 전환하는 것이 좋습니다.

2단계 인증 사용안함으로 설정해야 보안이 낮은 앱의 액세스 설정이 가능함



← 보안 수준이 낮은 앱의 액세스

일부 앱이나 기기에서 보안 수준이 낮은 로그인 기술을 사용하여 계정 보안이 더 취약해질 수 있습니다. Google은 이러한 앱에 액세스를 허용하지 않을 것을 권장하지만, 위험을 감수하고 사용하려면 액세스 권한을 사용 설정할 수 있습니다. 액세스를 사용하고 있지 않으시다면 자동으로 설정을 사용 중지하도록 하겠습니다. [자세히 알아보기](#)

보안 수준이 낮은 앱 허용: 사용



파이썬 openpyxl로 다른이름으로 엑셀 파일 저장하기

2020. 3. 18. 12:25

openpyxl다른이름저장, 파이썬shutil, 파이썬다른이름으로엑셀파일저장, 파이썬다른이름으로파일저장

Excel 파일에 파이썬 크롤링으로 데이터를 저장한 후 다른 이름으로 저장을 할 필요가 생겼다.

하기 예와 같이 openpyxl로 다른 이름을 저장하는 것은 매우 간단했다.

```
from openpyxl import load_workbook
```

```
wb = load_workbook('D:/test1.xlsx')
```

```
wb.save('D:/test2.xlsx')
```

```
wb.close()
```

단지 저장할때 이름만 바꾸면 된다.

그런데 단순한 텍스트만 있는 엑셀파일에서는 크게 문제가 없지만, 여러가지 데이터가 복잡하게 삽입된 엑셀파일에서는 문제가 발생한다.

본인의 경우에는 다음과 같은 문제점이 발생했다.

당채 무슨 소린지 모르겠다.

한참 고민을 하다가... 문득 한 가지 생각이 떠올랐다.

그래서, 작은 발상의 전환을 해보았다.

먼저, 파일을 복사 하여 다른 이름으로 저장하고, 그 파일에 필요한 데이터를 openpxy를 이용해 넣는거다.

그렇게 간단한 방법이 있었다.. -_-;;

다음과 같이 코드를 입력하면 파일을 다른 이름으로 복사할 수 있다.

```
import os
```

```
import shutil
```

```
import datetime
```

```
shutil.copy('D:/test1.xlsx', 'D:/test1.xlsx')
```

역시... 쉬운 방법은 따로 있었다.