

파이썬 학습 환경 구축

파이썬 학습 환경 구축

목차

- 파이썬 설치
- IDLE 사용
- 코랩 사용
- 파이참 설치

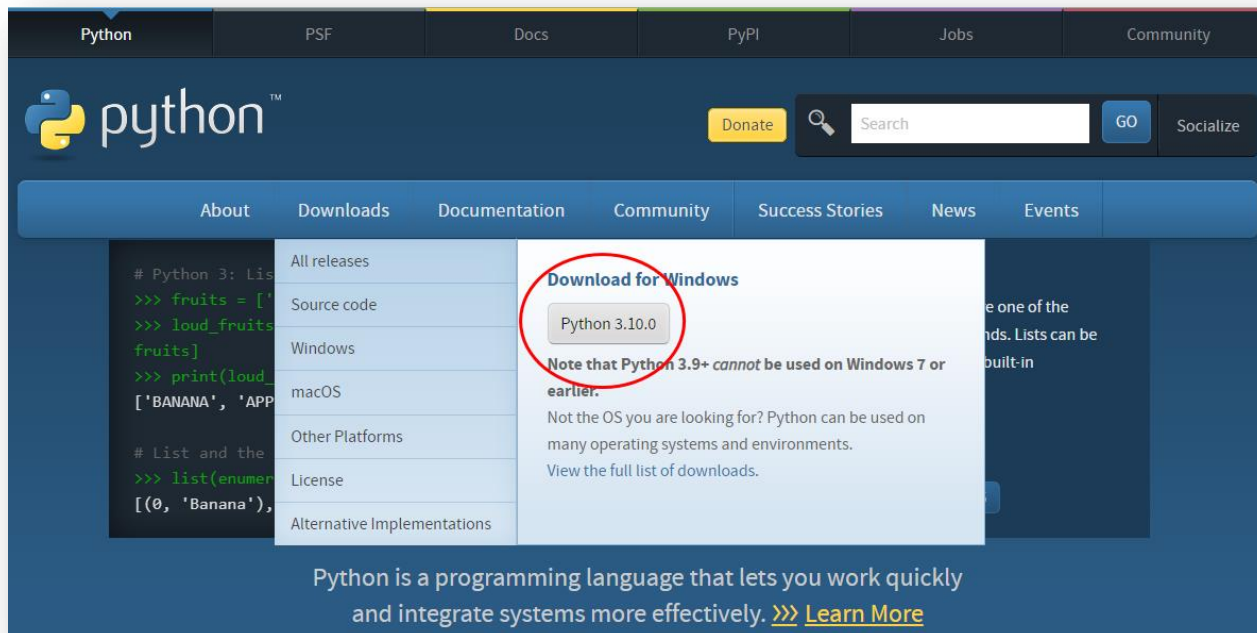
여러가지 파이썬 학습 환경

1. PC에 파이
2. 썬 설치 후 판다스 사용
3. 파이참 IDE 설치 후 판다스 사용
4. PC에 아나콘다 설치 후 판다스 사용
5. 구글 colab에서 판다스 사용

프로그램 설치

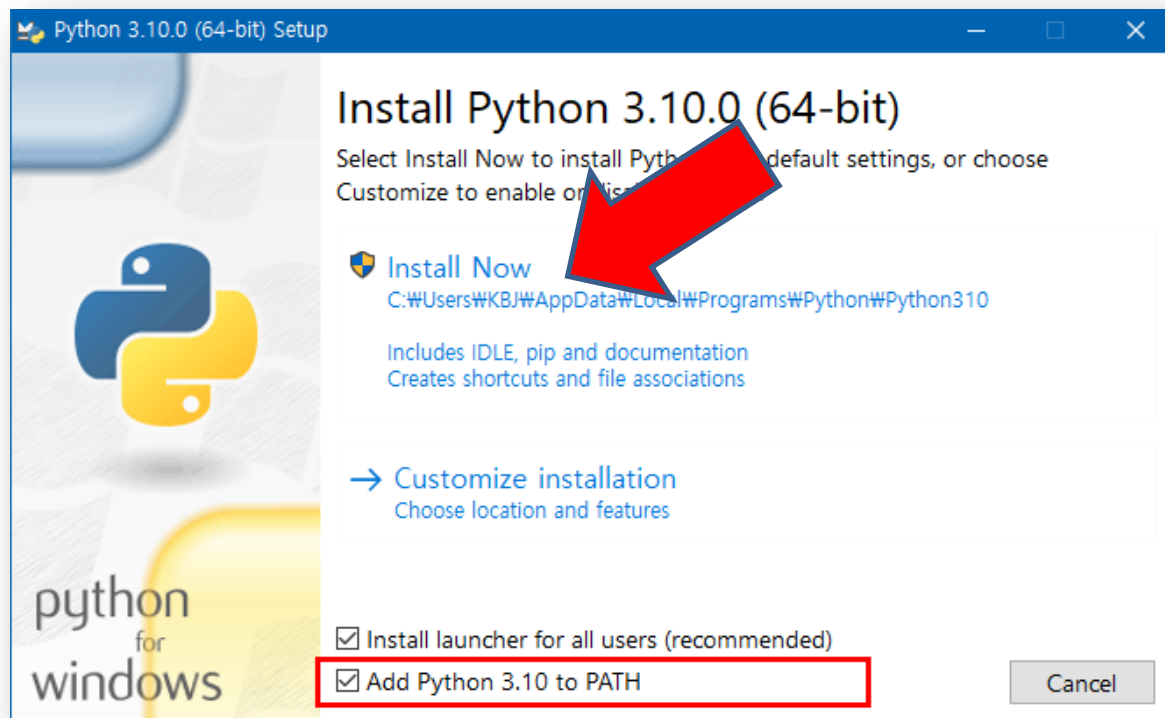
프로그램 설치

- 파이썬 공식 페이지
 - <https://www.python.org/>
 - 이전 버전은 웹사이트 아래쪽에 링크 이용.



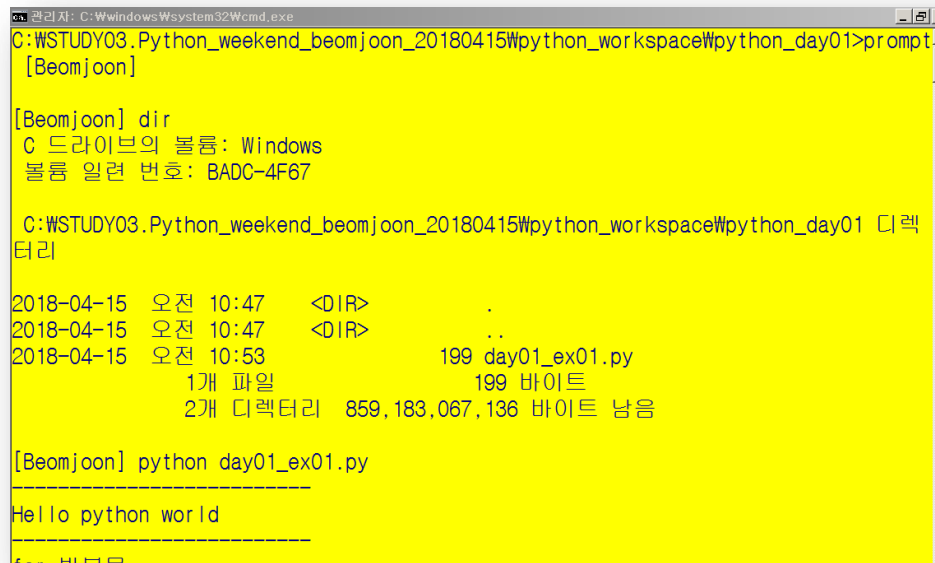
프로그램 설치

- 설치 진행 전에 Add Python 3.x to PATH 체크
- Install Now를 눌러서 설치 진행



파이썬 편집기 사용

- CMD에서 바로 Python 실행하기
- Python CLI 실행
- Python IDLE 실행
- 파이참에서 실행
- 구글 코랩(Colab)



```
C:\WINDOWS\system32\cmd.exe
C:\WSTUDY03.Python_weekend_beomjoon_20180415\python_workspace\python_day01>prompt
[Beomjoon]

[Beomjoon] dir
C 드라이브의 볼륨: Windows
볼륨 일련 번호: BADC-4F67

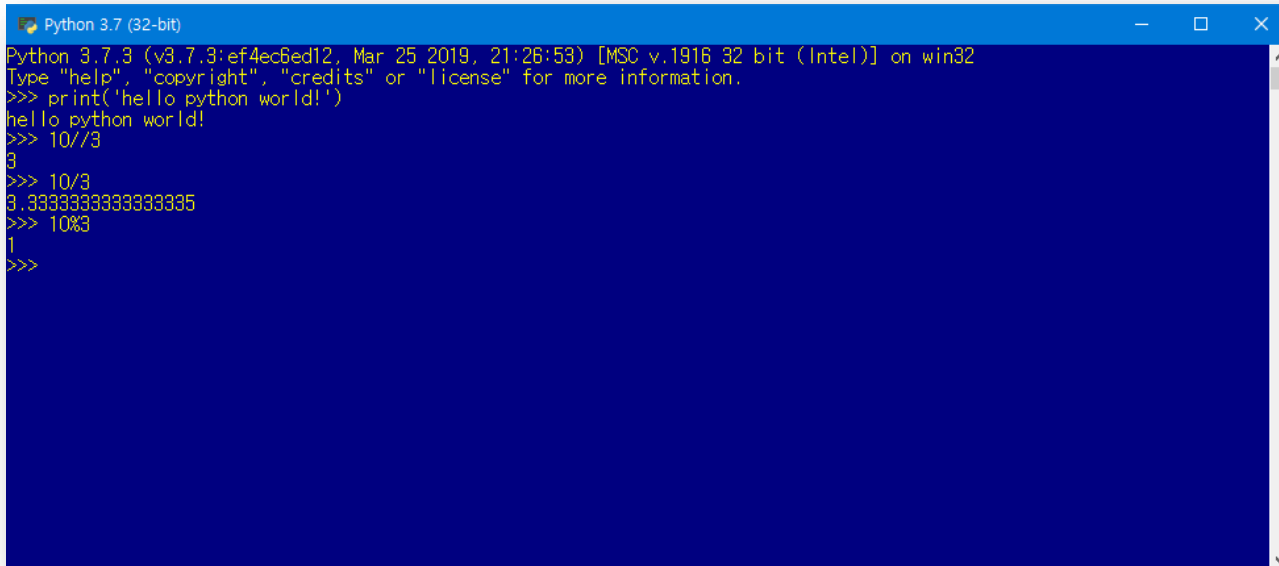
C:\WSTUDY03.Python_weekend_beomjoon_20180415\python_workspace\python_day01 디렉
터리

2018-04-15 오전 10:47    <DIR>          .
2018-04-15 오전 10:47    <DIR>          ..
2018-04-15 오전 10:53                199 day01_ex01.py
                        1개 파일              199 바이트
                        2개 디렉터리 859,183,067,136 바이트 남음

[Beomjoon] python day01_ex01.py
-----
Hello python world
-----
60 바이트
```

Python CLI 실행

- 파이썬 설치 확인
 - CMD 창에서 확인 가능 > python --version 명령어
- window > Python 3.x > Python 3.x(32bit) 실행
 - print('hello python world!')
 - 10//3 등의 명령어 테스트 가능
 - 파이썬 소스코드 파일을 바로 실행 가능

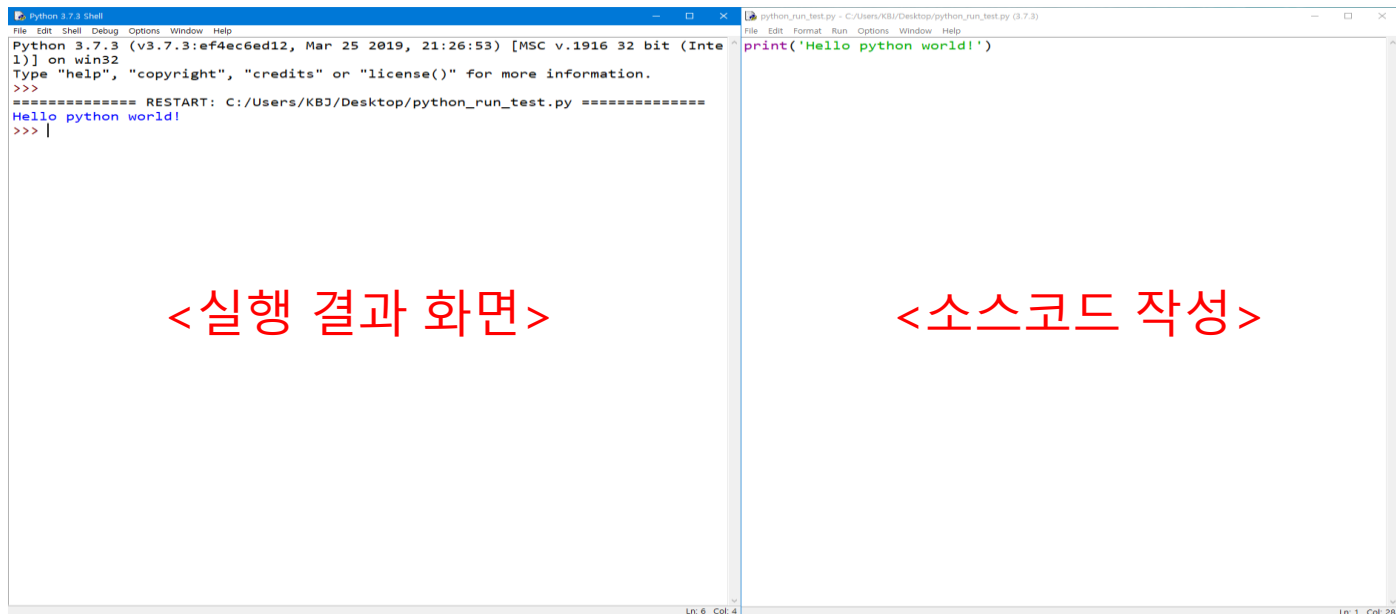


```
Python 3.7 (32-bit)
Python 3.7.3 (v3.7.3:ef4ec6d12, Mar 25, 2019, 21:26:53) [MSC v.1916 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> print('hello python world!')
hello python world!
>>> 10//3
3
>>> 10/3
3.3333333333333335
>>> 10%3
1
>>>
```


IDLE 사용

파이썬 내장 편집기 IDLE실행

- window > Python 3.x > IDLE 실행
- 명령 프롬프트에서 바로 실행 하거나
- 소프 파일을 새로 생성해서 소스코드를 작성
- 소스파일 실행 > run > Run Module F5

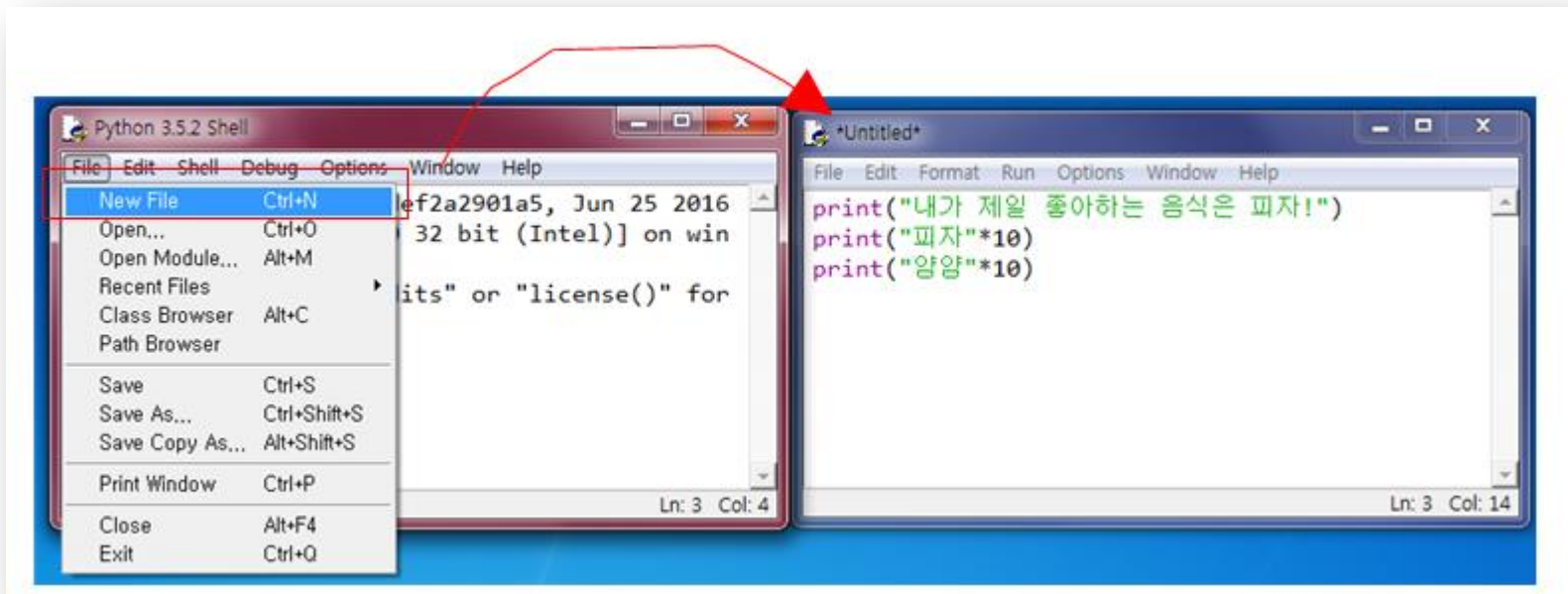


<실행 결과 화면>

<소스코드 작성>

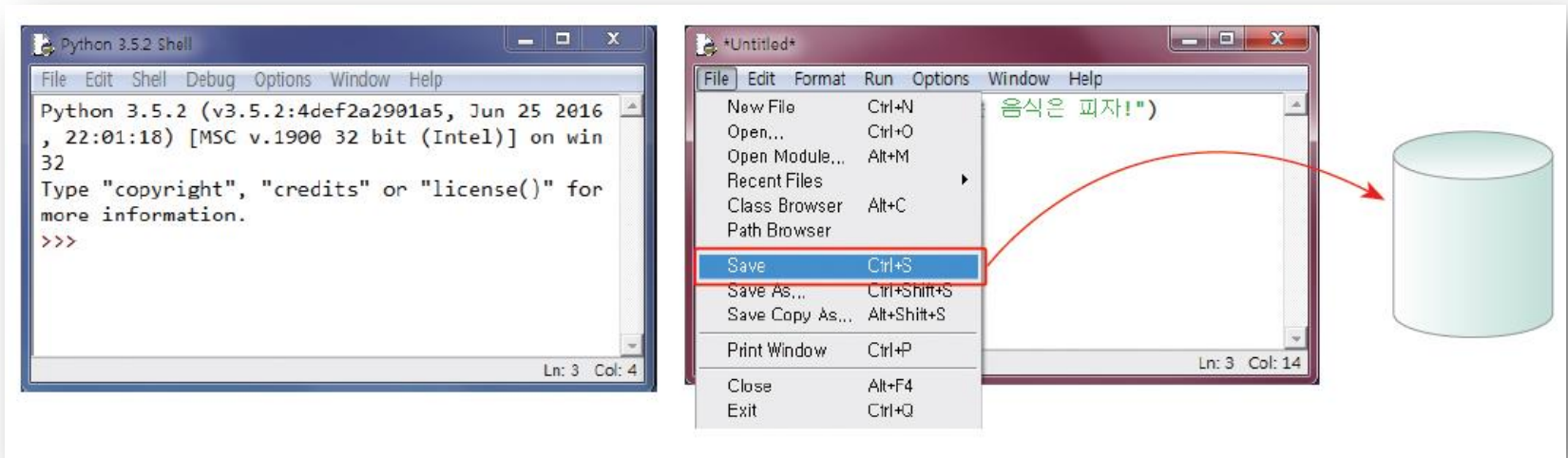
IDLE를 이용한 소스 파일 작성

- 파이썬 쉘의 메뉴 중에서 [File] -> [New File]을 선택한다.



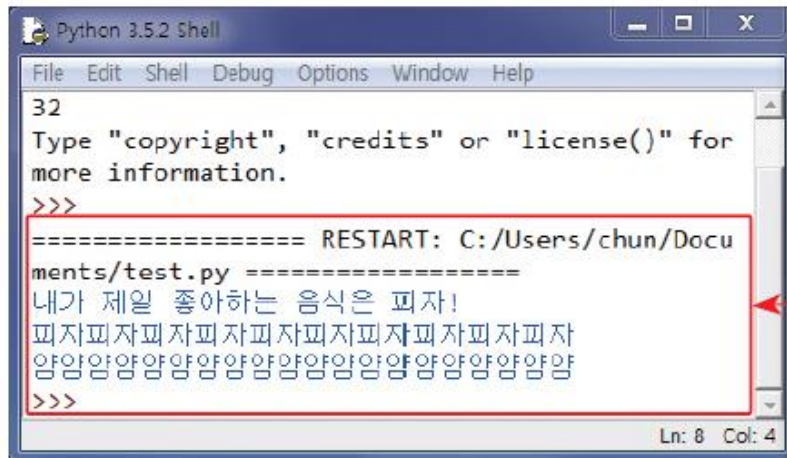
IDLE를 이용한 소스 파일 저장

- 텍스트 에디터의 [File]->[Save] 메뉴를 선택하여 코드를 파일로 저장한다.



IDLE를 이용한 소스 파일 실행

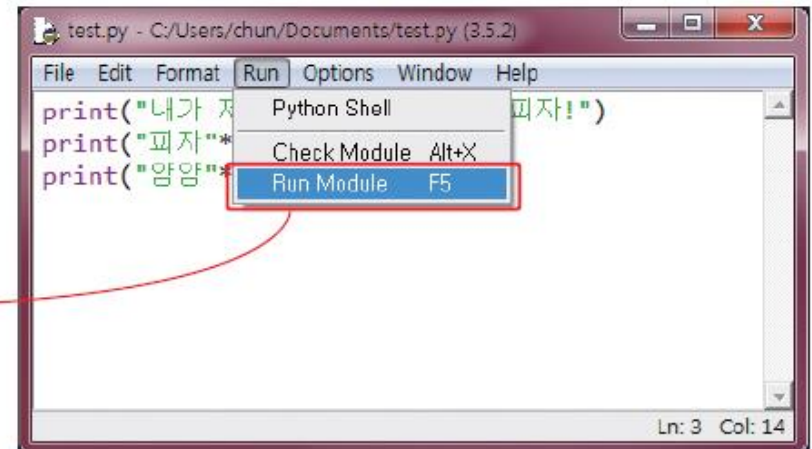
- 파이썬 셸의 메뉴 중에서 메뉴 [Run]->[Run Module]을 선택한다.



Python 3.5.2 Shell

```
File Edit Shell Debug Options Window Help
32
Type "copyright", "credits" or "license()" for
more information.
>>>
===== RESTART: C:/Users/chun/Documents/test.py =====
내가 제일 좋아하는 음식은 피자!
피자피자피자피자피자피자피자피자피자피자
암암암암암암암암암암암암암암암암암암암암암암
>>>
```

Ln: 8 Col: 4



test.py - C:/Users/chun/Documents/test.py (3.5.2)

```
File Edit Format Run Options Window Help
print("내가 제일 좋아하는 음식은 피자!")
print("피자")
print("암암")
```

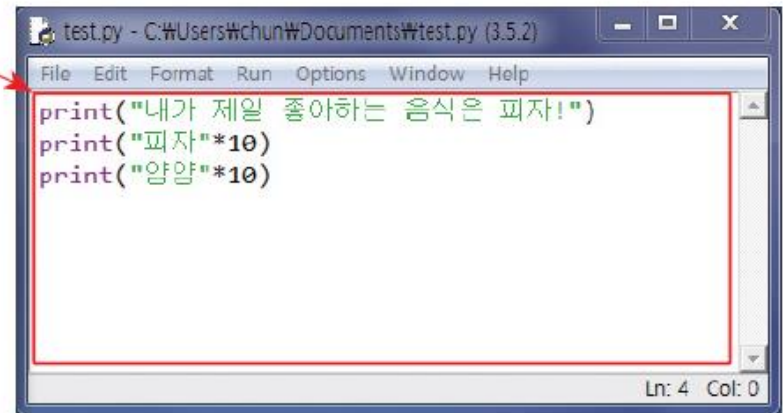
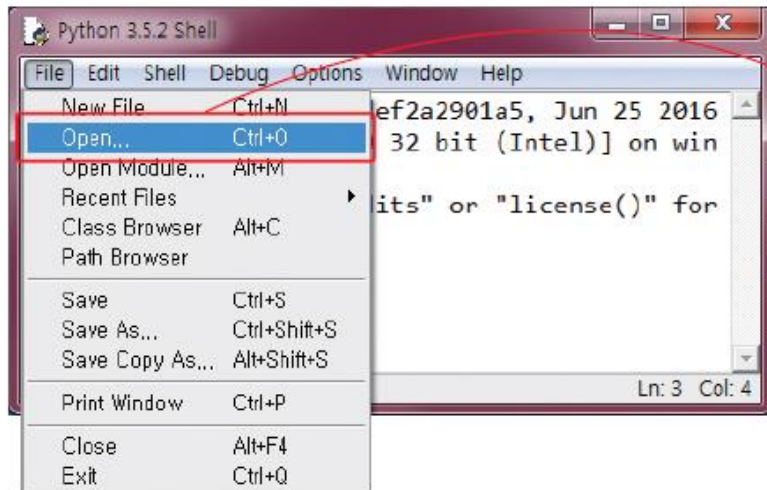
The 'Run' menu is open, showing the following options:

- Python Shell
- Check Module Alt+X
- Run Module F5

Ln: 3 Col: 14

소스 파일 다시 열기

- IDLE의 [File]->[Open] 메뉴를 선택한다.
- 우리가 저장하였던 폴더로 가서 원하는 파일을 선택한다.
- 텍스트 에디터가 나오고 우리가 입력하였던 소스가 다시 보일 것이다.



구글 코랩과 구글 드라이브

Google 코랩과 드라이브 사용

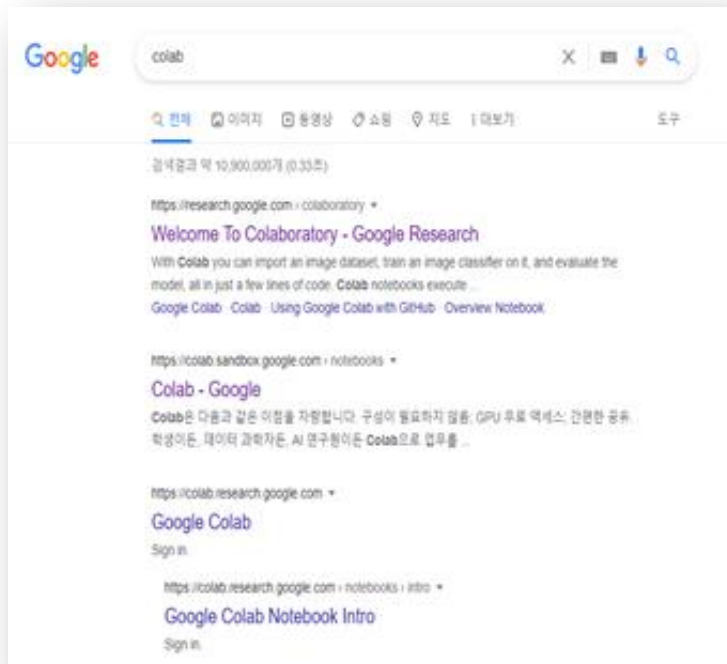
코랩 사용 자료는 블로그에서 확인 가능
<https://blog.naver.com/comstudy21/222558242553>

코랩과 구글 드라이브 연동

- 구글 코랩은 구글에서 colab을 검색 해서 바로 사용 가능합니다.
- 코랩을 사용하는 다른 방법으로 구글 드라이브에서 새로 만들기 버튼을 눌러서 코랩 파일을 생성해서 사용할 수 있습니다.
- 구글 드라이브와 구글 colab은 구글 회원 가입만 하면 무료로 사용 가능합니다.
- 이 문서는 구글 코랩에서 구글 드라이브를 마운트해서 사용하는 방법을 알려 줍니다.
- 구글 드라이브를 코랩에서 마운트하면 구글 드라이브에 파일 저장 및 파일 불러오기가 가능 해 집니다.

Google에서 검색해서 사용

구글에서 colab을 검색해서 바로 사용 가능
(<https://colab.research.google.com/>)

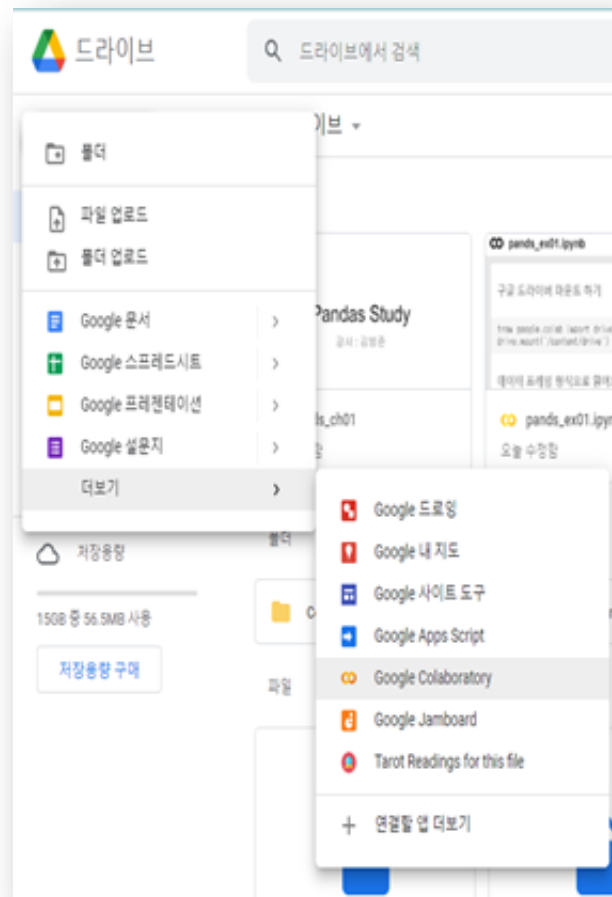


구글 드라이브에서 Colab 사용하기

1. 구글 회원가입 (이미 가입되었다면 기존 계정 사용)
2. 구글 드라이브 사용
 - a. 파일 관리
 - b. 구글 드라이브의 유용한 기능들
 - c. 구글 워크스페이스
3. 구글 코랩 사용
 - a. 새 노트
 - b. 구글 드라이브에서 확인
 - c. 구글 드라이브 mount
 - d. 구글 드라이브 파일 i/o
 - e. 구글 드라이브 데이터 파일 사용

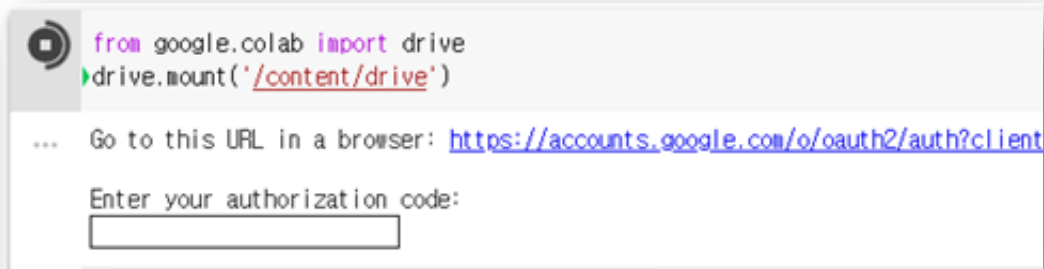
구글 드라이브에서 코랩 파일 생성

새로만들기 > 더보기 > Google Colaboratory



구글 드라이브 마운트

- 코랩에서 구글 드라이브를 마운트 하기 위해서는 인증 코드가 필요 함.



```
from google.colab import drive
drive.mount('/content/drive')
```

... Go to this URL in a browser: <https://accounts.google.com/o/oauth2/auth?client>

Enter your authorization code:

구글 드라이브 마운트

- Go to This URL in The Browser 링크를 이용해서 로그인 하면 구글 드라이브 마운트에 사용 될 인증코드를 바로 확인 가능합니다.



구글 드라이브 마운트

- 로그인 후 보이는 인증코드를 복사해서 인증하면 구글 코랩에서 구글 드라이브로 파일 입출력이 가능하게 됩니다.

```
[ ] from google.colab import drive  
    drive.mount('/content/drive')
```

```
Mounted at /content/drive
```

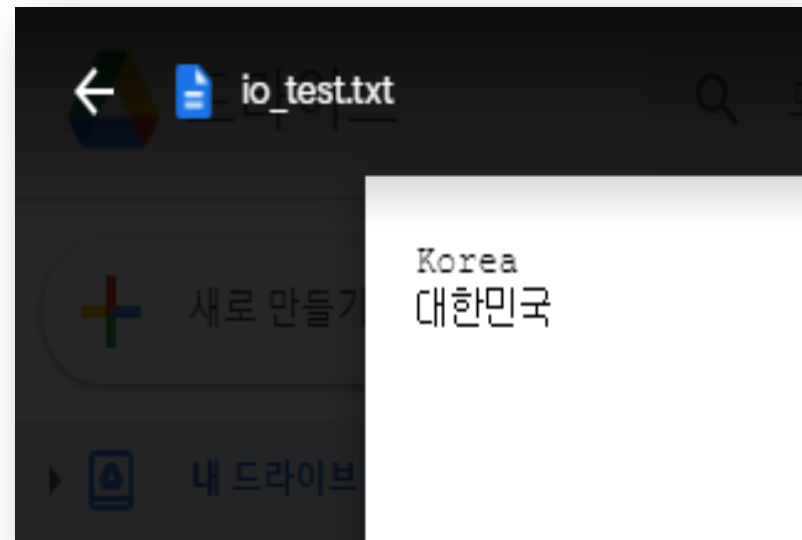
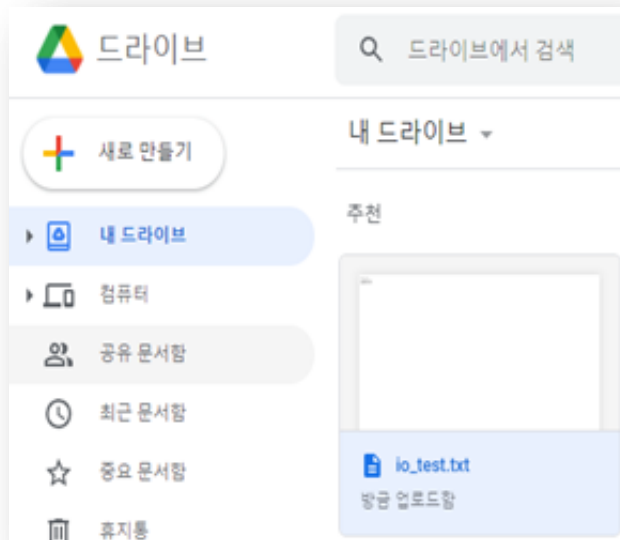
코랩에서 구글 드라이브에 저장

```
from google.colab import drive
drive.mount('/content/drive')

fp = open('/content/drive/MyDrive/io_test.txt', 'w', encoding='utf-8')

fp.write('Korea\n')
fp.write('대한민국\n')

fp.close()
```



구글 드라이브에서 데이터 읽어오기

```
filepath = '/content/drive/MyDrive/gapminder.tsv'
import pandas as pd

df=pd.read_csv(filepath, '\t')

print(df.head())
print(df.tail())
```

	country	continent	year	lifeExp	pop	gdpPercap
0	Afghanistan	Asia	1952	28.801	8425333	779.445314
1	Afghanistan	Asia	1957	30.332	9240934	820.853030
2	Afghanistan	Asia	1962	31.997	10267083	853.100710
3	Afghanistan	Asia	1967	34.020	11537966	836.197138
4	Afghanistan	Asia	1972	36.088	13079460	739.981106
	country	continent	year	lifeExp	pop	gdpPercap
1699	Zimbabwe	Africa	1987	62.351	9216418	706.157306
1700	Zimbabwe	Africa	1992	60.377	10704340	693.420786
1701	Zimbabwe	Africa	1997	46.809	11404948	792.449960
1702	Zimbabwe	Africa	2002	39.989	11926563	672.038623
1703	Zimbabwe	Africa	2007	43.487	12311143	469.709298

분석한 데이터 구글 드라이브로 저장

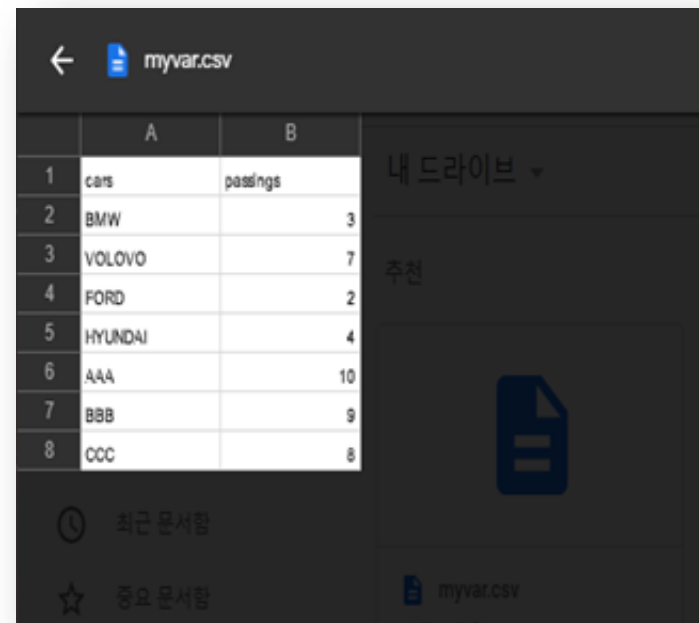
```
from google.colab import drive
drive.mount('/content/drive')

import pandas as pd

mydataset = {
    'cars': ['BMW', 'VOLOVO', 'FORD', 'HYUNDAI', 'AAA', 'BBB', 'CCC'],
    'passings': [3, 7, 2, 4, 10, 9, 8]
}

#int(data), float(data), string(data)
df = pd.DataFrame(mydataset)
print(df.head())

savepath = '/content/drive/MyDrive/myvar.csv'
df.to_csv(savepath, index=False, encoding='utf-8')
```



Series 형식으로 변환

```
a = [11, 22, 33, 44, 55]
print(type(a))
```

```
myvar = pd.Series(a)
print(type(myvar))
```

```
a = [10, 70, 20]
myvar = pd.Series(a, index=['x', 'y', 'z'])
print(myvar)
```

```
calories = {'day1': 420, 'day2': 380, 'day3': 290}
myvar = pd.Series(calories)
print(myvar)
```

```
myvar2 = pd.Series(calories, index=['day1', 'day2'])
print(myvar2)
```

DataFrame 형식으로 변환

```
mydataset = {  
    'cars': ['BMW', 'VOLVO', 'FORD', 'HYUNDAI', 'AAA', 'BBB', 'CCC'],  
    'passings': [3, 7, 2, 4, 10, 9, 8]  
}  
  
#int(data), float(data), string(data)  
myvar = pd.DataFrame(mydataset)  
print(type(myvar))  
  
#print(myvar.head())  
  
print(myvar.loc[2])
```

```
<class 'pandas.core.frame.DataFrame'>  
cars      FORD  
passings    2  
Name: 2, dtype: object
```

웹 크롤링(1)

```
from google.colab import drive
drive.mount('/content/drive')

import requests
from bs4 import BeautifulSoup

response = requests.get('https://www.naver.com/')
if response.status_code == 200 :
    soup = BeautifulSoup(response.text, 'html.parser')
    print(soup.select('ul.list_nav.'))
```

웹 크롤링(2)

```
response = requests.get('https://www.naver.com/')

if response.status_code == 200 :
    soup = BeautifulSoup(response.text, 'html.parser')
    # print(soup.select('ul.list_nav.type_fix li a'))
    data_list = soup.select('ul.list_nav.type_fix li a')
    title_lis = []
    link_lis = []
    for i, item in enumerate(data_list) :
        title_lis.append(item.text)
        link_lis.append(item['href'])

#print(title_lis, link_lis)
dataset = {
    "title": title_lis,
    "link" : link_lis
}
```

	title	link
0	메일	https://mail.naver.com/
1	카페	https://section.cafe.naver.com/
2	블로그	https://section.blog.naver.com/
3	지식IN	https://kin.naver.com/
4	쇼핑	https://shopping.naver.com/

구글 드라이브로 저장

코랩에서 크롤링 한 내용을 구글 드라이브에 파일 데이터로 저장하는 예시입니다.

```
import pandas as pd
```

```
df = pd.DataFrame(dataset)
```

```
print(df.head())
```

```
df.to_csv('/content/drive/MyDrive/naver_tile.csv', index=False, encoding='utf-8')
```

저장된 내용 확인

구글 드라이브에서 저장 된 파일을 실행 하면 내용을 확인 할 수 있습니다.



The image shows a preview of a CSV file named 'naver_tile.csv' in Google Drive. The file contains a table with two columns, A and B, listing various Naver services and their corresponding URLs. The table has 9 rows of data, with the first row serving as headers.

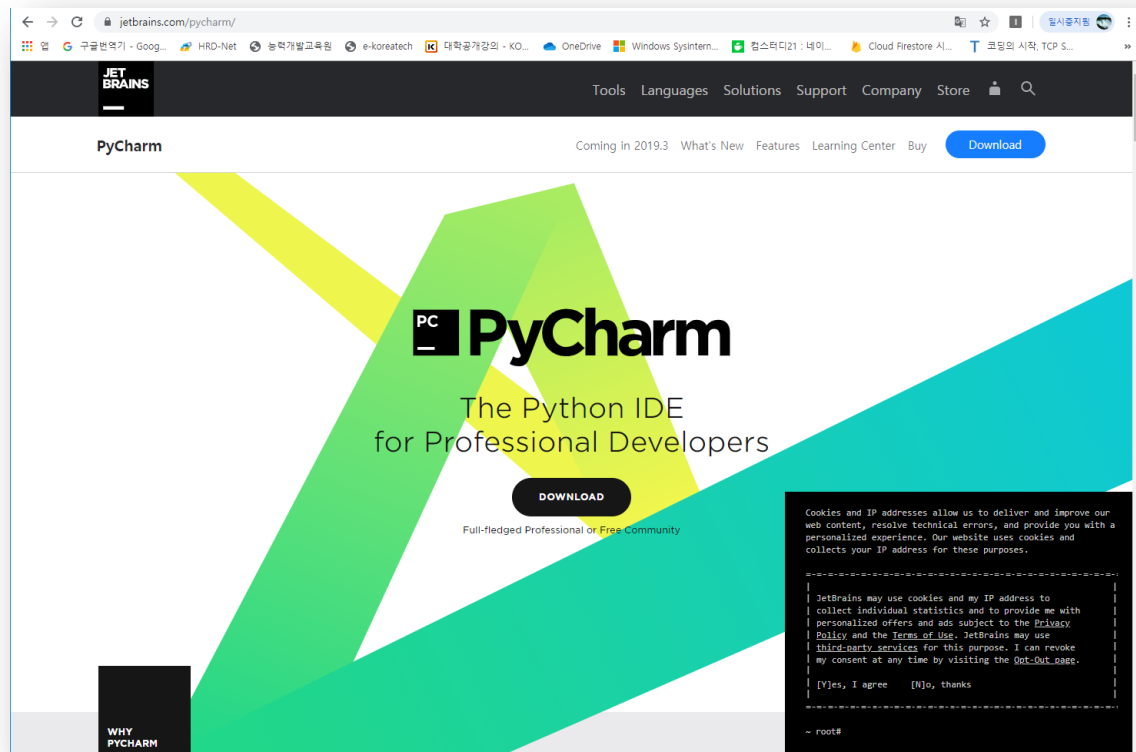
	A	B
1	title	link
2	메일	https://mail.naver.co
3	카페	https://section.cafe.n
4	블로그	https://section.blog.n
5	지식IN	https://kn.naver.com
6	쇼핑	https://shopping.nave
7	쇼핑LIVE	https://shoppinglive.r
8	Pay	https://order.pay.navi
9	TV	https://tv.naver.com/

파이참 설치

파이참 설치에 관한 내용은 부록 pdf 파일에 자세히 작성 되었습니다.

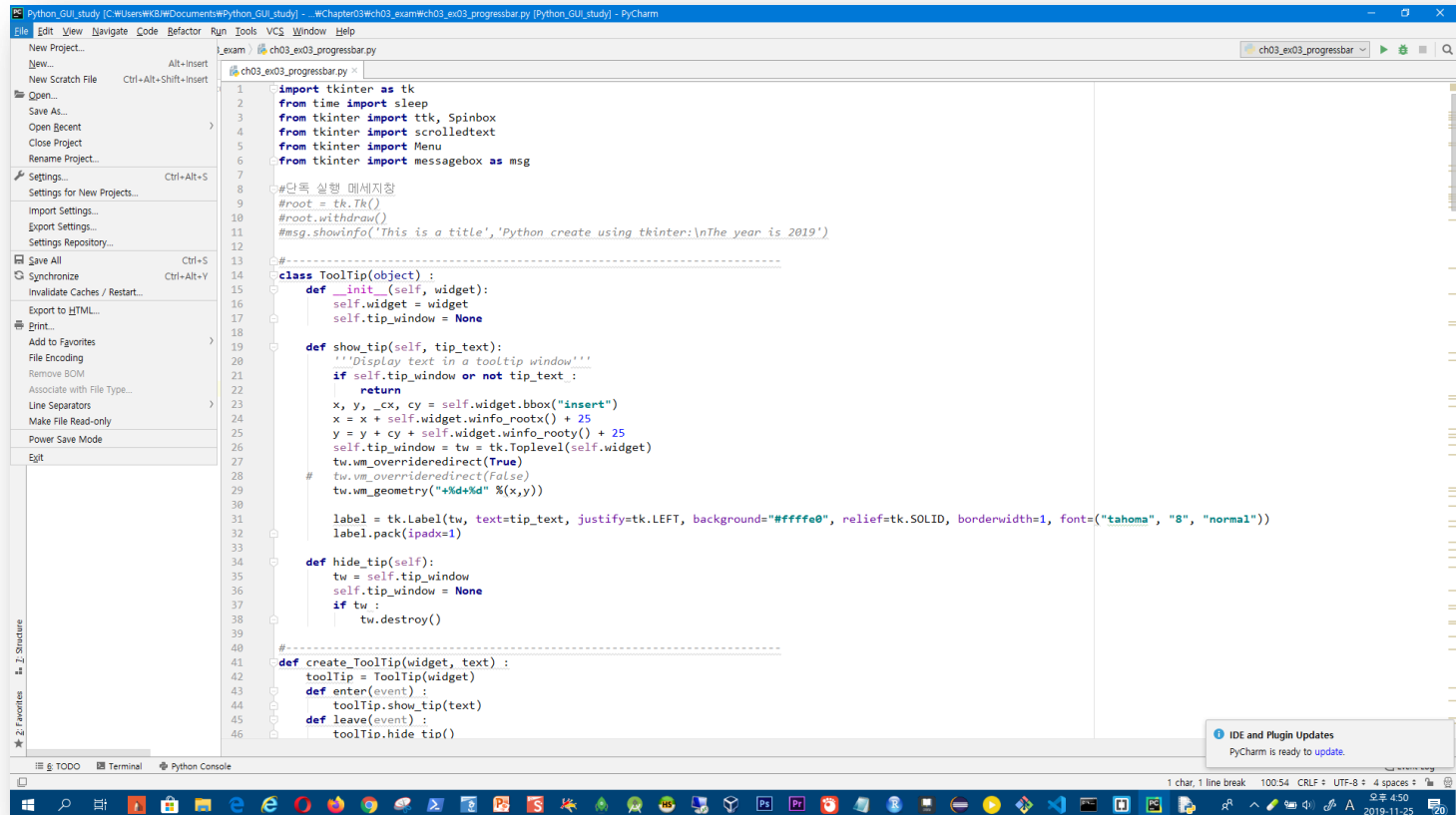
파이참 설치

- 파이참 설치파일 다운로드 및 설치
 - <https://www.jetbrains.com/pycharm/>



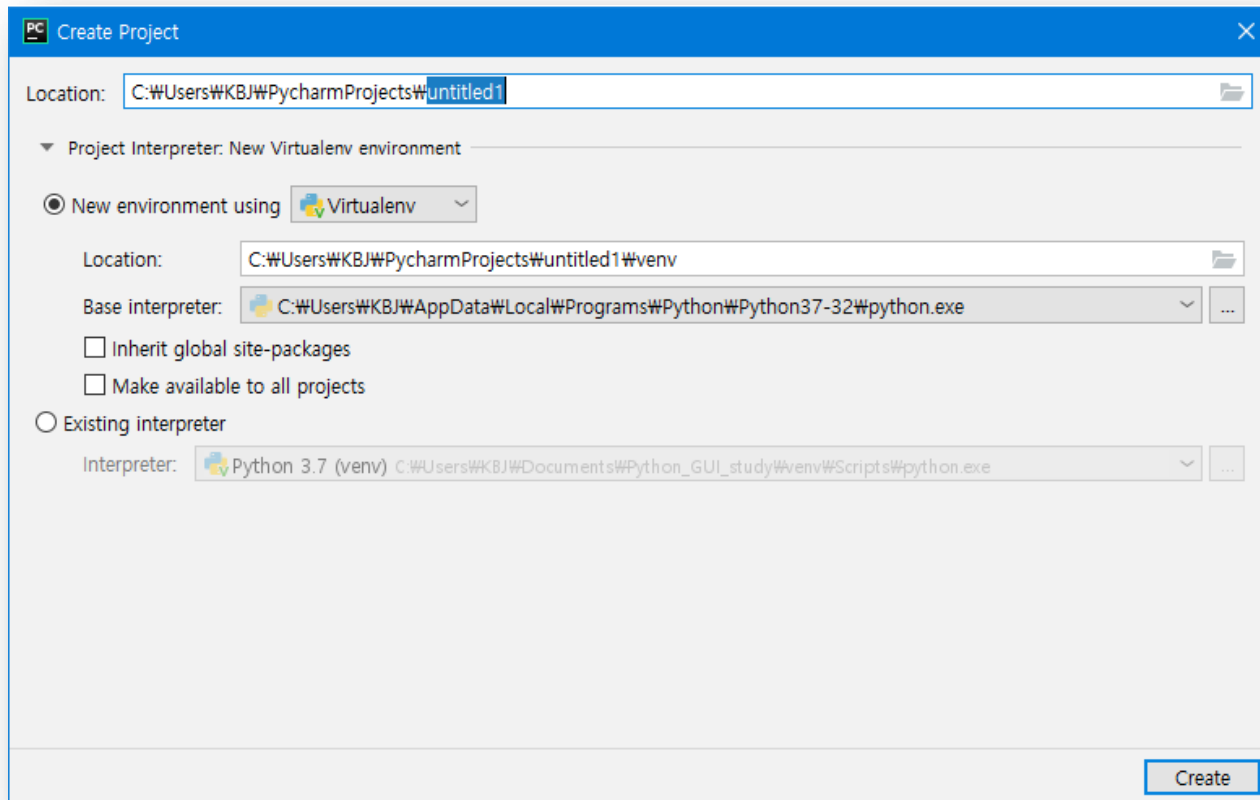
파이참 실행

- File > New Project 메뉴로 새로운 프로젝트 생성



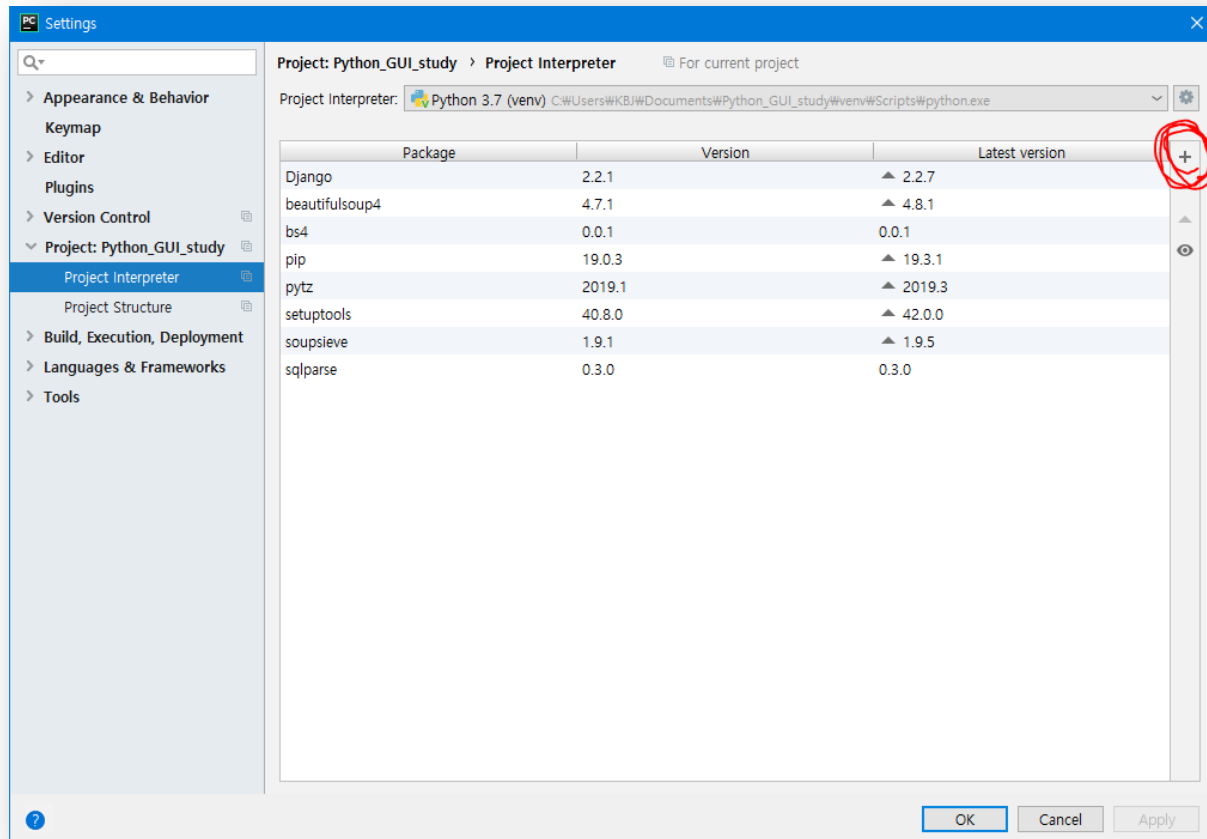
파이참 프로젝트 생성

- 새 프로젝트 생성 시 Python Interpreter 지정
- 파이썬 설치 경로 또는 프로젝트 자체 venv (가상환경)



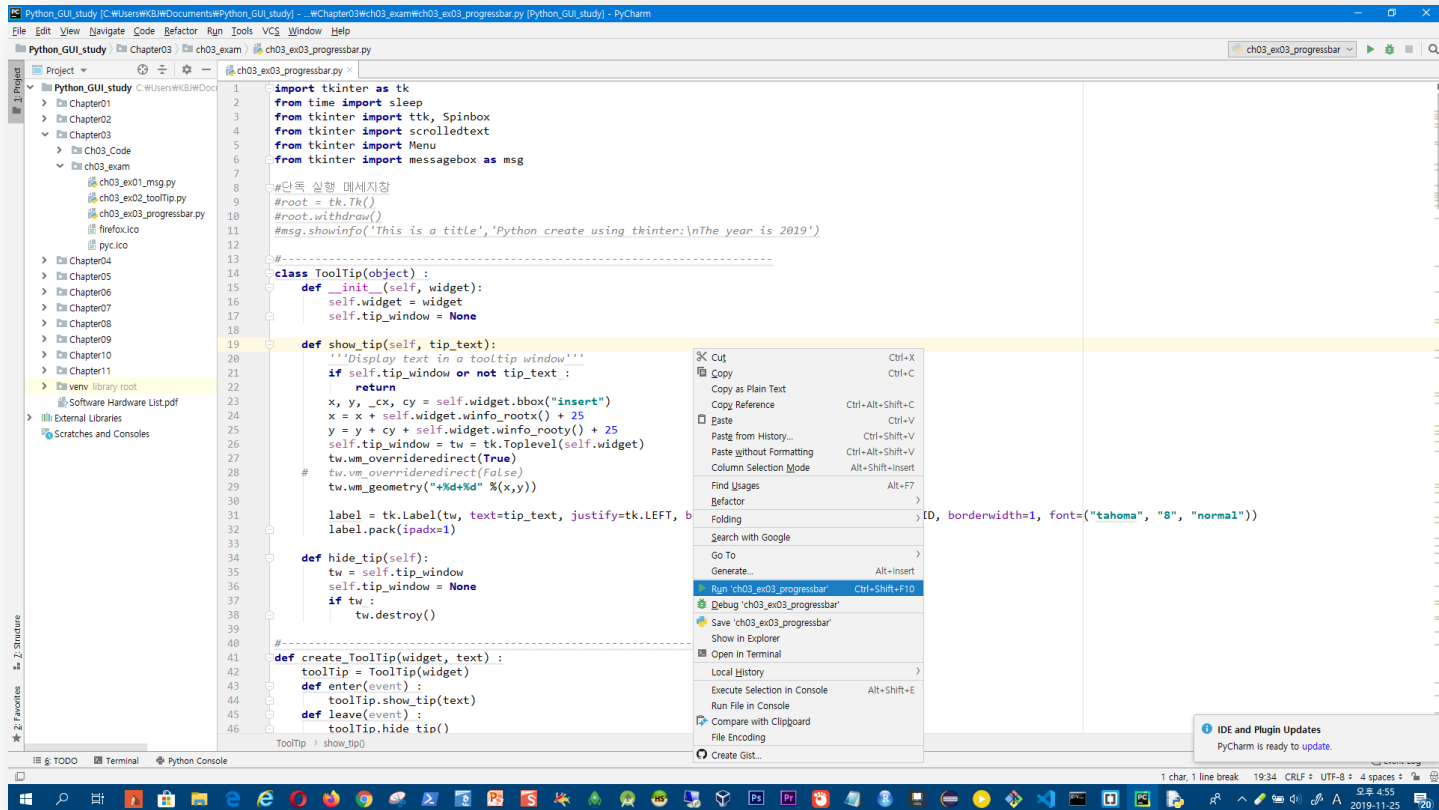
프로젝트에 외부 모듈 설치

- File > Settings ...



파일참에서 소스코드 실행

- 소스코드 화면에서 마우스 우클릭 > Run 파일명



파이참에서 소스코드 실행

- 파이참에서 즉시 소스 파일 실행이 가능하다

