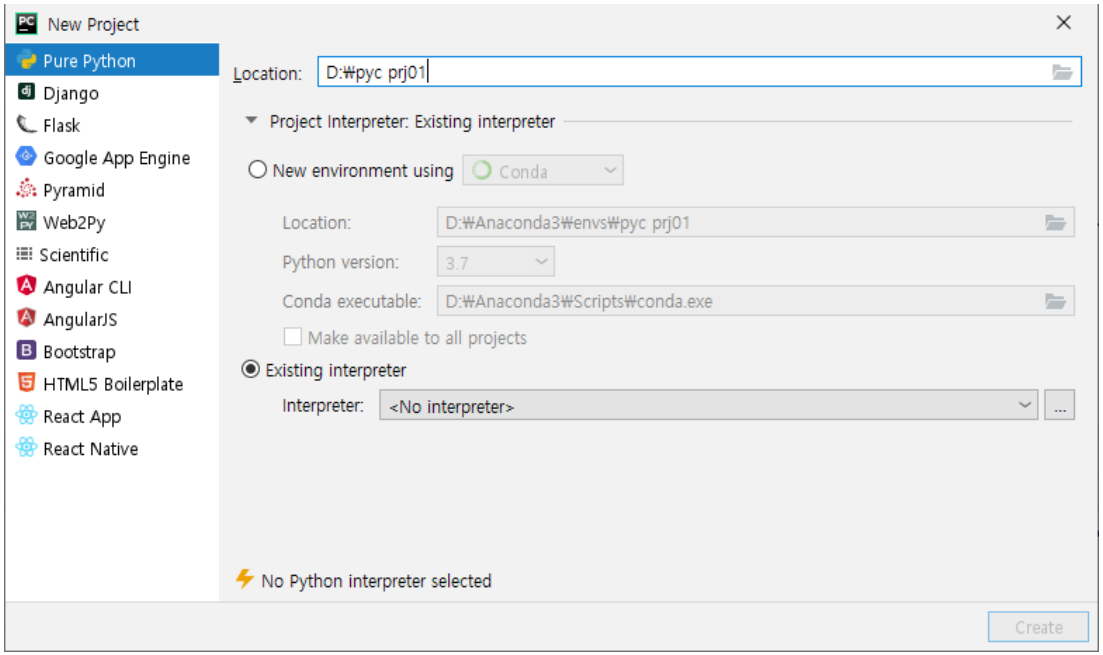
파이참 프로젝트에
이미 생성된 가상환경 설정

3개의 파이참 프로젝트 생성과 Existing Interpreter 지정

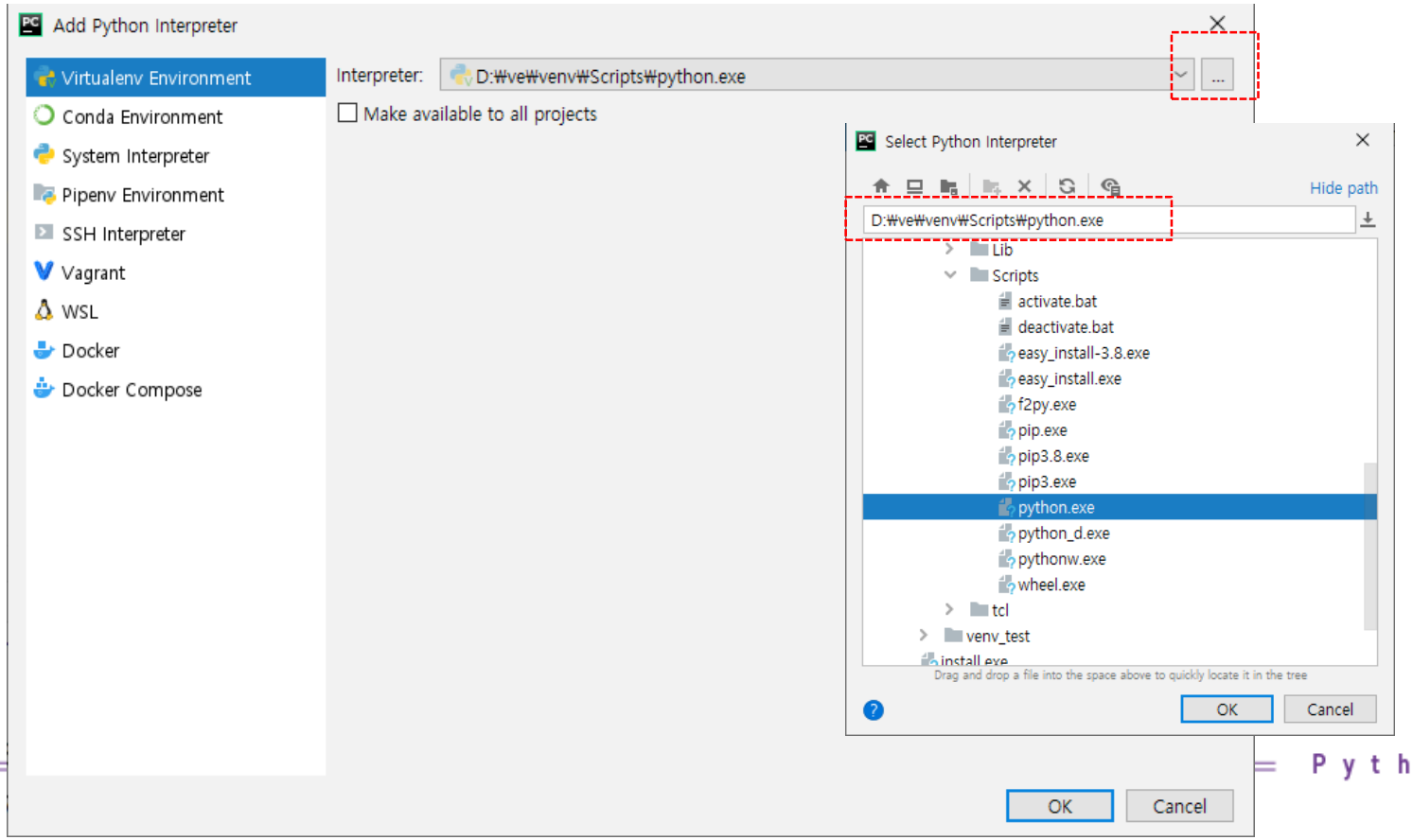
- **D:₩pyc prj01**
  - 가상환경 venv 지정
    - **Virtualenv로 만든 가상환경**
- **D:₩pyc prj02**
  - 가상환경 venv_test 지정
    - **Venv로 만든 가상환경**
- **D:₩pyc prj03**
  - 가상환경 penv 지정
    - **pipenv로 만든 가상환경**



- **Existing interpreter**
  - … 선택
    - **자신이 만든 가상환경 폴더의 scripts 하부 python.exe를 지정**

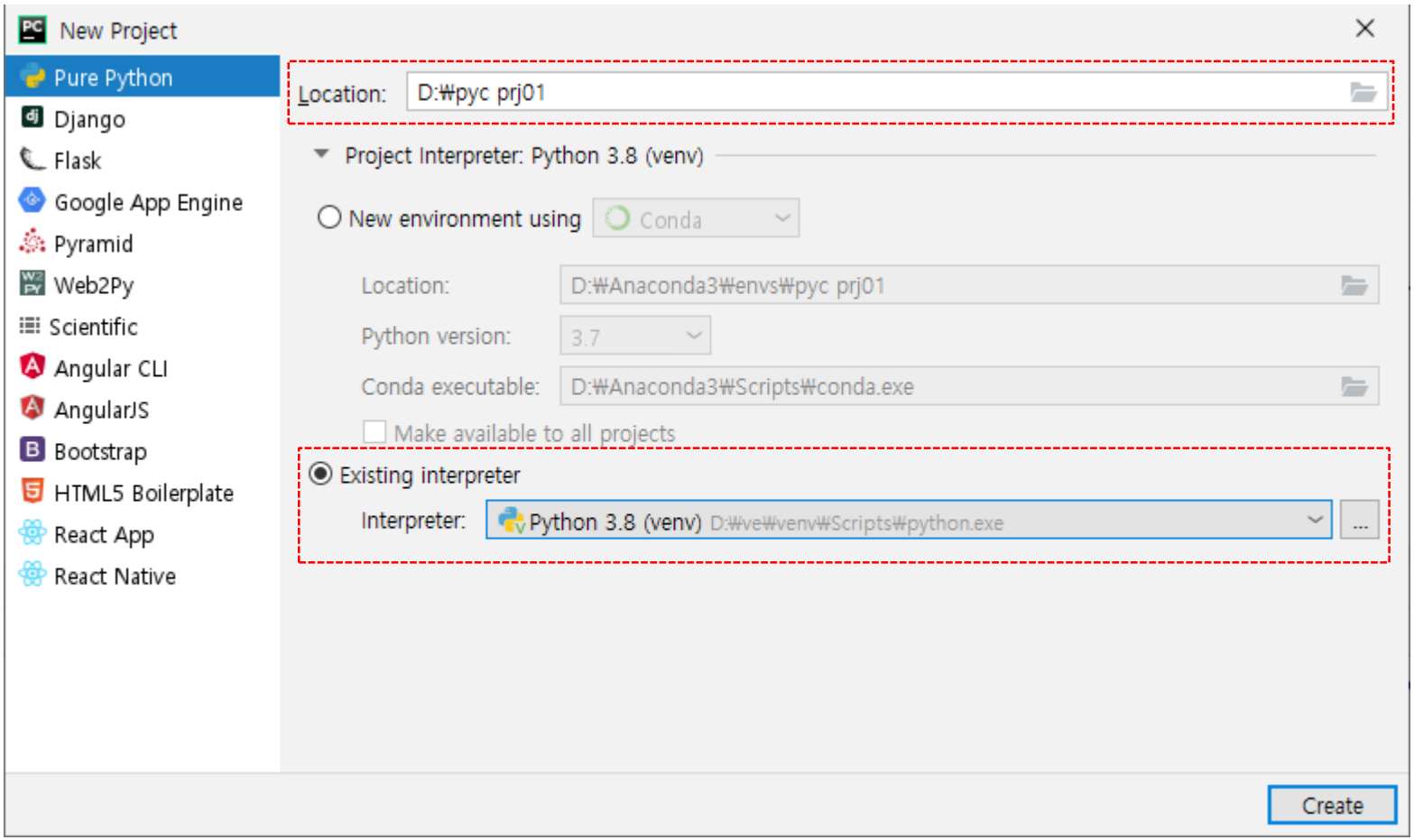
파이참 프로젝트 pyc prj01 만들기(1)

- **D:₩pyc prj01**
  - 가상환경 venv의 인터프리터 지정
    - **Virtualenv로 만든 venv 지정**
      - **...을 눌러 자신이 virtualenv로 직접 만든 가상환경의 scripts 폴더의 python.exe를 지정**



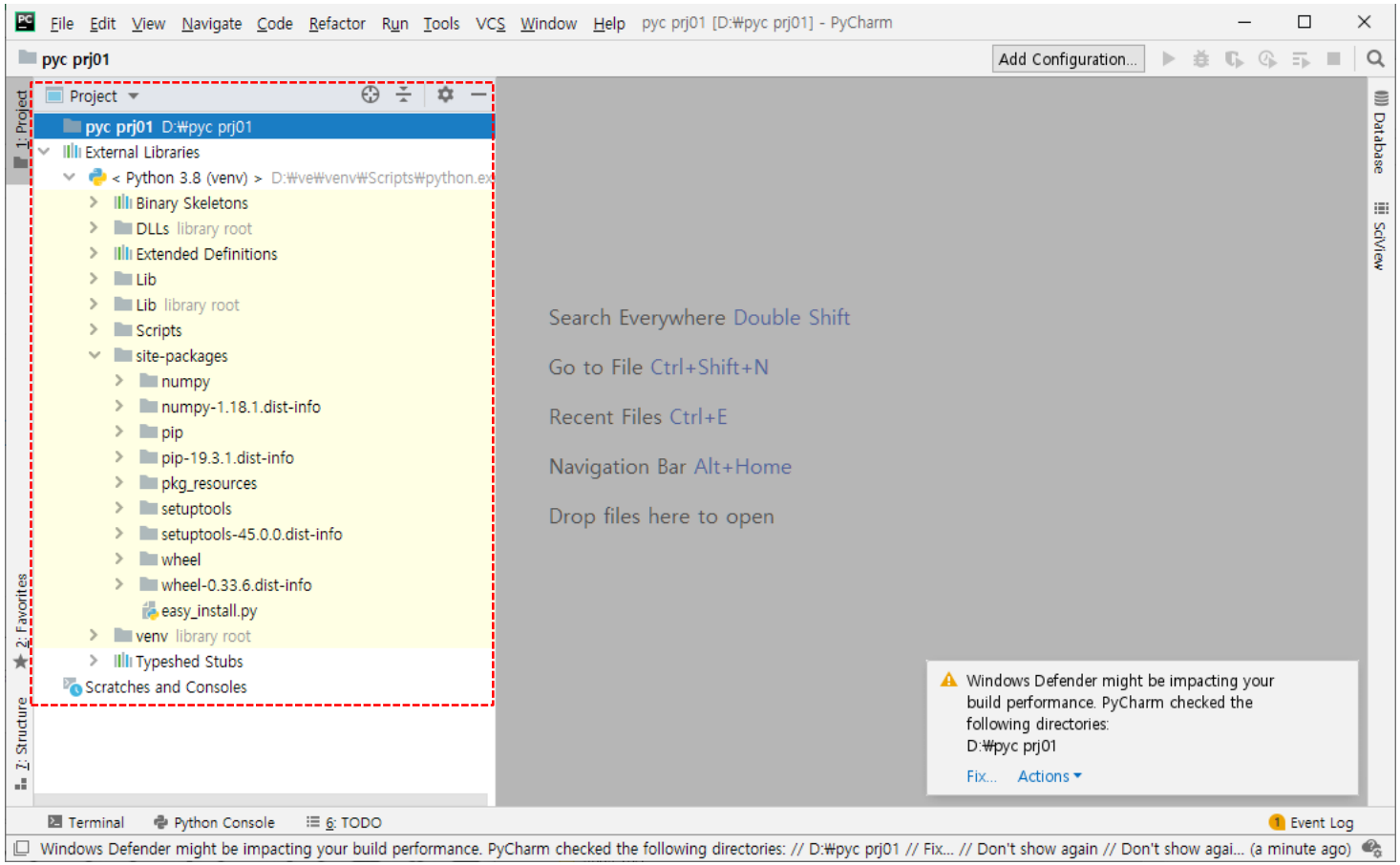
파이참 프로젝트 pyc prj01 만들기(2)

- 자신이 만든 가상환경이 지정된 New Project 대화상자

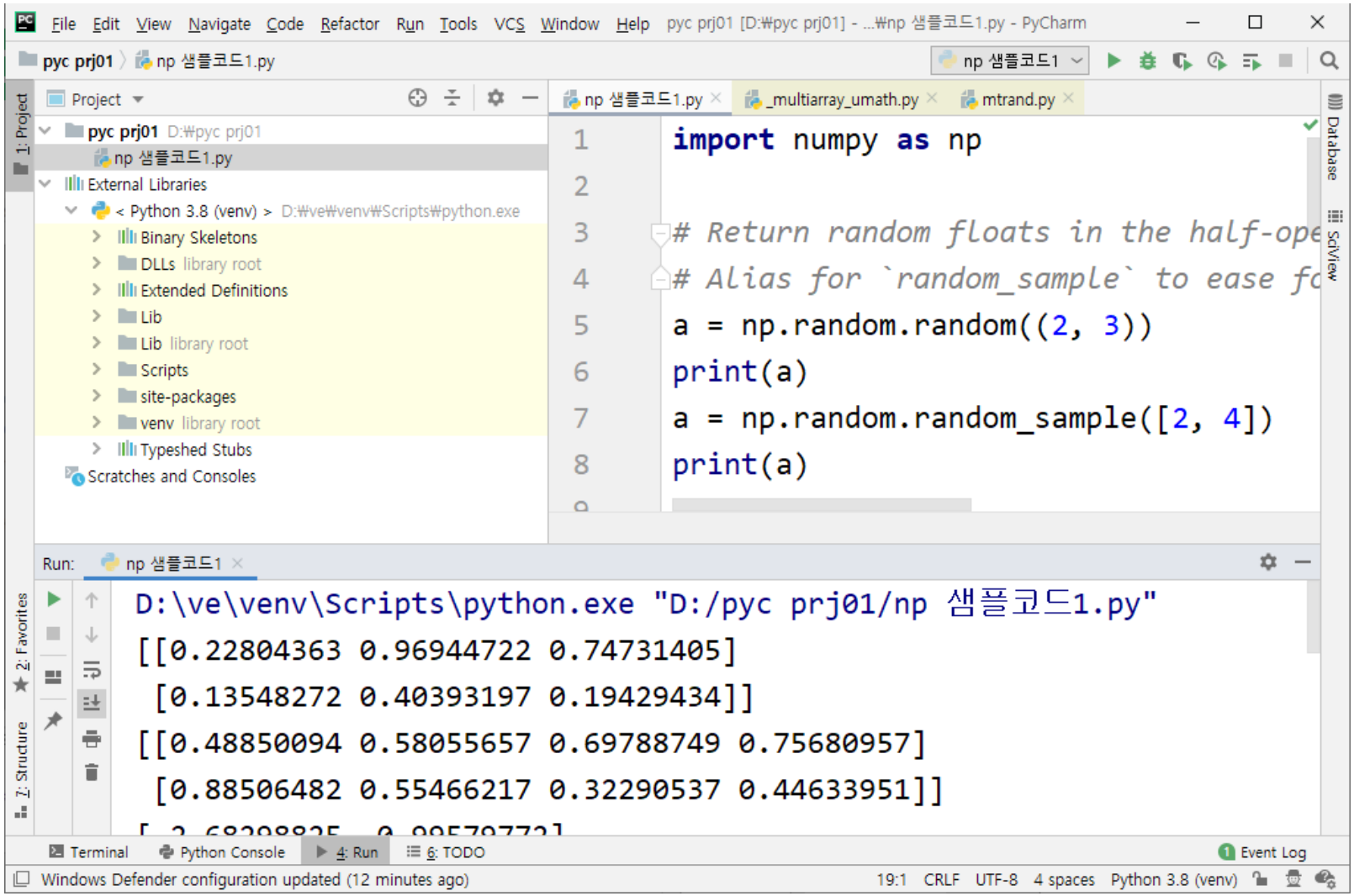


파이참 프로젝트 pyc prj01 만들기(3)

- 가상환경이 설정된 프로젝트



프로젝트 pyc prj01, 소스 np 샘플코드1.py



프로젝트 pyc prj01, 소스 이해와 결과

```python
import numpy as np

# Return random floats in the half-open interval
[0.0, 1.0).
# Alias for `random_sample` to ease forward-
porting to the new random API.
a = np.random.random((2, 3))
print(a)
a = np.random.random_sample([2, 4])
print(a)

# 정규분포의 난수 생성
b = np.random.randn(2)
print(b)
b = np.random.randn(2, 3)
print(b)
b = np.random.randn(3, 4)
print(b)

# 값이 모두 1인 텐서
c = np.ones(3)
print(c)

# 값이 모두 0인 텐서
c = np.zeros((2, 3))
print(c)
c = np.zeros([3, 4])
print(c)
```

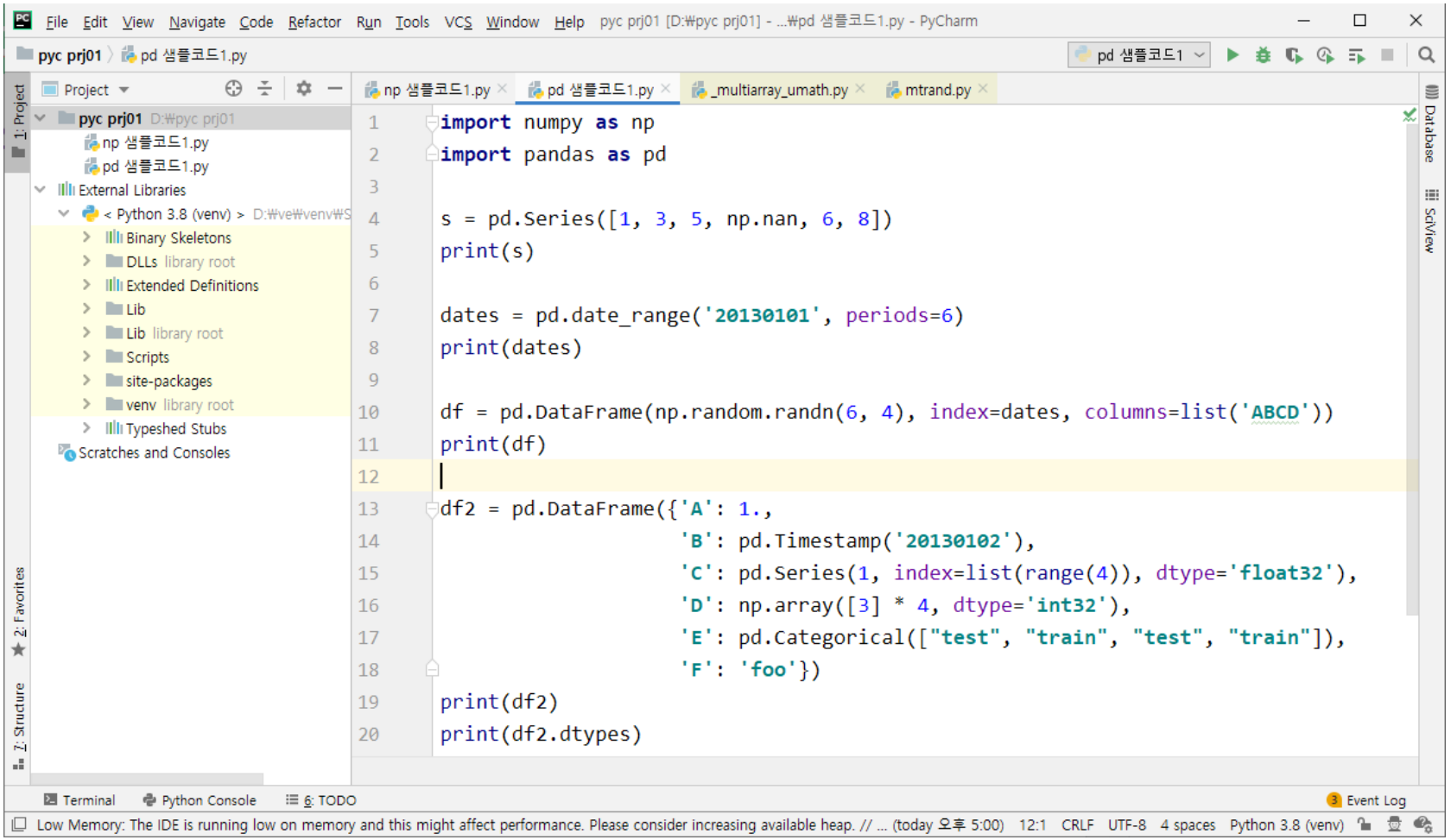
```
D:₩ve₩venv₩Scripts₩python.exe "D:/pyc prj01/np 샘플코드1.py"
[[0.22804363 0.96944722 0.74731405]
 [0.13548272 0.40393197 0.19429434]]
[[0.48850094 0.58055657 0.69788749 0.75680957]
 [0.88506482 0.55466217 0.32290537 0.44633951]]
[-2.68298825 -0.99579772]
[[-0.16099173  1.29978199  1.66469465]
 [ 0.68463138 -1.58433849 -0.06550564]]
[[ 0.02758157 -0.77454657  0.49862683  0.70299809]
 [ 1.57338867 -0.71034582  0.06683078 -1.51798292]
 [-0.20716991 -0.45346911  0.20735756 -0.76737582]]
[1. 1. 1.]
[[0. 0. 0.]
 [0. 0. 0.]]
[[0. 0. 0. 0.]
 [0. 0. 0. 0.]
 [0. 0. 0. 0.]]

Process finished with exit code 0
```

프로젝트 pyc prj01, 터미널 활용 모듈 pandas 추가

```
(venv) D:₩pyc prj01>where pip
D:₩ve₩venv₩Scripts₩pip.exe
D:₩Python38-32₩Scripts₩pip.exe
D:₩Anaconda3₩Scripts₩pip.exe
C:₩ProgramData₩Anaconda3₩Scripts₩pip.exe

(venv) D:₩pyc prj01>pip --version
pip 19.3.1 from d:₩ve₩venv₩lib₩site-packages₩pip
(python 3.8)

(venv) D:₩pyc prj01>pip list
Package    Version
---------- -------
numpy      1.18.1
pip        19.3.1
setuptools 45.0.0
wheel      0.33.6

(venv) D:₩pyc prj01>pip install --upgrade pip
Requirement already up-to-date: pip in
d:₩ve₩venv₩lib₩site-packages (19.3.1)

... pip install pandas

(venv) D:₩pyc prj01>pip list
Package         Version
--------------- -------
numpy           1.18.1
pandas          0.25.3
pip             19.3.1
python-dateutil 2.8.1
pytz            2019.3
setuptools      45.0.0
six             1.14.0
wheel           0.33.6
```

```
(venv) D:₩pyc prj01>pip install pandas
Collecting pandas
  Using cached
https://files.pythonhosted.org/packages/78/b9/a304328ea14cd172a5c
f681b634b99e24a5b4e24de83204b713b088f02d5/pandas-0.25.3-
cp38-cp38-win32.whl

Collecting python-dateutil>=2.6.1
  Using cached
https://files.pythonhosted.org/packages/d4/70/d60450c3dd48ef87586
924207ae8907090de0b306af2bce5d134d78615cb/python_dateutil-
2.8.1-py2.py3-none-any.whl

Requirement already satisfied: numpy>=1.13.3 in
d:₩ve₩venv₩lib₩site-packages (from pandas) (1.18.1)

Collecting pytz>=2017.2
  Using cached
https://files.pythonhosted.org/packages/e7/f9/f0b53f88060247251bf4
81fa6ea62cd0d25bf1b11a87888e53ce5b7c8ad2/pytz-2019.3-py2.py3-
none-any.whl

Collecting six>=1.5
  Downloading
https://files.pythonhosted.org/packages/65/eb/1f97cb97bfc2390a2769
69c6fae16075da282f5058082d4cb10c6c5c1dba/six-1.14.0-py2.py3-
none-any.whl

Installing collected packages: six, python-dateutil, pytz, pandas
Successfully installed pandas-0.25.3 python-dateutil-2.8.1 pytz-2019.3
six-1.14.0
```

프로젝트 pyc prj01, 소스 pd 샘플코드1.py



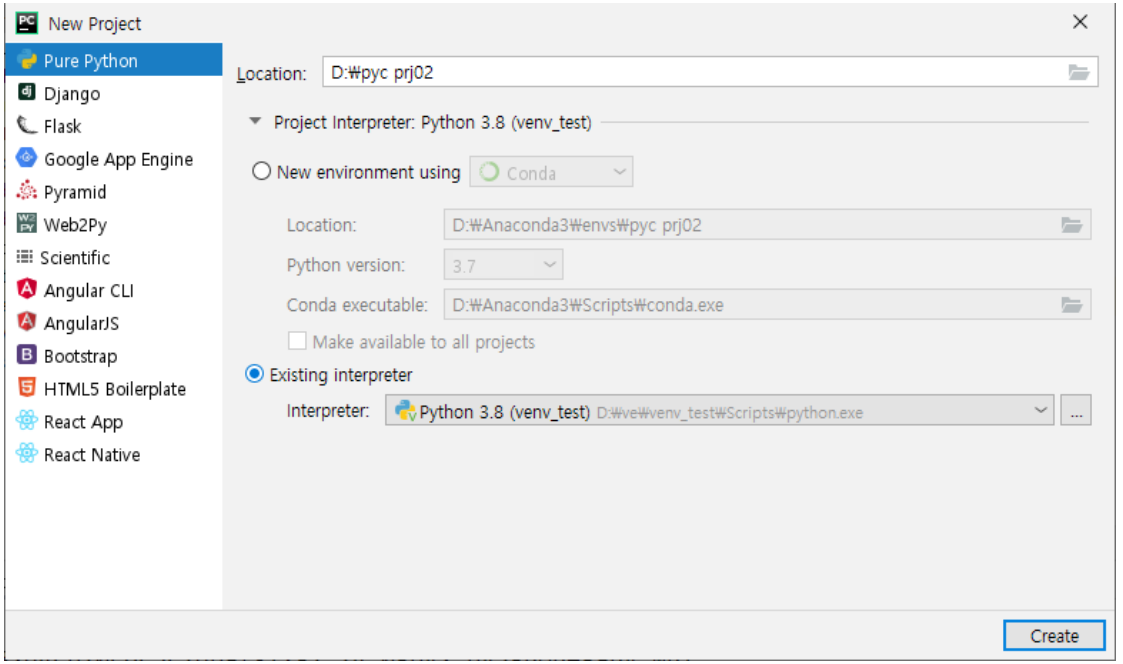
프로젝트 pyc prj01, 소스 pd 샘플코드1.py 이해

```
import numpy as np
import pandas as pd

s = pd.Series([1, 3, 5, np.nan, 6, 8])
print(s)

dates = pd.date_range('20130101', periods=6)
print(dates)

df = pd.DataFrame(np.random.randn(6, 4), index=dates,
                                         columns=list('ABCD'))
print(df)

df2 = pd.DataFrame(
         {'A': 1.,
          'B': pd.Timestamp('20130102'),
          'C': pd.Series(1, index=list(range(4)), dtype='float32'),
          'D': np.array([3] * 4, dtype='int32'),
          'E': pd.Categorical(["test", "train", "test", "train"]),
          'F': 'foo'})
print(df2)
print(df2.dtypes)
```

```
D:₩ve₩venv₩Scripts₩python.exe "D:/pyc prj01/pd 샘플코드1.py"
0    1.0
1    3.0
2    5.0
3    NaN
4    6.0
5    8.0
dtype: float64
DatetimeIndex(['2013-01-01', '2013-01-02', '2013-01-03', '2013-01-04',
               '2013-01-05', '2013-01-06'],
              dtype='datetime64[ns]', freq='D')
                   A         B         C         D
2013-01-01  0.617621  0.711937  0.555614  1.261003
2013-01-02 -1.378531 -0.325374  0.788234  1.223037
2013-01-03 -1.228281 -0.682720 -0.564663  0.107814
2013-01-04  0.552391 -0.354623 -0.488619 -0.072650
2013-01-05 -1.099271  0.108479 -2.062795  2.163172
2013-01-06 -0.034693  0.705632  0.194938  0.800431
     A          B    C  D      E    F
0  1.0 2013-01-02  1.0  3   test  foo
1  1.0 2013-01-02  1.0  3  train  foo
2  1.0 2013-01-02  1.0  3   test  foo
3  1.0 2013-01-02  1.0  3  train  foo
A           float64
B    datetime64[ns]
C           float32
D             int32
E          category
F            object
dtype: object

Process finished with exit code 0
```

3개의 파이참 프로젝트 생성과 Existing Interpreter 지정

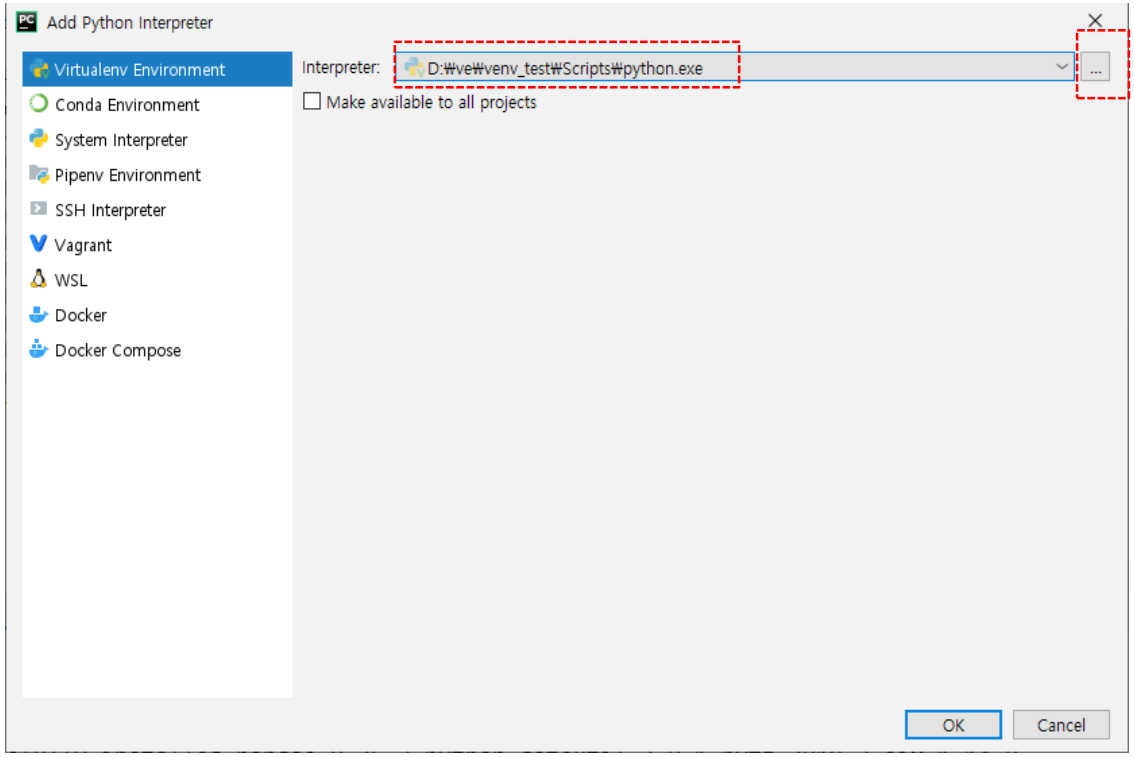
- **D:₩pyc prj01**
  - 가상환경 venv 지정
    - **Virtualenv로 만든 가상환경**
- **D:₩pyc prj02**
  - 가상환경 venv_test 지정
    - **Venv로 만든 가상환경**
- **D:₩pyc prj03**
  - 가상환경 penv 지정
    - **pipenv로 만든 가상환경**



- **Existing interpreter**
  - … 선택
    - **자신이 만든 가상환경 폴더의 scripts 하부 python.exe를 지정**

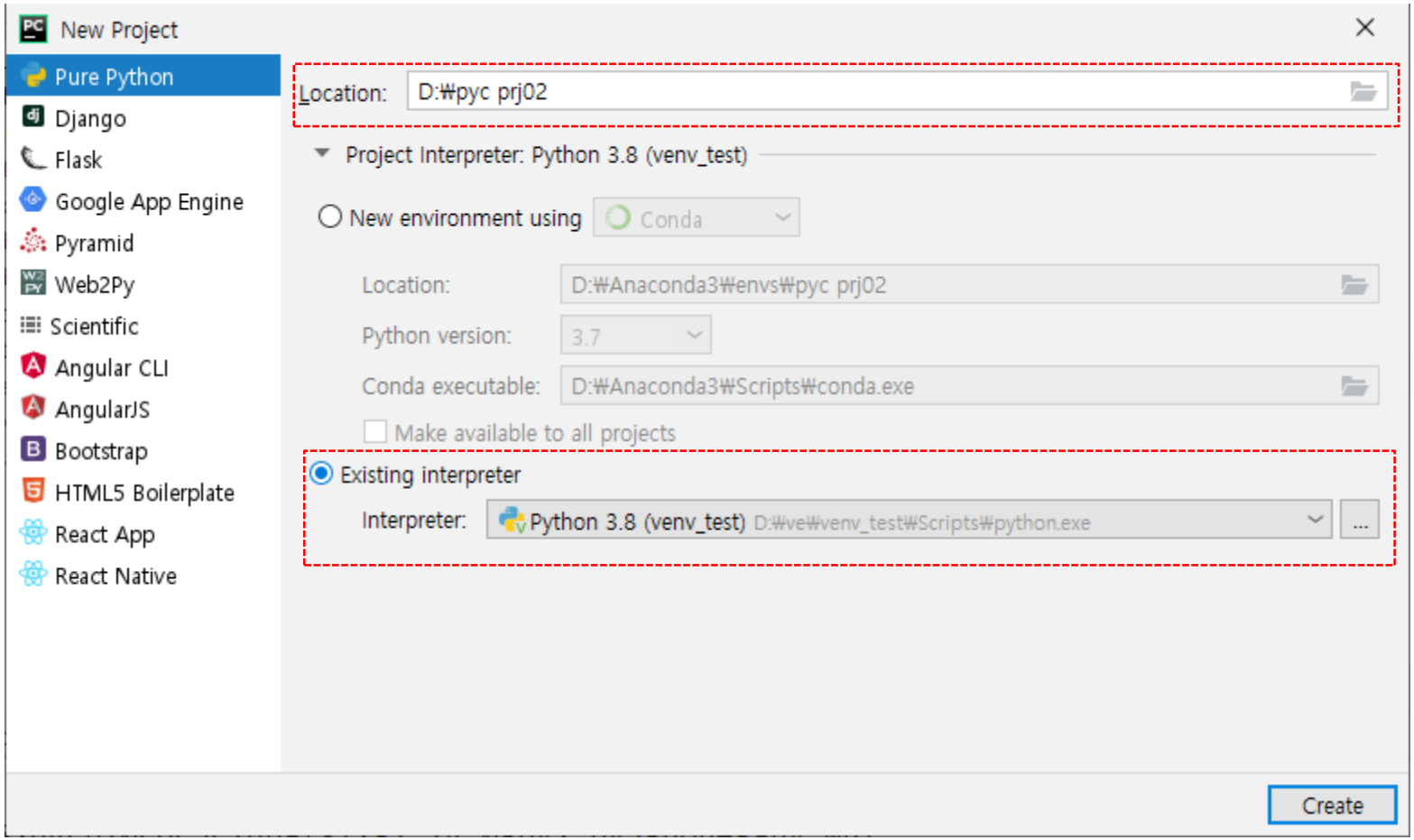
파이참 프로젝트 pyc prj02 만들기(1)

- **D:₩pyc prj02**
  - 가상환경 venv_test의 인터프리터 지정
    - **Venv로 만든 venv_test 지정**
      - **...을 눌러 자신이 venv로 직접 만든 가상환경의 scripts 폴더의 python.exe를 지정**

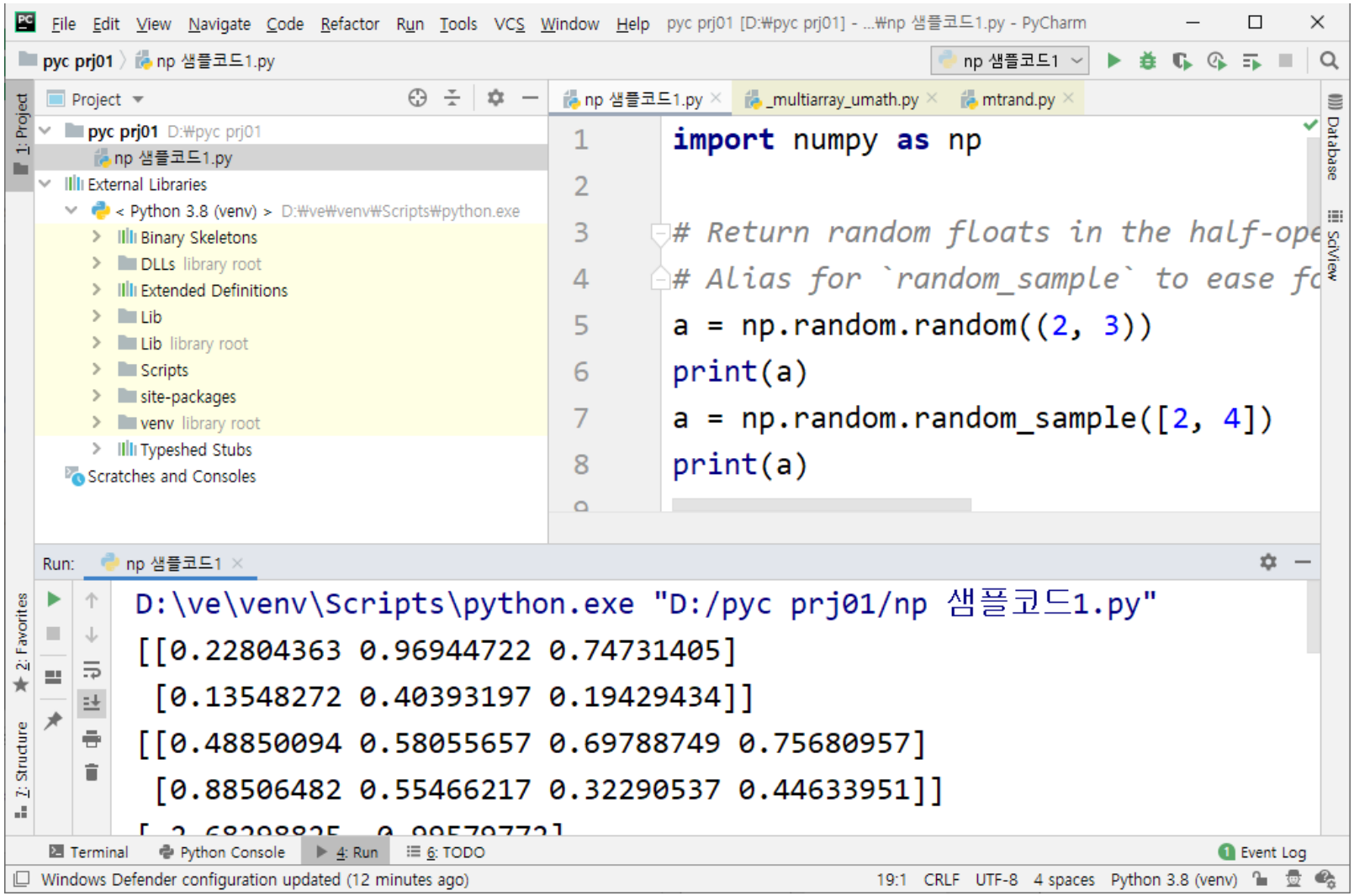


파이참 프로젝트 pyc prj02 만들기(2)

- 자신이 만든 가상환경이 지정된 New Project 대화상자



프로젝트 pyc prj02, 소스 np 샘플코드2.py



프로젝트 pyc prj02, 소스 이해와 결과

```python
import numpy as np

# [0, 1) 난수 생성
a = np.random.random((2, 3))
print(type(a))
print(a)
b = a.reshape(3, 2)
print(b)

# 정규분포의 난수 생성
c = np.random.randn(3, 4)
print(c.reshape(2, 6))

# 값이 모두 1인 텐서
d = np.ones((4, 5))
print(d.reshape(2, 5, 2))

# 값이 모두 0인 텐서
e = np.zeros((3, 4))
print(e.reshape(2, 3, 2))
```

```
D:₩ve₩venv_test₩Scripts₩python.exe "D:/pyc prj02/np 샘플코드2.py"
<class 'numpy.ndarray'>
[[0.82253794 0.94805217 0.18727646]
 [0.93734334 0.64931534 0.34043917]]
[[0.82253794 0.94805217]
 [0.18727646 0.93734334]
 [0.64931534 0.34043917]]
[[ 0.75831872  0.48717349  0.28570681 -1.52723675 -0.44313711 -0.95670083]
 [-0.12955399  0.60647143  0.33132981  0.63755303  1.12009118  0.09013185]]
[[[1. 1.]
  [1. 1.]
  [1. 1.]
  [1. 1.]
  [1. 1.]]

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

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

Process finished with exit code 0
```

프로젝트 pyc prj02, 터미널 활용 모듈 pandas 추가

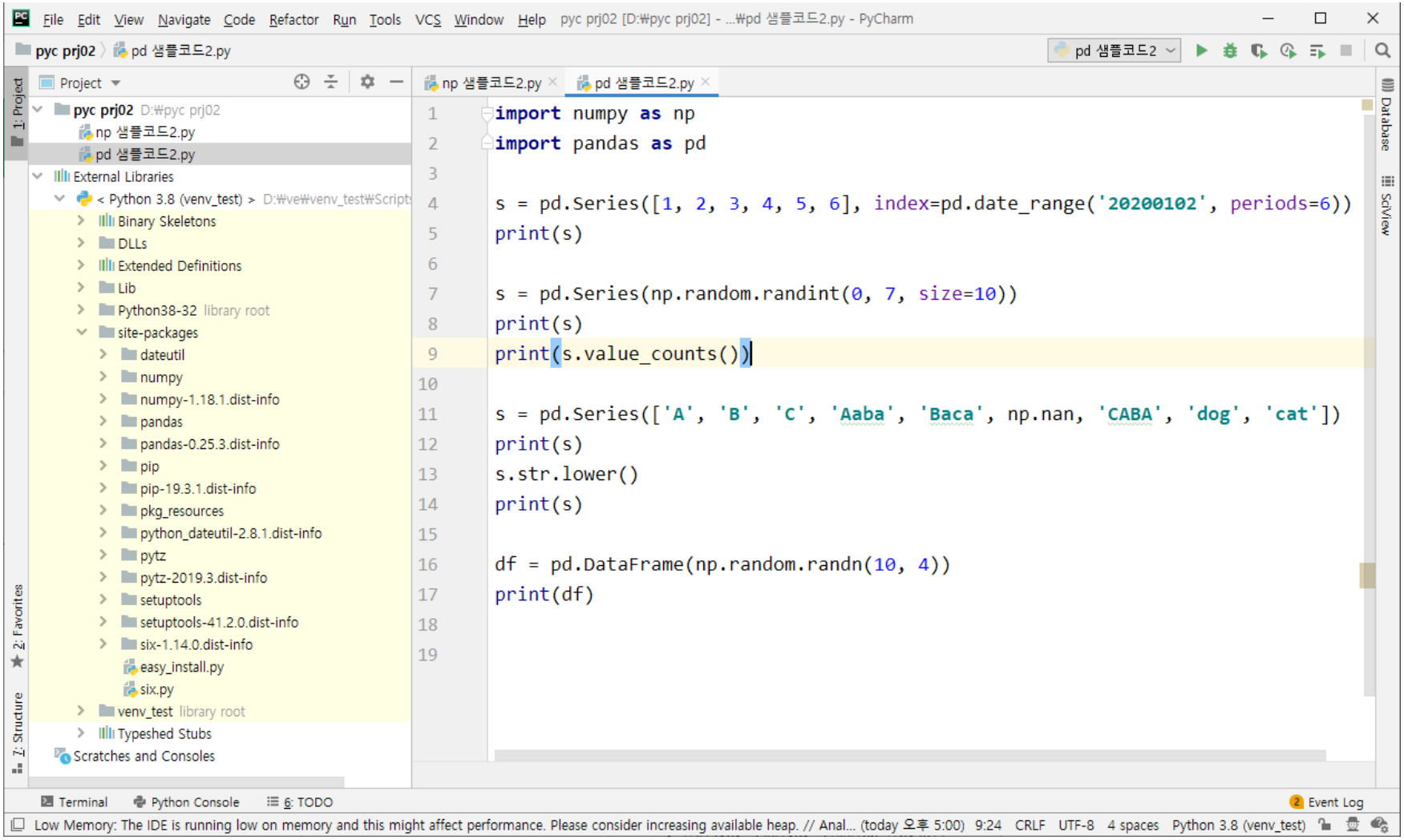
```
(venv_test) D:₩pyc prj02>where pip
D:₩ve₩venv_test₩Scripts₩pip.exe
D:₩Python38-32₩Scripts₩pip.exe
D:₩Anaconda3₩Scripts₩pip.exe
C:₩ProgramData₩Anaconda3₩Scripts₩pip.exe
(venv_test) D:₩pyc prj02>pip --version
pip 19.3.1 from d:₩ve₩venv_test₩lib₩site-packages₩pip
(python 3.8)
(venv_test) D:₩pyc prj02>pip list
Package     Version
----------- -------
numpy       1.18.1
pip         19.3.1
setuptools 41.2.0
```

... pip install pandas

```
(venv_test) D:₩pyc prj02>pip list
Package           Version
----------------- -------
numpy             1.18.1
pandas            0.25.3
pip               19.3.1
python-dateutil 2.8.1
pytz              2019.3
setuptools        41.2.0
six               1.14.0
```

```
(venv_test) D:₩pyc prj02>pip install pandas
  Using cached
https://files.pythonhosted.org/packages/e7/f9/f0b53f88060247251bf4
81fa6ea62cd0d25bf1b11a87888e53ce5b7c8ad2/pytz-2019.3-py2.py3-
none-any.whl

Requirement already satisfied: numpy>=1.13.3 in
d:₩ve₩venv_test₩lib₩site-packages (from pandas) (1.18.1)

Collecting python-dateutil>=2.6.1
  Using cached
https://files.pythonhosted.org/packages/d4/70/d60450c3dd48ef87586
924207ae8907090de0b306af2bce5d134d78615cb/python_dateutil-
2.8.1-py2.py3-none-any.whl

Collecting six>=1.5
  Using cached
https://files.pythonhosted.org/packages/65/eb/1f97cb97bfc2390a2769
69c6fae16075da282f5058082d4cb10c6c5c1dba/six-1.14.0-py2.py3-
none-any.whl

Installing collected packages: pytz, six, python-dateutil, pandas
Successfully installed pandas-0.25.3 python-dateutil-2.8.1 pytz-2019.3
six-1.14.0
```

프로젝트 pyc prj02, 소스 pd 샘플코드2.py



프로젝트 pyc prj02, 소스 pd 샘플코드2.py 이해

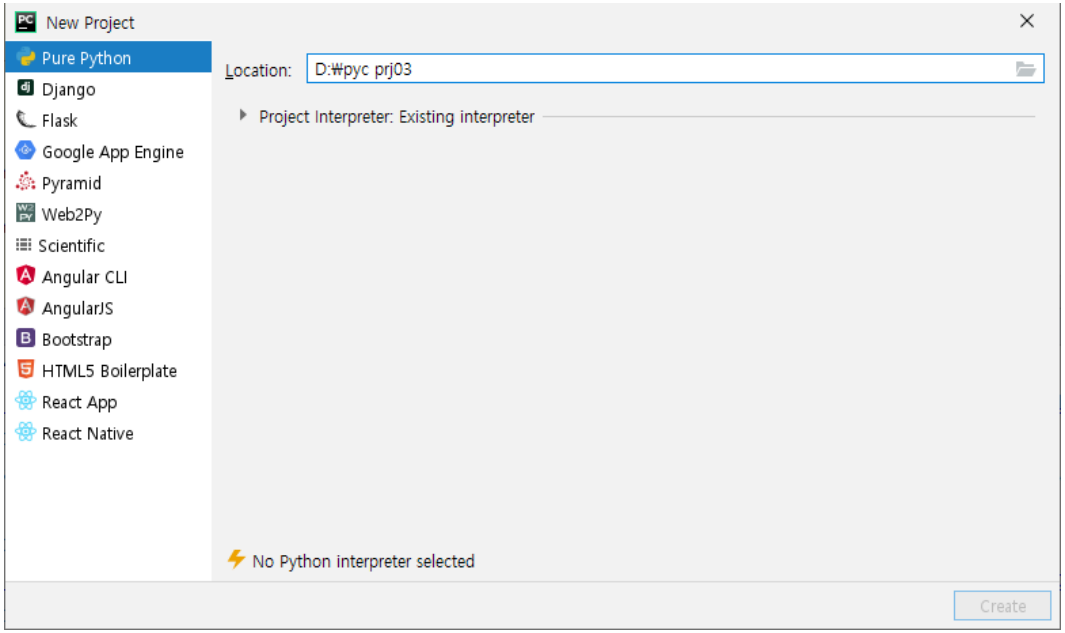
```
import numpy as np
import pandas as pd

s = pd.Series([1, 2, 3, 4, 5, 6], index=pd.date_range('20200102', periods=6))
print(s)

s = pd.Series(np.random.randint(0, 7, size=10))
print(s)
print(s.value_counts())

s = pd.Series(['A', 'B', 'C', 'Aaba', 'Baca', np.nan, 'CABA', 'dog', 'cat'])
print(s)
s.str.lower()
print(s)

df = pd.DataFrame(np.random.randn(10, 4))
print(df)
```

```
D:₩ve₩venv_test₩Scripts₩python.exe
"D:/pyc prj02/pd 샘플코드2.py"
2020-01-02    1
2020-01-03    2
2020-01-04    3
2020-01-05    4
2020-01-06    5
2020-01-07    6
Freq: D, dtype: int64
0    5
1    2
2    0
3    5
4    6
5    1
6    6
7    6
8    2
9    4
dtype: int32
6    3
5    2
2    2
4    1
1    1
0    1
dtype: int64
```

```
0       A
1       B
2       C
3    Aaba
4    Baca
5     NaN
6    CABA
7     dog
8     cat
dtype: object
0       A
1       B
2       C
3    Aaba
4    Baca
5     NaN
6    CABA
7     dog
8     cat
dtype: object
          0         1         2         3
0 -0.827856 -0.591318 -0.446506  1.639843
1 -0.455133  0.652168 -0.542553  0.015321
2 -0.790744  0.088498  0.499716 -0.355695
3  0.252766  0.853125  1.609860 -1.235949
4 -0.778862  0.734792 -0.559469  2.637026
5 -0.066913 -2.701452  0.196265 -1.475756
6 -1.171109 -1.312982 -0.123534 -0.467198
7 -0.560191 -0.025275  0.336903 -0.202051
8 -0.472363  2.441893  2.044766  0.685911
9 -0.899807 -1.176664  0.391078  0.148584

Process finished with exit code 0
```

3개의 파이참 프로젝트 생성과 Existing Interpreter 지정

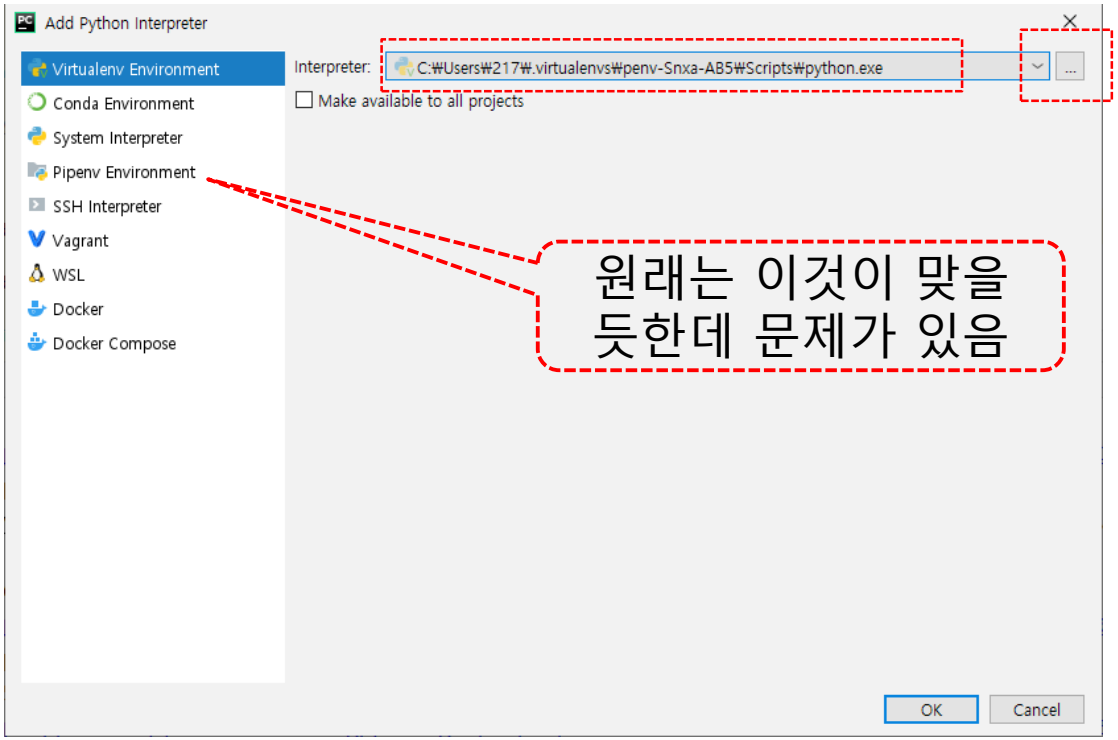
- **D:₩pyc prj01**
  - 가상환경 venv 지정
    - **Virtualenv로 만든 가상환경**
- **D:₩pyc prj02**
  - 가상환경 venv_test 지정
    - **Venv로 만든 가상환경**
- **D:₩pyc prj03**
  - 가상환경 penv 지정
    - **pipenv로 만든 가상환경**



- **Existing interpreter**
  - … 선택
    - **자신이 만든 가상환경 폴더의 scripts 하부 python.exe를 지정**

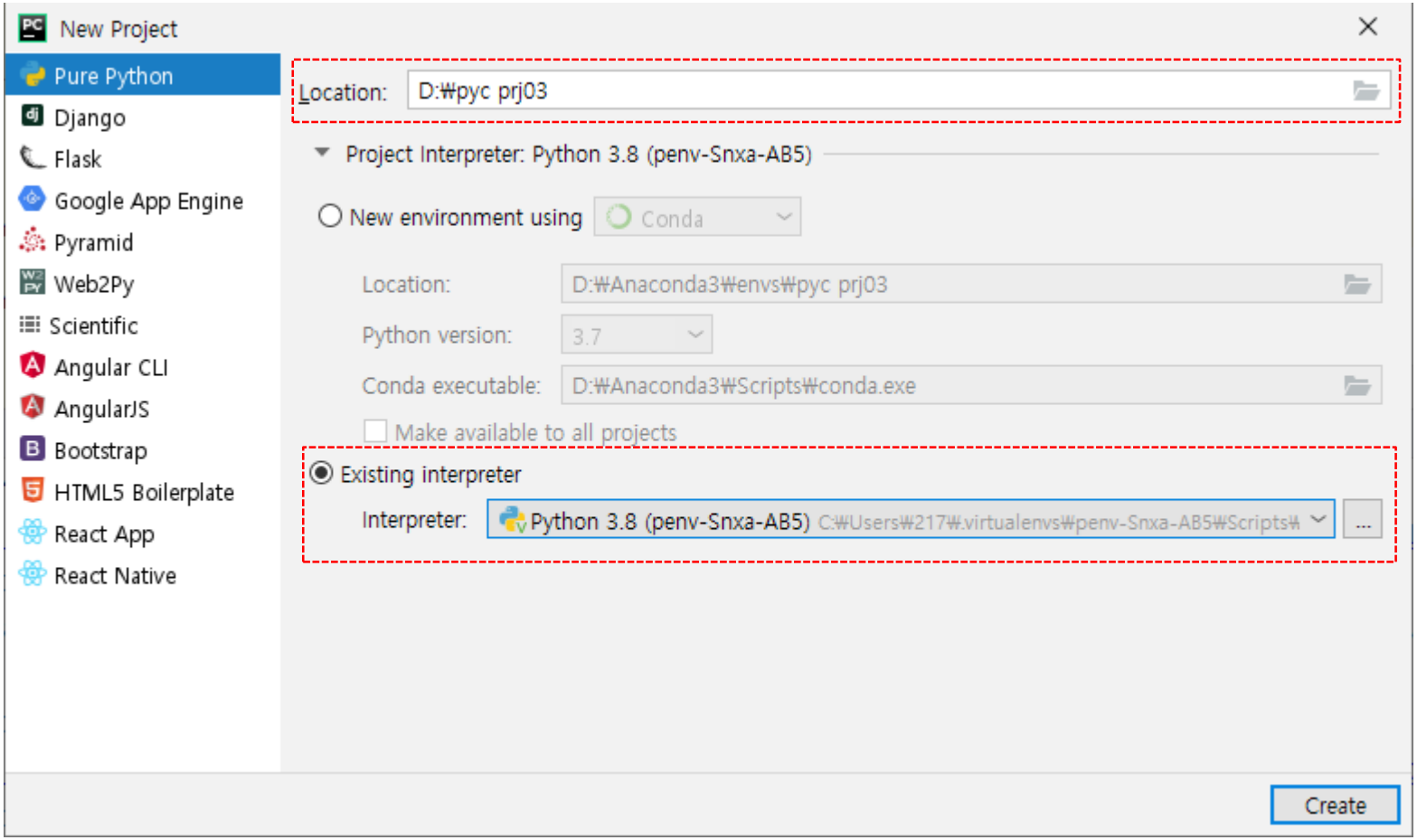
파이참 프로젝트 pyc prj03 만들기(1)

- **D:₩pyc prj03**
  - 가상환경 penv의 인터프리터 지정
    - **₩users₩217₩.virtualenv₩penv-00000₩Scrips₩python 지정**
    - **pipenv로 만든 위 인터프리터**
      - **...을 눌러 자신이 pipenv로 직접 만든 가상환경의 scripts 폴더의 python.exe를 지정**



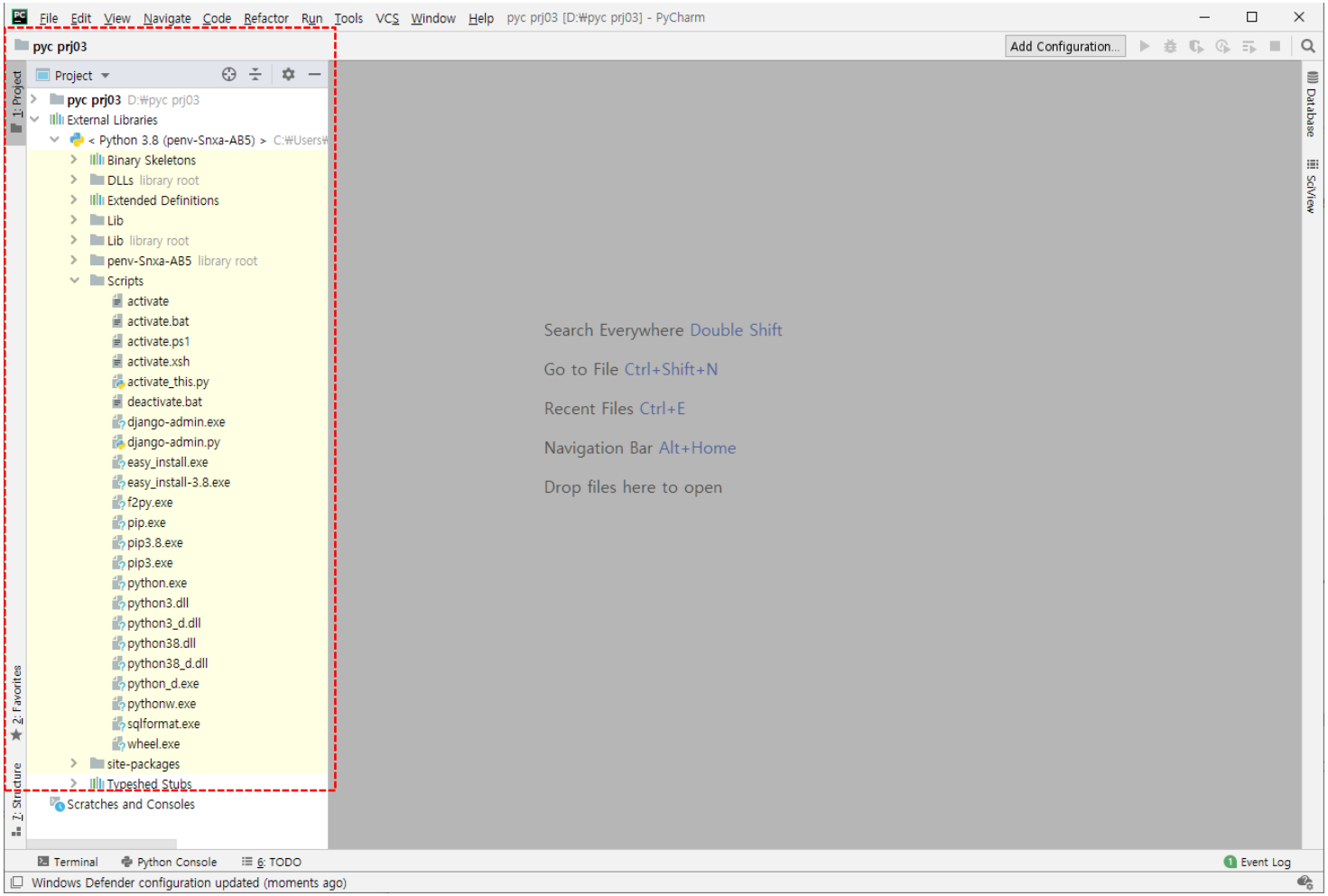
파이참 프로젝트 pyc prj03 만들기(2)

- 자신이 만든 가상환경이 지정된 New Project 대화상자

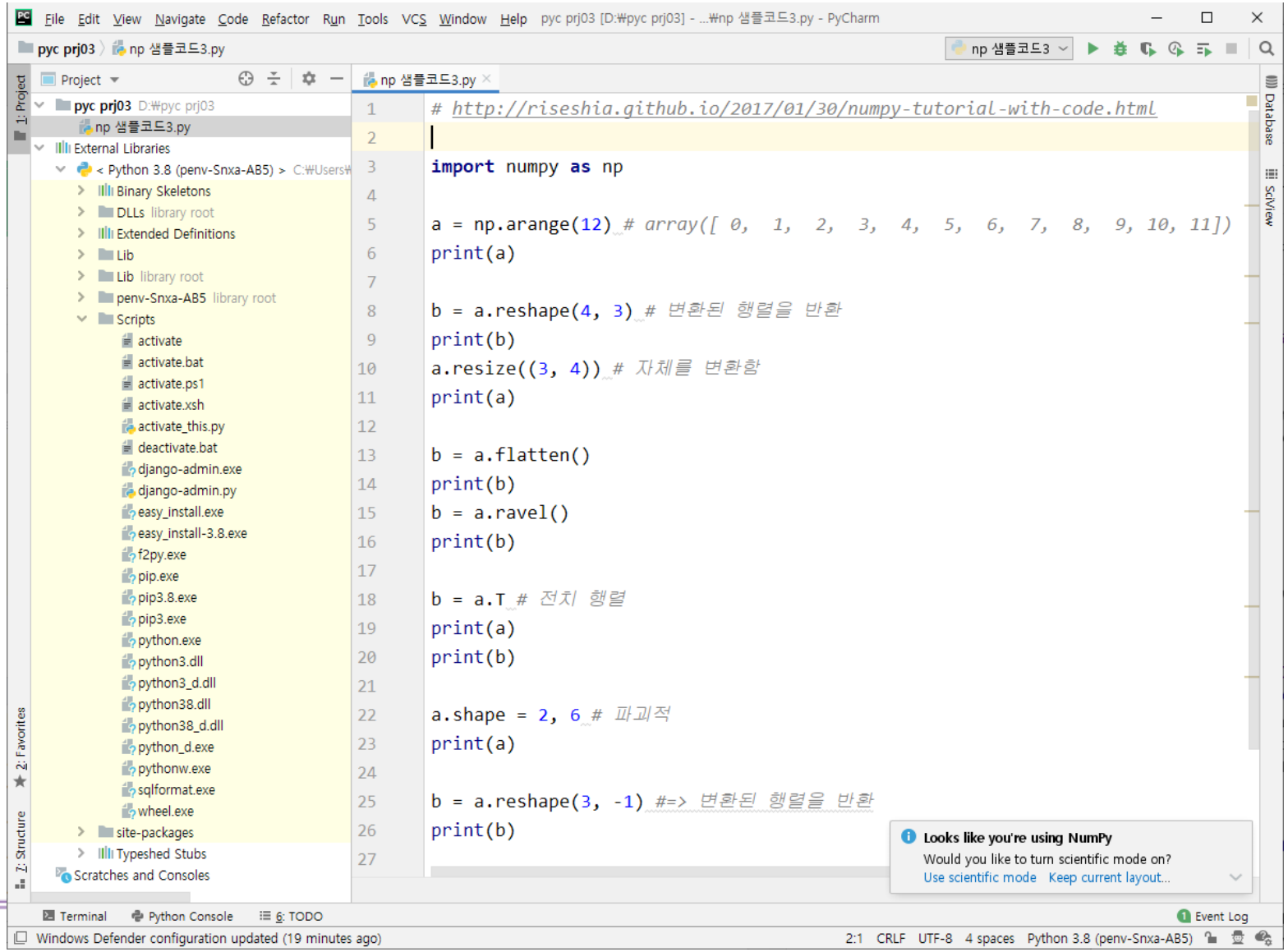


파이참 프로젝트 pyc prj03 만들기(3)

- 가상환경이 설정된 프로젝트



프로젝트 pyc prj03, 소스 np 샘플코드3.py



프로젝트 pyc prj03, 소스 이해와 결과

```
# http://riseshia.github.io/2017/01/30/numpy-tutorial-with-code.html

import numpy as np

a = np.arange(12) # array([ 0, … 11])
print(a)

b = a.reshape(4, 3) # 변환된 행렬을 반환
print(b)
a.resize((3, 4)) # 자체를 변환함
print(a)

b = a.flatten()
print(b)
b = a.ravel()
print(b)

b = a.T # 전치 행렬
print(a)
print(b)

a.shape = 2, 6 # 파괴적
print(a)

b = a.reshape(3, -1) # 변환된 행렬을 반환
print(b)
```

```
C:₩Users₩217₩.virtualenvs₩penv-Snxa-AB5₩Scripts₩python.exe "D:/pyc prj03/np 샘플코드3.py"
[ 0  1  2  3  4  5  6  7  8  9 10 11]
[[ 0  1  2]
 [ 3  4  5]
 [ 6  7  8]
 [ 9 10 11]]
[[ 0  1  2  3]
 [ 4  5  6  7]
 [ 8  9 10 11]]
[ 0  1  2  3  4  5  6  7  8  9 10 11]
[ 0  1  2  3  4  5  6  7  8  9 10 11]
[[ 0  1  2  3]
 [ 4  5  6  7]
 [ 8  9 10 11]]
[[ 0  4  8]
 [ 1  5  9]
 [ 2  6 10]
 [ 3  7 11]]
[[ 0  1  2  3  4  5]
 [ 6  7  8  9 10 11]]
[[ 0  1  2  3]
 [ 4  5  6  7]
 [ 8  9 10 11]]

Process finished with exit code 0
```

프로젝트 pyc prj03, 터미널 활용 모듈 bokeh 추가

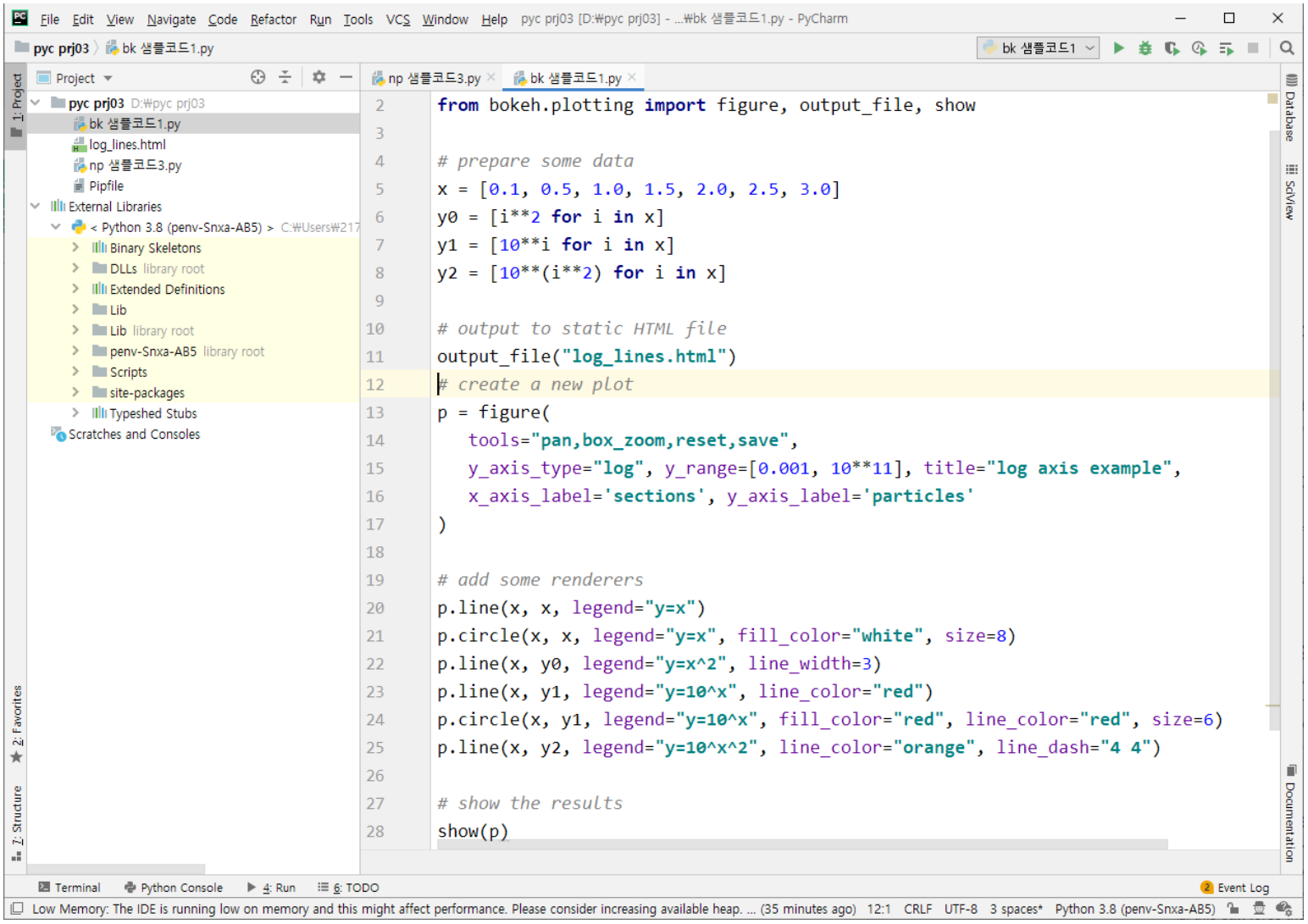
```
(penv) D:₩pyc prj03>pip show pandas
Name: pandas
Version: 0.25.3
Summary: Powerful data structures for data analysis, time series, and statistics
Home-page: http://pandas.pydata.org
Author: None
Author-email: None
License: BSD
Location: c:₩users₩217₩.virtualenvs₩penv-snxa-ab5₩lib₩site-packages
Requires: python-dateutil, numpy, pytz
Required-by:
```

```
(penv) D:₩pyc prj03>pip install bokeh
```

```
(penv) D:₩pyc prj03>pip show bokeh
Name: bokeh
Version: 1.4.0
Summary: Interactive plots and applications in the browser from Python
Home-page: http://github.com/bokeh/bokeh
Author: Bokeh Team
Author-email: info@bokeh.org
License: BSD-3-Clause
Location: c:₩users₩217₩.virtualenvs₩penv-snxa-ab5₩lib₩site-packages
Requires: Jinja2, python-dateutil, six, PyYAML, numpy, tornado, packaging, pillow
Required-by:
```

```
(penv) D:₩pyc prj03>pip install bokeh
Collecting bokeh
  Using cached https://files.pythonhosted.org/packages/de/70/fdd4b186d8570a737372487cc5547aac885a1270626e3ebf03db1808e4ed/bokeh-1.4.0.tar.gz
Requirement already satisfied: six>=1.5.2 in c:₩users₩217₩.virtualenvs₩penv-snxa-ab5₩lib₩site-packages (from bokeh) (1.14.0)
Collecting PyYAML>=3.10
  Downloading https://files.pythonhosted.org/packages/d1/2c/bc6625326e966aa2de85a085f91121330410588fd0bb1fe7603e822e6905/PyYAML-5.3-cp38-cp38-win32.whl (198kB)
     |████████████████████████████████| 204kB 242kB/s
Requirement already satisfied: python-dateutil>=2.1 in c:₩users₩217₩.virtualenvs₩penv-snxa-ab5₩lib₩site-packages (from bokeh) (2.8.1)
Collecting Jinja2>=2.7
  Using cached https://files.pythonhosted.org/packages/65/e0/eb35e762802015cab1ccee04e8a277b03f1d8e53da3ec3106882ec42558b/Jinja2-2.10.3-py2.py3-none-any.whl
Requirement already satisfied: numpy>=1.7.1 in c:₩users₩217₩.virtualenvs₩penv-snxa-ab5₩lib₩site-packages (from bokeh) (1.18.1)
Collecting pillow>=4.0
  Downloading https://files.pythonhosted.org/packages/a0/f5/943da9f188d1abdbd83f73dfba7ed8c1935161e8f9b4ef6fc9cea0b3e14b/Pillow-7.0.0-cp38-cp38-win32.whl (1.8MB)
     |████████████████████████████████| 1.8MB 3.3MB/s
Collecting packaging>=16.8
  Using cached https://files.pythonhosted.org/packages/d8/5b/3098db49a61ccc8583ffead6aedc226f08ff56dc03106b6ec54451e27a30/packaging-20.0-py2.py3-none-any.whl
Collecting tornado>=4.3
  Downloading https://files.pythonhosted.org/packages/30/78/2d2823598496127b21423baffaa186b668f73cd91887fcef78b6eade136b/tornado-6.0.3.tar.gz (482kB)
     |████████████████████████████████| 491kB 6.8MB/s
Collecting MarkupSafe>=0.23
  Downloading https://files.pythonhosted.org/packages/b9/2e/64db92e53b86efccfaea71321f597fa2e1b2bd3853d8ce658568f7a13094/MarkupSafe-1.1.1.tar.gz
Requirement already satisfied: pyparsing>=2.0.2 in c:₩users₩217₩.virtualenvs₩penv-snxa-ab5₩lib₩site-packages (from packaging>=16.8->bokeh) (2.4.6)
Building wheels for collected packages: bokeh, tornado, MarkupSafe
  Building wheel for bokeh (setup.py) ... done
  Created wheel for bokeh: filename=bokeh-1.4.0-cp38-none-any.whl size=23689221 sha256=e3777f86c58c996a998d4fdca2de98e1facdc47ee36b5e4f66c67a7f2ddabb46
  Stored in directory: C:₩Users₩217₩AppData₩Local₩pip₩Cache₩wheels₩fb₩f8₩47₩09700d9a19cbcbf0b7a3130690b75c0d6ff80fbda0b1774c7c
  Building wheel for tornado (setup.py) ... done
  Created wheel for tornado: filename=tornado-6.0.3-cp38-cp38-win32.whl size=415170 sha256=8429821cef96a6e48bd6c9973e2939f20cbee1bf0149466ed0ea76a984b5d32f
  Stored in directory: C:₩Users₩217₩AppData₩Local₩pip₩Cache₩wheels₩84₩bf₩40₩2f6ef700f48401ca40e5e3dd7d0e3c0a90e064897b7fe5fc08
  Building wheel for MarkupSafe (setup.py) ... done
  Created wheel for MarkupSafe: filename=MarkupSafe-1.1.1-cp38-cp38-win32.whl size=18939 sha256=a7fe4fde6d57712384d0566d79e396099cb675c9e7b3eb0f0f116fd79fdc4426
  Stored in directory: C:₩Users₩217₩AppData₩Local₩pip₩Cache₩wheels₩f2₩aa₩04₩0edf07a1b8a5f5f1aed7580fffb69ce8972edc16a505916a77
Successfully built bokeh tornado MarkupSafe
Installing collected packages: PyYAML, MarkupSafe, Jinja2, pillow, packaging, tornado, bokeh
Successfully installed Jinja2-2.10.3 MarkupSafe-1.1.1 PyYAML-5.3 bokeh-1.4.0 packaging-20.0 pillow-7.0.0 tornado-6.0.3
```

프로젝트 pyc prj03, 소스 bk 샘플코드1.py



프로젝트 pyc prj03, 소스 bk 샘플코드1.py 이해와 결과

```
#
https://docs.bokeh.org/en/latest/docs/user_guide/quickstart.html#userguide-
quickstart
from bokeh.plotting import figure, output_file, show

# prepare some data
x = [0.1, 0.5, 1.0, 1.5, 2.0, 2.5, 3.0]
y0 = [i**2 for i in x]
y1 = [10**i for i in x]
y2 = [10**(i**2) for i in x]

# output to static HTML file
output_file("log_lines.html")

# create a new plot
p = figure(
   tools="pan,box_zoom,reset,save",
   y_axis_type="log", y_range=[0.001, 10**11],
   title="log axis example",
   x_axis_label='sections', y_axis_label='particles'
)

# add some renderers
p.line(x, x, legend="y=x")
p.circle(x, x, legend="y=x", fill_color="white", size=8)
p.line(x, y0, legend="y=x^2", line_width=3)
p.line(x, y1, legend="y=10^x", line_color="red")
p.circle(x, y1, legend="y=10^x", fill_color="red", line_color="red", size=6)
p.line(x, y2, legend="y=10^x^2", line_color="orange", line_dash="4 4")

# show the results
show(p)
```