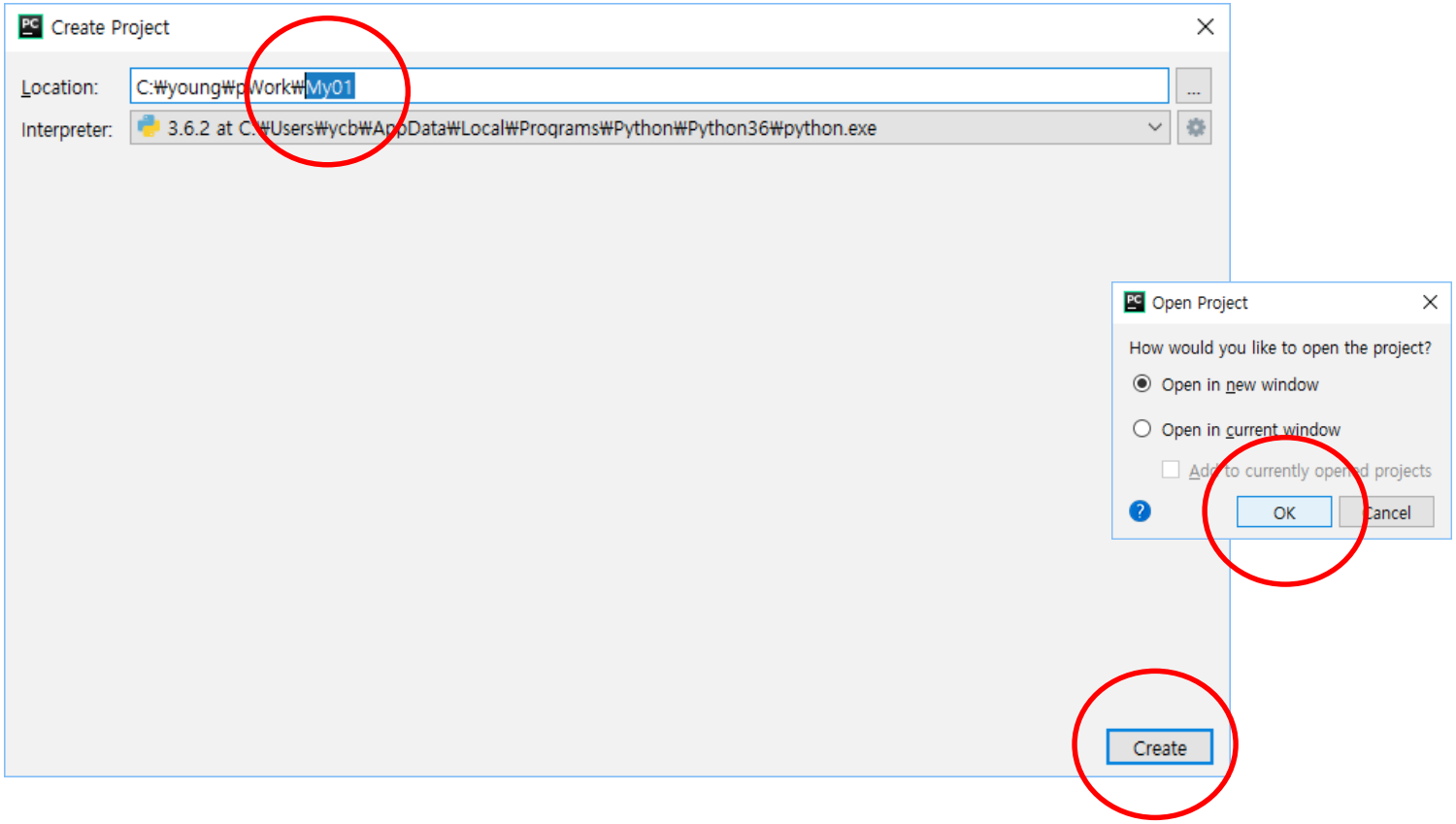
Python 클래스 작성 및 모듈화

변영철 교수
(ycb@jejunu.ac.kr)

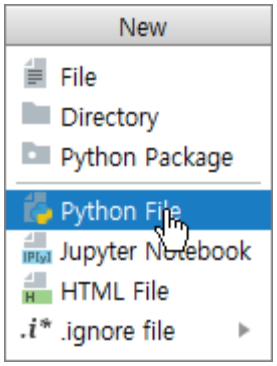
1절. 아주 간단한 Python 프로그램

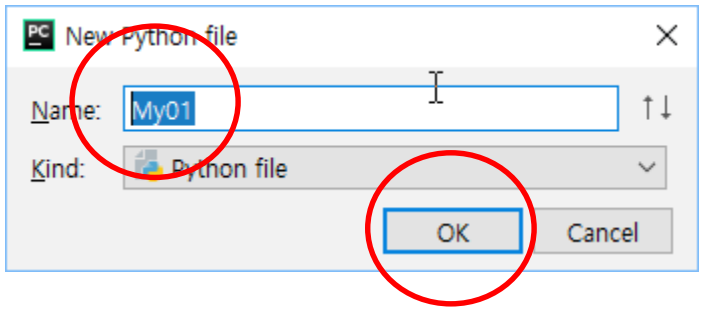
- File | New Project



1절. 아주 간단한 Python 프로그램

- File | New...





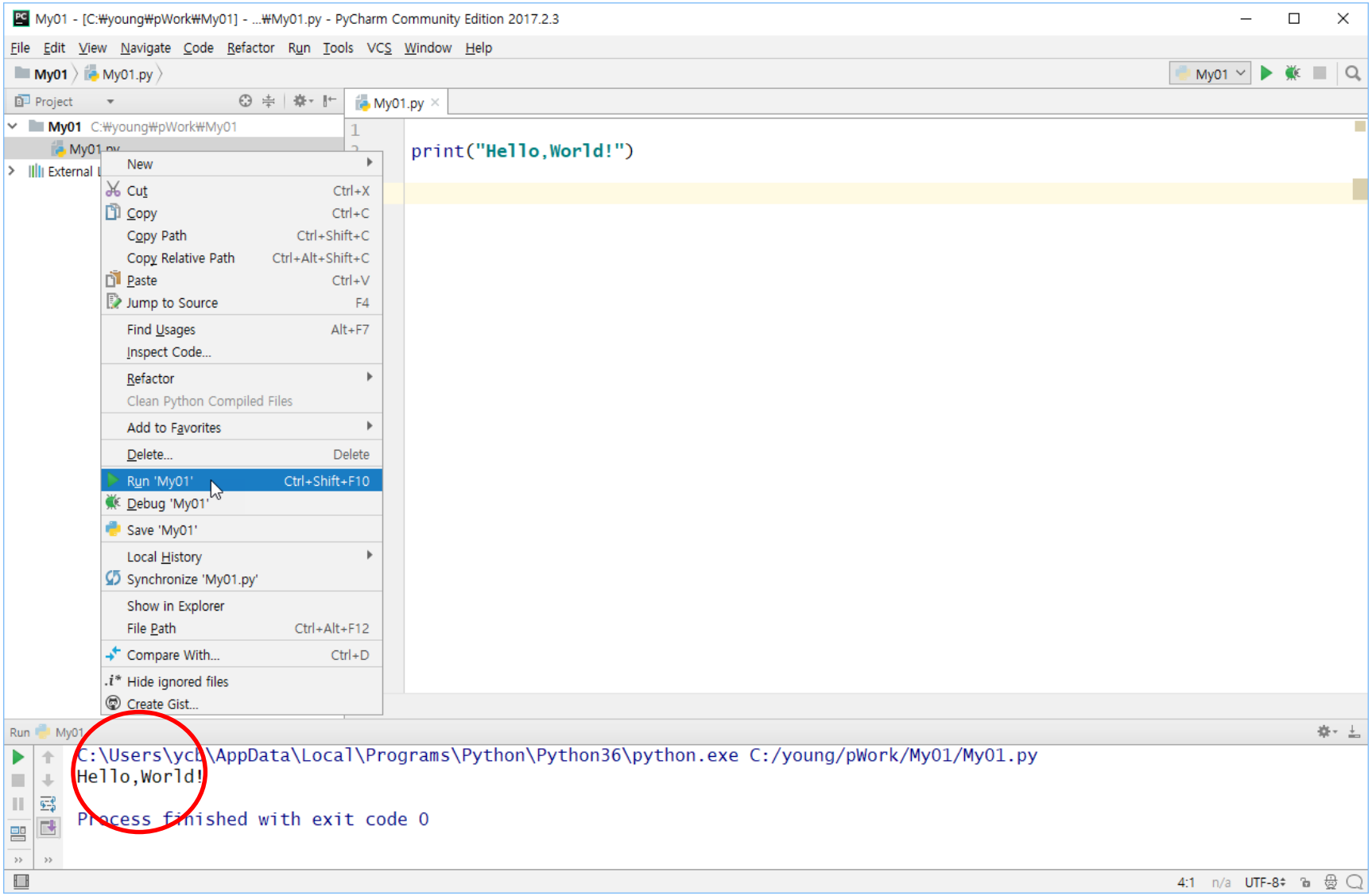
1절. 아주 간단한 Python 프로그램

- **코드 입력**

```
print("Hello,World!")
```

- 파이썬은 실행이 시작되는 메인 함수가 없음.
- 그 대신 들여쓰기 하지 않은 모든 코드(level 0 코드)가 실행됨.

1절. 아주 간단한 Python 프로그램



1절. 아주 간단한 Python 프로그램

"직접 실행되었다면"

```
if __name__ == "__main__"
    print("Hello,World!")
```

- __name__ 내장 변수
- 해당 코드가 직접 실행되는 경우 __name__에 __main__이 자동으로 설정됨.

1절. 아주 간단한 Python 프로그램

- **코드 입력**

```
def main():
    print("Hello,World!")

main()
```

2절. 조금 복잡한 Python 프로그램

- 아래와 같이 2개의 지역변수를 이용하자.

```
def main():
    iX = 2
    iY = 3
    iResult = iX + iY

    print("Sum = ", iResult)

main()
```

2절. 조금 복잡한 Python 프로그램

- 전역변수 만들기(?)

```
iX = 0
iY = 0

def main():
    iX = 2
    iY = 3
    iResult = iX + iY

    print("Sum = ", iResult)

main()
```

3절. 함수를 이용한 프로그램

- 추상화(abstraction)
  - 복잡한 내용을 간단하게 줄여서 표현하는 것
  - 예) 어제 무엇을 했나요?
- 코드 추상화
  - 복잡한 코드를 간단히 표현하는 것
  - 코드 추상화 하기 : Assign, Add

3절. 함수를 이용한 프로그램

- 코드 추상화 – Assign(), Add()

```
iX = 0
iY = 0

def Assign(a, b):
    iX = a
    iY = b

def Add():
    return iX + iY

def main():
    Assign(2, 3)
    iResult = Add()

    print("Sum = ", iResult)
```

실행결과:

```
Sum =  0
Process finished with exit code 0
```

3절. 함수를 이용한 프로그램

```
iX = 0
iY = 0

def Assign(a, b):
    global iX, iY
    iX = a
    iY = b

def Add():
    global iX, iY
    return iX + iY

def main():
    Assign(2, 3)
    iResult = Add()

    print("Sum = ", iResult)

main()
```

실행결과:

```
Sum =  5
Process finished with exit code 0
```

4절. 클래스 모듈화

```
class Point:
    iX = 0
    iY = 0

    def Assign(self, a, b):
        self.iX = a
        self.iY = b

    def Add(self):
        return self.iX + self.iY

def main():
    gildong = Point()
    gildong.Assign(2, 3)
    iResult = gildong.Add()
    print("Sum:", iResult)

main()
```

4절. 다른 파일로 분리

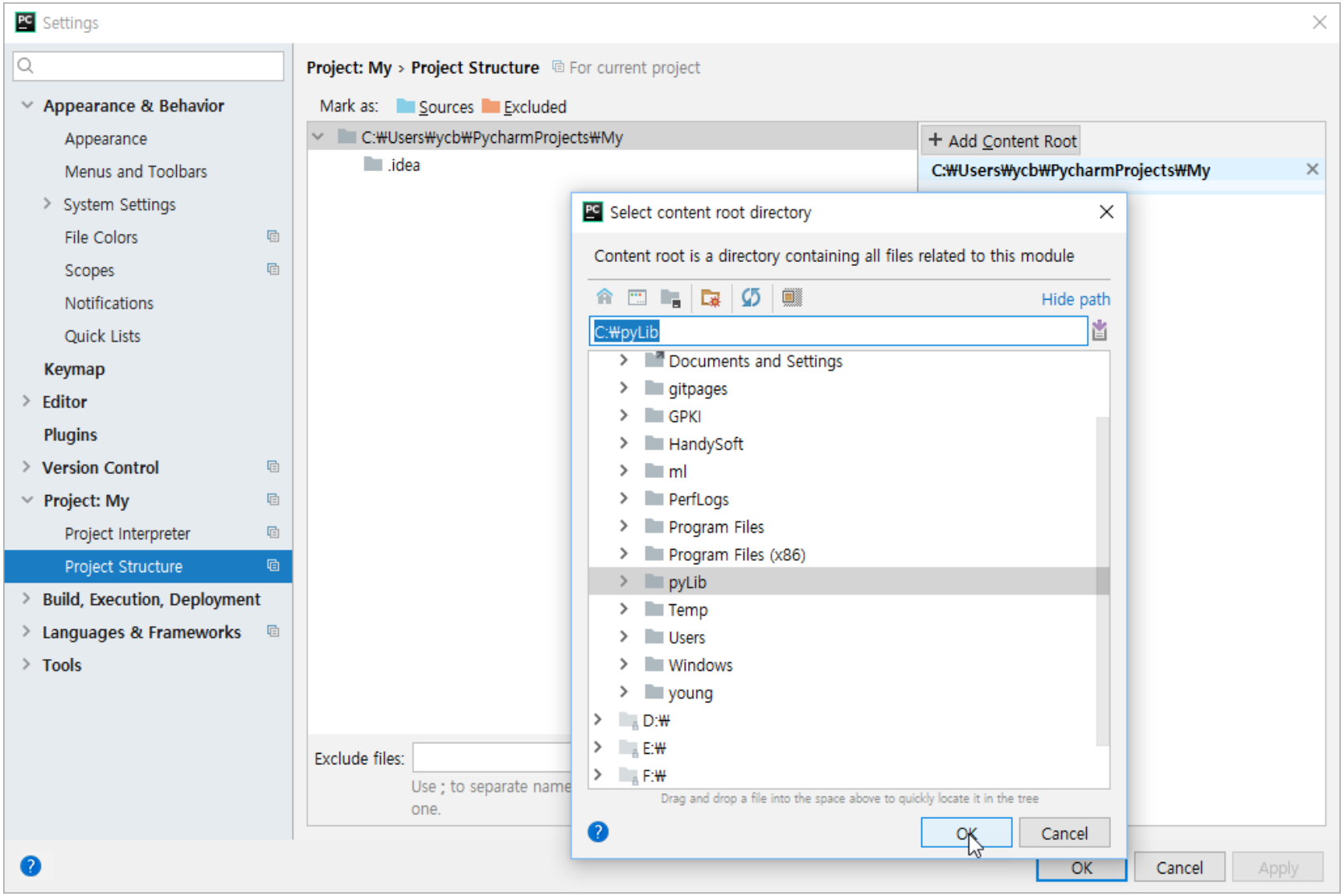
```
#My.py
import point

def main():
    gildong = point.Point()
    gildong.Assign(2, 3)
    iResult = gildong.Add()
    print("Sum:", iResult)

main()
```

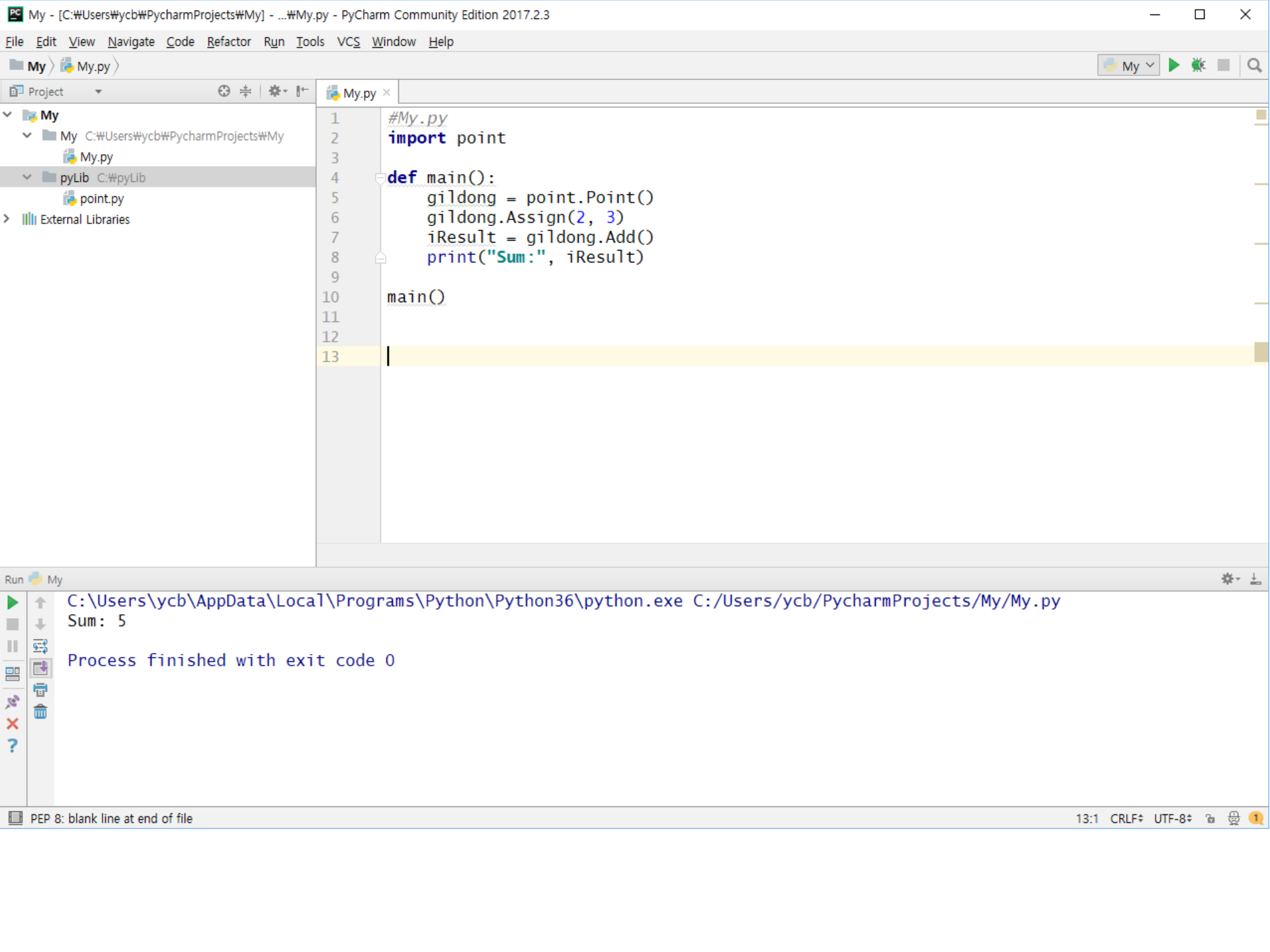
```
#point.py
class Point:
    iX = 0
    iY = 0

    def Assign(self, a, b):
        self.iX = a
        self.iY = b

    def Add(self):
        return self.iX + self.iY
```

5절. 나만의 라이브러리 폴더로 저장

- c:₩pyLib 폴더 생성 후 point.py를 그곳으로 이동

5절. 나만의 라이브러리 폴더 설정




My - [C:₩Users₩ycb₩PycharmProjects₩My] - ...₩My.py - PyCharm Community Edition 2017.2.3
File Edit View Navigate Code Refactor Run Tools VCS Window Help
My My.py
My
Project
My
My C:₩Users₩ycb₩PycharmProjects₩My
My.py
pyLib C:₩pyLib
point.py
External Libraries
My.py
#My.py
import point
def main():
gildong = point.Point()
gildong.Assign(2, 3)
iResult = gildong.Add()
print("Sum:", iResult)
main()
Run My
C:\Users\ycb\AppData\Local\Programs\Python\Python36\python.exe C:/Users/ycb/PycharmProjects/My/My.py
Sum: 5
Process finished with exit code 0
PEP 8: blank line at end of file
13:1 CRLF UTF-8

6절. import as

```
#My.py
import point as p

def main():
    gildong = p.Point()
    gildong.Assign(2, 3)
    iResult = gildong.Add()
    print("Sum:", iResult)

main()
```

```
import tensorflow as tf

#----- training data
x_data = [[1]]
y_data = [[1]]

#----- a neuron
w = tf.Variable(tf.random_normal([1,1]))
hypo = w * x_data

#----- learning
cost = (hypo - y_data) ** 2

train = tf.train.GradientDescentOptimizer(learning_rate=0.01).minimize(cost)

sess = tf.Session()
sess.run(tf.global_variables_initializer())

for i in range(1001):
    sess.run(train)

    if i % 100 == 0:
        print('w:', sess.run(w), 'cost:', sess.run(cost))

#----- testing(prediction)
x_data = [3]
print(sess.run(x_data * w))
```

모듈화 과정

- 전체 코드를 xxx 함수로 추상화
- My 클래스로 추상화
- 멤버 변수로 선언 → x_data, y_data, w, hypo, cost, train, sess 등
- set_data 멤버함수(메소드)
- init_network 메소드
- learn 메소드
- test 메소드