

Python 변수, 함수, 클래스, 모듈 이해하기

변영철 교수

(ycb@jejunu.ac.kr)

Python 설치하기
구글에서 'Python' 검색

PyCharm 설치하기
'PyCharm down' 검색

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



Installation Options

Configure your PyCharm Community Edition installation

Create Desktop Shortcut

☒ 64-bit launcher

Update PATH variable (restart needed)

☒ Add launchers dir to the PATH

Update context menu

☒ Add "Open Folder as Project"

Create Associations

☒ .py

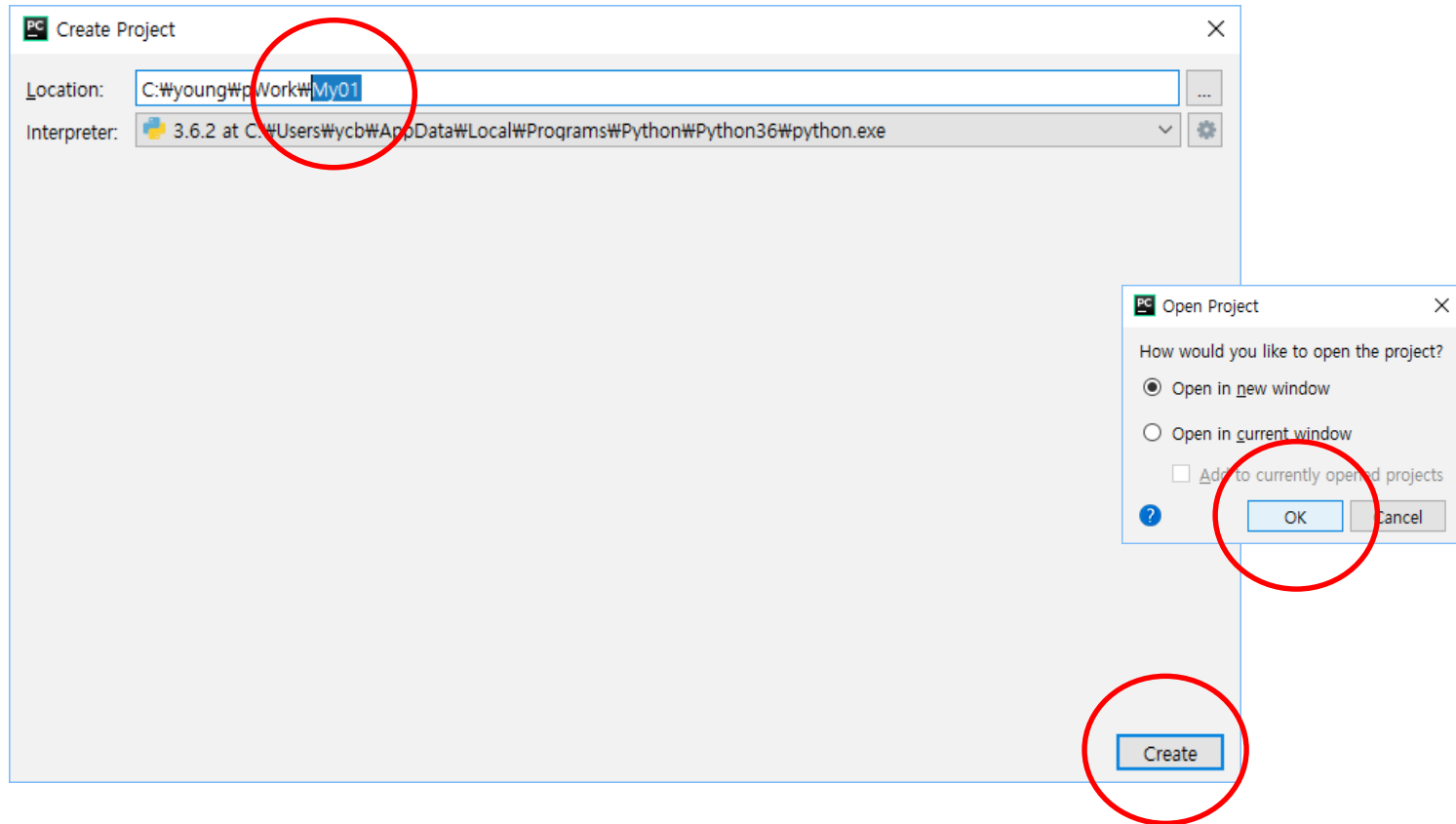
< Back

Next >

Cancel

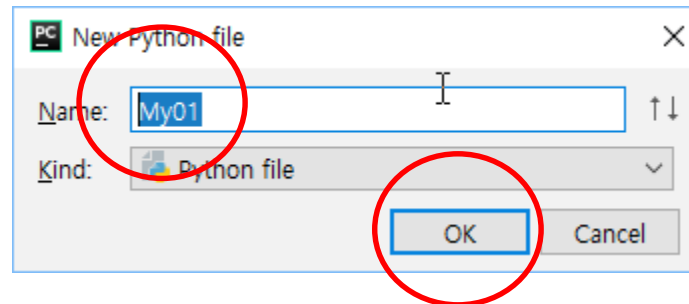
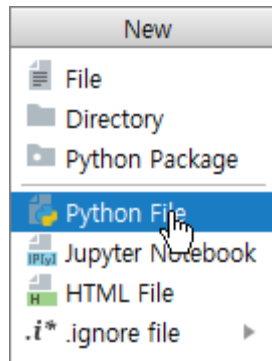
1절. 아주 간단한 Python 프로그램

- File | New Project



1절. 아주 간단한 Python 프로그램

- File | New...



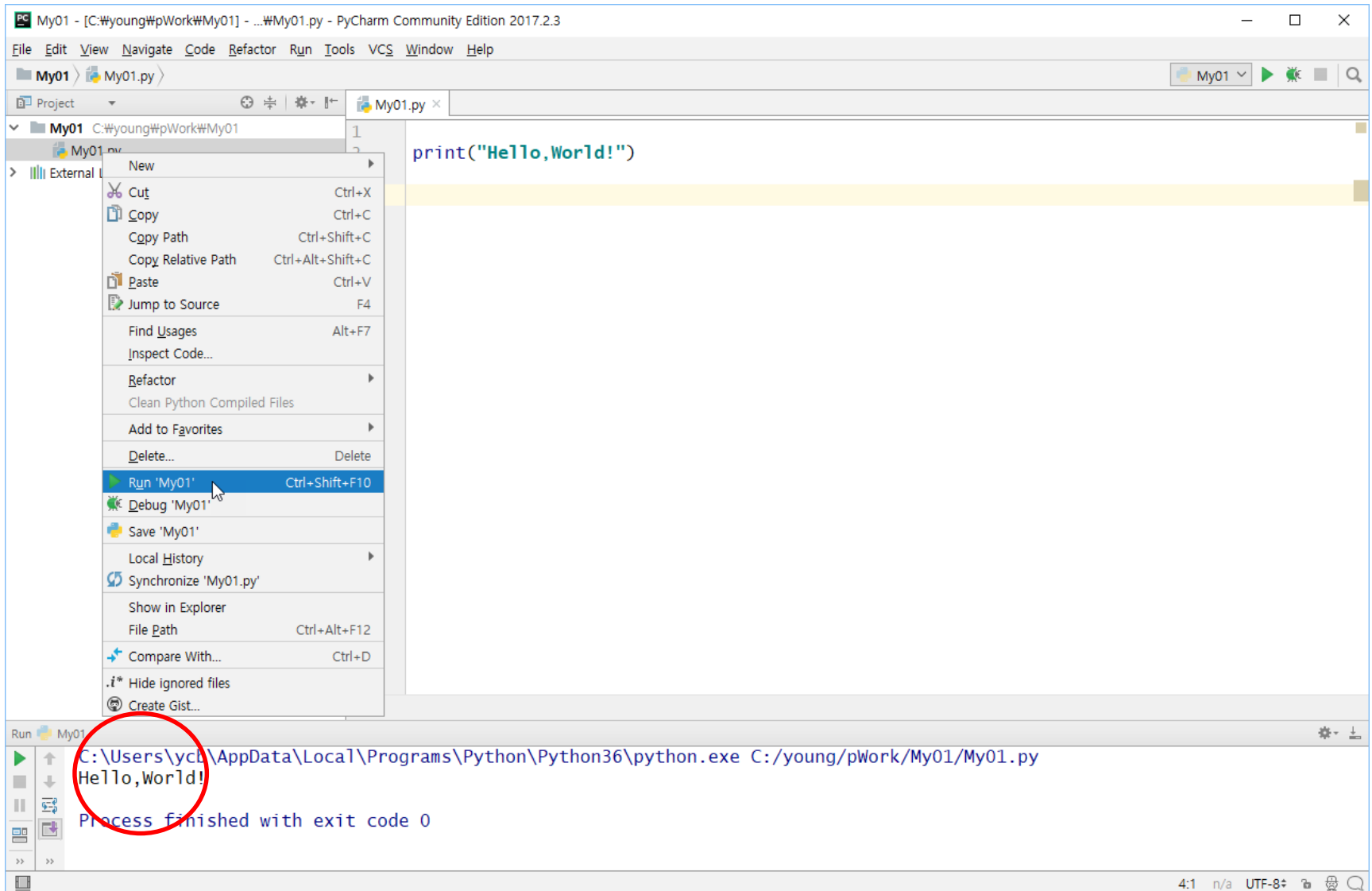
1절. 아주 간단한 Python 프로그램

- 코드 입력

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

- 파이썬은 실행이 시작 되는 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

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

"실행된 코드라면"

- `__name__` 내장 변수
- 프로그램을 직접 실행하면 `__name__`에 `__main__`이 자동으로 설정됨.
- `import`하여 실행할 경우에는 모듈 이름이 설정됨.

main.py

```
print('Hello,World!')  
print(__name__)
```

main.py

```
import module
```

module.py

```
print('Hello,World!')  
print(__name__)
```

main.py

```
import module
```

module.py

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

```
print(__name__)
```

main.py

```
import module
```

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

module.py

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

```
print(__name__)
```

1절. 아주 간단한 Python 프로그램

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

2절. 조금 복잡한 Python 프로그램

```
iX = 2
```

```
iY = 3
```

```
iResult = iX + iY
```

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


변수란 무엇? 값을 담는 그릇
변수는 언제 만들어질까?

변수가 만들어지는 곳은?

iX = 2

iY = 3

iResult = iX + iY

print("Sum = ", iResult)

어제 뭐했어요?

iX = 2

iY = 3

iResult = iX + iY

print("Sum = ", iResult)

3절. 추상화와 함수

```
def assign():
```

```
    iX = 2
```

```
    iY = 3
```

```
assign()
```

```
iResult = iX + iY
```

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

- 추상화(abstraction)

- 복잡한 내용을 묶어서 간단히 표현하는 것

- 예) 어제 무엇을 했나요?

- 코드 추상화

- 복잡한 코드를 간단히 표현하는 것

- 코드 추상화 하기 : Assign, Add

무슨 문제?

```
def assign():
```

```
    iX = 2
```

```
    iY = 3
```

```
assign()
```

```
iResult = iX + iY
```

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



iX = 0

iY = 0

```
def assign():
```

```
    iX = 2
```

```
    iY = 3
```

변수가 만들어지는 곳은?

```
assign()
```

```
iResult = iX + iY
```

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

iX = 0

iY = 0

변수가 만들어지는 곳은?

```
def assign():
```

```
    global iX, iY
```

```
    iX = 2
```

```
    iY = 3
```

```
assign()
```

```
iResult = iX + iY
```

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


global 의미는?

지역변수 만들지 말고
전역변수 이용하라!

```
iX = 0
```

```
iY = 0
```

```
def assign():  
    global iX, iY  
    iX = 2  
    iY = 3
```

```
def add():  
    return iX + iY
```

```
assign()  
iResult = add()
```

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



어떻게 해야 할까?



4절. 두번째 추상화

문자 시작

```
iX = 0
```

```
iY = 0
```

```
def assign():
```

```
    global iX, iY
```

```
    iX = 2
```

```
    iY = 3
```

```
def add():
```

```
    return iX + iY
```

끝

```
assign()
```

```
iResult = add()
```

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

묶자 **XXX** 시작

iX = 0

iY = 0

```
def assign():  
    global iX, iY  
    iX = 2  
    iY = 3
```

```
def add():  
    return iX + iY
```

끝

```
assign()  
iResult = add()
```

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

코드뿐만 아니라
변수(데이터)까지도
한나로 묶어서 추상화

코드 변수(데이터) 추상화
변수(데이터) 추상화
데이터 추상화

뭉어서 만든 XXX를
무엇이라고 할까?

추상 자료형

자료형은 뭐하라고 있는 것?

변수 만들라고 있는 것

gildong = XXX()

변수를 여러 개 만들어보자.

gildong = XXX()

youngja = XXX()

cheolsu = XXX()

```
gildong = 사람()  
youngja = 사람()  
cheolsu = 사람()
```

사람이라는 부류(class)

XXX라는 부류(class)

부류 = class

```
class XXX:
    iX = 0
    iY = 0

    def assign():
        global iX, iY
        iX = 2
        iY = 3

    def add():
        return iX + iY

gildong = XXX()

gildong.assign()
iResult = gildong.add()

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

```
class XXX:
    iX = 0
    iY = 0

    def assign(self):
        self.iX = 2
        self.iY = 3

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

gildong = XXX()

gildong.assign()
iResult = gildong.add()

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

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

    def assign(self):
        self.iX = 2
        self.iY = 3

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

gildong = Point()

gildong.assign()
iResult = gildong.add()

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

5절. 다른 파일로 분리

```
import point
```

```
gildong = point.Point()
```

```
gildong.assign()
```

```
iResult = gildong.add()
```

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

```
#point.py
```

```
class Point:
```

```
    iX = 0
```

```
    iY = 0
```

```
    def assign(self):
```

```
        self.iX = 2
```

```
        self.iY = 3
```

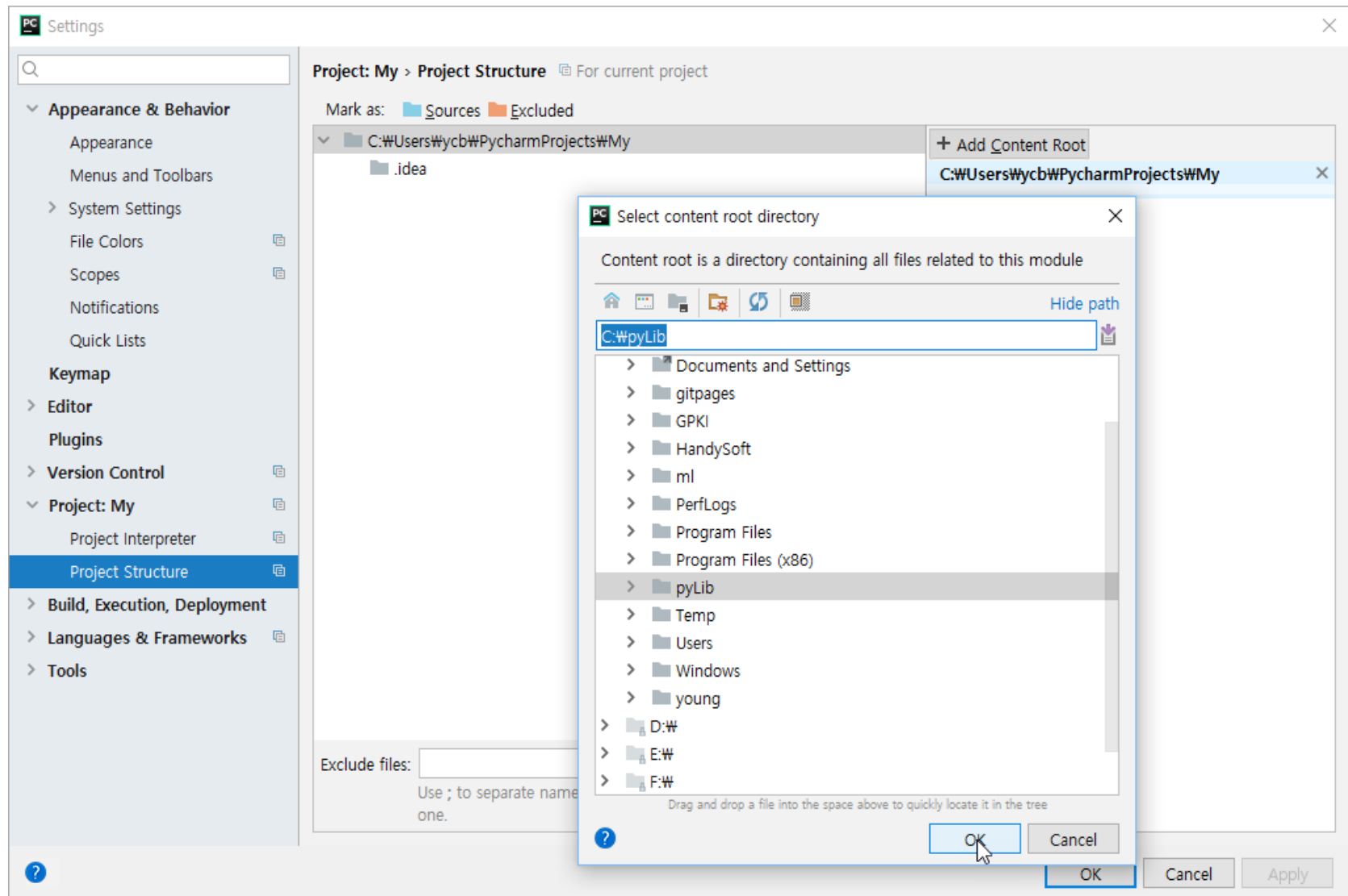
```
    def add(self):
```

```
        return self.iX + self.iY
```

6절. 나만의 라이브러리 폴더로 저장

- c:\WpyLib 폴더 생성 후 point.py를 그곳으로 이동

6절. 나만의 라이브러리 폴더 설정



My > My.py

Project

- My
 - My C:\Users\ycb\PycharmProjects\My
 - My.py
 - pyLib C:\pyLib
 - point.py
- External Libraries

```
1 #My.py
2 import point
3
4 def main():
5     gildong = point.Point()
6     gildong.Assign(2, 3)
7     iResult = gildong.Add()
8     print("Sum:", iResult)
9
10 main()
11
12
13
```

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

7절. import as

```
import point
```

```
gildong = point.Point()
```

```
gildong.assign()
```

```
iResult = gildong.add()
```

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

```
import point as p
```

```
gildong = p.Point()
```

```
gildong.assign()
```

```
iResult = gildong.add()
```

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

```
from point import Point
```

```
gildong = Point()
```

```
gildong.assign()
```

```
iResult = gildong.add()
```

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