

C API

Visual studio for debug

Application Settings

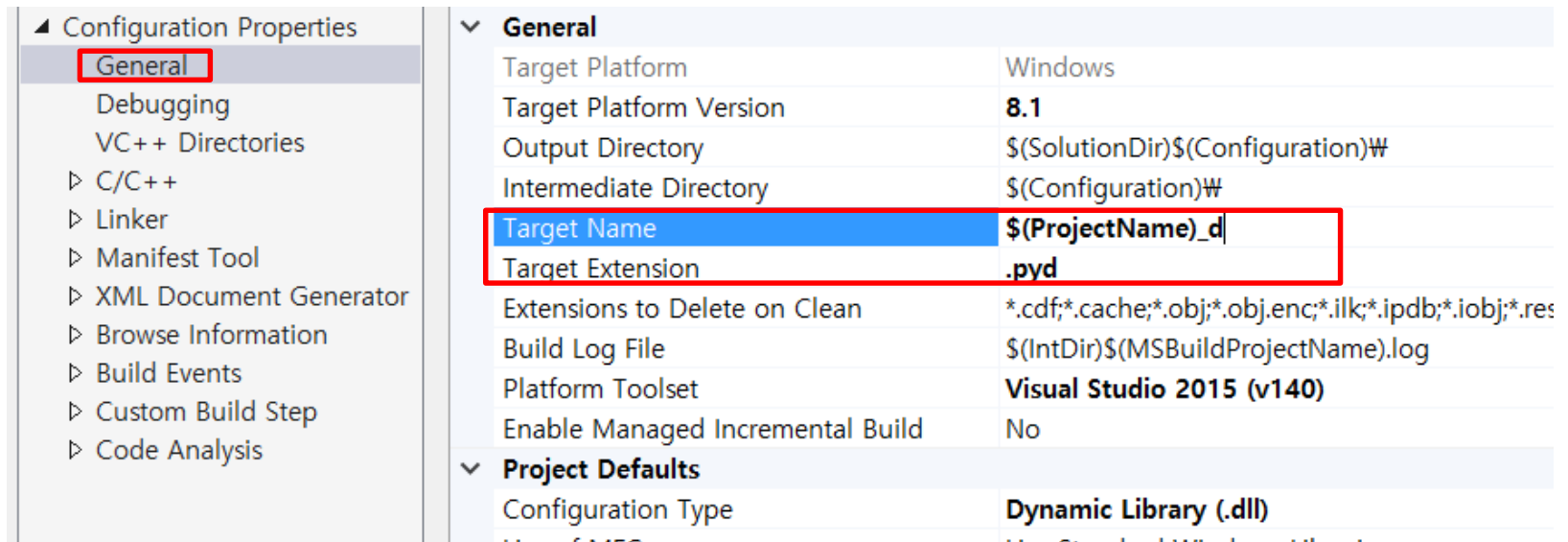
Application type:	Add common header files for:
<input type="radio"/> <u>W</u> indows application	<input type="checkbox"/> <u>A</u> TL
<input type="radio"/> <u>C</u> onsole application	<input type="checkbox"/> <u>M</u> FC
<input checked="" type="radio"/> <u>D</u> LL	
<input type="radio"/> <u>S</u> tatic library	
Additional options:	
<input checked="" type="checkbox"/> <u>E</u> mpy project	
<input type="checkbox"/> <u>E</u> xport symbols	
<input checked="" type="checkbox"/> <u>P</u> recompiled header	
<input type="checkbox"/> <u>S</u> ecurity Development Lifecycle (SDL) checks	

Visual studio for debug

Name:	<input type="text" value="evaluate.c"/>
Location:	<input type="text" value="C:\Users\User\Desktop\C_API_python\evaluate"/>

evaluate.c 생성

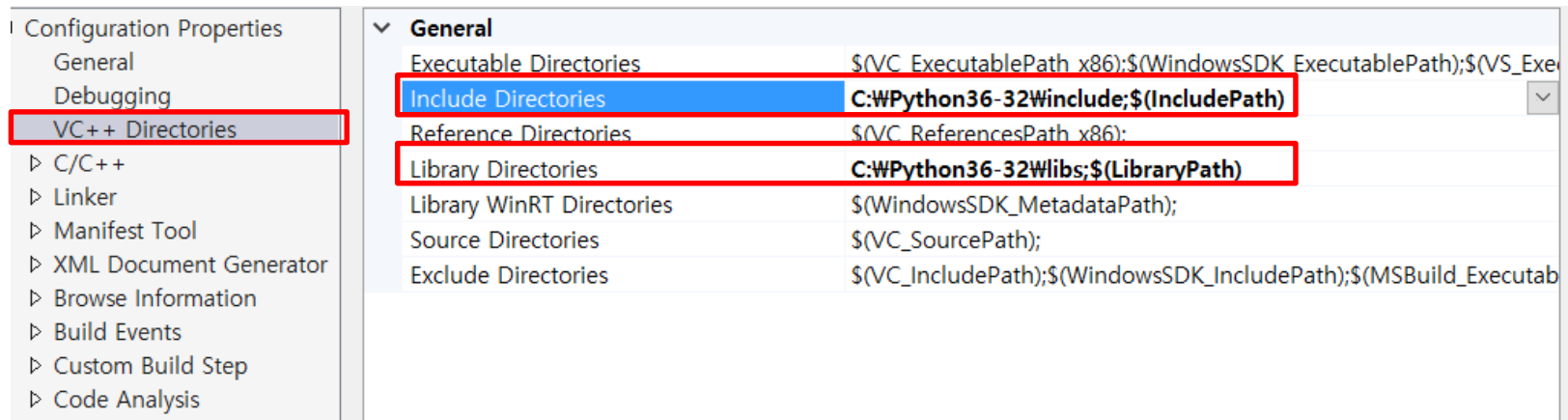
Visual studio for debug



The screenshot shows the 'Configuration Properties' dialog box in Visual Studio. The 'General' tab is selected and highlighted with a red rectangle. The 'Target Name' and 'Target Extension' fields are also highlighted with a red rectangle. The 'Target Name' field contains the text '\$(ProjectName)_d' and the 'Target Extension' field contains the text '.pyd'. The 'Platform Toolset' is set to 'Visual Studio 2015 (v140)'.

Configuration Properties	
General	
Target Platform	Windows
Target Platform Version	8.1
Output Directory	\$(SolutionDir)\$(Configuration)\#
Intermediate Directory	\$(Configuration)\#
Target Name	\$(ProjectName)_d
Target Extension	.pyd
Extensions to Delete on Clean	*.cdf;*.cache;*.obj;*.obj.enc;*.ilk;*.ipdb;*.iobj;*.res
Build Log File	\$(IntDir)\$(MSBuildProjectName).log
Platform Toolset	Visual Studio 2015 (v140)
Enable Managed Incremental Build	No
Project Defaults	
Configuration Type	Dynamic Library (.dll)

Visual studio for debug



Visual studio for debug

```
//python 함수에서 인자를 받아 c 자료형으로 바꾼 후 연산
//연산이 끝나면 다시 python 변수로 변환하여 반환
//실제 python 모듈은 이 함수를 호출한다
static PyObject * average(PyObject * self, PyObject * args)
{
    printf("avarage() in cpp is running \n");
    PyObject * scores;
    //한 반에 100명은 안 넘겠지.....
    int scoreArr[100];
    if (!PyArg_ParseTuple(args, "O", &scores))
    {
        printf("PyArg_ParseTuple() error! in average ");
        exit(-1);
    }
}
```

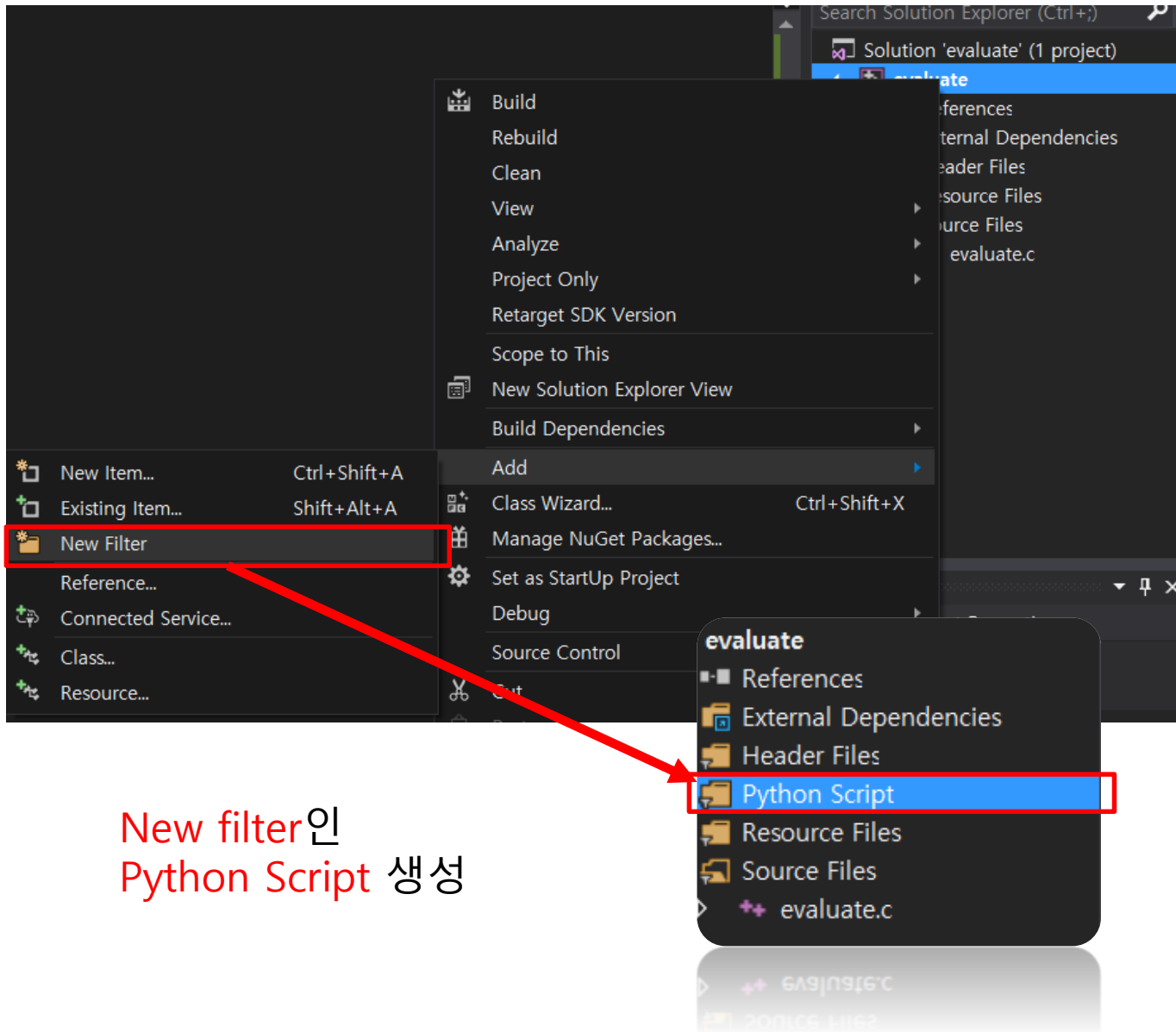
Python 함수를 작성 in C 파일

Visual studio for debug

```
output from: Build
C:\Users\User\Desktop\C_API_python\evaluate\evaluate.c(84): warning C4244: 'function': conversion from 'int' to 'double', possible loss of precision
Creating library C:\Users\User\Desktop\C_API_python\evaluate\Debug\evaluate_d.lib and object C:\Users\User\Desktop\C_API_python\evaluate\Debug\evaluate_d.pdb
evaluate.vcxproj -> C:\Users\User\Desktop\C_API_python\evaluate\Debug\evaluate_d.pdb
evaluate.vcxproj -> C:\Users\User\Desktop\C_API_python\evaluate\Debug\evaluate_d.pdb (Full PDB)
==== Build: 1 succeeded, 0 failed, 0 up-to-date, 0 skipped =====
```

Build 성공!!

Visual studio for debug



Visual studio for debug

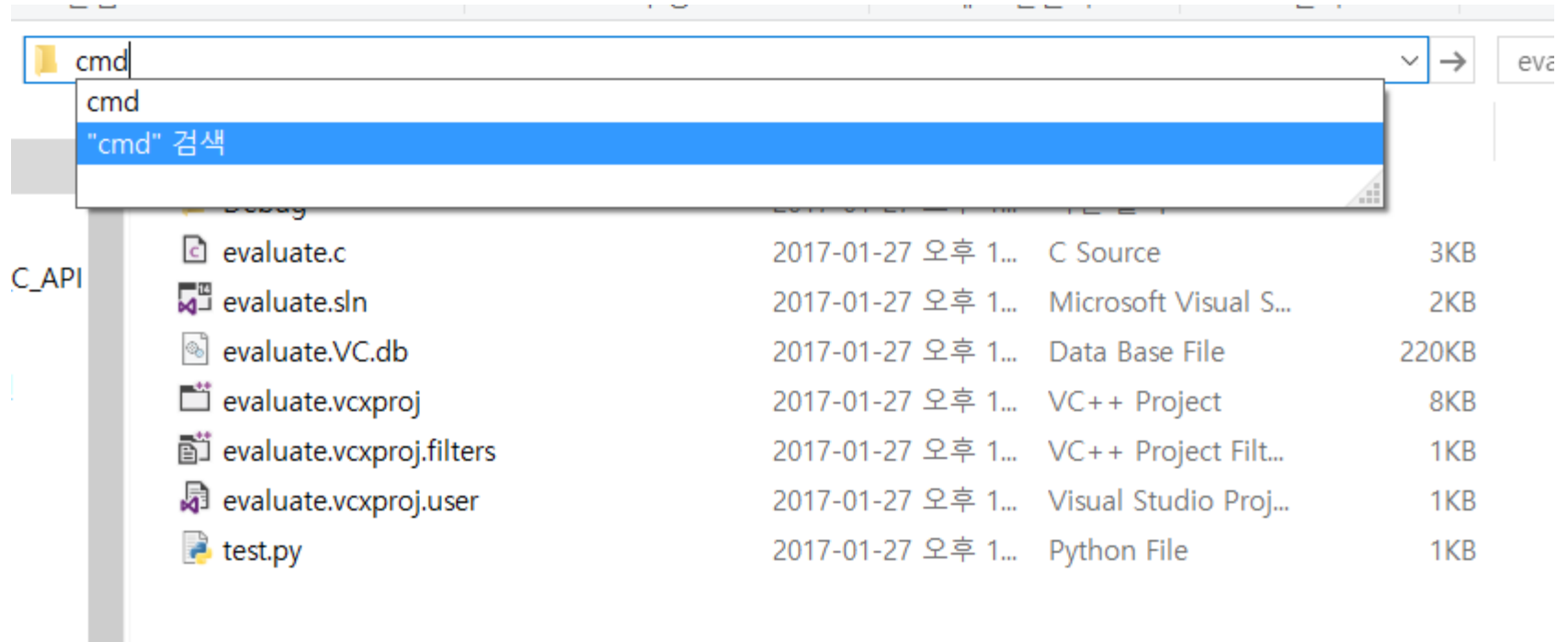
```
import evaluate

scores = [10, 15, 12, 13, 12]
avrg = evaluate.average(scores)
var = evaluate.variance(scores)

print(avrg, var)
```

Python script 안에 test.py 생성 후
테스트 코드 작성

Visual studio for debug



test.py가 있는 폴더로 가서 cmd 실행

Visual studio for debug

```
C:\Windows\System32\cmd.exe
Microsoft Windows [Version 10.0.14393]
(c) 2016 Microsoft Corporation. All rights reserved.

C:\Users\User\Desktop\WC_API_python\Wevaluate>set PYTHONPATH=Debug
C:\Users\User\Desktop\WC_API_python\Wevaluate>python_d test.py
avarage() in cpp is running
variance() in cpp is running!!!
12.399999618530273 2.799999952316284

C:\Users\User\Desktop\WC_API_python\Wevaluate>
```

환경변수 설정 -> 디버깅이 가능한 인터프리터인 python_d로
test.py 실행

Visual studio for debug

Debugging 환경에서 잘 작동하는 것을 확인했으니
이제 디버깅을 해볼 차례입니다!
(물론 잘 작동하기 디버깅이 필요 없지만)
Breakpoint를 통해 어떻게 작동하는지 살펴보죠!

Visual studio for debug

Configuration Properties

- General
- Debugging**
- VC++ Directories
 - ▷ C/C++
 - ▷ Linker
 - ▷ Manifest Tool
 - ▷ XML Document Generator
 - ▷ Browse Information
 - ▷ Build Events
 - ▷ Custom Build Step
 - ▷ Code Analysis

Debugger to launch:

Local Windows Debugger

Command	C:\Python36-32\python_d.exe
Command Arguments	test.py
Working Directory	\$(ProjectDir)
Attach	No
Debugger Type	Auto
Environment	PYTHONPATH=\$(OutputPath)\$(Loca
Merge Environment	Yes
SQL Debugging	No
Amp Default Accelerator	WARP software accelerator

Visual studio for debug

Environment ? ×

PYTHONPATH=\$(OutputPath)

Evaluated value:
PYTHONPATH=C:\Users\User\Desktop\WC_API_python\evaluate\Del

Inherited values:

☒ Inherit from parent or project defaults

DEBUG

Name	Value
\$(MpiDebuggerSchedul...	localhost/1
\$(MpiDebuggerSchedul...	5000
\$(OutDir)	C:\Users\User\Desktop\WC_API_python\Weva
\$(OutputPath)	C:\Users\User\Desktop\WC_API_python\Weva
\$(RemoteDebuggerAtta...	False
\$(RemoteDebuggerCon...	RemoteWithAuthentication
\$(RemoteDebuggerDeb...	Auto
\$(RemoteDebuggerDep...	true
\$(RemoteDebuggerServ...	DESKTOP-G8ER5AD

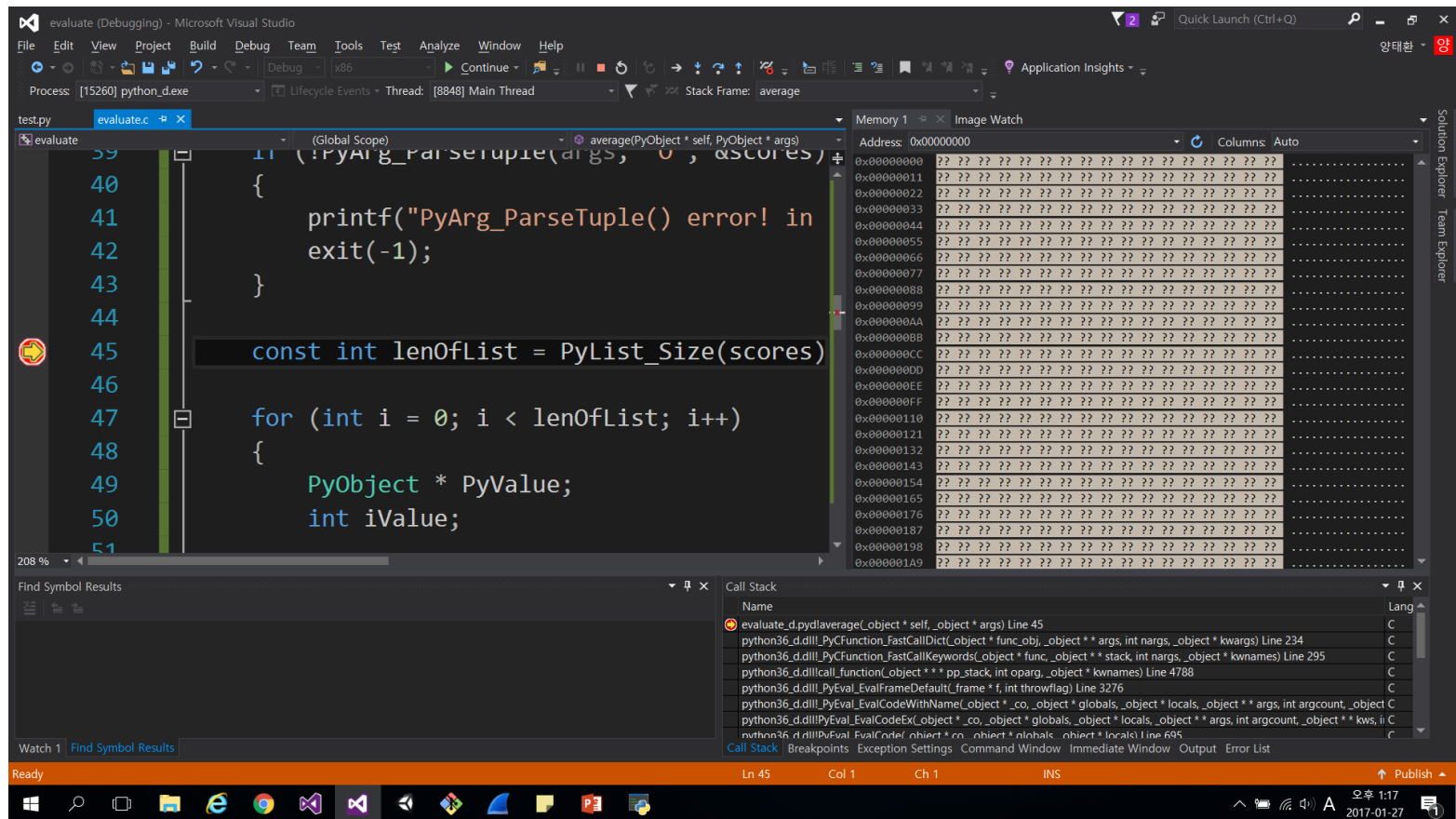
Macros <<

Insert OK Cancel

Visual studio for debug

```
38     int scoreArr[100],
39     if (!PyArg_ParseTuple(args, "0", &scores))
40     {
41         printf("PyArg_ParseTuple() error! in aver
42         exit(-1);
43     }
44
45     const int lenOfList = PyList_Size(scores);
46
47     for (int i = 0; i < lenOfList; i++)
```

Visual studio for debug

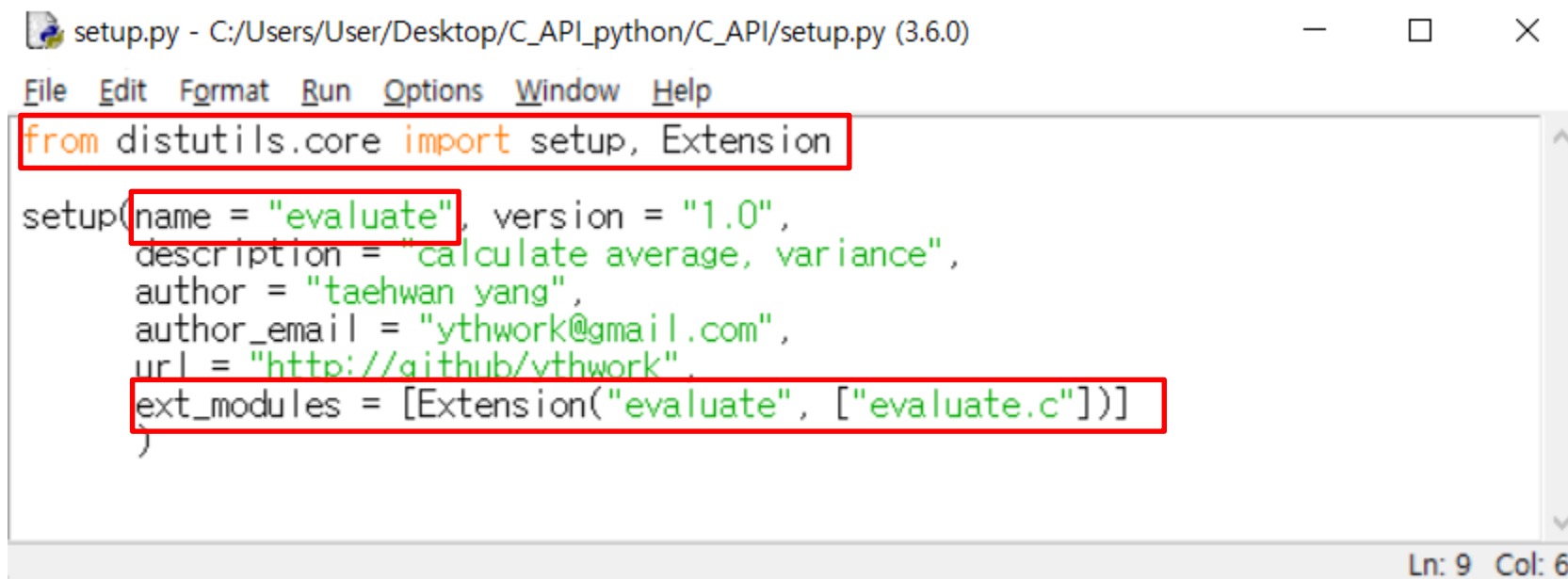


디버깅 성공!

C 확장 모듈

디버깅이 끝난 후
Data_analysis 프로그램에 실제로 적용해보죠!

C 확장 모듈



The screenshot shows a Python IDE window titled "setup.py - C:/Users/User/Desktop/C_API_python/C_API/setup.py (3.6.0)". The menu bar includes File, Edit, Format, Run, Options, Window, and Help. The code in the editor is as follows:

```
from distutils.core import setup, Extension

setup(name = "evaluate", version = "1.0",
      description = "calculate average, variance",
      author = "taehwan yang",
      author_email = "ythwork@gmail.com",
      url = "http://github/ythwork",
      ext_modules = [Extension("evaluate", ["evaluate.c"])]
)
```

Red boxes highlight the following parts of the code:

- The import statement: `from distutils.core import setup, Extension`
- The `name` parameter in the `setup` function: `name = "evaluate"`
- The `ext_modules` parameter in the `setup` function: `ext_modules = [Extension("evaluate", ["evaluate.c"])]`

The status bar at the bottom right indicates "Ln: 9 Col: 6".

setup.py 파일 작성

C 확장 모듈

C:\Windows\system32\cmd.exe

```
Microsoft Windows [Version 10.0.14393]  
(c) 2016 Microsoft Corporation. All rights reserved.
```









```
C:\Users\User\Desktop\C_API_python\C_API>python setup.py build_ext --inplace_
```

setup.py 빌드

C 확장 모듈

```
python36-32\PCbuild\win32 "/LIBPATH:C:\Program Files (x86)\Microsoft Visual Studio 14.0\VC\LIB" "/LIBPATH:C:\Program Files (x86)\Microsoft Visual Studio 14.0\VC\ATLMFC\LIB" "/LIBPATH:C:\Program Files (x86)\Windows Kits\10\Lib\10.0.10240.0\ucrt\x86" "/LIBPATH:C:\Program Files (x86)\Windows Kits\NETFXSDK\4.6.1\Lib\um\x86" "/LIBPATH:C:\Program Files (x86)\Windows Kits\8.1\Lib\winv6.3\um\x86" /EXPORT:PyInit_evaluate build\temp.win32-3.6\Release\evaluate.obj /OUT:C:\Users\User\Desktop\C_API_python\C_API\evaluate.cp36-win32.pyd /IMPLIB:build\temp.win32-3.6\Release\evaluate.cp36-win32.lib
    build\temp.win32-3.6\Release\evaluate.cp36-win32.lib 라이브러리 및 build\temp.win32-3.6\Release\evaluate.cp36-win32.exp 개체를 생성하고 있습니다.
코드 생성하고 있습니다.
코드 생성했습니다.
C:\Users\User\Desktop\C_API_python\C_API>
```

C 확장 모듈

 build	2017-01-27 오후 1...	파일 폴더	
 class_A.bin	2017-01-26 오후 2...	BIN 파일	1KB
 DataHandlerClass.py	2017-01-27 오전 6...	Python File	4KB
 evaluate.c	2017-01-27 오후 1...	C Source	3KB
 evaluate.cp36-win32.pyd	2017-01-27 오후 1...	Python Extension ...	10KB
 EvaluateClass.py	2017-01-10 오후 8...	Python File	1KB
 main.py	2017-01-26 오후 2...	Python File	1KB
 setup.py	2017-01-27 오후 1...	Python File	1KB

모듈 생성

C 확장 모듈

```
import evaluate

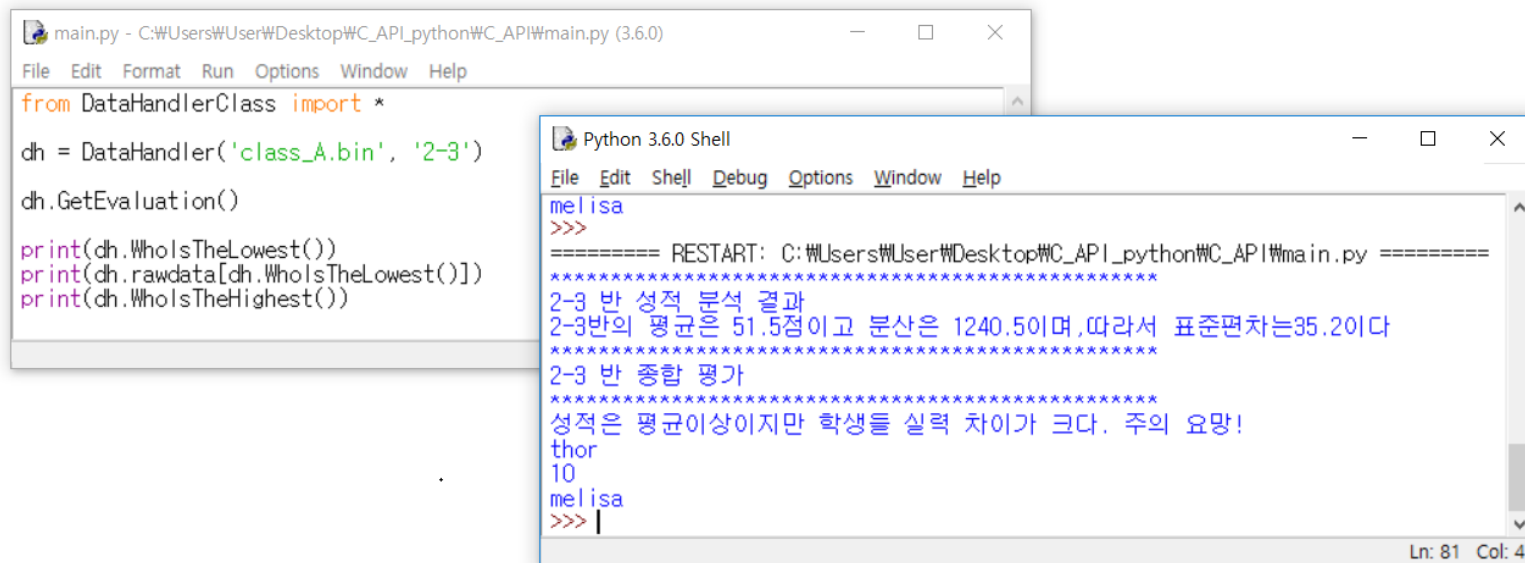
class Evaluate:
    def average(self, scores):
        return evaluate.average(scores)

    def variance(self, scores):
        return evaluate.variance(scores)

    def evaluateClass(self, avrg, std_dev):
        if avrg < 50 and std_dev > 20:
            print("성적이 너무 저조하고 학생들의 실력 차이가 너무 크다.")
        elif avrg > 50 and std_dev > 20:
            print("성적은 평균이상이지만 학생들 실력 차이가 크다. 주의 요망!")
        elif avrg < 50 and std_dev < 20:
            print("학생들간 실력차는 나지 않으나 성적이 너무 저조하다. 주의 요망")
        elif avrg > 50 and std_dev < 20:
            print("성적도 평균 이상이고 학생들의 실력차도 크지 않다.")
```

이제 실제 연산은 C 코드에 맡기시다!

C 확장 모듈



The image shows two windows. The top window is a text editor titled 'main.py - C:\Users\User\Desktop\C_API_python\C_API\main.py (3.6.0)'. It contains the following Python code:

```
from DataHandlerClass import *  
  
dh = DataHandler('class_A.bin', '2-3')  
dh.GetEvaluation()  
  
print(dh.WhoIsTheLowest())  
print(dh.rawdata[dh.WhoIsTheLowest()])  
print(dh.WhoIsTheHighest())
```

The bottom window is a 'Python 3.6.0 Shell' showing the execution output. It starts with a restart message and then displays the results of the script's execution:

```
=====  
*****  
2-3 반 성적 분석 결과  
2-3반의 평균은 51.5점이고 분산은 1240.5이며,따라서 표준편차는35.2이다  
*****  
2-3 반 종합 평가  
*****  
성적은 평균이상이지만 학생들 실력 차이가 크다. 주의 요망!  
thor  
10  
melisa  
>>> |
```

The status bar at the bottom right of the shell window indicates 'Ln: 81 Col: 4'.

잘 작동하네요!!