Python Programming

Python 시작하기

파이선 환경설치
파이썬 인터프리터 사용 사용하기

Yoonjoong Kim
Department of Computer Engineering, Hanbat National University
yjkim@hanbat.ac.kr

내용

1. 파이썬(Python) 개요
2. Visual Studio 2019 설치
3. Python Project 생성
4. Python 실행환경
    1. 개요
        - 배포자 웹사이트, 대화형창 열기, PowerShell에서 열기, 환경저장 위치
    2. 패키지
        - 패키지(라이브러리) 설치
5. 파이썬(Python) 인터프리터 실행
    1. IDLE Python
    2. PowerShell Python
    3. 대화형 Python
6. 파이썬 응용프로그램 맛보기
    1. Python Application 프로젝트 “PythonExamples” 생성
    2. Python 모듈(파일)에서 코드 작성 맛보기
    3. 대화형창에서 코드실행
    4. 파일썬의 기본연산
    5. 모듈사용법
7. 연습문제

1. Python 개요

- 1991년에 발표된 인터프리터 방식의 프로그래밍 언어. 공식홈페이지[link], 공식 github저장소[link]
- 창시자는 귀도 반 로섬(Guido van Rossum). 네덜란드어 발음으로 '히도 판 로쉼' 1989년 크리스마스 주에 "연구실이 닫혀 있어서 심심한 김에 만들었다." 농담이 아니고 반 로섬을 유럽에서는 애덤 스미스에 비견할 정도다. 2000년에는 Python 2, 2008년에는 Python 3가 나왔다.
- 대학교 프로그래밍 교육은, 예전에는 한국을 포함하여 대부분의 학교에서 C, C++나 Java 같은 실제 회사에서 많이 쓰는 언어 위주로 수업을 개설했으나, 2019년, 서울대학교가 기초 프로그래밍 수업을 Python으로 대체(기존에는 Java였다(...))하였다.
- 미국 대학들은 개론 수업 언어를 Python으로 옮겨가는 추세다. UC 버클리 컴퓨터과학 개론수업인 CS 61A를 인터넷 강의로 들어보자! 참고로, 연세대학교,카이스트, 성균관대학교 부산대학교, 국민대학교,서강대학교, 경희대학교, 상명대학교, 아주대학교, 인천대학교, 한양대학교 ERICA 캠퍼스, 한국외국어대학교, 세종대학교, 경북대학교에서는 프로그래밍 개론 수업을 Python으로 진행한다.
-

https://namu.wiki/w/Python

1. Python 개요 (cont.)

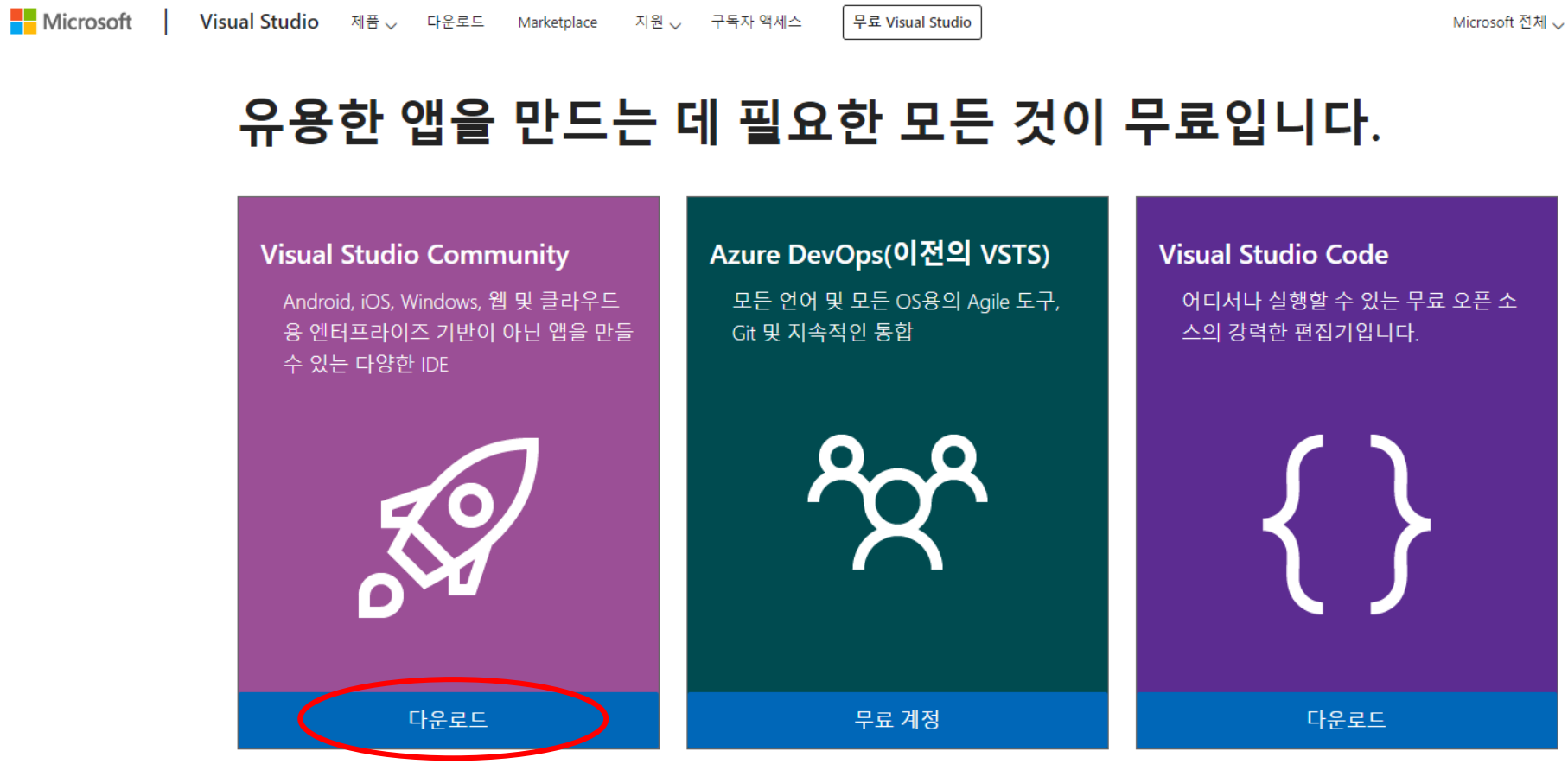
- Python의 구현 종류, Python.org
  - **CPython** : c언어로 구현된 표준 Python, **Jython** : JVM용 Python
  - **IronPython** : .NET framework
    - Microsoft .NET Framework의 가상머신인 CLR 상에서 구현된 DLR이라는 프레임워크 기반이다.
    - 제작자 Jim Hugunin은 Jython의 제작자이며, NumPy의 전신인 Numeric 라이브러리의 제작자이기도 하다.
    - .NET Framework 환경에서 제작된 DLL과 결합성이 매우 좋다. Jython과 마찬가지로 병렬 프로그래밍 환경에서 GIL 때문에 고민할 필요가 없다.
  - **Python for .NET**
    - Visual Studio에서 실행, Managed code (IL가 아니다. CPython (기반)위에서 돌라간다.
    - This approach allows you to use CLR services and continue to use existing Python code and C-API extensions while maintaining native execution speeds for Python code.

- 라이브러리
  - 수학 : SymPy, NumPy, SciPy
  - 데이터 : Pandas, Dask, StatsModels
  - 그래프챠트그리기 : matplotlib
  - Machine learning : Tensorflow, Keras, scikit-learn, PyTorch
  - GUI programming : Tkinter, wxPython, PyQt, PySide[28], PyGTK, SFML,eel
- 개발환경
  - Visual Studio IDE
    - Text editor, interpreter …

https://namu.wiki/w/Python

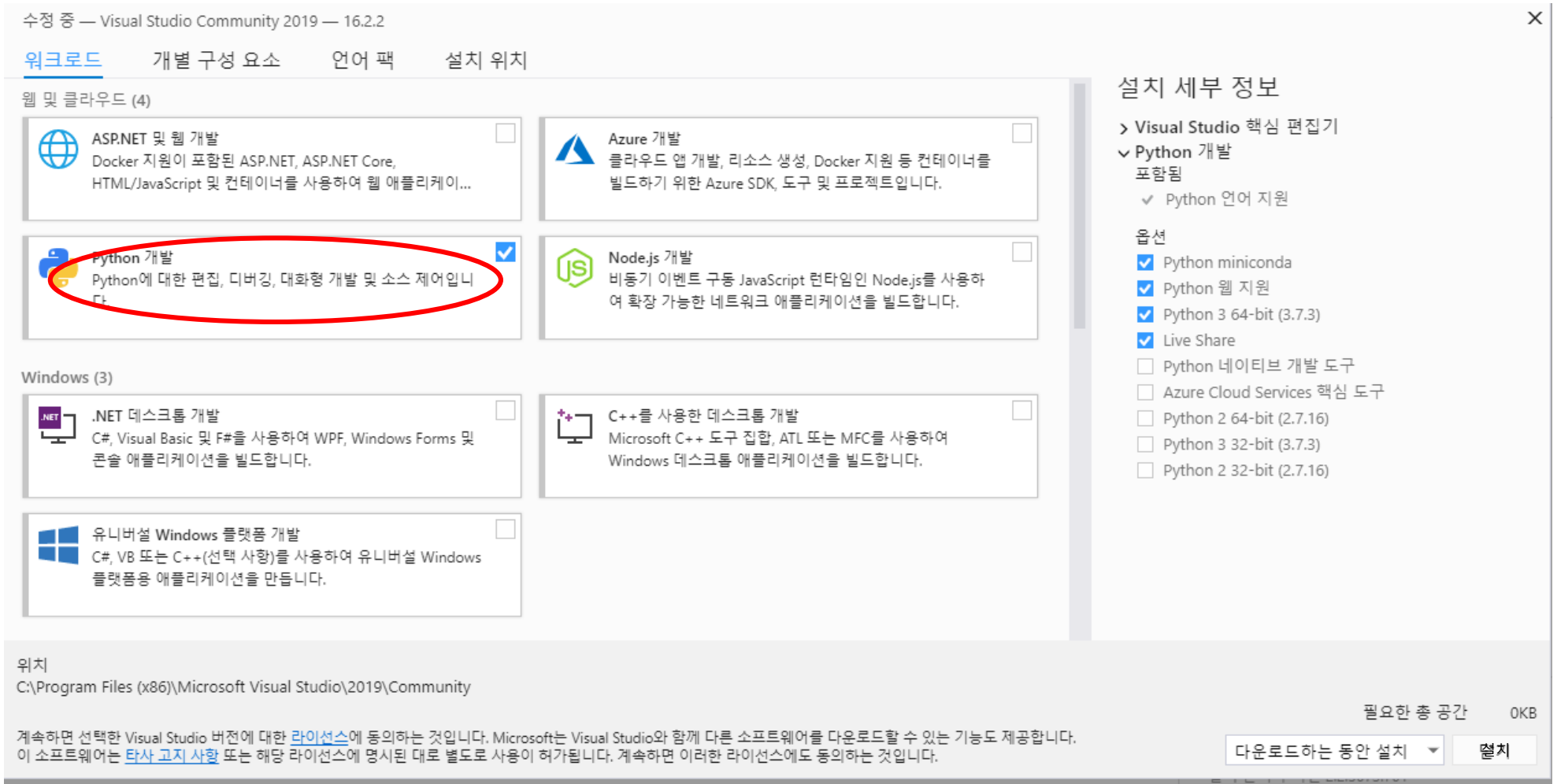
2. Visual Studio 2019 설치

- Microsoft Home page에 가서 [Visual Studio Community] 를 다운로드 하여 설치한다.



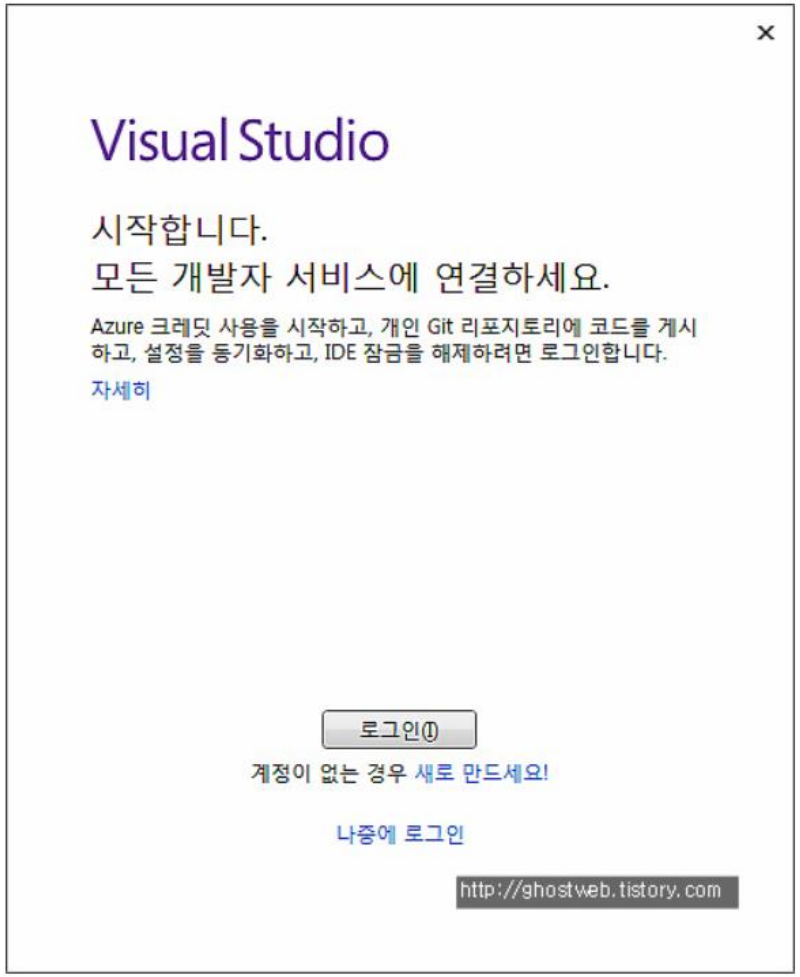
2. Visual Studio 2019 설치 (cont.)

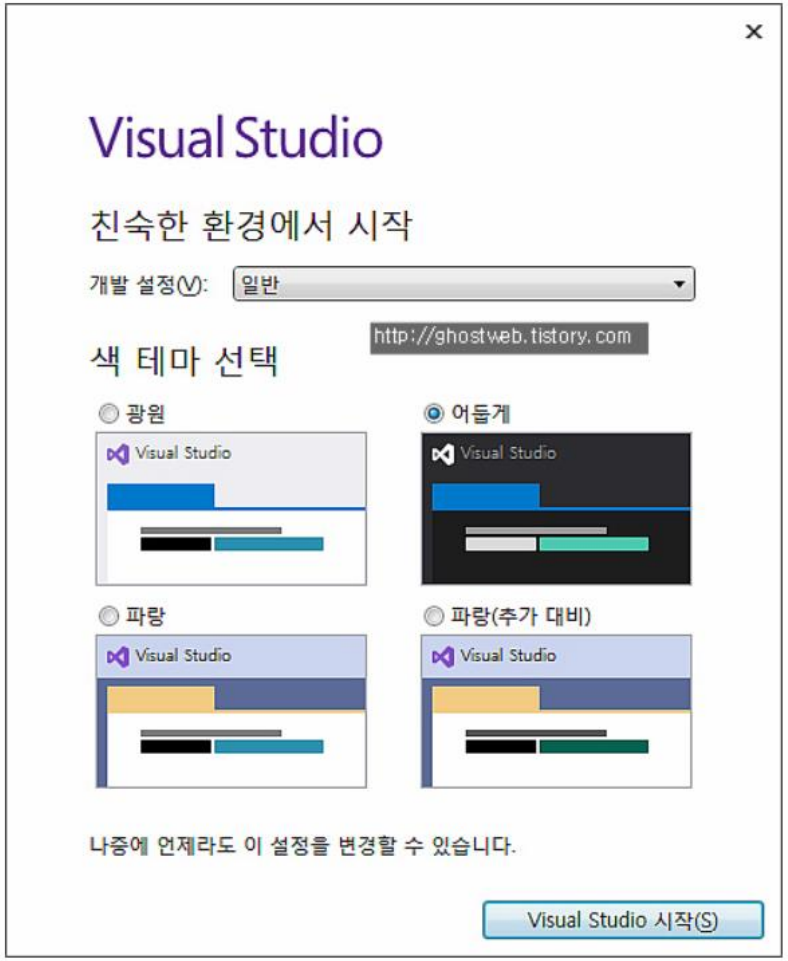
- 설치 과정 중에 개발환경구축(Visual Studio Installer)이 나오는데 <Python 개발>체크한다. (나중에 변경 도가능)



2. Visual Studio 2019 설치 (cont.)

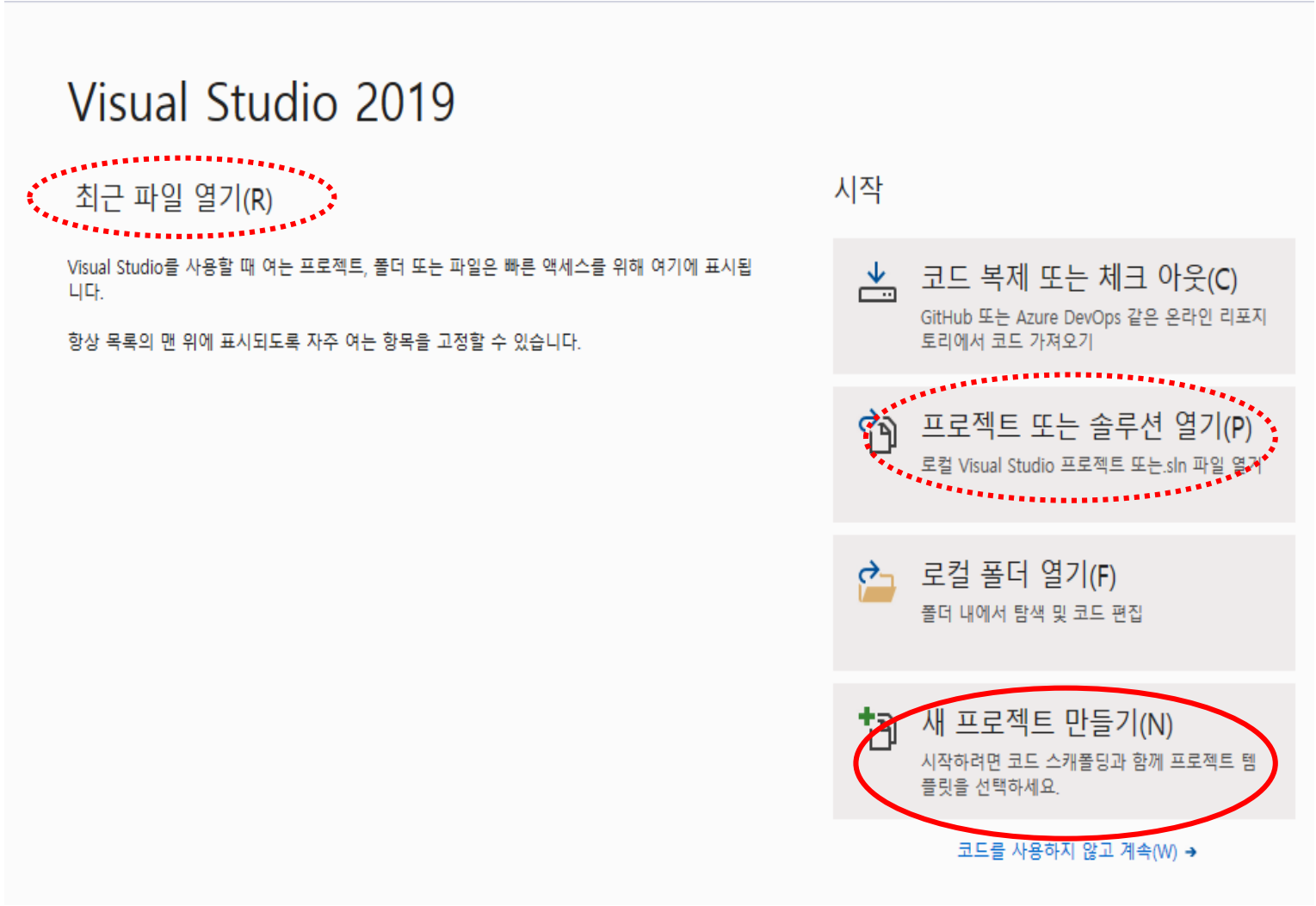
- 설치가 완료되면 비주얼 스튜디오가 실행된다.
  마이크로소프트계정에 로그인하고 색테마를 선택한다.





3 Python Project의 생성

- Visual Studio 2019 실행
- [Visual Studio 2019] IDE
  - 최근파일열기 또는
  - 시작 방법 선택
    - **<새프로젝트 만들기>** 선택
    - 코드사용하지 않고 계속

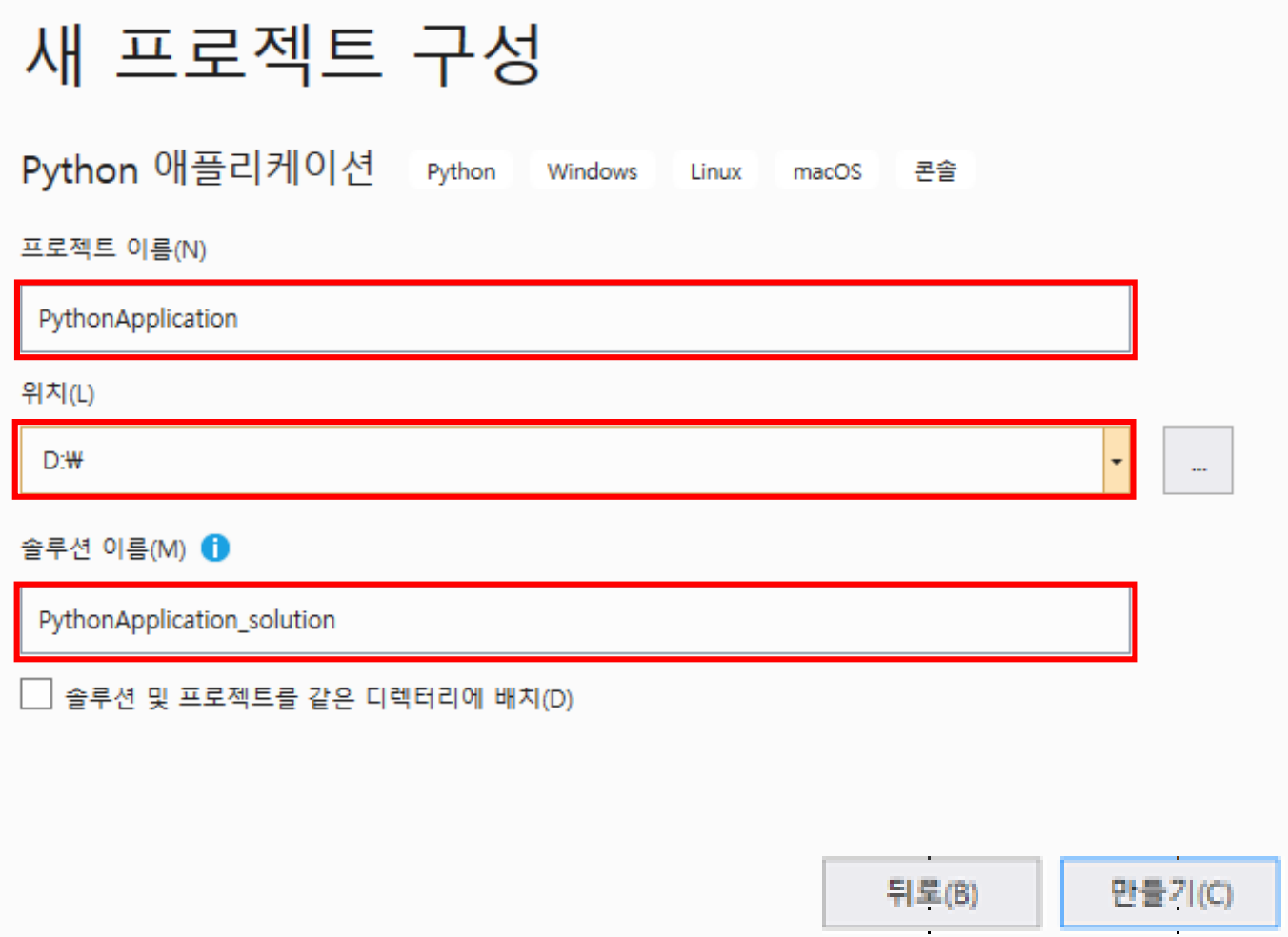


3 Python Project의 생성 (cont.)

- Visual Studio 2019 실행
- [Visual Studio 2019] IDE
  - 최근파일열기 또는
  - 시작 방법 선택
    - <새프로젝트 만들기> 선택
- **[새프로젝트 만들기]**
  - 최근 프로젝트 템플릿
    - Python 애플리케이션
  - **Python 애플리케이션** 선택
  - **<다음>**

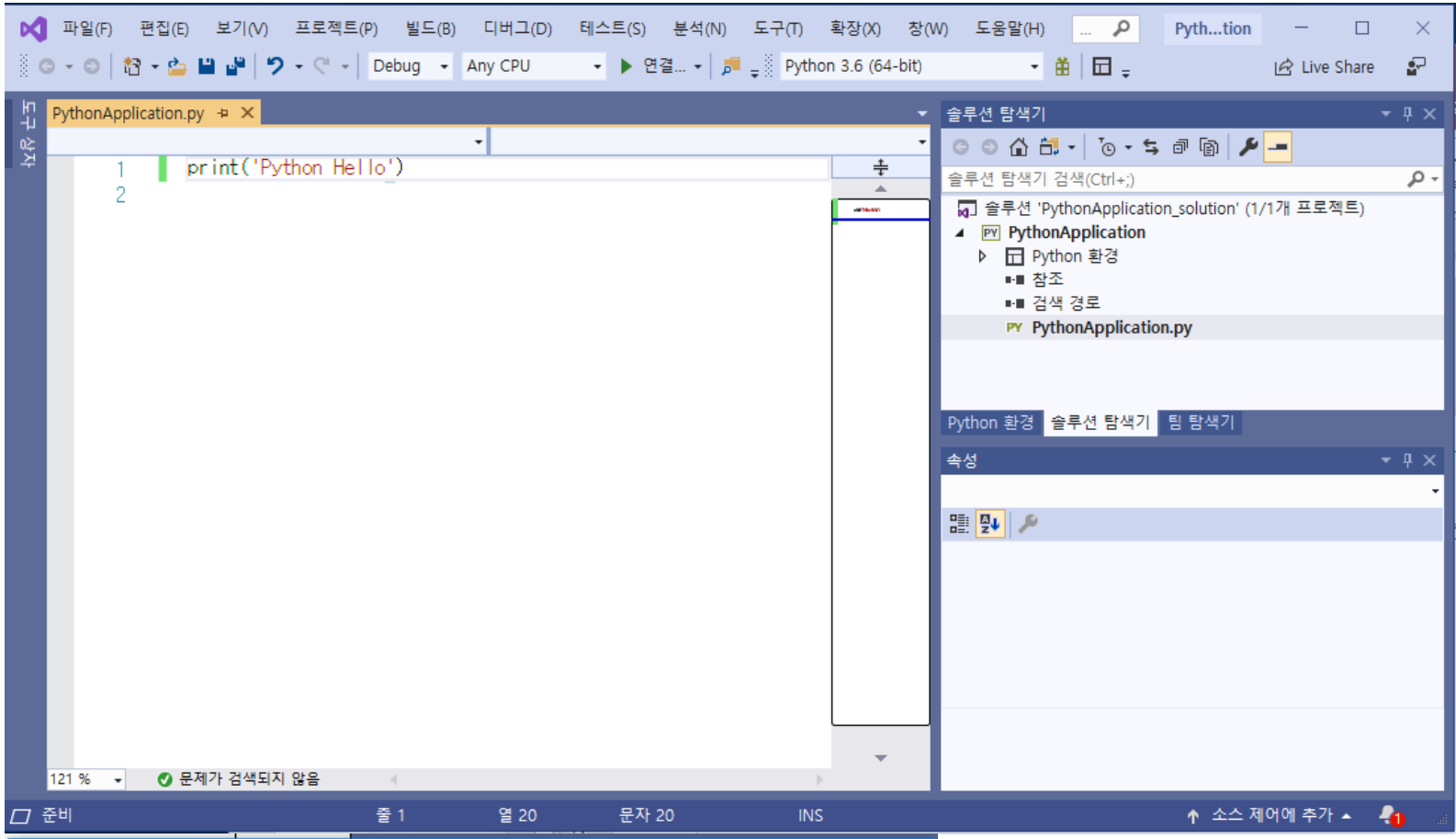
3 Python Project의 생성 (cont.)

- 비주얼스튜디오를 실행
- [Visual Studio 2019]
  - 최근파일열기 또는
  - 시작 방법 선택
    - <새프로젝트 만들기> 선택
- **[새프로젝**트 만들기]
  - **Python 애플리케이션** 선택
- [새프로젝트 구성]
  - 프로젝트이름 : PythonApplication
  - 위치 : D\
  - 솔루션이름 : PythonApplication_Solution
  - <만들기>



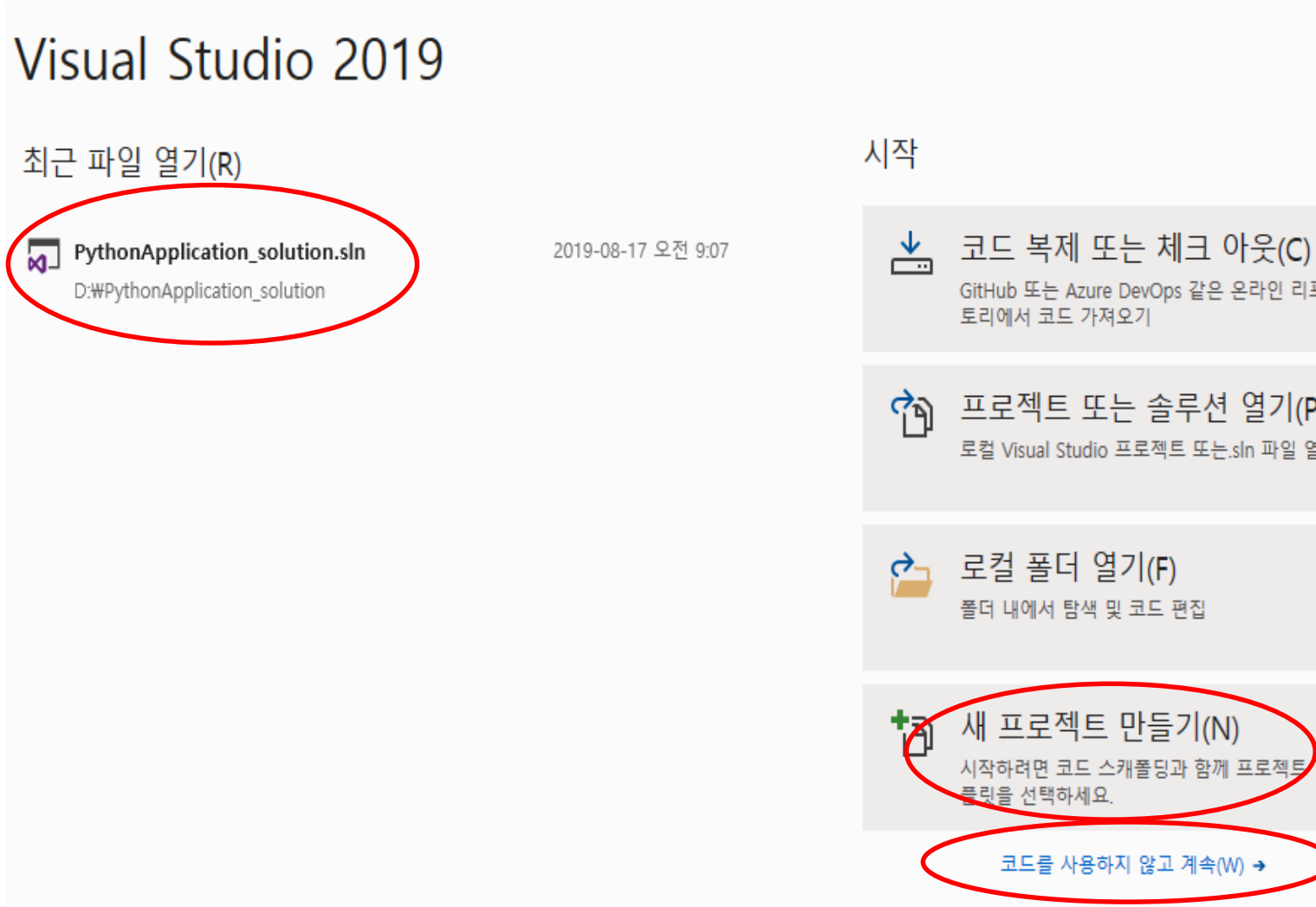
3 Python Project의 생성 (cont.)

- 화면구성
  - [솔루션창]
    PythonApplication_Solution
    - (프로젝트)
      PythonApplication
      - Python 환경
      - 참조
      - PythonApplication.py
  - Python 환경
  - [편집창]
  - [속성창]
- 테스트용 python 코드 작성 및 실행 하기
  - 프로젝트 “PythonApplication”에서 모듈 “PythonApplication.py”을 클릭
  - 편집창에서 코드 작성
    print(‘Hello Python !!’)
  - <시작>(<연결>, CTRL+F5)을 클릭하여 모듈의 코드 실행
  - 콘솔창에 실행결과가 출력된다.



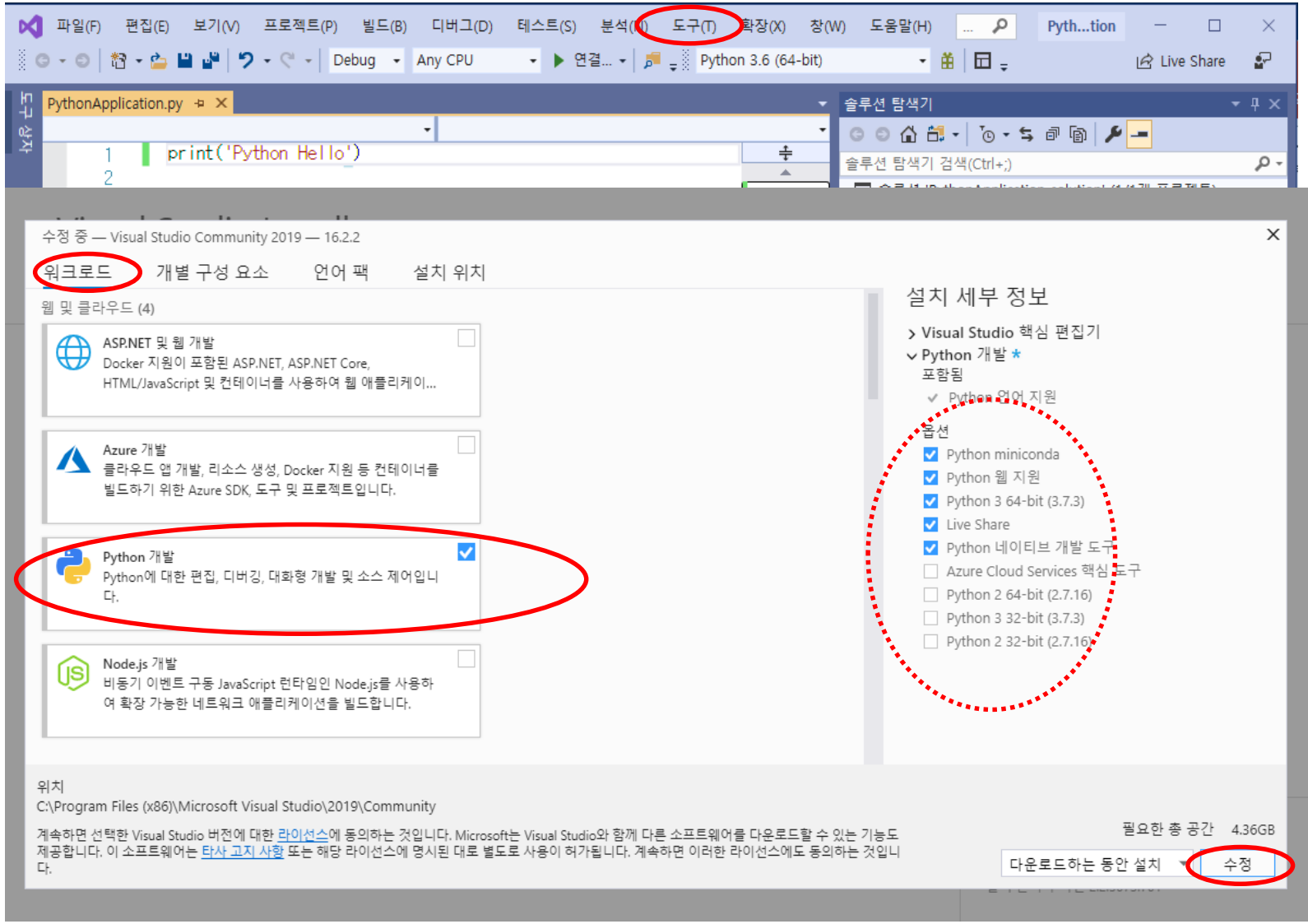
4. Python 환경 - Visual Studio Installer 열기

- Visual Studio Installer 열기
  - Visual Studio 2019 (관리자권한으로) 실행
  - Visual Studio 2019
    - 최근파일 열기 또는
      - PythonApplication_solution.sln
    - 새프로젝트 만들기
    - **코드를 사용하지 않고 계속**
  - Visual Studio 2019 IDE
    - <도구>/<도구 및 기능 가져오기>
- 워크플로우 Python 설정
  - Visual Studio Installer /**워크플로우**
    - [Python 개발] 선택 옵션선택
      - Python 3 64-bit(3.7.3)
    - [수정] 클릭



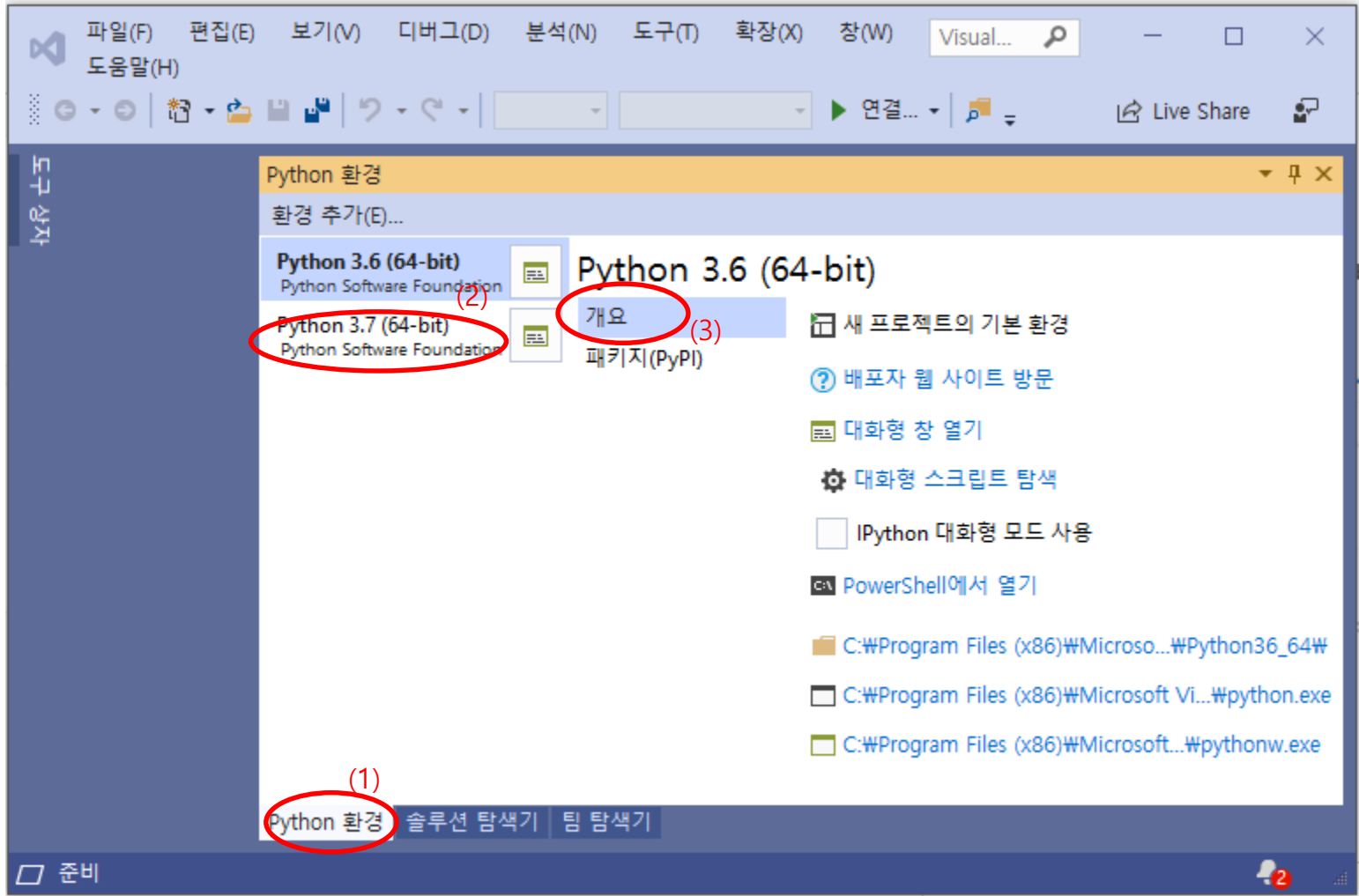
4. Python 환경

- Python 실행환경 만들기
  Python 3 64-bit(3.7.3)
- Visual Studio Installer 열기
  - Visual Studio 2019
    (관리자권한으로) 실행
  - Visual Studio 2019
    - 최근파일 열기 또는
      - PythonApplication_solution.sln
    - 새프로젝트 만들기
    - **코드를 사용하지 않고 계속**
  - Visual Studio 2019 IDE
    - <도구>/<도구 및 기능 가져오기>
- Visual Studio Installer
  - **워크플로우**
  - [Python 개발] 선택
  - 옵션선택
    - Python 3 64-bit(3.7.3)
  - [수정] 클릭



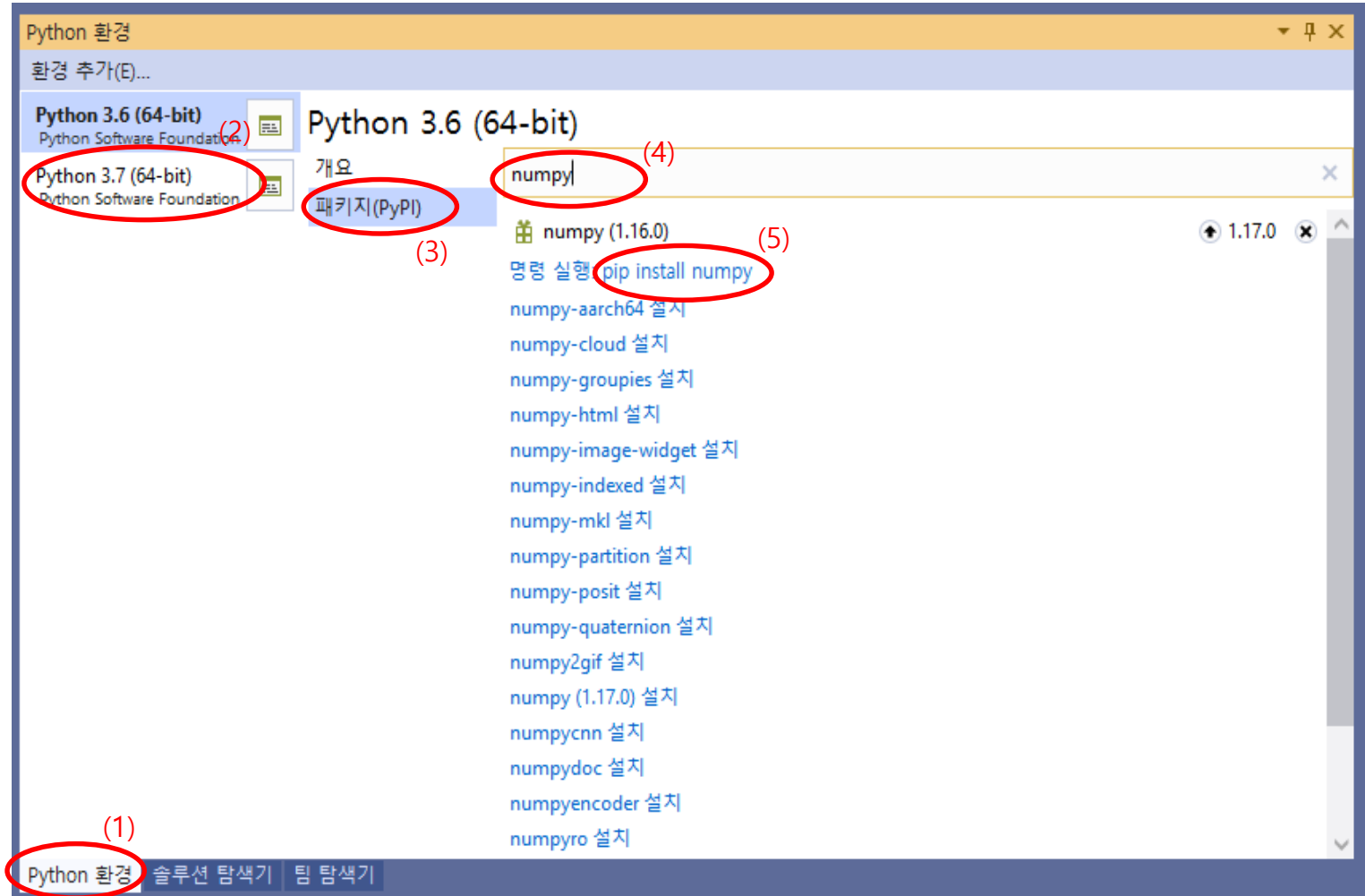
4. Python 환경 – 4.1 개요

- Python 환경열기(1-2)
  - 솔루션 창에서 [Python환경] 탭 클릭
  - Python 3.6(64bit)
  - Python 3.7(64bit)
- 4.1 개요 (3)
  - 배포자 웹사이트
  - 대화형창 열기
  - PowerShell에서 열기
  - 환경저장 위치
  - C:\Program Files (x86)\Microsoft Visual Studio\Shared\Python36_64
  - C:\Program Files (x86)\Microsoft Visual Studio\Shared\Python36_64\python.exe
  - C:\Program Files (x86)\Microsoft Visual Studio\Shared\Python36_64\pythonw.exe
- 4.2 패키지



4. Python 실행환경 4.2패키지

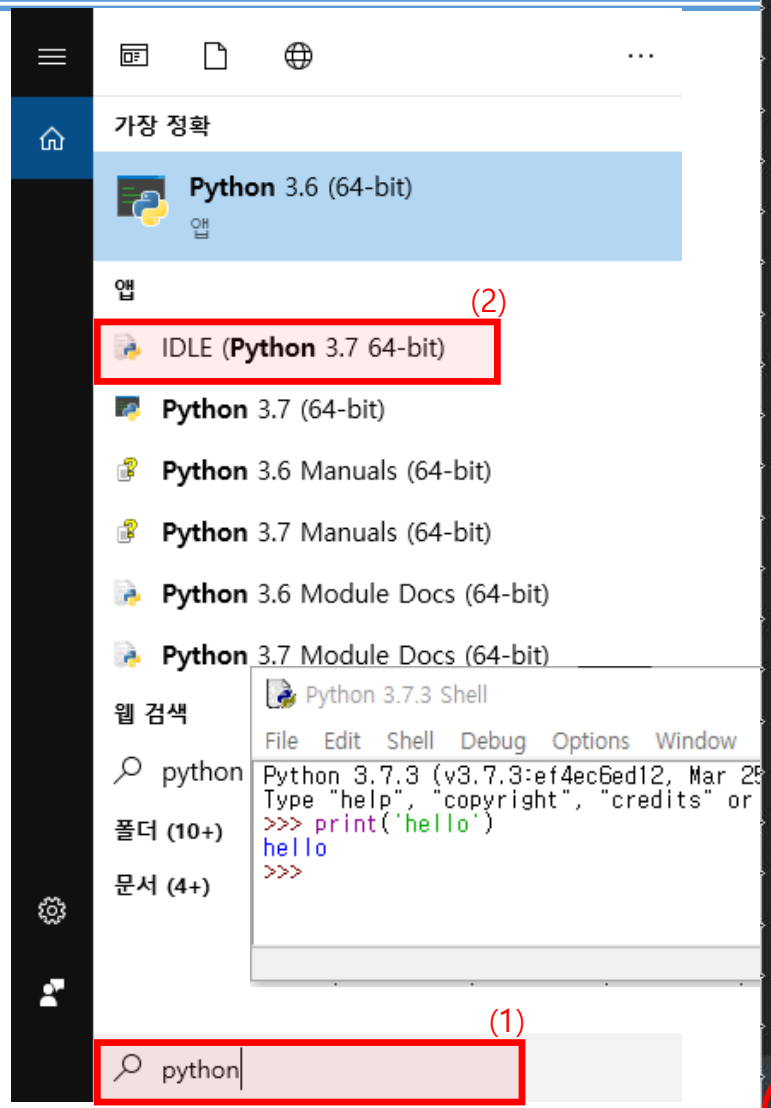
- Python 환경 열기(1-2)
  - Python 3.6(64bit)
  - Python 3.7(64bit)
- 4.1 개요
- 4.2 패키지(3)
  - 패키지(라이브러리)다운로드
    - 패키지 이름 타입(4)
      'numpy'
    - 명령실행 클릭(5)
      'pip install numpy'

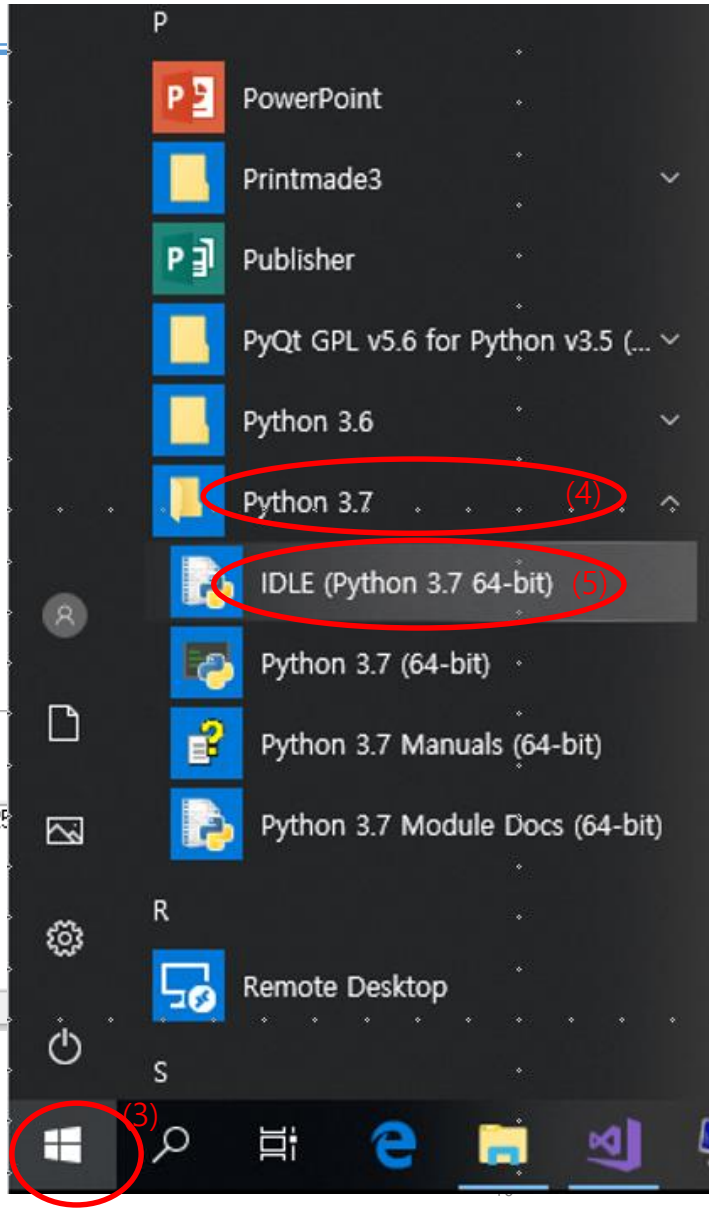


5. Python interpreter 실행 - 5.1 IDLE Python

- Interpreter 종류
  1. IDLE Python 3.7(64-bit)
  2. PowerShell Python 3.7(64-bit)
  3. 대화형창열기 Python 3.7(64-bit)

- 5.1 IDLE Python 3.7(64-bit)
  - 검색창에서 (1-2)
    - Python 타입 (1)
    - IDLE Python 3.7(64-bit) 클릭 (2)
  - 시작 버튼에서 (3-5)
    - Python 3.7
    - IDLE Python 3.7 (64-bit) 클릭

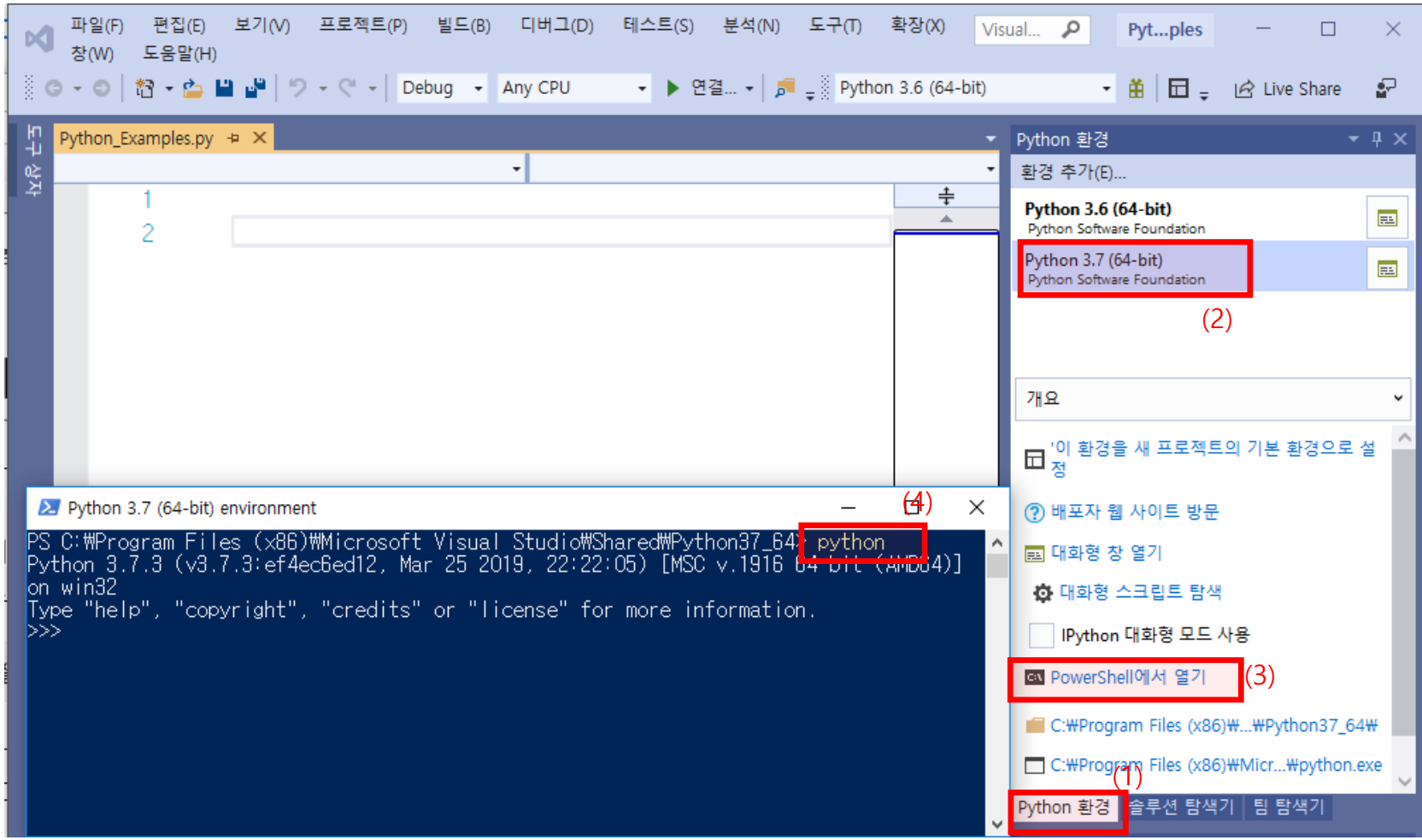




5.2 PowerShell Python

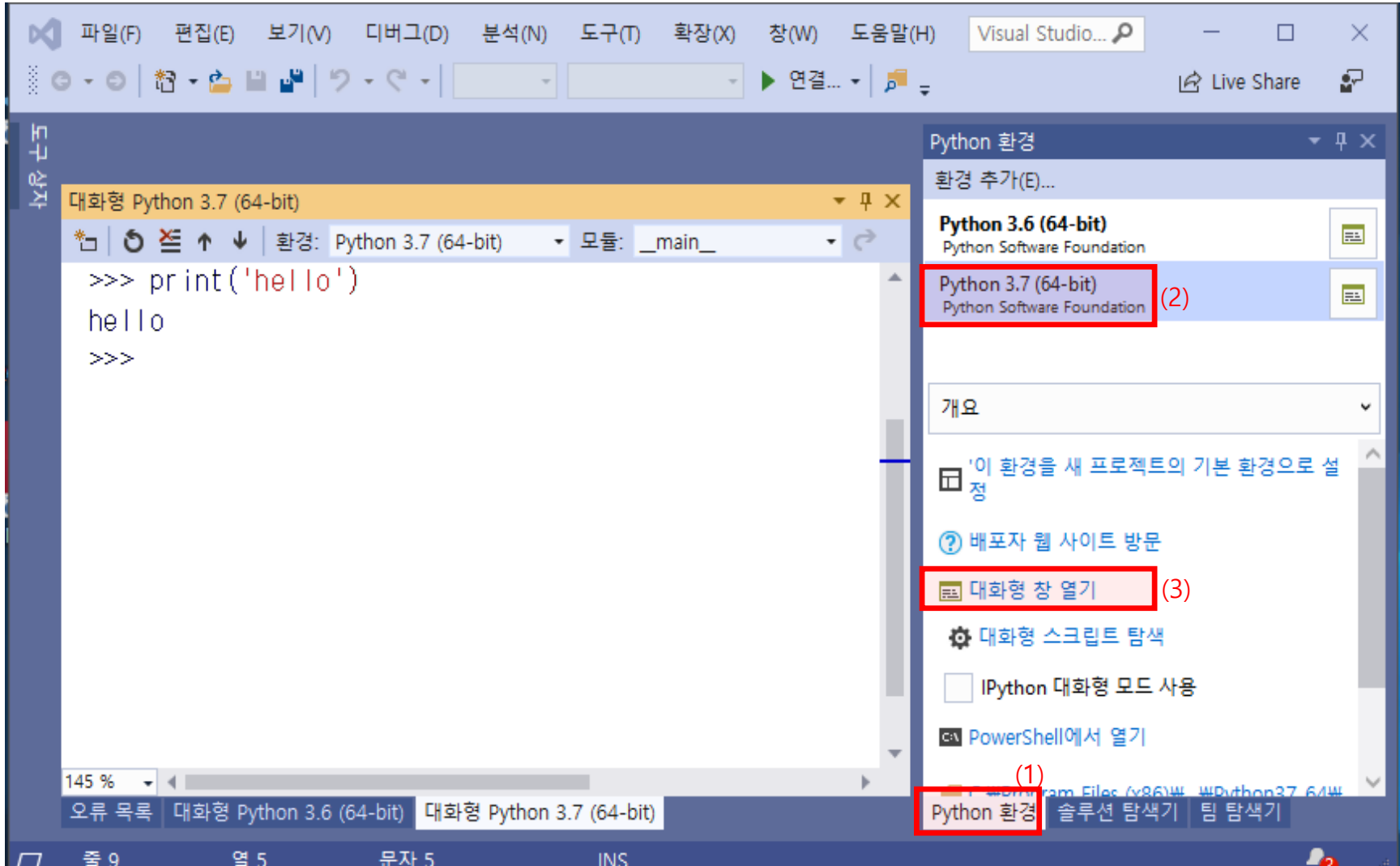
- 5.2 PowerShell Python 3.7(64-bit) 인터프리터 환경열기
  - Visual Studio 2019 실행
  - 코드를 사용하지 않고 계속
  - Python 환경
  - Python 3.7(64-bit)
  - Pworshell에서 열기

  - Python 3.7(64-bit) environment
    - python 타입



5.3 대화형창 Python

- 5.3 대화형창열기 Python 3.7(64-bit)
  - Visual Studio 2019 실행
  - 코드를 사용하지 않고 계속
  - Python 환경
  - Python 3.7(64-bit)
  - 대화형창 열기

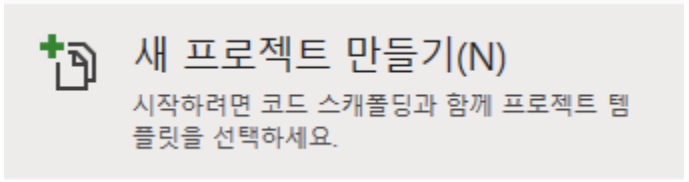


6. Python 프로그램 맛보기

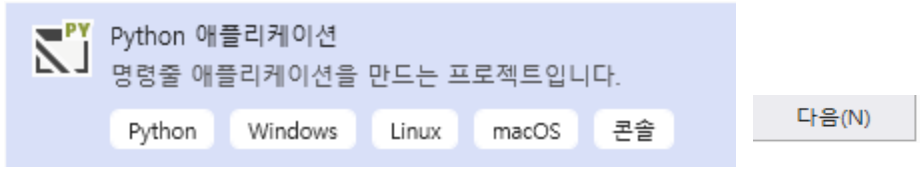
6.1 Python Application 프로젝트 PyhonExamples 생성

- Visual Studio 2019 IDE 실행

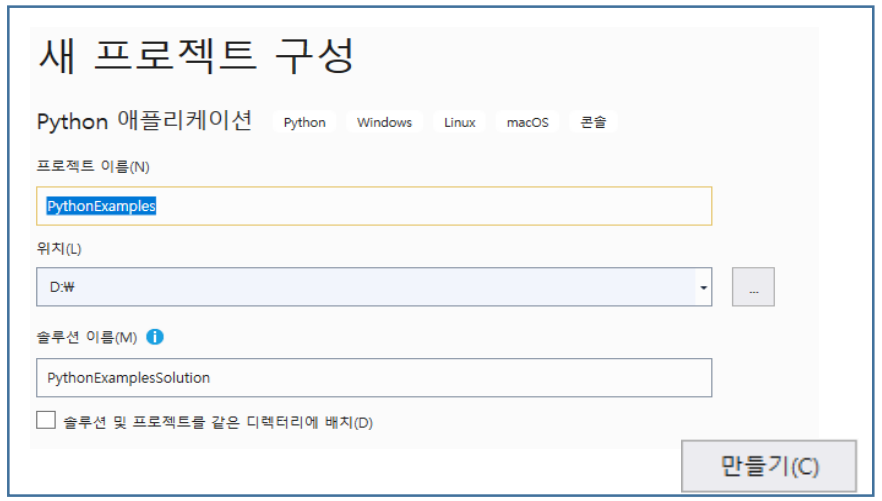
- [Visual Studio 2019]
  - <새프로젝트 만들기>



- [새프로젝트 만들기]
  - <Python 애플리케이션> <다음>

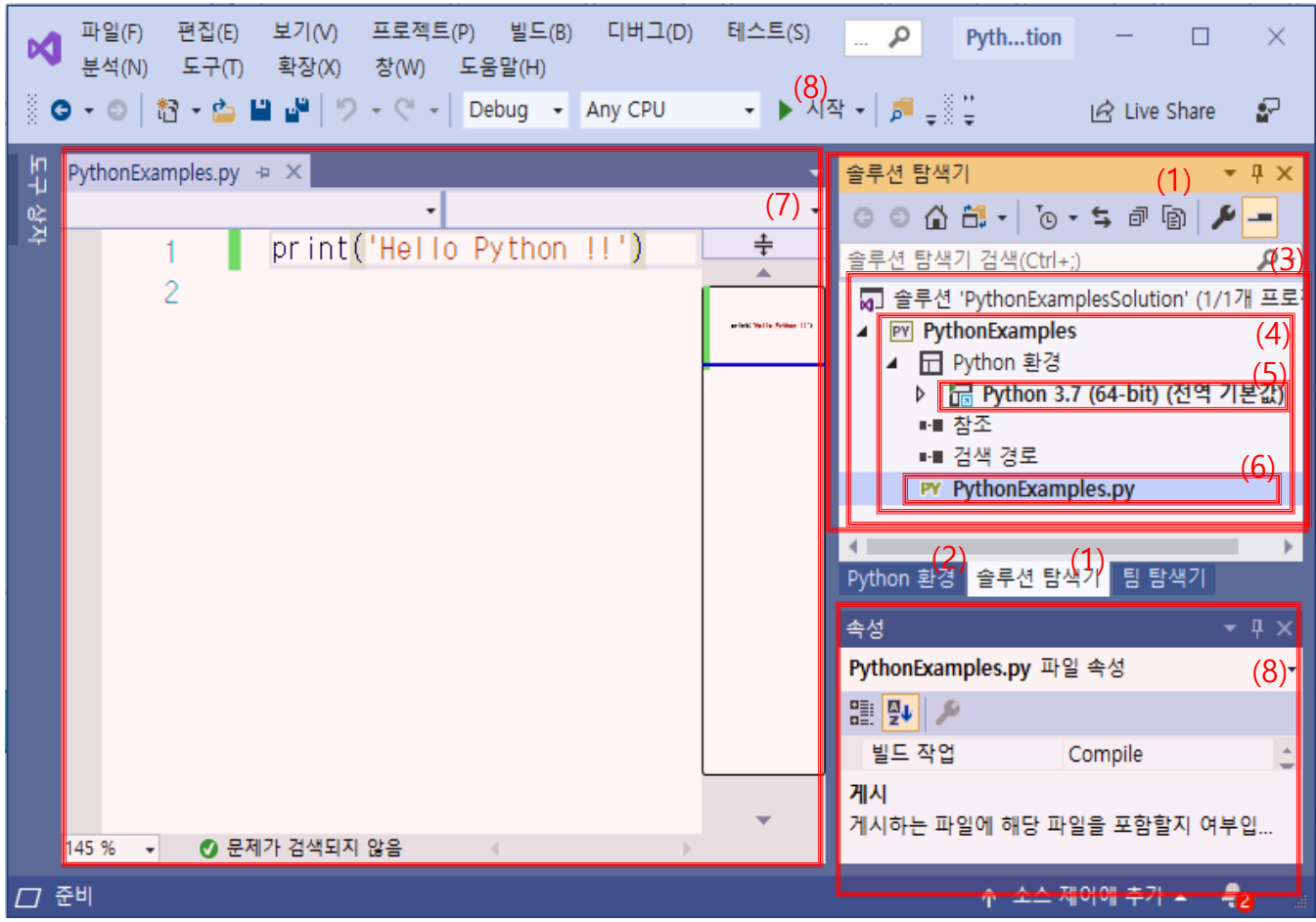


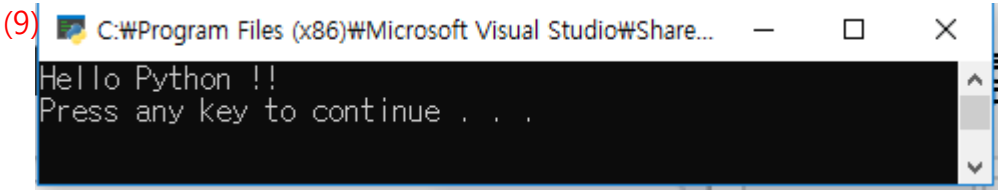
- [새프로젝트 구성]
  - 프로젝트이름 : PyhonExamples
  - 위치 : D:
  - 솔루션 (옵션) 선택
  - 솔루션이름 : PythonExamplesSolution
  - <만들기>



6.2 Python 모듈(파일)에서 코드 작성 맛보기(Hell Python)

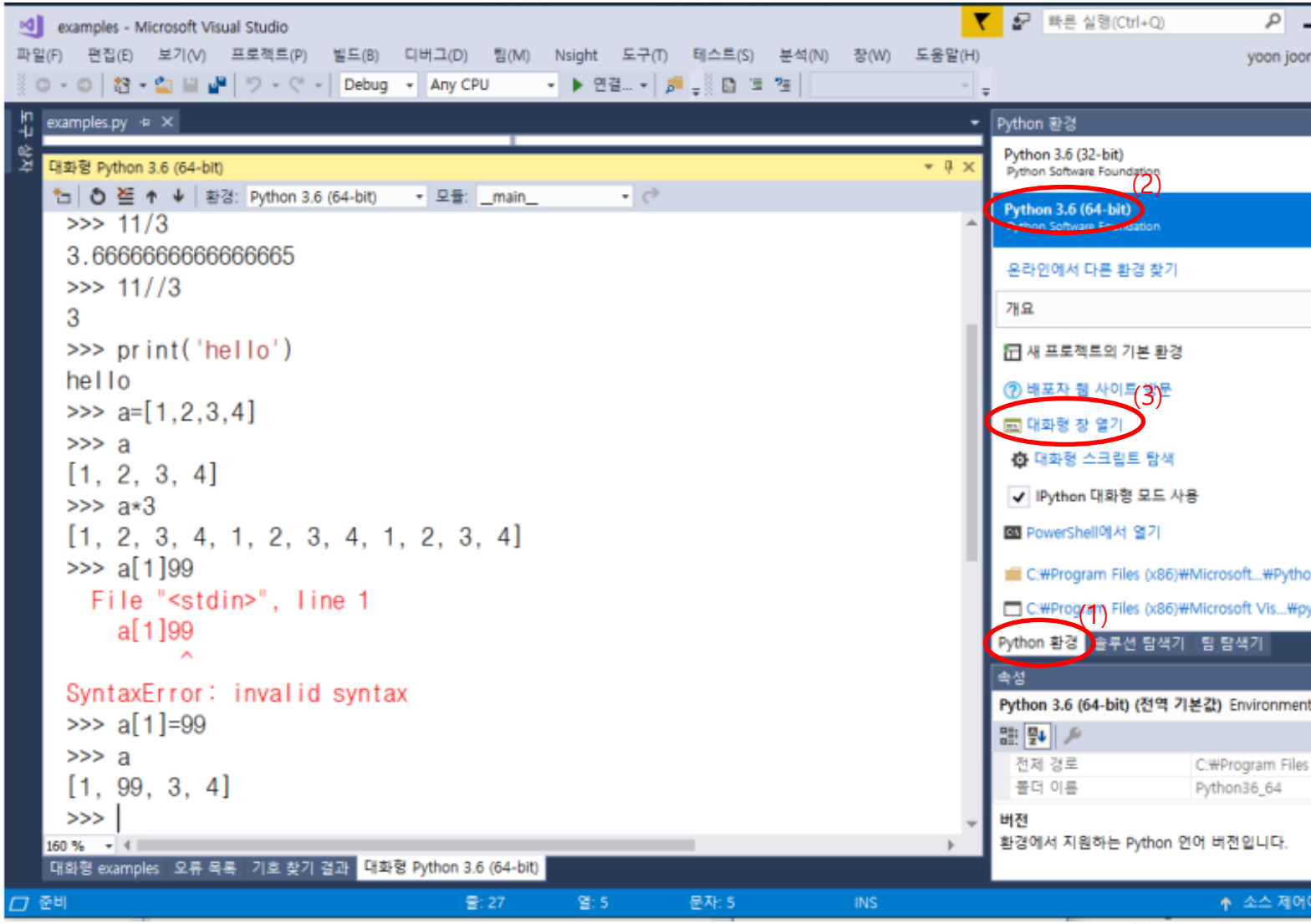
- Visual Studio 2019 IDE 구조
  - 솔루션탐색기(1)
    - 솔루션 PythonExamplesSolution(3) 은 복수 개의 실행단위 프로젝트로 구성된다.
    - 프로젝트 PythonExamples (4) 에는 Python 환경 "Python 3.7 (64-bit)" (5)와 파이썬 모듈들로 구성된다.(PythonExampls.py)(6)
  - Python 환경(2)
    - Python 라이브러리 다운로드 등등…
  - 편집창(7)
    - 선택된 모듈의 코드 편집
  - 속성창(8)
- 모듈 코드 작성 및 실행
  - 프로젝트 "PythonExamples" (4)에서 모듈 "PythonExamples.py" (6) 을 클릭
  - 편집창(7)에서 코드 작성
    print('Hello Python !!')
  - <시작>(8)을 클릭하여 모듈의 코드 실행
    콘솔창(9)에 실행결과가 출력된다.





6.3 대화형창에서 코드실행

- 6.3 대화형창에서 코드실행
  - <Python 환경> (1) 탭 크릭
  - Python 환경 (2)
    - Python 3.x(64-bit)
  - 대화형 창 열기 (3)
    (인터프리터 실행환경)
  - 대화형식으로 코드 작성 및 실행
    - 하나의 명령을 입력하고 엔터를 치면 실행된다. 하나의 명렬단위로 실행된다.
    - 임의로 입력해서 결과를 비교하는 시도를 해보자
      더하기,나누기,print, 리스트 …
    - Python은 들여쓰기(indent)로 프로그램의 구조를 표현합니다.
    - 이는 글쓰기에서 띄어쓰기와 유사합니다.
    - 파이썬의 문장의 시작은 처음 열부터 시작해야합니다.



6.4 파이썬의 기본연산

연산	연산자	예	결과
덧셈	+	2 + 3	5
뺄셈	-	5 - 3	2
곱셈	*	5 * 6	30
정수 나눗셈	//	8/3	2
실수 나눗셈	/	8.0/3	2.666...
나눗셈 후의 나머지 연산	%	8%3	2
지수승	**	3 **2	9

```
>>> 1+3
4
>>> 3-2
1
>>> 3*3
9
>>> 5/3
1.6666666666666667
```

```
>>> (50-5*6)/4
5
>>> 5/3.
1.6666666666666667
>>> 2*5
10
>>>
```

```
>>> divmod(11,3)
(3, 2)
>>> divmod(15.0, 3.5)
(4.0, 1.0)
>>> divmod(15.0, 3.2)
(4.0, 2.1999999999999993)
>>> 15.0 // 3.5
4.0
>>> 15.0 % 3.5
1.0
>>> 2**3
8
>>> pow(2,3)
8
```

```
>>> 11 // 3
3
>>> 11 / 3
3.6666666666666665
>>> 11./3
3.6666666666666665
>>> 11.5 // 4.3
2.0
>>> 11.5 // 4.3
2.0
>>> 11.5 / 4.3
2.674418604651163
>>>
```

6.5 모듈사용법

- 모듈은
  - 파이썬의 기본명령에 추가되는 명령들을 모아 놓은 파일에 저장되는 일종의 라이브러리 또는 namespace이다.
  - 예 math : 수학에 사용되는 다양한 연산 및 상수자료를 가지고 있다.

```
>>> import math
>>> math.pi
3.141592653589793
>>> math.sqrt(16)
4.0
>>>
```

```
>>> import math as m
>>> m.pi
3.141592653589793
>>> m.sqrt(16)
4.0
>>>
```

```
>>> from math import *
>>> pi
3.141592653589793
>>> sqrt(16)
4.0
>>>
```

```
>>> from math import pi, sqrt
>>> pi
3.141592653589793
>>> sqrt(16)
4.0
>>>
```

7. 연습문제

1. 파이썬의 환경을 직접 설치한다.
2. 파이썬 응용프로그램 프로젝트를 생성하고 모듈(파일)에 자신의 이름을 출력하는 코드를 작성하고 콘솔에 출력해 보세요. (6.1,6.2참조)
3. 대화형 Python 3.7(64-bit) 실행하여 인터프리터환경에서 자신의 이름을 출력해 보세요. (6.3 참조)
4. 사과 2500원하는 사과 10개를 사고 1000원을 할인을 받았을 경우 지불해야 하는 가격을 계산하는 프로그램을 작성해보세요. (6.3,6.4 참조)
5. 다음 다섯 가지 파이썬 표현 중 정상적으로 화면에 값이 출력되는 것을 고르시오.

```
1) print('I am a boy')
2) print("I am a boy")
3) print("I am a boy')
4) print("I live in 'Daejon'")
5) print('I live in "Daejon"')
6) print([hello])
7) print({hello})
```