



Jin-Soo Kim  
[\(jinsoo.kim@snu.ac.kr\)](mailto:jinsoo.kim@snu.ac.kr)

Systems Software &  
Architecture Lab.  
Seoul National University

Dec 16 – 20, 2019

# Basic Data Types

# Index

- Introduction
- Python 3 설치 (60')
  - Python Shell
- 개발 환경 구축 (60')
  - IDLE
  - Visual Studio Code
- Basic Lab (30') × 5
- Advanced Lab (45') × 2

# TA Introduction

- 이정민 ([tothesky7@snu.ac.kr](mailto:tothesky7@snu.ac.kr))
- 정연규 ([81887821@snu.ac.kr](mailto:81887821@snu.ac.kr))
- 장태준 ([tae7130@snu.ac.kr](mailto:tae7130@snu.ac.kr))

# Python 3 설치

# Python Version

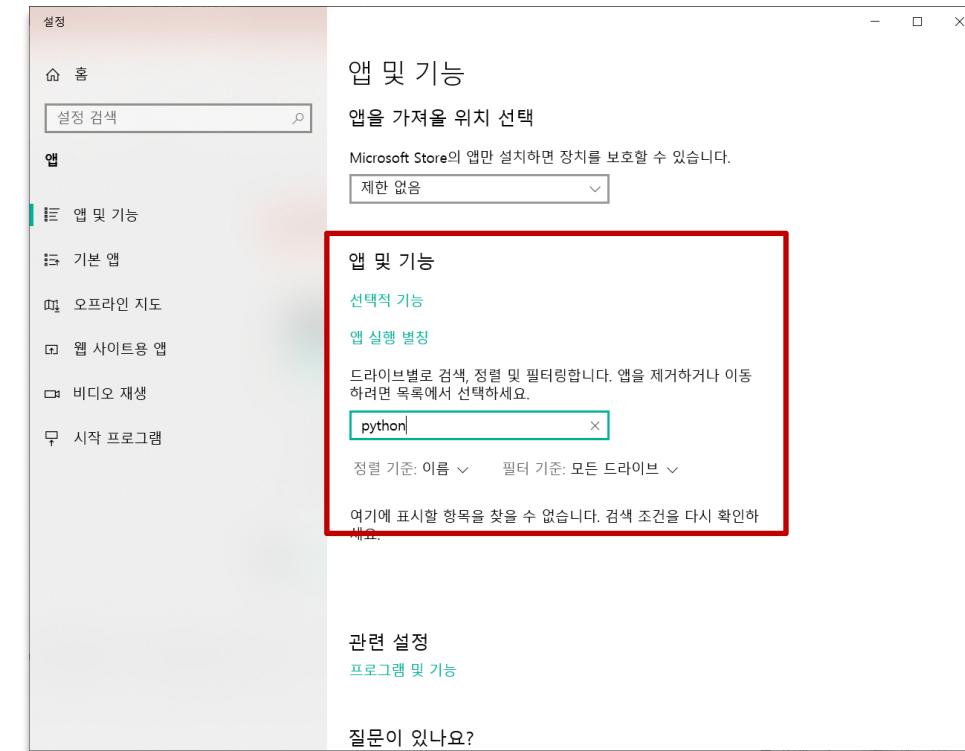
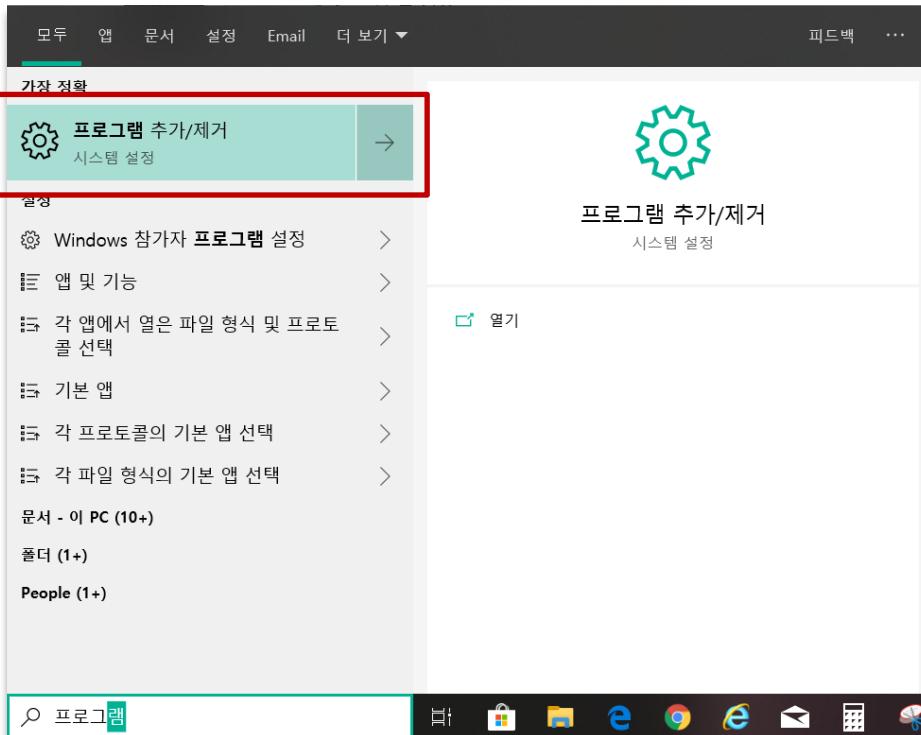
- 2019. 12. 16. 기준 최신 버전은 3.8
- 파이썬 버전 별로 문법이 달라짐
  - “`python --version`” 명령어로 설치된 파이썬의 버전 확인할 것
- 실습 수업은 Python 3 기준으로 진행 (Python 2 불가)
  - **Python 3.7 이상 권장**

# Python Packages

- ‘풍부한 라이브러리’는 파이썬의 대표적인 장점
  - **NumPy, pandas** : 데이터 분석에 활용되는 대표적인 라이브러리
- 현업에서 개발할 때는 많은 패키지를 동시에 관리할 필요가 있음
  - 패키지 관리를 도와주는 도구로는 conda 등이 있음
- 본 실습에서는 별도의 설치가 필요한 라이브러리를 사용하지 않음
  - 필요한 경우 Python 내장 라이브러리만 한정적으로 사용

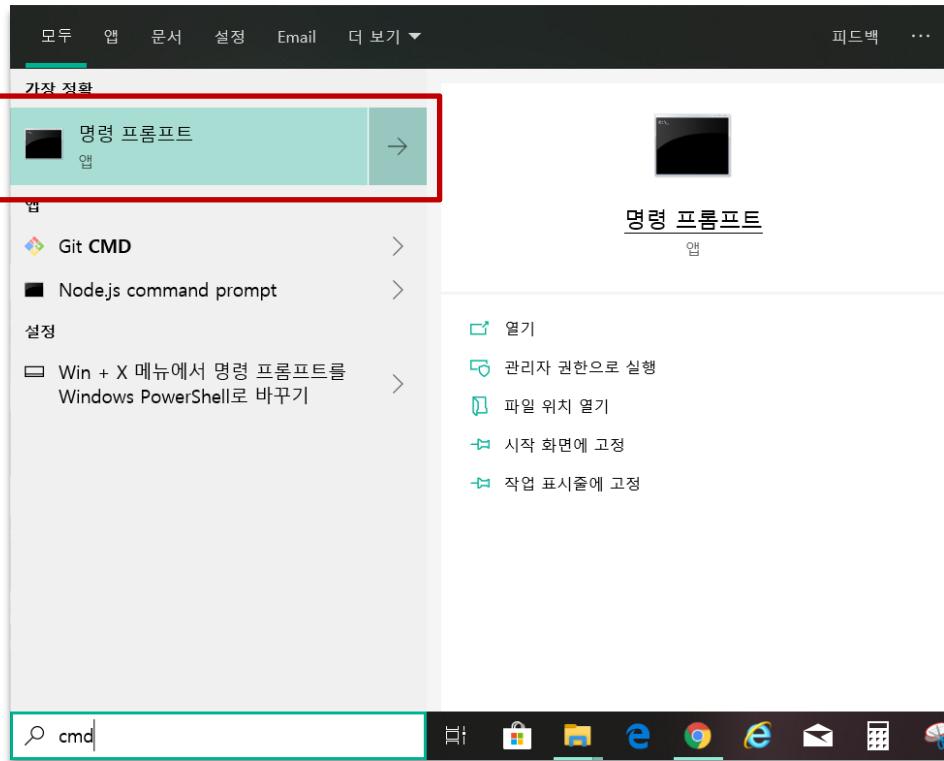
# Python 설치 (Windows)

- 컴퓨터에 이미 설치된 Python이 있는지 확인
  - 설치되어 있다면, 혹시 모를 충돌 예방을 위해 제거 권장
  - 윈도우 키 → “프로그램 추가/제거” 입력 후 실행 → “python” 검색



# Python 설치 (Windows)

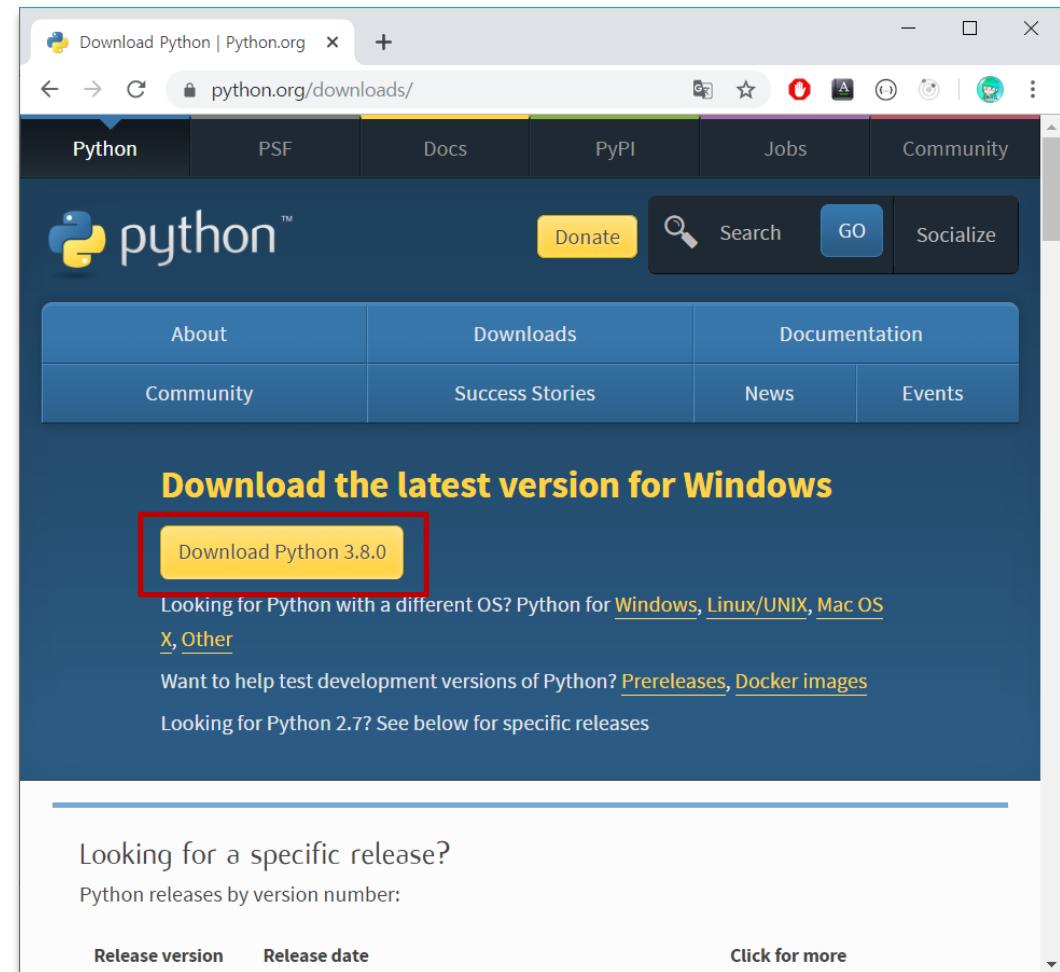
- 컴퓨터에 이미 설치된 Python이 있는지 확인
  - 설치되어 있다면, 혹시 모를 충돌 예방을 위해 제거 권장
  - 윈도우 키 → “Python” 입력 후 실행 → “python” 입력 후 엔터 키



A screenshot of a Windows Command Prompt window titled "cmd" with the path "C:\WINDOWS\system32\cmd.exe". The window displays the following error message:  
C:\Users\lg>python  
'python'은(는) 내부 또는 외부 명령, 실행할 수 있는 프로그램, 또는 배치 파일이 아닙니다.

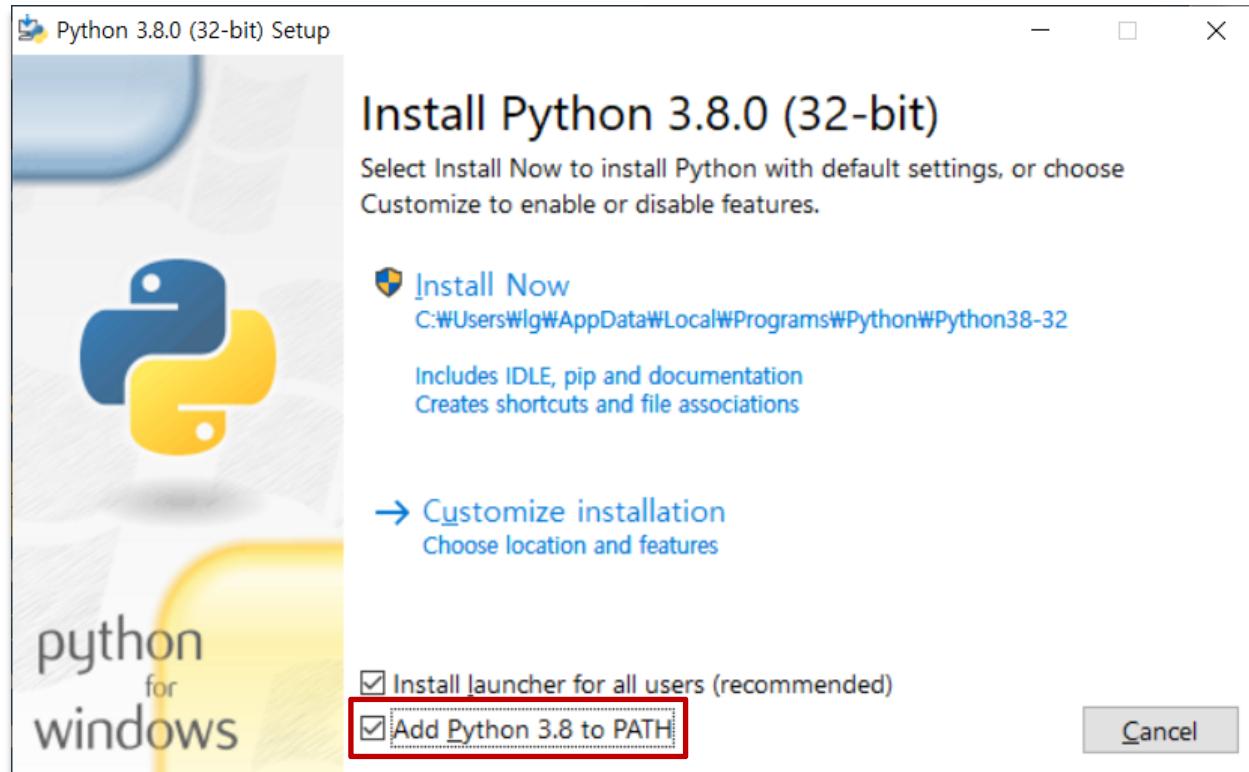
# Python 설치 (Windows)

- <https://www.python.org/downloads/>
  - 2019. 12. 16. 기준 Python 최신 버전은 3.8
  - “Download Python 3.8.0” 클릭



# Python 설치 (Windows)

- Python 3.8.0 설치 프로그램 실행
  - 첫 화면에서 **Add Python 3.8 to PATH** 부분 체크



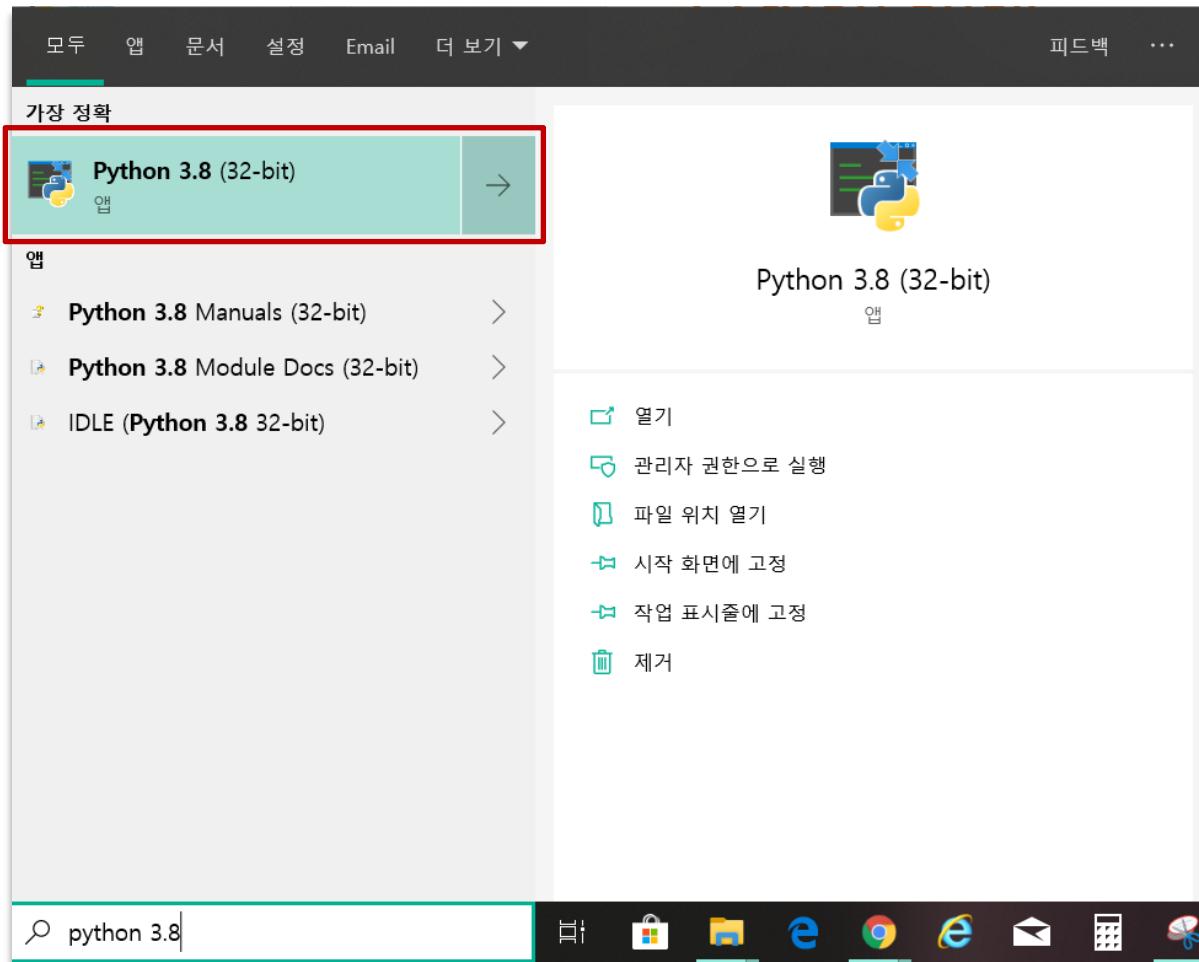
# Python 설치 (Windows)

- Python 3.8.0 설치 프로그램 실행
  - Install Now 클릭

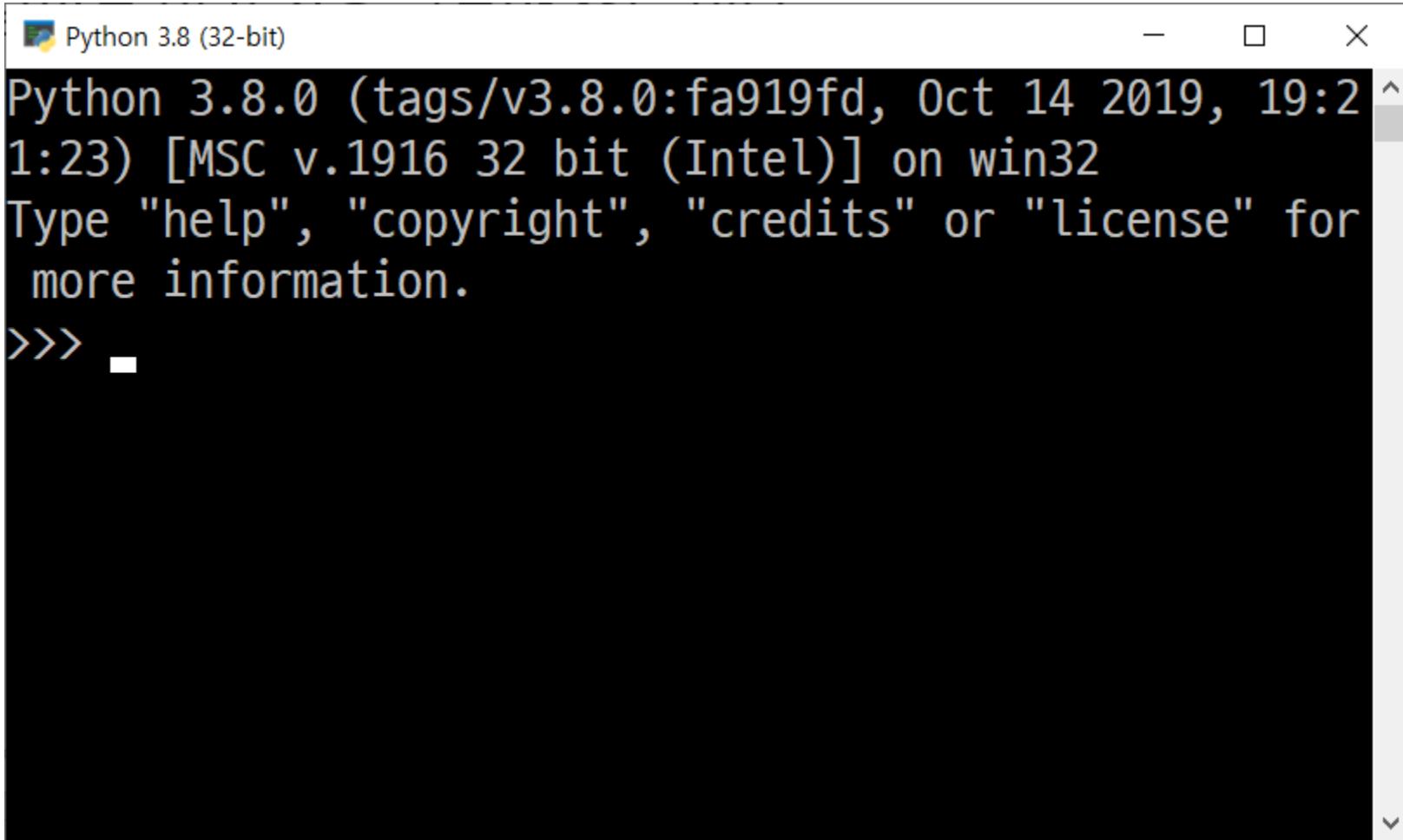


# Python Shell

- 원도우 키 – “python 3.8” 입력 – “Python 3.8” 앱 클릭



# Python Shell

A screenshot of a Windows-style application window titled "Python 3.8 (32-bit)". The window contains the Python 3.8.0 startup message:

Python 3.8.0 (tags/v3.8.0:fa919fd, Oct 14 2019, 19:21:23) [MSC v.1916 32 bit (Intel)] on win32  
Type "help", "copyright", "credits" or "license" for more information.

Below the message, the Python prompt ">>> " is visible, followed by a small black square cursor. The window has standard minimize, maximize, and close buttons at the top right, and a vertical scroll bar on the right side.

```
Python 3.8.0 (tags/v3.8.0:fa919fd, Oct 14 2019, 19:21:23) [MSC v.1916 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license" for more information.

>>> ■
```

# Python Shell

## ■ 산수 문제 풀기

- $2019 - 1999$ 는?
- $39800$ 의  $80\%$ 는?
- $3 + 2 \times 3$ 은?
- $15 \div 5 + 2 \times 3$ 은?
- $4.3, 2.7, 3.3$ 의 평균은?
- $2$ 의  $8$ 제곱은?
- $123$ 을  $17$ 로 나눈 몫과 나머지는?
- $27 \times 4$ 는  $17 \times 7$ 보다 큰가?
- $1$ 을  $0$ 으로 나누면 무슨 일이 벌어지는가?
- CAFE ( $16$ 진수)는  $51966$  ( $10$ 진수)와 같은가

# Python Shell

## ■ 문자열 다루기

- 다음 코드의 결과 예측하고 실행하기
  - 에러 발생 시 오류 메시지 확인하기

```
>>> "1" + "2"
>>> "이" + '정민'
>>> '300' - '1'
>>> 5 * '20'
>>> '20' * 5
>>> "올해는 " + 2019 + "년"
>>> prefix = "Py"
>>> prefix + "thon"
```

# Python Shell

## ■ 변수 활용하기

- 두 변수의 값을 바꾸는(swap) 방법

```
>>> x = "y라는 변수에 들어가고 싶은 문자열"
>>> y = "변수 x의 값이 되고 싶은 문자열"
>>> tmp = x      # Swap Start
>>> x = y
>>> y = tmp      # Swap End
>>> x
'변수 x의 값이 되고 싶은 문자열'
>>> y
'y라는 변수에 들어가고 싶은 문자열'
```

# Python Shell

## ■ 형변환

- Precision Issue

```
>>> large_int = 2 ** 100 + 10
>>> int(float(large_int)) == large_int
False
```

```
>>> large_float = 1.2345e300
>>> float(int(large_float)) == large_float
True
```

# Python Shell

## ■ 형변환

- Truthy Vs. Falsy

```
>>> bool(1), bool(0), bool(-1)
>>> bool(0.0000_0000_0000_0000_0001)
>>> bool('참'), bool('True')
>>> bool('False'), bool('0')
>>> bool('')
>>> bool(str(False)) == False
>>> str(bool('')) ==
>>> not '참', not 0.0, not -1
```

# Python Shell

## ■ 비교 연산자와 논리 연산자

```
>>> 1 > 0.3
>>> 1 == 1.0
>>> 1 is 1.0
>>> True == 1
>>> True == -1
>>> False < True
>>> 1 < '2'
>>> 'apple' < 'banana'
>>> True or False
>>> 1 or 0
>>> 0 or 2
```

# 개발 환경 구축

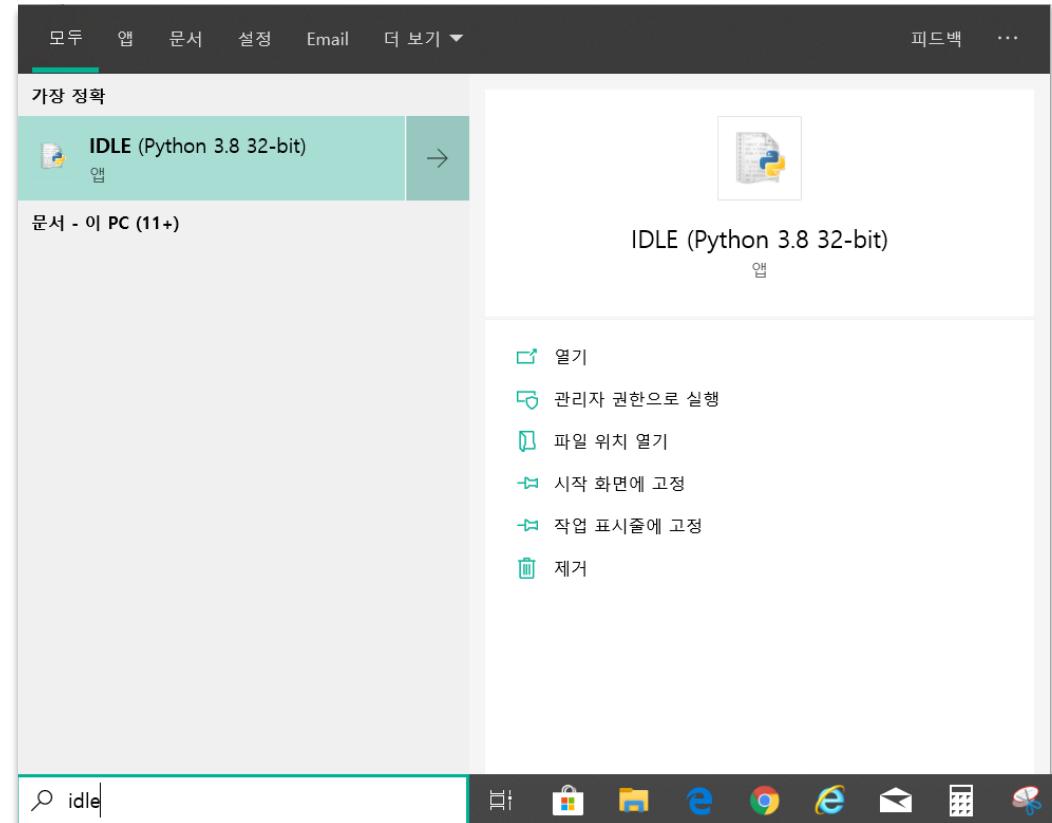
# Integrated Development Environment

- ‘파이썬 프로그래밍’을 위해서는 다양한 작업이 필요함
  - Python 3 설치, 코드 작성, 실행, 테스트, 빌드, 배포, ...
- 코드 편집기(Code Editor)
  - 위 작업 중 “코드 작성”을 중점적으로 도와주는 프로그램
  - 가장 단순한 사례로 메모장을 들 수 있음
- 통합 개발 환경(Integrated Development Environment)
  - 개발부터 프로그램 배포에 이르기까지 일련의 과정을 하나의 프로그램 안에서 통합적으로 처리할 수 있게 만든 환경

# Python IDLE

## ■ Python 설치 파일에 내장된 기본 개발 환경

- Integrated Development and Learning Environment
- 윈도우 키 → “IDLE” 입력해서 실행



# Python IDLE Shell

## ■ 다양한 편의 기능

- 구문 강조(Syntax Highlighting) : 문법에 맞춰 코드의 글자색이 달라짐
- 자동완성(Auto Completion) : 코드 작성 중 Tab 키
- 함수 도움말(Call Tip) : 함수 이름 입력 후 "(" 입력

The screenshot shows the Python 3.8.0 Shell window. The title bar reads "\*Python 3.8.0 Shell\*". The menu bar includes File, Edit, Shell, Debug, Options, Window, and Help. The main area displays the Python version information and a help message. Below that, a user types "2 + 2 = " + str(2 + 2) and '2 + 2 = 4'. Then, they type "print(" and press the Tab key, which triggers a code completion dropdown menu.

```
File Edit Shell Debug Options Window Help
Python 3.8.0 (tags/v3.8.0:fa919fd, Oct 14 2019, 19
:21:23) [MSC v.1916 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license()"
for more information.
>>> "2 + 2 = " + str(2 + 2)
'2 + 2 = 4'
>>> print(
    print(value, ..., sep=' ', end='\n', file=s)
```

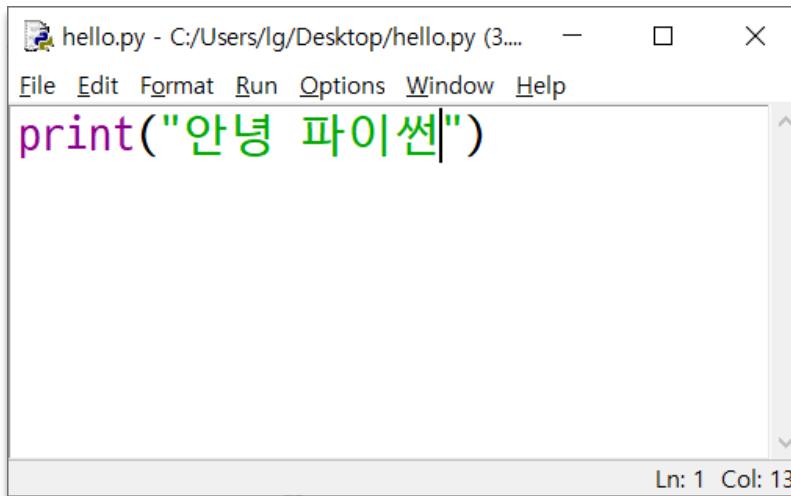
The screenshot shows the Python 3.8.0 Shell window. The user has typed "1 + 1" and pressed Enter, resulting in a SyntaxError: unexpected indent. Below the error message, another code completion dropdown menu is shown, listing various built-in functions like print, property, quit, range, repr, reversed, round, set, setattr, slice, etc. The word "print" is highlighted in blue, indicating it is the currently selected suggestion.

```
>>> 1 + 1
SyntaxError: unexpected indent
///
>>> print
print
property
quit
range
repr
reversed
round
set
setattr
slice
>>> pr
```

# Python IDLE Shell

## ■ 스크립트 파일 작성하기

- 메뉴 바에서 “File” → “New File (Ctrl + N)”로 새로운 창을 띄워서
  - 스크립트를 작성한 뒤 hello.py로 적당한 경로에 저장 (이후 Shell에서 Ctrl + O로 열 수 있음)
  - (새로운 창) 메뉴 바에서 “Run” → “Run Module (F5)” 실행



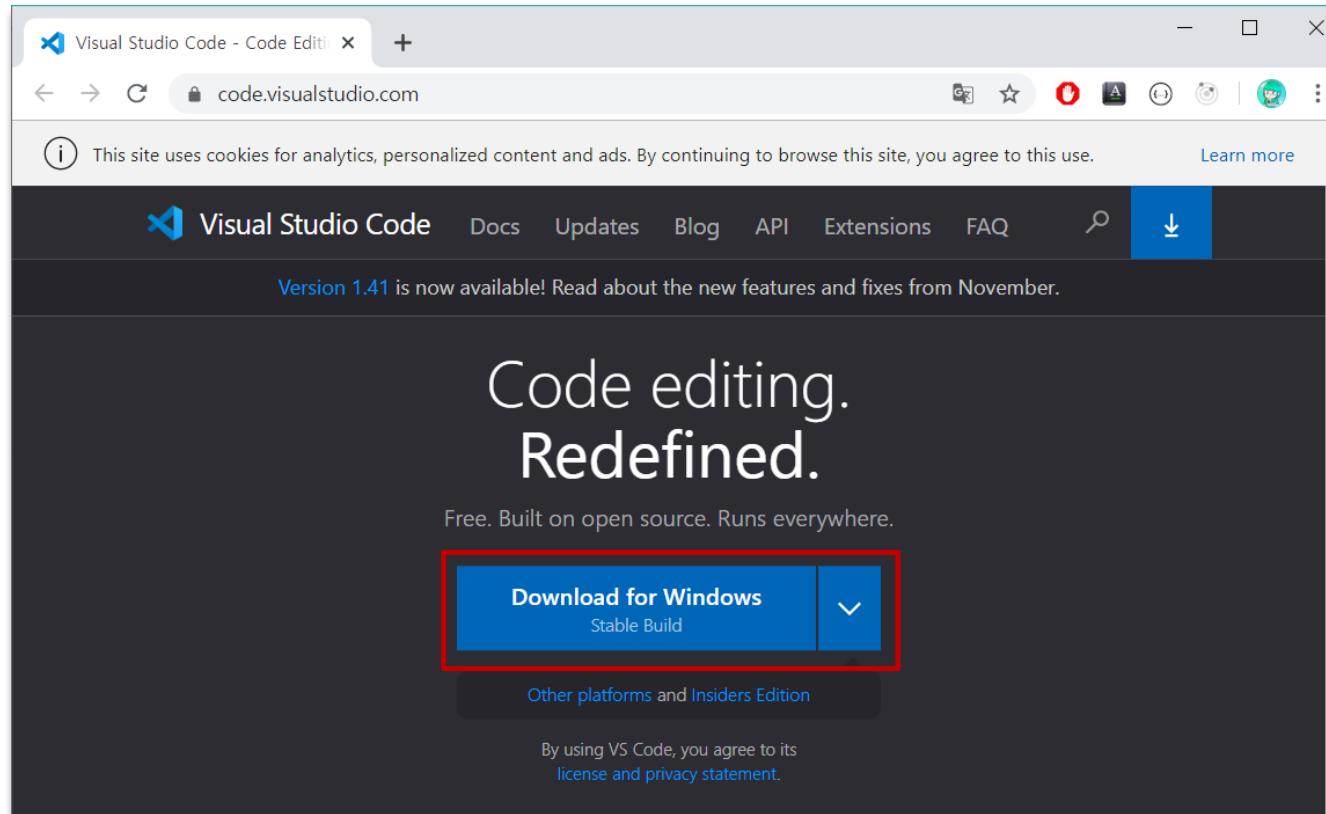
A screenshot of the Python 3.8.0 Shell interface. On the left is a code editor window titled "hello.py - C:/Users/lg/Desktop/hello.py (3...)" containing the single line of code: `print("안녕 파이썬")`. The status bar at the bottom indicates "Ln: 1 Col: 13". On the right is the shell window with a title bar "Python 3.8.0 Shell" and menu bar "File Edit Shell Debug Options Window Help". The shell window displays the following output:  
```  
>>>  
>>>  
>>>  
>>>  
>>>  
>>>  
>>>  
===== RESTART: C:/Users/lg/Desktop/hello.py =====  
안녕 파이썬  
>>>  
===== RESTART: C:/Users/lg/Desktop/hello.py =====  
안녕 파이썬  
>>> |  
Ln: 33 Col: 4  
```

# 다른 개발 환경

- **Visual Studio Code**
- PyCharm
- Atom
- Sublime Text
- Jupyter Notebook

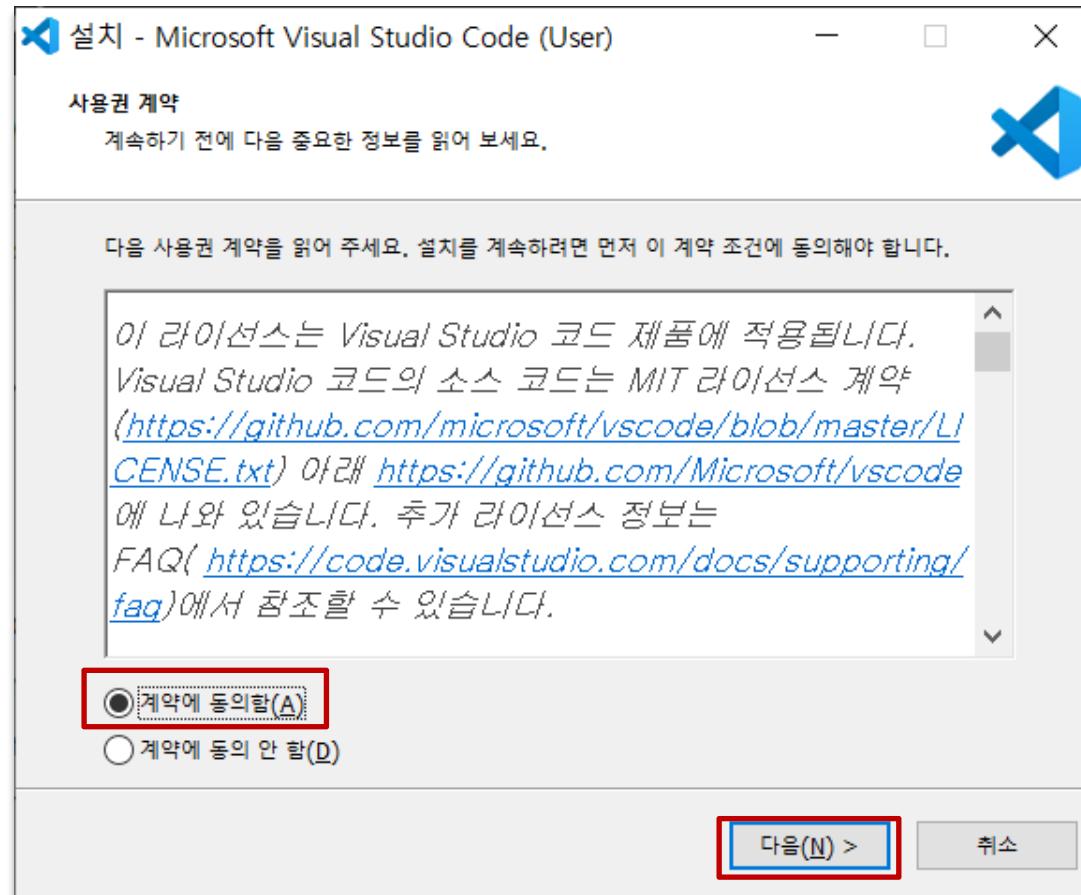
# Visual Studio Code 설치

- <https://code.visualstudio.com/>
  - Python IDLE 이상으로 다양하고 강력한 편의 기능 제공



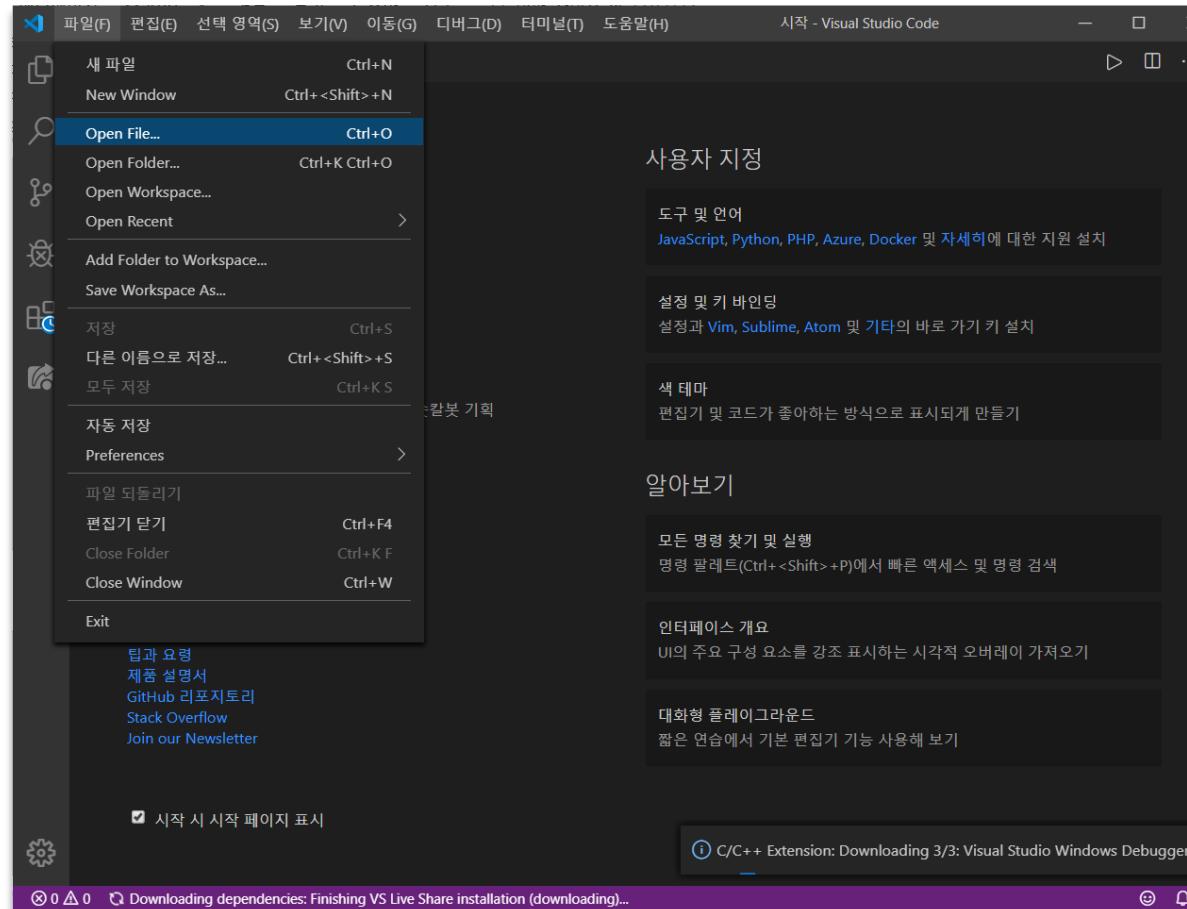
# Visual Studio Code

- 사용권 계약에 동의한 후 별다른 옵션 변경 없이 설치 진행



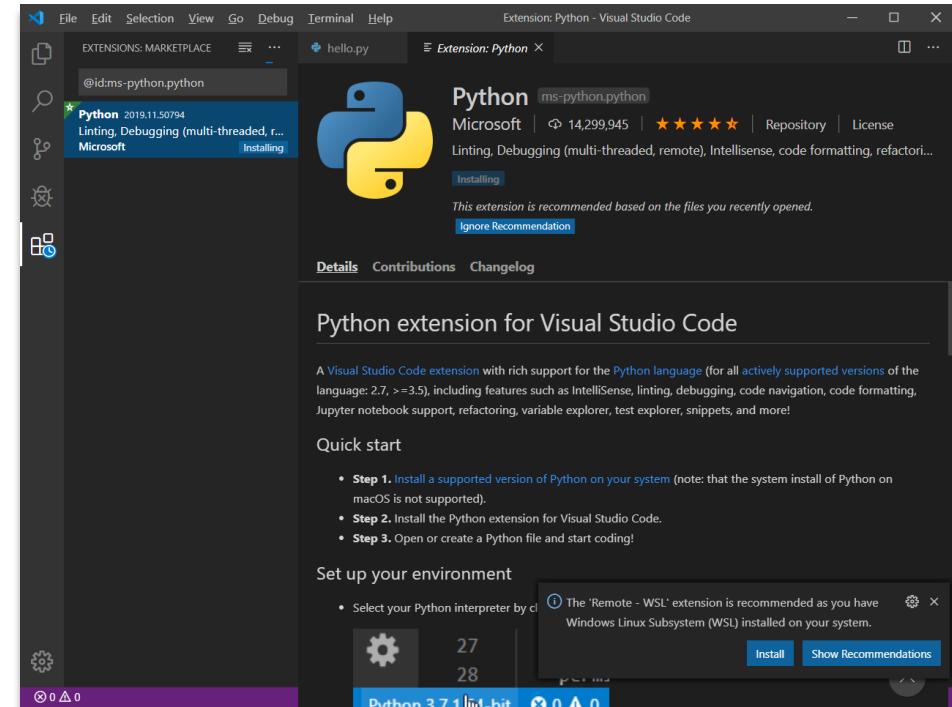
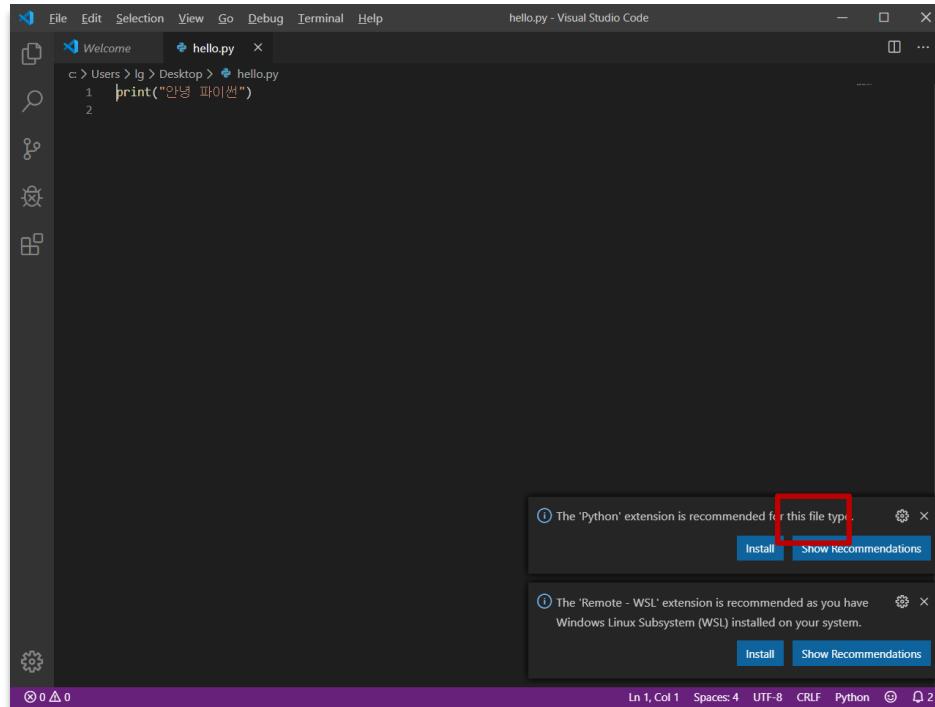
# Visual Studio Code

- “파일(F)” – “Open File... (Ctrl + O)”에서 hello.py 열어보기



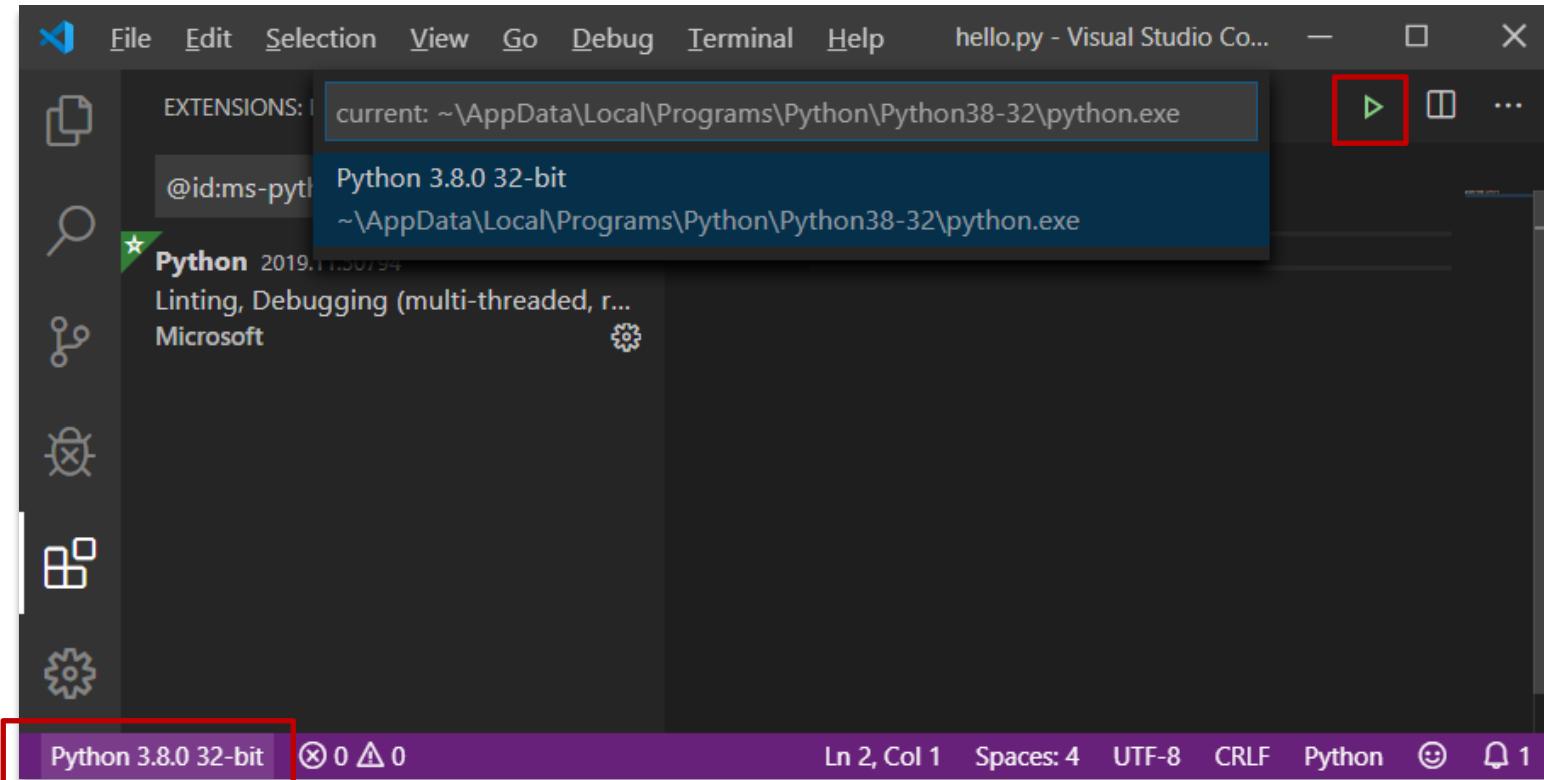
# Visual Studio Code

- “파일(F)” – “Open File... (Ctrl + O)”에서 hello.py 열어보기
- Microsoft의 “Python” Extension 설치하기
  - VSCode는 다양한 확장 프로그램을 설치해서 IDE처럼 사용할 수 있음



# Visual Studio Code

- 화면 좌측 하단에 Python이 제대로 인식되었는지 확인하기
- 화면 우측 상단에 코드 실행 버튼 클릭하기



# Visual Studio Code

- 화면 우측 하단에 터미널 창이 새로 나타난 것을 확인할 수 있음

The screenshot shows the Visual Studio Code interface. On the left is the Explorer sidebar with a file named 'hello.py' selected. The main editor area displays the following Python code:

```
c: > Users > lg > Desktop > hello.py
1 print("안녕 파이썬")
2 
```

Below the editor is the Terminal panel, which shows the output of running the script:

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

새로운 크로스 플랫폼 PowerShell 사용 https://aka.ms/pscore6

PS C:\Users\lg> & C:/Users/lg/AppData/Local/Programs/Python/Python38-32/python.exe
c:/Users/lg/Desktop/hello.py
안녕 파이썬
PS C:\Users\lg>
```

The status bar at the bottom indicates 'Python 3.8.0 32-bit' and shows the current file is 'hello.py'.

# Visual Studio Code

- 터미널 창에 텍스트를 입력하여 상호작용할 수도 있음

The screenshot shows the Visual Studio Code interface with the following details:

- File Menu:** File, Edit, Selection, View, Go, Debug, Terminal, Help.
- Title Bar:** hello.py - Visual Studio Code.
- Explorer Panel:** Shows "hello.py" in the "OPEN EDITORS" section. The path is "c: > Users > lg > Desktop > hello.py". The code content is "1 print(input())" and "2".
- Terminal Panel:** Shows a Python terminal session:

```
Copyright (C) Microsoft Corporation. All rights reserved.  
새로운 크로스 플랫폼 PowerShell 사용 https://aka.ms/pscore6  
PS C:\Users\lg> & C:/Users/lg/AppData/Local/Programs/Python/Python38-32/python.exe  
c:/Users/Desktop/hello.py  
안녕 파이썬  
PS C:\Users\lg> & C:/Users/lg/AppData/Local/Programs/Python/Python38-32/python.exe  
c:/Users/Desktop/hello.py  
test  
test  
PS C:\Users\lg>
```
- Status Bar:** Python 3.8.0 32-bit, 0 errors, 0 warnings, Ln 2, Col 1, Spaces: 4, UTF-8, CRLF, Python, 1 notification.

# Basic Lab

# Lab I-I. 복리 이율 계산하기

- 복리 이율 계산식

$$A = \left(1 + \frac{r}{100}\right)^n P$$

$A$ : 만기 금액

$P$ : 원금

$r$ : 연이율 (%)

$n$ : 만기 기간 (년)

- 아래와 같이 출력하는 프로그램 만들기

원금을 입력하세요: *10000* ↴

연이율을 입력하세요 (% 단위, 숫자만 입력): *2.5* ↴

만기까지의 기간을 입력하세요 (년): *10* ↴

만기 금액은 *12800.845441963567*입니다.

# Lab I-I. 복리 이율 계산하기

## ■ 복리 이율 계산식

$$A = \left(1 + \frac{r}{100}\right)^n P$$

$A$ : 만기 금액

$P$ : 원금

$r$ : 연이율 (%)

$n$ : 만기 기간 (년)

## ■ 몇 개의 “변수”가 필요한가?

- 사용자에게 값을 직접 입력 받아야하는 변수는 몇 개인가?
- 각 변수마다 적합한 자료형은 무엇인가? (int, bool, float, str)

## ■ 이 문제에서 “상수”는 필요한가?

# Lab I-I. 복리 이율 계산하기

- 복리 이율 계산식

$$A = \left(1 + \frac{r}{100}\right)^n P$$

$A$ : 만기 금액

$P$ : 원금

$r$ : 연이율 (%)

$n$ : 만기 기간 (년)

- 위 수식을 어떻게 Python 코드로 변환하는가? (참고 : \*\* 연산자)

```
1 P = float(input("원금을 입력하세요: "))
2 r = float(input("연이율을 입력하세요 (% 단위, 숫자만 입력): "))
3 n = int(input("만기까지의 기간을 입력하세요 (년): "))
4
5 A = ?????????
```

# Lab I-2. 요일 계산하기

- 문자열 연산을 자세히 배우지 않았기에, 요일을 숫자에 대응

월요일	화요일	수요일	목요일	금요일	토요일	일요일
1	2	3	4	5	6	7

- 아래와 같이 출력하는 프로그램 만들기
  - 금요일(5)의 3일 뒤는 월요일(1)

오늘은 무슨 요일입니까? (월: 1 ~ 일: 7): 5↙

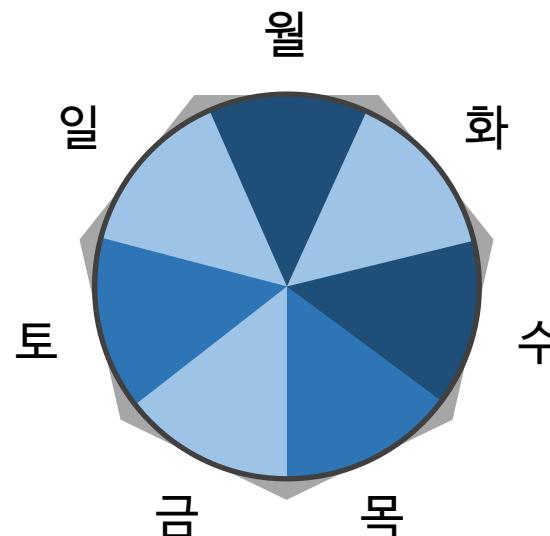
며칠 뒤의曜일을 알고 싶습니까?: 3↙

그날의曜일은 1입니다. (월: 1 ~ 일: 7)

# Lab I-2. 요일 계산하기

## ■ 나머지 연산 $a \% b$ 를 이용

- Python에서는 나머지 연산의 결과가 항상 **0 이상 b 미만**임이 보장됨
- 그러나 주어진 문제 상황은 **0~6**이 아닌 **1~7**로 숫자가 배정됨



```
1 today = int(input("오늘은 무슨曜일입니까? (월: 1 ~ 일: 7): "))
2 duration = int(input("며칠 뒤의曜일을 알고 싶습니까?: "))
3
4 future = (today + duration) % 7      # WRONG CODE
5 print("그 날의曜일은 " + str(future) + "입니다. (월: 1 ~ 일: 7)")
```

# Lab I-3. 수 반올림하기

- 아래와 같이 출력하는 프로그램 만들기

- 소수점 아래 자리수가 0.5 이상인 경우, 올림
- 소수점 아래 자리수가 0.5 미만인 경우, 버림

반올림할 수를 입력해 주세요: 1.67↙  
반올림 결과는 2입니다.

반올림할 수를 입력해 주세요: 6.49↙  
반올림 결과는 6입니다.

# Lab I-3. 수 반올림하기

- "소수점 위"와 "소수점 아래"를 분리하는 방법?
  - Python의 경우, **float** 자료형 간의 나머지 연산도 지원함

```
>>> 1 % 0.7  
>>> 3.2 % 2  
>>> 1.5 % 0.4
```

- **float** 자료형에 대한 **int()** 형변환 함수는 무조건 버림
- 수를 올림하는 방법?
  - 소수점 아래 자리수가 0.5 이상일 경우, 버림값에 1을 더함

# Lab I-3. 수 반올림하기

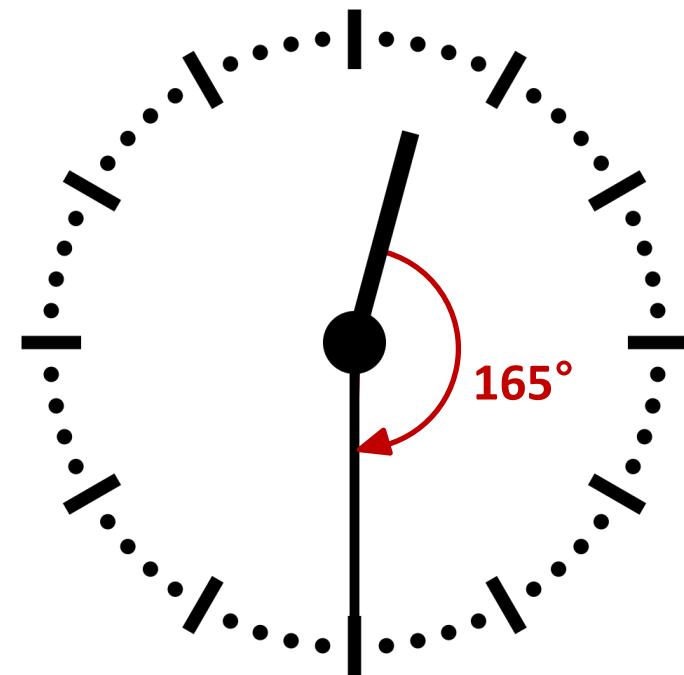
- 사실, Python에는 **round()**함수가 기본 제공됨
  - `round(x)`  
== `int(x) + int((x % 1) // 0.5)`  
== `int(x + 0.5)`
- $x.0$ 인 경우,  $x.1$ 인 경우,  $x.5$ 인 경우,  $x.9$ 인 경우  
네 가지 경우에 대해 수식을 대입해보며 이해하기

# Lab I-4. 시계 시침 분침 각도 구하기

- 아래와 같이 출력하는 프로그램 만들기
  - 시침에서 분침으로 이동하는데 시계방향으로 얼마나 이동해야 하는 지

시침 값을 입력해 주세요: 12 ↵  
분침 값을 입력해 주세요: 30 ↵  
시침에 대한 분침의 각도는 165.0도입니다.

시침 값을 입력해 주세요: 3 ↵  
분침 값을 입력해 주세요: 0 ↵  
시침에 대한 분침의 각도는 270.0도입니다.



© Micthev, Wikimedia Commons. CC BY-SA 4.0

# Lab I-4. 시계 시침 분침 각도 구하기

## ■ 수식 세우기

- 1분이 지날때마다 분침은  $360^\circ \div 60 = 6^\circ$ 씩 회전한다.
- 1시간이 지날때마다 시침은  $360^\circ \div 12 = 30^\circ$ 씩 회전한다.
- 1분이 지날때마다 시침은  $30^\circ \div 60 = 0.5^\circ$ 씩 회전한다.

## ■ 즉, H시 M분일 때

- 시침의 각도는  $30H + 0.5M$
- 분침의 각도는  $6M$
- 시침에 대한 분침의 각도는  $6M - (30H + 0.5M) = 5.5M - 30H$

## ■ 추가적인 고려사항은?

# Lab I-5. 계산 시험

- 아래와 같이 출력하는 프로그램 만들기

```
71 + 42 = 113↙
```

```
153 + 188 = 331↙
```

```
321 + 1016 = 1337↙
```

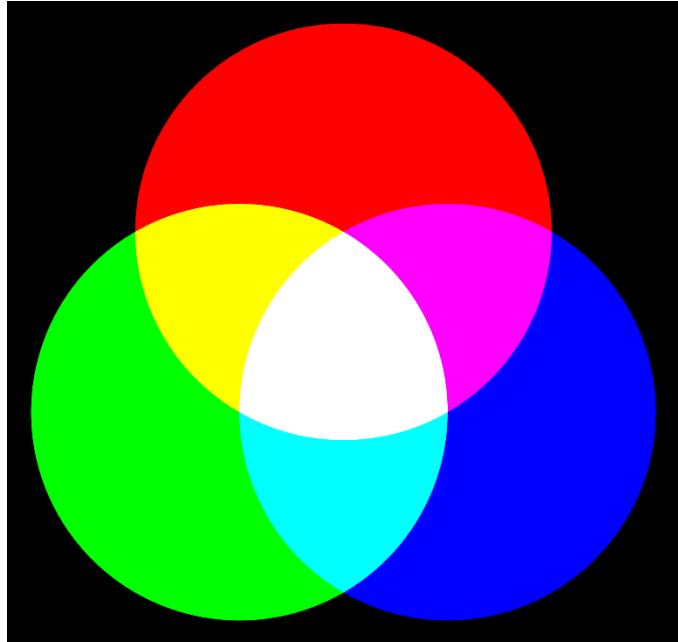
3개 중 2개 맞았습니다.

- 아직 안 배운 조건문(if)을 사용하지 않고 구현하기
  - Hint: `bool`을 `int()`로 형변환
  - Hint: `a = a + b`는 `a += b`와 동치

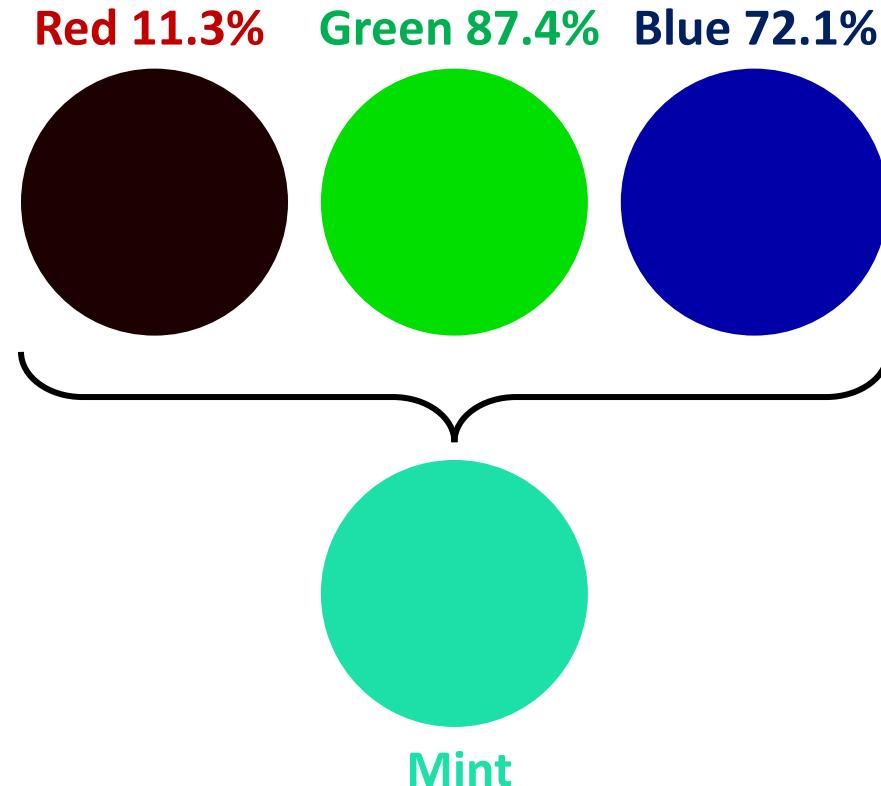
# Advanced Lab

# Lab I-6. 색상 코드

- RGB는 컴퓨터에서 색을 표현하는 방법 중 한 가지
  - 빨강(Red), 초록(Green), 파랑(Blue) 세 광원의 밝기를 조절하여 조합



© Quark67, Wikimedia Commons. CC BY-SA 3.0



# Lab I-6. 색상 코드

- 통상적으로 삼색광 광원의 세기는 0~255의 정수로 표현함
  - 0이 가장 어둡고(0%, 광원 없음), 255가 가장 밝음(100%)
  - 컴퓨터 상에서 8비트로 표현 가능
  - 16진수 두 자리로 표현하기 용이함 (0x00~0xFF)
- 색을 RGB로 표현하는 방법은 크게 두 가지가 통용됨
  - 16진수로 붙여 표현하는 방법 #RRGGBB (ex: #C0FEE)
  - 10진수로 표현하는 방법 rgb(RRR, GGG, BBB) (ex: rgb(192,255,238))

# Lab I-6. 색상 코드

- 16진수 색상 코드를 입력받아, 10진수 색상코드로 변환하고자 함
- 입출력 예시는 다음과 같음
  - 10진수 변환 결과는 꼭 세 자리일 필요 없음
  - Hint: 형변환 함수 `int()` 활용

색상 코드 1-2째 자리: `c0↙`

색상 코드 3-4째 자리: `ff↙`

색상 코드 5-6째 자리: `ee↙`

`rgb(192, 255, 238)`

# Lab I-7. 막대 그래프 그리기

## ■ 입출력 예시는 다음과 같음

- 시험 점수를 십의 자리 밑에서 반올림하여 10점당 '#' 하나씩
- 오른쪽 점수가 정렬되었음에 유의
- Hint: 'a' \* 5 == 'aaaaa', 'a' \* -1 == ''

국어 점수: 85 ↴

수학 점수: 23 ↴

영어 점수: 100 ↴

국어 ##### 85

수학 ## 23

영어 ##### 100

# Lab I-7. 막대 그래프 그리기

## ■ 십의 자리 밑에서 반올림하기

- 85를 십의 자리 밑에서 반올림하는 문제는  
8.5를 일의 자리 밑에서 반올림하는 문제로 환원할 수 있음
- `round(x) == int(x + 0.5) == (x + 0.5) // 1`
- 그렇다면, 십의 자리 밑에서는?

## ■ 수 정렬하기

- 점수 `score`를 십의 자리에서 반올림한 값을 `rounded`로 정의
- `rounded`와 상수 `10`을 이용하여 수식 설계하기