

1주

파이썬 프로그래밍 소개 및 환경 구축



염 희 균

파이썬 교육 일정

no.	일시			내용
1	2020.7.1(목)	14:00~17:00	3H	Python 프로그래밍 소개 및 개발 환경 구축
2	2020.7.2(금)	14:00~17:00	3H	Python 자료형과 문자열 - 기본 자료형과 컨테이너 자료형 - 문자열과 연산자 - 숫자 - 리스트, 튜플, 딕셔너리
3	2020.7.5(월)	14:00~17:00	3H	Python 변수, 다양한 내장함수
4	2020.7.6(화)	14:00~17:00	3H	Python 조건문 - 불 자료형과 if조건문 - if~else와 elif구문
5	2020.7.7(수)	14:00~17:00	3H	Python 반복문 - 리스트와 반복문 - 딕셔너리와 반복문
6	2020.7.8(목)	14:00~17:00	3H	Python for 반복문과 while 반복문
7	2020.7.9(금)	14:00~17:00	3H	Python 다양한 함수 - 튜플과 람다 - 파일 처리
8	2020.7.12(월)	14:00~17:00	3H	Python 예외 처리
9	2020.7.13(화)	14:00~17:00	3H	Python 표준 모듈
10	2020.7.14(수)	14:00~17:00	3H	Python 외부 모듈 - Flask 모듈 - BeautifulSoup 모듈
11	2020.7.15(목)	14:00~17:00	3H	Python 모듈 만들기
12	2020.7.16(금)	14:00~17:00	3H	Python 객체지향 프로그래밍 방법1 - 클래스와 객체
13	2020.7.19(월)	14:00~17:00	3H	Python 객체지향 프로그래밍 방법2 - 상속
14	2020.7.20(화)	14:00~17:00	3H	Python 예제 프로그램 개발1
15	2020.7.21(수)	14:00~17:00	3H	Python 예제 프로그램 개발2

- 파이썬 소개
- 파이썬 설치하기
- 파이썬 실행하기 : 인터렉티브 셸
- 텍스트 에디터 사용하기 : 파이썬 IDLE 에디터
- 텍스트 전문 에디터 사용하기 : Visual Studio Code Install
- 예약어와 주석
- print()
- 확인문제

- 파이썬 (Python)

- 1991년 귀도 반 로섬 (Guido van Rossum)이 개발
- 초보자가 쉽게 배울 수 있는 프로그래밍 언어



- 파이썬의 장점

- 비전공자도 쉽게 배울 수 있음
- 다양한 분야에서 활용할 수 있음
- 대부분의 운영체제에서 동일하게 사용됨

- 파이썬의 단점

- C언어에 비해 일반적으로 10~350배 느림
- 최근에는 컴퓨터 성능이 좋아져 연산이 많이 필요한 프로그램이 아니라면 차이 크게 느낄 수 없음

파이썬으로 할 수 있는 일

- GUI 프로그래밍

- GUI 프로그래밍을 위한 도구들이 지원됨
- 파이썬 프로그래밍과 함께 설치되는 Tkinter가 있다.

- 웹프로그래밍

- 데이터 분석, 사물인터넷

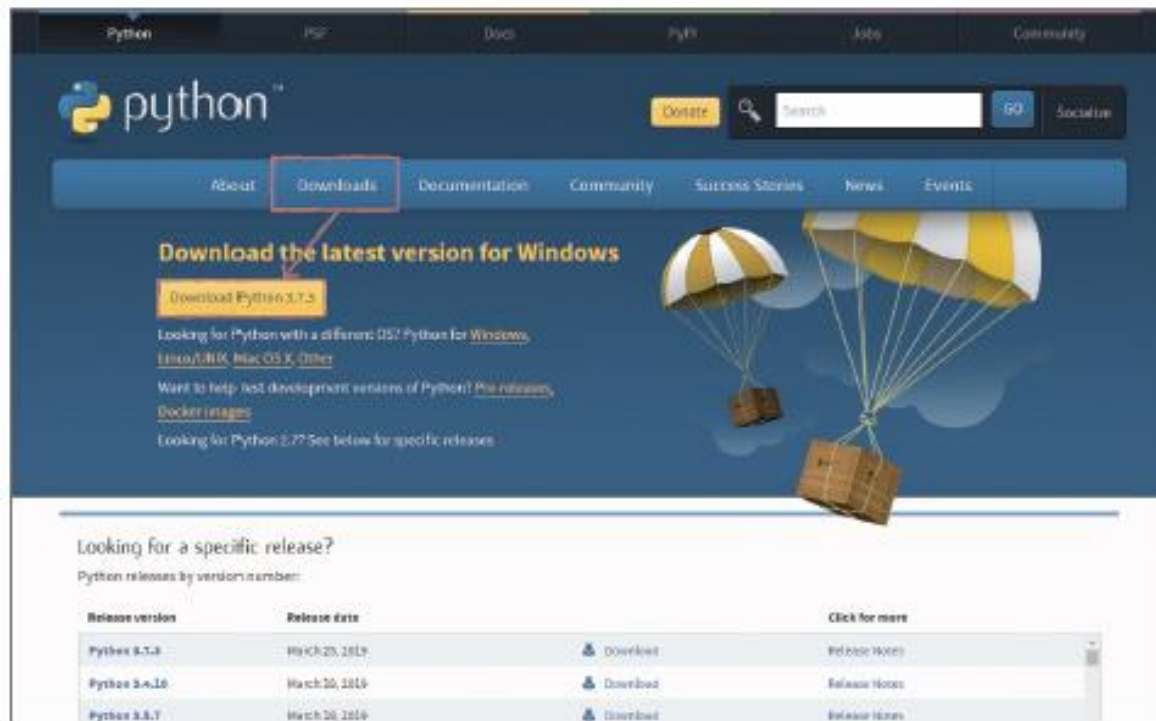
- 파이썬으로 만든 '판다스' 모듈을 사용하면 데이터 분석을 더 쉽고 효과적으로 할 수 있다.
- 라즈베리파이를 제어하는 도구로 파이썬을 사용한다. (예: 모터작동, LED 불 들어오게 하기등)

파이썬 설치하기

- 파이썬 설치 프로그램 다운로드

1) 파이썬 공식 홈페이지 (<http://www.python.org>) 접속 - [Downloads]

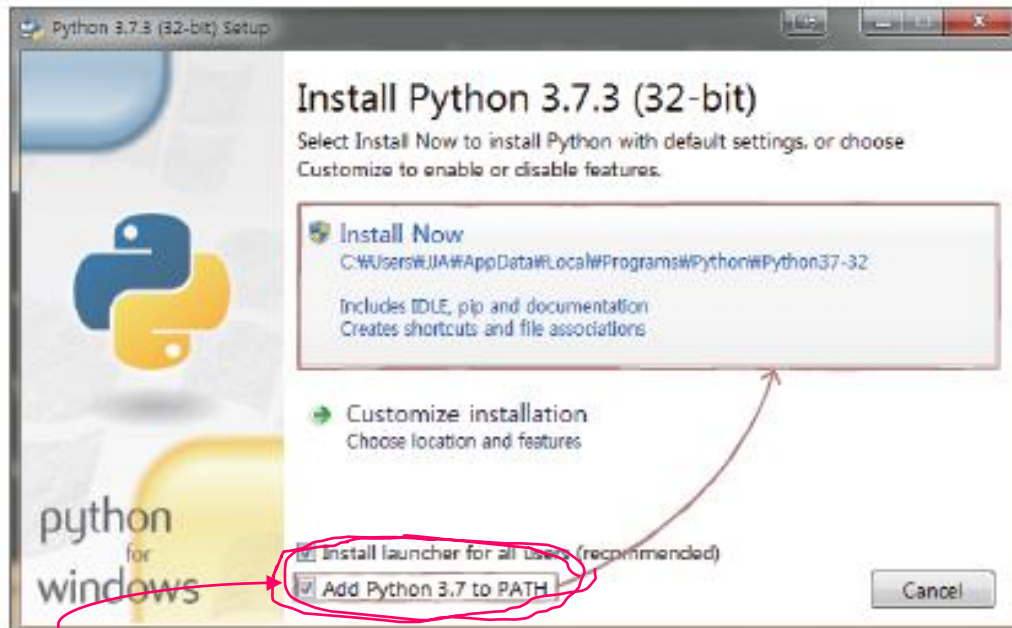
2) [Download Python 3.9.x] 클릭



파이썬 설치하기

- 파이썬 설치하기

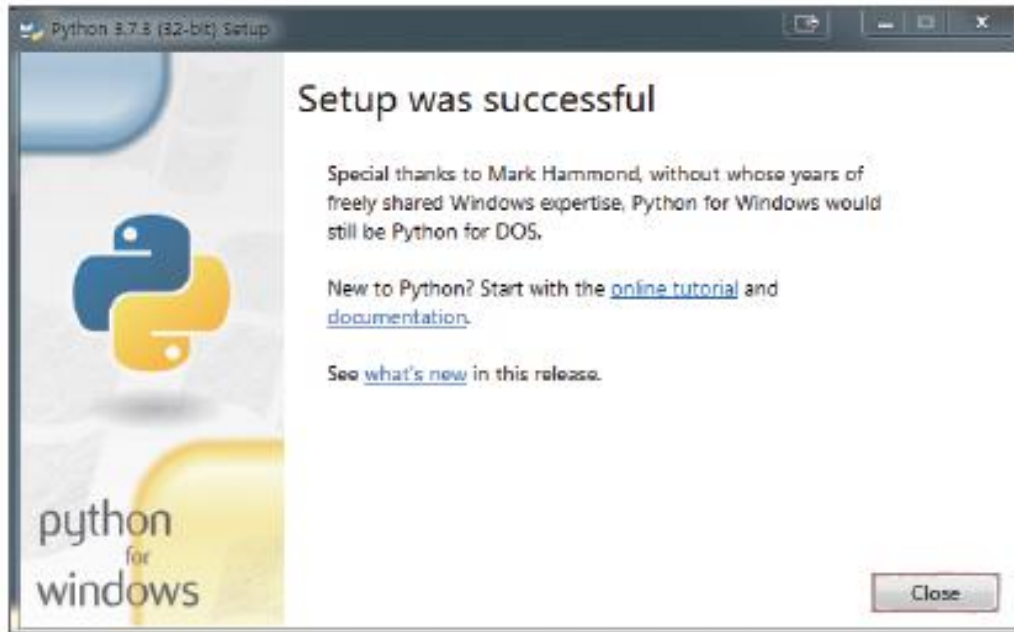
- 1) 설치 프로그램 실행하여 아래 화면에서 [Add Python 3.9 to PATH] 체크
- 2) [Install Now] 클릭



여기를 체크하지 않고 설치하면 파이썬이 실행되지 않으므로 꼭 확인합니다.

파이썬 설치하기

3) 설치 완료 화면이 나타나면 [Close] 클릭



4) 윈도우 [시작] 메뉴에서 [Python 3.9] 프로그램 확인



파이썬 실행하기 : 파이썬 인터랙티브 셸

- 인터프리터 (interpreter)

- 파이썬으로 작성된 코드를 실행해주는 프로그램
- [시작]-[Python3.x(00-bit)프로그램 선택하기

- 파이썬 인터랙티브 셸

- ```
I Python 3.7.2 (tags/v3.7.2:9a3ffc0492, Dec 23 2018, 22:20:52) [MSC v.1916 32 bit
(Intel)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>>
```

# 파이썬 실행하기 : 파이썬 인터랙티브 셸

- 프롬프트 (prompt)

- >>>

- 코드를 한 줄씩 입력

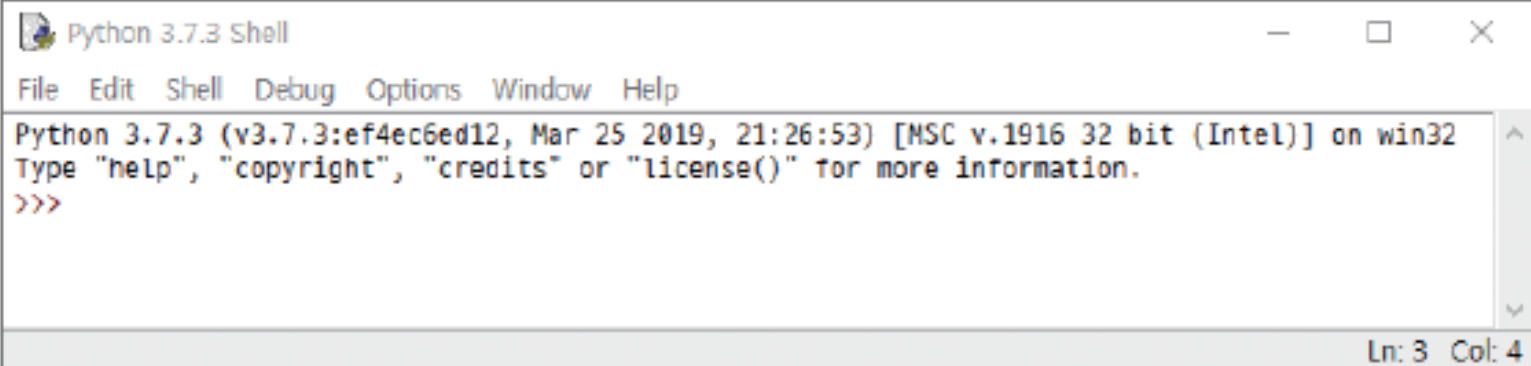
- 인터랙티브 셸 = 대화형 셸

: 컴퓨터와 상호 작용하는 공간이며, 한 마디씩 주고받는 것처럼 대화한다고 하여 대화형 셸로 불리기도 함

```
>>> 10 + 10 [Enter] → 10 + 10을 입력하니
20 → 10과 10을 더해 20을 출력합니다.
>>> "Hello" * 3 [Enter] → Hello라는 문자열을 3번 출력하라는 의미이며,
'HelloHelloHello' → 'HelloHelloHello'를 출력합니다.
>>>
```

# 텍스트 에디터 사용하기 : 파이썬 IDLE 에디터

- 텍스트 에디터 (text editor)
    - 긴 코드를 입력하거나 코드를 저장해야 하는 경우 사용
    - 글자를 적을 수 있는 모든 종류의 프로그램
  - IDLE 에디터에서 코드 작성하고 실행하기
    - 파이썬은 통합 개발 환경으로써 IDLE 제공
- 1) [시작 메뉴] – [Python 3.9] – [IDLE]

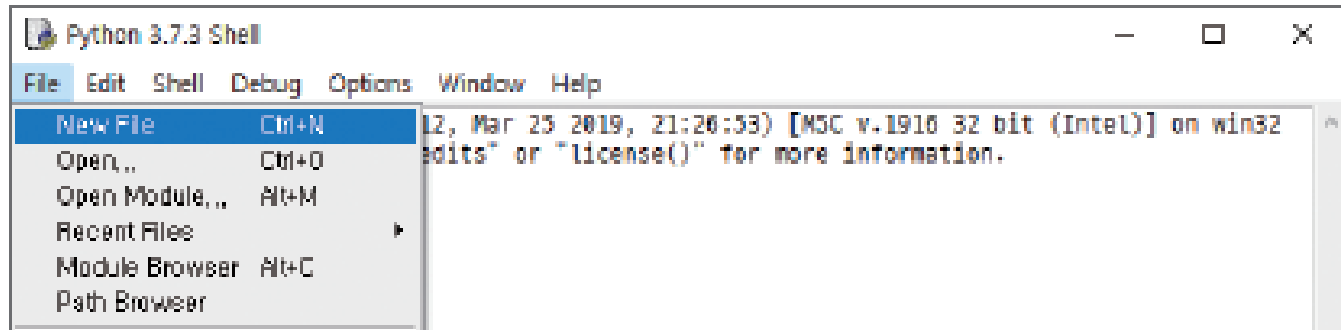


```
Python 3.7.3 Shell
File Edit Shell Debug Options Window Help
Python 3.7.3 (v3.7.3:ef4ec6ed12, Mar 25 2019, 21:26:53) [MSC v.1916 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
```

Ln: 3 Col: 4

# 텍스트 에디터 사용하기 : 파이썬 IDLE 에디터

## 2) [File] – [New File] 메뉴 선택

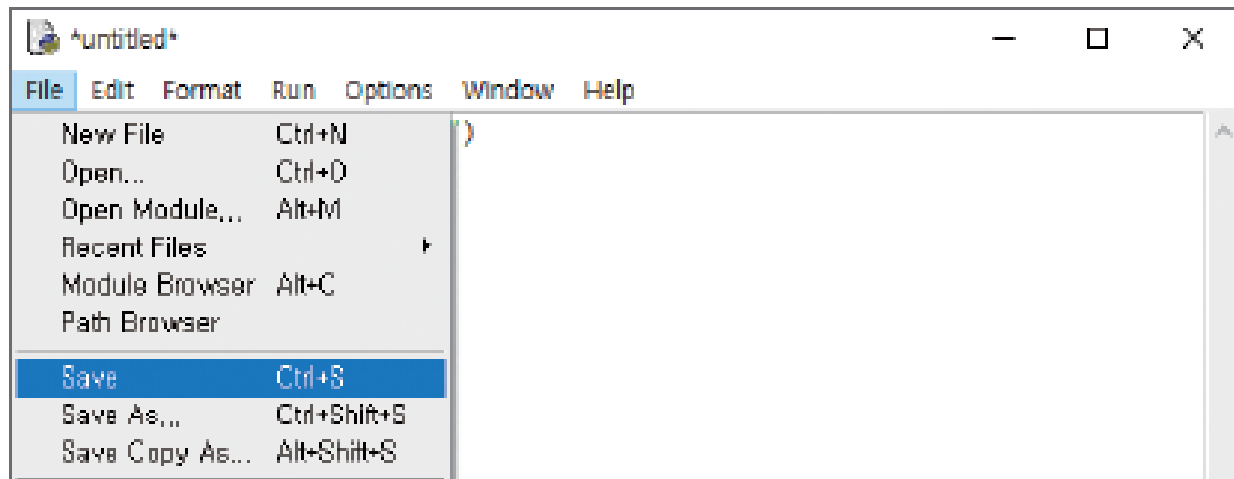


## 3) 아래와 같이 입력

```
print("IDLE에서 파이썬 코드를")
print("작성해서 출력해 보는")
print("예제입니다")
```

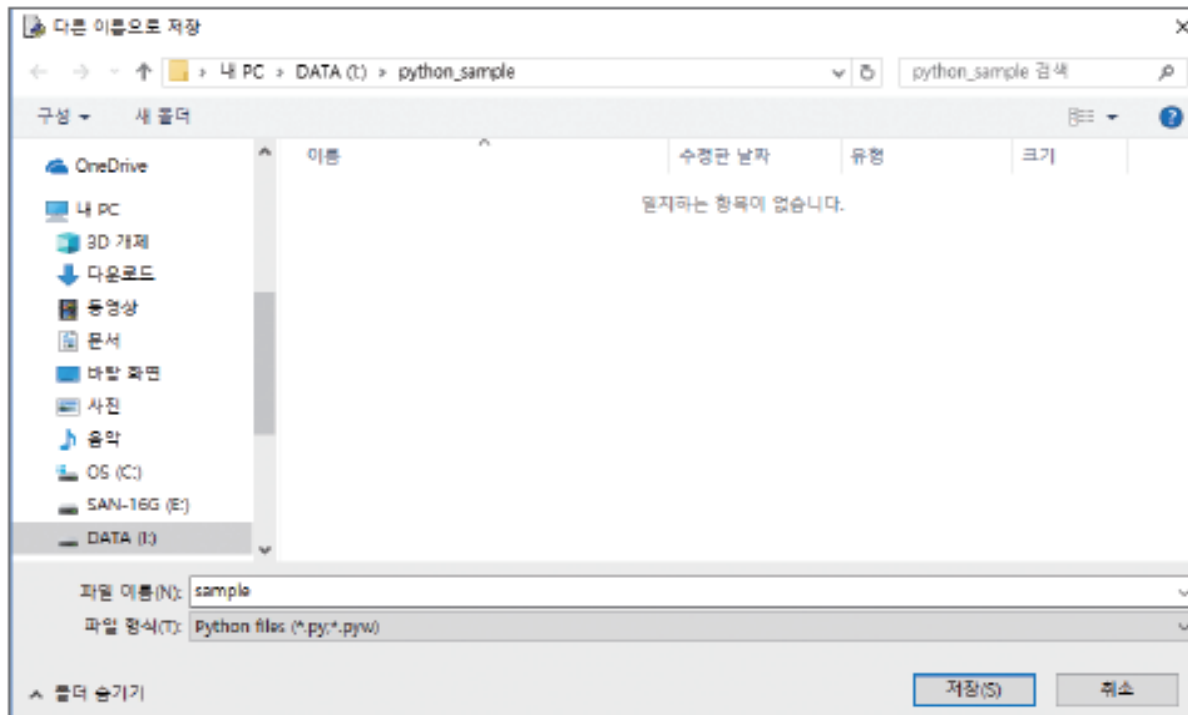
# 텍스트 에디터 사용하기 : 파이썬 IDLE 에디터

## 4) [File] – [Save] 메뉴 선택



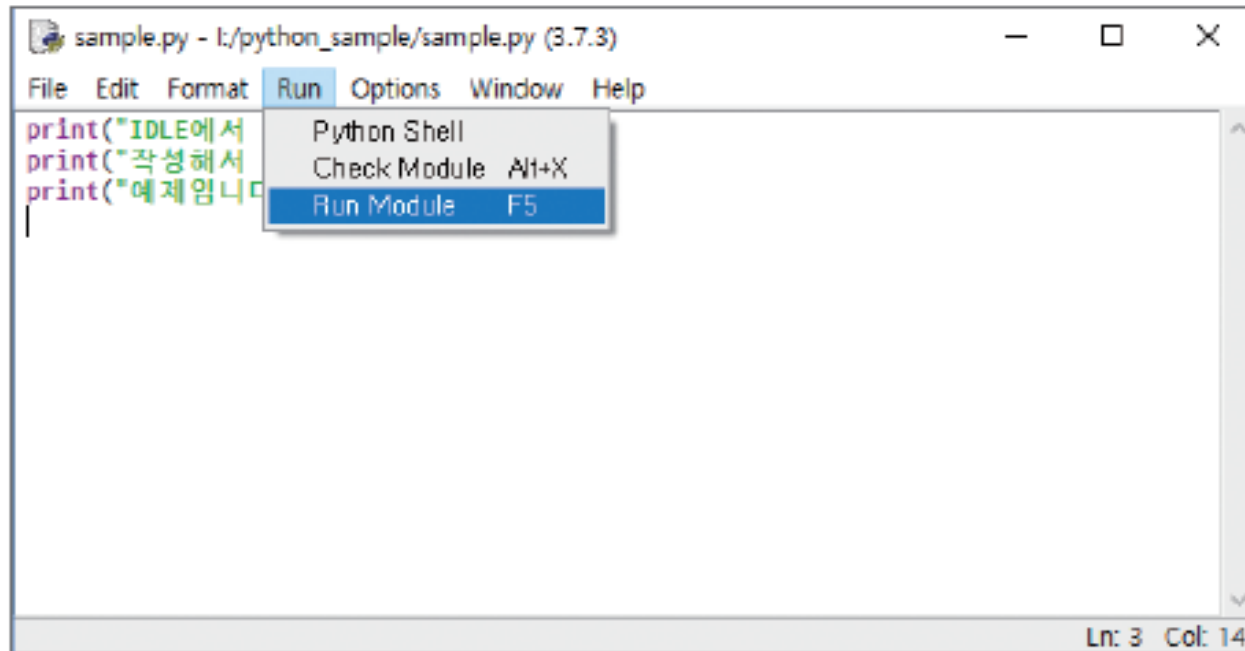
## 텍스트 에디터 사용하기 : 파이썬 IDLE 에디터

5) [다른 이름으로 저장] 대화상자에서 파일 이름을 'sample'로 저장



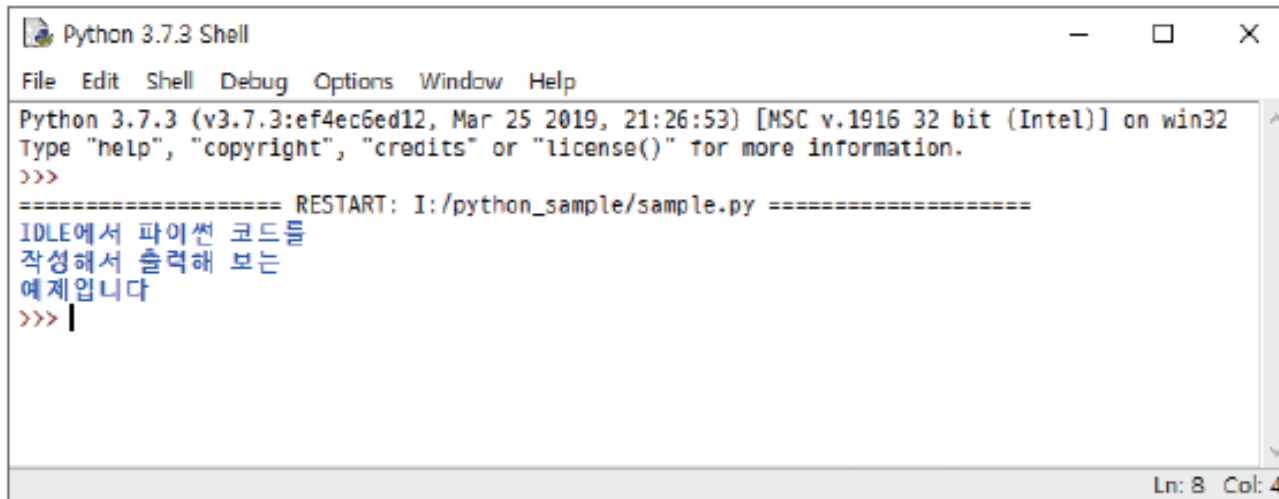
## 텍스트 에디터 사용하기 : 파이썬 IDLE 에디터

6) [Run] – [Run Module] 메뉴 선택(혹은 [F5] 단축키)



# 텍스트 에디터 사용하기 : 파이썬 IDLE 에디터

## 7) 파이썬 코드가 실행됨



The screenshot shows a 'Python 3.7.3 Shell' window. The menu bar includes 'File', 'Edit', 'Shell', 'Debug', 'Options', 'Window', and 'Help'. The main text area displays the following content:

```
Python 3.7.3 (v3.7.3:ef4ec6ed12, Mar 25 2019, 21:26:53) [MSC v.1916 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: I:/python_sample/sample.py =====
IDLE에서 파이썬 코드들
작성해서 출력해 보는
예제입니다
>>> |
```

The status bar at the bottom right indicates 'Ln: 8 Col: 4'.

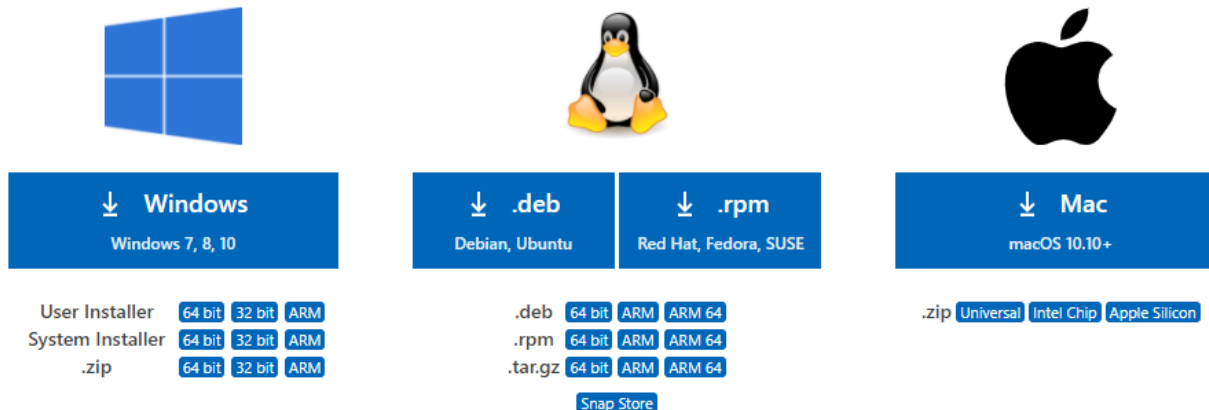


# 텍스트 전문 에디터 사용하기 : 비주얼 스튜디오 코드

- 비주얼 스튜디오 코드 공식 다운로드 사이트

[1] 홈페이지 (<https://code.visualstudio.com/download>) 에서 다운로드

Free and built on open source. Integrated Git, debugging and extensions.

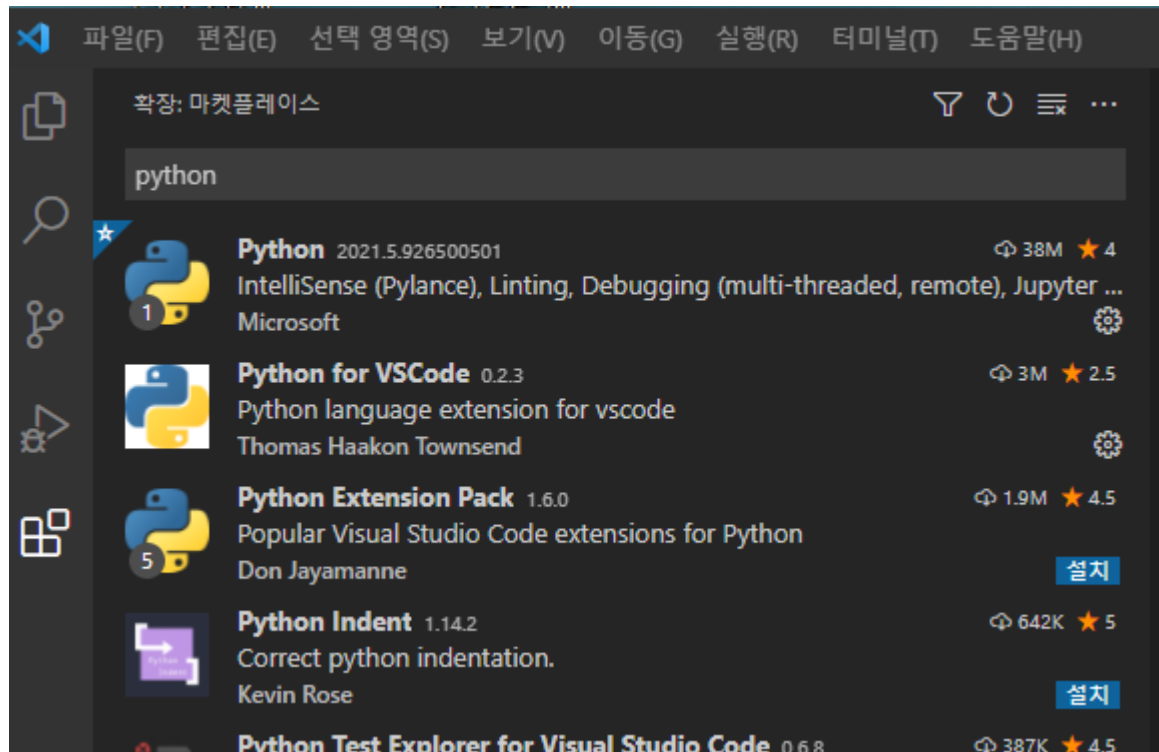


The image shows the download section of the Visual Studio Code website. It is organized into three main columns for different operating systems: Windows, Linux, and Mac. Each column features a platform icon at the top, followed by a blue button with a download arrow and the platform name. Below these buttons are links to various installer formats and architectures.

| Platform | Download Options                                                                                                                           |
|----------|--------------------------------------------------------------------------------------------------------------------------------------------|
| Windows  | User Installer (64 bit, 32 bit, ARM), System Installer (64 bit, 32 bit, ARM), .zip (64 bit, 32 bit, ARM)                                   |
| Linux    | .deb (Debian, Ubuntu) (64 bit, ARM, ARM 64), .rpm (Red Hat, Fedora, SUSE) (64 bit, ARM, ARM 64), .tar.gz (64 bit, ARM, ARM 64), Snap Store |
| Mac      | .zip (Universal, Intel Chip, Apple Silicon)                                                                                                |

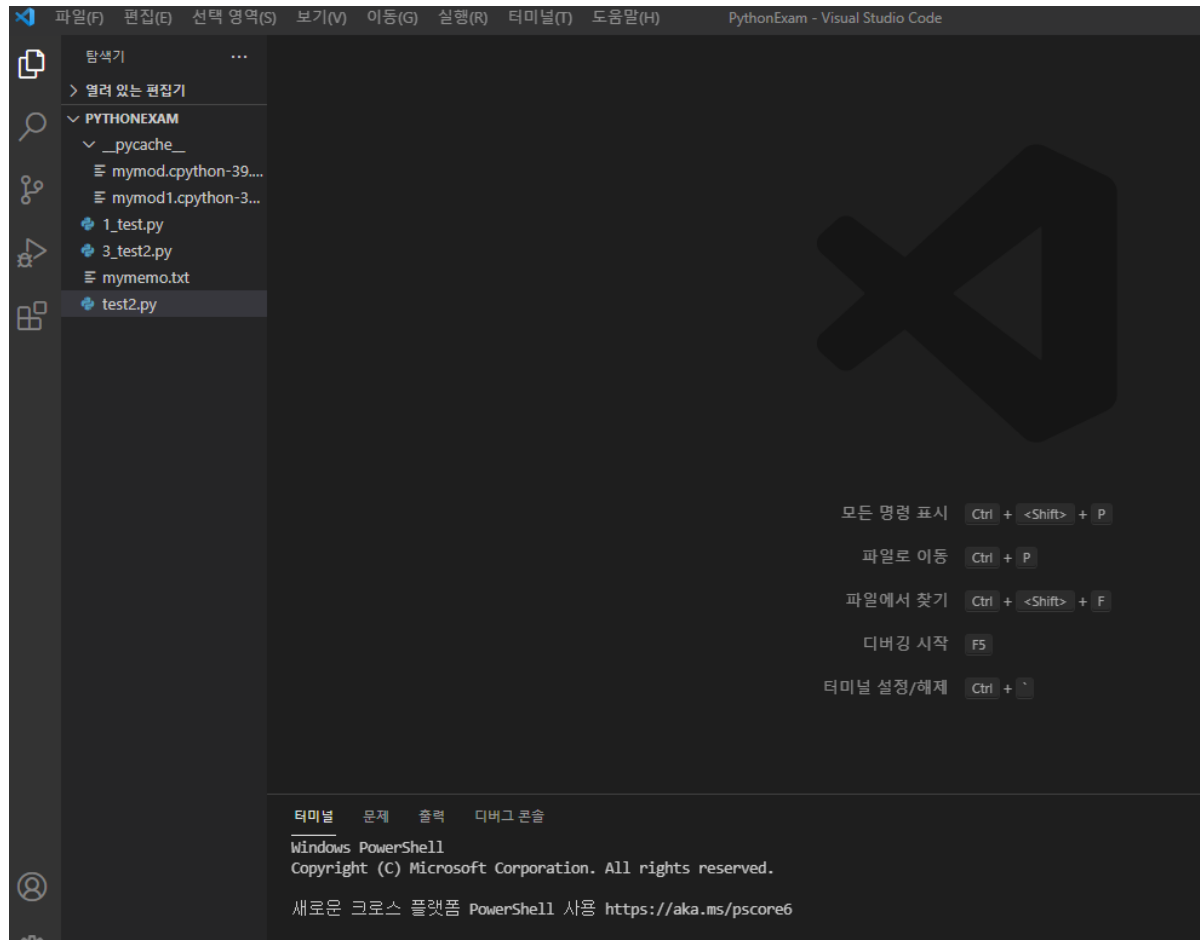
# 텍스트 전문 에디터 사용하기 : 비주얼 스튜디오 코드

## [2] 비주얼 스튜디오 코드- 파이썬 플러그인 설치



# 텍스트 전문 에디터 사용하기 : 비주얼 스튜디오 코드

## [3] PythonExam 작업 폴더 생성 후 폴더 열기]



# 파이썬 예약어

- 키워드 (keyword)

- 특별한 의미가 부여된 단어
- 파이썬에서 이미 특정 의미로 사용하기로 예약해 놓은 것
- 프로그래밍 언어에서 이름 정할 때 똑같이 사용할 수 없음

|       |        |          |       |        |          |
|-------|--------|----------|-------|--------|----------|
| False | None   | True     | and   | as     | assert   |
| break | class  | continue | def   | del    | elif     |
| else  | except | finally  | for   | from   | global   |
| if    | import | in       | is    | lambda | nonlocal |
| not   | or     | pass     | raise | return | try      |
| while | with   | yield    |       |        |          |

- 대소문자 구별

# 식별자

- 식별자 (identifier)

- 프로그래밍 언어에서 이름 붙일 때 사용하는 단어
- 변수 또는 함수 이름 등으로 사용
- 키워드 사용 불가
- 특수문자는 언더바(\_)만 허용
- 숫자로 시작 불가
- 공백 포함 불가
- 알파벳 사용이 관례
- 의미 있는 단어로 할 것

| 사용 가능한 단어                                    | 사용 불가능한 단어                                                                        |
|----------------------------------------------|-----------------------------------------------------------------------------------|
| alpha<br>alpha10<br>_alpha<br>AlpHa<br>ALPHA | break → 키워드라서 안 됩니다.<br>273alpha → 숫자로 시작해서 안 됩니다.<br>has space → 공백을 포함해서 안 됩니다. |

# 주석

- 주석 (comment)

- 프로그램 진행에 영향 주지 않는 코드
- 프로그램 설명 위해 사용
- # 기호를 주석으로 처리하고자 하는 부분 앞에 붙임 -> 한 줄 주석

```
>>> # 간단히 출력하는 예입니다.
>>> print("Hello! Python Programming...") # 문자열을 출력합니다.
Hello! Python Programming...
```

→ # 기호 뒷부분이 주석 처리됩니다.

- """ 기호를 3번 붙이거나 '''3번 붙임 -> 여러 줄 주석

```
myStr = 'Hello python'
myChar = 'a'
myInt = 123
myFloat = 3.14
myBool = True

print('myStr= ', myStr)
#print('myChar= ', myChar)
print('myInt= ', myInt)
"""
print('myFloat= ', myFloat)
print('myBool= ', myBool)
"""

===== RESTART: C
t.py =====
myStr= Hello python
myChar= a
myInt= 123
myFloat= 3.14
myBool= True
>>>
===== RESTART: C
t.py =====
myStr= Hello python
myInt= 123
myFloat= 3.14
myBool= True
>>>
===== RESTART: C
t.py =====
myStr= Hello python
myInt= 123
>>>
===== RESTART: C
t.py =====
myStr= Hello python
myInt= 123
>>>|
```

## 출력 : print()

- 여러 개 출력하기

```
>>> print(52, 273, "Hello")
52 273 Hello
>>> print("안녕하세요", "저의", "이름은", "윤인성입니다!")
안녕하세요 저의 이름은 윤인성입니다!
```

- 줄바꿈하기

```
>>> print()
 → 빈 줄을 출력합니다.
>>>
```

## 출력 : print()

- 예시 - 기본 출력

```
01 # 하나만 출력합니다.
02 print("# 하나만 출력합니다.")
03 print("Hello Python Programming...!")
04 print()
05
06 # 여러 개를 출력합니다.
07 print("# 여러 개를 출력합니다.")
08 print(10, 20, 30, 40, 50)
09 print("안녕하세요", "저의", "이름은", "윤인성입니다!")
10 print()
11
12 # 아무것도 입력하지 않으면 단순히 줄바꿈합니다.
13 print("# 아무것도 출력하지 않습니다.")
14 print("— 확인 전용선 —")
15 print()
16 print()
17 print("— 확인 전용선 —")
```



# 출력 : print()

- 비주얼 스튜디오 코드에서 실행

```
hello.py x
1 # 하나만 출력합니다.
2 print('# 하나만 출력합니다.')
3 print("Hello Python Programming..!")
4 print()
5
6 # 여러 개를 출력합니다.
7 print('# 여러 개를 출력합니다.')
8 print(10,20,30,40,50)
9 print('안녕하세요 ' + '저의 ' + '이름은 ' + '염희균입니다!')
10 print()
11
12 # 아무 것도 입력하지 않으면 단순히 줄바꿈 합니다.
13 print('# 아무 것도 입력하지 않으면 단순히 줄바꿈 합니다.')
14 print('--- 확인 전용선---')
15 print()
16 print()
17 print('--- 확인 전용선---')
18
```

```
hello x
C:\Users\HeeGyun\PycharmProjects\HelloPycharm\venv\Scripts\
하나만 출력합니다.
Hello Python Programming..!

여러 개를 출력합니다.
10 20 30 40 50
안녕하세요 저의 이름은 염희균입니다!

아무 것도 입력하지 않으면 단순히 줄바꿈 합니다.
--- 확인 전용선---

--- 확인 전용선---

Process finished with exit code 0
```

## 확인문제

- 파이썬 IDLE 에디터 실행 후 다음 명령을 입력했을 때 나오는 결과를 빈칸에 적어 보세요.

```
>>> print("Hello Python")
```

- 파이참 에디터에서 다음 소스 코드를 입력하고 ex01.py로 저장한 후 실행해 나오는 결과물을 빈칸에 적어 보세요.

```
print("Hello! " * 3)
print("혼자 공부하다 모르면 동영상 강의를 참고하세요!")
```

