

파이썬 (Python)

강사 : 이승태



파이썬이란 무엇인가?

...

01-1 파이썬이란?

01-2 파이썬의 특징

01-3 파이썬으로 무엇을 할 수 있을까?

01-4 파이썬 설치하기

01-5 파이썬 둘러보기

01-6 파이썬과 에디터

▶ 파이썬(Python)

- ▶ 1990년 암스테르담의 귀도 반 로섬(Guido Van Rossum)이 개발한 인터프리터 언어
- ▶ 파이썬의 사전적 의미
 - ▶ 고대 신화에 나오는 파르나소스 산의 동굴에 살던 큰 뱀
- ▶ 구글에서 만든 소프트웨어의 50% 이상이 파이썬으로 작성됨
- ▶ 인스타그램(Instagram), 넷플릭스(Netflix), 아마존(Amazon) 등에서 사용
- ▶ 공동 작업과 유지 보수가 매우 쉽고 편리함

- ▶ 파이썬은 인간다운 언어이다
- ▶ 파이썬은 사람이 생각하는 방식을 그대로 표현할 수 있는 언어

```
if 4 in [1, 2, 3, 4]: print("4가 있습니다")
```



만약 4가 1, 2, 3, 4 중에 있으면 '4가 있습니다'를 출력한다.

▶ 문법이 쉬워 빠르게 배울 수 있다

- ▶ 문법 자체가 아주 쉽고 간결하며 사람의 사고 체계와 매우 닮아 있음
- ▶ 유명한 프로그래머인 에릭 레이먼드(Eric Raymond)는 공부한 지 단 하루 만에 자신이 원하는 프로그램을 작성!
- ▶ 무료이지만 강력하다
- ▶ 오픈 소스 → 무료로 언제 어디서든 파이썬을 다운로드하여 사용 가능
- ▶ 파이썬과 C는 찰떡 궁합
 - ▶ 프로그램의 전반적인 뼈대는 파이썬으로 만들고,
빠른 실행 속도가 필요한 부분은 C로 만들어서 파이썬 프로그램 안에 포함

▶ 간결하다

펄(Perl)	파이썬
100가지 방법으로 1가지 일 처리	가장 좋은 방법 1가지만 사용

- ▶ 프로그램이 실행되게 하려면
줄(들여쓰기)을 반드시 맞추어야 함

▶ 가독성 ↑

- ▶ 프로그래밍을 즐기게 해 준다
 - ▶ 프로그래머가 만들고자 하는 기능에만 집중할 수 있음
- ▶ 개발 속도가 빠르다

"Life is too short, You need Python."

인생은 너무 짧으니 파이썬이 필요해.

- ▶ 파이썬의 엄청나게 빠른 개발 속도를 두고 유행처럼 퍼진 말

▶ 파이썬으로 할 수 있는 일

▶ 웹 프로그래밍

- ▶ 파이썬은 웹 프로그램을 만들기에 매우 적합한 도구
- ▶ 파이썬으로 제작한 웹 사이트는 셀 수 없이 많을 정도

▶ 인공지능과 머신러닝

- ▶ 자연어 처리, 음성 인식, 이미지 인식과 같은 인공지능 기술 구현
- ▶ 인공지능과 머신러닝 프로그래밍을 쉽게 할 수 있도록 여러 라이브러리 구현

▶ 수치 연산 프로그래밍

- ▶ C로 작성된 넘파이(Numpy) 수치 연산 모듈을 통해 빠른 수치 연산 가능

▶ 파이썬으로 할 수 있는 일

▶ 데이터 분석

- ▶ NumPy, Pandas, Matplotlib 등의 라이브러리를 활용해 데이터 처리, 통계 분석, 시각화를 손쉽게 수행
- ▶ 판다스가 등장한 이래 데이터 분석에 R보다 파이썬을 사용하는 사례 증가

▶ 데이터베이스 프로그래밍

- ▶ Sybase, Infomix, Oracle, MySQL, PostgreSQL 등의 데이터베이스에 접근하기 위한 도구 제공
- ▶ 자료를 변형 없이 그대로 파일에 저장하고 불러오는 파이썬 모듈 피클(pickle)

▶ 시스템 유틸리티 제작하기

- ▶ 운영체제(윈도우, 리눅스 등)의 시스템 명령어를 사용하는 도구를 통한 시스템 유틸리티 제작

▶ 파이썬으로 할 수 있는 일

▶ **GUI(Graphic User Interface) 프로그래밍**

- ▶ 화면에 윈도우 창을 만들고 프로그램을 동작시킬 수 있는 메뉴나 버튼, 그림 등을 추가하는 것
- ▶ GUI 프로그래밍을 위한 도구들을 갖추고 있어, GUI 프로그램을 만들기 쉬움 (예 - Tkinter(티케이인터))

▶ **C/C++와 결합하기**

- ▶ C나 C++로 만든 프로그램을 파이썬에서, 파이썬으로 만든 프로그램을 C나 C++에서 사용 가능

▶ **사물 인터넷**

- ▶ 라즈베리파이를 제어하며 사물 인터넷 구현

▶ 파이썬으로 할 수 없는 일

▶ 시스템과 밀접한 프로그래밍 영역

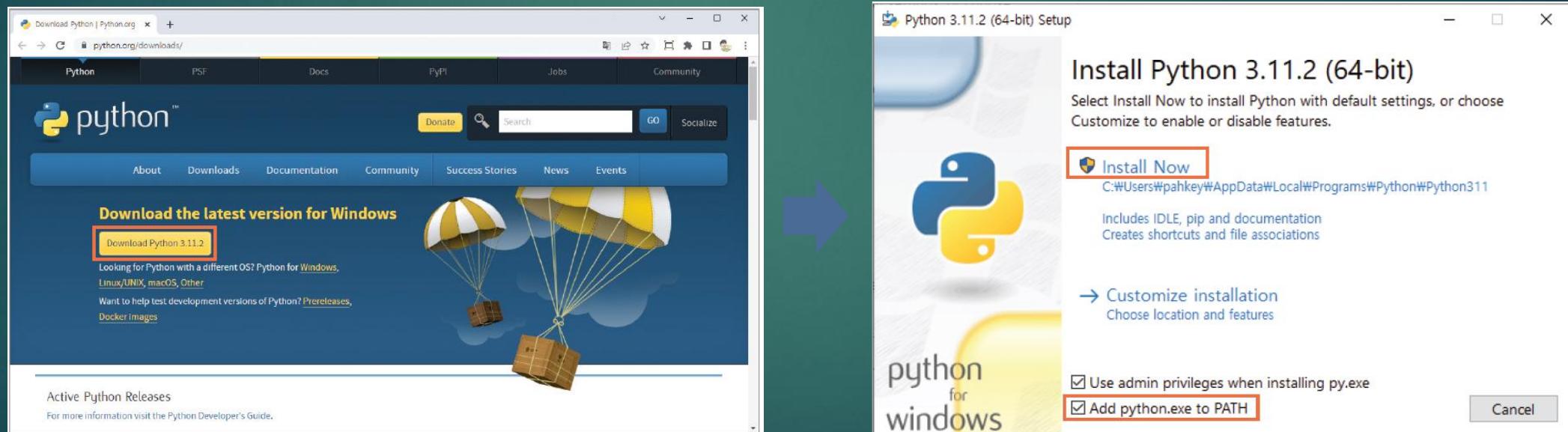
- ▶ 운영체제, 엄청난 횟수의 반복과 연산이 필요한 프로그램 등
매우 빠른 속도를 요구하거나 하드웨어를 직접 건드려야 하는 프로그램에는 적합하지 않음

▶ 모바일 프로그래밍

- ▶ 안드로이드 네이티브 앱(android native app) 개발에는 아직 역부족
- ▶ 아이폰 앱 개발은 할 수 없음

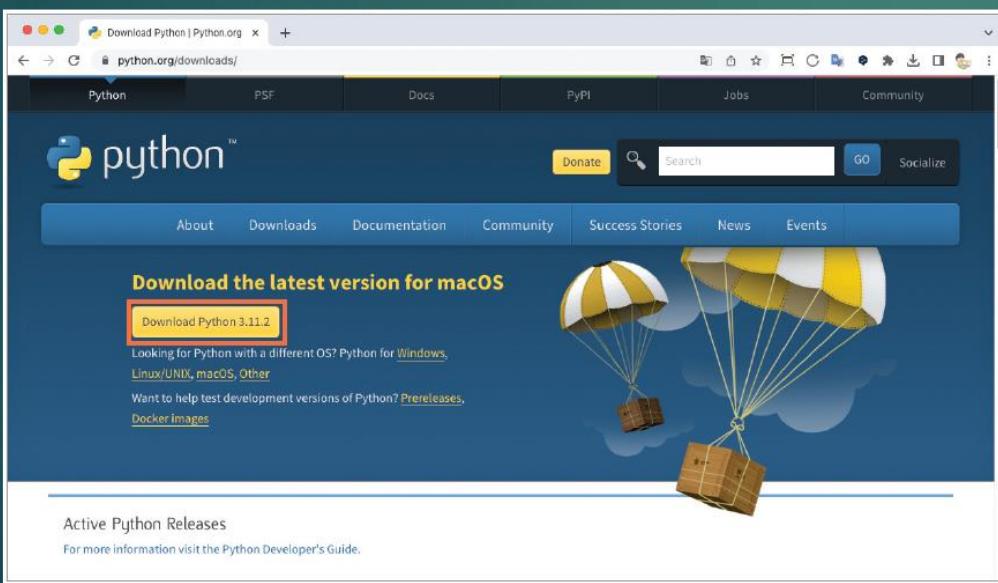
▶ 윈도우에서 파이썬 설치하기

1. 파이썬 공식 홈페이지 (www.python.org/downloads/)에서 Python 3.x 최신 버전 다운로드
2. 파이썬이 어느 곳에서든지 실행될 수 있도록 '**Add python.exe to PATH**' 옵션 선택



▶ 맥에서 파이썬 설치하기

1. 파이썬 공식 홈페이지 (www.python.org)의 [Downloads] 메뉴에서 맥용 설치 파일 다운로드
2. 내려받은 **python-3.11.x-macos1.pkg** 파일을 실행하여 설치
3. 터미널에서 명령을 입력해 자신의 맥에 설치된 파이썬 버전 확인



pahkey@mymac ~ % **python3 -V** ━━━━━━━━━━━━━━━━ 직접 입력해 보자.

Python 3.11.2

```
Last login: Sat Apr 15 10:13:55 on ttys000
pahkey@mymac ~ % python3 -V
Python 3.11.2
pahkey@mymac ~ %
```

▶ 파이썬 기초 실습 준비하기

▶ 파이썬 대화형 인터프리터 실행

- ▶ [시작] → 'python'을 검색 → 'Python 3.11(64-bit)'을 찾아 실행
※ 맥 사용자는 터미널에서 'python3'을 입력



▶ 파이썬 대화형 인터프리터(파이썬 셀)

- ▶ 사용자가 입력한 소스 코드를 실행하는 환경

```
Python 3.11.2 (tags/v3.11.2:878ead1, Feb 7 2023, 16:38:35) [MSC v.1934 64 bit (AMD64)] on win32
```

```
Type "help", "copyright", "credits" or "license" for more information.
```

>>> ━━━━━━ 프롬프트(>>>)가 있는 곳이 입력하는 부분

▶ 파이썬 기초 실습 준비하기

▶ 수식 입력

▶ 입력에 따른 결괏값이 바로 출력됨

```
>>> 1 + 1 ← 수식 입력  
2 ← 결괏값 바로 출력  
>>>
```

■ 대화형 인터프리터 종료

- 1) Ctrl + Z → Enter
- 2) 내장함수 quit()이나 exit()를 사용
- 3) sys 모듈 사용

```
>>> import sys  
>>> sys.exit()
```

▶ 파이썬 기초 문법 따라 해 보기

▶ 사칙 연산

▶ 덧셈(+), 뺄셈(-), 나눗셈(/), 곱셈(*)

```
>>> 1 + 2
3
```

```
>>> 3 / 2.4
1.25
>>> 3 * 9
27
```

▶ 변수에 숫자 대입하고 계산하기

```
>>> a = 1
>>> b = 2
>>> a + b
3
```

▶ 변수에 문자 대입하고 출력하기

```
>>> a = "Python"
>>> print(a)
Python
```

```
>>> a
'Python'
```

▶ 조건문 if

```
>>> a = 3
>>> if a > 1:
...     print("a는 1보다 큽니다.")
...
a는 1보다 큽니다.
```

들여쓰기(Spacebar) 4칸
Enter 입력

▶ 파이썬 기초 문법 따라 해 보기

▶ 반복문 for

- ▶ 대괄호([]) 사이에 있는 값을 하나씩 출력

```
>>> for a in [1, 2, 3]:  
...     print(a) ↓  
... ↓  
1  
2  
3
```

▪ 반복문 while

- ▶ 일정한 조건이 만족하는 동안 문장을 반복 수행

```
>>> i = 0  
>>> while i < 3:  
...     i = i + 1 ↓  
...     print(i) ↓  
... ↓  
1  
2  
3
```

▶ 파이썬 기초 문법 따라 해 보기

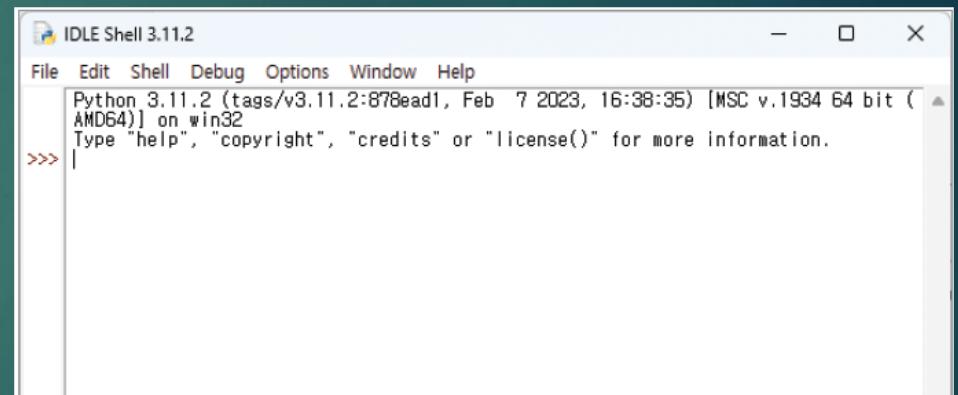
▶ 함수

```
>>> def add(a, b):  
...     return a + b  
...  
>>> add(3, 4)  
7
```

- ▶ 파이썬에서 함수를 정의할 때 사용하는 예약어 `def`
※ 예약어: 프로그래밍 언어에서 이미 문법적인 용도로 사용하고 있는 단어
- ▶ `return`은 함수에서 값을 반환할 때 사용

01-6 파일과 에디터

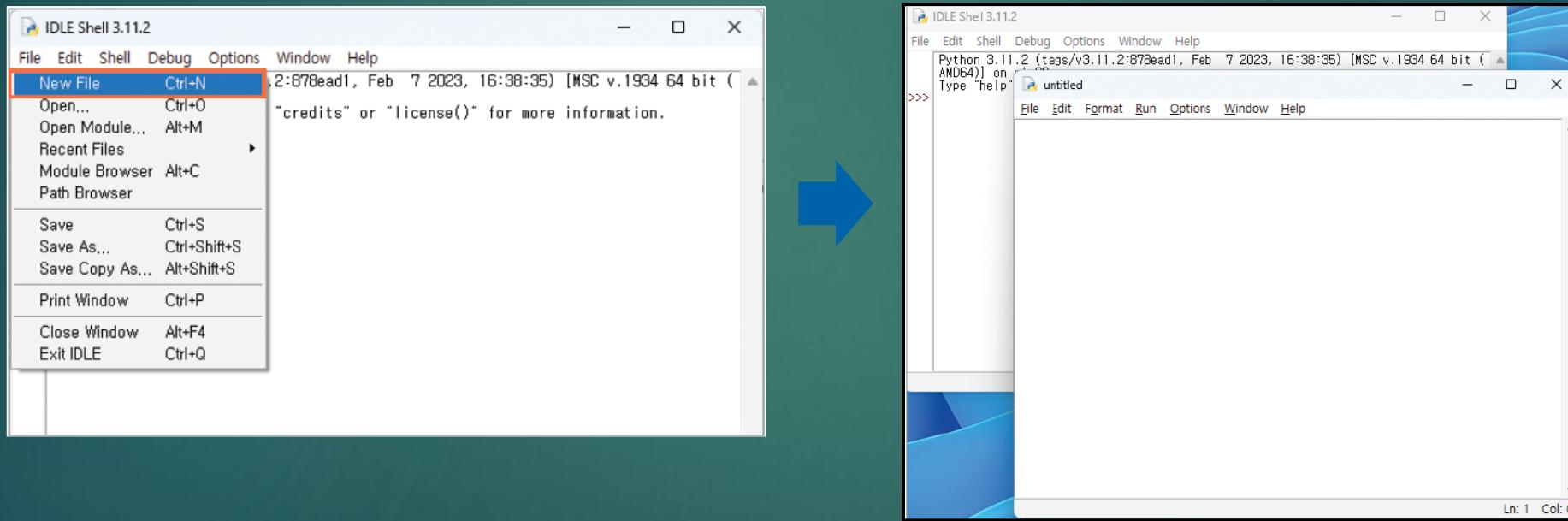
- ▶ IDLE 에디터로 파일 프로그램 작성하기
- ▶ 파일 IDLE 실행
 - ▶ IDLE(Integrated Development and Learning Environment)
 - ▶ 파일 프로그램 작성 도와주는 통합 개발 환경
 - ▶ [시작] → 'idle' 검색



01-6 파일과 에디터

- ▶ IDLE 에디터로 파일 프로그램 작성하기

- ▶ IDLE 에디터(Editor) 실행



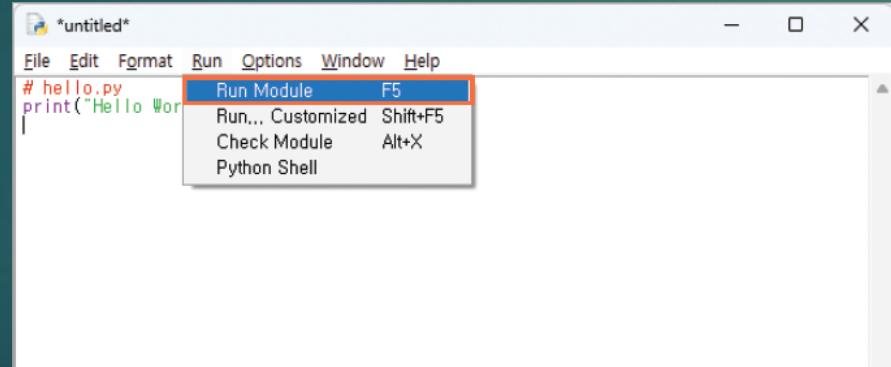
01-6 파일과 에디터

▶ IDLE 에디터로 파일 프로그램 작성하기

1)

```
# hello.py  
print("Hello World")
```

2)



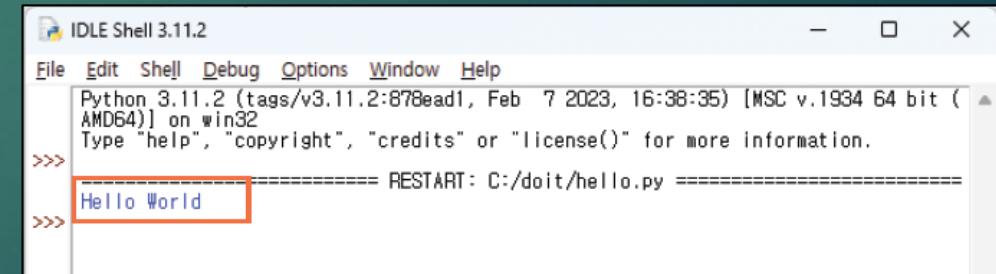
3) 파일 저장

- 경로) C:\doit\hello.py



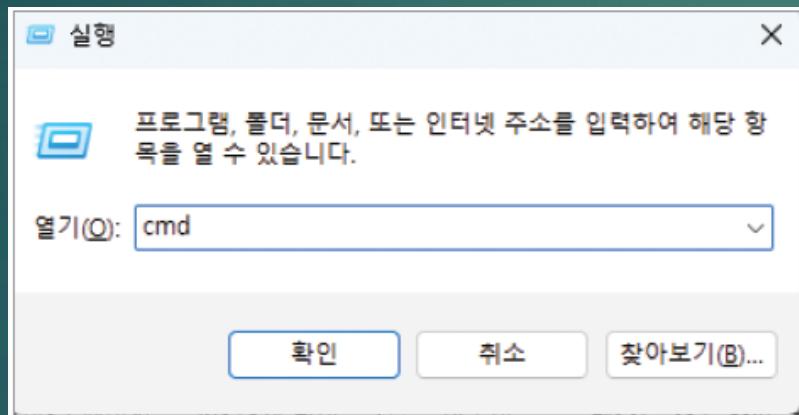
4) 실행 결과

- IDLE 셸 창에 표시됨



▶ 명령 프롬프트 창에서 파이썬 프로그램 실행하기

- 1) 명령 프롬프트 창(command prompt) 열기
▶ 윈도우 키 + R → 'cmd' 입력 후 Enter



- 2) hello.py 저장한 경로로 이동하여 실행

A screenshot of a Windows Command Prompt window titled '명령 프롬프트'. It shows the following text:
Microsoft Windows [Version 10.0.22621.1413]
(c) Microsoft Corporation. All rights reserved.
C:\Users\박 응용>cd c:\doit
C:\doit>python hello.py
Hello World
C:\doit>

01-6 파일과 에디터

▶ 추천하는 에디터

▶ 비주얼 스튜디오 코드(code.visualstudio.com)

The screenshot shows the Visual Studio Code interface. On the left is the Explorer sidebar with 'simple.py' listed. The main editor area contains the following Python code:

```
language = ['python', 'perl', 'c', 'java']
for lang in languages:
    if lang in ['python', 'perl']:
        print("%s need interpreter" % lang)
    elif lang in ['c', 'java']:
        print("%s need compiler" % lang)
    else:
        print("should not reach here")
```

Below the editor is the Terminal tab, which displays the output of running the script:

```
PS C:\dev\jptpy2> & C:/Users/user/AppData/Local/Programs/Python/Python311/python.exe c:/dev/jptpy2/simple.py
python need interpreter
perl need interpreter
c need compiler
java need compiler
PS C:\dev\jptpy2>
```

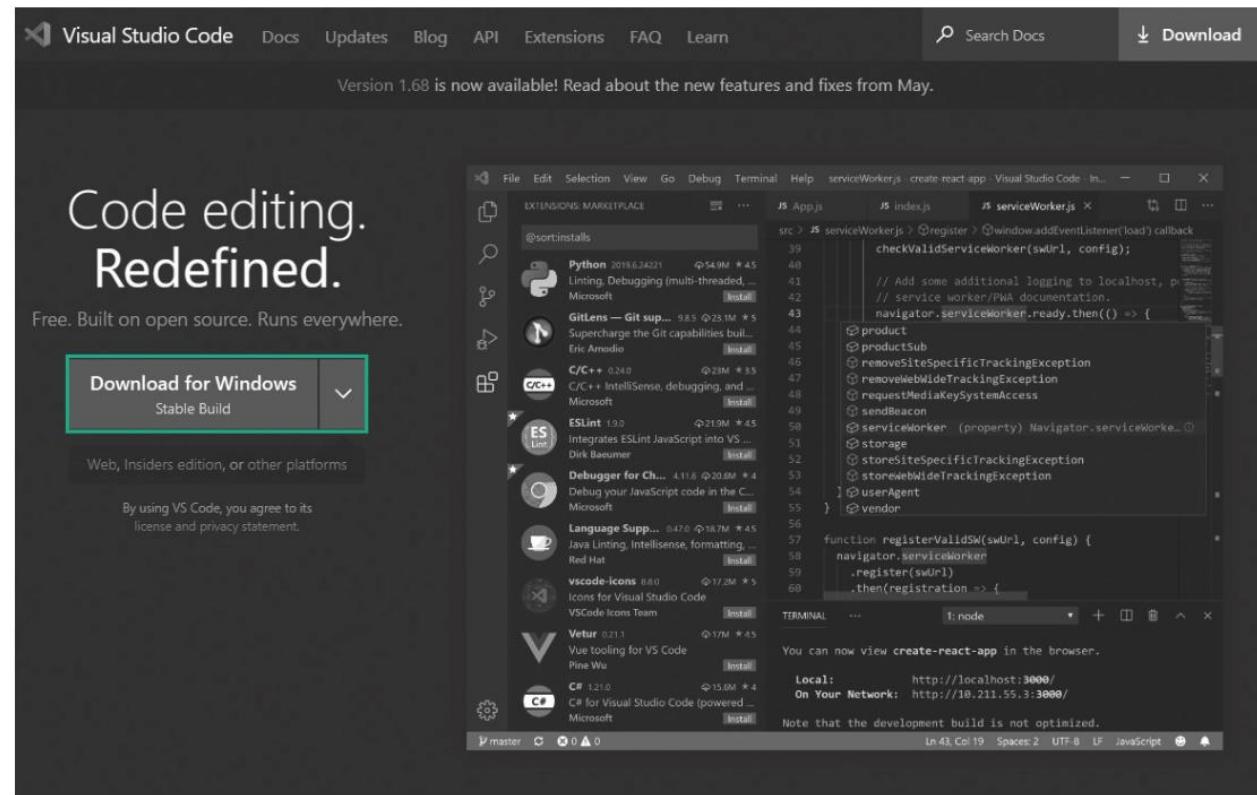
The status bar at the bottom shows the file is 10 lines long, has 6 errors, and is in Python mode.

▶ 파일

(www.jetbrains.com/pycharm/download)

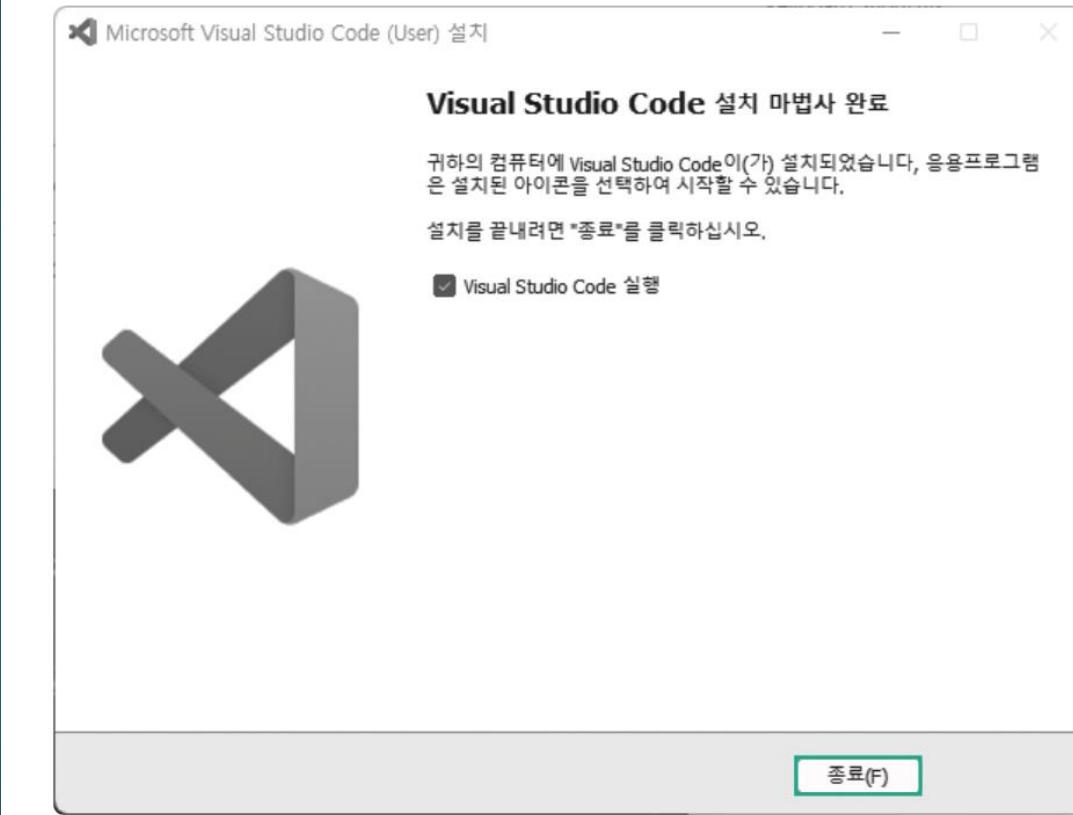
01-6 코딩 준비하기

1. VS 코드 공식 홈페이지(<https://code.visualstudio.com/>)에 접속해서 설치 파일을 내려받습니다. 자신의 운영체제(여기서는 윈도우)에 맞는 내려받기 링크를 자동으로 만들어 주므로 [Download for Windows(Stable Build)]를 클릭하면 됩니다.



01-6 코딩 준비하기

2. 이후 과정에서 따로 설정할 필요가 없으므로 진행 순서에 따라 [다음]과 [설치]를 클릭합니다. 그리고 [종료]를 클릭하여 VS 코드를 실행합니다.



설치 중 <추가 작업 선택> 단계에서 '바탕 화면에 바로가기 만들기'를 선택하면 실습할 때 좀 더 편리하게 실행할 수 있습니다.

01-6 코딩 준비하기

P27. 확장 기능 설치하기

vs 코드는 다양한 프로그래밍 언어를 지원하므로 파이썬을 사용한다고 알려야 한다.

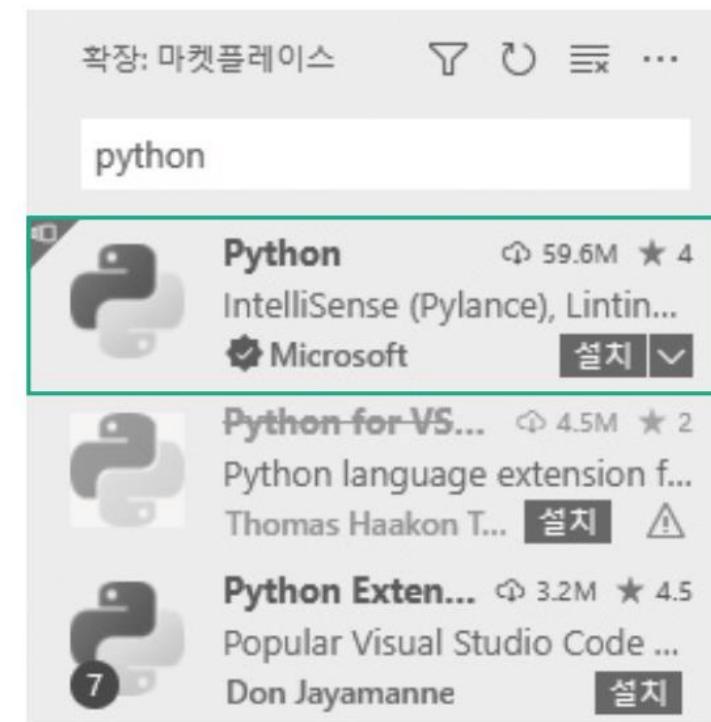
다음 과정을 따라 하며 vs 코드에 확장 기능을 설치해보자.

1. VS 코드를 실행하면 환영 메시지와 함께 튜토리얼, 파일 등의 링크를 표시합니다. 화면
왼쪽 위에 있는 아이콘 중 가장 밑에 있는 [확장] 아이콘을 클릭해 봅시다.



01-6 코딩 준비하기

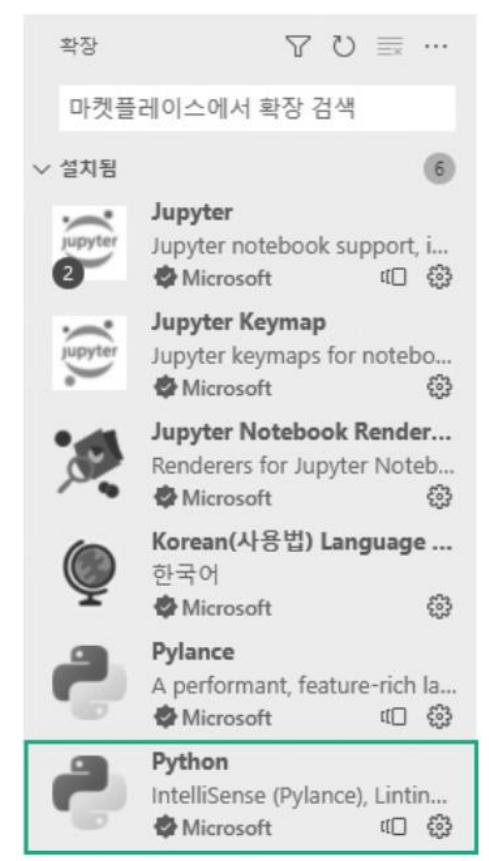
2. [확장] 아이콘을 클릭하면 [확장] 패널이 나타납니다. 검색창에 ‘python’을 입력하면 다음과 같은 목록이 등장합니다. Python 항목의 [설치] 버튼을 클릭합니다.



[확장] 패널은 사용자의 운영체제나 버전에 따라 다르게 보일 수 있습니다.

01-6 코딩 준비하기

3. Python이 설치되었다면 [설치됨] 목록에 다음과 같이 [Python]이 표시됩니다.

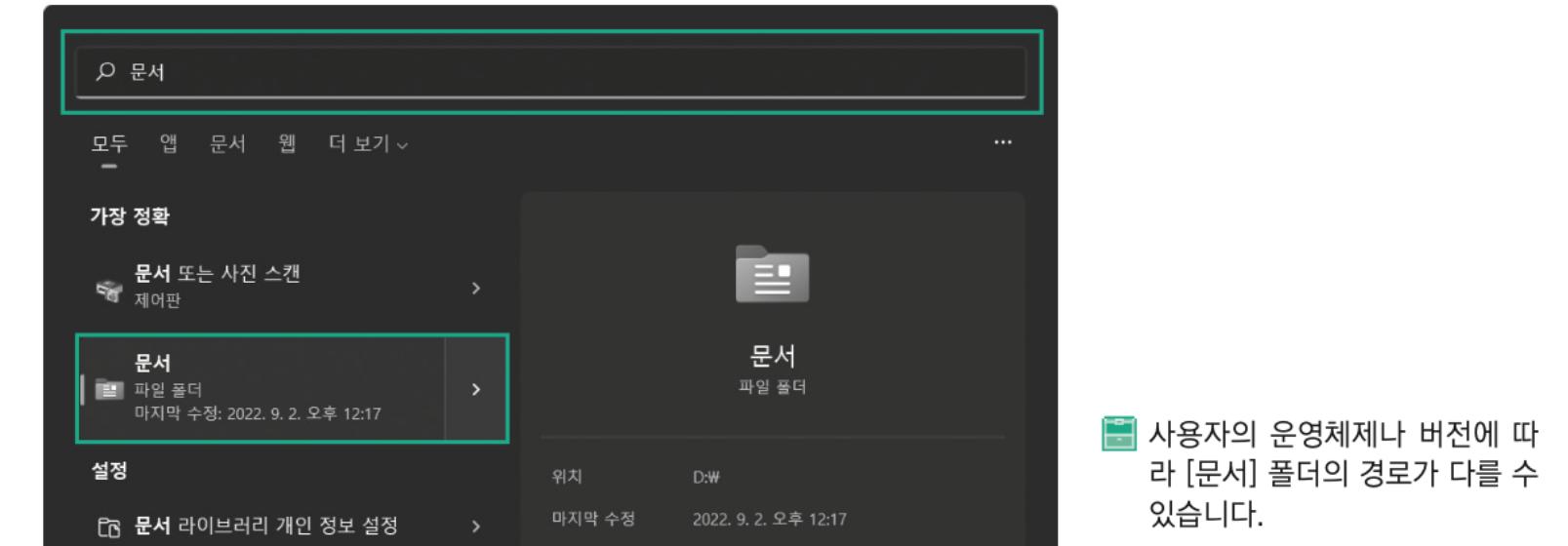


나의 첫 파이션 프로그램 만들기

P29. 작업 폴더 만들기

일반적으로 모든 프로젝트를 저장하는 주 폴더를 만들고 그 아래에 하위 폴더를 두어 개별 프로젝트나 애플리케이션을 저장한다.

1. 윈도우 아래 창문 모양의 [시작] 버튼 을 클릭하여 ‘문서’를 검색합니다. 그리고 [문서]를 클릭합니다.



나의 첫 파이션 프로그램 만들기

2. 문서 폴더에서 [새로 만들기 → 폴더]를 클릭하여 새 폴더를 만듭니다. 폴더 이름은 Python으로 합니다.

