

# 단원 01

## 파이썬 언어의 개요와 개발 환경 설치

인공지능소프트웨어학과

강환수 교수



## Section 1. 파이썬의 개요

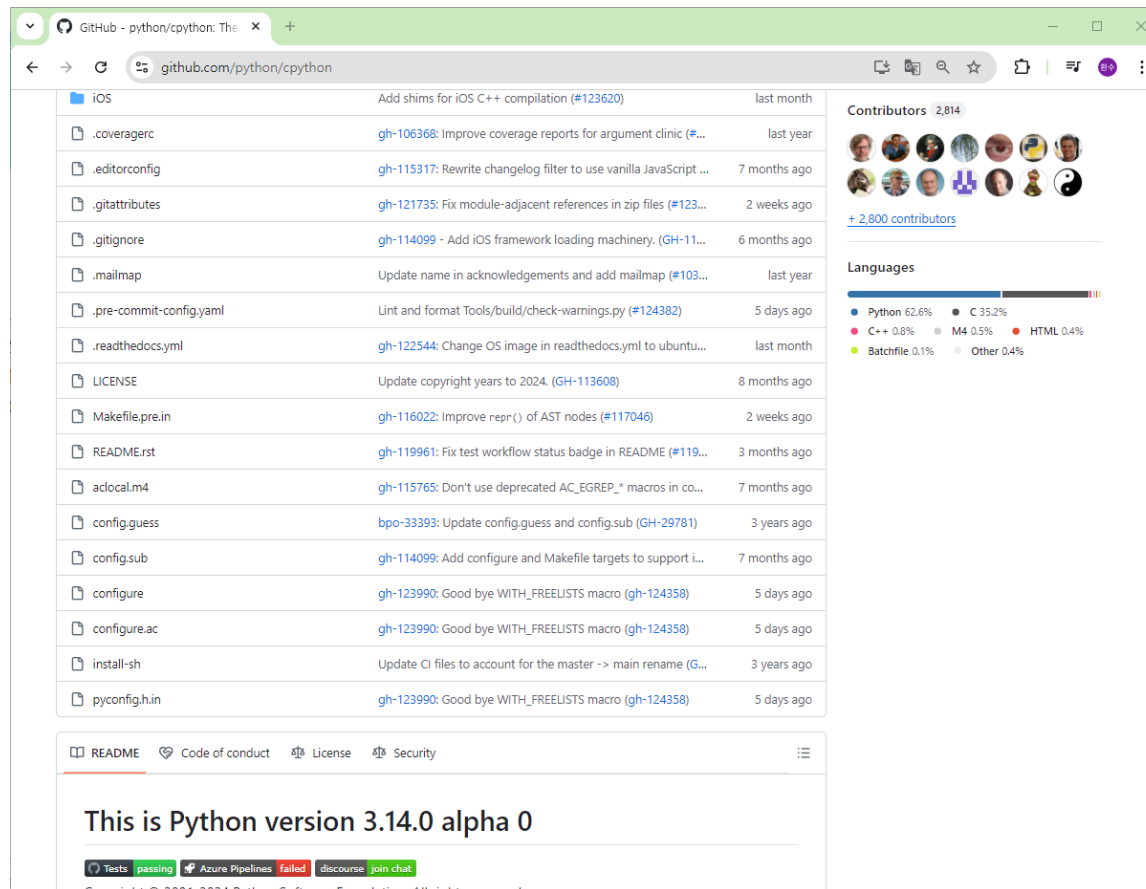
-



# 파이썬 언어의 개요

배우기 쉽고 강력한 프로그래밍 언어

- 1991년 네덜란드의 귀도 반 로섬(Guido van Rossum)이 개발
  - 무료이자 오픈 소스(open source) 소프트웨어
    - 깃허브 사이트([www.github.com/python/cpython](https://www.github.com/python/cpython))에 모든 소스가 공개



## 쉽고 강력한 언어

- 누구나 배우기 쉬운 언어

- 첫째, 간결하고 배우기 쉽다
- 둘째, 무료이고 생산성이 높다
- 셋째, 강력한 라이브러리를 제공한다
- 넷째, 다양한 프로그램에 잘 붙는 풀 언어(glue language)
  - 다른 언어로 개발하거나 이미 개발되어 쓰이고 있는 모듈들을 연결

- 빅데이터 처리와 인공지능 등 다양한 분야에 적합한 언어

- 교육과 학술, 실무 등 다양한 분야에 사용
- 인공지능 구현과 빅데이터 분석 처리 분야에 사용

- 주요 라이브러리:

- NumPy: 배열 처리 및 수치 계산을 위한 필수적인 라이브러리
- SciPy: 고급 과학 계산 함수를 제공하며, NumPy를 기반으로 구축
- Matplotlib: 데이터 시각화를 위한 라이브러리
- Scikit-learn: 머신 러닝 알고리즘 구현 및 모델 훈련을 위한 라이브러리
- TensorFlow: 딥 러닝 모델 구축 및 학습에 사용되는 라이브러리

## 다양한 플랫폼에서 파이썬 개발 환경 활용

- 파이썬은 다양한 플랫폼에서 실행 가능하며, 무료로 사용
  - 윈도우, macOS, 리눅스, 유닉스 등 주요 운영 체제를 지원하며, 모바일 환경에서도 파이썬을 사용
- 파이썬 개발 환경 종류:
  - 파이썬 셸
    - 간단한 코드를 실행하고 테스트할 수 있는 대화형 환경
  - 파이썬 통합 개발 환경 (IDE)
    - 코드 편집, 디버깅, 실행, 버전 관리 등 다양한 기능을 제공하는 개발 환경
    - PyDev: Eclipse 기반 IDE
    - PyCharm: JetBrains에서 개발한 전문적인 파이썬 IDE
    - Spyder: 과학 계산 및 데이터 분석에 특화된 IDE
    - Jupyter Notebook: 코드, 결과, 설명을 함께 작성할 수 있는 대화형 환경
    - Thonny: 초보자를 위한 간편한 IDE
  - 텍스트 편집기
    - 코드 작성 및 편집에 사용 가능하며, 파이썬 플러그인을 통해 개발 환경 구축 가능
    - Sublime Text, Visual Studio Code, Notepad++, Atom, Vim 등

## 아이파이썬(IPython, [ipython.org](http://ipython.org))

실제 인터프리터로는 CPython이 필요

- 아이파이썬(IPython, [ipython.org](http://ipython.org))

- Jupyter라고도 부름
- 파이썬 기능에서 대화형인 REPL의 기능을 확장한 파이썬
- 기존의 콘솔 환경에서 웹(web)을 기반으로 하는 주피터 노트북(jupyter notebook) 등 클라이언트를 분리하기 위해서 개발된 파이썬
- 특징
  - 강력한 대화형 셸
    - 주피터 노트북 지원
  - 주피터(jupyter) 커널 활용

- 아이파이썬을 쉽게 사용하는 방법

- 아나콘다(anaconda)라는 배포판을 설치해 활용

- Vscode

- 주피터(ipython) 관련 확장 설치해 사용 가능

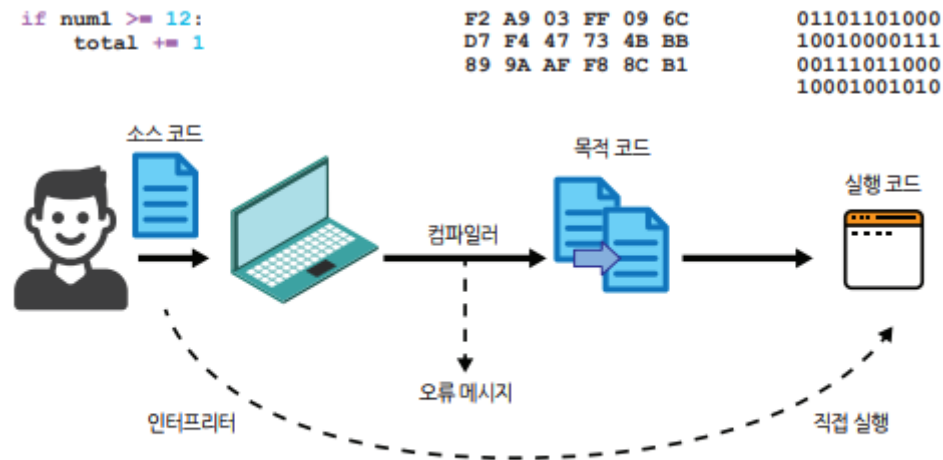
## 컴파일러 언어와 인터프리터 언어

### • 파이썬은 인터프리터(interpreter) 방식으로 실행되는 언어

- 소스 코드를 한 줄씩 해석하여 실행
- 실행 속도는 느리지만, 코드 수정 및 테스트가 용이하다는 장점
- 빠른 개발과 수정이 가능하여 생산성을 향상시키는 데 유리

### • 컴파일러(compiler) 방식

- 소스 코드를 기계어로 변환하여 실행 파일을 생성
- 실행 속도가 빠르지만, 코드 수정 시 다시 컴파일해야 하는 번거로움이 있음
- 대표적 언어 C언어



▲ 그림 5 인터프리터와 컴파일러의 차이



## Section 2. 표준 파이썬과 비주얼 스튜디오 코드 설치



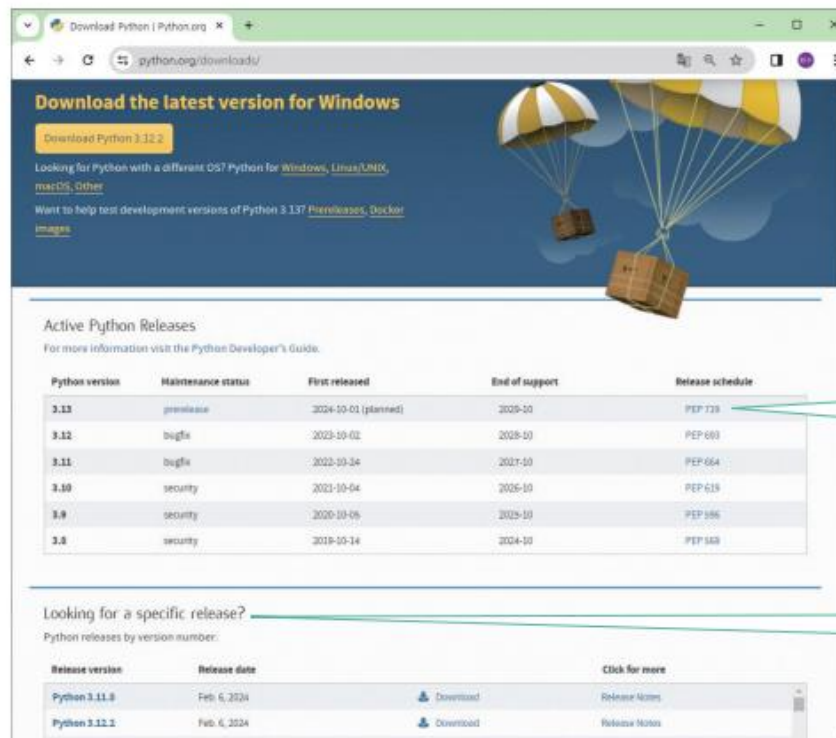


## 표준 파이썬 설치

파이썬 공식 웹사이트([www.python.org](http://www.python.org))

- 윈도우용 최신 버전의 파이썬 설치 파일을 다운로드
  - 32비트 또는 64비트 운영체제에 맞는 설치 파일을 선택
  - 다운로드 페이지에서 "Download Python 3.O.O"을 클릭
    - 설치 파일(예: `python-3.O.O-amd64.exe`)을 다운로드

■ 설치 파일: `python-3.O.O-amd64.exe`



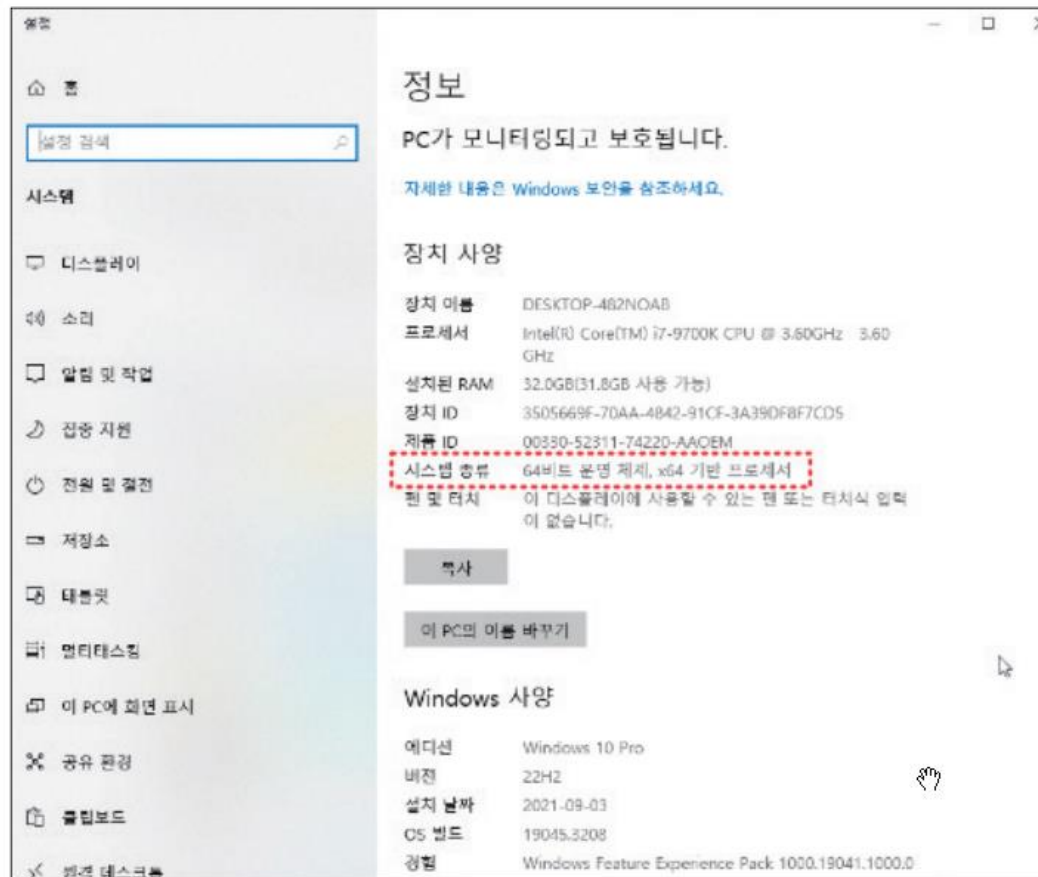
향후 출시 예정인 버전도 다운로드할 수 있다.



과거에 출시된 특정 버전도 다운로드할 수 있다.

▲ 그림 6 표준 파이썬 다운로드 페이지

## 윈도 시스템 종류

- 정보 > 장치 사양 > 시스템 종류
  - 현재 윈도가 32비트인지 64비트인지 확인



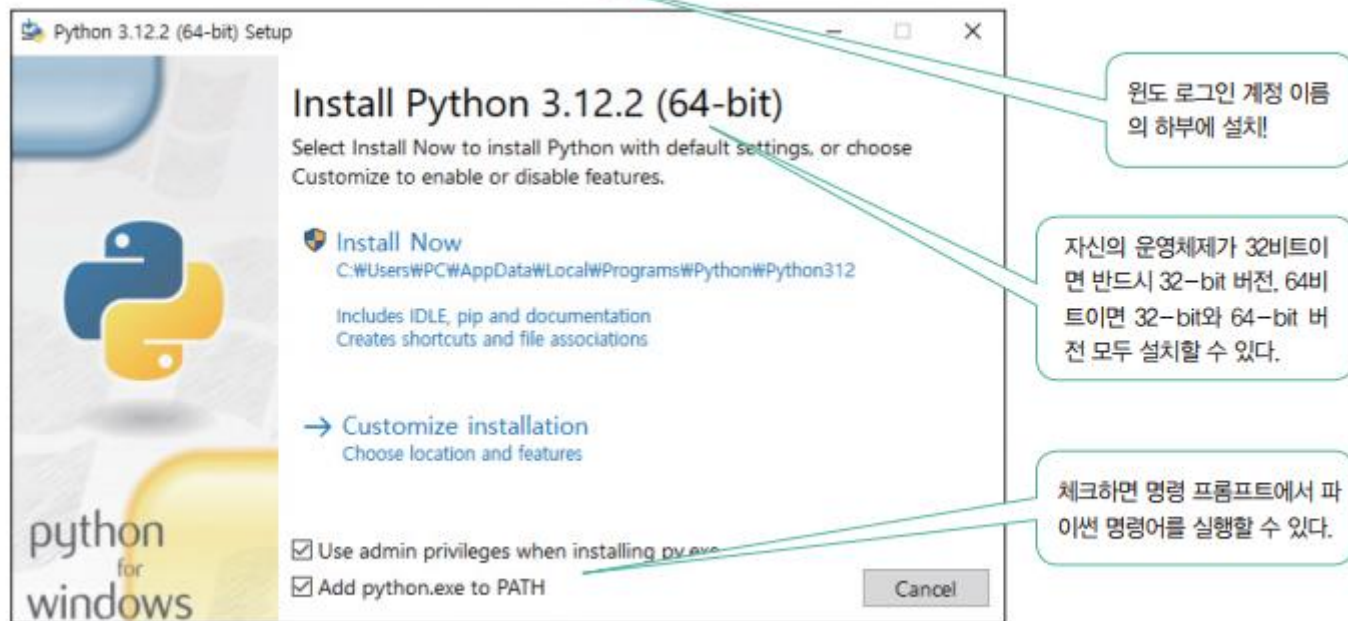
▲ 그림 14 윈도 시스템 종류 확인 화면과 단축 키  + 

## 설치 과정

- 다운로드 받은 설치 파일을 실행하면 설치 마법사가 표시

- "Install Now"를 선택하여 기본 설정으로 설치
- "Customize installation"을 선택하면 설치 위치, 추가 기능 등을 개별적으로 설정

■ 기본 파이썬 설치 폴더: C:\Users\kanghs\AppData\Local\Programs\Python\Python37-32

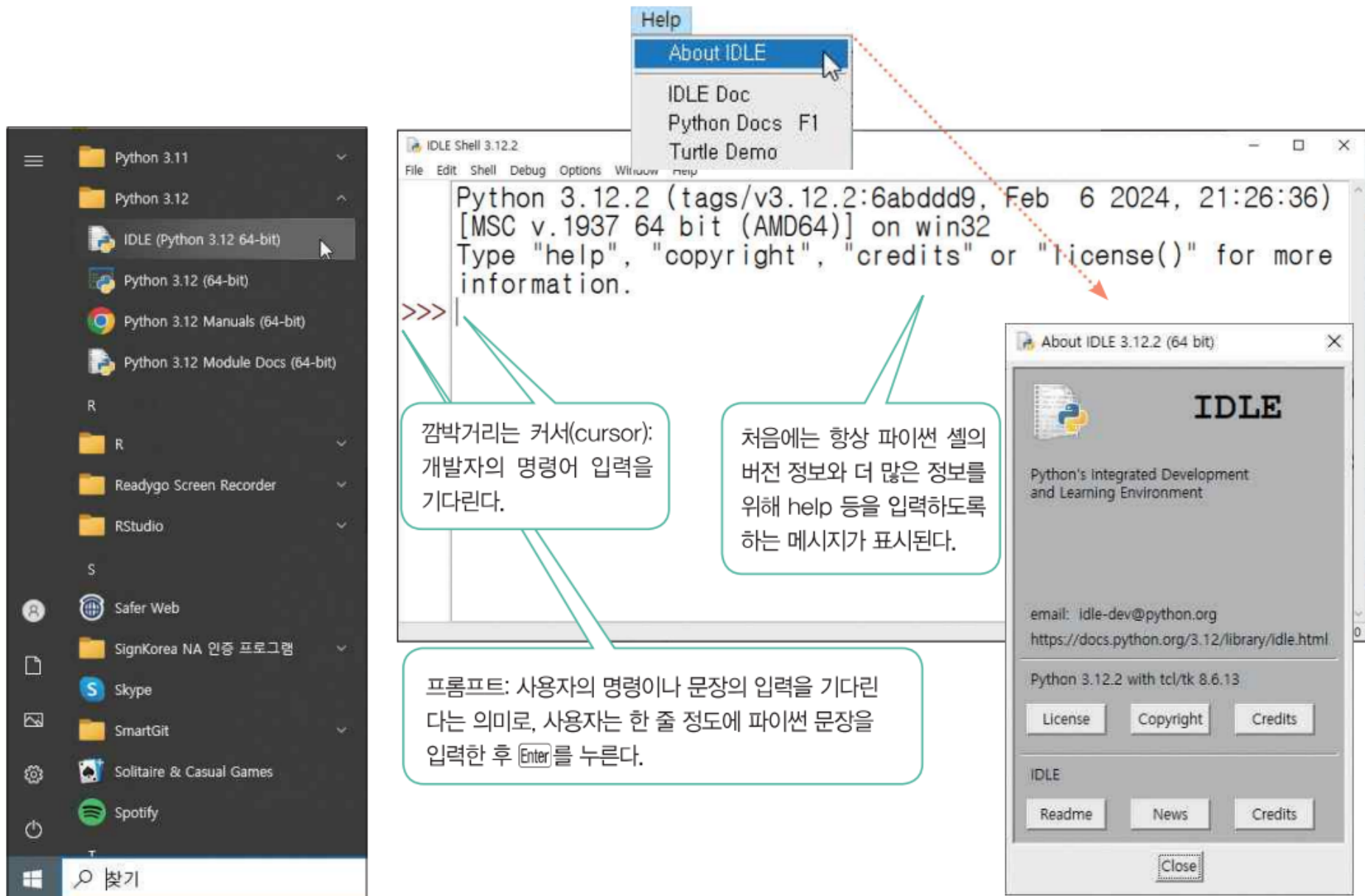


▲ 그림 7 파이썬 설치 시작 화면

## 파이썬 쉘 실행

- 파이썬을 설치하면 IDLE이라는 기본 개발 환경이 함께 설치
  - IDLE은 "Integrated Development Learning Environment"의 약자
    - 파이썬 학습 및 개발에 필요한 기능을 제공
  - IDLE을 실행(최신 버전인 3.13인 경우)
    - [시작] 메뉴에서 "Python 3.13 > IDLE (Python 3.13 64-bit)"를 선택

# Python IDLE 셸 실행과 화면



## REPL (Read-Eval-Print Loop)

- 파이썬 셸은 REPL (Read-Eval-Print Loop) 방식으로 동작
  - 사용자로부터 입력을 받아(Read), 평가하고(Eval),
  - 결과를 출력하고(Print), 다시 입력을 받는(Loop) 과정을 반복
- REPL 환경
  - 코드를 한 줄씩 입력하고 바로 실행 결과를 확인할 수 있기 때문에
  - 파이썬 문법을 익히거나 간단한 코드를 테스트하는 데 유용
- 대화형 모드와 스크립트 모드
  - 파이썬은 REPL 환경에서 코드를 한 줄씩 실행하는 대화형(interactive) 모드
    - 대화형 모드는 코드를 빠르게 테스트하거나 디버깅하는 데 유용
  - 미리 작성된 코드 파일(.py)을 실행하는 스크립트(script) 모드를 모두 지원
    - 스크립트 모드는 완성된 프로그램을 실행하는 데 사용

## 비주얼 스튜디오 코드(Visual Studio Code) 개요와 설치

### • 비주얼 스튜디오 코드(Visual Studio Code)

- 마이크로소프트에서 개발한 가볍지만 강력한 오픈소스 코드 편집기
- Electron 프레임워크를 기반으로 만들어져 웹 기술을 활용
- 다양한 프로그래밍 언어를 지원하며, 확장 기능을 통해 더 많은 기능을 추가

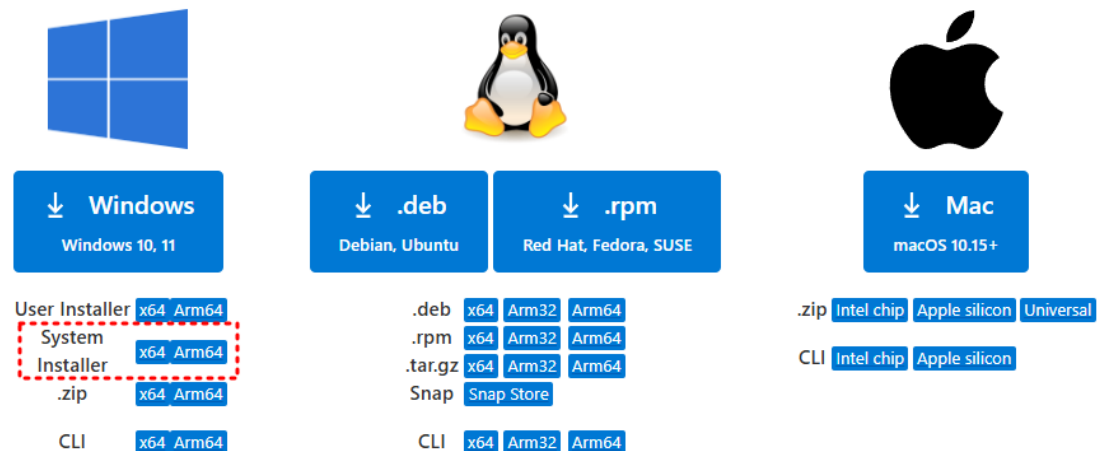
 Get [GitHub Copilot Free](#) in VS Code!

### • 비주얼 스튜디오 코드 설치

- <https://code.visualstudio.com/download>

## Download Visual Studio Code

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



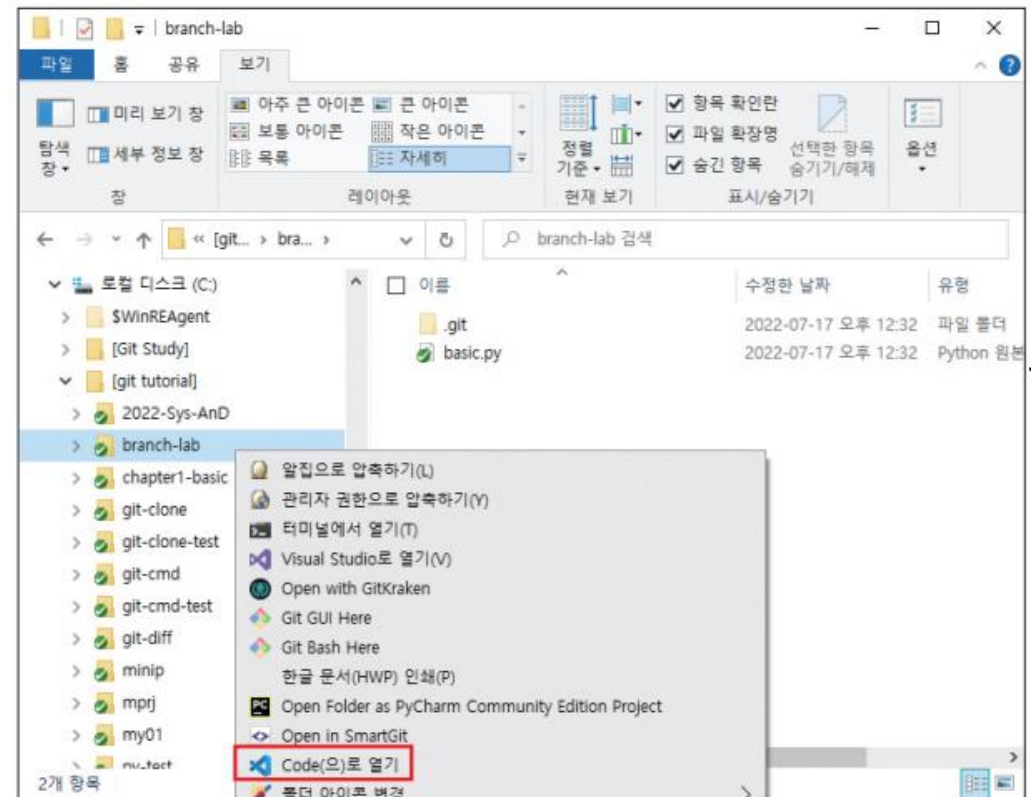
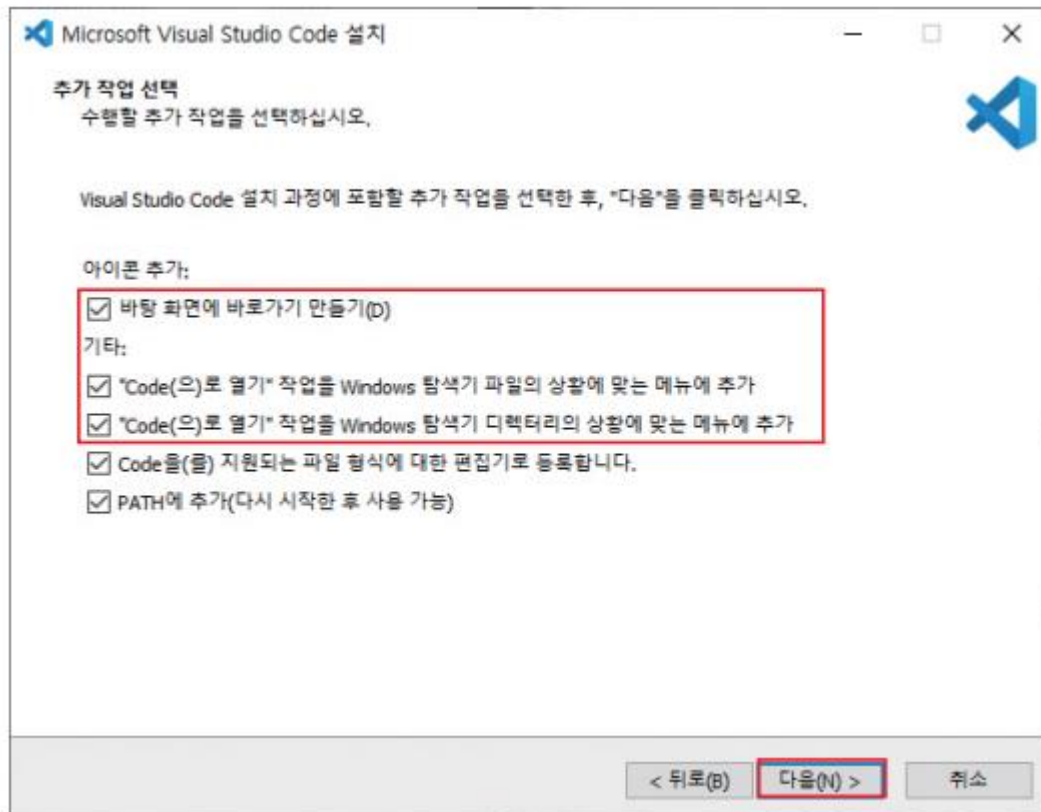
The image shows the download page for Visual Studio Code, organized into three main sections: Windows, Linux, and Mac. Each section has a header icon (Windows logo, Tux penguin, and Apple logo respectively) and a primary download button. Below these are lists of alternative installation methods and their supported architectures.

Platform	Primary Download	Alternative Methods	Supported Architectures
Windows	Windows 10, 11	User Installer, System Installer, .zip, CLI	x64, Arm64
Linux	.deb (Debian, Ubuntu), .rpm (Red Hat, Fedora, SUSE)	.deb, .rpm, .tar.gz, Snap, CLI	x64, Arm32, Arm64
Mac	macOS 10.15+	.zip, CLI	Intel chip, Apple silicon, Universal



## 설치 시 유의사항

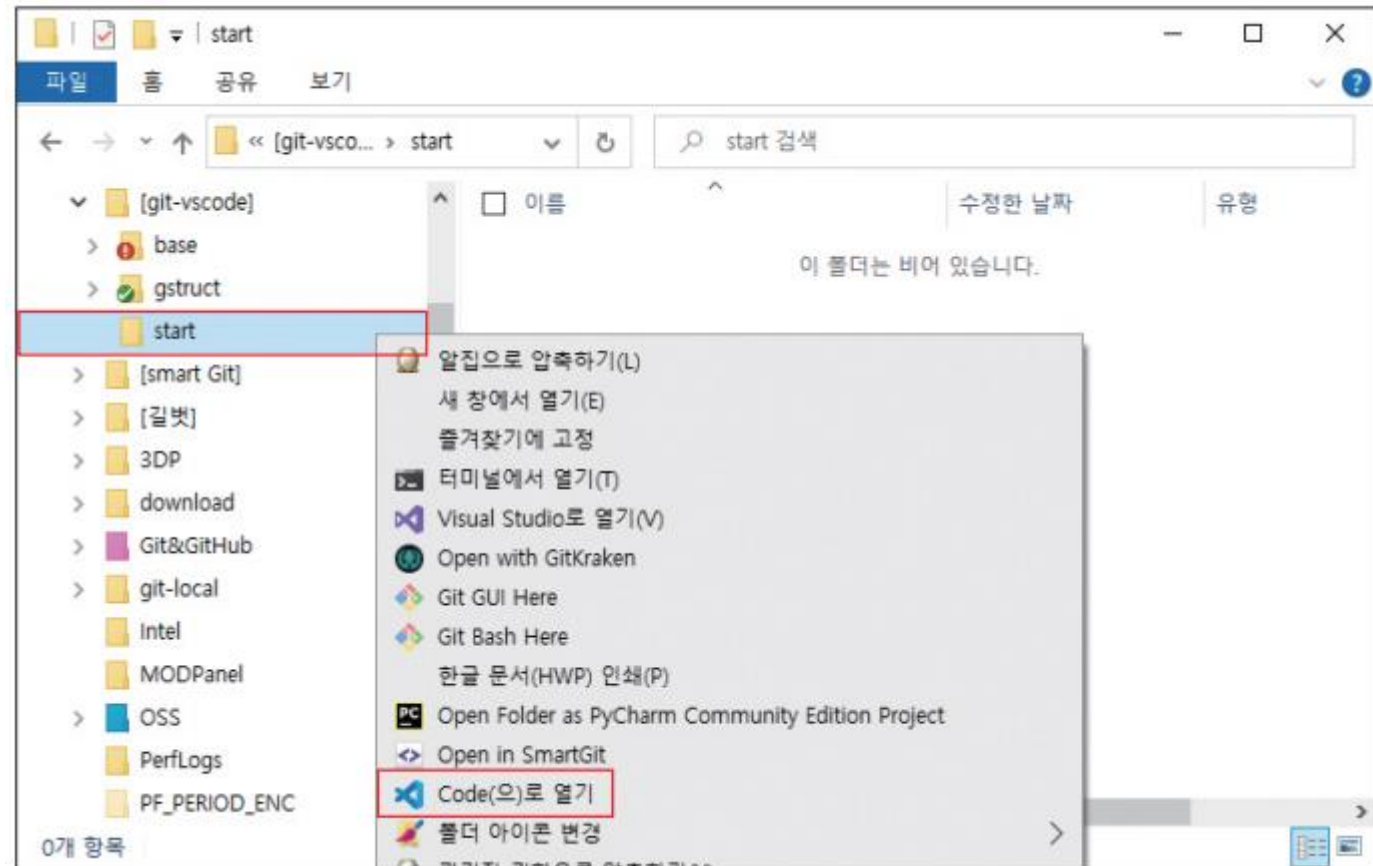
- 탐색기의 파일 또는 디렉터리에서 바로 [Code(으)로 열기] 메뉴를 사용할 수 있도록 선택



▲ 그림 18 파일 또는 폴더의 우측 메뉴에서 [Code(으)로 열기] 선택

## 탐색기의 원하는 폴더에서 팝업 메뉴인 [Code(으)로 열기]로 직접 실행

- 탐색기의 원하는 폴더에서 오른쪽 마우스 버튼을 누르면
  - 나타나는 팝업 메뉴에서 [Code(으)로 열기]를 선택
- 우리 수업에서는
  - 적당한 폴더를 선정



## 비주얼 스튜디오 코드 화면 인터페이스 구성

VS Code의 화면 인터페이스는 크게 다음과 같은 구성 요소

- 메뉴

- 파일, 편집, 선택, 보기, 이동, 실행, 터미널 등 다양한 메뉴를 제공합니다.

- 명령 팔레트

- 단축키 또는 F1 키를 눌러 실행할 수 있으며, 다양한 명령을 검색하고 실행할 수 있습니다.

- 활동 표시줄

- 탐색, 검색, 소스 제어, 실행, 확장 등 주요 기능에 빠르게 접근할 수 있습니다.

- 사이드 바

- 파일 탐색기, 검색 결과, 디버깅 정보 등 다양한 정보를 표시합니다.

- 편집기

- 코드를 작성하고 편집하는 공간입니다.

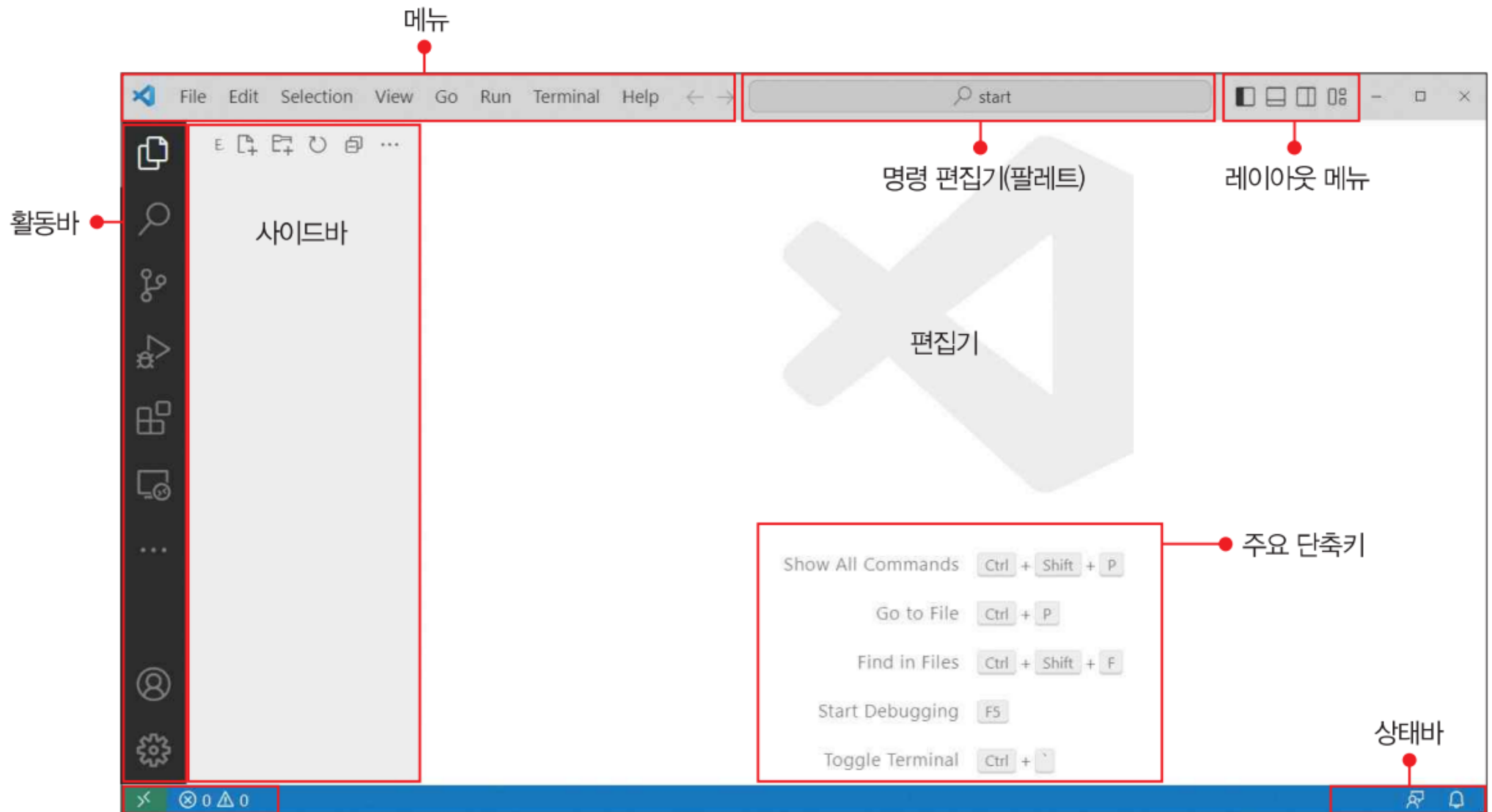
- 상태 표시줄

- 파일 정보, 커서 위치, 줄 번호 등 현재 상태 정보를 표시합니다.

- 패널

- 터미널, 출력, 문제 등 다양한 정보를 표시합니다.

## Vs code의 구성 요소



## Section 3. 첫 파이썬 프로그래밍

-



## 첫 대화형 코딩: 정수와 실수의 간단한 연산 수행

IDLE: 파이썬 자체 내장된 기본 개발 환경으로, 간단한 코드 작성 및 실행에 유용

- **REPL: Read-Eval-Print Loop**

- 사용자 입력을 받아 실행하고 결과를 출력하는 인터랙티브 환경
- IDLE 에서 이를 통해 즉각적인 코드 테스트 및 학습이 가능

- **기본 연산**

- 사칙 연산, 지수 연산 등 수학적 계산을 지원

- **문자열**

- "안녕" 과 같이 큰따옴표로 감싼 텍스트

- **출력**

- print() 함수를 이용: print("Hello World!")
  - "Hello World!" 와 같은 문자열을 화면에 출력

- **주석**

- # 기호를 사용하여 코드에 설명을 추가하며, 실행에 영향을 주지 않음

- **파이썬 셸의 명령어는 첫 칸부터 입력**

- 오류
  - **SyntaxError** 와 같은 오류 발생 시 원인을 파악하고 수정해야 함

## 오류 발생

처음부터 문제 없이 파란색의 결과만 표시되지 않을 수 있다. 아니 그게 당연하다.

- 여러 가지 이유로 잘못 입력해 파이썬 셸이 인식할 수 없는 명령어가 전달되면
  - 다음과 같이 한 줄에서 네 줄 정도 빨간색의 결과 메시지가 표시
  - 파이썬 셸이 개발자에게 오류를 알리는 메시지
    - 이러한 오류 메시지는 개발자에게 오류를 알려 수정할 수 있도록 해 주는 하나의 방법

```

>>> prin('Hello World!')
Traceback (most recent call last):
  File "<pyshell#1>", line 1, in <module>
    prin('Hello World!')
NameError: name 'prin' is not defined. Did you mean: 'print'?
>>> print('Hello World!')
...
SyntaxError: unterminated string literal (detected at line 1)
>>>
  
```

잘못 입력된 함수 이름으로 print를 권유

출력 글자 Hi, python에서 앞 홑따옴표는 있지만 맨 뒤에 홑따옴표가 없어 출력 글자의 끝이 명시되지 않아 발생하는 오류라는 것을 표시하는데, 이를 간단히 unterminated string literal으로 표기

▲ 그림 27 그림 오류 등 오류 메시지의 두 가지 예



## 오류 메시지 이해

### • 오류 메시지

- 소스 파일 이름, 추정되는 오류가 있는 줄 번호, 추정 오류 문장, 오류 이름: 에러 메시지 등이 표시
- 'SyntaxError: unterminated string literal'
  - 'SyntaxError'가 오류 이름이며, 'unterminated string literal'은 오류 메시지
  - 오류 중 'SyntaxError'는 앞으로 자주 표시될 오류로, 구문 오류라고 부르며, 주로 한 줄로 표시
  - 특히 오류 메시지의 마지막 줄, 콜론(:) 이후 오류 원인이 표시되는 오류 메시지를 읽어 보면 무엇이 문제인지 어느 정도 이해할 수 있음
  - 아직은 몰라도 상관없으니 무조건 읽어 보도록 하자.

```

IDLE Shell 3.12.2
File Edit Shell Debug Options Window Help
>>> prin('Hello World!')
Traceback (most recent call last):
  File "<pyshell#1>", line 1, in <module>
    prin('Hello World!')
NameError: name 'prin' is not defined. Did you mean: 'print'?
>>> print(Hello World!)
...
SyntaxError: unterminated string literal (detected at line 1)
>>>
  
```

잘못 입력된 함수 이름으로 print를 권유

출력 글자 Hi, python에서 앞 홑따옴표는 있지만 맨 뒤에 홑따옴표가 없어 출력 글자의 끝이 명시되지 않아 발생하는 오류라는 것을 표시하는데, 이를 간단히 unterminated string literal으로 표기

▲ 그림 27 그림 오류 등 오류 메시지의 두 가지 예

## 디버깅(debugging)

프로그램의 잘못을 수정하는 과정

- 디버깅 과정에서 오류 메시지는 잘못을 찾는 중요한 단서를 제공
  - 오류 또는 에러(error)
    - 프로그램의 여러 잘못된 문제
    - 많은 문제가 철자나 기호 문자를 잘못 입력해서 발생하는 문제
    - 빨간색의 오류 메시지는 항상 읽어 보는 습관을 기르도록 권장
    - 오류 메시지를 통해 오류를 쉽게 수정 가능

```

IDLE Shell 3.12.2
File Edit Shell Debug Options Window Help
>>> prin('Hello World!')
Traceback (most recent call last):
  File "<pyshe11#1>", line 1, in <module>
    prin('Hello World!')
NameError: name 'prin' is not defined. Did you mean: 'print'?
>>> print(Hello World!)
SyntaxError: unterminated string literal (detected at line 1)
>>>
  
```

잘못 입력된 함수 이름으로 print를 권유

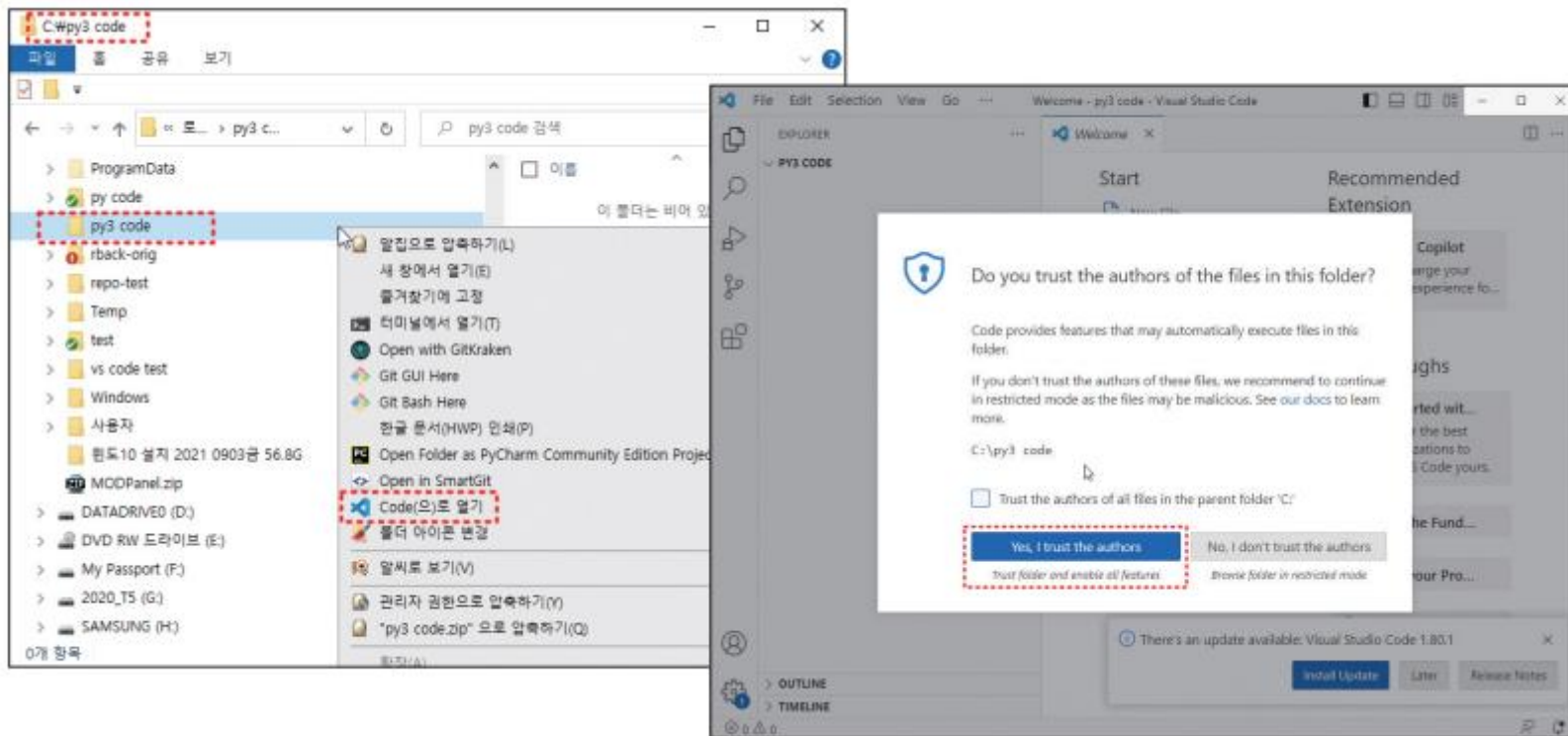
출력 글자 Hi, python에서 앞 홑따옴표는 있지만 맨 뒤에 홑따옴표가 없어 출력 글자의 끝이 명시되지 않아 발생하는 오류라는 것을 표시하는데, 이를 간단히 unterminated string literal으로 표기

▲ 그림 27 그림 오류 등 오류 메시지의 두 가지 예

## Vs code 실행

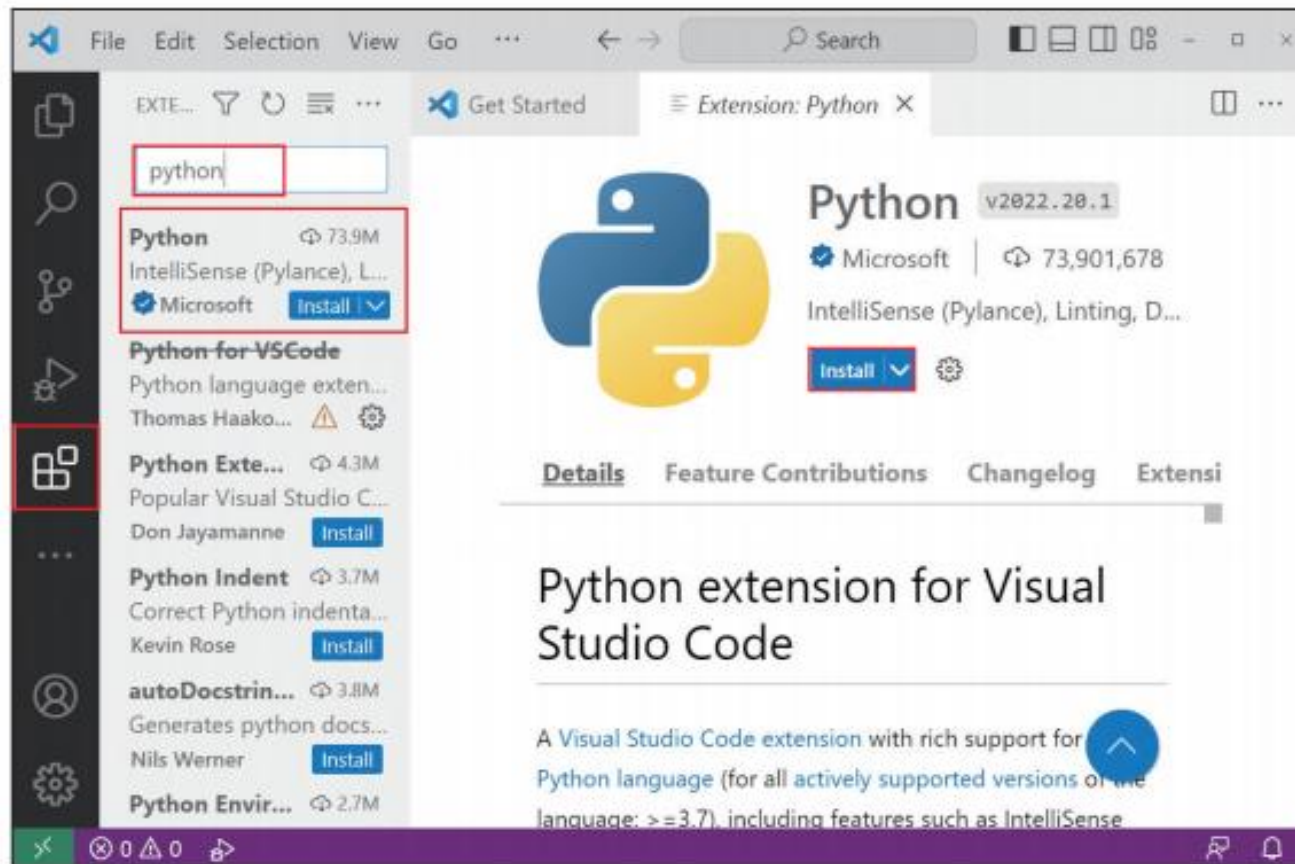
### • 탐색기의 원하는 폴더

- 오른쪽 마우스 버튼을 누르면 나타나는 팝업 메뉴에서 [Code(으)로 열기]를 직접 실행



## 파이썬 확장 설치

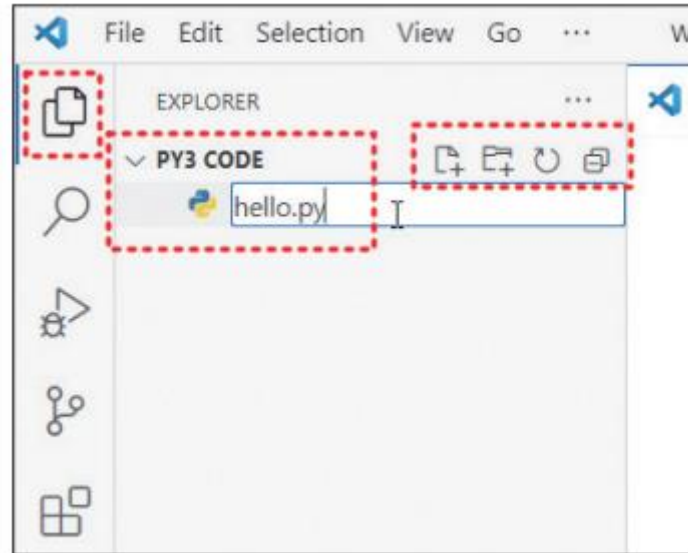
- VS Code 에서 파이썬 코드 작성 및 실행을 위한 필수 핵심 확장 기능
  - IntelliSense, Linting, Debugging 기능을 제공.



▲ 그림 30 Python 확장 설치

## 코드 작성 및 실행

- VS Code 에서 코드를 작성하고 `print()` 함수를 이용해 원하는 출력 결과 확인 가능



▲ 그림 32 활동바 탐색기의 사이드바 메뉴와 파일 생성

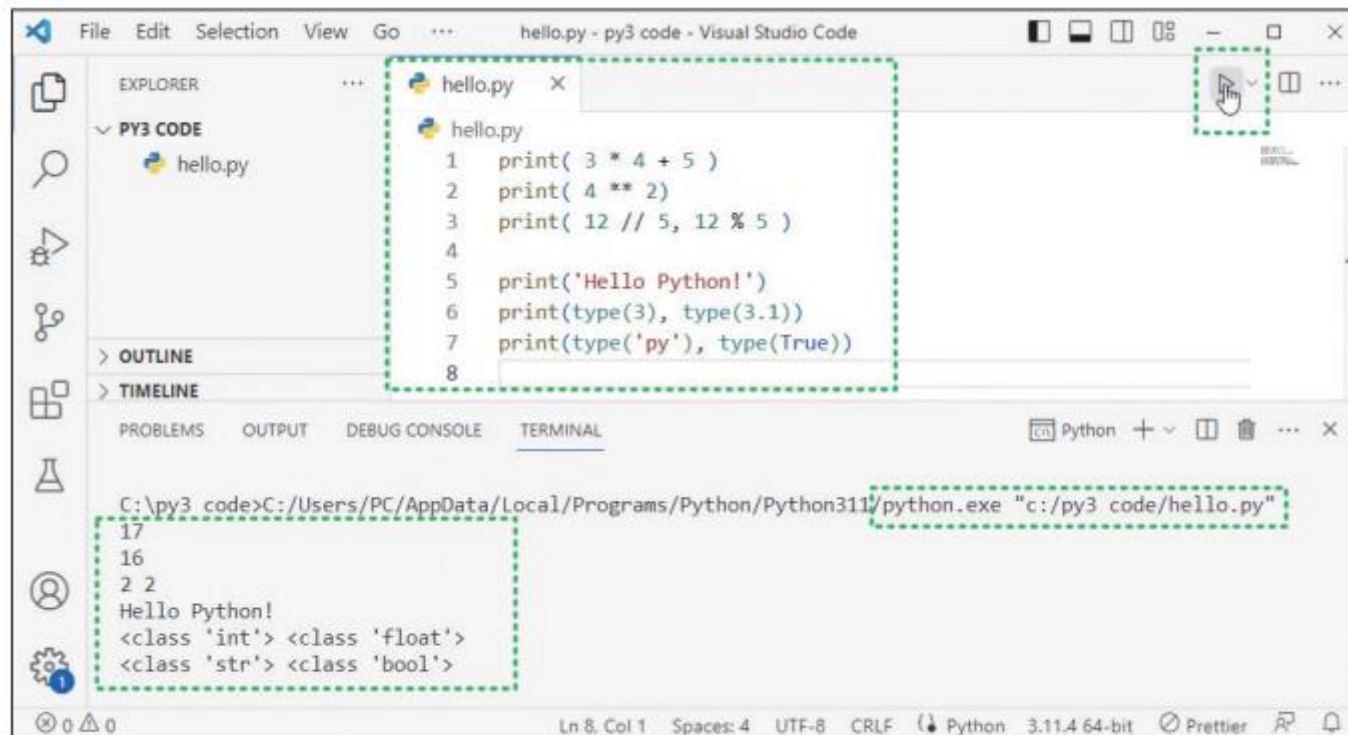
## 파이썬 코드 hello.py 실행

### • 실행 방법

- 저장된 파이썬 소스 파일의 실행은 쉽게 우측 상단 삼각형이나
- 하부 메뉴에서 Run Python File을 선택

### • 결과 표시

- 오류 없이 실행되면 편집기의 하단 패널에 터미널( Terminal )이 실행 되고 결과가 표시



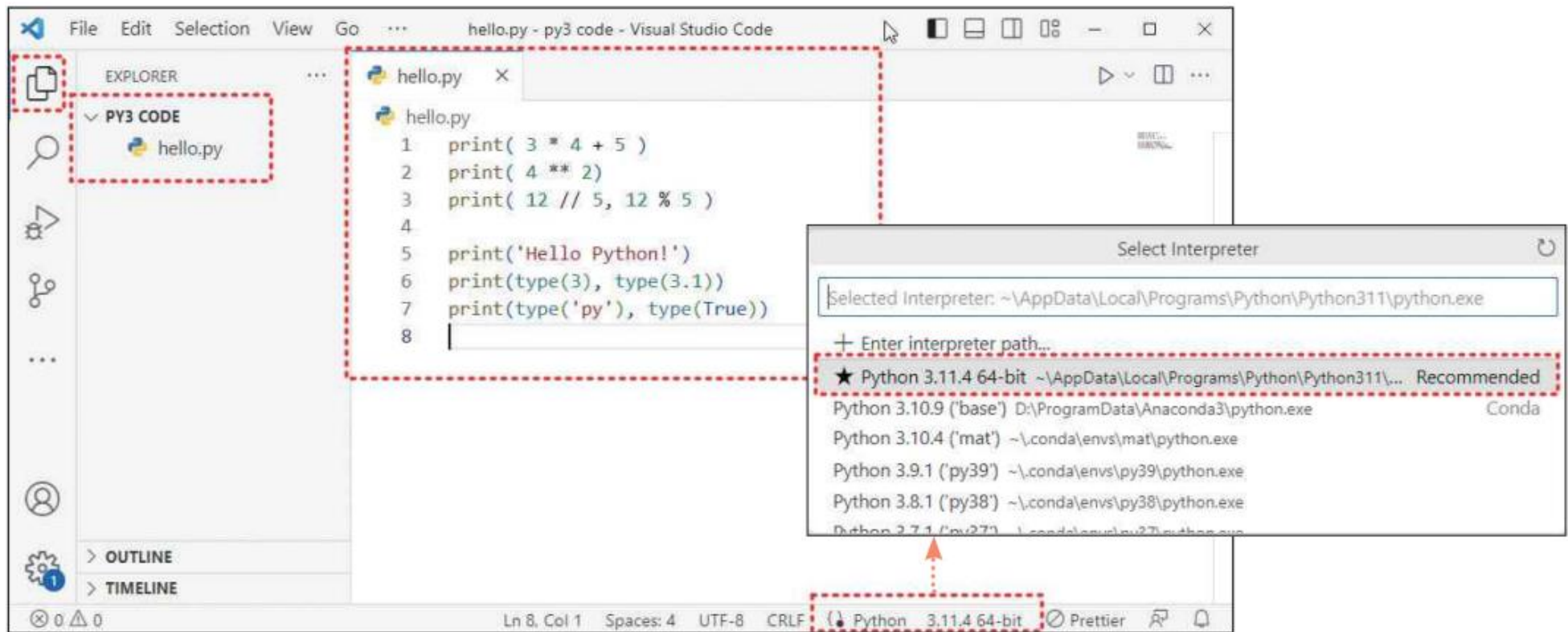
▲ 그림 34 파이썬 소스 파일 실행: 우측 상단 삼각형이나 하부 메뉴에서 [Run Python File]을 선택



## 설정된 인터프리터 확인

### • 상태바의 우측 하단

- 현재의 프로그래밍 언어 Python과 설정된 인터프리터인 3.00.0 64-bit를 확인





## 주요 설정과 단축키

- 주요 설정

- 활동바 설정의 하부 메뉴인 [Settings]에서 수행
- 단축키 ctrl + , 로도 설정 화면을 표시
  - 'Editor: Font Size'
    - 편집기 폰트의 크기를 조정
- 화면에 표시되는 글자와 메뉴 글자의 크기를 조정( 또는 줌인 줌아웃 기능 )
  - `Ctrl` + `+`: 확대, `Ctrl` + `-`: 축소