

파이썬 프로그래밍 강의 노트 #01

프로그래밍, 파이썬, 주석문

프로그래밍이란? – Webster 사전 검색

programming **noun**

pro·gram·ming ('prō-ˌgra-mɪŋ) -grə-

variants *or less commonly* **programing**

- 1 : the planning, scheduling, or performing of a **program**
- 2 **a** : the process of instructing or learning by means of an instructional program
b : the process of preparing an instructional program for a device (such as a computer) <https://www.merriam-webster.com/dictionary/programming>

□ 프로그래밍

- 프로그램을 계획(planning), 작업의 순서를 정하거나 실행하는 것
- "주어진 문제를 해결하기 위해" 작업의 순서를 계획하고 수행하는 것



MARK
CREATED facebook

프로그래밍이란?

□ 프로그래밍이란?

- 주어진 문제를 해결하기 위한 계획 및 진행 과정
- 문제?
 - 오늘 점심 뭐 먹을까?
 - 집에서 학교에 오는 것?
 - 학교에서 지금 파이썬 수업을 듣는 강의실에 찾아오기
 - 라면 끓여 먹기

□ 프로그래밍의 핵심

- 요구사항 분석
- 문제를 해결하는 방법(알고리즘)을 창자야 함
- 검수

컴퓨터 프로그래밍

□ 컴퓨터 프로그래밍

- 컴퓨터가 주어진 문제를 해결할 수 있도록 사람이 지시해야 하는 일

□ 사람이 하는 일

- 문제 해석
- 요구사항 분석
- 계획
- 처리 과정(명령) 지시 (코딩) 및 디버깅

□ 컴퓨터가 하는 일

- 주어진 명령을 처리

프로그래밍 언어

- 저수준(Low-level), 고수준(High-level) 언어로 구분
 - 사람에게 가까운지(고수준) 컴퓨터에게 가까운지(저수준)을 결정
- 저수준 프로그래밍 언어
 - 사람보다 컴퓨터가 이해하기 쉬움
 - 하드웨어 종속적
 - 이진수로 구성된 기계어에 거의 1:1로 사상시킨 어셈블리 언어
- 고수준 프로그래밍 언어
 - 컴퓨터보다 사람이 이해하기 쉽게 만들어진 언어
 - 하드웨어에 비종속적
 - 파이썬(Python), C, C++, 자바(Java) 등

프로그래밍 언어

- 자연어 (Natural Language)
 - 사람이 이해할 수 있는 언어
- 기계어 (Machine Language)
 - 컴퓨터가 이해할 수 있는 언어
- 프로그래밍 언어 (언어)
 - 컴퓨터가 이해할 수 있는 언어로 번역 가능한 언어
 - 자연어에 비해 훨씬 단순해야 함
 - 문법
 - 단어 개수

프로그래밍언어의 요소

- ❑ 입력: 프로그램에서 처리하려고 하는 데이터
- ❑ 출력: 프로그램에서 처리된 후의 결과를 사용자에게 보여주는 것
- ❑ 순차: 순서에 따라 처리하는 과정. 일을 순서대로 나누어서 수행하는 과정.
- ❑ 조건: 경우에 따라 구분하여 처리하는 과정. 예를 들어 조건이 만족될 때에는 어떤 일을 해야 하고 조건이 만족되지 않을 때에는 다른 일을 해야 함
- ❑ 반복: 같은 일을 여러 번 다시 하는 과정
- ❑ 재사용: 프로그램의 일부를 고치지 않고 다시 활용하는 과정. 프로그래밍 언어의 용어를 빌리면 함수 등을 만들어서 코드를 재사용함

100 *SECONDS OF*



파이썬(Python)

□ 파이썬

- 1991년에 귀도 반 로섬(Guido van Rossum)이라는 사람이 만든 언어
- 최신 프로그래밍 트렌드에 맞게 지속적으로 발전되고 있는 고급 프로그래밍 언어

□ 파이썬의 버전

- 2.x와 3.x버전이 있음. 여기서는 3.x를 사용 (2.x는 지원 끊김)

파이썬(Python)

□ PYPL (PopularitY of Programming Language)

■ <http://pypl.github.io/PYPL.html>











Worldwide, Feb 2023 compared to a year ago:

Rank	Change	Language	Share	Trend
1		Python	27.7 %	-0.7 %
2		Java	16.79 %	-1.3 %
3		JavaScript	9.65 %	+0.6 %
4	↑	C#	6.97 %	-0.5 %
5	↓	C/C++	6.87 %	-0.6 %
6		PHP	5.23 %	-0.8 %
7		R	4.11 %	-0.1 %
8	↑↑	TypeScript	2.83 %	+0.8 %
9		Swift	2.27 %	+0.3 %
10	↓↓	Objective-C	2.25 %	-0.1 %

파이썬(Python)

□ TIOBE Index for Jan. 2022

■ <https://www.tiobe.com/tiobe-index/>

Feb 2023	Feb 2022	Change	Programming Language		Ratings	Change
1	1			Python	15.49%	+0.16%
2	2			C	15.39%	+1.31%
3	4	▲		C++	13.94%	+5.93%
4	3	▼		Java	13.21%	+1.07%
5	5			C#	6.38%	+1.01%
6	6			Visual Basic	4.14%	-1.09%
7	7			JavaScript	2.52%	+0.70%
8	10	▲		SQL	2.12%	+0.58%
9	9			Assembly language	1.38%	-0.21%
10	8	▼		PHP	1.29%	-0.49%

파이썬(Python)

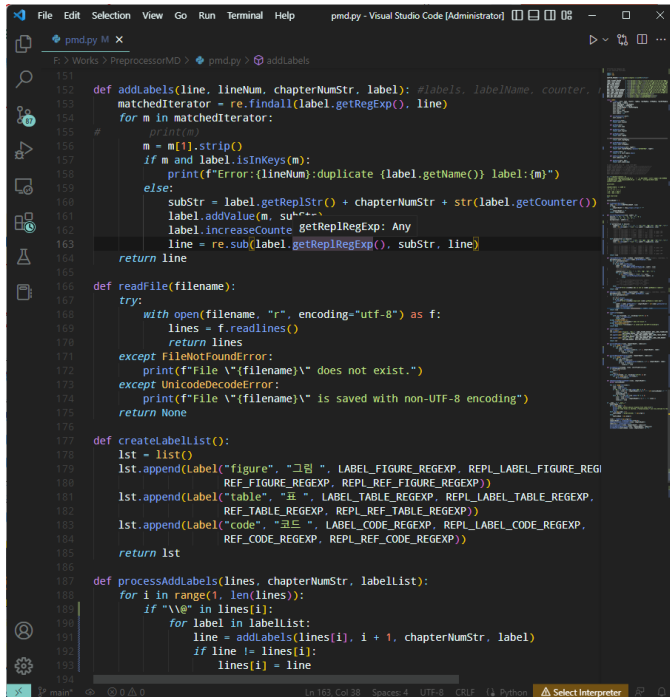
□ 왜 파이썬을 사용하려 하는가?

- 초보자가 배우기 쉬운 언어
 - 인터프리터 언어이고 문법이 쉬움
- 최신 프로그래밍 기법 지원
 - 객체 지향언어
 - 함수형 언어의 일부 기능 지원
- 다양한 분야에서 사용됨
 - 다양한 분야의 라이브러리(library)
 - 웹, 금융, 인공지능, 컴퓨터비전, 생명공학, 물리, 화학, 수학 등 거의 모든 전공에서 사용 가능
- 만능은 아님
 - 빠른 시간에 프로그램을 만들 수 있지만, 실행 속도가 느림

프로그램 작성

□ 소스 코드 작성

- 프로그래밍 언어 사용
- 텍스트 에디터 (메모장, IDLE, Visual Studio Code 등)
- 언어에 따라 확장자를 다르게 붙임(파이썬은 .py 사용)



```
151
152 def addLabels(line, lineNumber, chapterNumStr, label): #labels, labelName, counter,
153     matchedIterator = re.findall(label.getRegExp(), line)
154     for m in matchedIterator:
155         #
156         m = m[1].strip()
157         if m and label.isInKeys(m):
158             print(f"Error:(lineNum):duplicate {label.getName()} label:(m)")
159         else:
160             subStr = label.getReplStr() + chapterNumStr + str(label.getCounter())
161             label.addValue(m, subStr)
162             label.increaseCounter()
163             line = re.sub(label.getReplRegExp(), subStr, line)
164     return line
165
166 def readFile(filename):
167     try:
168         with open(filename, "r", encoding="utf-8") as f:
169             lines = f.readlines()
170             return lines
171     except FileNotFoundError:
172         print(f"File \"{filename}\" does not exist.")
173     except UnicodeDecodeError:
174         print(f"File \"{filename}\" is saved with non-UTF-8 encoding")
175     return None
176
177 def createLabelList():
178     lst = list()
179     lst.append(Label("figure", "그림 ", LABEL_FIGURE_REGEXP, REPL_LABEL_FIGURE_REGEXP,
180                     REF_FIGURE_REGEXP, REPL_REF_FIGURE_REGEXP))
181     lst.append(Label("table", "표 ", LABEL_TABLE_REGEXP, REPL_LABEL_TABLE_REGEXP,
182                     REF_TABLE_REGEXP, REPL_REF_TABLE_REGEXP))
183     lst.append(Label("code", "코드 ", LABEL_CODE_REGEXP, REPL_LABEL_CODE_REGEXP,
184                     REF_CODE_REGEXP, REPL_REF_CODE_REGEXP))
185     return lst
186
187 def processAddLabels(lines, chapterNumStr, labelList):
188     for i in range(1, len(lines)):
189         if "\n\n" in lines[i]:
190             for label in labelList:
191                 line = addLabels(lines[i], i + 1, chapterNumStr, label)
192                 if line != lines[i]:
193                     lines[i] = line
194     return lines
```

□ 소스 파일

- 파일에 저장된 소스 코드
- 텍스트 파일 저장 방법(인코딩)은 언어마다 다름
 - 파이썬 소스 코드는 UTF-8로 저장

프로그램은 어떻게 실행하는가?

□ 실행

■ 인터프리터

- 한 줄씩 통역해서 전달
- 파이썬

■ 컴파일러(Compiler) 또는 번역기(Translator)

- 전체를 번역해서 전달
- C, Java 등

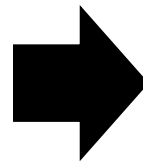
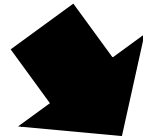
프로그램은 어떻게 실행되는가?

소스 코드/파일
(Source Code)

```
import sys
import os

f = open(sys.argv[2])
print(sys.argv[2])
count = 0
found = False
for line in f:
    line = line.rstrip()
    count = count + 1
    if sys.argv[1] in line:
        idx = line.index(sys.argv[1])
        print(str(count) + ":" + str(idx + 1) +
              found = True
if not found:
    print("Not found here")
```

실행
(Execution)



```
100011111011011101
1111101101101101000
011000011011000111
011011000111110110
110110001111101101
110001111101101110
001101100011111011
100001101100011111
```

인터프리터 또는 컴파일러
(Interpreter or Compiler)

이진 코드
(Binary Code)



**When you land in the world of computers with
their strange convoluted machine language**

이상하고 복잡한 기계 언어를 사용하는 컴퓨터의 세계에 착륙하는 것은 마치

<https://youtu.be/Dx2tSsd3aFc>

프로그램은 어떻게 실행하는가?

- 파이썬 프로그램을 실행하는 방법 두 가지
 - 파이썬 셸(shell) 사용
 - 한 문장 단위로 파이썬 인터프리터가 처리
 - 파이썬 코드 파일을 실행
 - 텍스트 에디터를 이용해서 파이썬 코드를 작성해서 파일로 저장
 - 파이썬 인터프리터로 하여금 코드 파일을 읽어서 한 문장씩 실행시키도록 함

파이썬 사이트 접속 (http://www.python.org)

Welcome to Python.org

Python Software Foundation [US] | https://www.python.org

Free Learning - Free | Google 스프레드시트 | How to make HTTP P

Python PSF Docs PyPI Jobs Community

python™

Donate Search GO Socialize

About Downloads Documentation Community Success Stories News Events

```
# Python 3: Simple arithmetic
>>> 1 / 2
0.5
>>> 2 ** 3
8
>>> 17 / 3 # classic division returns a float
5.666666666666667
>>> 17 // 3 # floor division
5
```

Intuitive Interpretation

Calculations are simple with Python, and expression syntax is straightforward: the operators `+`, `-`, `*` and `/` work as expected; parentheses `()` can be used for grouping. [More about simple math functions in Python 3.](#)

1 2 3 4 5

Python is a programming language that lets you work quickly and integrate systems more effectively. >>> [Learn More](#)

Get Started
Whether you're new to programming or an experienced developer, it's easy to learn and use Python.
Start with our [Beginner's Guide](#)

Download
Python source code and installers are available for download for all versions!
Latest: Python 3.7.2

Docs
Documentation for Python's standard library, along with tutorials and guides, are available online.
[docs.python.org](#)

Jobs
Looking for work or have a Python related position that you're trying to hire for? Our **relaunched community-run job board** is the place to go.


파이썬 다운로드

Welcome to Python.org

Python Software Foundation [US] | https://www.python.org

Free Learning - Free F | Google 스프레드시트 | How to make HTTP P

PythonPSFDocsPyPIJobsCommunity

 [Donate](#) [GO](#) [Socialize](#)

AboutDownloadsDocumentationCommunitySuccess StoriesNewsEvents

Python 3: Sim
>>> print("Hell
Hello, I'm Pyth

Input, assign
>>> name = input
>>> print('Hi,
What is your na
Python
Hi, Python.

All releases
Source code
Windows
Mac OS X
Other Platforms
License
Alternative Implementations


Download for Windows

Python 3.7.2

Note that Python 3.5+ cannot be used on Windows XP or earlier.

Not the OS you are looking for? Python can be used on many operating systems and environments.
[View the full list of downloads.](#)


Python is a programming language that lets you work quickly and integrate systems more effectively. >>> [Learn More](#)



Get Started

Whether you're new to programming or an experienced developer, it's easy to learn and use Python.


[Start with our Beginner's Guide](#)



Download

Python source code and installers are available for download for all versions!


Latest: Python 3.7.2



Docs

Documentation for Python's standard library, along with tutorials and guides, are available online.

[docs.python.org](#)

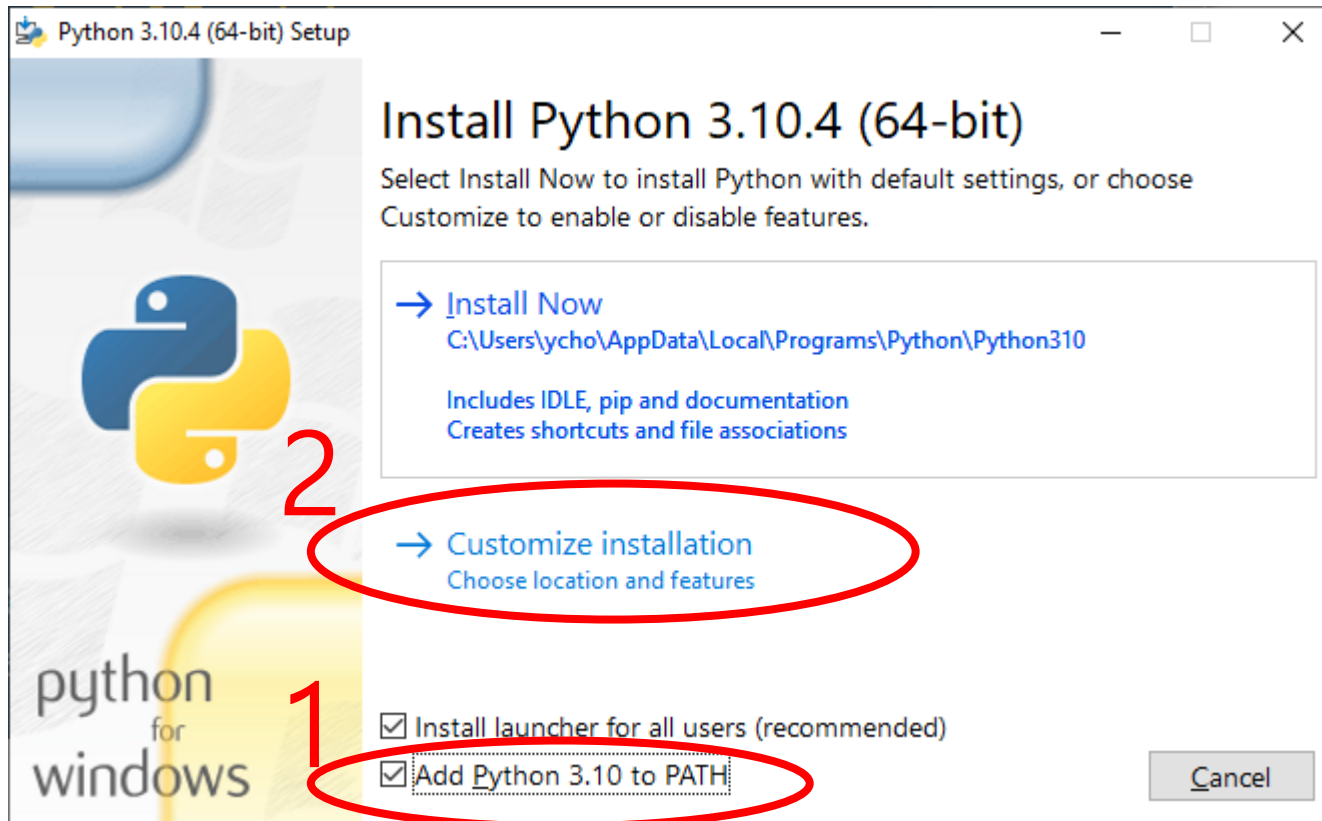


Jobs

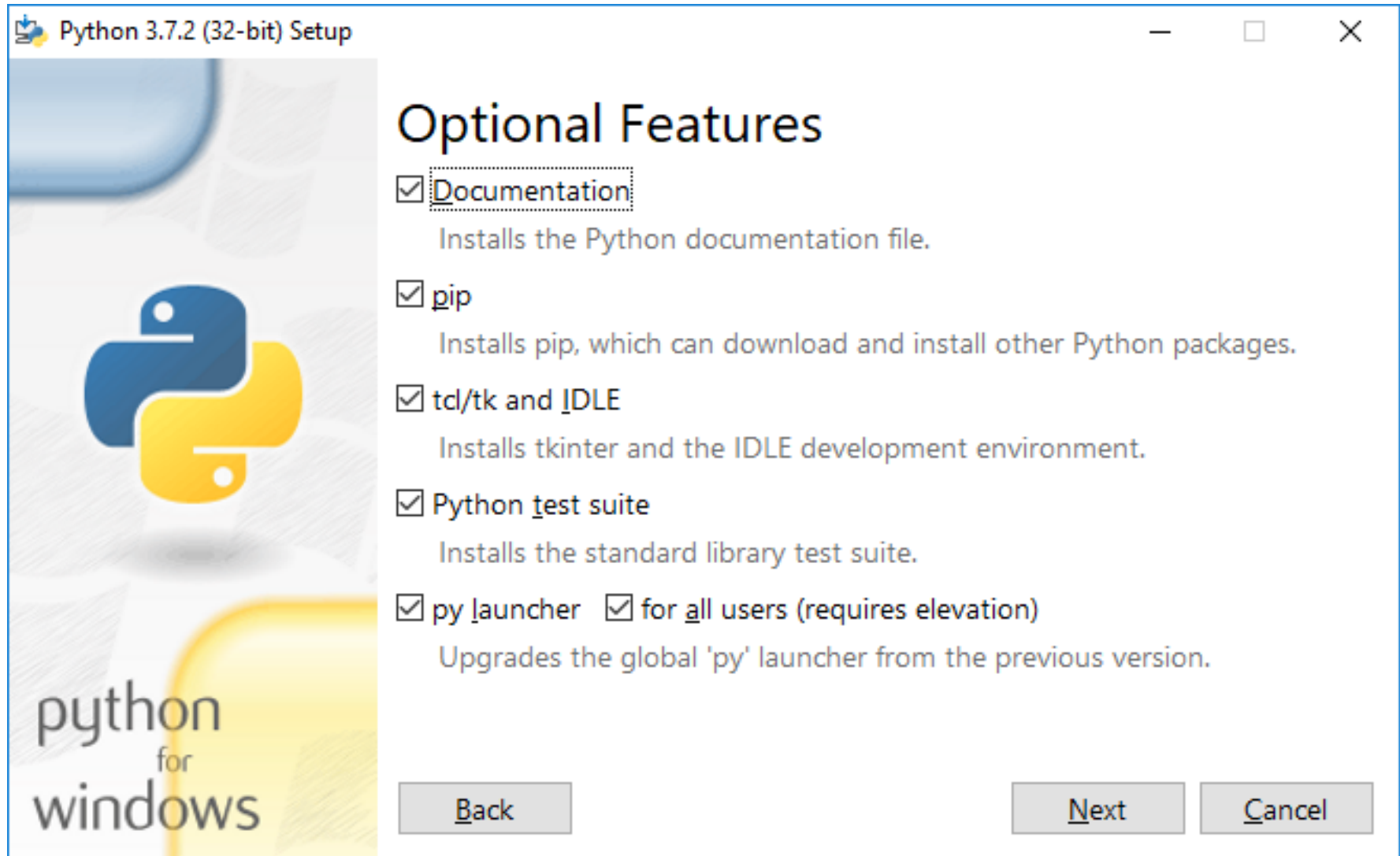
Looking for work or have a Python related position that you're trying to hire for? Our **relaunched community-run job board** is the place to go.

<https://www.python.org/ftp/python/3.7.2/python-3.7.2.exe>

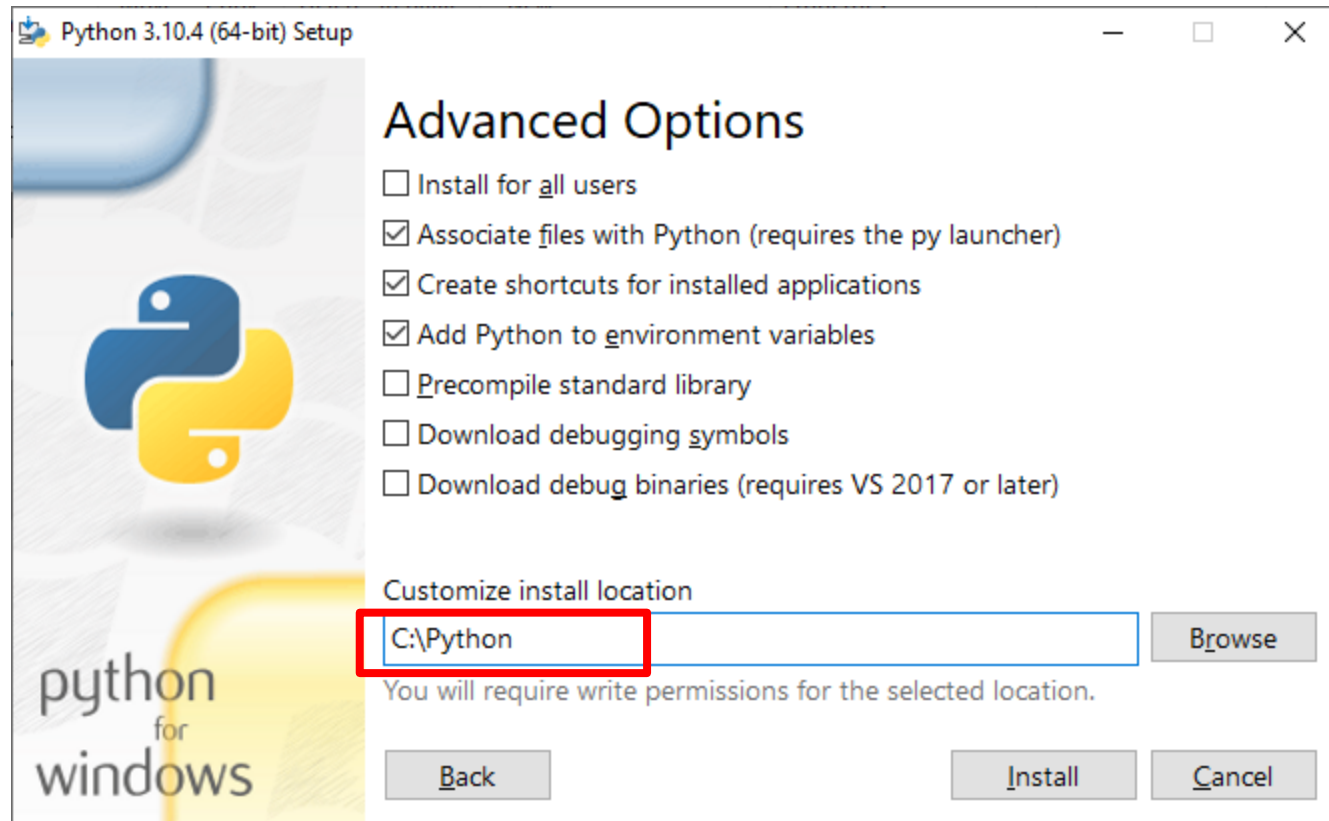
파이썬 설치



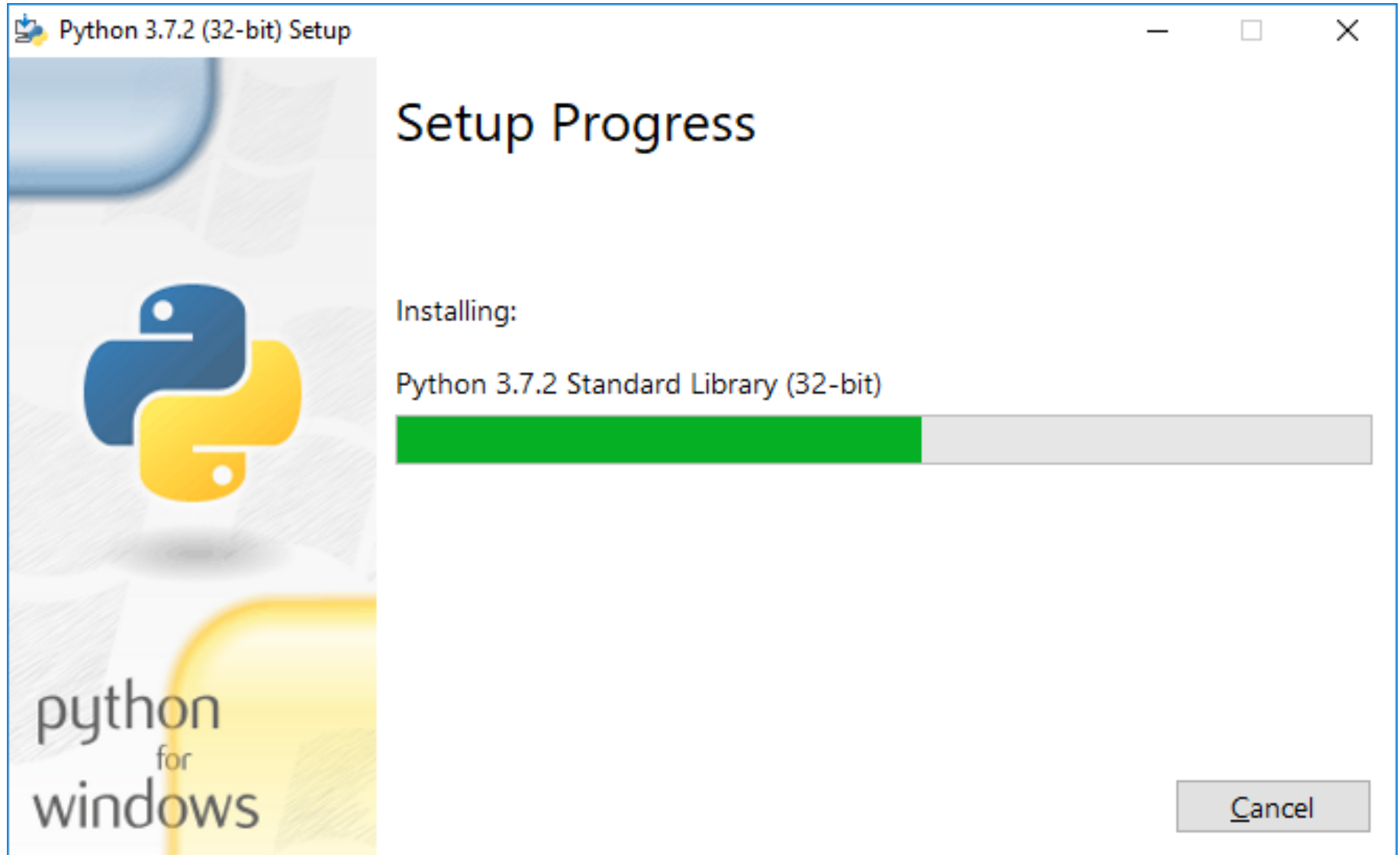
파이썬 설치



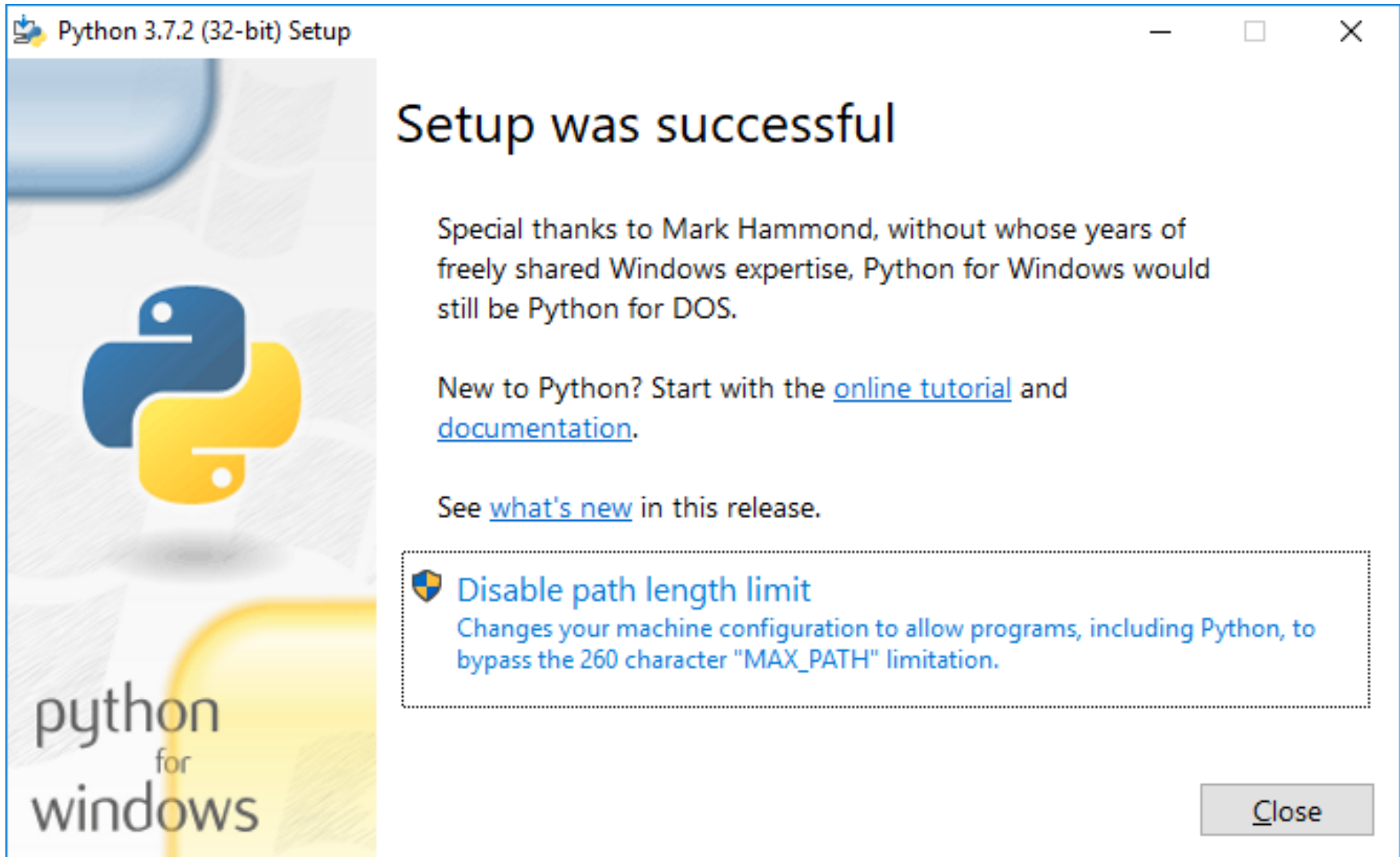
파이썬 설치



파이썬 설치

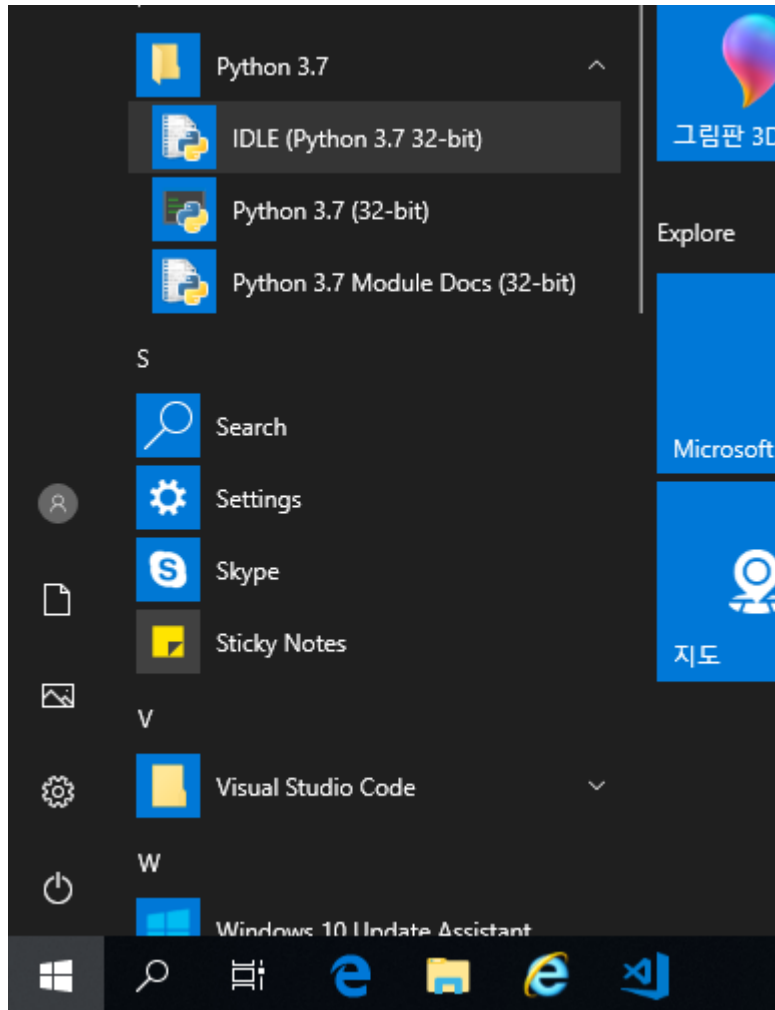


파이썬 설치



IDLE 사용

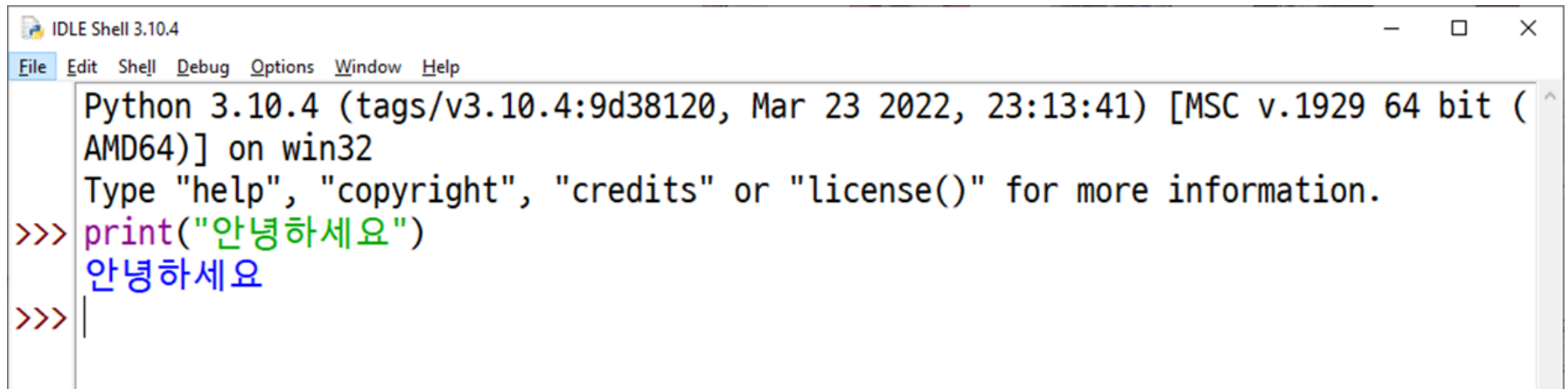
▣ 시작 메뉴에서 IDLE 실행



IDLE 사용

- ▣ IDLE 실행 후 코드 입력
- ▣ 셸 모드(shell mode)에서 실행

```
print("안녕하세요")
```

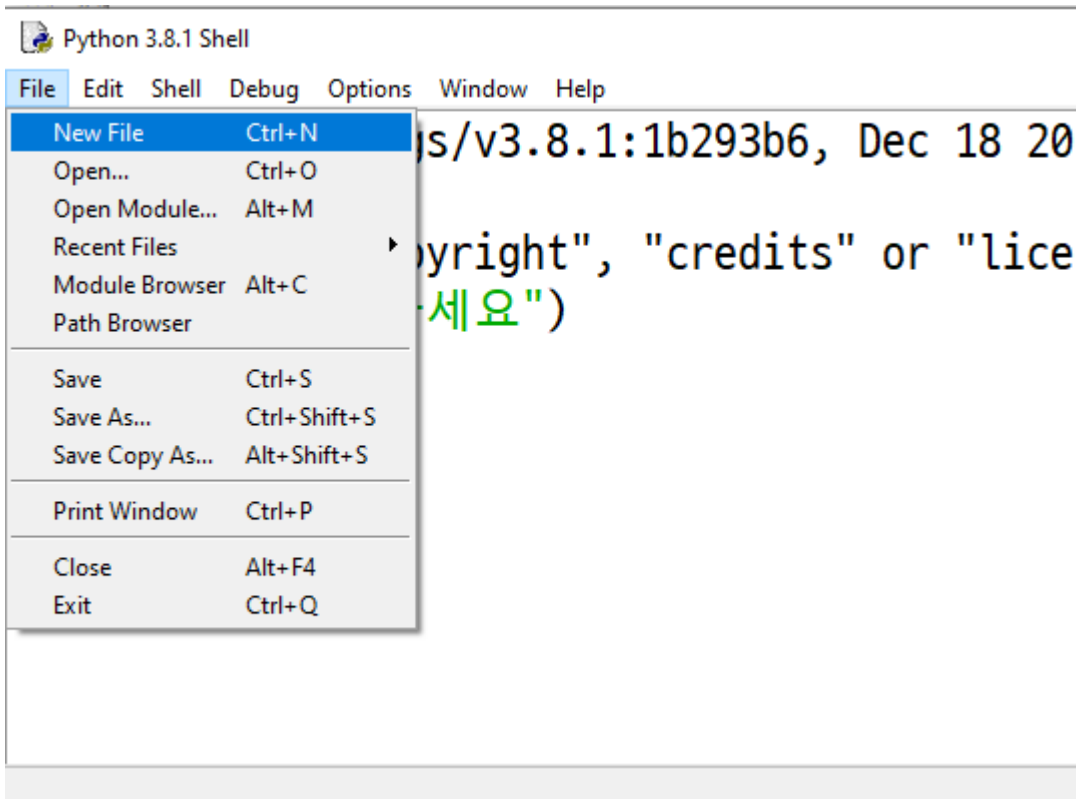


The screenshot shows the IDLE Shell 3.10.4 window. The title bar reads "IDLE Shell 3.10.4". The menu bar includes "File", "Edit", "Shell", "Debug", "Options", "Window", and "Help". The main text area displays the following content:

```
Python 3.10.4 (tags/v3.10.4:9d38120, Mar 23 2022, 23:13:41) [MSC v.1929 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> print("안녕하세요")
안녕하세요
>>> |
```

IDLE에서 파일로 실행시키기

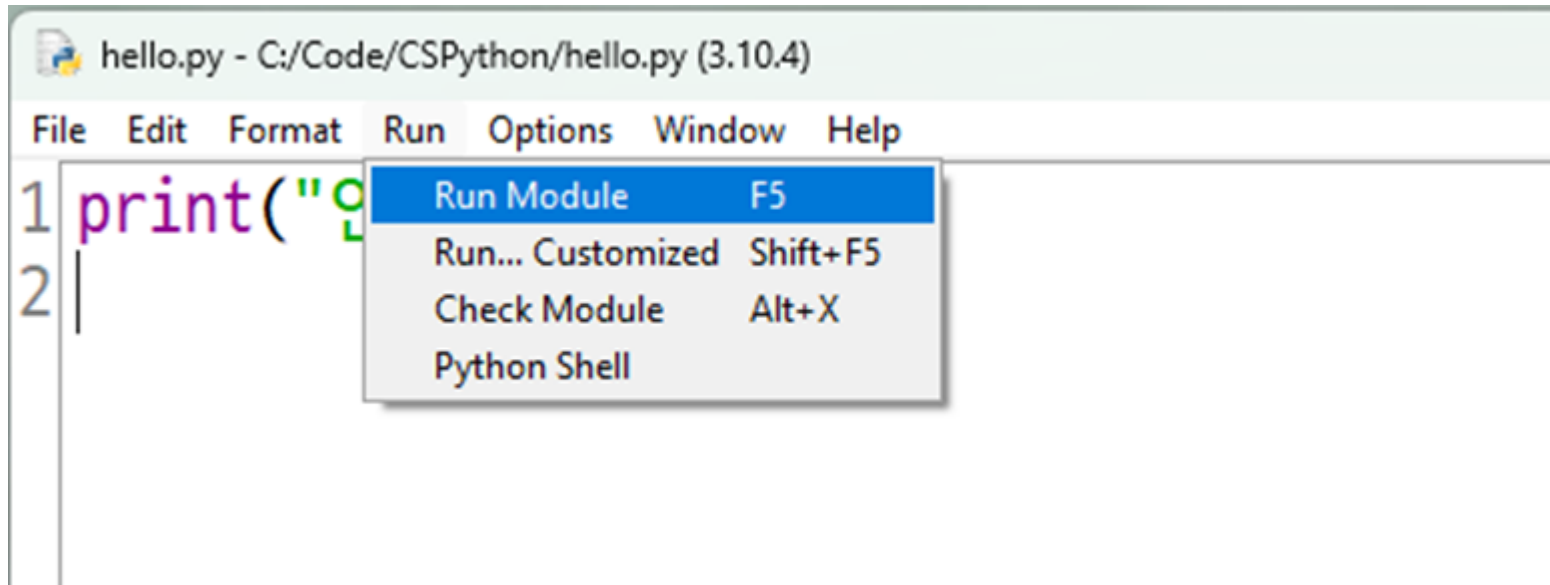
□ File | New File 메뉴 선택(또는 Ctrl+N)



□ 코드 작성 후 hello.py로 저장

IDLE에서 파일로 실행시키기

□ 실행



IDLE vs 전문적인 프로그래밍 에디터

□ IDLE

- 파이썬 프로그래밍에 최적화
- 통합 개발 환경
- 파이썬을 처음 배우는 초보자에게 적합

□ 다른 통합 개발 환경(Integrated Development Environment)

- 코드 작성, 실행, 디버깅 등은 한 곳에서 할 수 있도록 만들어진 프로그래밍 도구
- PyCharm, Spyder, PyDev

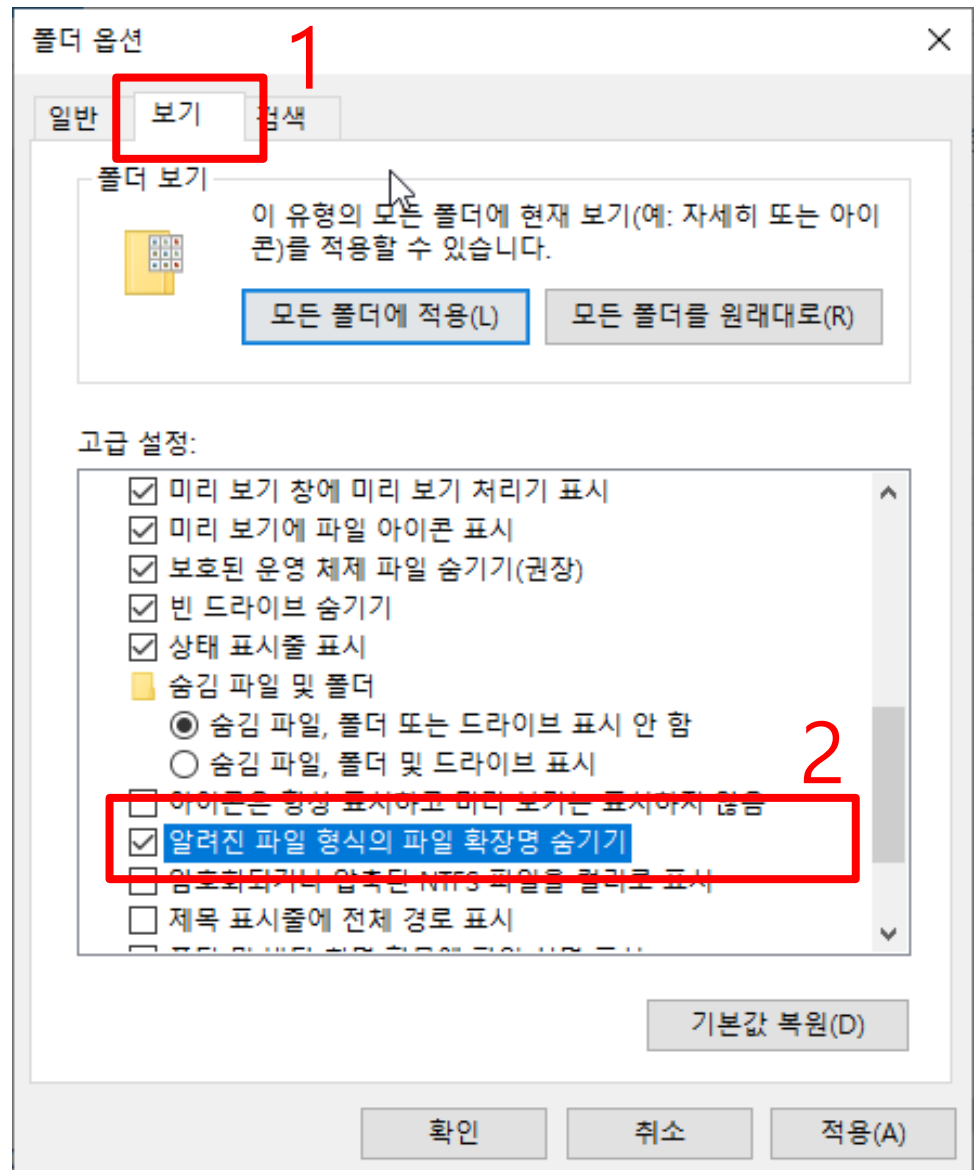
IDLE vs 전문적인 프로그래밍 에디터

▣ 전문적인 프로그래밍 에디터

- 다양한 언어 지원
 - ▣ 언어별 확장 기능 설치가 필요할 수 있음
- 전문적인 프로그래밍 지원 기능 제공
- 통합 개발 환경 기능 제공
- 여러 가지 언어를 다루게 되는 사람들에게 적합
- Visual Studio Code, Sublime Text 등

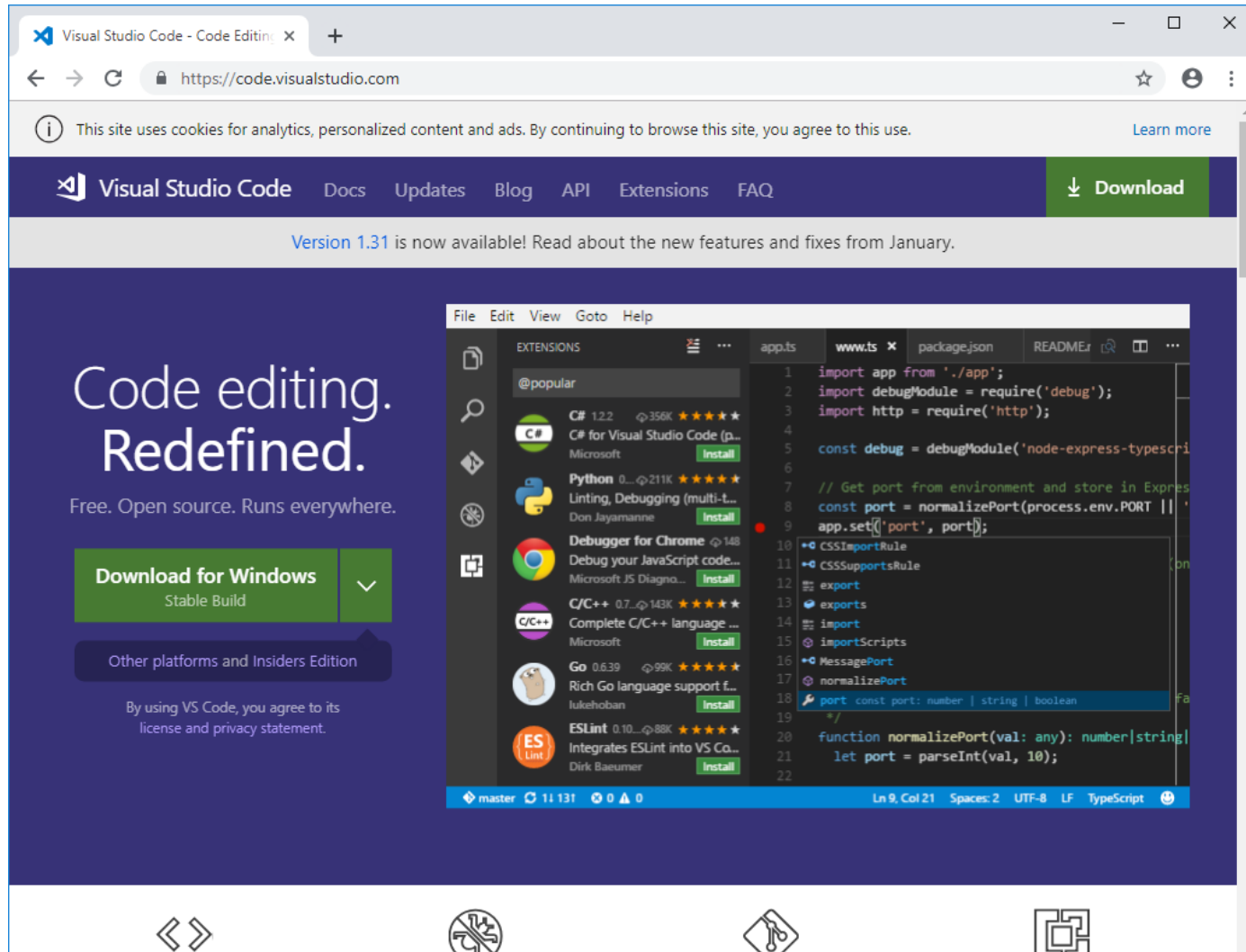
윈도우 탐색기 설정 변경

- 프로그래밍 할 때
파일의 확장자를
보이게 하는 것이
여러 모로 편함
- 탐색기 설정 변경

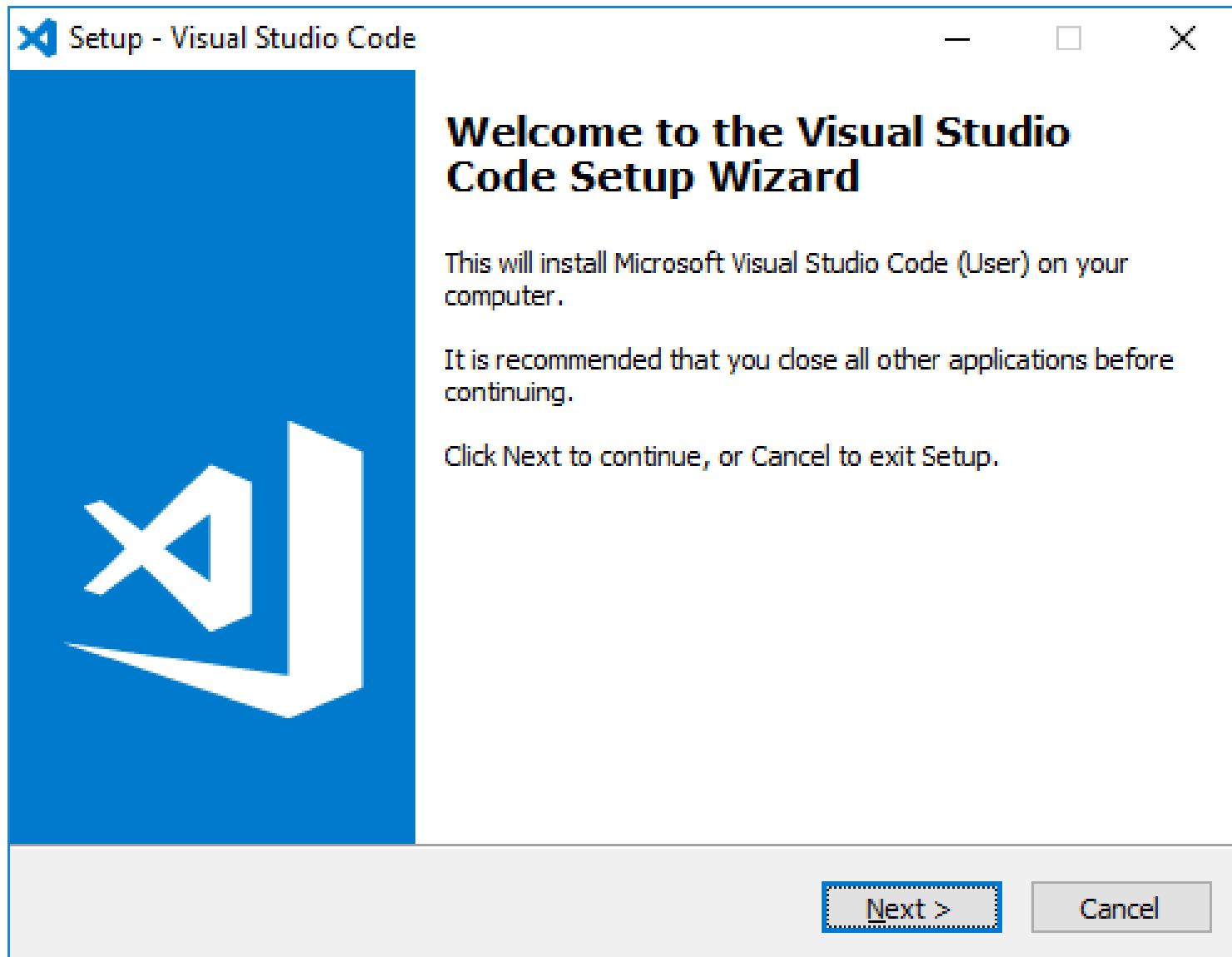


Visual Studio Code 다운로드

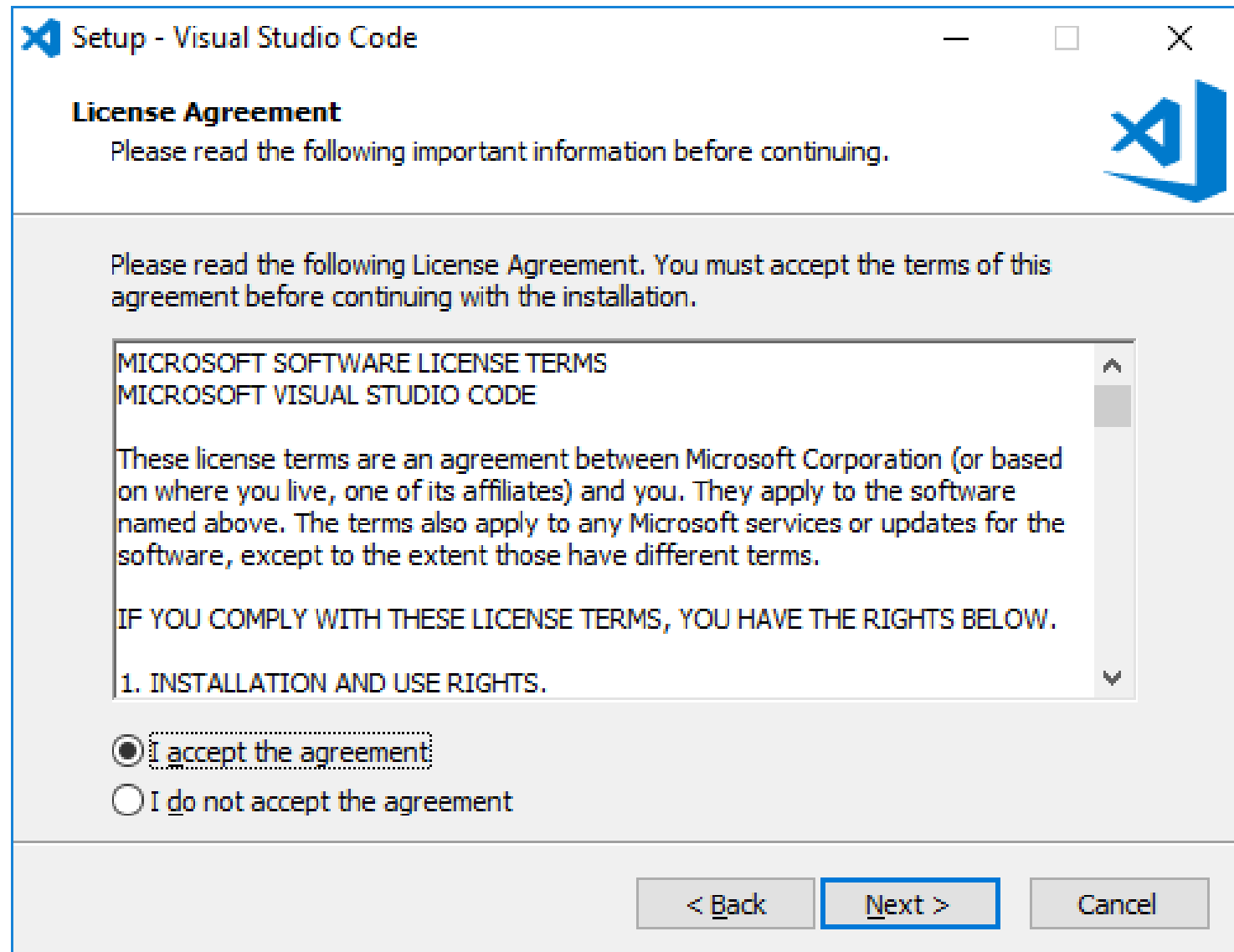
□ <http://code.visualstudio.com>



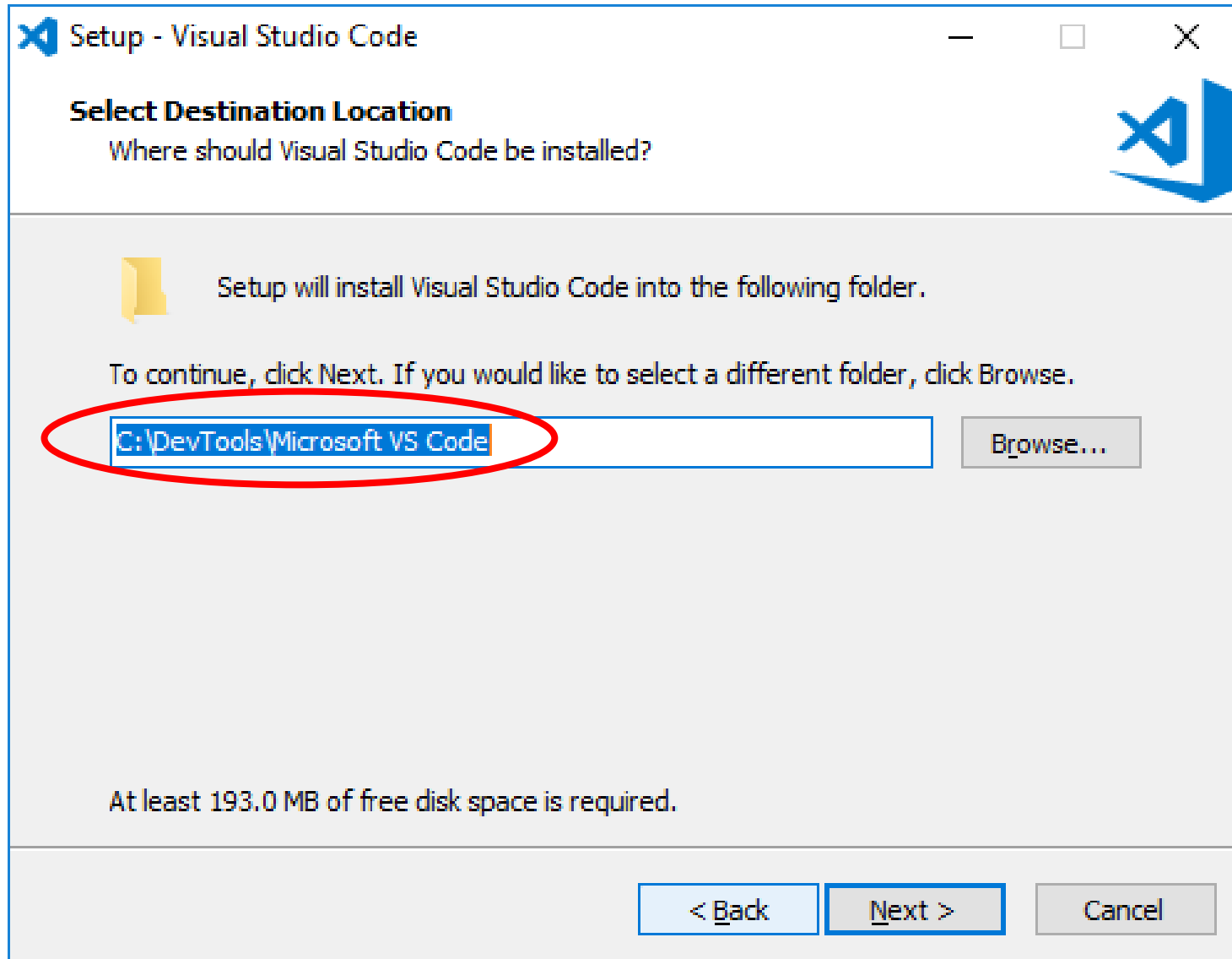
Visual Studio Code 설치



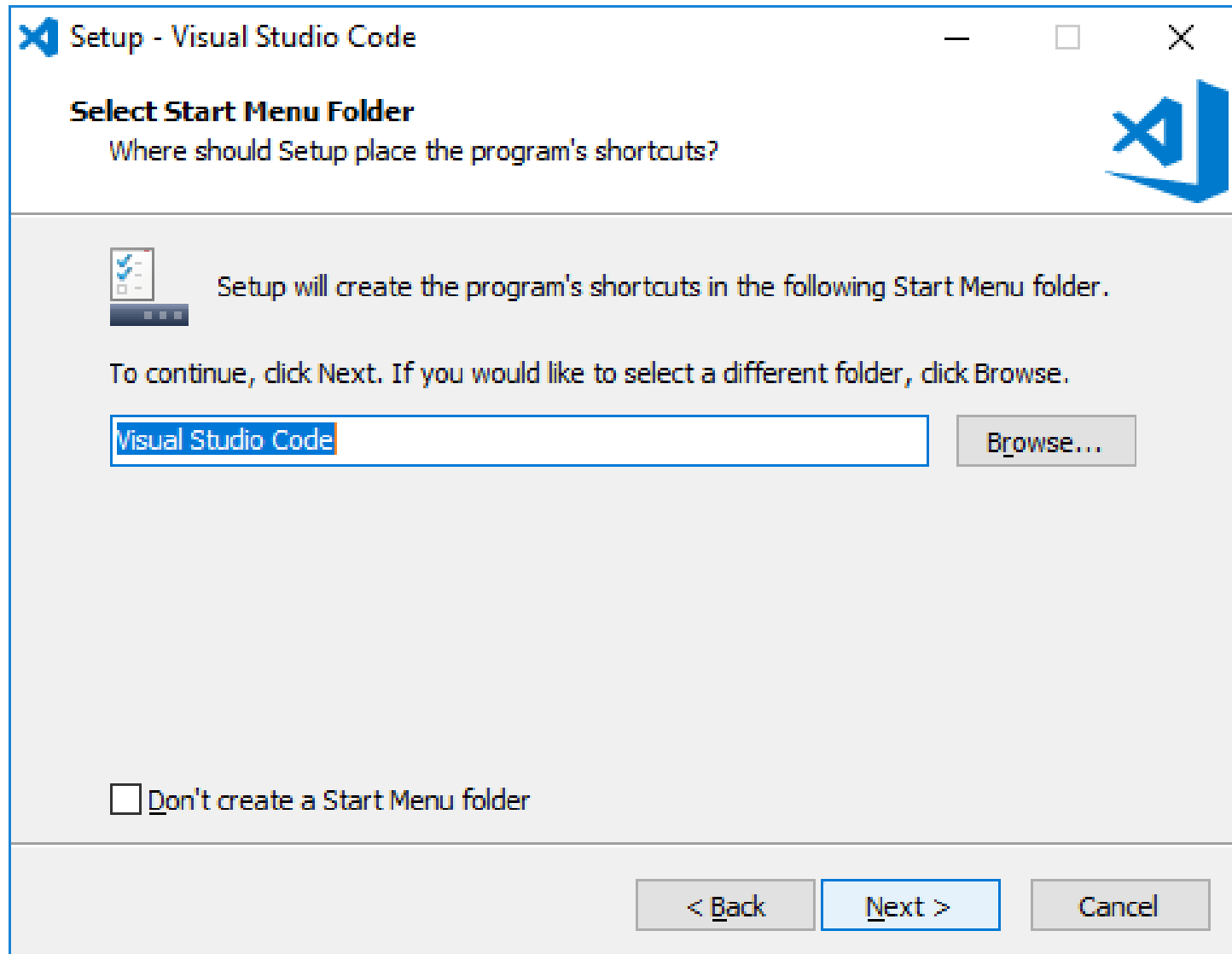
Visual Studio Code 설치



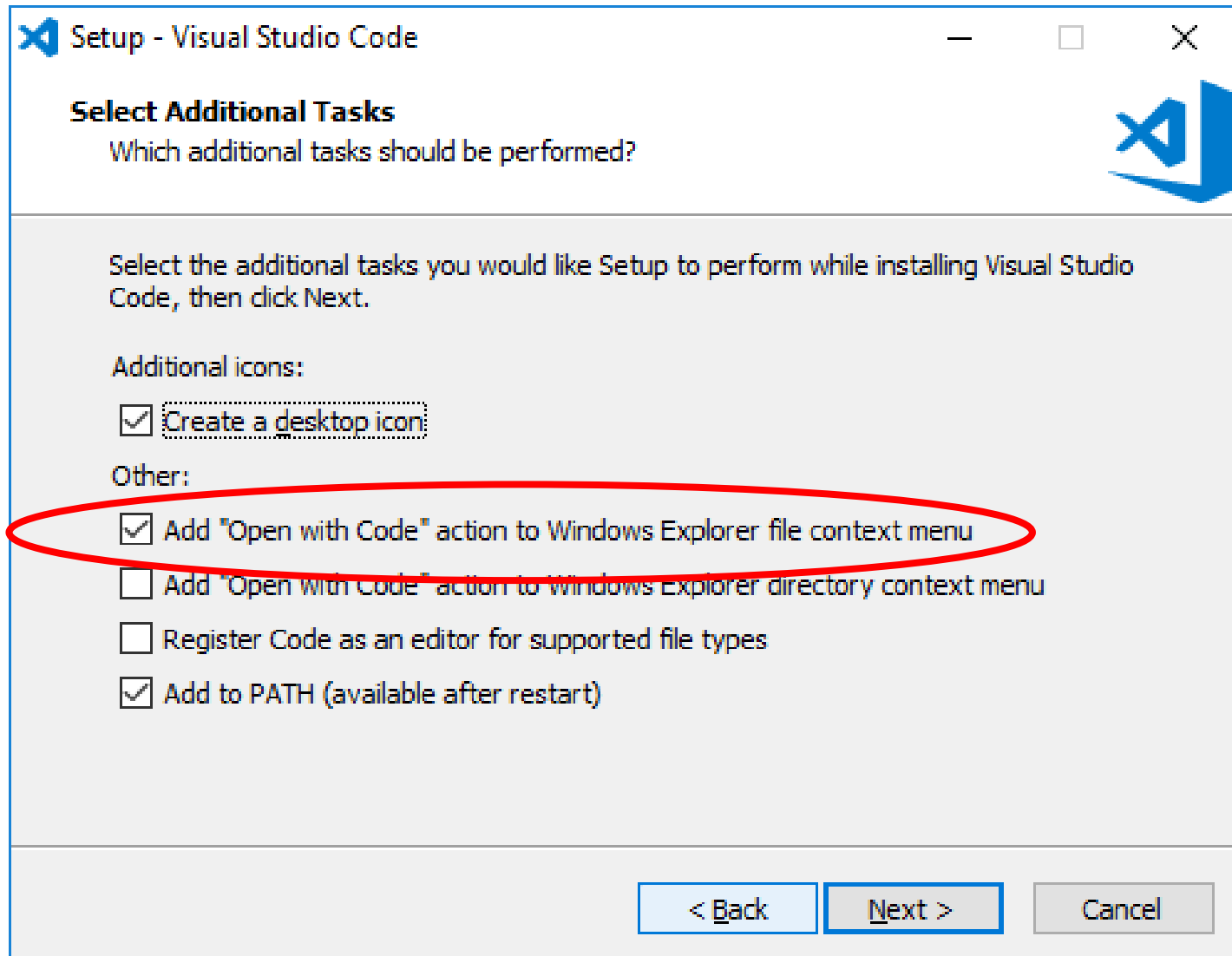
Visual Studio Code 설치



Visual Studio Code 설치



Visual Studio Code 설치



The image shows the 'Setup - Visual Studio Code' window. The title bar includes the Visual Studio Code logo and standard window controls. The main heading is 'Select Additional Tasks' with the subtitle 'Which additional tasks should be performed?'. A large blue Visual Studio Code logo is on the right. The instructions state: 'Select the additional tasks you would like Setup to perform while installing Visual Studio Code, then click Next.' There are two sections: 'Additional icons:' with a checked checkbox for 'Create a desktop icon'; and 'Other:' with four checkboxes. The first checkbox, 'Add "Open with Code" action to Windows Explorer file context menu', is circled in red. The other checkboxes are for 'Add "Open with Code" action to Windows Explorer directory context menu', 'Register Code as an editor for supported file types', and 'Add to PATH (available after restart)'. At the bottom are buttons for '< Back', 'Next >', and 'Cancel'.

Setup - Visual Studio Code

Select Additional Tasks
Which additional tasks should be performed?

Select the additional tasks you would like Setup to perform while installing Visual Studio Code, then click Next.

Additional icons:

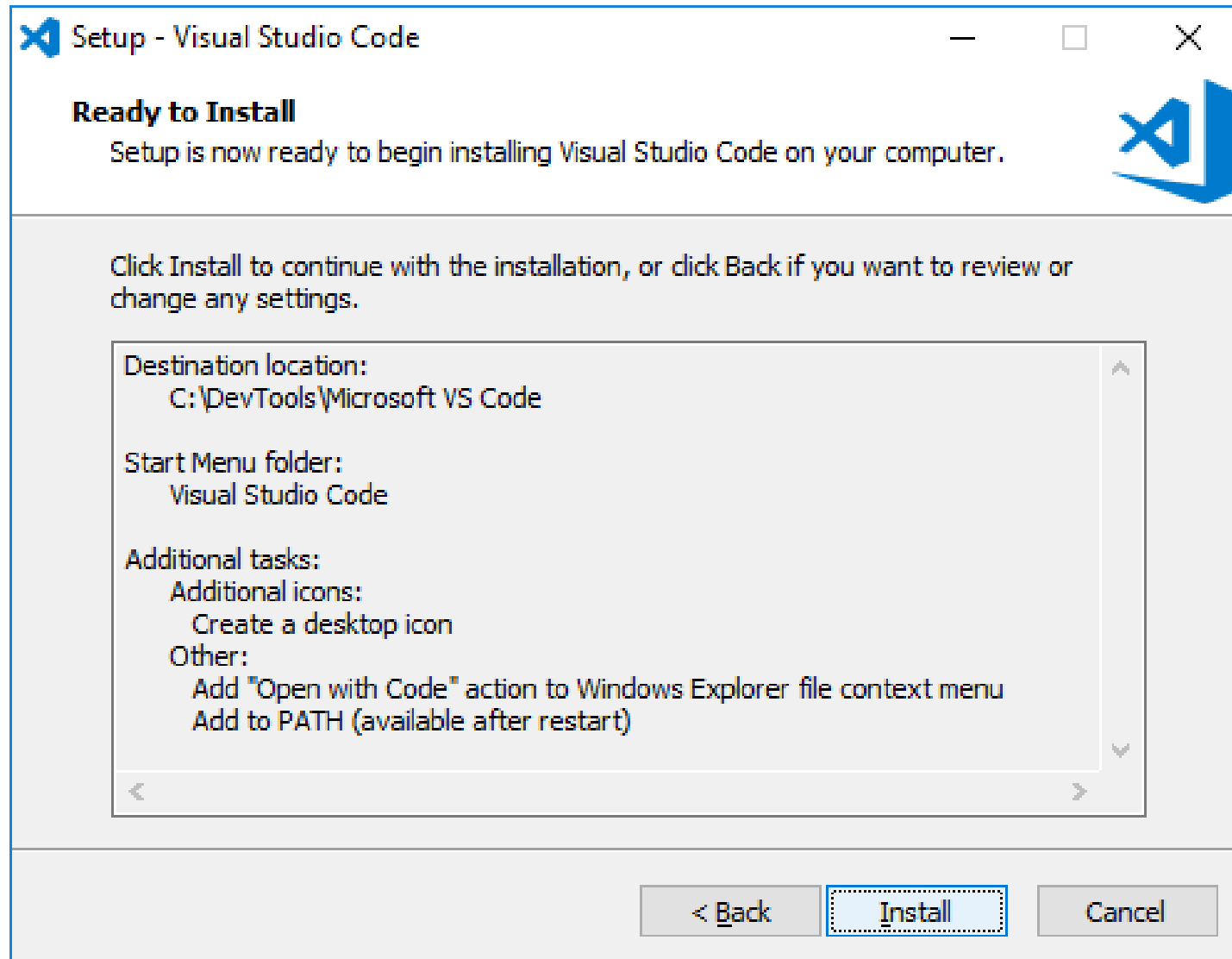
- ☒ Create a desktop icon

Other:

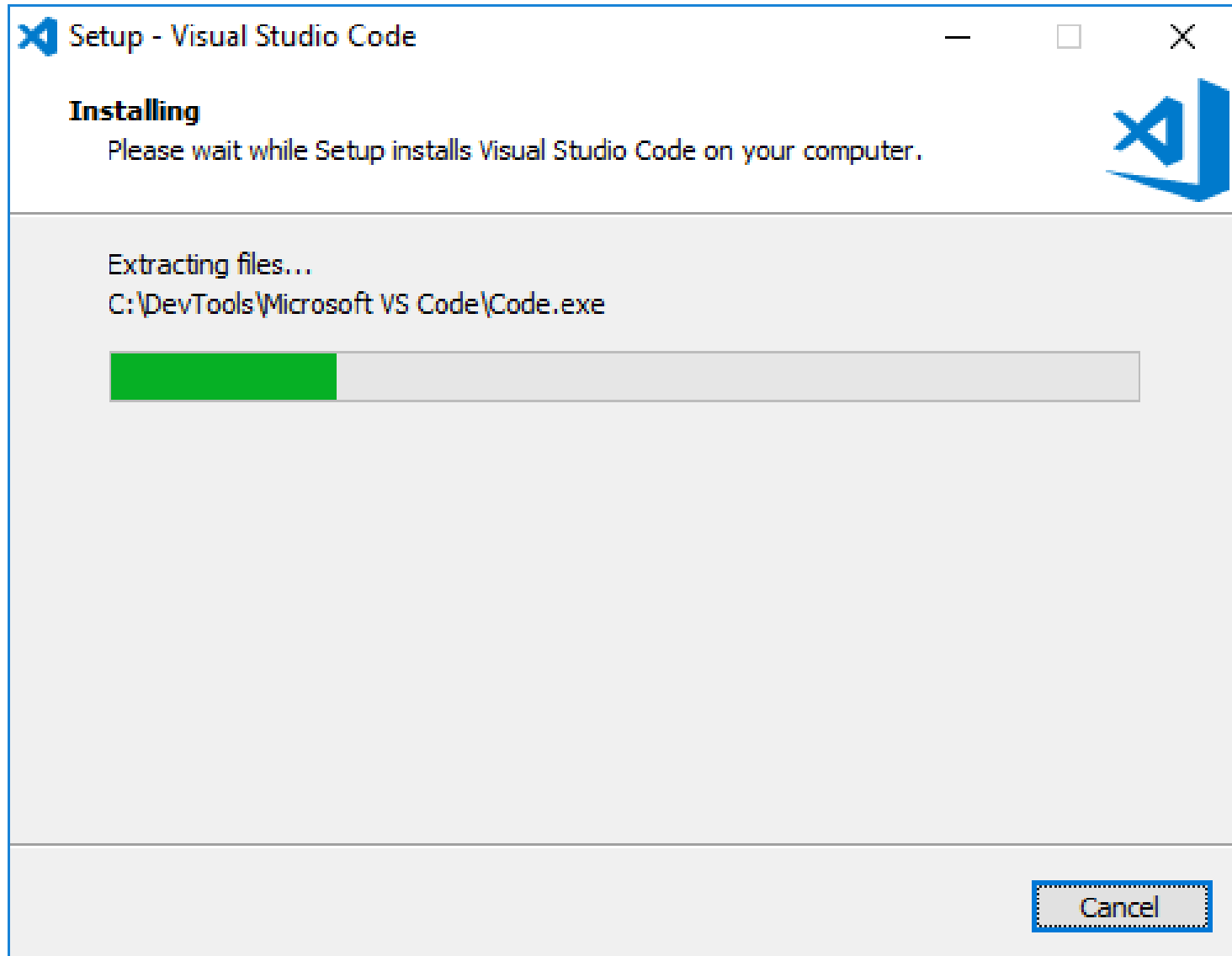
- ☒ Add "Open with Code" action to Windows Explorer file context menu
- ☐ Add "Open with Code" action to Windows Explorer directory context menu
- ☐ Register Code as an editor for supported file types
- ☒ Add to PATH (available after restart)

< Back Next > Cancel

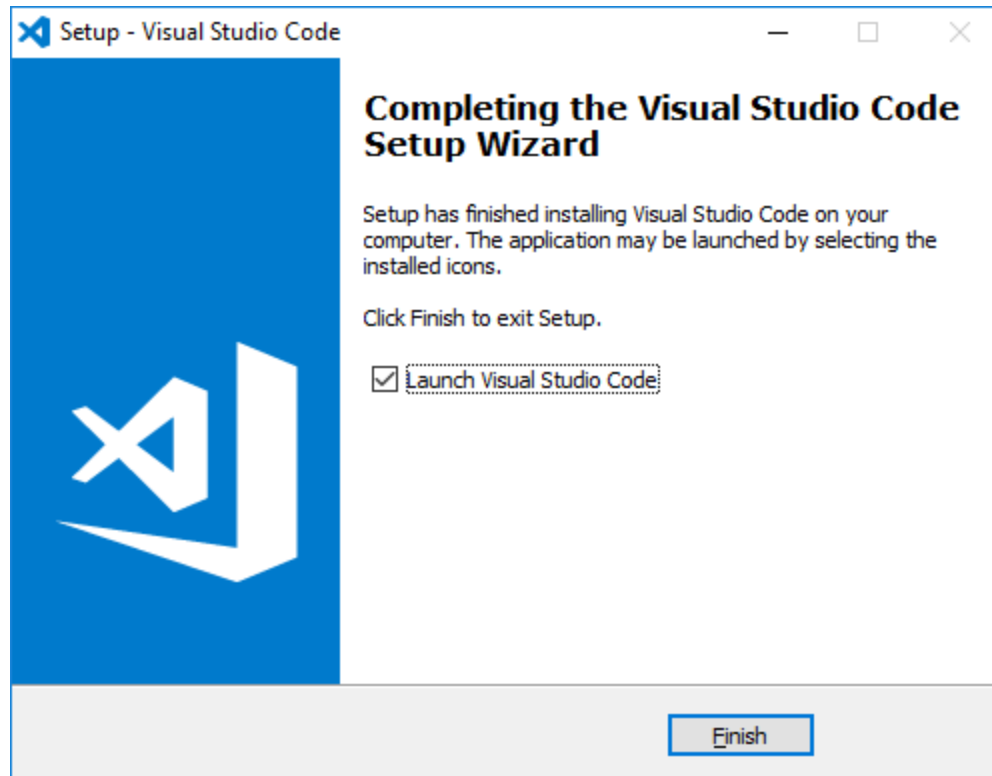
Visual Studio Code 설치



Visual Studio Code 설치

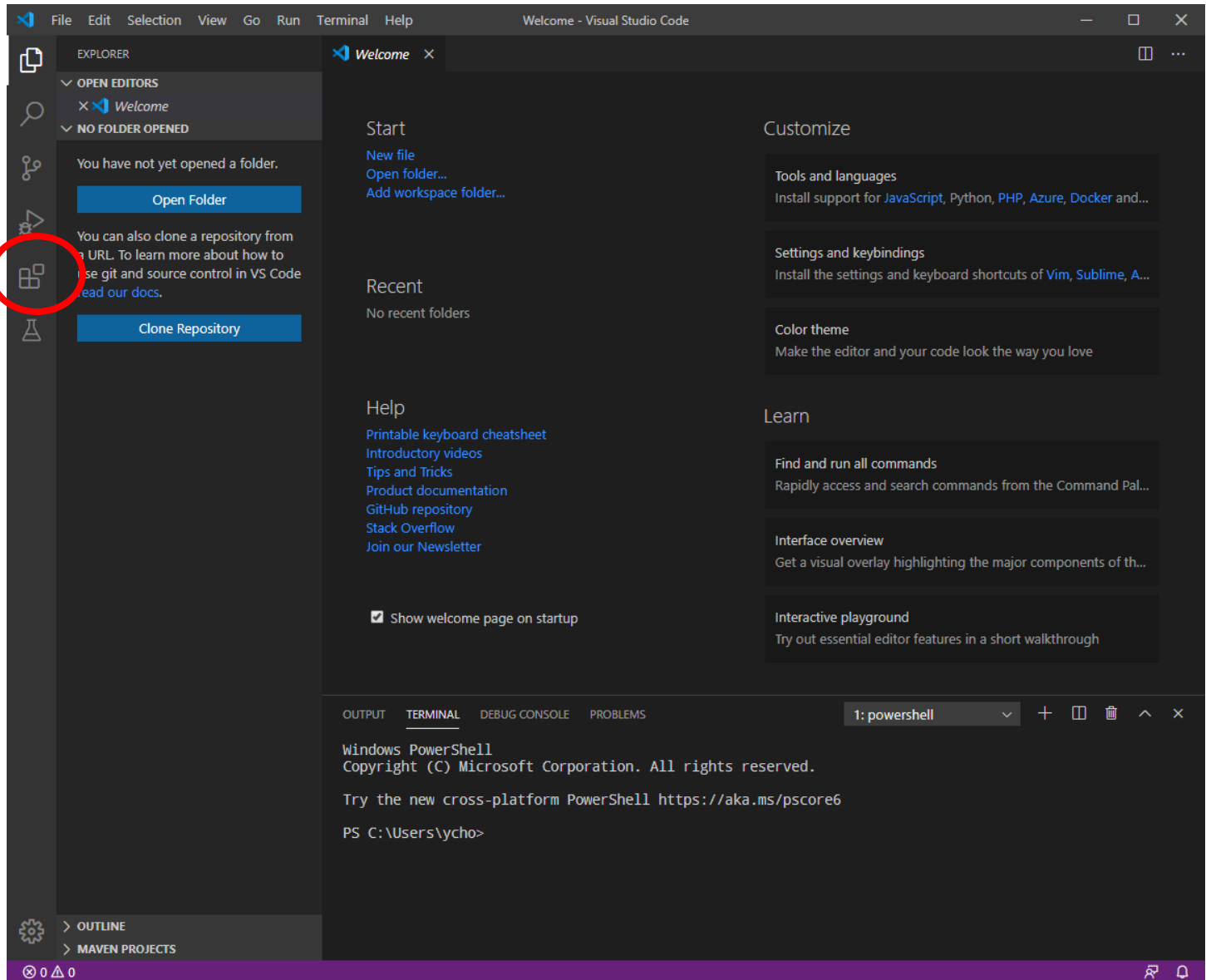


Visual Studio Code 설치



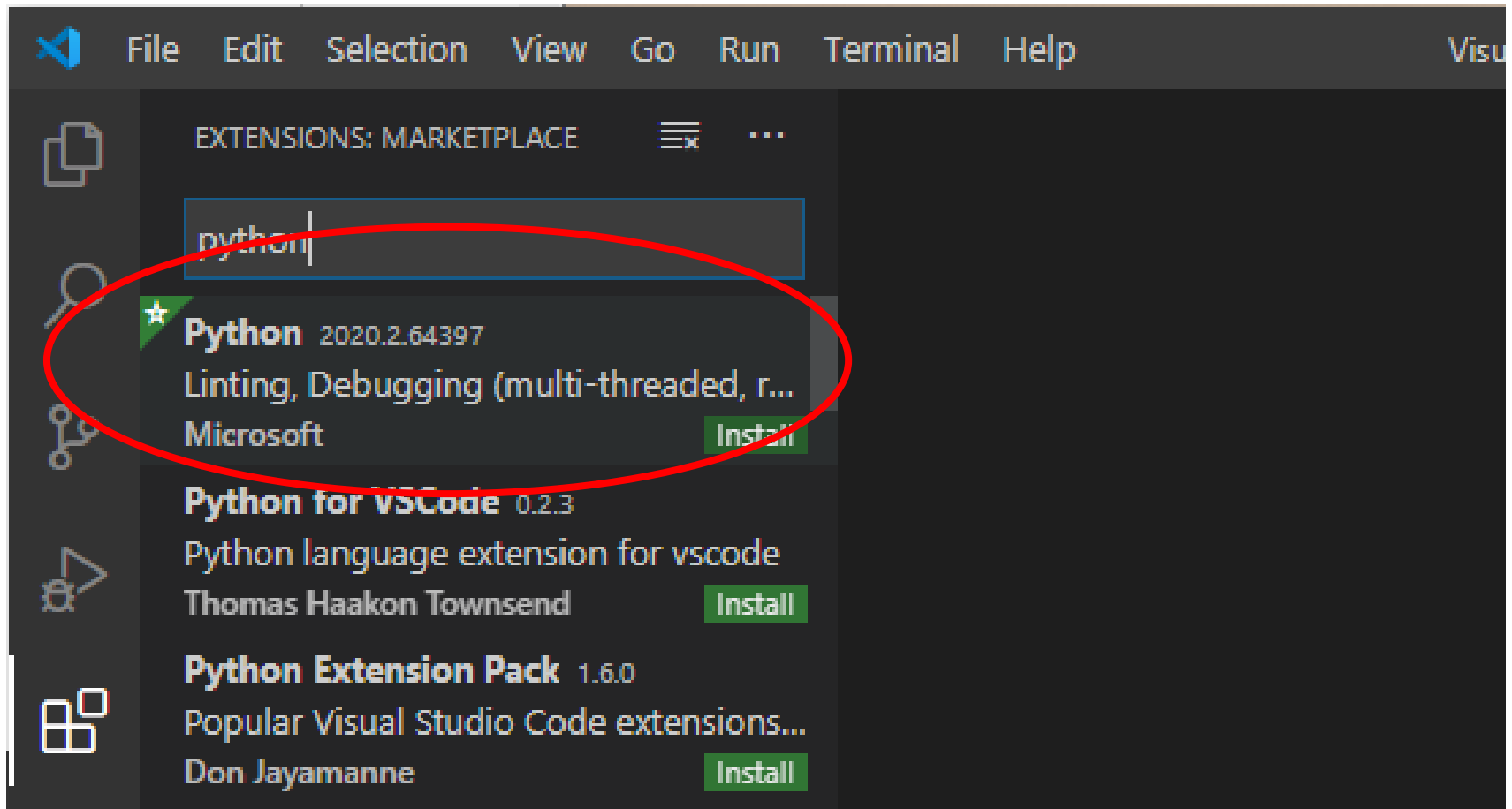
Visual Studio Code 설치

Extensions



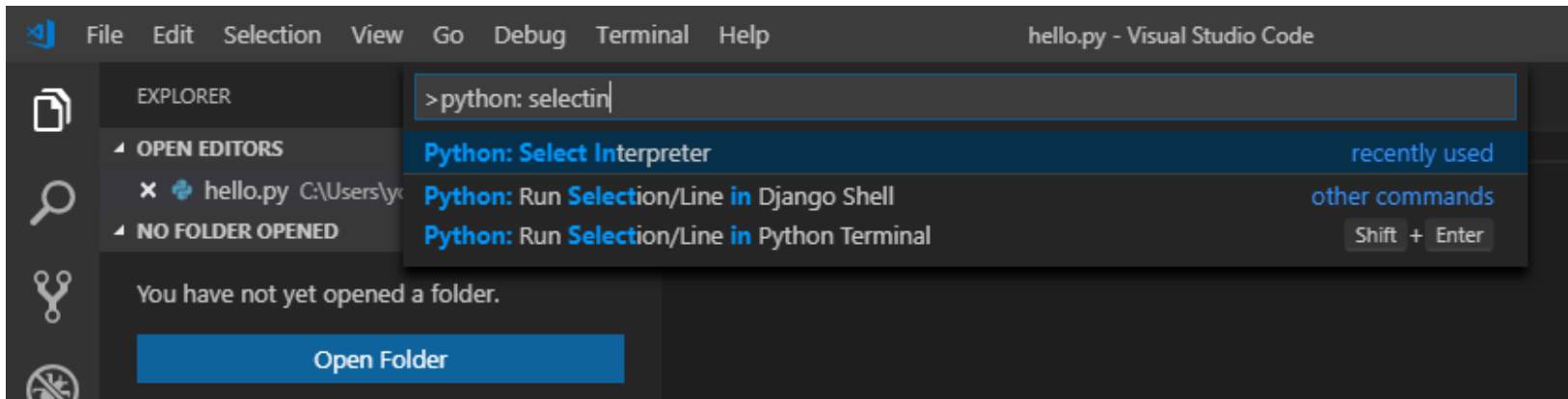
파이썬 extension 설치

- python으로 검색 후, Microsoft에서 제공하는 패키지 설치

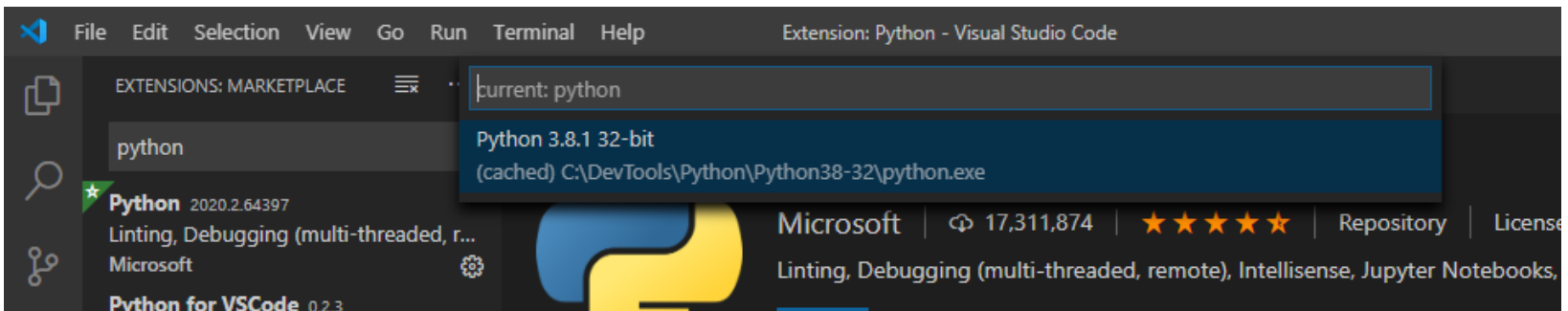


파이썬 인터프리터 선택

- Ctrl+Shift+P 입력
- python: select interpreter 선택



- 앞에서 설치한 파이썬 선택



파이썬 코드 작성 및 실행

- 새 파일을 열고 아래에서 보인 것처럼 입력하고 저장

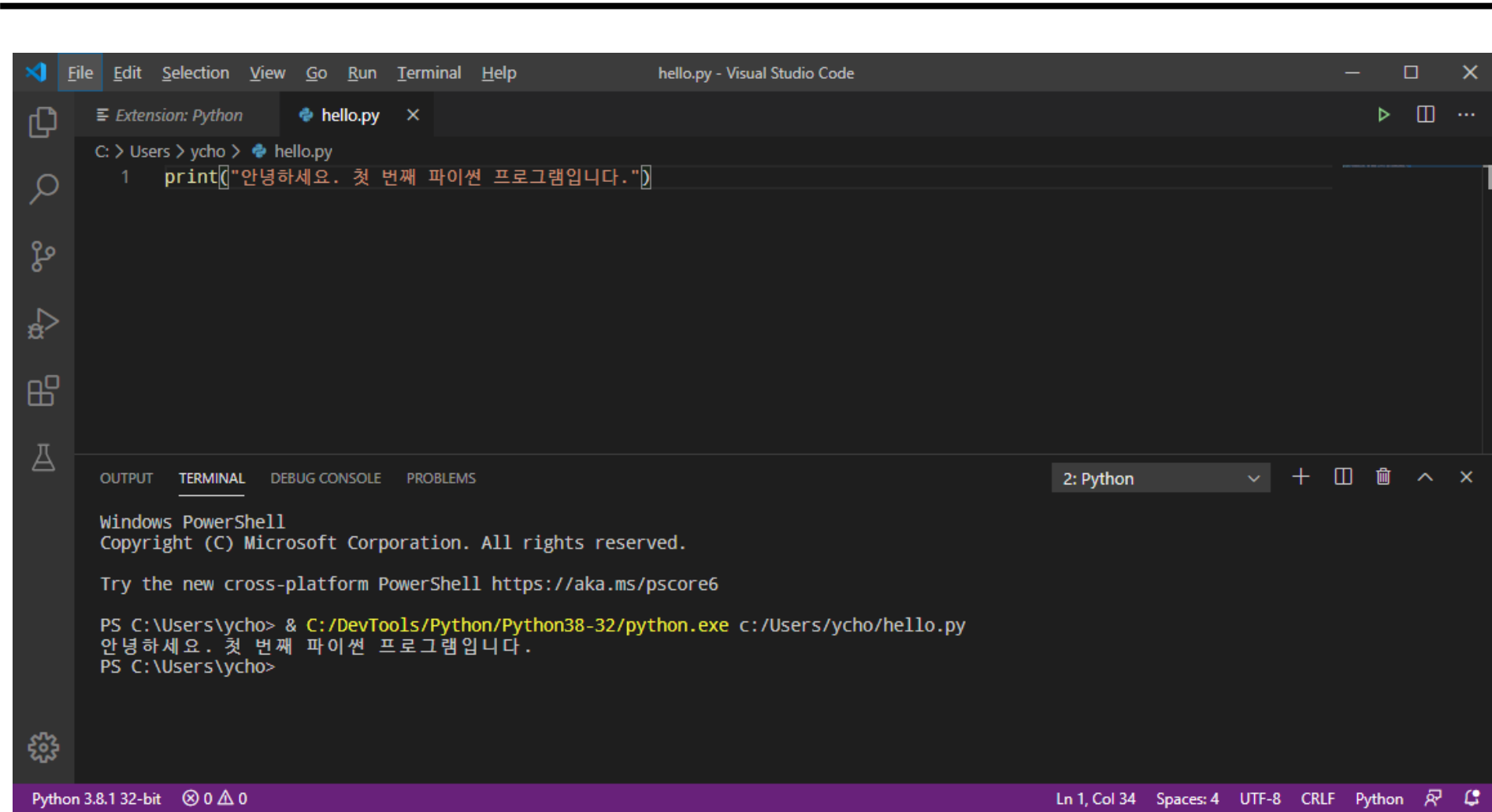
```
print("안녕하세요. 첫 번째 파이썬 프로그램입니다")
```

- 파일을 저장할 때 확장자는 .py로 지정

- 저장 후 실행

- 오른쪽 마우스 버튼 클릭 후 "Run Python File in Terminal" 선택

파이썬 코드 작성 및 실행



The image shows the Visual Studio Code interface with a Python file named `hello.py` open. The code in the editor is:

```
1 print("안녕하세요. 첫 번째 파이썬 프로그램입니다.")
```

The terminal window at the bottom shows the command to run the script and the output:

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

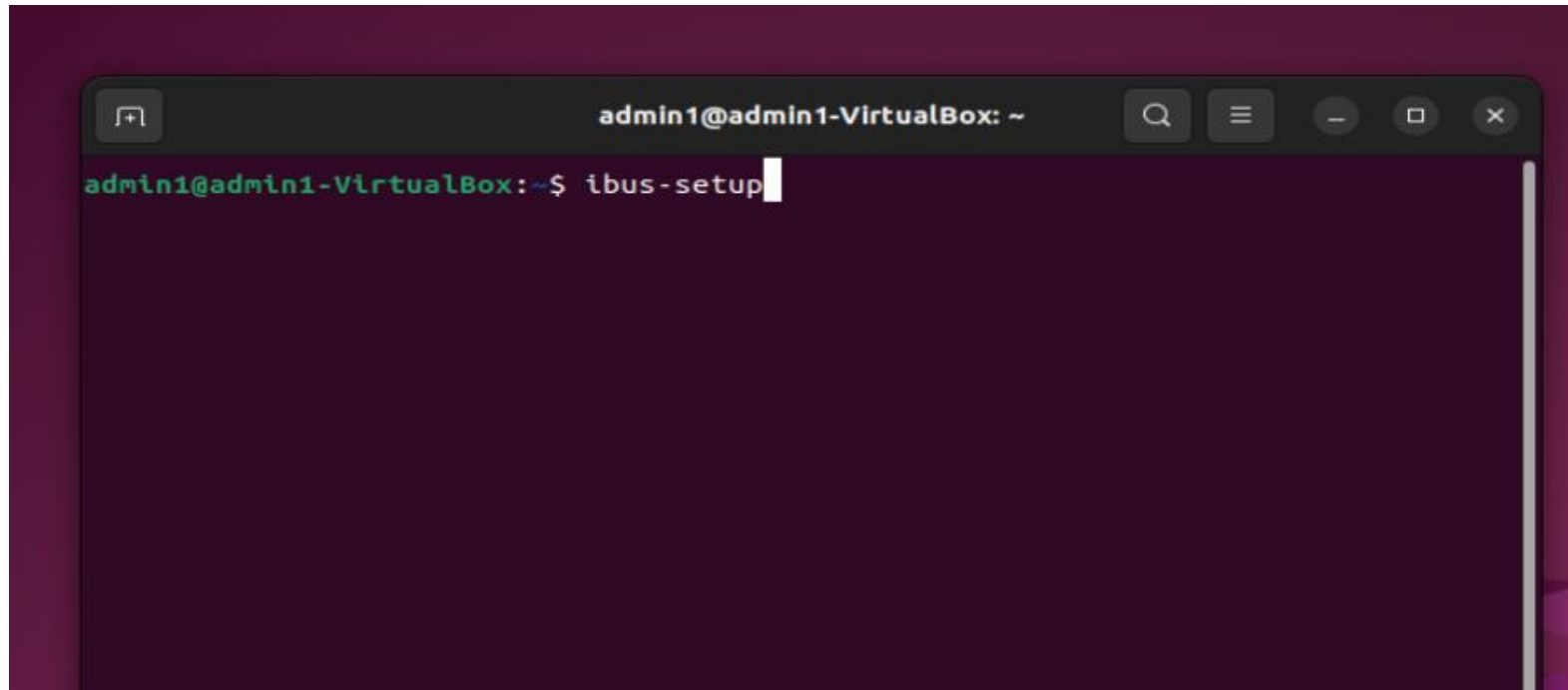
Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\Users\ycho> & C:/DevTools/Python/Python38-32/python.exe c:/Users/ycho/hello.py
안녕하세요. 첫 번째 파이썬 프로그램입니다.
PS C:\Users\ycho>
```

The status bar at the bottom indicates the Python version is 3.8.1 32-bit, and the file is encoded in UTF-8 with CRLF line endings.

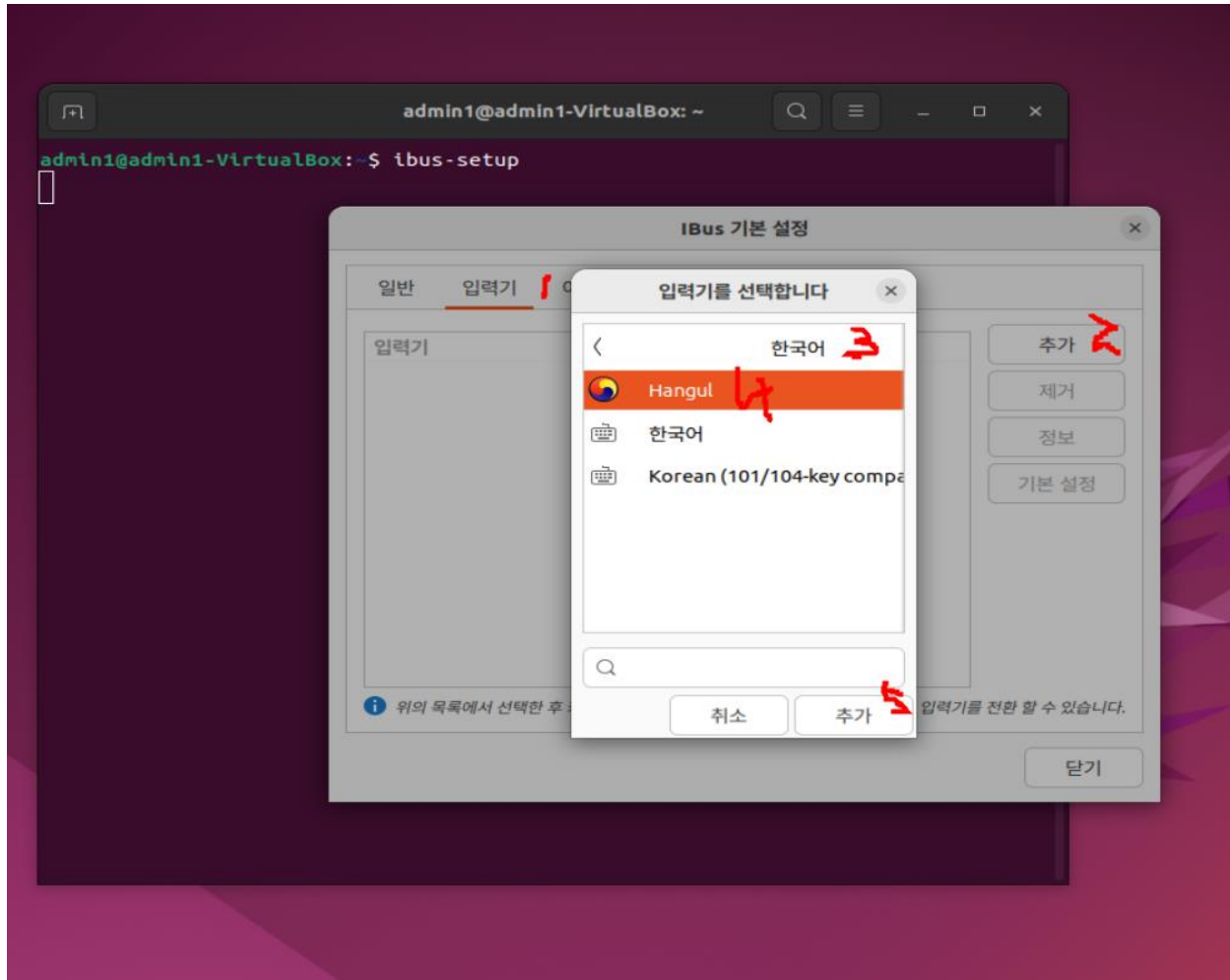
한글 입력 환경 설정

- ▣ Setup → teminal 실행 후
- ▣ ibus-setup 타이핑



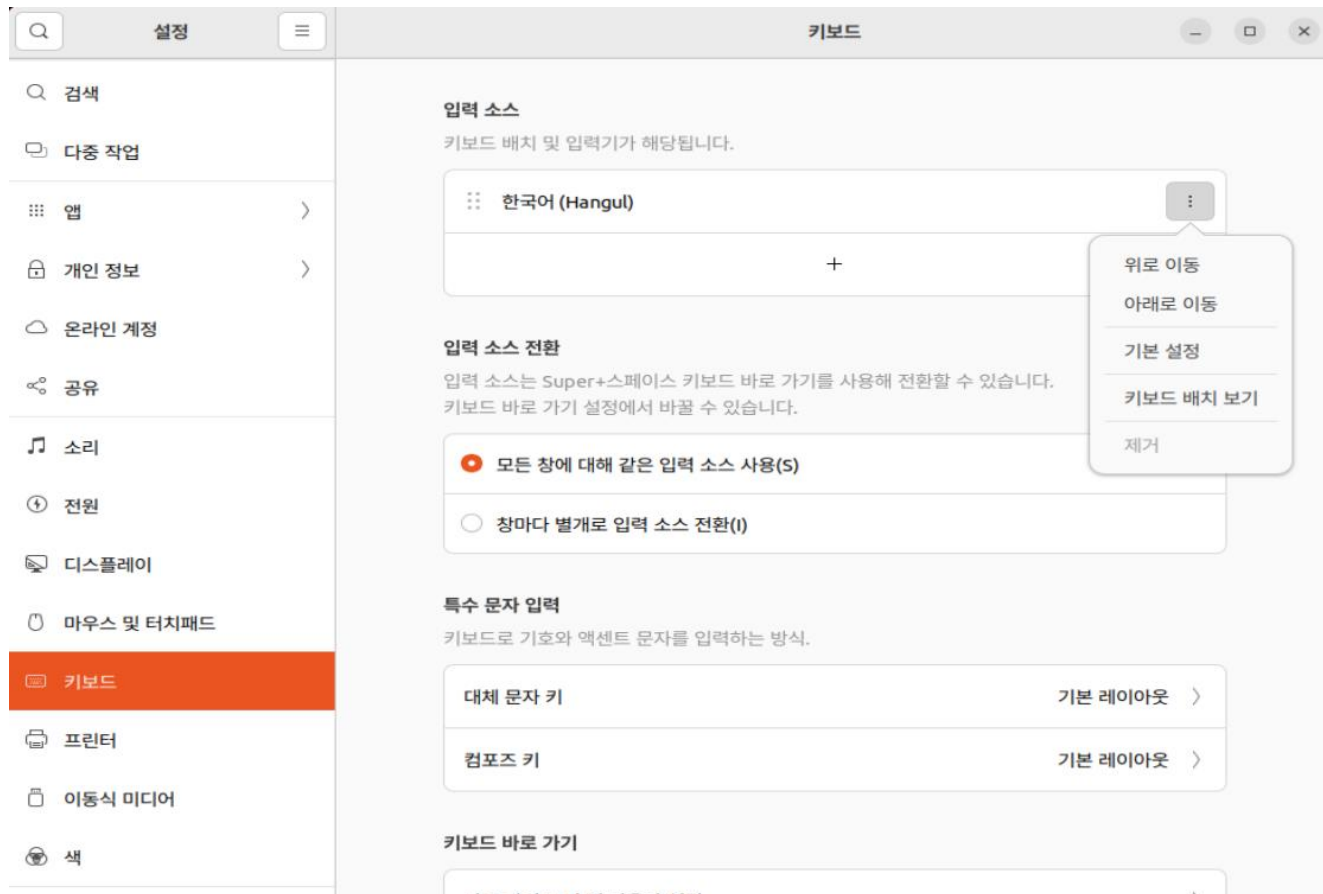
한글 입력 환경 설정

- 다음 단계를 거쳐서 hangul 을 찾고 설정 완료한다.



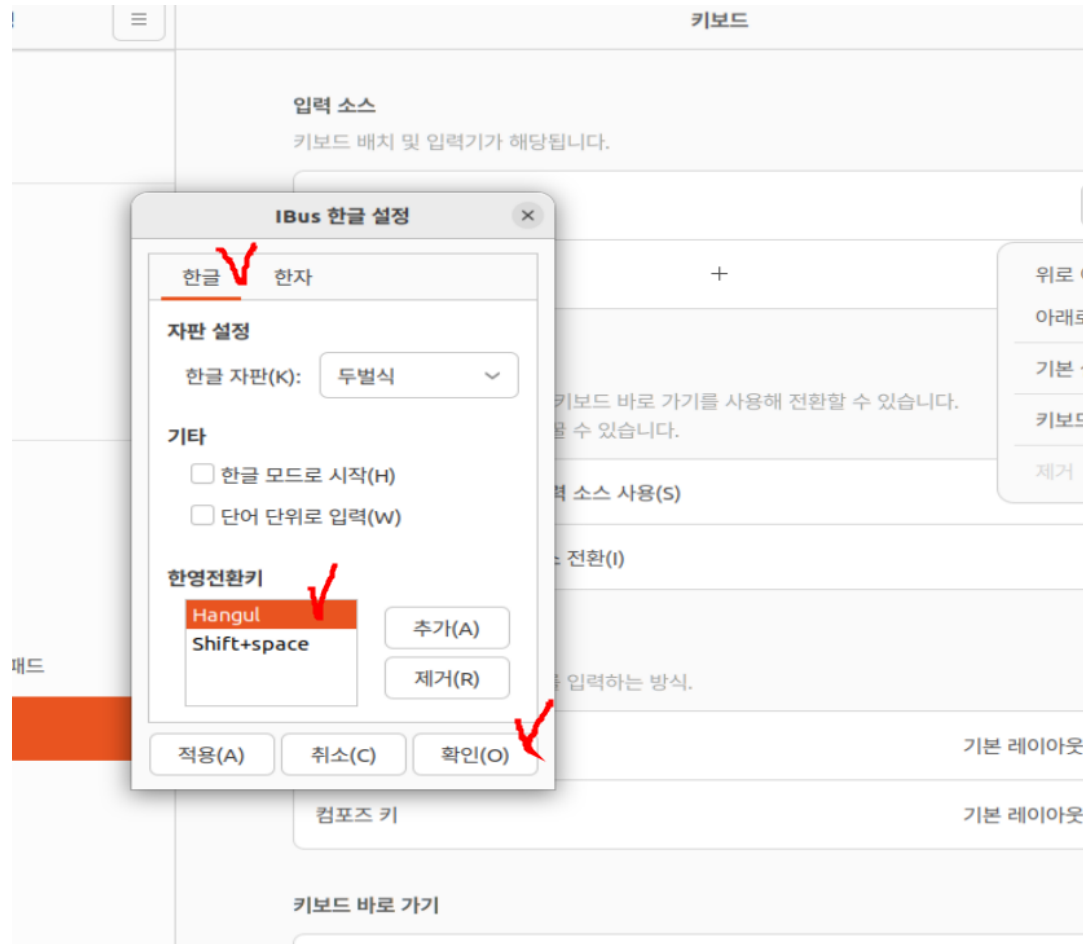
한글 입력 환경 설정

▣ 키보드 배치:설정→키보드→ Hangul →기본설정 이동



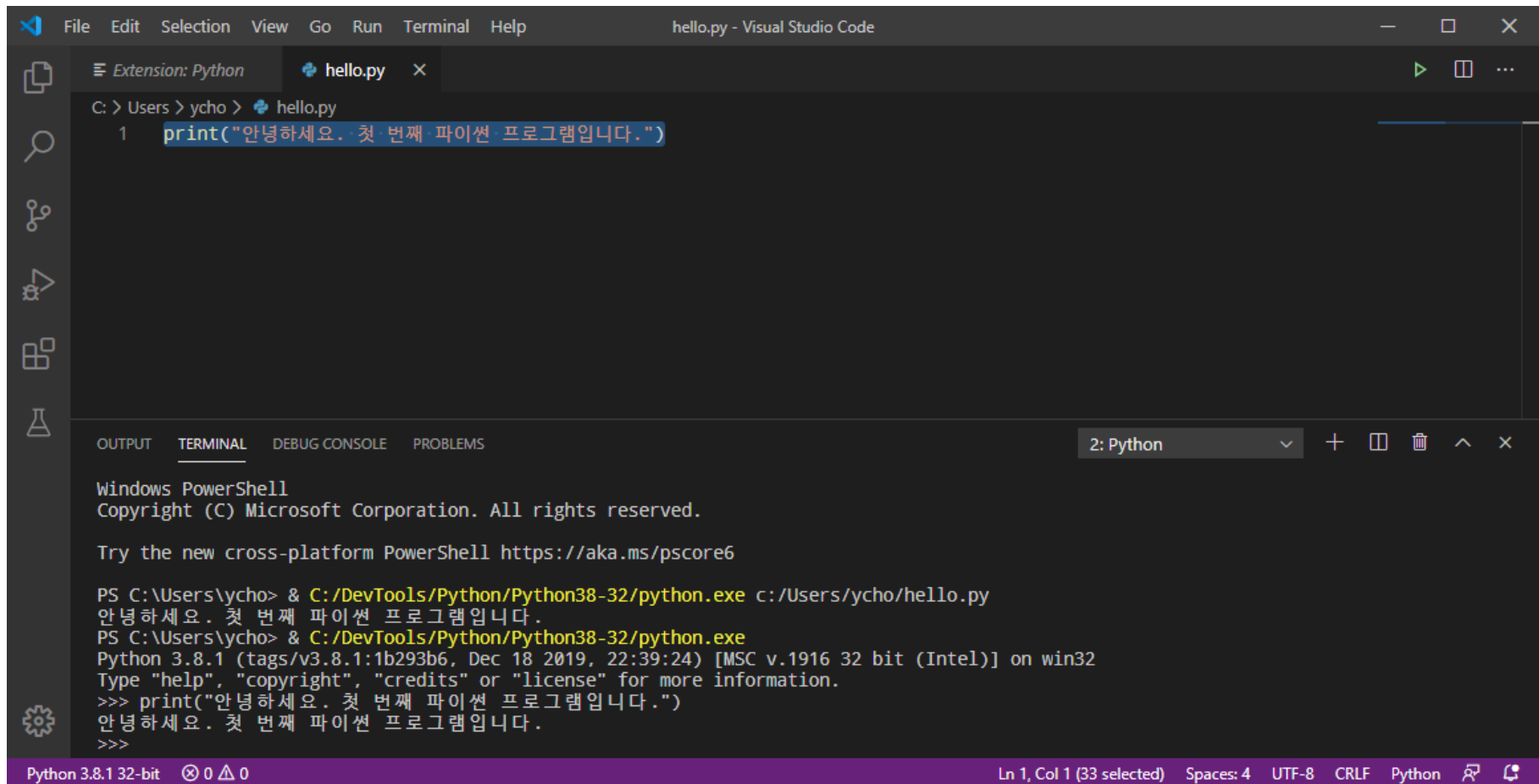
한글 입력 환경 설정

- 키보드 배치:전환키로 사용할 버튼을 선택하고 '확인' 완료



파이썬 코드 작성 및 실행

- ▣ 일부만 실행하고 싶으면 코드 선택 후 Shift-Enter



The screenshot shows the Visual Studio Code editor with a file named 'hello.py' open. The code in the file is:

```
1 print("안녕하세요. 첫 번째 파이썬 프로그램입니다.")
```

The code is selected, and the terminal at the bottom shows the execution output:

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

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\Users\ycho> & C:/DevTools/Python/Python38-32/python.exe c:/Users/ycho/hello.py
안녕하세요. 첫 번째 파이썬 프로그램입니다.
PS C:\Users\ycho> & C:/DevTools/Python/Python38-32/python.exe
Python 3.8.1 (tags/v3.8.1:1b293b6, Dec 18 2019, 22:39:24) [MSC v.1916 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> print("안녕하세요. 첫 번째 파이썬 프로그램입니다.")
안녕하세요. 첫 번째 파이썬 프로그램입니다.
>>>
```

The status bar at the bottom indicates 'Python 3.8.1 32-bit' and 'Ln 1, Col 1 (33 selected)'.

파이썬 코드를 저장할 때 주의할 점

□ 인코딩

- 문자를 컴퓨터가 어떻게 저장하는지를 나타냄
- 예: 우리가 'A'라고 보는 것은 실제로 컴퓨터에서는 65라고 저장될 가능성이 높음

□ 디코딩

- 숫자로 저장되어 있는 글자를 우리가 인식할 수 있는 문자로 변환

□ 인코딩 방식은 다양하게 있음

□ 파이썬 코드

- UTF-8 형식으로 저장
- 특히 코드에 한글이 들어있을 때 조심해야 함

파이썬 코드를 저장할 때 주의할 점

□ UTF-8

- 전 세계 문자를 표기할 수 있도록 만들어진 인코딩 방식
- 주로 웹에서 많이 사용됨
- 맥os나 리눅스는 기본적으로 UTF-8을 사용하는 경우가 많음

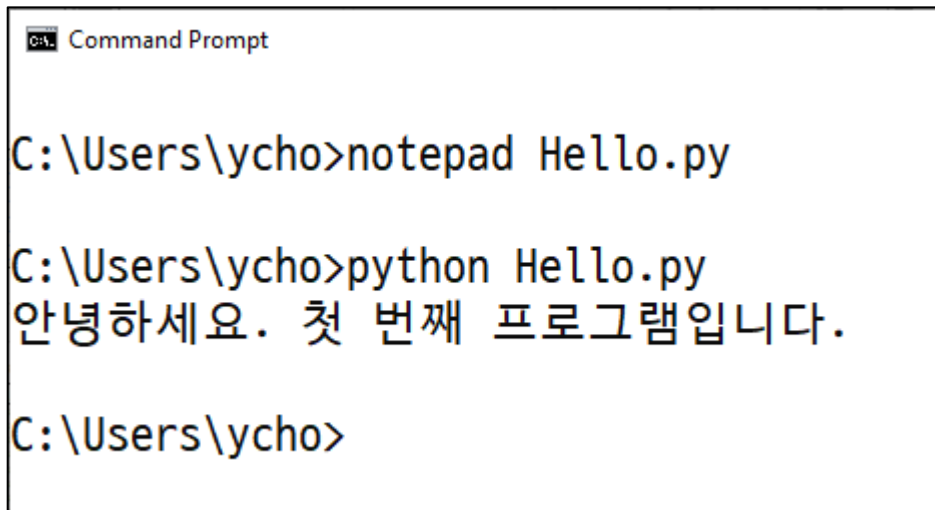
1	(START OF HEADING)	33	21	!	65	41	A	97	61	a
2	(START OF TEXT)	34	22	"	66	42	B	98	62	b
3	(END OF TEXT)	35	23	#	67	43	C	99	63	c
4	(END OF TRANSMISSION)	36	24	\$	68	44	D	100	64	d
5	(ENQUIRY)	37	25	%	69	45	E	101	65	e
6	(ACKNOWLEDGE)							102	66	f
7	(BELL)							103	67	g
8	(BACKSPACE)				72			104	68	h
9	(HORIZONTAL TAB)				73			105	69	i
A	(LINE FEED)							106	6A	j
B	(VERTICAL TAB)	43						107	6B	k
C	(FORM FEED)	44	2C		76	4C		108	6C	l
D	(DEVICE CONTROL 1)	45	2D			4D		109	6D	m
E	(DEVICE CONTROL 2)	46	2E	.		4E		110	6E	n
F	(DEVICE CONTROL 3)	47	2F	,		4F		111	6F	o
10	(DEVICE CONTROL 4)	48	30	0		50		112	70	p
11	(DEVICE CONTROL 5)	49	31			51		113	71	q
12	(DEVICE CONTROL 6)	50	32		82	52		114	72	r
13	(DEVICE CONTROL 7)	51				53		115	73	s
14	(DEVICE CONTROL 8)	52				54		116	74	t
15	(NEGATIVE ACKNOWLEDGE)	53				55		117	75	u
16	(SYNCHRONOUS IDLE)	54						118	76	v
17	(END OF TRANS. BLOCK)	55			87			119	77	w
18	(CANCEL)			8	88			120	78	x
19	(END OF MEDIUM)			9	89	59		121	79	y
1A	(SUBSTITUTE)	58	3A	:	90	5A	Z	122	7A	z
1B	(ESCAPE)	59	3B	;	91	5B	[123	7B	{
1C	(FILE SEPARATOR)	60	3C	<	92	5C	\	124	7C	
1D	(GROUP SEPARATOR)	61	3D	=	93	5D]	125	7D	~

ASCII UNICODE UTF-8

Cmd 창에서 실행

- Visual Studio Code 등을 이용해서 Hello.py 파일 생성
- Hello.py에는 다음 코드를 입력

```
print("안녕하세요. 첫 번째 프로그램입니다")
```
- 저장 후 Cmd창에서 실행



```
Command Prompt

C:\Users\ycho>notepad Hello.py

C:\Users\ycho>python Hello.py
안녕하세요. 첫 번째 프로그램입니다.

C:\Users\ycho>
```

첫 번째 파이썬 프로그램 분석

```
print("안녕하세요")
```

□ print()는 파이썬이 제공하는 명령

□ 명령

- 어떤 작업을 컴퓨터로 하여금 수행할 수 있도록 만들어진 코드의 묶음
- print() 명령은 주어진(전달된) 내용을 화면에 출력함
- 명령에 전달되는 내용을 전달인자(argument) 혹은 인자라고 함
- 파이썬에서 키워드(keyword)가 아닌 이름에 괄호()가 붙어 있으면 명령이라고 생각하면 됨
 - 키워드는 파이썬 언어에서 중요한 의미로 사용하겠다고 지정한 단어들

첫 번째 파이썬 프로그램 분석

- 큰 따옴표("") 또는 작은 따옴표(')는 코드에서 문자열 표시 방법
- 문자열
 - 문자들의 조합 또는 나열
 - 파이썬에서는 "... "와 '...'를 구별 안 함
 - 아래는 같은 결과를 출력하는 코드임

```
print("안녕하세요")  
print('안녕하세요')
```

첫 번째 파이썬 프로그램 분석

- 파이썬에서는 대문자와 소문자를 구별함

```
print("안녕하세요")  
Print('안녕하세요')
```

- 명령 이름을 잘못 쓰면 오류 발생

```
>>> Print("안녕하세요")  
Traceback (most recent call last):  
  File "<pyshell#0>", line 1, in <module>  
    Print("안녕하세요")  
NameError: name 'Print' is not defined
```

print() 명령

- print() 명령은 주어진 내용을 화면에 출력

```
>>> print(2)
2
>>> print(2.3)
2.3
```

- print() 명령에 산술 계산식을 전달하면 결과값 출력

```
>>> print(2 + 5)
7
>>> print(2 - 5)
-3
```

print() 명령

□ print() 명령과 줄바꿈 문자(new line character)

- print() 명령은 내용 출력 후 줄바꿈 문자를 출력

```
>>> print(2); print(3)
2
3
```

□ print() 명령과 콤마(',')

- 출력하는 내용을 콤마로 분리해서 인자로 전달하면 값들 사이를 공백 문자(' ')로 분리

```
>>> print(2, 3)
2 3
>>> print("원주율값: ", 3.1415)
원주율값: 3.1415
```

주석(Comments)

- 프로그래밍을 하다 보면 코드에 설명을 붙여야 하는 경우가 있음
 - 코드 저작 정보 표시 (저자, 수정 날짜, 버전 정보 등)
 - 코드 구현 방법 설명
 - 함수에 입력, 출력, 동작 내용 등 설명
- 프로그래밍 언어에서는 주석이라고 부름
- 파이썬에서 주석은 '#'문자로부터 시작됨
 - '#' 이후 줄바꿈(new line)이 일어날 때까지 코드가 실행 안됨
- 여러 줄로 구성된 주석은 ''' ... ''' 또는 """ ... """ 형태로 쓸 수 있음 – 본래 문자열 표기 방법 중 한 가지

주석(Comments)

```
#print(2.3 + 5.2) # 실행 안됨
print(2.3 + 5.2) # print() 명령은 실행 '#' 이후 무시

# 여러 줄로 주석 작성
'''

print(2.3 + 5.2)
print(2.3 - 5.2)
'''

# 큰 따옴표도 사용 가능
"""

print(2.3 + 5.2)
print(2.3 - 5.2)
"""
```

오류

□ 문법 오류(syntax error)

- 문법에 적합하지 않아 오류 발생

```
>>> print(2 : 5)
File "<stdin>", line 1
    print(2 : 5)
           ^
SyntaxError: invalid syntax
```

```
print(3 / 2) # 3 // 2대신 잘못 사용
```

```
print(2 / 0)
```

오류

- 실행 중 오류(버그)
 - 논리 오류(logical error)
 - 코드를 잘못 작성
 - 런타임 오류(run-time error) 또는 예외 (exception)
 - 시스템적인 문제나 사용자 잘못으로 나타나는 경우가 많음
- 실행 중 나타나는 오류를 수정하는 작업
 - 디버거(Debugger)를 사용하기도 함

실습문제 1

□ 문제

- 본인의 이름과 생년월일을 화면에 출력

□ 요구사항

- 생년월일은 년도-월-일 형태로 출력

□ 코드

```
print("홍길동")  
print("2000-1-1")
```

실습문제 2

□ 문제

- 본인의 이름과 생년월일을 출력하되, 날짜를 "2000년 1월 1일" 형태로 출력

□ 요구사항

- print() 명령의 콤마를 사용해서 년, 월, 일을 분리

□ 코드

```
print("홍길동")  
print("2000년", "1월", "1일")
```

print() 명령의 다른 사용법

□ end 옵션

- 줄바꿈 문자 대신 다른 것 출력하기

```
print(출력_내용, end="다른_문자(열)")
```

- 줄바꿈 문자 대체

```
>>> print("홍길동", end=" ")
홍길동 >>>
>>> print("홍길동", end = " (끝)")
홍길동 (끝)>>>
>>> print("홍길동", end = " (끝)\n")
홍길동 (끝)
>>>
```

print() 명령의 다른 사용법

□ sep 옵션

- 공백 문자 대신 다른 문자(열)로 분리

```
print(출력_내용, sep="다른_문자(열)")
```

- 공백 문자 대체

```
>>> print("홍", "길동")
```

```
홍 길동
```

```
>>> print("홍", "길동", sep = "\n")
```

```
홍
```

```
길동
```

```
>>> print("성: 홍", "길동", sep = " 이름: ")
```

```
성: 홍 이름: 길동
```

print() 명령의 다른 사용법

```
>>> print("홍", "길", "동", sep = "\n")  
홍  
길  
동
```

- print() 명령에 sep와 end를 같이 넣을 수 있음

```
>>> print("원주율 값", "3.1415", sep=": ", end="15\n")  
원주율 값: 3.141515
```

실습문제 3

□ 문제

- 본인의 이름과 생년월일을 출력하되, 날짜를 "2000/1/1" 형태로 출력

□ 요구사항

- 년, 월, 일은 숫자로 출력
- 생년월일은 이름과 같은 줄에 출력
- print() 명령은 분리해서 사용(두 개의 print() 명령 사용)

□ 코드

```
print("홍길동", end = ' ')\nprint(2000, 1, 1, sep = '/')
```

실습문제 4

▣ Seq 활용

기본적으로 sep은 공백입니다.
print("Hello", "World")

쉼표로 구분

print("Hello", "World", sep=", ")

하이픈으로 구분

print("Hello", "World", sep="-")

실습문제 5

기본적으로 end는 줄바꿈입니다.

```
print("Hello")
```

```
print("World")
```

공백으로 끝내기

```
print("Hello", end=" ")
```

```
print("World")
```

다른 문자열로 끝내기

```
print("Hello", end="!!!")
```

```
print("World")
```


실습문제 6

#한줄 출력

```
print("Python", "is", "fun", sep="-", end="!!!\n")
```

여러 줄을 연결하여 출력하기

```
print("First line", end=", ")
```

```
print("Second line", end=", ")
```

```
print("Third line")
```

실습문제 7

▣ 사용자로부터 이름과 생년월일을 입력 받아 특정 형식으로 출력해보자

사용자 입력 받기

```
name = input("이름을 입력하세요: ")
```

```
birth_year = input("태어난 해를 입력하세요: ")
```

```
birth_month = input("태어난 달을 입력하세요: ")
```

```
birth_day = input("태어난 일을 입력하세요: ")
```

print, sep, end 사용하여 출력



실습문제 8

▣ 날짜를 입력받고, (/ , - , .) 3가지 형식으로 출력해 보자

날짜 입력 받기

```
year = input("연도를 입력하세요: ")
```

```
month = input("월을 입력하세요: ")
```

```
day = input("일을 입력하세요: ")
```

슬래시(/)로 구분하여 출력



실습문제 9

▣ 여러 문장을 한 줄에 출력

여러 문장 출력

```
print("Python is fun", end='. ')
```

```
print("Let's learn Python", end='. ')
```

```
print("Programming is powerful", end='!\n')
```

실습문제 10

▣ 두개의 값 사칙 연산 결과 출력

수학 연산

$A = 5$ $b = 3$

덧셈, 뺄셈, 곱셈, 나눗셈 결과 출력

