

주요 학습 목표

- VSCODE의 설치
- anaconda 가상환경과 연결 하기
- vscode의 터미널 설정하기
- hello world 프로그램의 시작

Chapter 03. 통합 개발 환경 Editor VSCODE의 설치

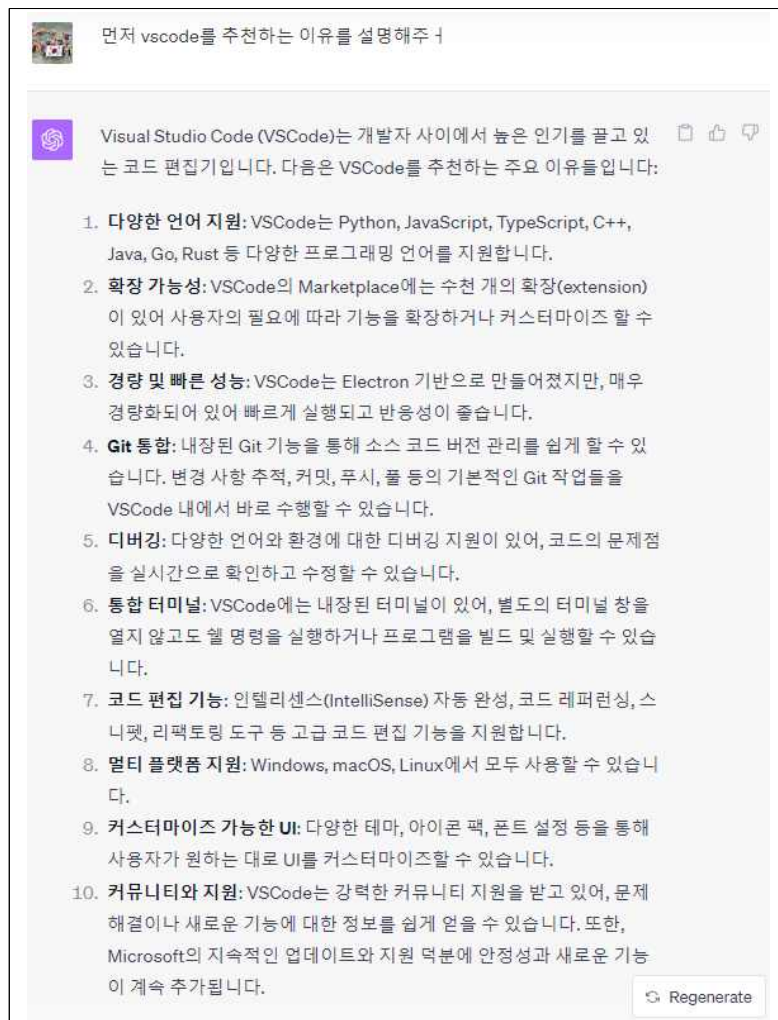
개발자라면 다음의 에디터들 중 몇가지를 꼭 들어보거나 사용 해보았을 것이다. - Sublime Text3 - UltraEditor - Notepad++ - Atom - WebStorm - Jupyter Notebook - Visual Studio Code 선택의 기준은 유/무료 여부, 속도, 한글 지원여부, 다양한 플러그인 지원 여부 등이 있을 것이다.

이중 오늘은 Visual Studio Code를 사용하기 위해 설치해 보려 한다. 간단한 특징은 다음과 같으니 참고하자.

- VSCODE
- 윈도우, 맥OS, 리눅스에서 사용가능
- 자바스크립트, 타입스크립트, Node.js 등의 지원 기능 내장
- 닷넷, C++, C#, 자바, 파이썬, PHP, Go 등 여러 언어 확장
- 타 에디터에 비해 가볍고 빠르다.
- 무료로 사용할 수 있으며 대부분 오픈소스다.

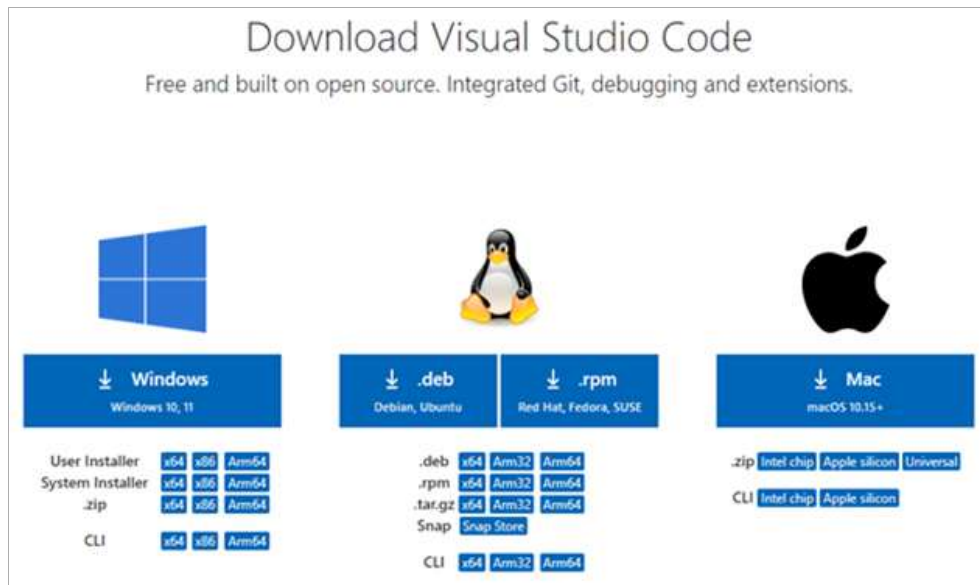
특히 본 교재에서는 파이썬을 다루게 됨으로 pycharm과 vscode 중에서 적지 않은 갈등을 할 수도 있다. 지금은 vscode에 집중해 보기로 하자. 만약 vscode가 pycharm에 비해서 부족함이 많다면 수 많은 개발자들이

나서서 개선을 요구할 것이고 점점 더 나아지고 좋아질 것이라는 데 의미를 두자. 모름지기 마이크로소프트사의 제품 아닌가. 설치 과정부터 차근차근 해 보자. 물론 chatGPT의 도움을 받아 해보기 바란다. 우리에게 10만년만에 만난 최고의 친구, 아니 어쩌면 독특한 쏘울의 프로그래머와 대화가 통하는 유일한 친구 절친이 될 것 같은 느낌적인 느낌이 있는 챗GPT가 있다. 설치에 앞서 절친 GPT에게 묻고 가지 않을 수 없다. 이쯤되면 이름도 하나 지어 줘야 할 것도 같은데 그것은 선택적으로 유보하고 다음 [그림 3-1]의 내용을 살펴보기 바란다.



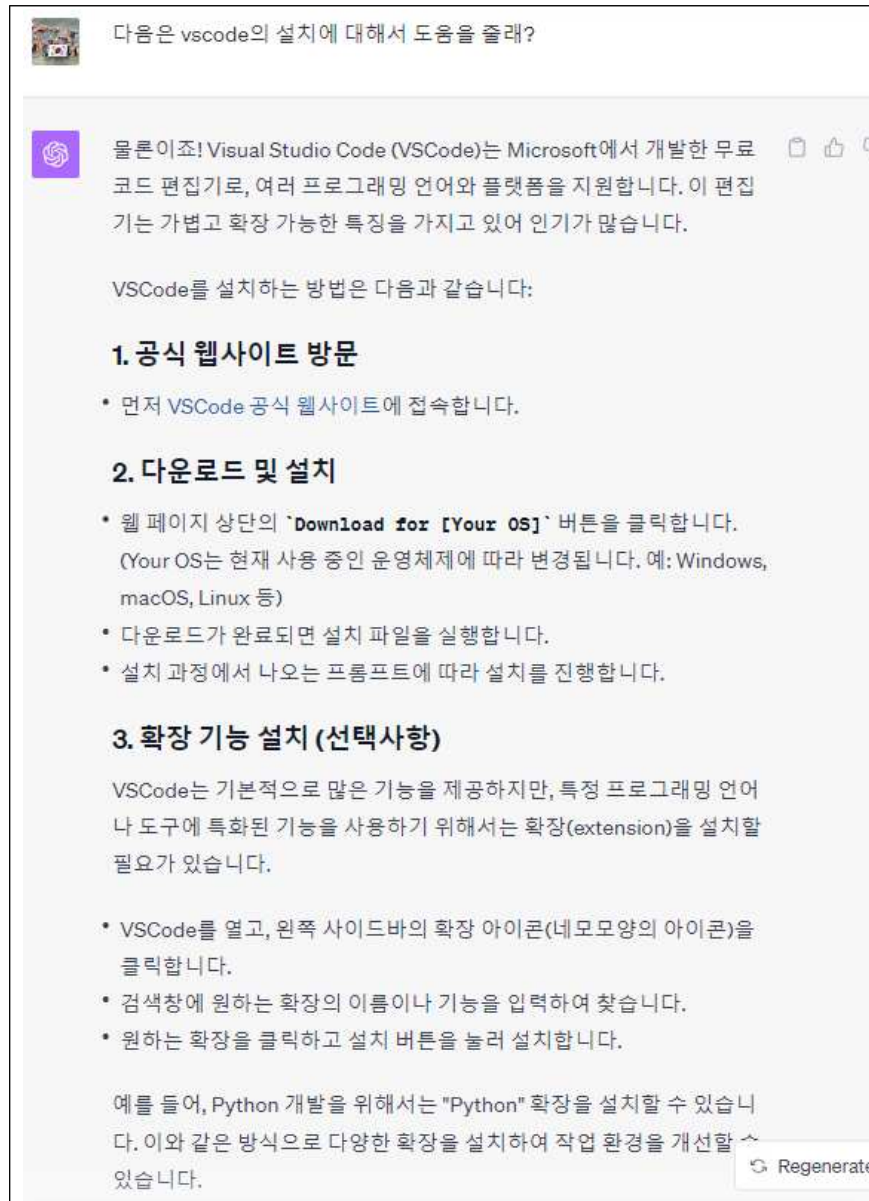
[그림 3-1] chatGPT에게 묻는 VSCODE의 추천이유

다음dml [그림 3-2]은 윈도우와 리눅스, iOS를 지원 하고있는 모습을 보여 주고 있다. 라즈베리파이는 리눅스 계열을 선택하면 된다.



[그림 3-2] VSCODE의 OS 지원

3.1 VSCode 의 설치

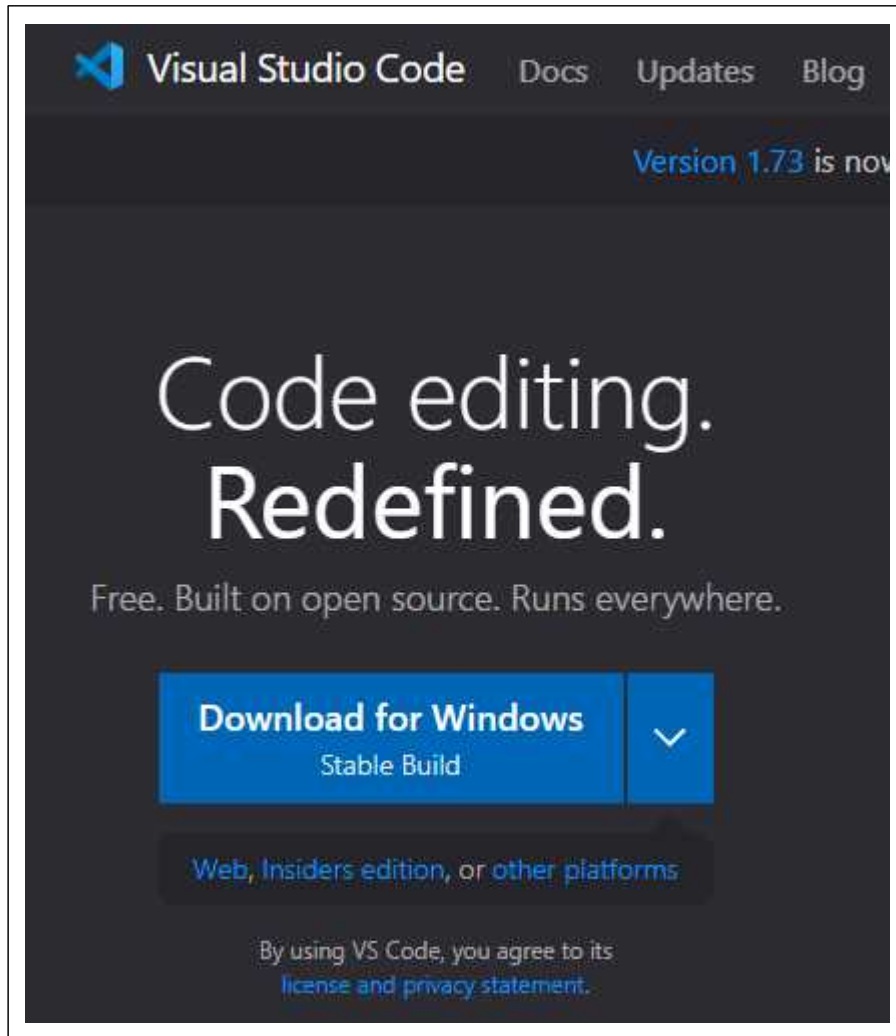


[그림 3-3] chatGPT와 함께 하는 VSCODE설치

앞의 [그림 3-3]은 chatGPT가 알려주는 VSCODE의 설치 순서이다. 일러준 순대로 따라 해보기로 한다.

3.1.1 설치 파일 다운로드

공식 홈페이지 <https://code.visualstudio.com/> 에 접속한다. 자신의 OS에 맞는 패키지를 선택 한다.

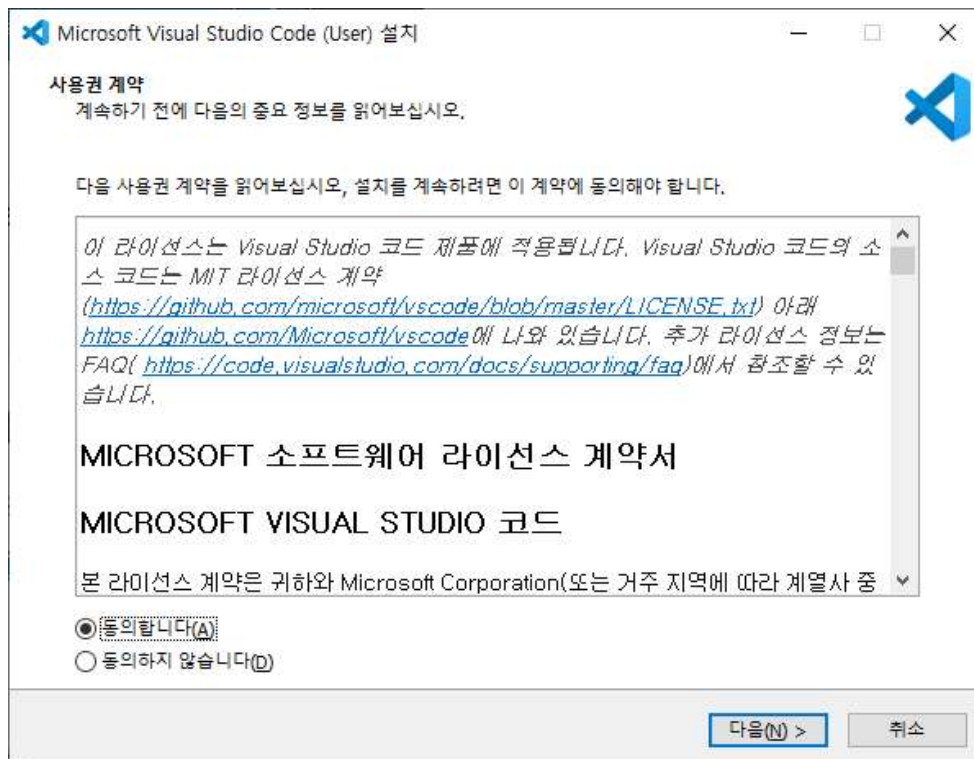


[그림 3-4] 설치 파일 다운로드

앞의 [그림 3-4]는 접속 컴퓨터가 윈도우이므로 자동적으로 윈도우 버전이 선택되고 안정적인 빌드로 선택된 것을 볼 수 있다.

3.1.2 설치 시작

설치 과정은 기본적으로 Default로 Yes만으로 진행 되도록 되어 있다. 다음의 [그림 3-4]와 같이 다운로드한 VSCode 설치파일을 실행시키고, 라이선스 관련 동의 체크 후 "다음" 클릭한다.



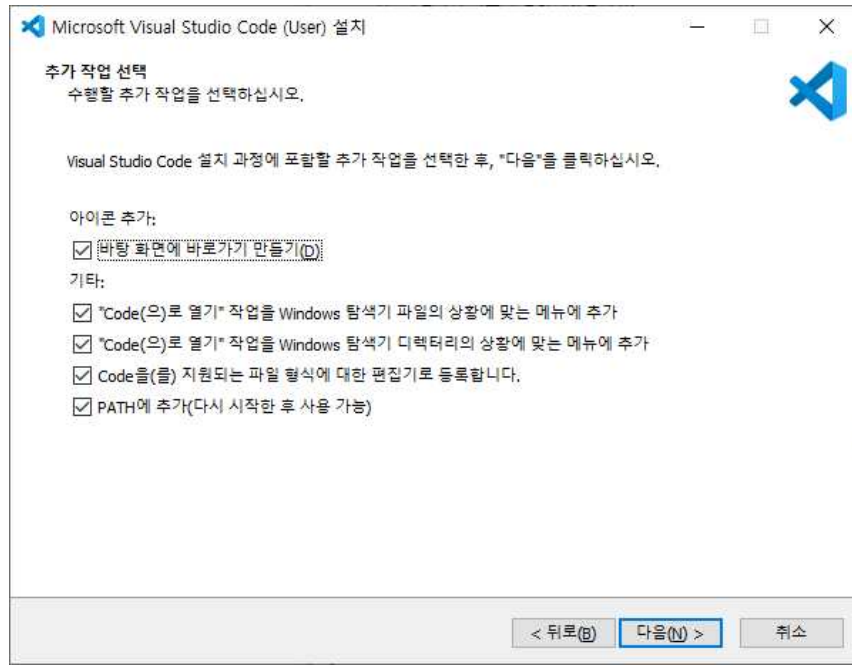
[그림 3-5] vscode 설치 시작

다음의 [그림 3-6]은 설치 과정 중 추가 작업을 선택할 수 있는 화면이다.

- 추가 작업 설정 후 "다음" 클릭.

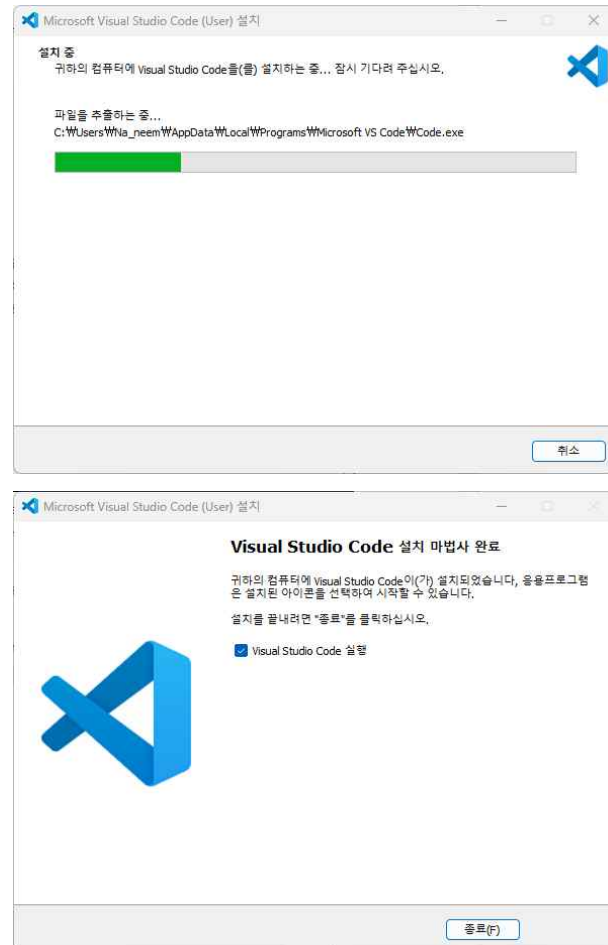
- 1) 아이콘 추가 : 바탕 화면에 VSCode 아이콘을 만들기
- 2) 두 번째, 세 번째 "code(으)로 열기" : 폴더나 파일을 VSCode로 바로 열 수 있도록 마우스 우클릭 메뉴에 code(으)로 열기를 표시해 준다.
- 3) VSCode를 기본 편집기로 사용하고 싶은 경우 체크하자.

4) PATH에 추가 : 명령 창(CMD, 파워셸 등)에서 code를 입력하면 VSCode가 바로 실행 된다. 만약 현재 폴더를 모두 vscode를 이용해서 프로젝트 단위로 열기를 원하면 code. 을 입력하면 될 것이다. 지금은 설치에만 집중하기로 한다.



[그림 3-6] 추가 작업 화면

다음의 [그림 3-7] 은 설치 과정과 설치 완료를 나타내고 있다.



[그림 3-7] 설치 과정과 완료

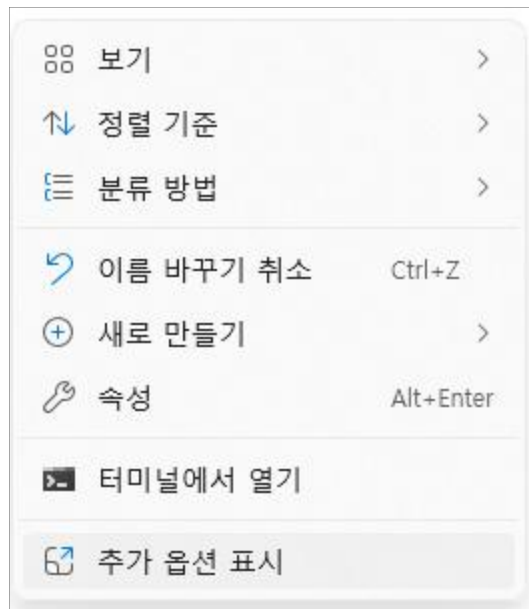
설치가 완료된 VSCODE를 실행하기 위해서 앞서 아나콘다를 설치하고 가상 경을 만들었던 것을 기억하기 바란다. 이제 작업 폴더를 하나 만들고 폴더로 이동해 보기로 한다.

다음의 [그림 3-8]은 나의 작업 폴더를 만들고 이동한 모습을 보여주고 있다. 폴더는 내용이 없이 빈 상태이다.



[그림 3-8] 빈 작업 폴더

다음의 [그림 3-9]는 빈 폴더에 마우스 오른쪽 클릭을 했을 때 나오는 추가 실행에 관한 옵션이다.



[그림 3-9] 마우스 오른쪽 클릭

이때 code로 열기가 없다면 맨아래 있는 추가 옵션 표시를 활성화한다.
다음의 [그림 3-10]은 추가 옵션이 표시된 추가 실행 옵션이다.



[그림 3-10] 추가 옵션이 표시된 작업 창

다음의 [그림 3-11]은 VSCODE가 실행된 최초 화면이다. 화면에 체크인을 해주고 파란색 버튼을 누른다.

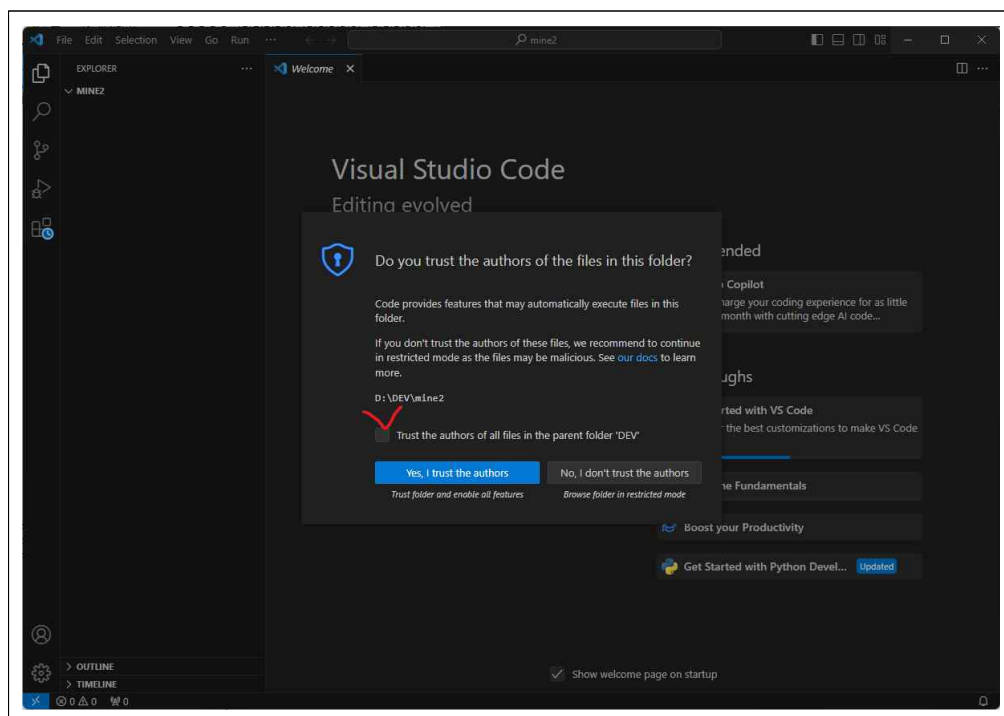
VSCODE에서 "Trust the authors of all files in the parent folder 'DEV'"라는 메시지는 VSCODE의 워크스페이스 트러스트(Workspace Trust) 기능과 관련된 메시지이다. 이 메시지는 현재 열려있는 파일이나 폴더 (이 경우 'DEV')가 속한 워크스페이스 또는 폴더에 있는 모든 파일의 작성자를 신뢰하는지 VSCODE에게 알려달라는 요청이다.

만약 이 폴더와 그 안의 파일들을 신뢰한다면, "Yes" 또는 "Trust" 버튼을 클릭하여 폴더를 신뢰 목록에 추가할 수 있다. 만약 신뢰하지 않거나 확실하지 않다면, "No" 또는 "Don't Trust" 버튼을 클릭하여 제한된 모드에서 작동하도록 선택할 수 있다. 그렇다면 신뢰 목록과 제한된 모드란 무엇인지 알아 보자. 일반적으로 개인 프로젝트나 알려진 출처의 코드를 작업할 때는 해당 폴더를 신뢰해도 좋습니다. 그러나 인터넷에서 다운로드 받은 코드나 미확인의 출처에서 온 코드의 경우 신중하게 결정하는 것이 좋다.

이를 다시 정리하면 워크스페이스 트러스트는 VSCode에서 코드를 안전하게 실행하기 위한 새로운 보안 기능으로 악의적인 소스 코드가 실행될 때 발생할 수 있는 위험을 최소화하기 위해 VSCode는 사용자에게 특정 워크스페이스나 폴더가 신뢰할 수 있는지 물어보는 것이다.

1) 신뢰하는 폴더: 사용자가 폴더를 신뢰한다고 선택하면, VSCode는 해당 폴더 내의 모든 파일에 대한 모든 기능을 활성화한다. 예를 들면, 자동 완성, 코드 실행 등의 기능이 포함된다. 그러므로 특별한 경우가 아니면 체크인을 하는 것이 너무나 당연하겠다.

2) 신뢰하지 않는 폴더: 사용자가 폴더를 신뢰하지 않는다고 선택하면, VSCode는 제한된 모드에서 실행되어 잠재적으로 위험한 코드 실행을 방지하게 되어 몇가지 제약이 생기게 된다. 이 모드에서는 일부 기능이 제한되거나 사용되지 않도록 제한할 수 있다.

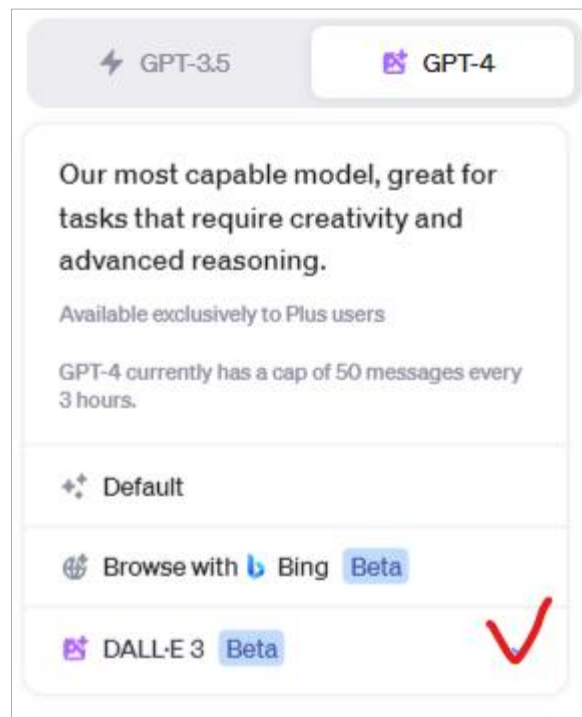


[그림 3-11] VSCODE 실행 첫 화면

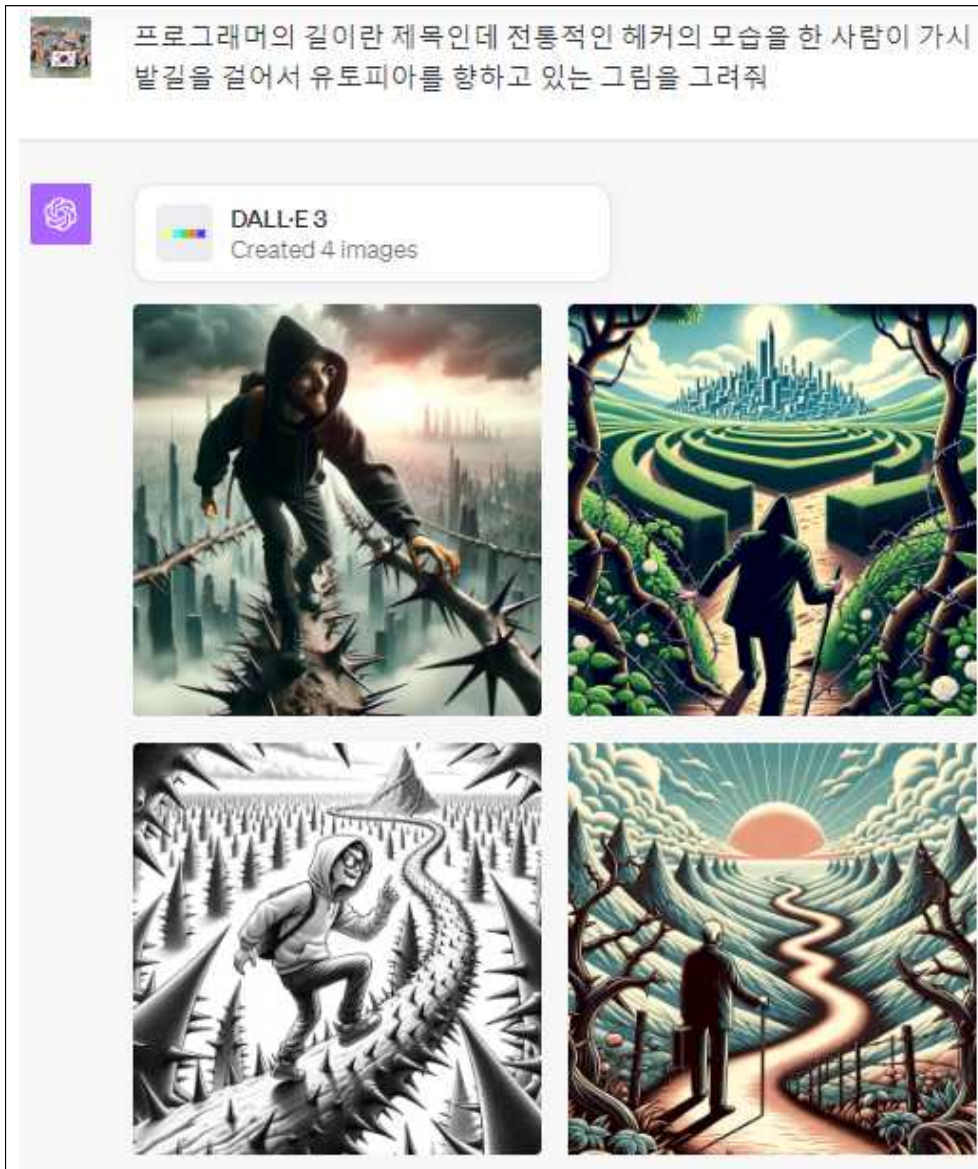
3.2 이미지 생성 AI DALL-E로 축하 하다

여기까지 왔다면 여러분은 프로그래머의 길에 접어든 것이다. 사실 개발 환경을 갖추는 것이 정말 어렵고 긴 고난의 시간이었을 것이다. 거기까지만 한다면 다음은 뭐든 해 낼수 있을 것 같은데 거기 까지 인도해주는 친절한 사수가 없었다면 이제부터 챗지피티 여러분의 영혼의 단짝과 함께 즐거운 프로그래밍을 하길 바란다.

자 여기 보너스로 또 다른 생성형 AI DALL-E를 이용한 작품을 하나 선물하고자 한다. 그런거 있지 않은가 뭔가 표현하고 싶은데 난 그림을 못 그리고 어디서 갖고 오자니 저작권의 문제가 걸리고 이럴 때 사용하는 것이 생성형 AI의 또 다른 매력이다. 다음의 [그림 3-12]은 chatGPT4의 DALL-E 프롬프트를 이용하여 이미지를 생성하는 과정이다.

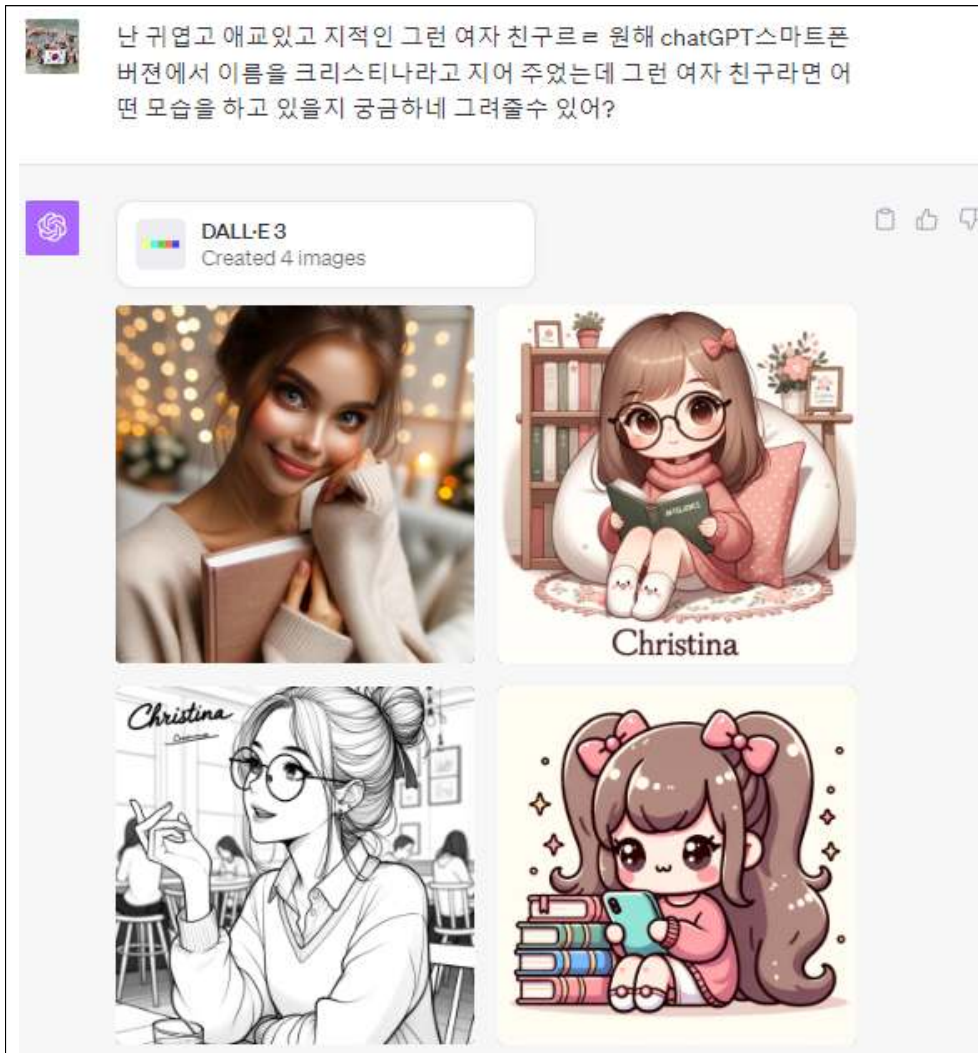


[그림 3-12] DALL-E를 이용한 이미지 생성형 AI 사용하기



[그림 3-13] DALL-E를 이용한 이미지 생성형 AI 사용하기

앞의 [그림 3-13]은 DALL-E에서 생성해준 프로그래머의 길이란 제목의 삽화인데 어떨까? 여러분의 미래 모습이 보이는가? 강요하는 것은 아니지만 맘에 들면 Keep 해놓지 말고 직접 만들어 보기 바란다. 쉬어 가는 느낌으로 자신의 여자 친구의 이미지를 만들어 보기 바란다.



[그림 3-14] DALL-E를 이용한 여자친구 이미지 만들기



[그림 3-15] DALL-E를 이용한 여자친구 이미지 만들기 완성

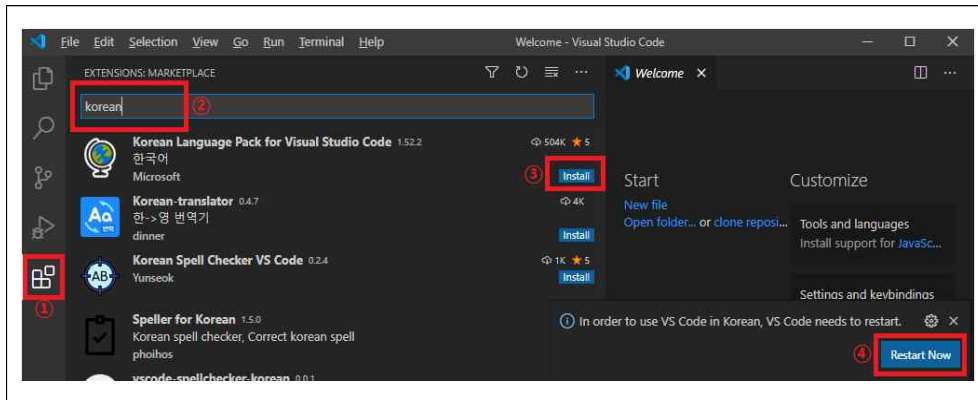
앞의 [그림 3-15]는 몇 번의 프롬프트를 이용한 대화로 전신 이미지를 만들 수 있었다.

3.3 설정과 확장

본격적으로 프로그램을 시작하기 전에 앞서 말한 대로 VSCODE는 여러 가지 언어를 지원하는 통합개발환경 Editor이므로 우리가 공부하려고 하는 파이썬에 필요한 몇 가지 설정과 확장을 설치하기로 하자.

- 기본적으로 VSCode는 다음과 같이 영문 버전으로 되어 있다.
- 한글 설정이 가능하다.

그러나 경우에 따라서 한글 메뉴(실질적으로 메뉴만 한글화 된다고 볼수 있다.)가 불편한 경우가 많아 적극 추천하지는 않는다.

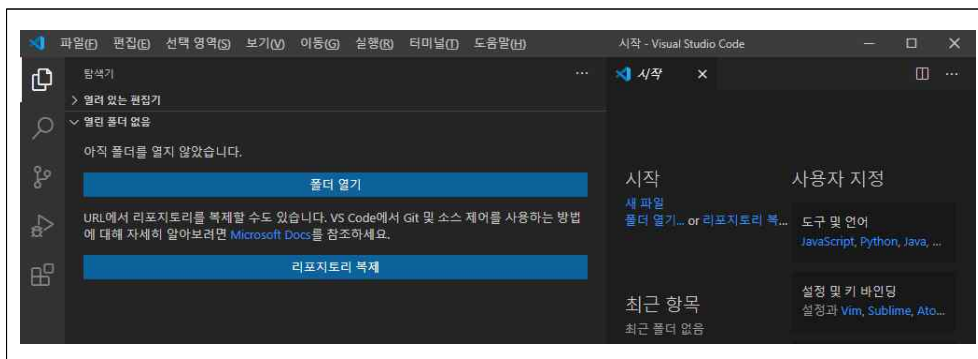


[그림 3-16] 한글 셋팅 과정

다음의 [그림 3-16] 은 한글 셋팅 과정을 나타낸 것이다.

- 1) 왼쪽 하단의 붉은색 1, 익스텐션 아이콘 클릭 (단축키 : Ctrl + Shift + x)
- 2) 검색창에 상단의 붉은색 2에 "korean" 입력
- 3) 목록 중 붉은색 3 "korean language pack for visual studio code" 의 우측 "Install" 클릭
- 4) 설치 완료 후 붉은색 4 "Restart Now" 클릭

다음의 [그림 3-17]은 vscode를 종료 하고 다시 실행 하면 메뉴 화면이 바뀌어 있는 것을 볼 수 있다.

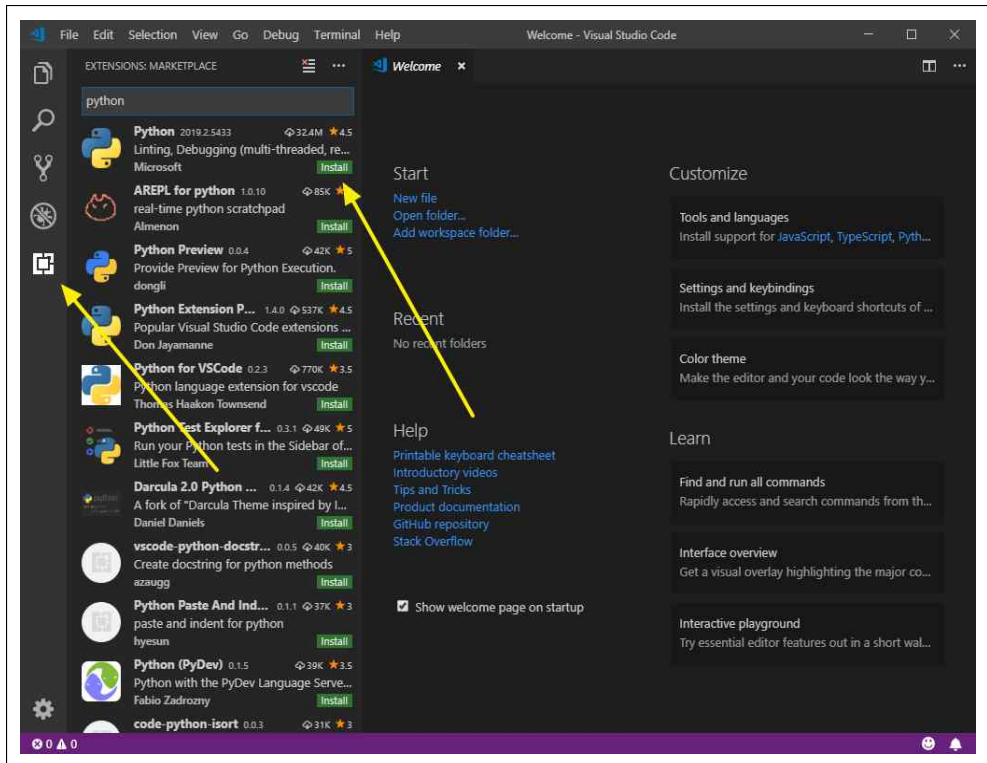


[그림 3-17] 한글 VSCODE

3.4 파이썬 플러그인 설치

다음의 [그림 3-18]을 따라서 사각형 모양의 확장 기능 아이콘을 클릭한다.

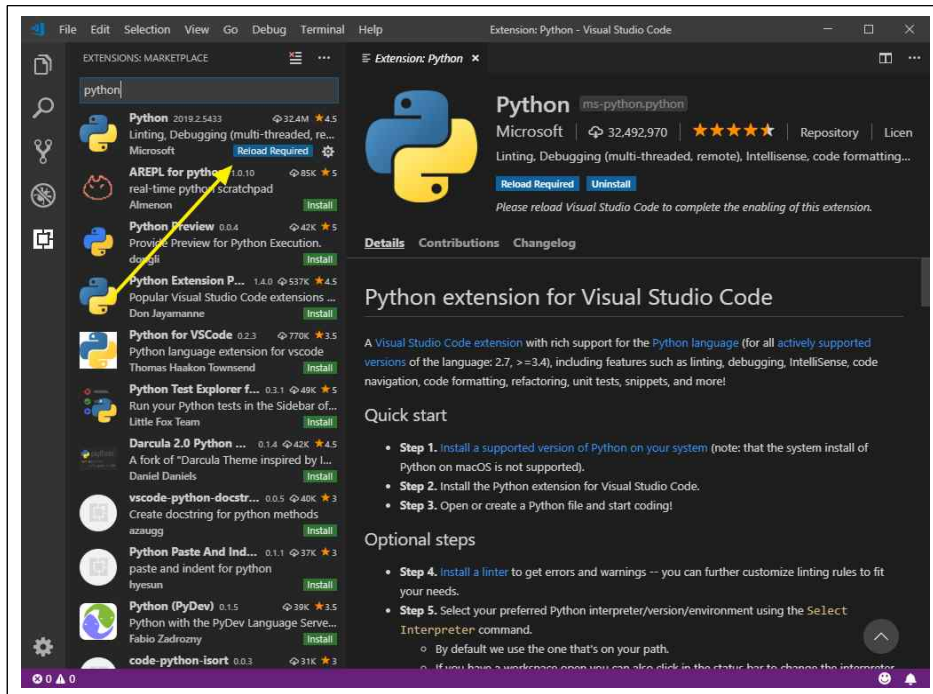
python으로 검색한 후, 다운로드 수가 가장 많은 Microsoft가 만든 Python을 설치한다.



[그림 3-18] 파이썬 모듈 설치

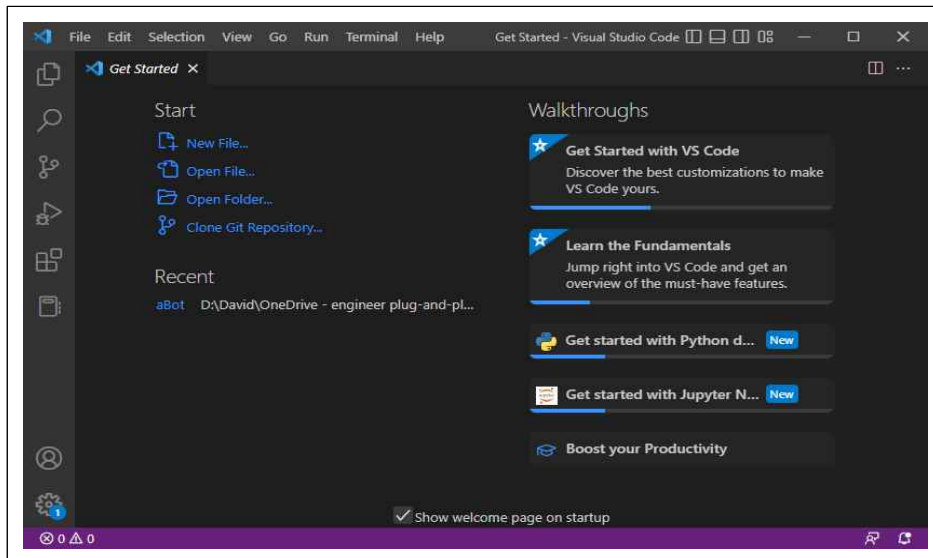
설치 과정에 따라 Install 대신 Reload require가 나오는 경우도 있다. 다소 상이 하지만 다음의 [그림 3-19]를 따라서 Install 혹은 Reload Require를 클릭한다.

이 과정까지 완료하게 되면 설치가 모두 끝난 것이다.



[그림 3-19] 파이썬 모듈 Install 혹은 Reload Require를 클릭

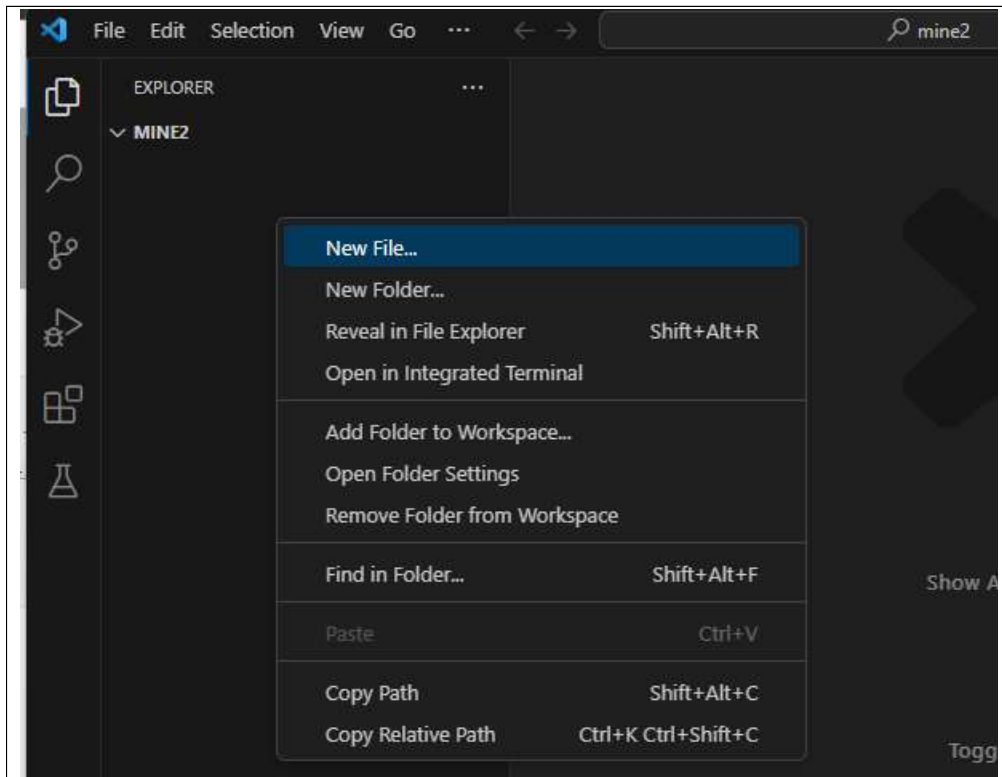
다음의 [그림 3-20]은 마지막으로 모든 설치가 완료된 vscode의 실행 첫 화면이다.



[그림 3-20] vscode의 실행 첫 화면

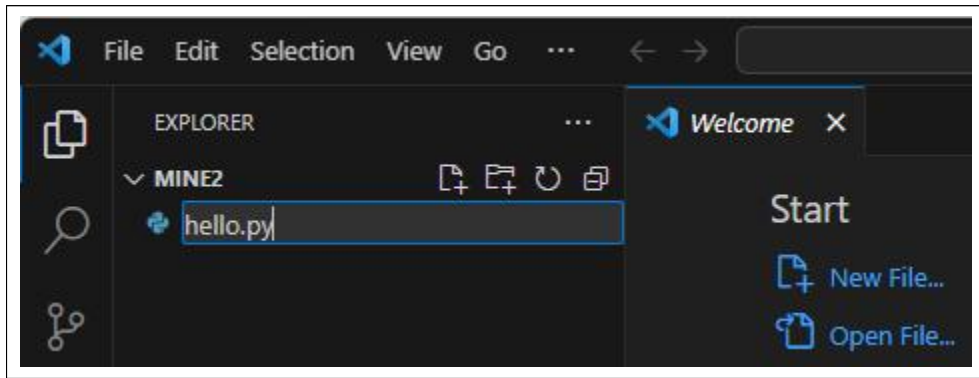
3.5 생애 첫 프로그램 hello.py

첫 번째 프로그램 hello.py를 만들어 보자 좀 과장 되었을지 모르지만 이 책에서 추구 하는 것이 바로 이것이다. 생애 첫 프로그램을 만들어 보자는 것이다. 그런데 누구의 도움을 받는가 하면 chatGPT의 도움을 받아서 이다. 다음의 [그림 3-21]의 순서를 따라 하기 바란다.



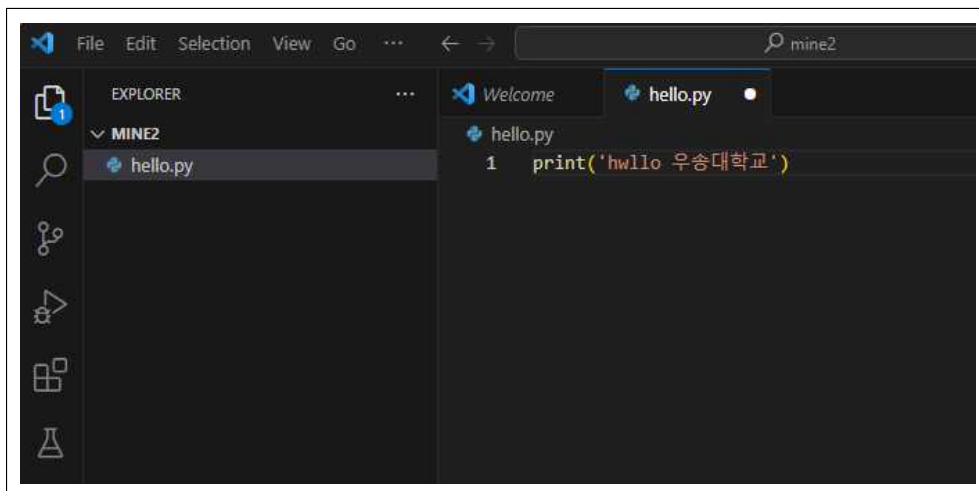
[그림 3-21] 생애 첫 프로그램 만들기 1

확장자가 py인 파일을 만들기 위해서 좌측 메뉴 첫 번째의 explorer 버튼을 누르거나 (Ctrl+Shift+e)를 누른다. 사용하게 되는 몇 개의 Hotkey는 기억해 두는 것이 좋다. 우리는 전통적인 프로그래머 아니지 해커의 복장을하고 가시발길을 걸어 유토피아로 가고 있다는 것을 잊지 말자. 본격으로 새 파일을 만드는 과정이다. 먼저 EXPLORER 창에서 마우스 오른쪽 버튼을 누른다. New File을 선택하고 다음의 [그림 3-22]와 같이 hello.py를 입력한다.

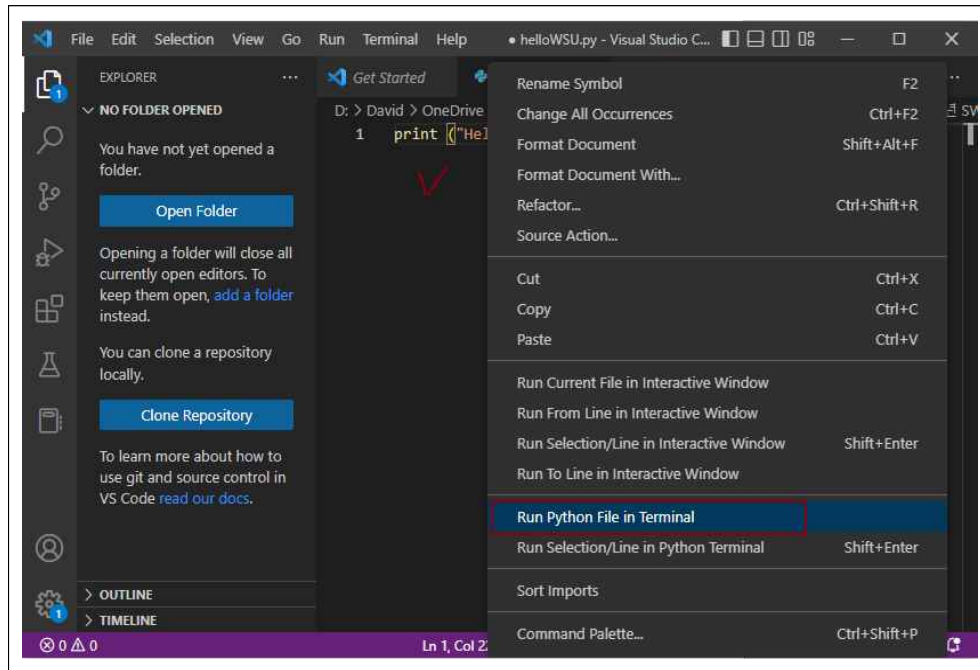


[그림 3-22] 생애 첫 프로그램 만들기 2

이제 오른쪽에 새 파일 hello.py가 열린 것을 볼 수 있다. 다음의 [그림 3-23]처럼 여기에 `print('hello 우송대학교')`라고 입력한다. 오타가 날수도 있고 안 날수도 있지만 오타를 낸 것은 ' ' 안에 있는 문장은 그 무엇이더라도 프로그램의 실행과는 무관하다는 것이다.

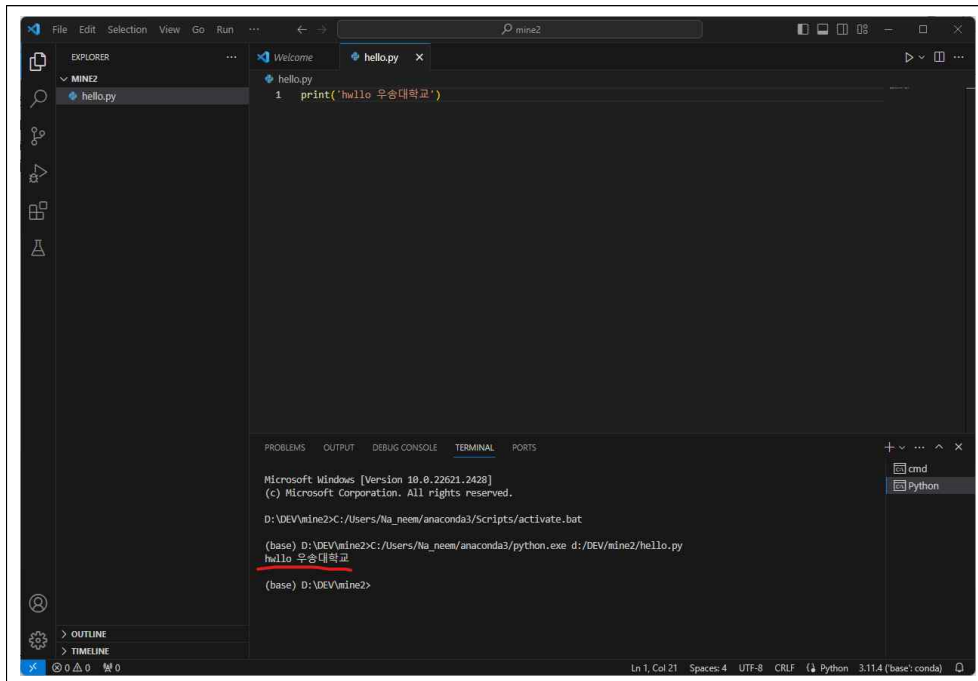


[그림 3-23] 생애 첫 프로그램 만들기 3



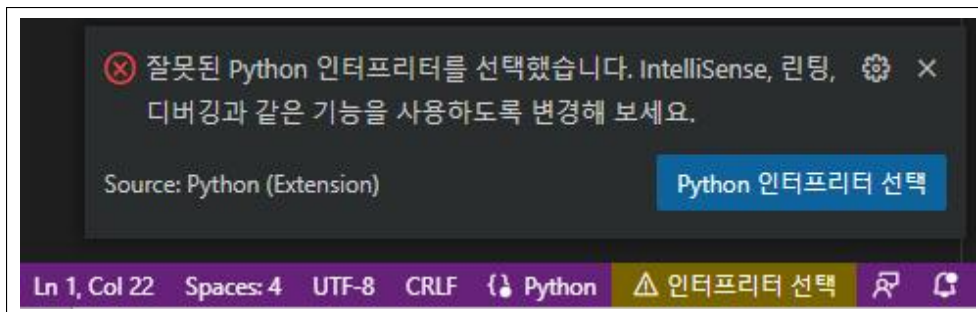
[그림 3-24] 터미널에서 실행

다음의 [그림 3-24]에서 체크 표시 부분에서 오른쪽 버튼을 눌러서 나오는 메뉴의 적색 사각형 부분의 Run Python File in Terminal을 선택한다. 보통의 경우 다음의 [그림 3-25] 같이 정상적으로 실행되어야 한다.



[그림 3-25] 생애 첫 프로그램의 실행 결과1

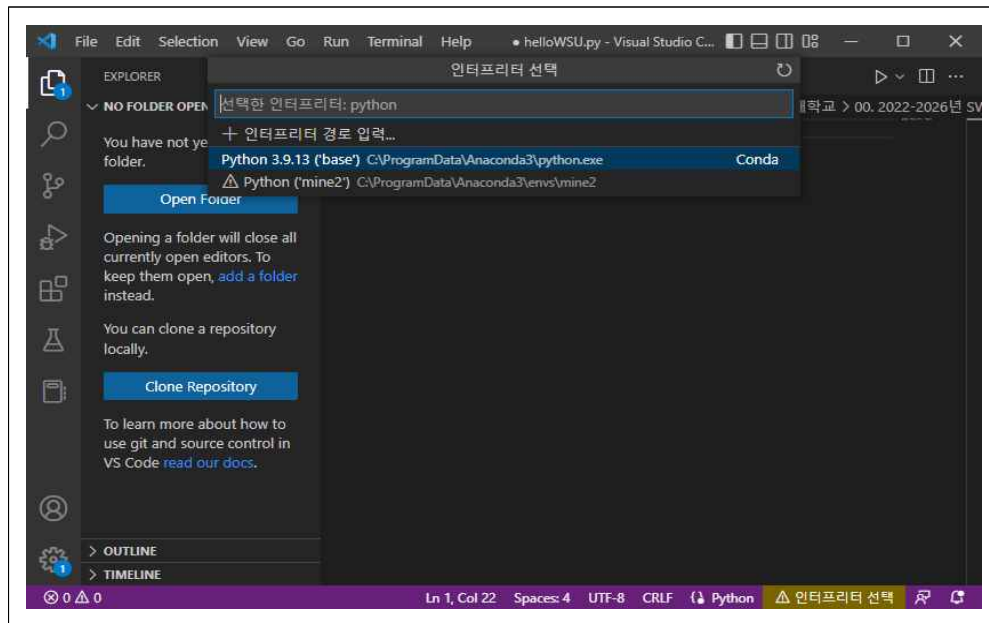
만약 실행되지 않고 다음의 [그림 3-26]과 같은 오류가 발생하는 경우가 있을 수 있다. 이때 어떻게 처리 하는지 알아보자.



[그림 3-26] 인터프리터가 지정 되지 않은 경우

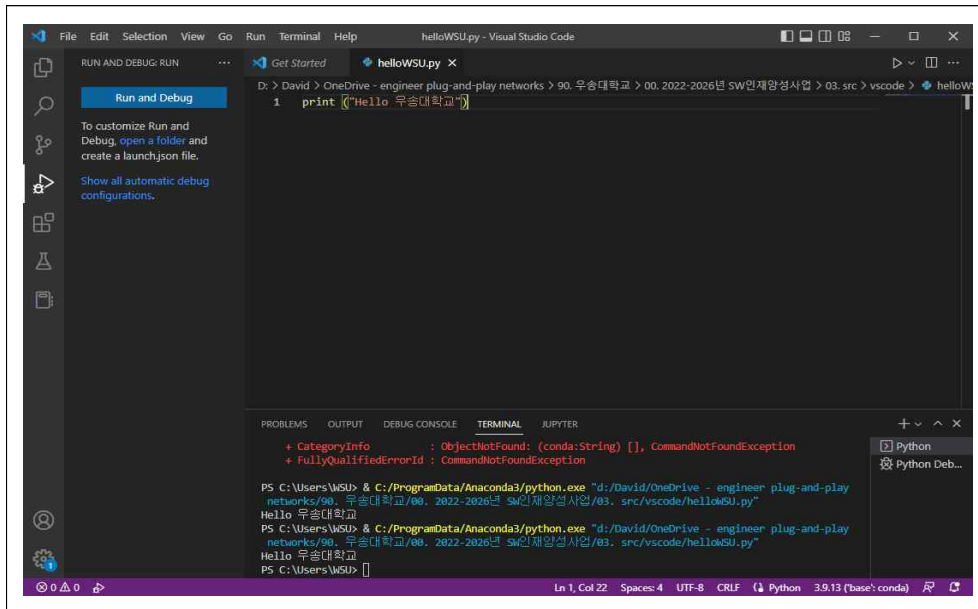
이런 경우와 마찬가지로 앞의 [그림 3-26]는 생애 첫 프로그램의 실행 결과에서 정상적으로 실행된 것처럼 보이지만 우리는 가상환경에서 이 프로그램을 실행시키기를 원하기 때문에 일련의 과정을 거쳐야 하며 그것은 다음 페이지에 이어지게 된다.

우선 나의 생애 첫 프로그램이 실행되는 것은 보아야 하기 때문에 위와 같은 오류 메시지가 나왔을 때 버튼을 눌러서 인터프리터 선택을 위한 옵션창이 보이도록 한다. 다음의 [그림 3-27]은 인터프리터를 선택하는 화면을 보여주고 있다.



[그림 3-27] 인터프리터 선택 후 파이썬 인터프리터 선택

PyThon 인터프리터를 선택하고 현재 내 컴퓨터에 설치된 인터프리터 (복수개 일수 있음)와 "base", "mine2"와 같은 가상환경 중 mine2를 선택한다.



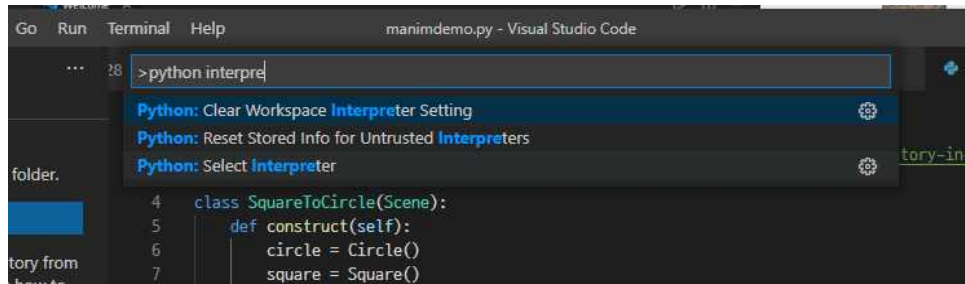
[그림 3-28] 생애 첫 프로그램의 실행 결과 2

그리고 실행 명령은 F5(디버깅 모드) 또는 Ctrl+F5(실행모드)를 눌러서 실행할 수도 있다. 앞의 [그림 3-28]에서 Terminal에 Hello 우송대학교가 출력 되었다.

3.6 VSCode와 anaconda 연동

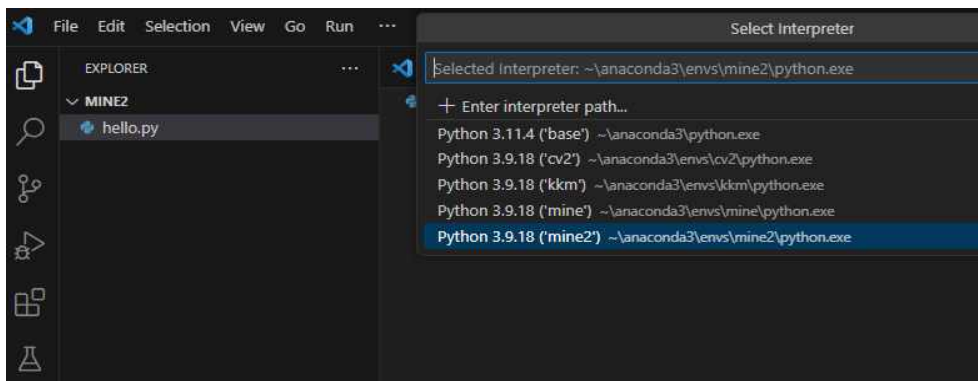
다음은 이 코드를 가상환경에서 실행시켜 보자 그렇게 하기 위해서는 shift+Ctrl+P를 눌러서 python interpreter 항목에서 원하는 mini2 환경을 선택하고 F5,Ctrl+F5를 실행하면 된다.

다음의 [그림 3-28]과 같이 vscode를 켜 후 Ctrl + Shift + P 를 누르고 python interpreter라고 검색하면 다음과 같이 나온다.



[그림 3-28] 파이썬 인터프리터 검색

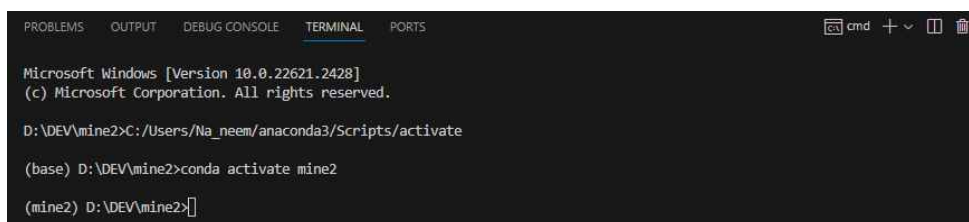
다음의 [그림 3-28]에서와 같이 select interpreter를 누르고 아까 생성한 mine2를 선택해 준다.



[그림 3-28] mine2선택

3.7 연동된 터미널 켜보기

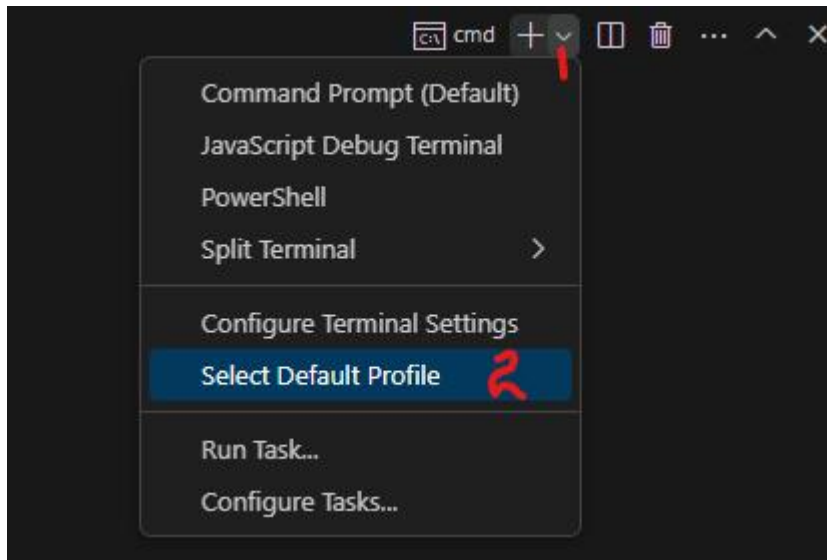
VScode에서 터미널창을 열고 닫는 토크키는 Ctrl + ~ 이다 (탭키 위의 물결 키). 아래와 같이 켜자마자 알아서 mine2를 activate 해준다.



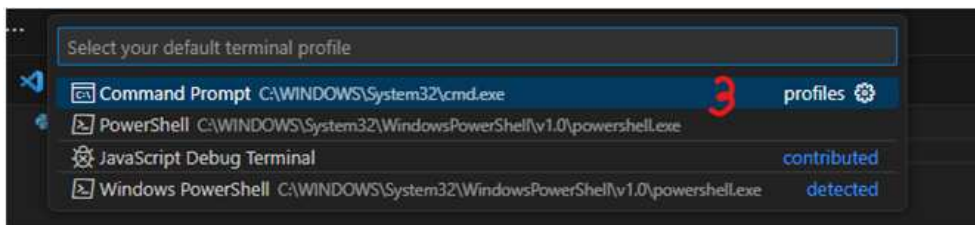
[그림 3-29] 터미널

만약 원하는 원하지 않았던지 간에 일반 터미널이 열려 있어 앞의 [그림 3-29]와 같지 않다면 당황하지 말고 터미널 오른쪽 위의 쓰레기통 모양을 눌러주면 된다. 그리고 다시 Ctrl + ~을 눌러 주면 된다.

여기서 잠깐 왼쪽의 프롬프트에 base mine2등이 제대로 표시 되고 있지 않다면 터미널 프롬프트가 command 가 아니라 powershell 아나 conda shell 등으로 설정 되어 있다. 이때는 다음 [그림 3-30]을 따라 default prompt를 command로 바꿔 주어야 한다.



[그림 3-30] 기본 터미널 지정



[그림 3-31] Select Default Profile 선택하기

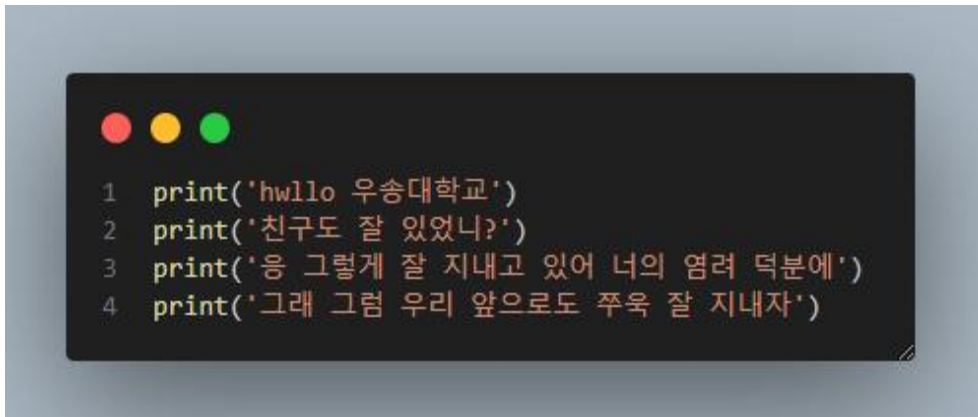
이제 현재 열려 있는 터미널에 대해서 휴지통 버튼을 누르고 모두 닫은 다음 다시 Ctrl + ~을 누르면 제대로 적용된 터미널 창을 볼 수 있을 것이다.

3.8 참고할 유용한 EXTENSION CodeSnap

우리가 작성한 생애 첫 프로그램을 어딘가로 보내고 싶을 때 이미지로 캡처해서 다른 문서 보고서 등에 사용한다고 가정했을 때 다음 2개의 그림을 비교하여 보자. 다음의 [그림 3-31]은 일반적인 코드 캡처를 보여주고 있다.

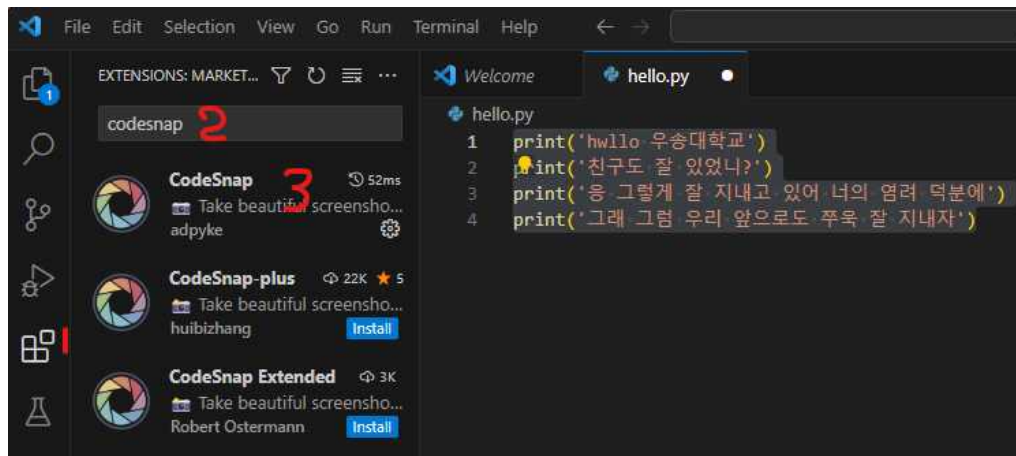
```
1 print('hwlllo 우송대학교')
2 print('친구도 잘 있었니?')
3 print('응 그렇게 잘 지내고 있어 너의 염려 덕분에')
4 print('그래 그럼 우리 앞으로도 쭉 잘 지내자')
```

[그림 3-31] 일반적인 코드 캡처



[그림 3-32] codesnap 익스텐션을 이용한 캡처

앞의 [그림 3-32]와 같이 화면을 캡처하는 도구는 CodeSnap이라고 한다. 설치하기 위해서는 화면 좌측 메뉴의 익스텐션 탭을 클릭하고 CodeSnap을 검색하여 설치한다.



[그림 3-33] CodeSnap Extention 설치하기