

How to use Python

CSED101 프로그래밍과 문제해결

Student ID: 20240505

Name: 공현성

POVIS ID: hyunseong

Phone Number: 010-9649-3020

3월 수업 일정

03/08 (토) (TODAY)

03/15 (토)

03/22 (토)

03/29 (토)

Content

1. 파이썬 설치하기
2. IDE 사용법
3. 파이썬 사용해보기

03/08	0T, 파이썬 설치, 개념 복습 (today)
03/15	수업
03/22	수업
03/29	수업
04/05	중간고사 대비
04/12	수업
04/19	수업
04/26	수업
05/03	수업
05/10	수업
05/17	수업
05/24	기말고사 대비

파이썬이란?

웹앱, 데이터 과학, 기계 학습에 널리 사용되는 프로그래밍 언어.

배우기 매우 쉬우며 여러 플랫폼에서 실행될 수 있다는 장점 때문에 널리 사용 중.



파이썬 설치하기

모든 언어에는 “문법”이라고 하는 것이 존재. 문법에 맞추어 코딩을 해야 됨.

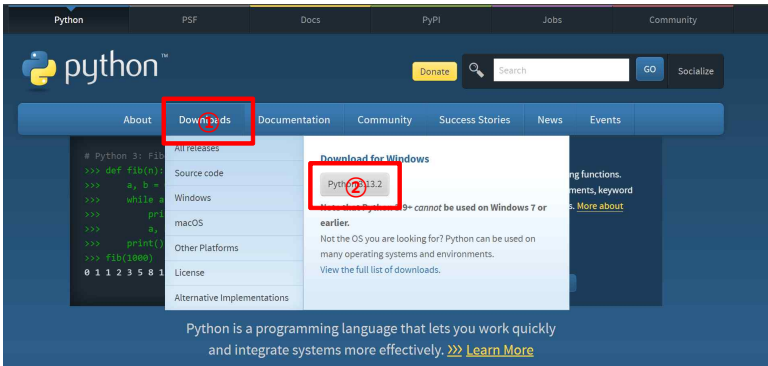
이 문법에 맞추어도, 작성된 파이썬 파일은 사실 그냥 텍스트에 불과함.
-> 컴퓨터가 이해할 수 있는 언어(바이너리 코드)로 변환을 해줘야 됨.

이러한 변환을 담당하는 프로그램을 “인터프리터”라고 부름.

보통 이 인터프리터를 설치하는 것을 “파이썬을 설치한다”고 함.

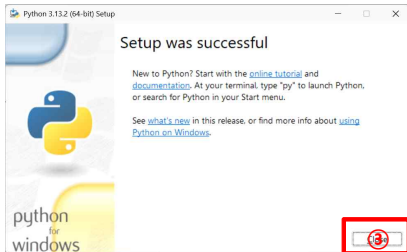
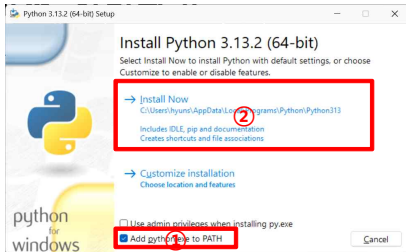
파이썬 설치하기

이러한 인터프리터는 <https://www.python.org/>에서 제공하고 있음.



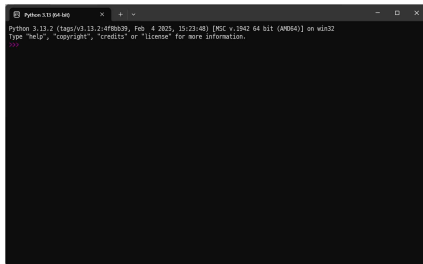
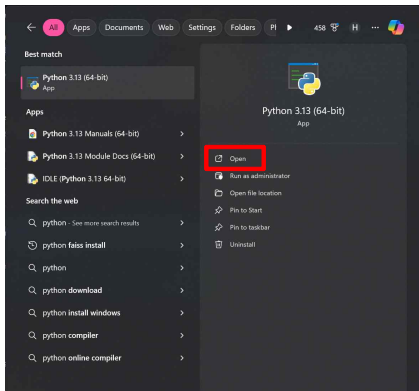
파이썬 설치하기

다운받은 파일 **python-3.13.2-amd64.exe** 관리자 권한으로 실행하기



파이썬 설치하기

설치한 인터프리터는 윈도우 검색에 python을 검색하여 실행해볼 수 있다.



파이썬 설치하기 - 써보기

현재 실행한 프로그램(Python 3.13)은 명령어를 1줄씩 입력이 가능한 창이다.

(실습1) 아래의 코드를 입력해보자.

입력:

```
print("Hello, World!")
```

출력:

```
Hello, World!
```

```
Python 3.13.2 (tags/v3.13.2:4f8bb3  
Type "help", "copyright", "credits  
>>> print("Hello, World!")  
Hello, World!  
>>> |
```


IDE란?

실제 코드는 몇십~몇백줄을 적게 된다.

그런데, 기본적으로 지원하는 창에서는 여러 한계점(한 줄 입력하면 더 이상 수정이 불가능)이 보인다.

그렇기 때문에, 사람들은 IDE(통합 개발 환경)이라고 하는 것을 만들었다.

정말 다양한 IDE가 존재하지만, 우리는 **Visual Code**라고 하는 것을 사용한다.

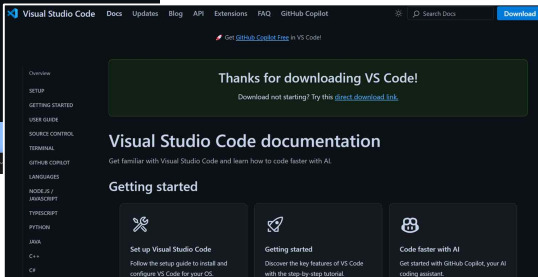
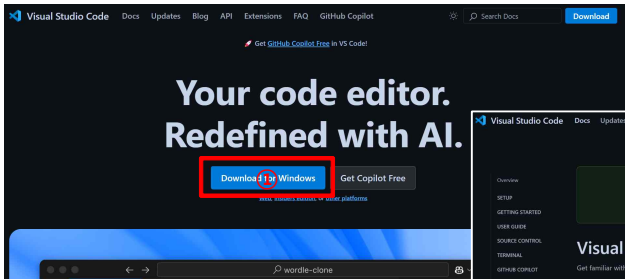


주의: IDE 초기 설정은 복잡할 수 있다.

다행히 한 번 설정하면, 컴퓨터를 초기화하기 전까지 다시 설정할 일은 없다.

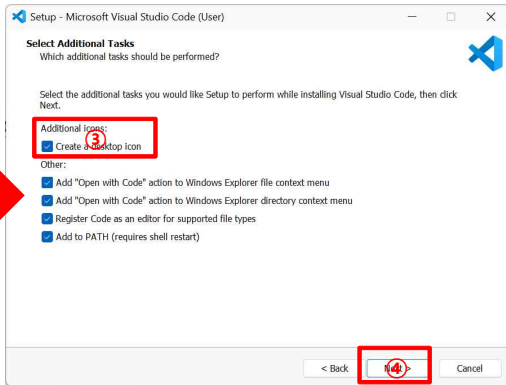
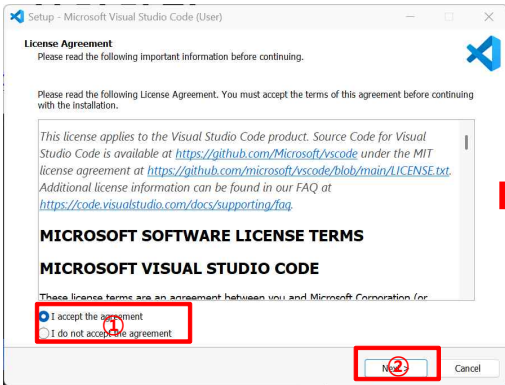
VSCode 설치하기

VSCode는 <https://code.visualstudio.com/>에서 제공하고 있음.



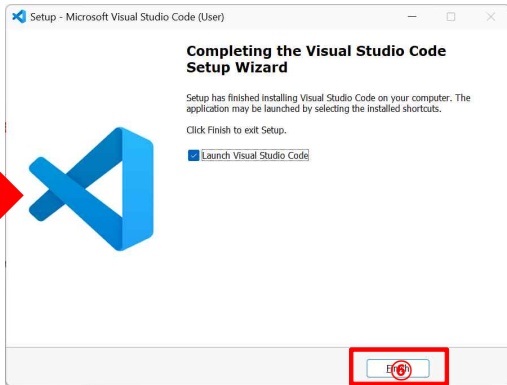
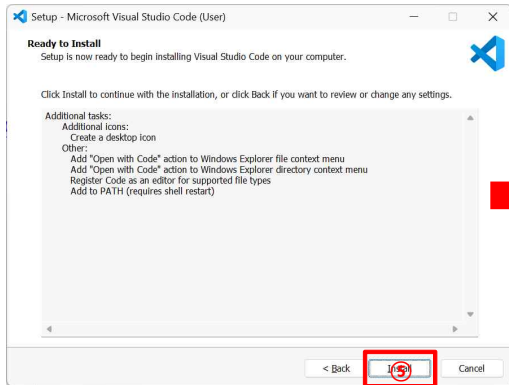
VSCode 설치하기

다운받은 파일 **VSCodeUserSetup-x64-1.97.2.exe** 관리자 권한으로 실행하기



VSCode 설치하기

다운받은 파일 **VSCodeUserSetup-x64-1.97.2.exe** 관리자 권한으로 실행하기

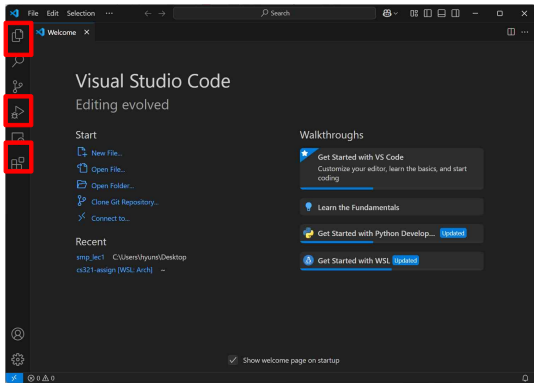


VSCode 사용하기 - 기본 화면

기본적으로 실행하면, 아래와 같은 화면이 나온다.

화면의 왼쪽의 바에는 버튼이 있다.
눌러보면서 기능을 익히자.

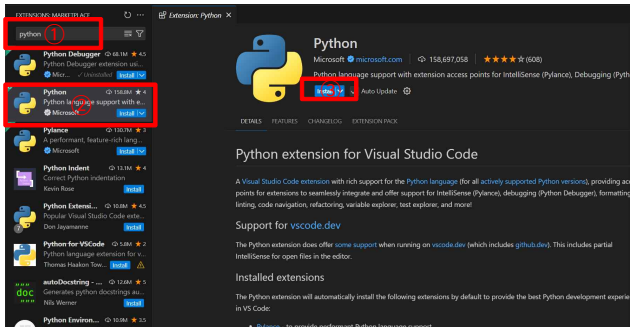
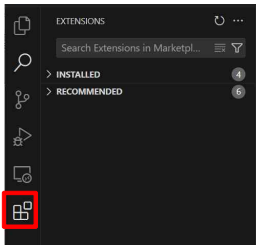
1. 폴더 관리
2. 코드 실행 & 디버깅
3. 확장 기능 설치



VSCode 사용하기 - 확장 기능 설치

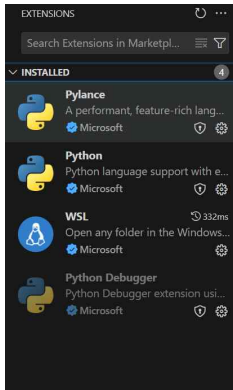
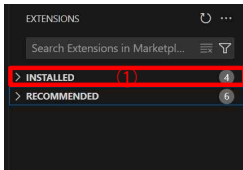
지금은 **확장 기능 설치**부터 사용한다.

검색창에 “python”을 입력하고, 이걸 설치하자.



VSCode 사용하기 - 확장 기능 설치

설치를 완료한 후, 검색창의 검색어를 지워서 빈칸으로 만들면, 설치한 확장 기능들을 확인할 수 있다.

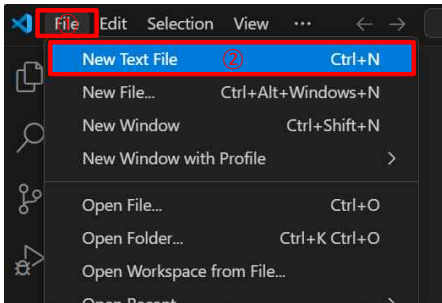


INSTALLED를 클릭하면,
이렇게 창이 펼쳐져서 설치된
기능들을 확인해볼 수 있다.

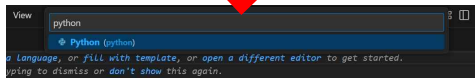
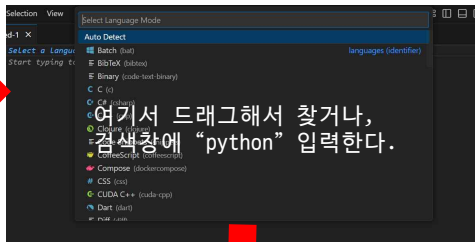
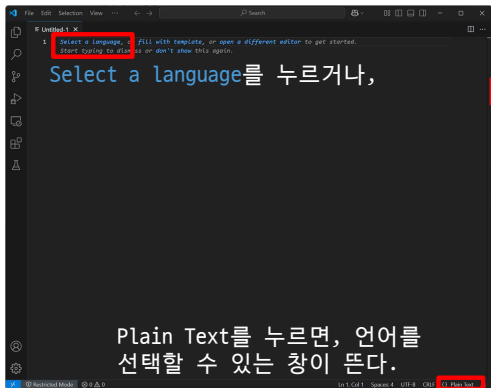
Python 확장 기능을 하나만
설치해도 이렇게 3개가
동시에 설치가 된다.

VSCode 사용하기 - 파일 만들기

이제, 준비가 끝났다. 새로운 파이썬 파일을 만들어서 테스트해보자.



VSCode 사용하기 - 파일 만들기

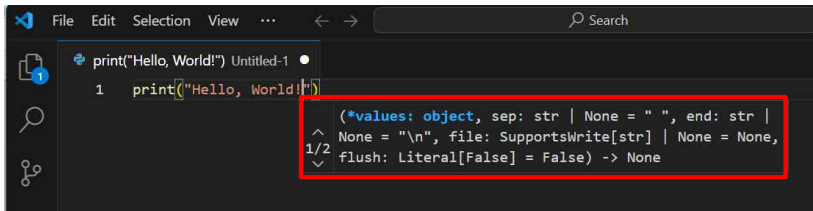


VSCode 사용하기 - 명령어 쓰고 실행하기

아까 쓴 명령어를 똑같이 한번 써보자.

커서의 위치로 이상한 팝업창이 따라다닌다. 아까 확장 기능의 역할이 바로 이거다.

쉽게 설명하자면, 코딩을 위한 하나의 가이드로 지금 팝업창은 print를 사용하는 방법을 설명하고 있다. 추후 더 자세히 설명할 예정.

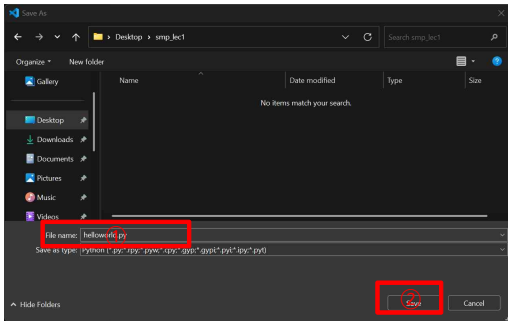


VSCode 사용하기 - 명령어 쓰고 실행하기

코드를 실행하려면 우선 파일을 저장해야 한다.

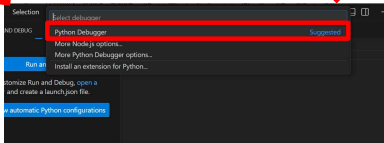
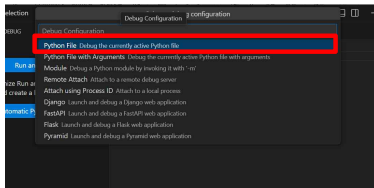
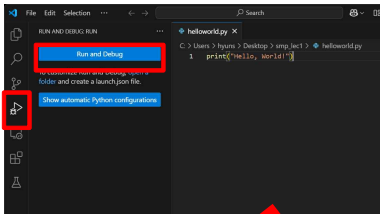
ctrl+s를 누르면 파일을 저장할 수 있는 창이 뜬다. 파일의 이름을 정하고 파일을 저장하자.

파이썬의 기본 확장자는 **.py**이다.



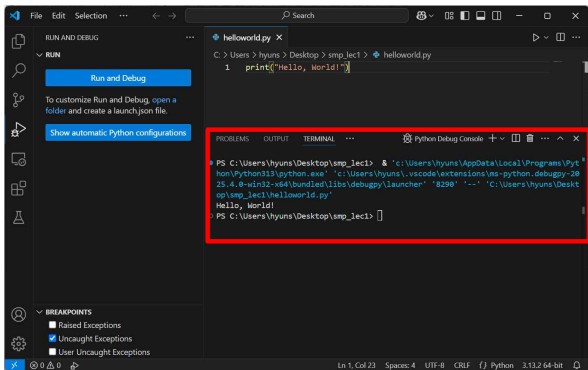
VSCode 사용하기 - 명령어 쓰고 실행하기

저장된 파일을 실행하려면, 실행 창으로 이동한다.



VSCode 사용하기 - 명령어 쓰고 실행하기

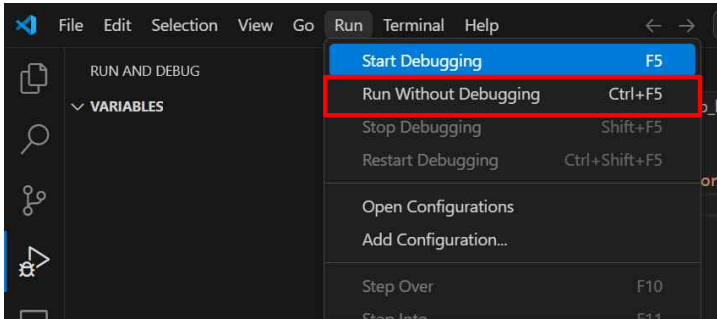
코드가 실행되면, 아래와 같이 **TERMINAL**이라고 하는 창이 새로 생긴다. 이 창에서 결과를 확인해볼 수 있다.



VSCode 사용하기 - 명령어 쓰고 실행하기

실행은 **CTRL+F5** 단축키를 누르거나,

Run -> Run Without Debugging을 이용해서도 실행해볼 수 있다.



VSCode 사용하기 - 프로젝트 만들기

실제 과제를 하거나 프로젝트를 진행할 때는 여러 개의 파일을 쓰고, 관리해야 할 때가 있다.

VSCode는 폴더 단위로 이러한 프로젝트를 관리할 수 있도록 한다.

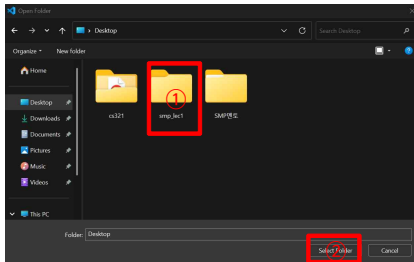
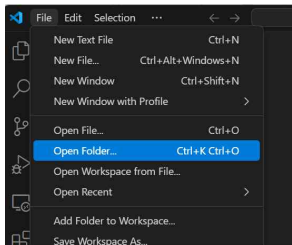
우선, 원하는 위치(예: 바탕화면)에 폴더 'smp_lec1'을 만들자.



VSCode 사용하기 - 프로젝트 만들기

VSCode에서 방금 만든 폴더를 열 것이다.

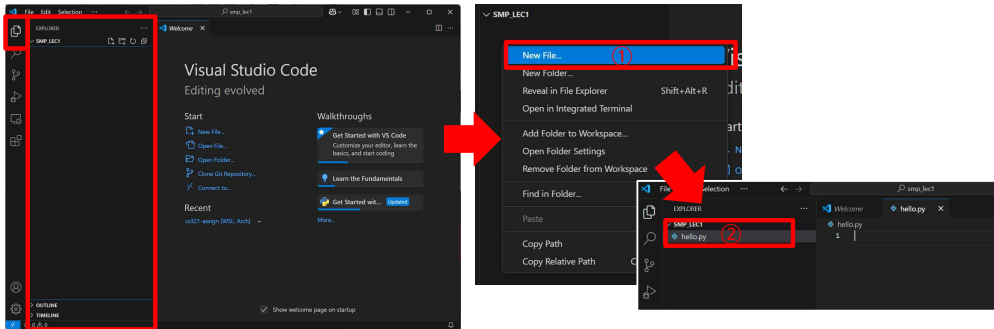
File -> Open Folder를 들어가 폴더를 찾아서 클릭하면 된다.



VSCode 사용하기 - 프로젝트 만들기

폴더 내부의 파일은 좌측의 **EXPLORER**로 확인할 수 있다.

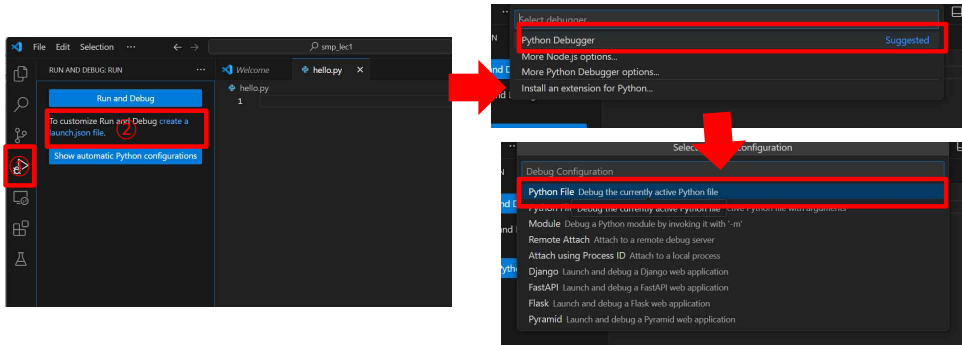
폴더를 처음 만들면 내부에 아무것도 없다. EXPLORER를 우클릭한 뒤, **New File**을 클릭하면 새로운 파일을 만들 수 있다.



VSCode 사용하기 - 프로젝트 만들기

그 후, RUN 창으로 가서 create a launch.json file을 클릭한다.

그리고 Python Debugger -> Python File을 순서대로 클릭한다.

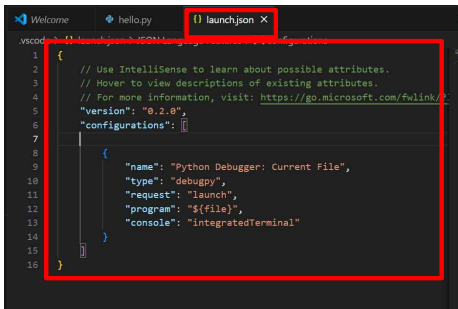


VSCode 사용하기 - 프로젝트 만들기

그러면 VSCode가 자동으로 **launch.json**이라고 하는 파일을 만들어준다.

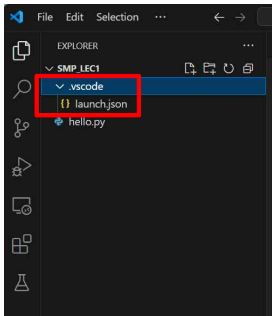
이 파일은 코드를 실행하는 설정을 담아놓은 파일로, VSCode에서 프로젝트 내의 코드를 실행할 때 이 파일을 참고하게 된다.

EXPLORER 창으로 돌아와서 확인하면 **.vscode** 폴더와 그 안의 **launch.json**가 생긴 것을 확인해볼 수 있다.



The screenshot shows the VS Code editor with the **launch.json** file open. The file content is as follows:

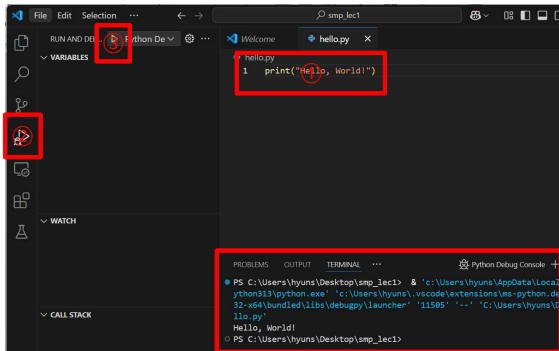
```
{
  // Use IntelliSense to learn about possible attributes.
  // Hover to view descriptions of existing attributes.
  // For more information, visit: https://go.microsoft.com/fwlink/?linkid=833080
  "version": "0.2.0",
  "configurations": [
    {
      "name": "Python Debugger: Current File",
      "type": "debugpy",
      "request": "launch",
      "program": "${file}",
      "console": "integratedTerminal"
    }
  ]
}
```



VSCode 사용하기 - 프로젝트 만들기

이제 `hello.py`에 아까의 코드를 입력한 후, **RUN** 창으로 가서 실행 버튼(초록색 삼각형)을 클릭하거나, 아까처럼 단축키(**SHIFT+F5**)를 이용하는 등의 방법을 이용하여 코드를 실행할 수 있다.

앞으로는 이렇게 폴더를 만들어 파일을 관리할 것이다.



텍스트 창에서 열려있는
파일이 실행되게 된다.

이상한 창을 열어놓으면
제대로 실행이 되지 않을
수 있으니 주의할 것.

디버깅이란?

수백줄이 넘는 코드를 작성하다보면 필연적으로 실수를 하기 마련이다.

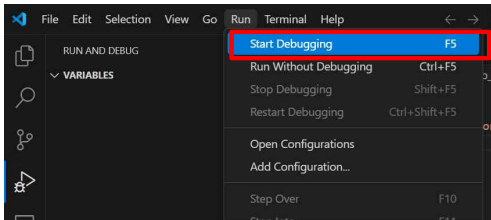
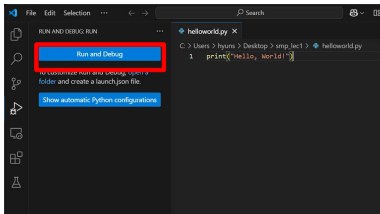
파이썬은 스크립트 언어로 인터프리터가 코드를 1줄씩 읽으면서 실행한다. 이는 코드를 빠르게 테스트해볼 수 있다는 장점이 있지만, **코드의 일부분에 문제가 발생해도** 그 코드에 도달하기 전까지는 **문제를 확인할 수 없다**는 단점이 있다.

디버깅은 개발 중에 발생할 수 있는 논리적 오류나 비정상적인 연산(버그)을 찾아내고 수정하는 과정이다.

디버깅 하는 법

사실 앞서 봤던 예제에서 **Run and Debug**를 눌러 코드를 실행했었다. 즉, 디버깅 기능을 이미 사용하고 있었던 것이다.

다만, 코드가 너무 짧아, 디버깅이 순식간에 끝나버렸다.



단축키 **F5**를 누르거나,
Run -> Start Debugging을 이용해서도
디버깅을 할 수 있다.

디버깅 하는 법

디버깅 실습을 위해 아래와 같은 코드를 입력해보자.

while은 반복문을 작성하기 위한 코드의 일종으로, 아래의 코드는 영원히 끝나지 않는다.

입력:

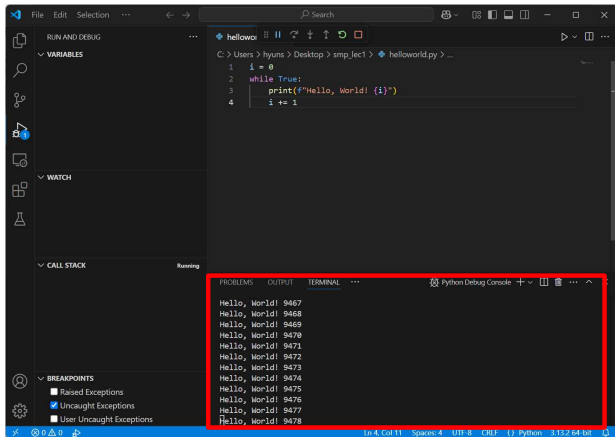
```
i = 0
while True:
    print(f"Hello, World! {i}")
    i += 1
```

출력:

```
Hello, World! 0
Hello, World! 1
Hello, World! 2
Hello, World! 3
...
```

디버깅 하는 법

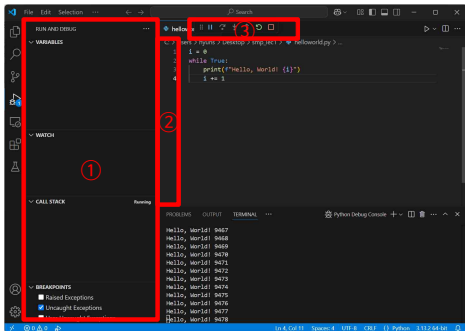
TERMINAL 창에서 숫자가 1씩 늘어나며 계속 출력이 이루어지는 모습을 볼 수 있다.



디버깅 하는 법

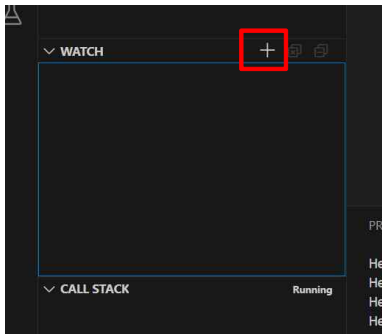
디버깅은 아래의 창을 이용한다.

- ① 변수의 값 등을 관찰할 수 있다.
- ② 코드의 단계적 실행을 확인할 수 있다.
- ③ 코드의 실행을 일시정지/재시작/중지할 수 있다.

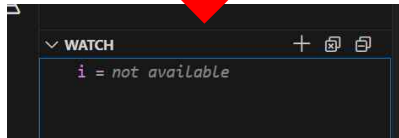
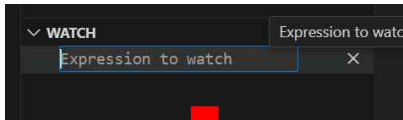


디버깅 하는 법 - 변수 값 관찰하기

①번 위치의 **WATCH** 창 안에 커서를 가져다대면 우측 위에 + 버튼이 생긴다.



+ 버튼을 누르고 “i”를 입력해보자



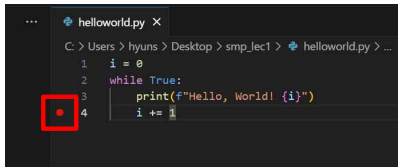
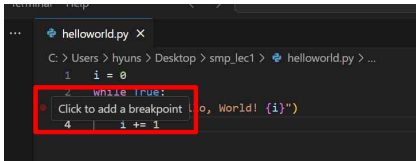
아직은 **not available**인 것을
확인할 수 있다.

디버깅 하는 법 - 변수 값 관찰하기

변수의 값을 추적하기 위해서는, ②번으로 가서 일시중지할 줄을 찾아야 한다.

줄을 알려주는 숫자(1,2,3,4) 왼쪽에 마우스 커서를 가져다 대면, Click to add a breakpoint라는 문구가 뜨게 된다.

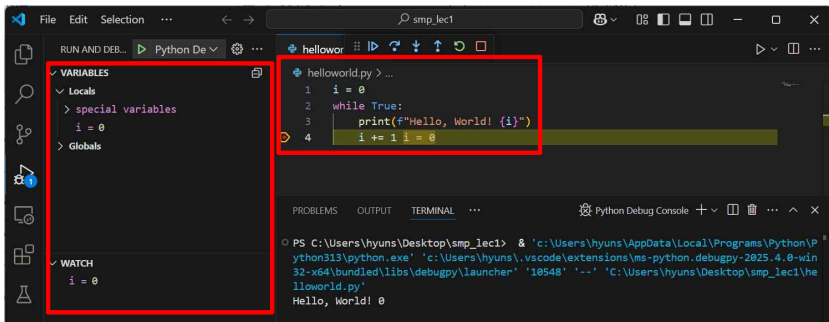
여기서 원클릭하면, 우측과 같이 빨간색 점이 점등된다.



디버깅 하는 법 - 변수 값 관찰하기

이제, 코드를 디버깅 모드로 다시 실행하자. 디버깅 모드의 단축키 **F5**를 외우면 편하니 다들 기억해두자.

코드의 실행이 4번째 줄에서 멈춘 것을 확인할 수 있다. 더불어, **i**의 값도 출력되기 시작했다.



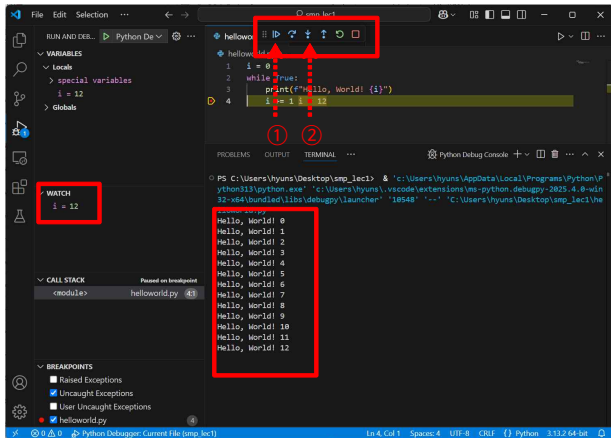
디버깅 하는 법 - 변수 값 관찰하기

이제, 버튼을 눌러

- ① 다음 빨간 점까지 실행하거나,
- ② 코드를 1줄씩 실행하며,

WATCH 창을 보면 변수 `i`의 값이 1씩 증가하는 것을 확인해 볼 수 있다.

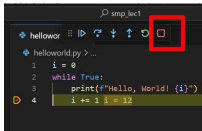
터미널 창에서 코드도 1줄씩 실행되는 것을 확인할 수 있다.



디버깅 종료

빨간색 네모 버튼을 클릭하면, 디버깅을 종료할 수 있다.

디버깅이 아닌 기본 실행에서도 마찬가지로 같은 버튼을 눌러 실행을 종료할 수 있다.

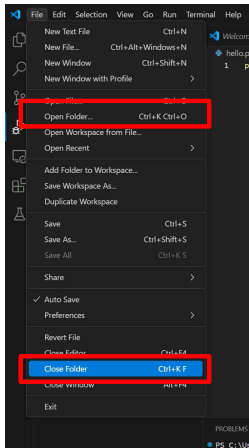


프로젝트 종료

프로젝트를 끝내려면

File -> Close Folder를 누른다.

반대로 시작하려면 앞서 봤던 것처럼 Open Folder를 누르면 된다.



Lecture1 끝
Q/A?