

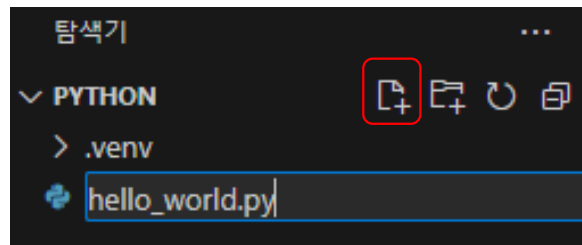
LLM 활용 인공지능 서비스 개발자 양성과정

도봉 SeSAC 캠퍼스 X **Saltlux**

강사 최동혁

Hello, Python!

- 개발환경 설정 드디어 끝!
- 파이썬 파일 만들기
 - Visual Studio Code 탐색기에 “새 파일” 버튼 누르고 파일 이름 입력



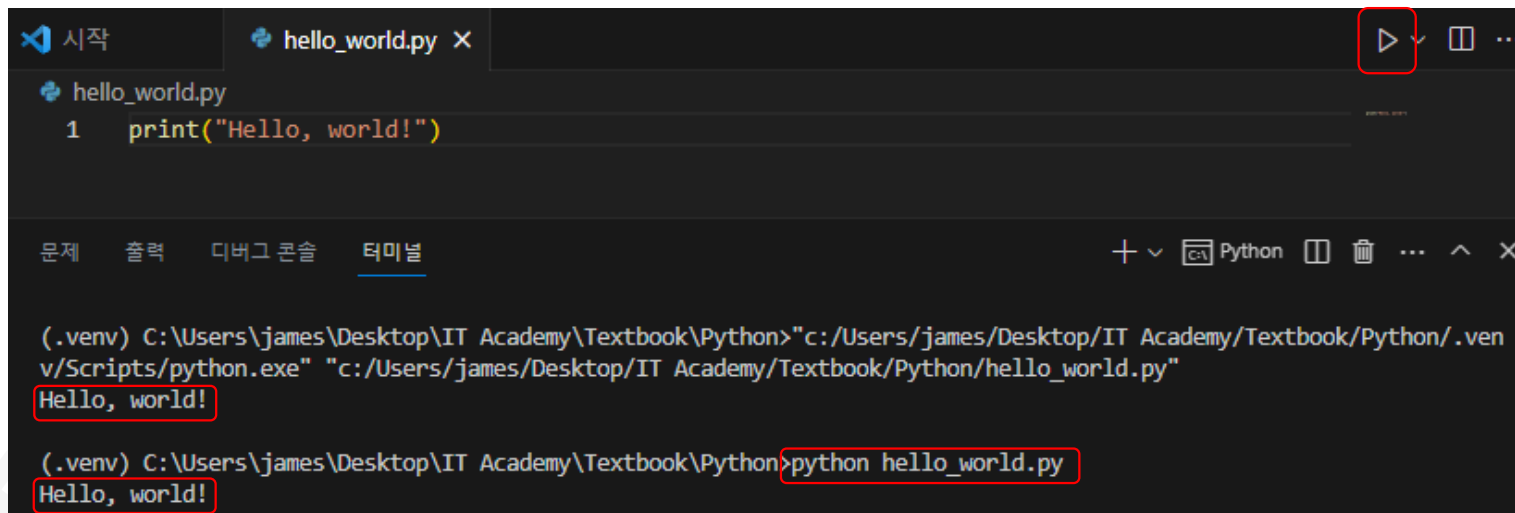
Hello, Python!

- 배울 첫 명령: `print()`
- 터미널에 `print()` 안에 들어가 있는 것을 출력
- 명령에 괄호가 있으면 함수라고 부른다
 - 수학 시간에 배우는 함수와 비슷하다
 - $y = f(x) \Rightarrow x$ 를 입력하면 y 가 나온다
 - 프로그래밍에서 보이는 함수들의 입력값을 인수, 출력값을 반환값이라 부른다

```
1 print("Hello, world!")
```

Hello, Python!

- 파이썬 코드를 실행하는 방법:
 - 우측 상단에 있는 실행 버튼 클릭
 - 터미널에서 "python hello_world.py" 입력
 - 여기서 "hello_world.py"가 상대 경로이고, python 명령어는 python 코드를 실행해주는 명령이다



The screenshot shows a code editor with a file named `hello_world.py` containing the following code:

```
1 print("Hello, world!")
```

Below the code editor is a terminal window. The terminal shows the command `python hello_world.py` being executed, and the output `Hello, world!` is displayed. The command and output are highlighted with red boxes.

Hello, Python!

- 줄이 여러 개면 순차적으로 하나씩 실행이 된다

```
1 print("Hello, world 1!")  
2 print("Hello, world 2!")  
3 print("Hello, world 3!")  
4 print("Hello, world 4!")
```

실행

```
Hello, world 1!  
Hello, world 2!  
Hello, world 3!  
Hello, world 4!
```

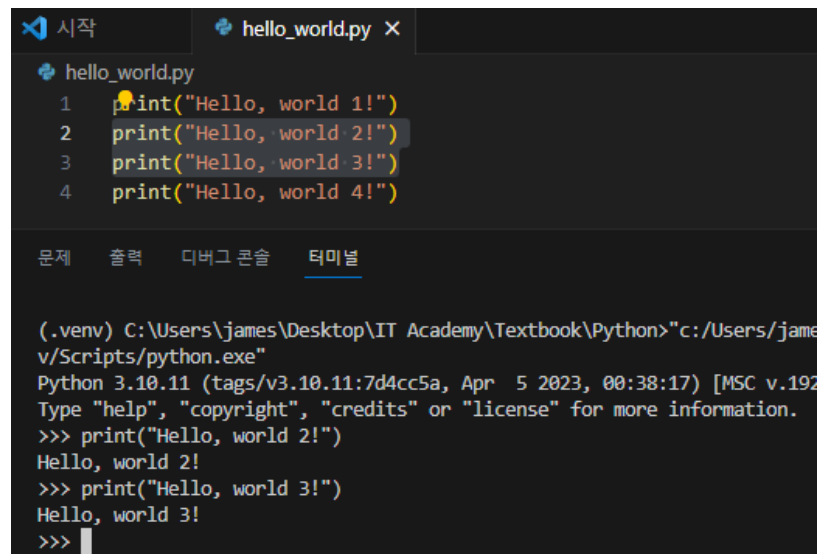
Hello, Python!

- 터미널에서 "python"만 입력하면 파이썬 인터프리터가 켜진다
 - 코드를 한줄씩 입력하면서 코드의 결과값을 쉽게 확인할 수 있는 프로그램이 켜진다
 - 파이썬 인터프리터를 끄려면 `exit()` 입력

```
(.venv) C:\Users\james\Desktop\IT Academy\Textbook\Python>python
Python 3.10.11 (tags/v3.10.11:7d4cc5a, Apr  5 2023, 00:38:17) [MSC v.19
Type "help", "copyright", "credits" or "license" for more information.
>>> print("Hello, world!")
Hello, world!
>>> █
```

Hello, Python!

- Visual Studio Code로 선택한 줄을 파이썬 인터프리터로 돌리는 기능도 있다
 - 실행하고 싶은 줄을 선택한 후 Shift+Enter 입력하거나 우측 클릭 > "Python 실행" > "파이썬 터미널에서 선택/줄 실행"



The screenshot shows the Visual Studio Code interface. The top editor pane displays a file named `hello_world.py` with the following code:

```
1 print("Hello, world 1!")
2 print("Hello, world 2!")
3 print("Hello, world 3!")
4 print("Hello, world 4!")
```

The bottom pane shows the `터미널` (Terminal) tab. The terminal output is as follows:

```
(.venv) C:\Users\james\Desktop\IT Academy\Textbook\Python>"c:/Users/jame
v/Scripts/python.exe"
Python 3.10.11 (tags/v3.10.11:7d4cc5a, Apr 5 2023, 00:38:17) [MSC v.192
Type "help", "copyright", "credits" or "license" for more information.
>>> print("Hello, world 1!")
Hello, world 1!
>>> print("Hello, world 2!")
Hello, world 2!
>>> print("Hello, world 3!")
Hello, world 3!
>>> 
```

디버깅 (Debugging)

- 버그란 (bug)?
 - 프로그램에서 잘못된 부분
- 디버깅이란?
 - 버그를 찾고 고치는 작업
- 코딩하면서 수많은 생각치 못한 오류가 생긴다
 - Syntax error: 파이썬이 프로그램을 이해 못해서 실행이 아예 안 되는 오류
 - Runtime error: 프로그램 실행 중에 에러 메시지가 뜨고 프로그램이 종료되는 오류
 - Semantic error: 프로그램 실행 중 에러 메시지는 없지만 예상치 못한 결과가 나오는 오류

디버깅 (Debugging)

```
hello_world.py
1 print("Hello, world 1!")
2 print("Hello, world 2!")
3 prin("Hello, world 3!")
4 print("Hello, world 4!")
```

어느 줄을 실행했을 때 오류
가 뜨는지 보여주는

Traceback

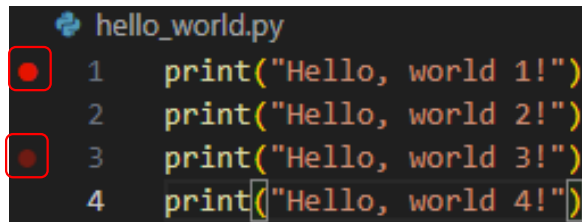
```
(.venv) C:\Users\james\Desktop\IT Academy\Textbook\Python>python hello_world.py
Hello, world 1!
Hello, world 2!
Traceback (most recent call last):
  File "C:\Users\james\Desktop\IT Academy\Textbook\Python\hello_world.py", line 3, in <module>
    prin("Hello, world 3!")
NameError: name 'prin' is not defined. Did you mean: 'print'?
```

어떤 종류의
오류인지

왜 이 오류
가 났는지에
대한 추가적

디버깅 (Debugging)

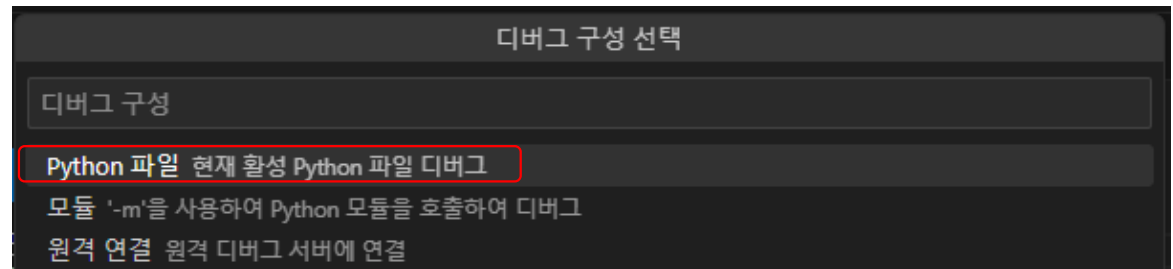
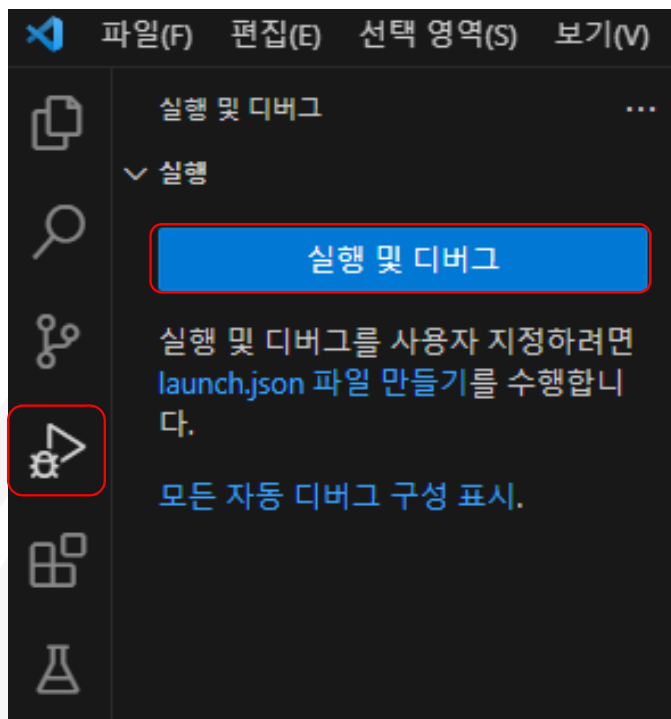
- Visual Studio Code에 디버그 도구가 있다
- 줄 숫자 왼쪽을 클릭하면 그 줄을 중단점(breakpoint)으로 만들 수 있다
 - 디버그 도구로 프로그램을 실행할 때 중단점인 줄이 실행되기 전에 멈춘다



```
hello_world.py
1  print("Hello, world 1!")
2  print("Hello, world 2!")
3  print("Hello, world 3!")
4  print("Hello, world 4!")
```

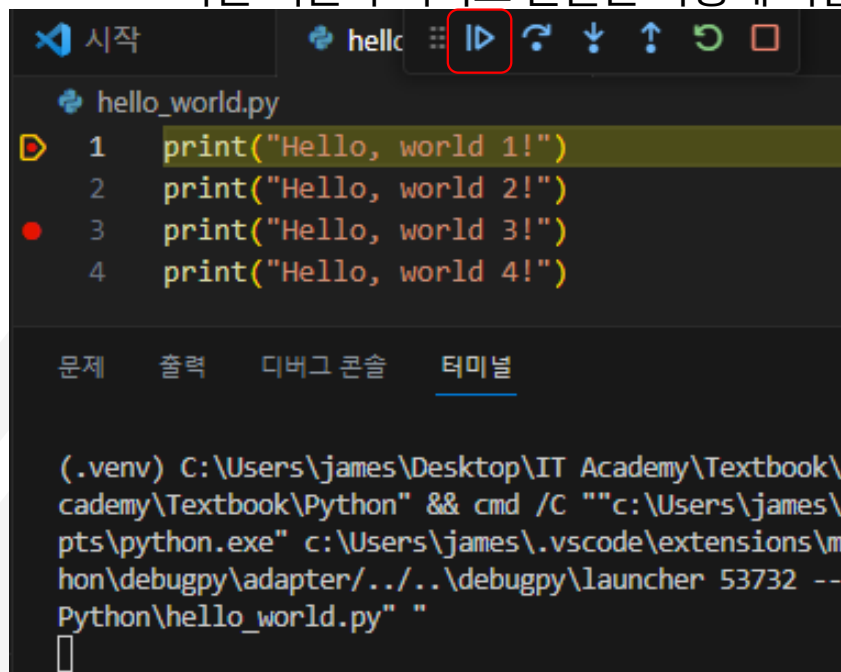
디버깅 (Debugging)

- 좌단에 실행 및 디버그 버튼 누르기 > 실행 및 디버그 > "Python 파일"



디버깅 (Debugging)

- 처음에 터미널에서 아무 출력도 안 뜬다 – 첫 줄 이전에 중단점 때문이다
- 계속 버튼을 누르면 다음 중단점까지 프로그램이 실행된다
 - 다른 버튼과 디버그 콘솔은 나중에 다룰 것이다

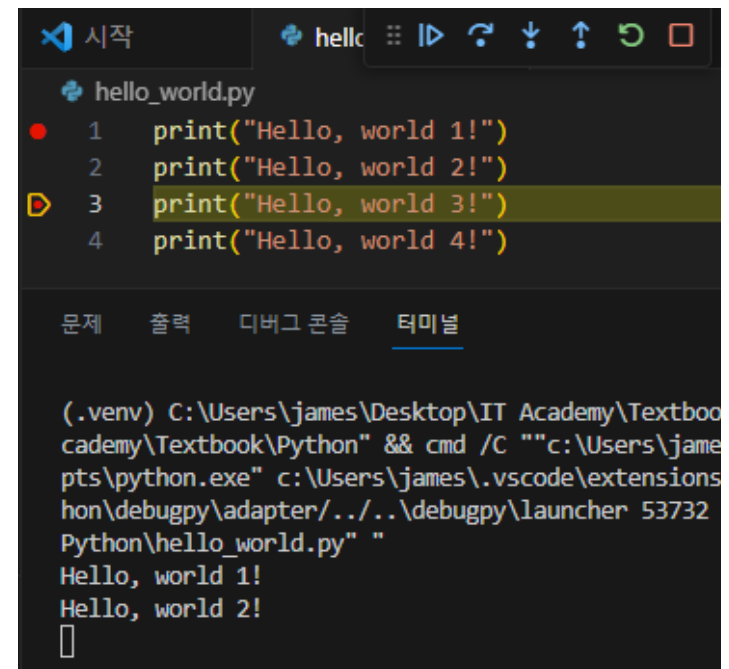


The screenshot shows the VS Code interface with a file named `hello_world.py` open. The code contains four lines of `print` statements. A red dot (breakpoint) is set on line 1. The 'Run and Debug' button (a play icon with a bug) in the top toolbar is highlighted with a red box. The bottom panel shows the 'Terminal' tab with the command prompt output.

```
hello_world.py
1 print("Hello, world 1!")
2 print("Hello, world 2!")
3 print("Hello, world 3!")
4 print("Hello, world 4!")

문제 출력 디버그 콘솔 터미널

(.venv) C:\Users\james\Desktop\IT Academy\Textbook\
cademy\Textbook\Python" && cmd /C ""c:\Users\jame
pts\python.exe" c:\Users\james\.vscode\extensions
hon\debugpy\adapter\..\..\debugpy\launcher 53732 --
Python\hello_world.py" "
```



The screenshot shows the same VS Code interface, but the breakpoint is now on line 3. The 'Continue' button (a right-pointing arrow) in the top toolbar is highlighted with a red box. The bottom panel shows the 'Terminal' tab with the output of the first two lines of the program.

```
hello_world.py
1 print("Hello, world 1!")
2 print("Hello, world 2!")
3 print("Hello, world 3!")
4 print("Hello, world 4!")

문제 출력 디버그 콘솔 터미널

(.venv) C:\Users\james\Desktop\IT Academy\Textboo
cademy\Textbook\Python" && cmd /C ""c:\Users\jame
pts\python.exe" c:\Users\james\.vscode\extensions
hon\debugpy\adapter\..\..\debugpy\launcher 53732
Python\hello_world.py" "
Hello, world 1!
Hello, world 2!

```


Hello, Python! 2

- input() 함수

- 괄호 안에 있는 것을 터미널에 출력한 후 사용자의 입력을 기다린다

```
1 input("Tell me something!")
```

사용자가 Hello, Python!을 입력



```
Tell me something! █
```

```
Tell me something! Hello, Python!
```

Hello, Python! 2

- input() 명령

- 사용자가 입력한 것을 반환을 한다

- print() 함수와 같이 쓰면 이 사실을 확인할 수 있다

```
1 print(input("Tell me something! "))
```

사용자가 Hello, Python!을 입력



```
Tell me something! █
```

```
Tell me something! Hello, Python!  
Hello, Python!
```

Hello, Python! 2

- 왜 이런 결과가 나올까?

- 프로그램이 각 줄을 왼쪽부터 오른쪽으로 읽고, 먼저 나오는 함수를 실행하려고 한다
- 인수나 피연산자는 값이어야 한다
- print() 함수 안에 있는 것을 출력하고 싶은데 print 함수에 들어가는 인수가 함수라 함수의 반환값을 받을 때까지 기다린다

```
1 print(input("Tell me something! "))
```

```
Tell me something! Hello, Python!  
Hello, Python!
```

Hello, Python! 3

- 파이썬 인터프리터로 간단한 연산을 할 수 있다

```
(.venv) C:\... python
Python 3.10.11 ...

>>> 3 + 3
6
>>> 3 - 3
0
>>> 3 * 3
9
>>> 30 - 2*6
18
>>> (30 - 2*6) / 3
6.0
>>> 17 / 3
5.666666666666667
>>> 17 // 3
5
>>> 17 % 3
2
>>> 2 ** 5
32
```


Hello, Python! 3

- 코드로 구현하고 싶다면 print() 함수를 사용할 수 있다

```
1    print(3 + 3)
2    print(3 - 3)
3    print(3 * 3)
4    print(30 - 2 * 6)
5    print((30 - 2 * 6) / 3)
6    print(17 / 3)
7    print(17 // 3)
8    print(17 % 3)
9    print(2 ** 5)
```

Hello, Python! 4

- 코드에 주석을 남길 수 있다
- # (한줄 주석)
- ''' ... ''' (여러줄 주석)

```
1 print("Hello world 1!")
2 # print("Hello world 2!")
3
4 '''
5 이것도
6 마찬가지로
7 '''
```

```
Hello world 1!
```

VS Code 유용한 단축키

- Ctrl + /
 - 블록지정하면 모든 코드를 주석처리 해준다.
- Ctrl + Tab
 - 파이썬 indentation 한번에
- Ctrl + F
 - 현재 에디터 내에서 문자열 검색
- Ctrl + H
 - 문자열 대체
- Ctrl + ,
 - 설정 창을 열어준다.
 - (참고) VS Code에서는 설정을 JSON 파일로 설정할 수 있다.

VS Code 유용한 단축키

- Ctrl + ` (숫자 1 옆에)
 - 새 터미널 열기
- Ctrl + (+, -)
 - 글자 크기 늘리기 / 줄이기
- Ctrl + Shift + P
 - 명령 팔레트
 - 주요 메뉴들을 빠르게 접근시켜 준다.
- Ctrl + K, F
 - 현재 워크스페이스 종료
 - (참고) 워크스페이스와 프로젝트 개념을 이해하자.

VS Code 유용한 단축키

- Ctrl + \
 - 화면 분할. (유용함)
- Ctrl + K, Ctrl + F
 - 코드 포매팅
 - (참고) Prettier를 쓰면 저장만 해도 포매팅이 된다.
- • Ctrl + Shift + L
 - 블록 씩운 문자열을 한번에 바꿔준다. (대단히 유용함.)