



# 4주차\_VS Code Debugging

## 공지사항

1. **팀구성 완료** : CSID-DGU에 초대 → 팀장이 repo 만들고 팀원들 모두 초대하기 **(04/04 23:59까지)**

repo 형식 : **2023-1-OSSProj-팀이름-팀번호**

**public**으로 만들어 다른 사람들도 볼 수 있게 해주세요!

repo가 만들어져있고, 모든 팀원이 초대받아져 있는 화면캡처를 pdf로 올려주시면 됩니다.

**팀구성과제\_팀번호.pdf**

TEAM	TEAM_NAME	팀장	팀원1	팀원2	팀원3
1	H3	정한길	이은학	윤영훈	
2	어?금지	이중원	박지혜	은정민	
3	MAC	민한결	안상연	최필환	
4	핫소스	고영웅	민헌준	이유빈	
5	Prodigy Project	홍범선	김유리	홍서이	
6	3PLUS	신서영	남윤형	이윤서	
7	MathComputer 더 맥스	이지민	박광렬	정관희	정원호

2. **VS Code Debugging과제** : 실습 과제 제출 → **(04/04 23:59까지)**

3. **제안서**

-팀별 제안서 작성 회의 및 시험 대비 질의/응답 시간 : **04/05(WED)**

-제안서 발표 : **04/12(WED)**

-제안서 및 발표자료 제출일 : **04/11(TUE) 21:00 까지**



발표자료: 팀장만 제출

발표시간: 총 12분으로 진행

팀당 9분이내(모든 팀원이 같은 시간동안 나누어 발표해야함) +질의응답 3분

제출할 파일 : **1) 제안서 2) 발표자료**

## 이전 팀들 자료

CSID DGU

Convergence Software Institute at Dongguk University - CSID DGU

<https://github.com/CSID-DGU?q=2022-2-OSSProj&type=all&language=&sort=>



**매주 월요일 오전 9시까지 회의록 받음!**

-중간 발표 : **05/17**

-최종 발표 : **06/14**



### <채점 기준>

제출해야할 파일 : 최종제안서, 최종발표자료, 데모영상, github 링크

데모영상은 Youtube나 google drive에 업로드하여 링크를 첨부!

Github : Readme에 프로젝트 제목, 내용, 실행 환경, 설치 방법, 코드 예제 및 실제 적용 사례, 연락처(팀원의 프로필등)등이 필수적으로 포함되어야 함

참고링크 : <https://github.com/matiassingers/awesome-readme>

### <채점기준>

Completeness: 해당 팀이 무엇을 하고 있고, 현재 진도가 얼마나 나갔는지 알 수 있어야 함

진행하고자 하는 프로젝트 내용이 포함되어 있어야 함

지금까지 무엇을 진행 했는지

팀원 각자의 역할 구분 있어야 함

전체 및 각자의 타임라인

3~4명이 두 달 동안 1주일에 4시간, 즉, 32시간 x 팀원 수 만큼의 프로젝트를 진행할 수 있는 양을 진행했어야 함

팀원간의 Github 협업활동기록

### <주의사항>

6분 발표이며, 팀원간 최대한 적절하게 시간분배를 해주세요!

6분 타이머를 맞춰놓고 알람이 울리면 더 이상 발표를 진행하지 못합니다!

## Debugging



### Debugging

Bug = 프로그램이 오작동하는 것

Debugging = 오작되는 현상들을 해결하는 것 → 오류들을 찾아내기 위한 테스트 과정

VS Code에서 debugging하는 법

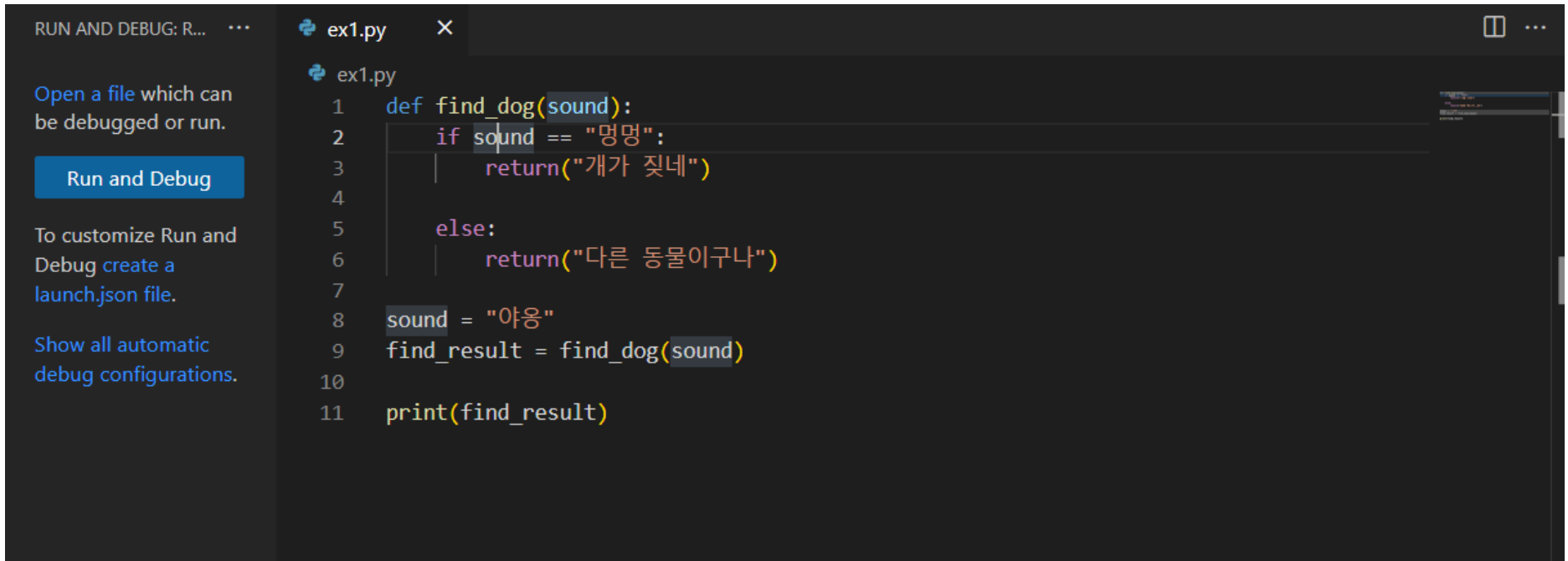
Python으로 간단한 실습!

```
def find_dog(sound):
    if sound == "멍멍":
        return("개가 짖네")

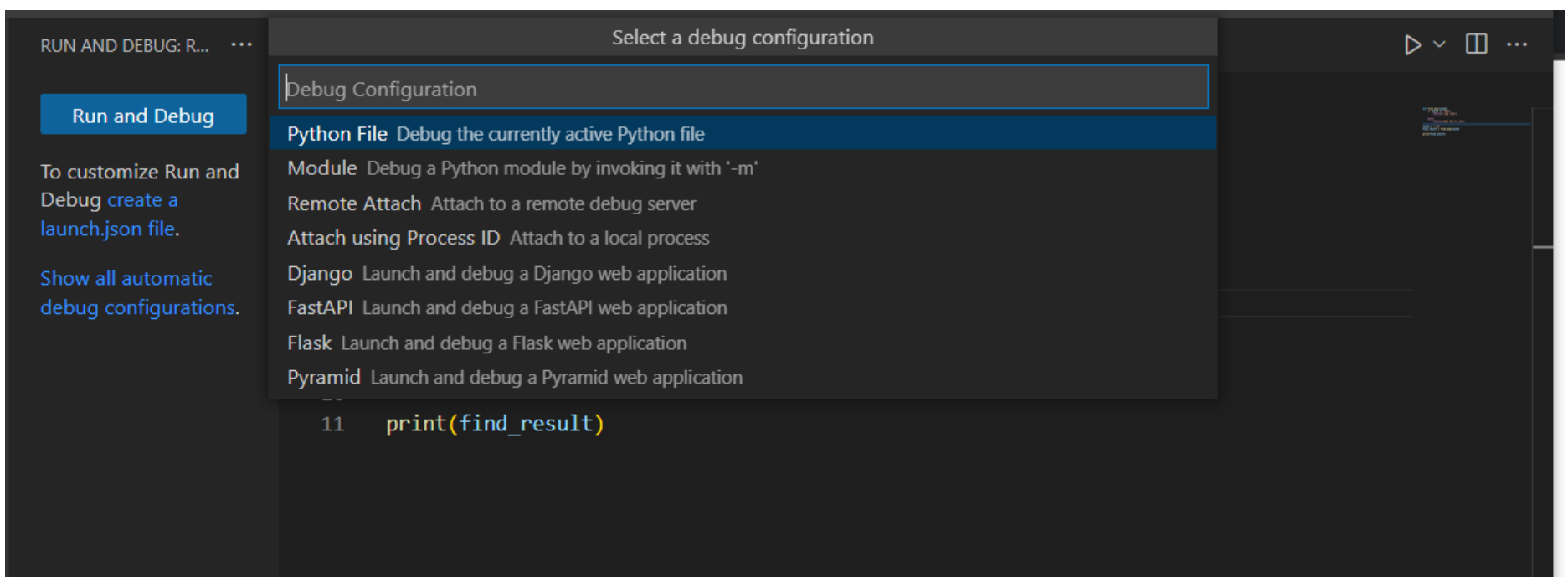
    else:
        return("다른 동물이구나")

sound = "야옹"
find_result = find_dog(sound)

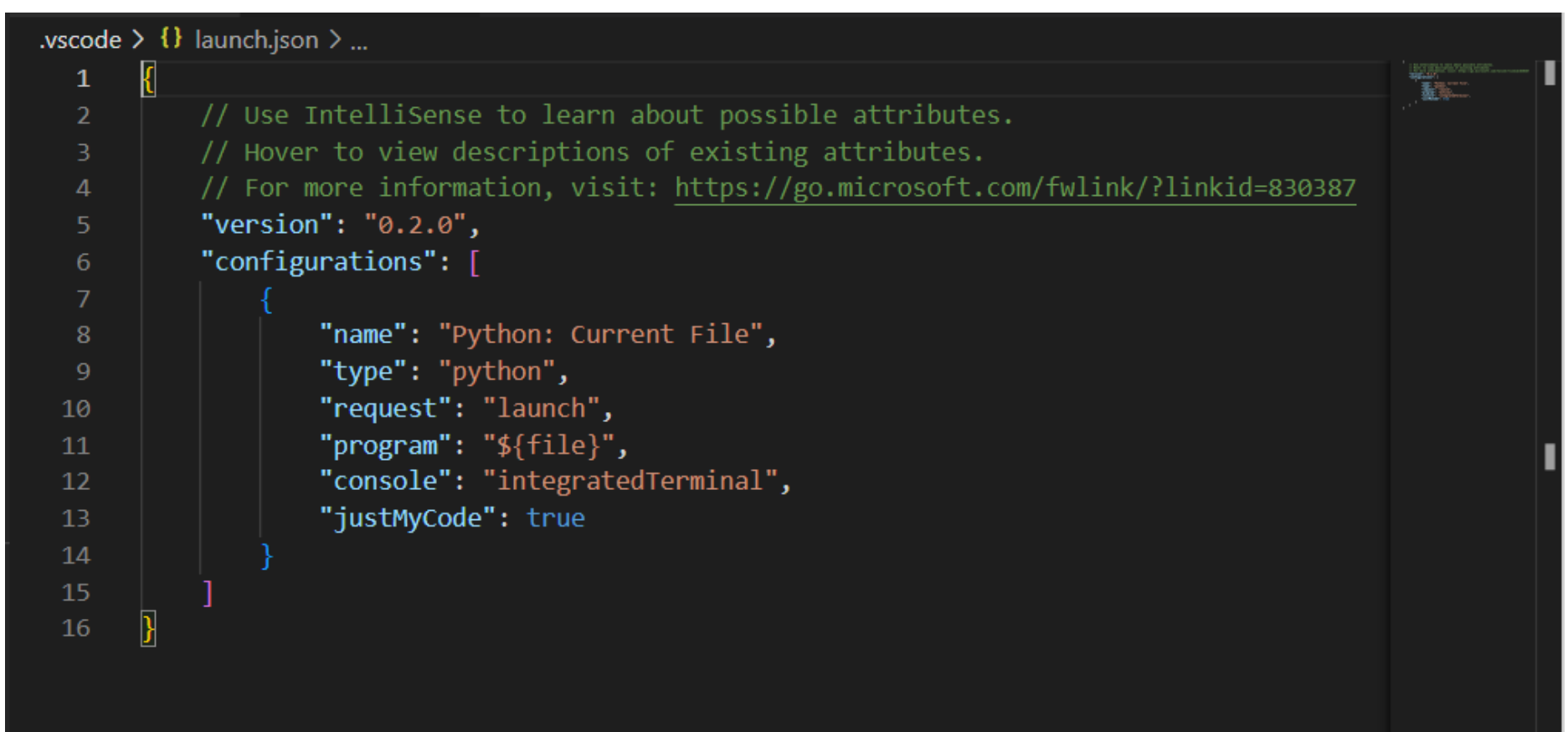
print(find_result)
```



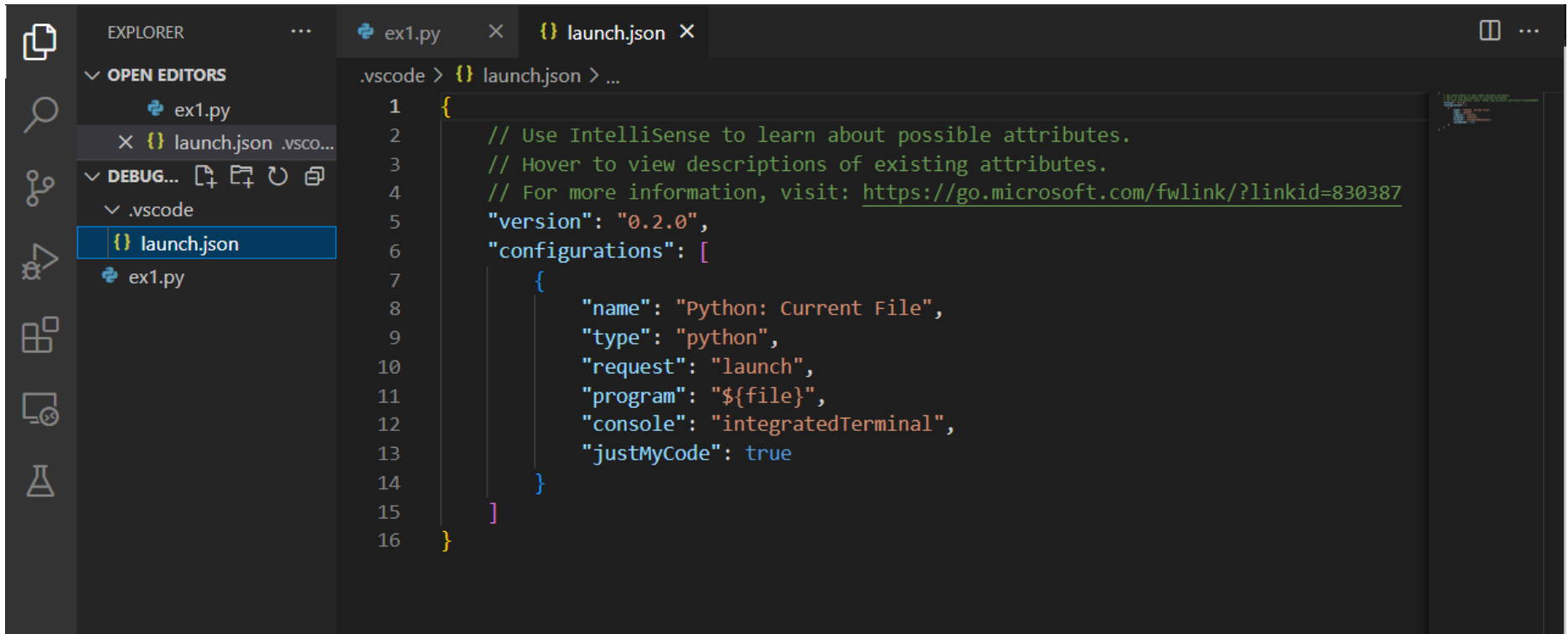
- 왼쪽의 Run and Debug 아래에 `create a launch.json` 클릭



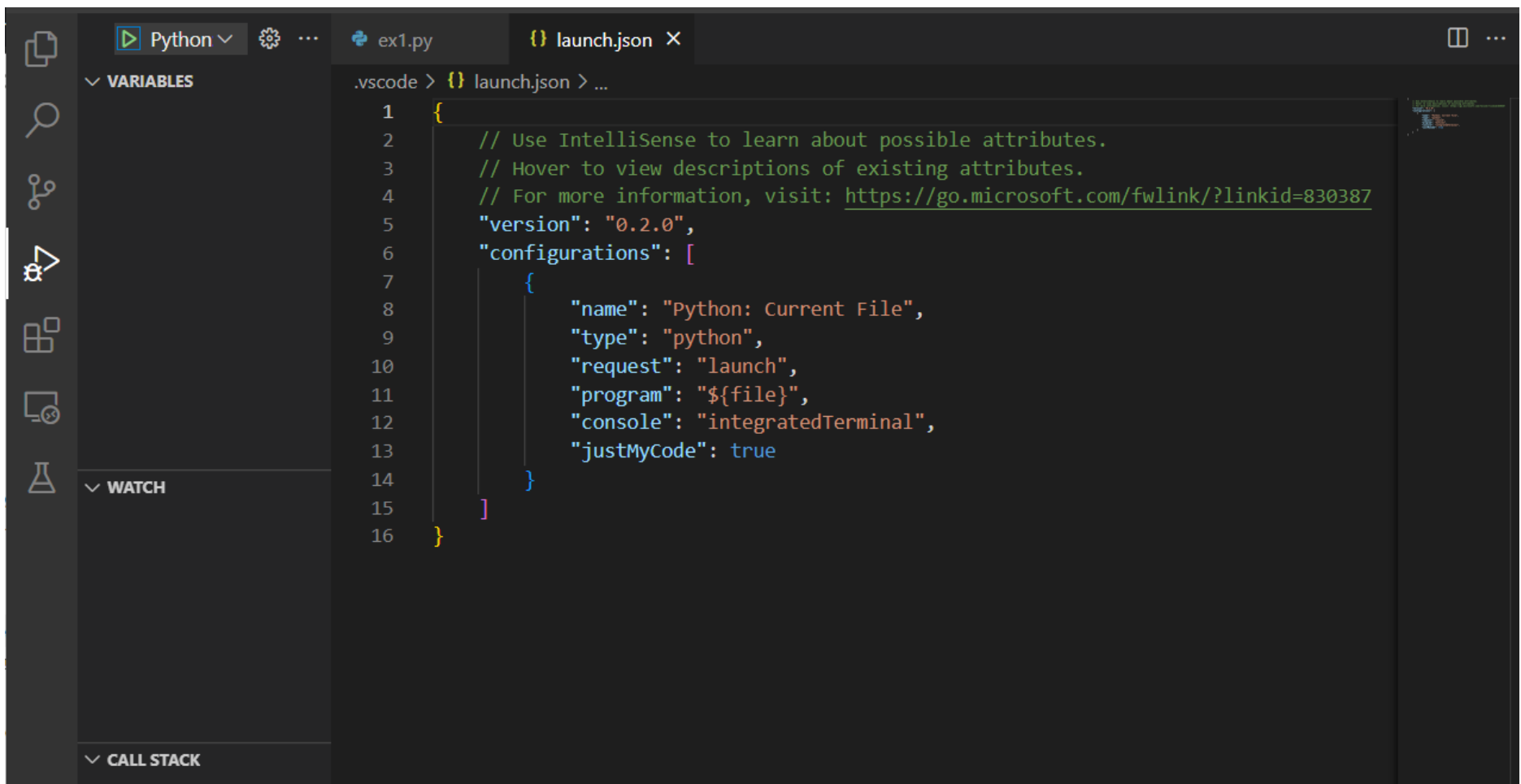
- Python File을 클릭하면 아래와 같은 화면이 나타남



- 아래와 같은 폴더 구조를 갖는 것을 볼 수 있음(해당 설정은 각 프로젝트 폴더에서만 유효! 다른 프로젝트 폴더를 생성한다면 새로 설정해야 합니다)



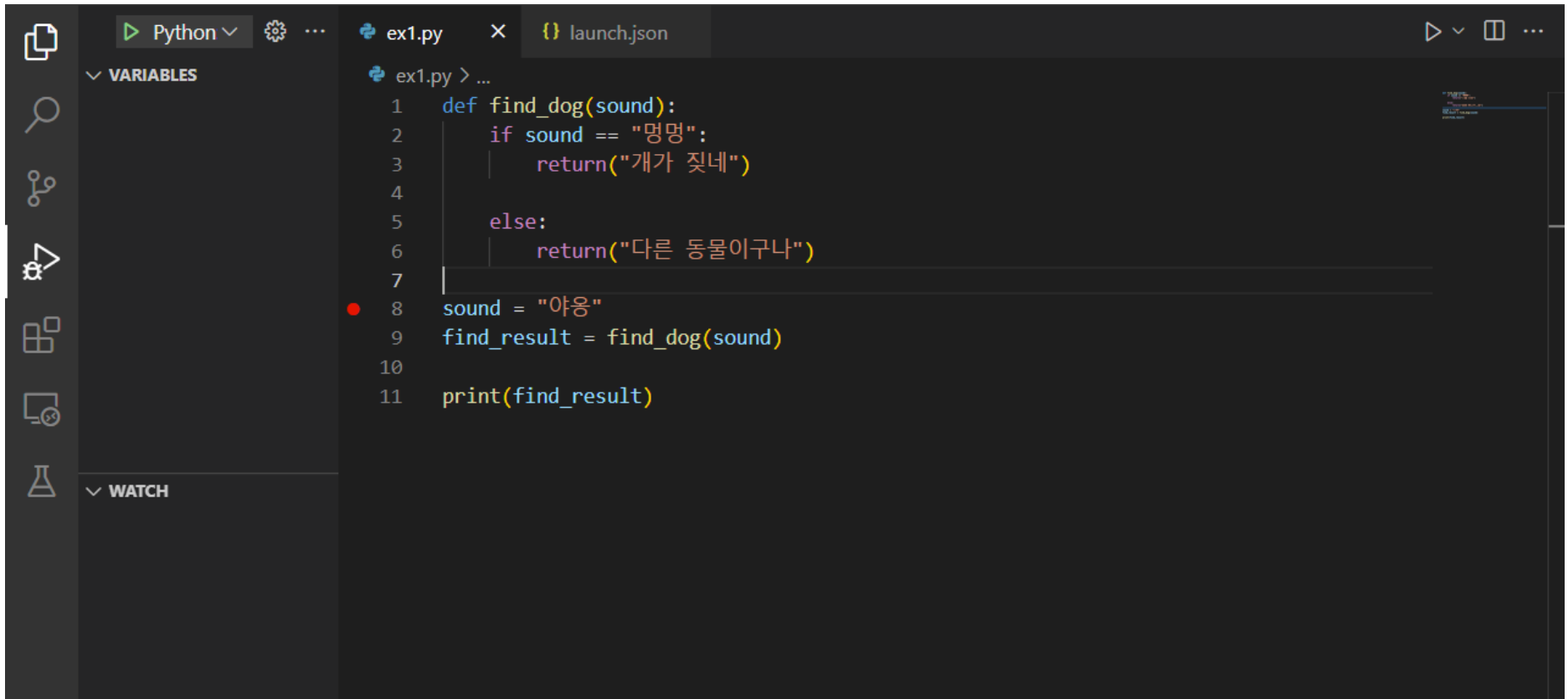
- 다시 debug 버튼을 클릭하면 이번에는 초록색 버튼이 담긴 것을 확인할 수 있음 : `variables`, `watch`, `call stack` 이 존재함



## 1. Break Point 지정하기

커서를 올려서 클릭하면 Break Point를 찍을 수 있음

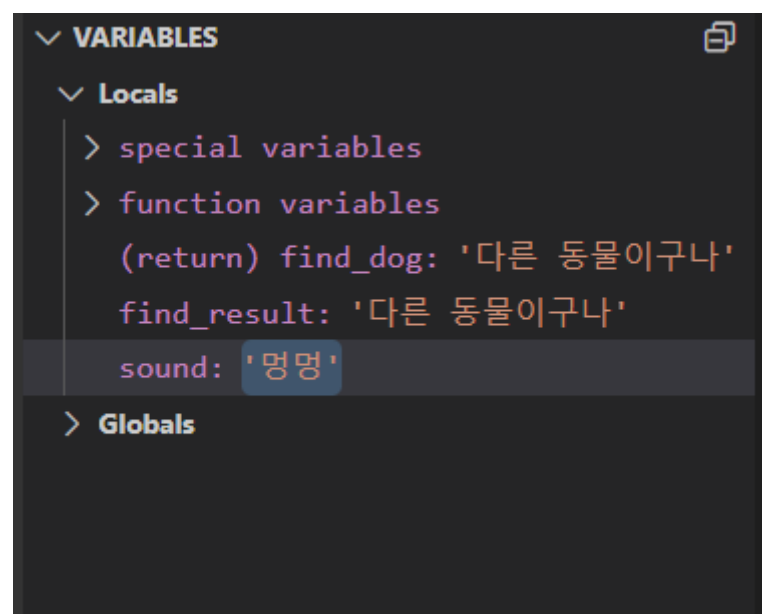
해당 행위의 의미는, Break Point(빨간점)으로 표시한 라인 전까지만 코드가 실행되기를 원한다는 의미



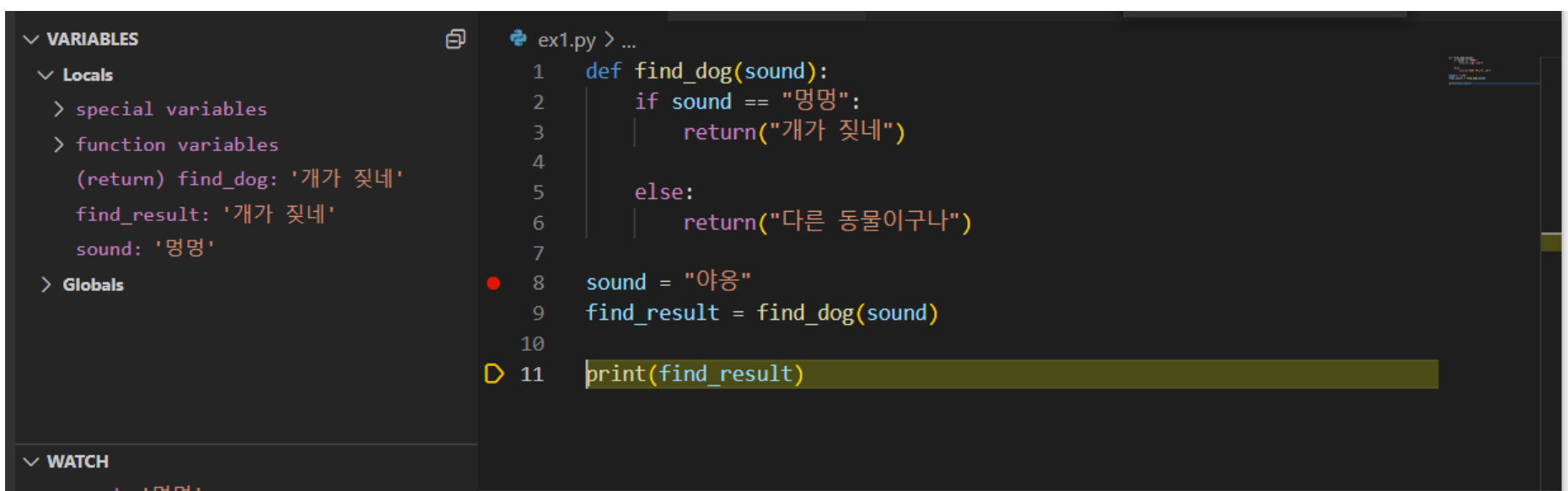
그냥 Run클릭하면 아직 한줄도 실행되지 않은 상태가 됨

따라서, step over를 눌러가면서 코드를 한줄 씩 실행시키면서 왼쪽 variable 이나 stack의 변화를 보면 됨

## 2. Variable을 더블클릭하면 변수명을 바꿀 수 있음

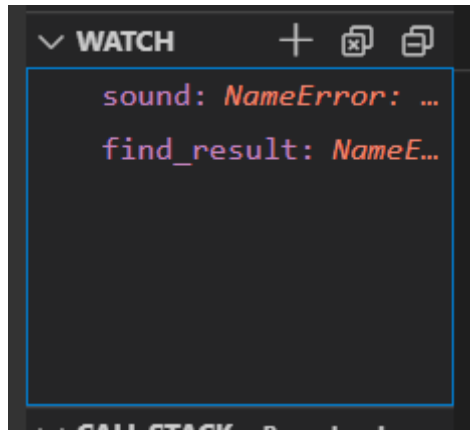


8번째 줄을 실행시킨 다음 '멍멍'으로 바꾸고 9번째줄을 step over를 눌러서 실행시키면 이번에는 아래와 같은 결과가 나오는 것을 볼 수 있음

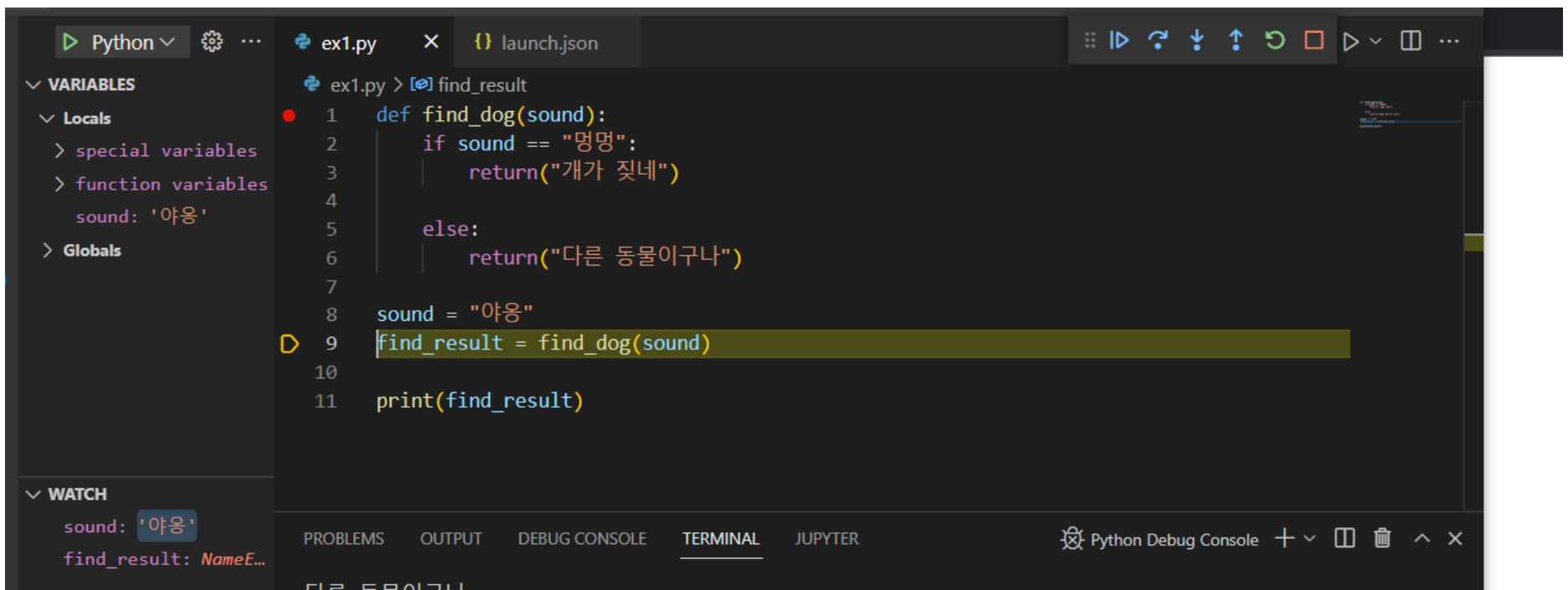


## 3. watch 부분에 확인하고 싶은 변수명 입력

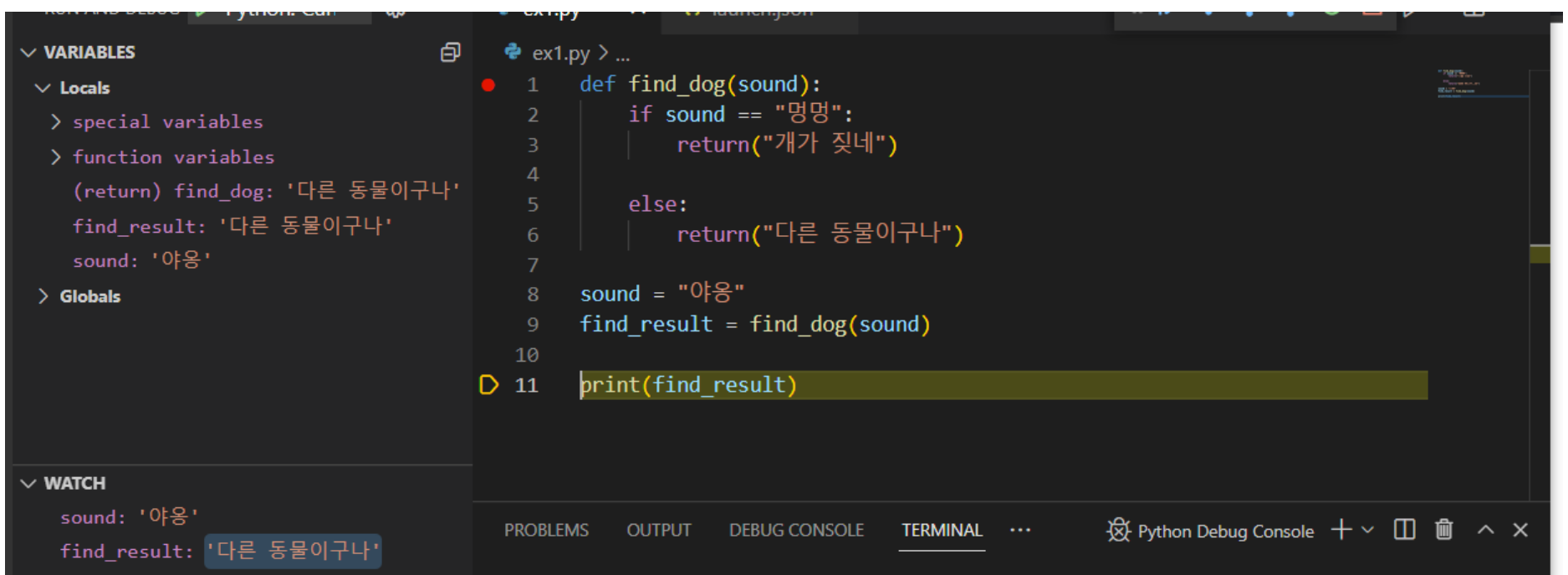
sound, find\_result 부분을 넣어주면 처음에는 변수들이 정의되지 않은 상태라면서 **NameError**를 발생시킴



첫줄에 break point를 달고 **step over** 를 클릭하면 아래와 같이 값을 받는 시점에 sound부분이 “야옹”으로 변하는 것을 확인할 수 있음



한번 더 step over를 눌렀더니 find\_result 값에 다른 동물이구나가 추가되는 것을 확인할 수 있음



## Practice1

### [문제]

섭씨 온도를 입력받아 화씨 온도로 변환해주는 프로그램을 제작하시오.

- 섭씨 온도를 입력받는 변수명은 celsius이다.
- 화씨 온도는 fahrenheit 변수에 대입하시오.
- 섭씨온도를 화씨온도로 변환하는 식:  $fahrenheit = (9/5) * celsius + 32$

### [코드]

```
print('## 섭씨 -> 화씨 변환 프로그램 ##')
#섭씨 온도를 celsius라는 변수에 저장한다.
celsius = input('섭씨 온도를 입력하세요. ')
fahrenheit=(5/9)*celsius+32
```

```
print('섭씨 온도:', celsius, 'C')
print('화씨 온도:', fahrenheit, 'F')
```

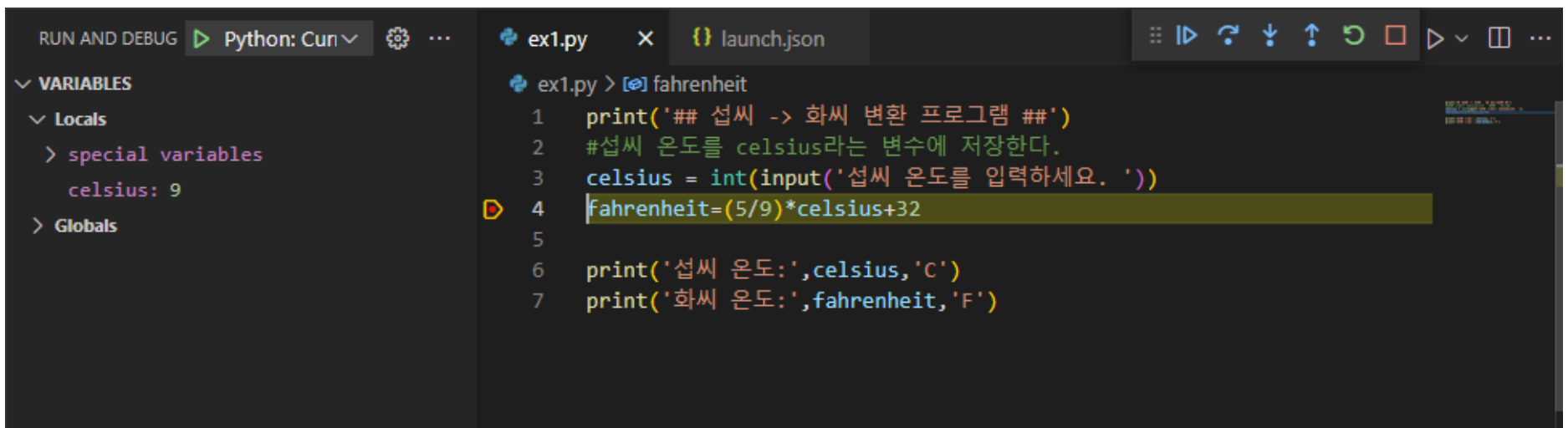
#### [Debugging]

1. 문제가 있어보이는 fahrenheit부분에 break point를 걸어둠

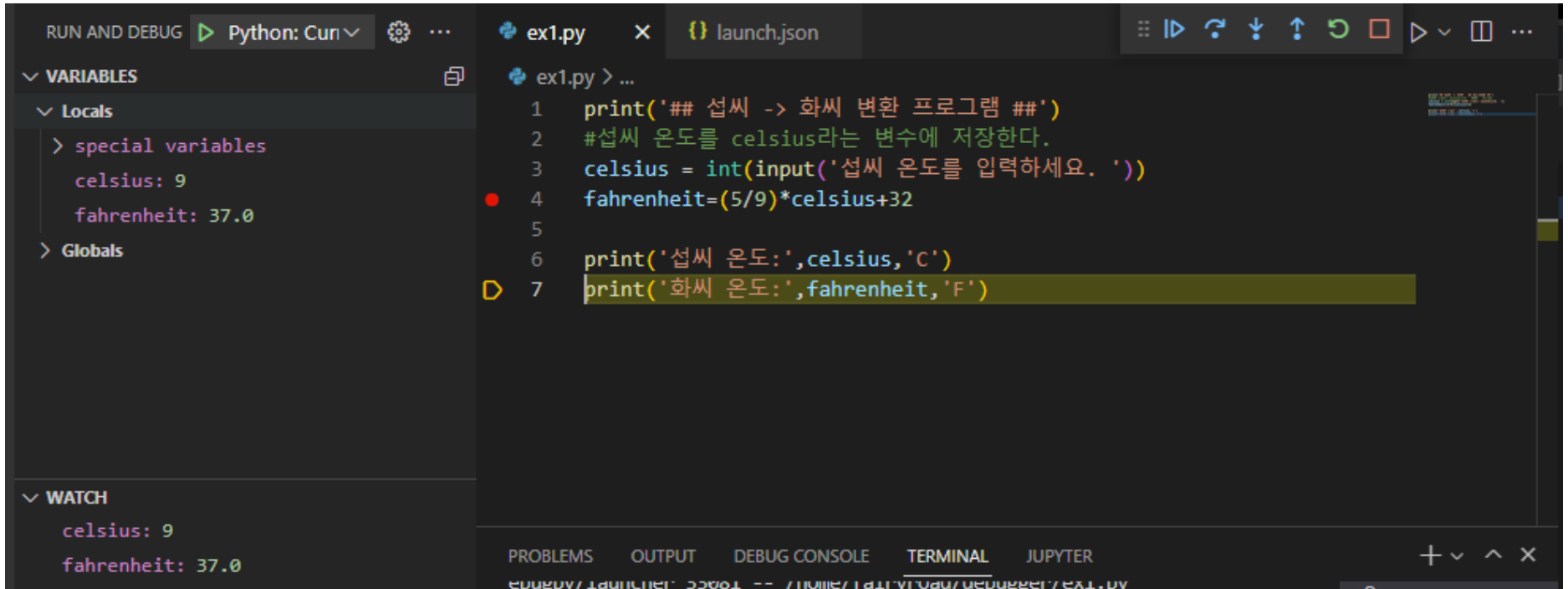
입력받는 창 : 9를 입력

```
fairyroad@DESKTOP-53ORR0M:~/debugger$ cd /home/fairyroad/debugger ; /
usr/bin/env /bin/python3 /home/fairyroad/.vscode-server/extensions/ms-
python.python-2022.14.0/pythonFiles/lib/python/debugpy/adapter/../../d
ebugpy/launcher 35071 -- /home/fairyroad/debugger/ex1.py
## 섭씨 -> 화씨 변환 프로그램 ##
섭씨 온도를 입력하세요. 9
```

2. Run and debug 버튼을 클릭하면 아래와 같은 화면이 나오게 됨



3. WATCH 부분에 Celsius와 fahrenheit를 추가해서 변수 값을 더 자세히 봄



4. step over를 클릭해서 fahrenheit가 어떻게 변하는 지 확인 : fahrenheit가 37.0 임 → 원래는 48.2 여야 함 → 해당 부분에서 오류가 있다는 것을 알 수 있음

#### [정답]

```
print('## 섭씨 -> 화씨 변환 프로그램 ##')
#섭씨 온도를 celsius라는 변수에 저장한다.
celsius = int(input('섭씨 온도를 입력하세요. '))
fahrenheit=(9/5)*celsius+32

print('섭씨 온도:', celsius, 'C')
print('화씨 온도:', fahrenheit, 'F')
```

## Practice2

### [문제]

Sentence 변수에 ‘ Goodbye Mr. My Holiday ‘ 문자열을 입력받고

1. 문자열의 공백을 없애 Sentence 변수에 대입하고 Sentence 변수 값을 출력하시오.
2. 문자열 슬라이싱만을 이용하여 Sentence 변수를 ‘Goodbye Mr. Hody’로 초기화하고 출력하시오.
3. 새로 초기화된 문자열의 길이를 출력하시오.
4. 바뀐 문자열을 모두 각각 소문자, 대문자로 출력하시오.
5. 문자열 내장함수를 사용하여 MR. HODY를 본인 이름으로 대체하시오. (ex. MS. 김땡땡)

### [코드]

```
sentence = input('문자열 입력 :')

# 공백 없애고 대입하기
sentence = sentence.strip()
print(sentence)
# 문자열 슬라이싱하여 초기화
slice1 = sentence[:12]
slice2 = sentence[14:16]
slice3 = sentence[19]
slice4 = sentence[22]
sentence = slice1 + slice2 + slice3 + slice4
print(sentence)
# 문자열 길이 출력
print('변경된 문자열의 길이:', len(sentence))
# 대소문자로 출력
print(sentence.lower())
print(sentence.upper())
# 내 이름으로 대체
print(sentence.upper().replace("MR. HODY", "MS. 김땡땡"))
```

### [Debugger]

1. 어디가 잘못되었는지 잘 모르겠음! Test Case를 생각 → Goodbye Mr. My Holiday 가 Goodbye MS. 정재혁 으로 바뀌어야 함 (자신의 이름으로 코드 변경해주기)

```
debug.py > ...
1  sentence = input('문자열 입력 :')
2
3  # 공백 없애고 대입하기
4  sentence = sentence.strip()
5  print(sentence)
6  # 문자열 슬라이싱하여 초기화
7  slice1 = sentence[:12]
8  slice2 = sentence[14:16]
9  slice3 = sentence[19]
10 slice4 = sentence[22]
11 sentence = slice1 + slice2 + slice3 + slice4
12 print(sentence)
13 # 문자열 길이 출력
14 print('변경된 문자열의 길이:', len(sentence))
15 # 대소문자로 출력
16 print(sentence.lower())
17 print(sentence.upper())
18 # 내 이름으로 대체
19 print(sentence.upper().replace("MR. HODY", "MS. 정재혁"))
```

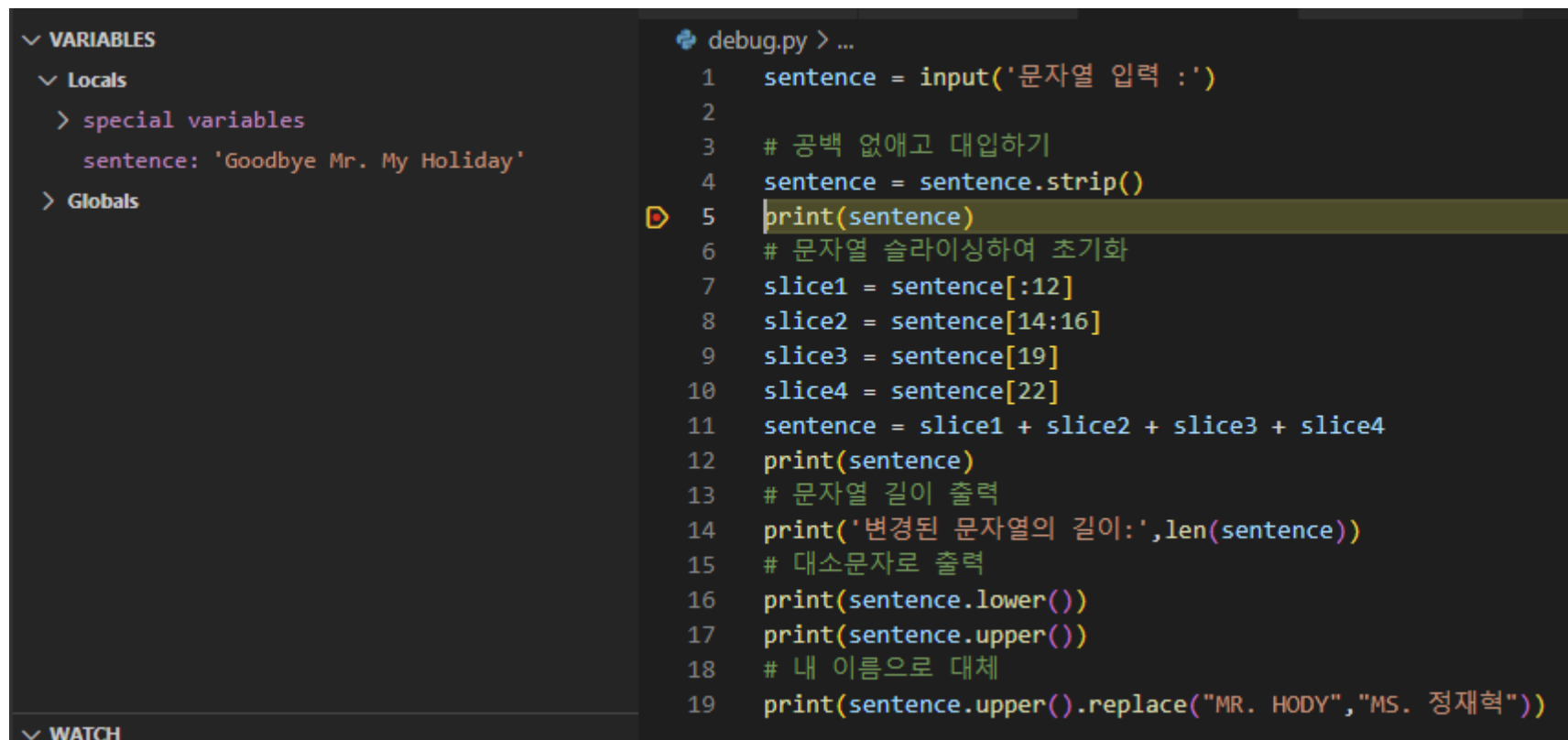
2. 처음부터 쪽 변수값이 어떻게 변화는 지 확인!

첫줄에 break point를 찍고 step over를 하면서 확인을 해봄

아래 부분의 값을 입력해주기 : Goodbye Mr. My Holiday

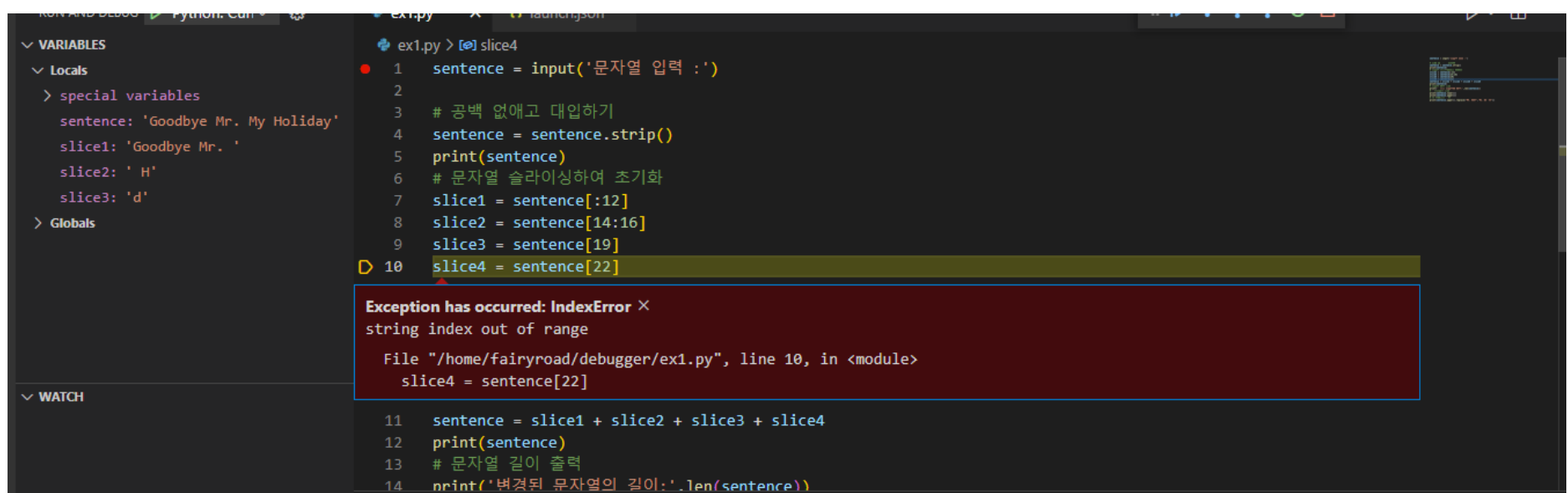
```
fairyroad@DESKTOP-53ORROM:~/debugger$ cd /home/fairyroad/debugger ; /usr/bin/env /bin/python3 /home/fairyroad/.vscode-server
/extensions/ms-python.python-2022.14.0/pythonFiles/lib/python/debugpy/adapters/.../debugpy/launcher 40603 -- /home/fairyroad
/debugger/ex1.py
문자열 입력 :Goodbye Mr. My Holiday
█
```





## 2. step over를 사용하면서 변수의 변화를 잘 보기

다음과 같이 변수 명에 어떻게 값이 들어가고 있고, index out of range가 발생하는 지도 확인할 수 있음



[정답]

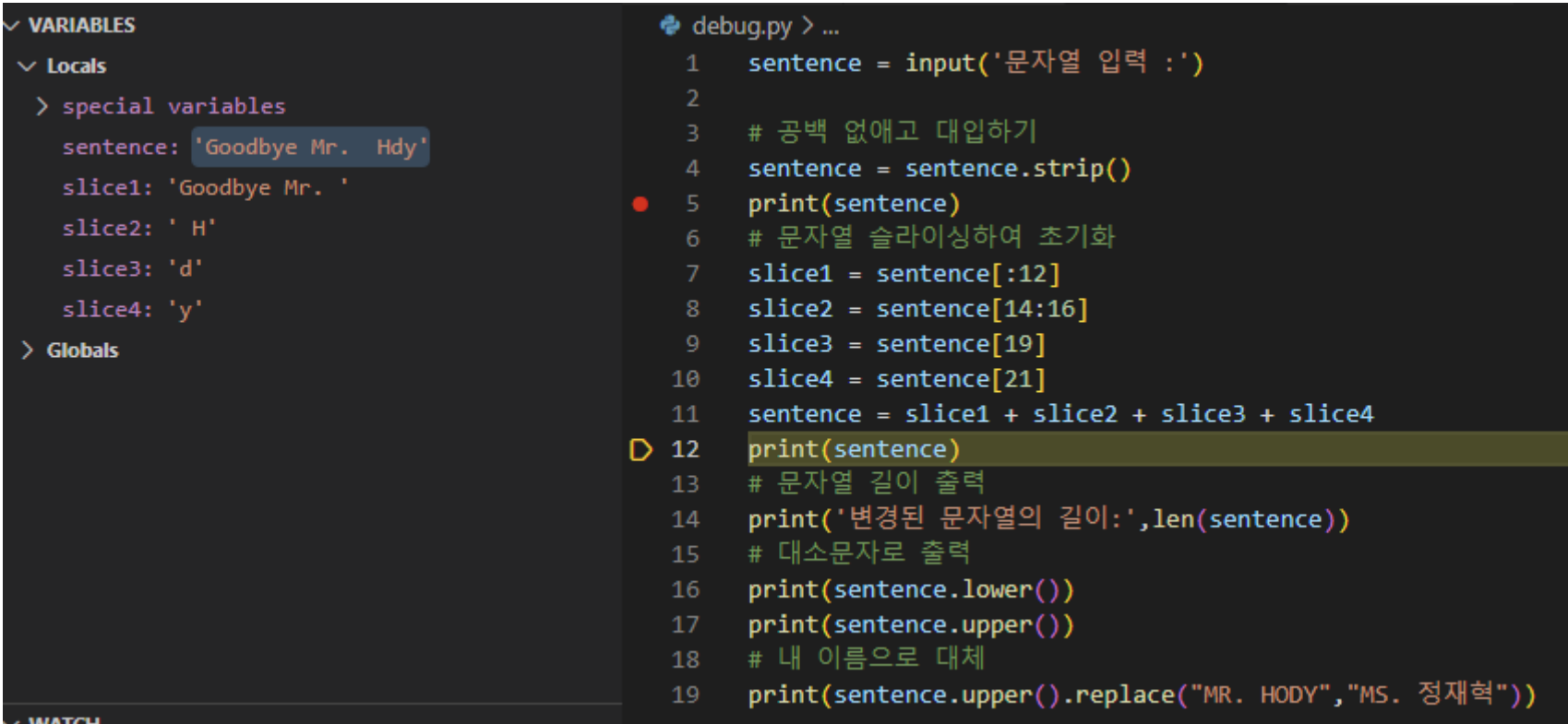
```
sentence = input('문자열 입력 :')

# 공백 없애고 대입하기
sentence = sentence.strip()
print(sentence)
# 문자열 슬라이싱하여 초기화
slice1 = sentence[:12]
slice2 = sentence[15:17]
slice3 = sentence[19]
slice4 = sentence[21]
sentence = slice1 + slice2 + slice3 + slice4
print(sentence)
# 문자열 길이 출력
print('변경된 문자열의 길이:', len(sentence))
# 대소문자로 출력
print(sentence.lower())
print(sentence.upper())
# 내 이름으로 대체
print(sentence.upper().replace("MR. HODY", "MS. 김유정"))
```

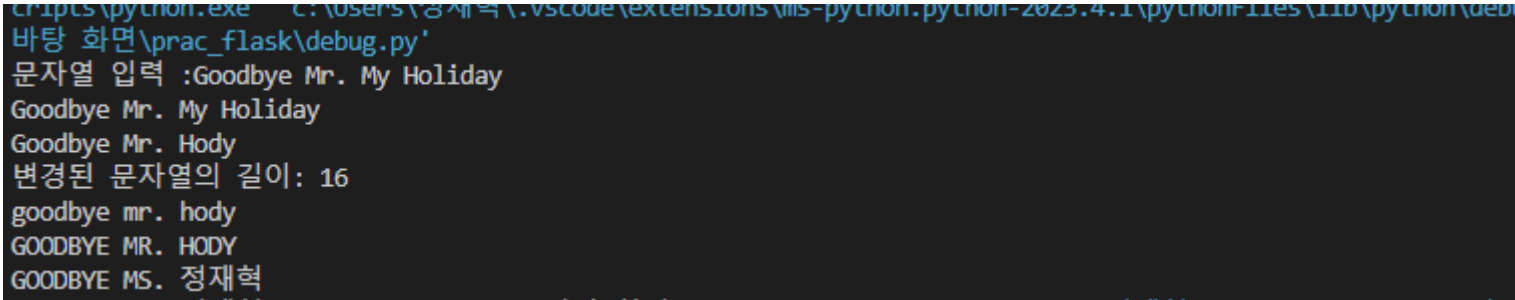
## 1. 다시 확인해 보기

```
fairyroad@DESKTOP-53ORROM:~/debugger$ cd /home/fairyroad/debugger ; /usr/bin/env /bin/python3 /home/fairyroad/.vscode-server
/extensions/ms-python.python-2022.14.0/pythonFiles/lib/python/debugpy/adapters/../../debugpy/launcher 45359 -- /home/fairyroad
/debugger/ex1.py
문자열 입력 : Goodbye Mr. My Holiday
```

맨 끝에 break point를 걸어놓고 run 시키면 다음과 같이 한꺼번에 변수값들이 나옴



마지막 step over를 클릭하면 다음과 같이 결과값을 확인할 수 있음



### Practice3

[문제]

set1={30, 20, 10, 40, 60, 20, 50, 10}, set2={50, 20, 10, 70, 10, 40, 50}, set3={10, 20, 30}과 같이 각각 집합을 생성 후

1. set1과 set2의 교집합, 합집합, 각각의 차집합(2개)을 출력하시오.
2. set3이 set1과 set2의 부분집합인지 판단하시오
3. set2에 원소 30, 60을 추가하고 출력하시오.
4. set1에서 원소 50과 60을 삭제하고 출력하시오.

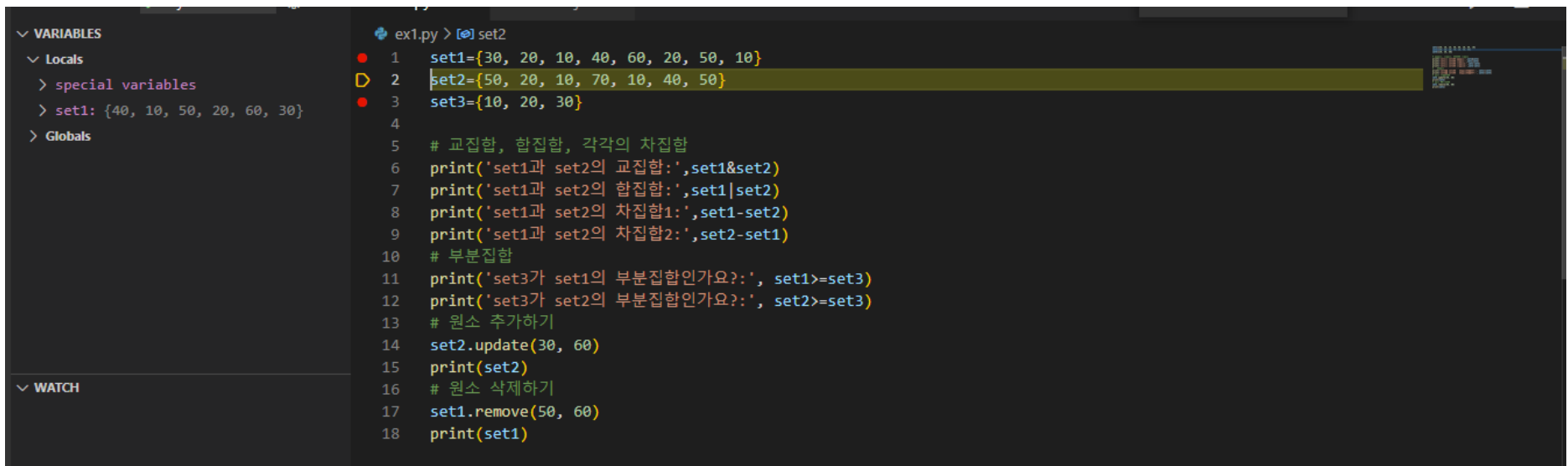
[코드]

```
set1={30, 20, 10, 40, 60, 20, 50, 10}
set2={50, 20, 10, 70, 10, 40, 50}
set3={10, 20, 30}

# 교집합, 합집합, 각각의 차집합
print('set1과 set2의 교집합:',set1&set2)
print('set1과 set2의 합집합:',set1|set2)
print('set1과 set2의 차집합1:',set1-set2)
print('set1과 set2의 차집합2:',set2-set1)
# 부분집합
print('set3가 set1의 부분집합인가?:', set1>=set3)
print('set3가 set2의 부분집합인가?:', set2>=set3)
# 원소 추가하기
set2.update(30, 60)
print(set2)
# 원소 삭제하기
set1.remove(50, 60)
print(set1)
```

## [Debugger]

1. 처음부터 실행시키는데 set1값이 중복되는 것은 잘 걸러서 들어가는 것을 확인할 수 있음

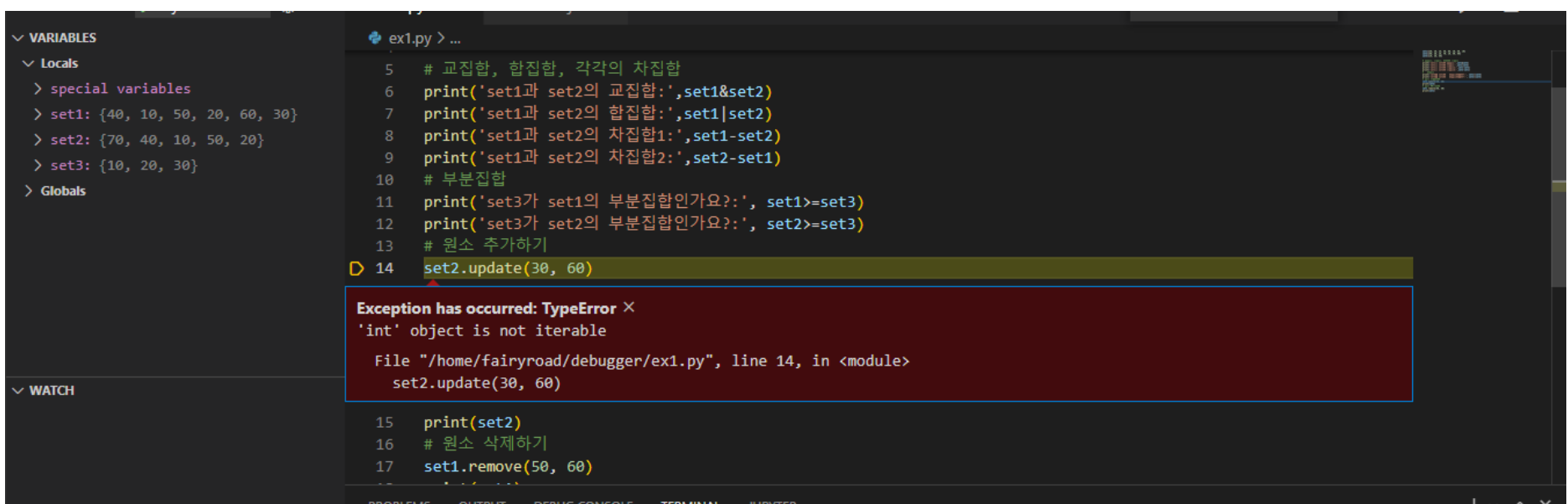


```
ex1.py > [0] set2
1  set1={30, 20, 10, 40, 60, 20, 50, 10}
2  set2={50, 20, 10, 70, 10, 40, 50}
3  set3={10, 20, 30}
4
5  # 교집합, 합집합, 각각의 차집합
6  print('set1과 set2의 교집합:',set1&set2)
7  print('set1과 set2의 합집합:',set1|set2)
8  print('set1과 set2의 차집합1:',set1-set2)
9  print('set1과 set2의 차집합2:',set2-set1)
10 # 부분집합
11 print('set3가 set1의 부분집합인가요?:', set1>=set3)
12 print('set3가 set2의 부분집합인가요?:', set2>=set3)
13 # 원소 추가하기
14 set2.update(30, 60)
15 print(set2)
16 # 원소 삭제하기
17 set1.remove(50, 60)
18 print(set1)
```

2. 하나씩 step over로 실행시키면서 변화를 확인해보기

```
fairyard@DESKTOP-53ORROM:~/debugger$ cd /home/fairyard/debugger ; /usr/bin/env /bin/python3 /home/fairyard/.vscode-server
/extensions/ms-python.python-2022.14.0/pythonFiles/lib/python/debugpy/adapters/.../debugpy/launcher 40231 -- /home/fairyard
/debugger/ex1.py
set1과 set2의 교집합: {40, 10, 50, 20}
set1과 set2의 합집합: {70, 40, 10, 50, 20, 60, 30}
set1과 set2의 차집합1: {60, 30}
set1과 set2의 차집합2: {70}
█
```

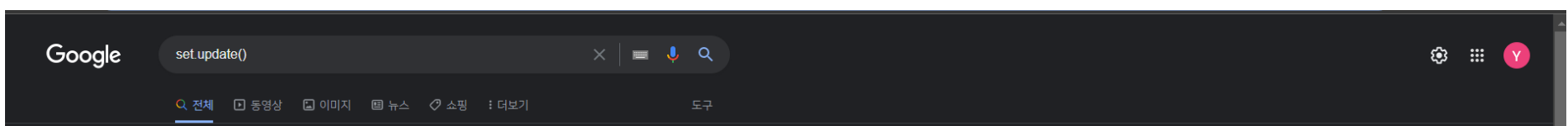
3. 아래와 같이 TypeError 오류가 발생하는 것을 볼 수 있음



```
ex1.py > ...
5  # 교집합, 합집합, 각각의 차집합
6  print('set1과 set2의 교집합:',set1&set2)
7  print('set1과 set2의 합집합:',set1|set2)
8  print('set1과 set2의 차집합1:',set1-set2)
9  print('set1과 set2의 차집합2:',set2-set1)
10 # 부분집합
11 print('set3가 set1의 부분집합인가요?:', set1>=set3)
12 print('set3가 set2의 부분집합인가요?:', set2>=set3)
13 # 원소 추가하기
14 set2.update(30, 60)
15 print(set2)
16 # 원소 삭제하기
17 set1.remove(50, 60)
18
```

Exception has occurred: TypeError <int> object is not iterable  
File "/home/fairyard/debugger/ex1.py", line 14, in <module>  
set2.update(30, 60)

int object로 넣으면 안되겠다는 것을 알게되었음 → 구글링



## 값 여러 개 추가하기(update)

여러 개의 값을 한꺼번에 추가(update)할 때는 다음과 같이 하면 된다.

```
>>> s1 = set([1, 2, 3])
>>> s1.update([4, 5, 6])
>>> s1
{1, 2, 3, 4, 5, 6}
```

update부분을 위와 같이 수정하면 된다는 것을 알게 되었음!

4. 아래와 같이 수정한 후 다시 step over를 해봄

```
5 # 교집합, 합집합, 각각의 차집합
6 print('set1과 set2의 교집합:', set1&set2)
7 print('set1과 set2의 합집합:', set1|set2)
8 print('set1과 set2의 차집합1:', set1-set2)
9 print('set1과 set2의 차집합2:', set2-set1)
10 # 부분집합
11 print('set3가 set1의 부분집합인가?:', set1>=set3)
12 print('set3가 set2의 부분집합인가?:', set2>=set3)
13 # 원소 추가하기
14 set2.update([30, 60])
15 print(set2)
16 # 원소 삭제하기
17 set1.remove(50, 60)
18 print(set1)
```

5. 아래와 같은 에러가 한번 더 발생함

```
9 print('set1과 set2의 차집합2:', set2-set1)
10 # 부분집합
11 print('set3가 set1의 부분집합인가?:', set1>=set3)
12 print('set3가 set2의 부분집합인가?:', set2>=set3)
13 # 원소 추가하기
14 set2.update([30, 60])
15 print(set2)
16 # 원소 삭제하기
17 set1.remove(50, 60)
18 print(set1)
```

Exception has occurred: TypeError ×  
remove() takes exactly one argument (2 given)  
File "/home/fairyroad/debugger/ex1.py", line 17, in <module>  
set1.remove(50, 60)

```
16 # 원소 삭제하기
17 set1.remove([50, 60])
18 print(set1)
```

Exception has occurred: TypeError ×  
unhashable type: 'list'  
File "/home/fairyroad/debugger/ex1.py", line 17, in <module>  
set1.remove([50, 60])

다시 구글링 해보면, 아래처럼 하나씩 remove해야하는 것을 알 수 있음

## 특정 값 제거하기(remove)

특정 값을 제거하고 싶을 때는 다음과 같이 하면 된다.

```
>>> s1 = set([1, 2, 3])
>>> s1.remove(2)
>>> s1
{1, 3}
```

[정답]

```
set1={30, 20, 10, 40, 60, 20, 50, 10}
set2={50, 20, 10, 70, 10, 40, 50}
set3={10, 20, 30}
```

```
# 출력하기
print('set1:', set1, 'set2:', set2)
# 교집합, 합집합, 각각의 차집합
print('set1과 set2의 교집합:', set1&set2)
print('set1과 set2의 합집합:', set1|set2)
print('set1과 set2의 차집합1:', set1-set2)
print('set1과 set2의 차집합2:', set2-set1)
# 부분집합
print('set3가 set1의 부분집합인가요?:', set1>=set3)
print('set3가 set2의 부분집합인가요?:', set2>=set3)
set2.update([30, 60])
print(set2)
# 원소 삭제하기
set1.remove(50)
set1.remove(60)
print(set1)
```

다시 debugging해보면 잘 돌아가는 것을 확인할 수 있음