

# 6주차(2)\_VS Code Debugging 예제

## 예제 1

### 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 = int(input('섭씨 온도를 입력하세요. '))
fahrenheit = (5/9)*celsius+32

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

### 디버깅

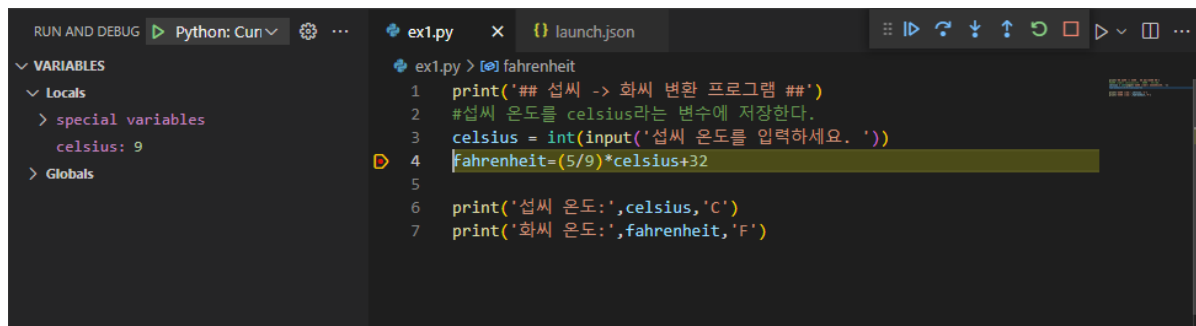
1. 문제가 있어보이는 fahrenheit부분에 break point를 걸어둠  
입력받는 창 : 9를 입력

```

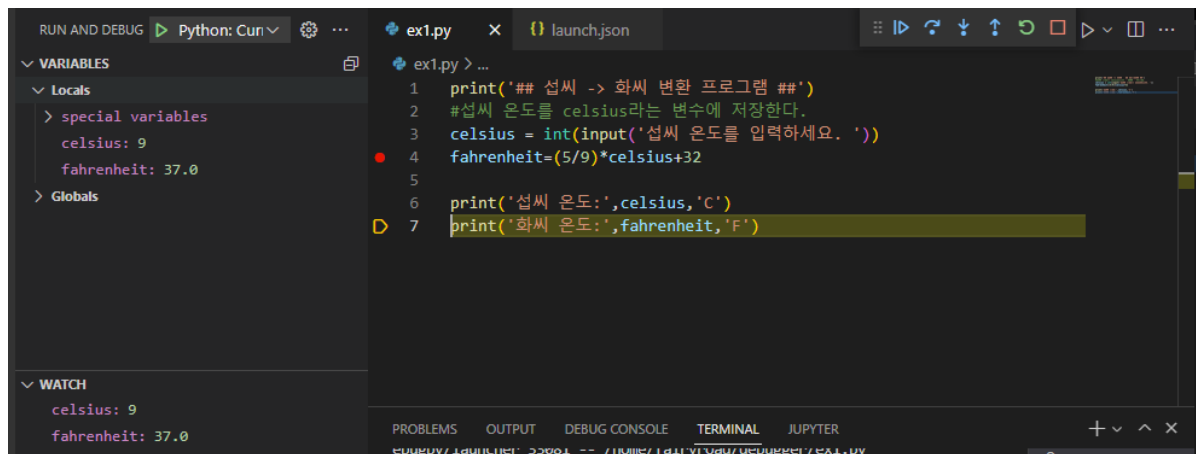
fairyard@DESKTOP-53ORROM:~/debugger$ cd /home/fairyard/debugger ; /
usr/bin/env /bin/python3 /home/fairyard/.vscode-server/extensions/ms-
python.python-2022.14.0/pythonFiles/lib/python/debugpy/adapter/../../d
ebugpy/launcher 35871 -- /home/fairyard/debugger/ex1.py
## 섭씨 -> 화씨 변환 프로그램 ##
섭씨 온도를 입력하세요. 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')
```

## 예제 2

### 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"))

```

## 디버깅

1. 어디가 잘못되었는지 잘 모르겠으면, Test Case를 생각해야함.

Goodbye Mr. My Holiday 가 Goodbye MS. 김민정 으로 바뀌어야 함.

본인의 이름으로 코드 변경해주기

```

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. 김유정"))

```

2. 처음부터 쭉 변수값이 어떻게 변화는 지 확인!

첫줄에 break point를 찍고 step over를 하면서 확인을 해봄

The screenshot shows the VS Code editor with a Python file named `ex1.py`. The script is as follows:

```
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. 김유정"))
```

The left sidebar shows the **VARIABLES** pane with the following content:

```
Locals
  > special variables
    sentence: 'Goodbye Mr. My Holiday'
  > Globals
```

The **WATCH** pane is empty.

아래 부분의 값을 입력해주기 : Goodbye Mr. My Holiday

The screenshot shows a terminal window with the following commands and output:

```
fairyroad@DESKTOP-530RROR:~/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/../../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

The screenshot shows the VS Code editor with the same Python file `ex1.py`. The **VARIABLES** pane shows the following content:

```
Locals
  > special variables
    sentence: 'Goodbye Mr. My Holiday'
    slice1: 'Goodbye Mr. '
    slice2: ' H'
    slice3: 'd'
  > Globals
```

The **WATCH** pane is empty.

The script is running, and an exception has occurred at line 10:

```
Exception has occurred: IndexError ×
string index out of range
File "/home/fairyroad/debugger/ex1.py", line 10, in <module>
    slice4 = sentence[22]
```

## 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. 김윤희"))

```

다시 확인해 보기

```

fairyroad@DESKTOP-530RROM:~/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/../../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

The screenshot shows the VS Code debugger interface. On the left, the 'VARIABLES' pane is expanded to 'Locals', showing the following variables and their values:

- special variables:
  - sentence: 'Goodbye Mr. Hody'
  - slice1: 'Goodbye Mr. '
  - slice2: 'Ho'
  - slice3: 'd'
  - slice4: 'y'
- Globals: (empty)

On the right, the 'WATCH' pane shows the current line of code being executed:

```

19 print(sentence.upper().replace("MR. HODY", "MS. 김윤희"))

```

마지막 step over를 클릭하면 다음과 같이 결과값을 확인할 수 있음

```

GOODBYE MR. 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/adapter/../../debugpy/launcher 45359 -- /home/fairyroad
/debugger/ex1.py
문자열 입력 : Goodbye Mr. My Holiday
Goodbye Mr. My Holiday
Goodbye Mr. Hody
변경된 문자열의 길이: 16
goodbye mr. hody
GOODBYE MR. HODY
GOODBYE MS. 김유정

```

## 예제 3

### 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}

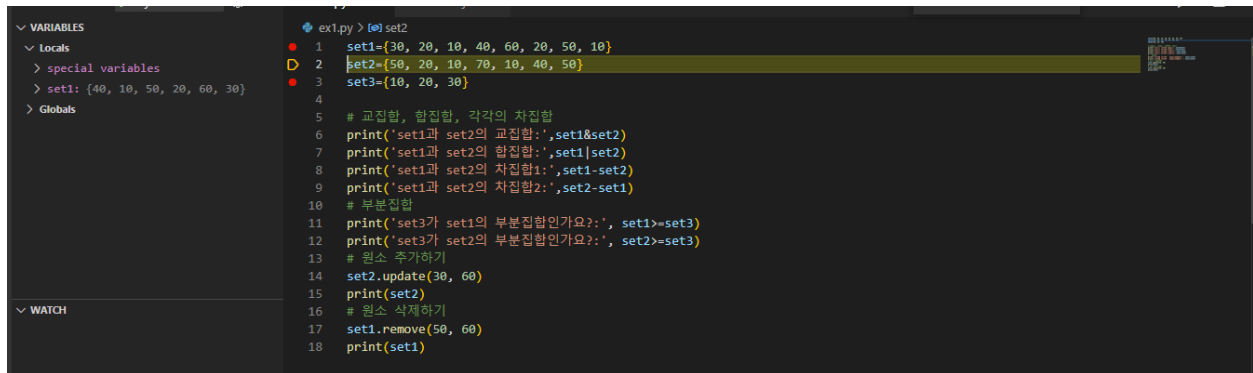
# 교집합, 합집합, 각각의 차집합
print('set1과 set2의 교집합:', set1&set2)
print('set1과 set2의 합집합:', set1|set2)
print('set1과 set2의 차집합1:', set1-set2)
print('set1과 set2의 차집합2:', set2-set1)
# 부분집합
print('set3가 set1의 부분집합인가?: ', set1>=set3)
print('set3가 set2의 부분집합인가?: ', set2>=set3)
# 원소 추가하기
set2.update(30, 60)

```

```
print(set2)
# 원소 삭제하기
set1.remove(50, 60)
print(set1)
```

## 디버깅

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

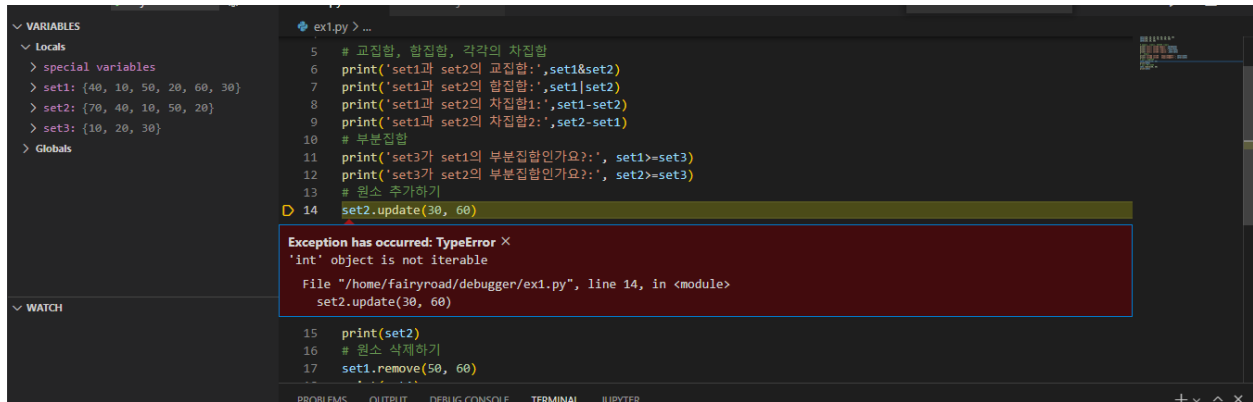


하나씩 step over로 실행시키면서 변화를 확인해보기

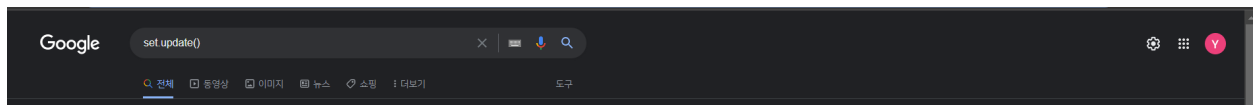
```
fairyroad@DESKTOP-53ORR0M:~/debugger$ cd /home/fairyroad/debugger ; /usr/bin/env /bin/python3 /home/fairyroad/.vscode-server
/extensions/ms-python.python-2022.14.0/pythonFiles/lib/python/debugpy/adapters/../../debugpy/launcher 40231 -- /home/fairyroad
/debugger/ex1.py
set1과 set2의 교집합: {40, 10, 50, 20}
set1과 set2의 합집합: {70, 40, 10, 50, 20, 60, 30}
set1과 set2의 차집합1: {60, 30}
set1과 set2의 차집합2: {70}
[]
```

아래와 같이 TypeError 오류가 발생하는 것을 볼 수 있음





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!

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)

```

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 부분집합인가?: ', set1>=set3)
print('set3가 set2의 부분집합인가?: ', set2>=set3)
set2.update([30, 60])
print(set2)
# 원소 삭제하기
set1.remove(50)
set1.remove(60)
print(set1)
```