14. debugging & scope

* using VSCode

* made by 세현 쌤

debugging?

- 에러를 발견 & 에러 위치를 알려준다
- 에러를 분석해서 더 나은 솔루션을 제공해줌
- 추후 컴퓨터 성능 저하 예방

demo code (1)

```
1   def add(a,b):
2       c = a +b
3       return c
4
5   def main():
6       x = 3
7       y = 5
8       z = add(x,y)
9       print(z)
10
11  if __name__ == '__main__' :
12      main()
13
```

explore debugger

RUN AND ... Python: Curre

VARIABLES

WATCH

CALL STACK

BREAKPOINTS

- Raised Exceptions
- Uncaught Exceptions
- User Uncaught Exceptions

locals & globals

VARIABLES

Locals

```
(return) add: 8
x: 3
y: 5
z: 8
```

Globals

```
> special variables
> function variables
```

오늘의 point!

① locals & globals 차이
② breakpoint 이해

breakpoints

BREAKPOINTS

- Raised Exceptions
- Uncaught Exceptions
- User Uncaught Exceptions
- 1.py C:\Users\harry\OneDrive\Des... 8

executing debugger

Q. Line 7 breakpoint 빨간 점 놓고 Debug Python File 실행하기

```
7    y = 5
```

Q. 명령줄의 step over 클릭하고 결과 비교해보기

```
7    y = 5
8    z = add(x,y)
```

executing debugger

Q. 명령줄의 step over 한 번 더 클릭하고 결과 비교해보기

```
7    y = 5
8    z = add(x,y)
9    print(z)
```

! 즉 line 7, 8, 9에 각각 breakpoint를 놓았을 때의 진행과정 점검

executing debugger

결과 확인하기

```
VARIABLES
  Locals
    x: 3
  Globals
    special variables
    function variables
      add: <function add at 0x03AB4F58>
      main: <function main at 0x03AB4FA0>
```

```
VARIABLES
  Locals
    x: 3
    y: 5
  Globals
    special variables
    function variables
      add: <function add at 0x03AB4F58>
      main: <function main at 0x03AB4FA0>
```

```
VARIABLES
  Locals
    (return) add: 8
    x: 3
    y: 5
    z: 8
  Globals
    special variables
    function variables
      add: <function add at 0x03AB4F58>
      main: <function main at 0x03AB4FA0>
```

demo code (2)

```python
# Python Program to Print Even Numbers from 1 to N

maximum = int(input(" Please Enter the Maximum Value : "))

for number in range(1, maximum+1):
    if(number % 2 == 0):
        answer = number
        print("{0}".format(answer))

```

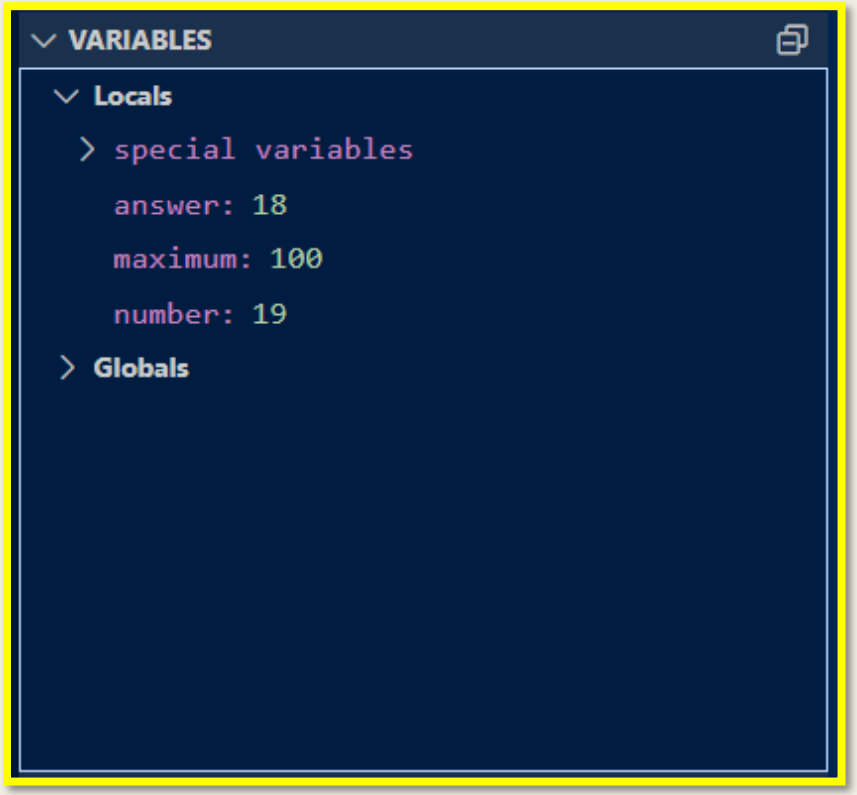
executing debugger

Q. Line 5에 breakpoint 설정하고
maximum을 100으로 설정한 다음,
step over를 계속 돌리자

```
5   for number in range(1, maximum+1):
6       if(number % 2 == 0):
7           answer = number
8           print("{0}".format(answer))
```

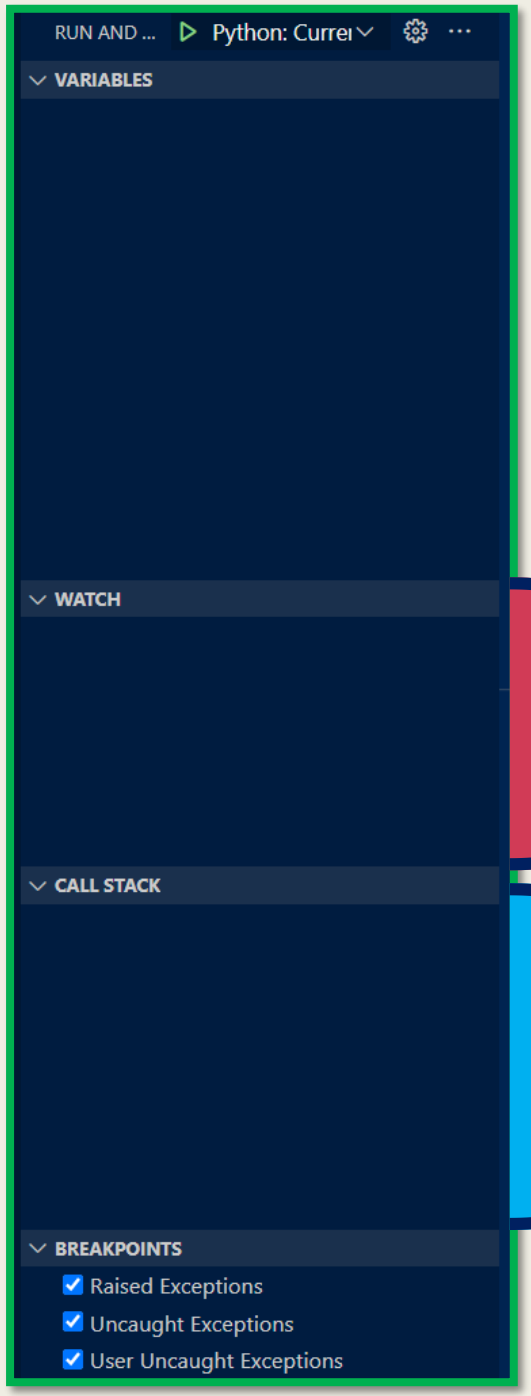
executing debugger

Q. Step over를 진행하면서 local variable answer, maximum, number 값의 변화를 확인!



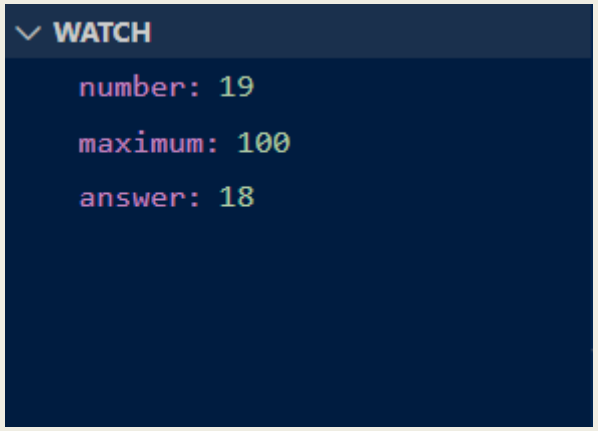
* watch에 변수명을 입력해서도 값을 알 수 있다. (+call stack도 확인)

! 실제 for loop이 계속 돌아가는 동안 변수 값의 변화를 눈으로 확인할 수 있다!



executing debugger

watch

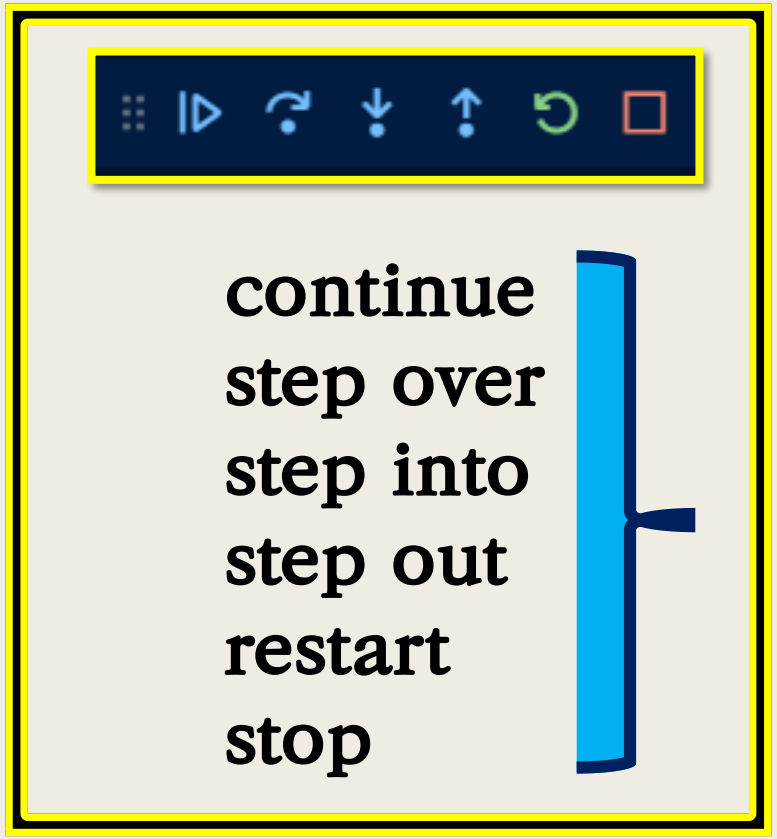


call stack

오늘의 point!

③ watch 이해
④ call stack 이해

executing debugger



오늘의 point!

⑤ step execution 이해

! 앞선 예제에서 각 기능들 실습 해보기

```python
import random

def average(d,n):
    avg = d/n
    return avg

def running_average(numbers):
    avgs = []
    total = 0
    for i, num in enumerate(numbers):
        total += num
        current_avg = average(total, i)
        avgs.append(current_avg)

    return avgs

def highest_running_average(numbers):
    averages = running_average(numbers)
    return max(averages)

def generate_numbers(n):
    return random.sample(range(1,300), n)

if __name__ == "__main__":
    nums = generate_numbers(100)
    print('Numbers: ', nums)
    print('Running Average: ', running_average(nums))
    print('Highest Average: ', highest_running_average(nums))
```

도전과제

Q. VSCode python debugger를 이용 - 각종 step execution 기능 및 앞서 배운 모든 개념들을 활용하여 해당 code의 error message 내용을 파악하고 code를 고쳐보자.

* error message 확인

도전과제

```
1   import random
2
3   def average(d,n):
4       avg = d/n
```

Exception has occurred: ZeroDivisionError (note: full exception trace is shown but execution is paused at: <module>)

```
division by zero
  File "C:\Users\harry\OneDrive\Desktop\ch14_ex3.py", line 4, in average
    avg = d/n
  File "C:\Users\harry\OneDrive\Desktop\ch14_ex3.py", line 12, in running_average
    current_avg = average(total, i)
  File "C:\Users\harry\OneDrive\Desktop\ch14_ex3.py", line 27, in <module> (Current frame)
    print('Running Average: ', running_average(nums))
```

```
5       return avg
6
7   def running_average(numbers):
8       avgs = []
9       total = 0
10      for i, num in enumerate(numbers):
11          total += num
12          current_avg = average(total, i)
13          avgs.append(current_avg)
14
15      return avgs
16
17  def highest_running_average(numbers):
18      averages = running_average(numbers)
19      return max(averages)
```

scope? – demo code (3)

Message는 local일까? global일까? – 출력결과 확인!

```
1   message = "Hello"
2
3   def say():
4       print("say:mesage = "+message)
5       obj_id = id(message)
6       print("say:id(message)={0:d}".format(obj_id))
7
8   def main():
9       say()
10      print("main:message="+message)
11      obj_id = id(message)
12      print("say:id(message)={0:d}".format(obj_id))
13
14  if __name__ == "__main__":
15      main()
```

scope? – demo code (4)

Say() 내의 message 변수 선언! – 출력결과 확인!

```
1   message = "Hello"
2
3   def say():
4       message = "Hi"
5       print("say:mesage = "+message)
6       obj_id = id(message)
7       print("say:id(message)={0:d}".format(obj_id))
8
9   def main():
10      say()
11      print("main:message="+message)
12      obj_id = id(message)
13      print("say:id(message)={0:d}".format(obj_id))
14
15  if __name__ == "__main__":
16      main()
```

Q. 지역과 전역변수 구분해보자

scope? – demo code (5)

global keyword 선언 – 달라진 출력결과 확인!

```
1   message = "Hello"
2
3   def say():
4       global message
5       message = "Hi"
6       print("say:mesage = "+message)
7       obj_id = id(message)
8       print("say:id(message)={0:d}".format(obj_id))
9
10  def main():
11      say()
12      print("main:message="+message)
13      obj_id = id(message)
14      print("say:id(message)={0:d}".format(obj_id))
15
16  if __name__ == "__main__":
17      main()
```