Jupyter notebook에서 디버깅 하기

모두의연구소

박은수 Research Director

pdb

- 기본적인 파이썬 디버거로 Jupyter notebook에서도 동작합니다

기본 코드

```
1  def modulabs_hi(x):
2      answer = "Hi~ "
3      answer += x
4      return answer
5
6  print(modulabs_hi("Eunsoo"))
```

```
Hi~ Eunsoo
```

pdb

- pdb 사용 방법 :
  - 먼저 임포트

```
import pdb
```

  - 멈추길 원하는 위치에서

```
pdb.set_trace()
```

pdb

- 적용 예시

```
1  def modulabs_hi(x):
2      answer = "Hi~ "
3      import pdb; pdb.set_trace()
4      answer += x
5      return answer
6
7  print(modulabs_hi("Eunsoo"))
```

pdb

- 적용 예시

In [*]:

```
1  def modulabs_hi(x):
2      answer = "Hi~ "
3      import pdb; pdb.set_trace()
4      answer += x
5      return answer
6
7  print(modulabs_hi("Eunsoo"))
```

```
(Pdb)
```

```
> <ipython-input-12-998c342c98fc>(4)modulabs_hi()
-> answer += x
```

1) answer += x 를 수행할 차례임을 보여줌

pdb

- 적용 예시

In [*]:

```
1  def modulabs_hi(x):
2      answer = "Hi~ "
3      import pdb; pdb.set_trace()
4      answer += x
5      return answer
6
7  print(modulabs_hi("Eunsoo"))
```

```
(Pdb)
```

2) 여기에 디버깅 관련 명령어를 입력 함

```
> <ipython-input-12-998c342c98fc>(4)modulabs_hi()
-> answer += x
```

pdb

- 적용 예시

In [*]:

```
1  def modulabs_hi(x):
2      answer = "Hi~ "
3      import pdb; pdb.set_trace()
4      answer += x
5      return answer
6
7  print(modulabs_hi("Eunsoo"))
```

```
> <ipython-input-12-998c342c98fc>(4)modulabs_hi()
-> answer += x
(Pdb) n
> <ipython-input-12-998c342c98fc>(5)modulabs_hi()
-> return answer
(Pdb)
```

1) n 을 입력한 모습 -> 한 줄 실행 임

2) 다음 실행 할 줄

pdb

Python Debugger Cheatsheet

Getting started

start pdb from within a script:
import pdb;pdb.set_trace()
start pdb from the commandline:
python -m pdb <file.py>

Basics

h(elp)	print available commands
h(elp) ***command***	print help about *command*
q(uit)	quit debugger

Examine

p(rint) ***expr***	print the value of *expr*
pp ***expr***	pretty-print the value of *expr*
w(here)	print current position (including stack trace)
l(ist)	list 11 lines of code around the current line
l(ist) ***first***, ***last***	list from *first* to *last* line number
a(rgs)	print the args of the current function

Movement

<ENTER>	repeat the last command
n(ext)	execute the current statement (step over)
s(tep)	execute and step into function
r(eturn)	continue execution until the current function returns
c(ontinue)	continue execution until a breakpoint is encountered
u(p)	move one level up in the stack trace
d(own)	move one level down in the stack trace

Breakpoints

b(reak)	show all breakpoints
b(reak) ***lineno***	set a breakpoint at *lineno*
b(reak) ***func***	set a breakpoint at the first line of a *func*

Manipulation

!*stmt* treat *stmt* as a Python statement instead of a pdb command

참고 : https://appletree.or.kr/quick_reference_cards/Python/Python%20Debugger%20Cheatsheet.pdf

pdb

- 적용 예시

In [*]:

```
1  def modulabs_hi(x):
2      answer = "Hi~ "
3      import pdb; pdb.set_trace()
4      answer += x
5      return answer
6
7  print(modulabs_hi("Eunsoo"))
```

```
> <ipython-input-12-998c342c98fc>(4)modulabs_hi()
-> answer += x
(Pdb) n
> <ipython-input-12-998c342c98fc>(5)modulabs_hi()
-> return answer
(Pdb)
```

1) n 을 입력한 모습 -> 한 줄 실행 임

2) 다음 실행 할 줄

pdb

- 적용 예시

In [*]:

```
1  def modulabs_hi(x):
2      answer = "Hi~ "
3      import pdb; pdb.set_trace()
4      answer += x
5      return answer
6
7  print(modulabs_hi("Eunsoo"))
```

```
> <ipython-input-12-998c342c98fc>(4)modulabs_hi()
-> answer += x
(Pdb) n
> <ipython-input-12-998c342c98fc>(5)modulabs_hi()
-> return answer

(Pdb) answer
```

← 변수 값을 볼 수 있고 연산도 가능

pdb

- 적용 예시

In [*]:

```
1  def modulabs_hi(x):
2      answer = "Hi~ "
3      import pdb; pdb.set_trace()
4      answer += x
5      return answer
6
7  print(modulabs_hi("Eunsoo"))
```

```
> <ipython-input-12-998c342c98fc>(4)modulabs_hi()
-> answer += x
(Pdb) n
> <ipython-input-12-998c342c98fc>(5)modulabs_hi()
-> return answer
(Pdb) answer
'Hi~ Eunsoo'
(Pdb)
```

변수 값을 볼 수 있고 연산도 가능

pdb

- 적용 예시

```
In [*]:
1  def modulabs_hi(x):
2      answer = "Hi~ "
3      import pdb; pdb.set_trace()
4      answer += x
5      return answer
6
7  print(modulabs_hi("Eunsoo"))
```

```
> <ipython-input-13-998c342c98fc>(4)modulabs_hi()
-> answer += x
(Pdb) next
> <ipython-input-13-998c342c98fc>(5)modulabs_hi()
-> return answer

(Pdb) answer + "Welcome to ModuLABS~~~!!"
```

pdb

- 적용 예시

```
In [*]:
1  def modulabs_hi(x):
2      answer = "Hi~ "
3      import pdb; pdb.set_trace()
4      answer += x
5      return answer
6
7  print(modulabs_hi("Eunsoo"))
```

```
> <ipython-input-13-998c342c98fc>(4)modulabs_hi()
-> answer += x
(Pdb) next
> <ipython-input-13-998c342c98fc>(5)modulabs_hi()
-> return answer
(Pdb) answer + "Welcome to ModuLABS~~~!!"
'Hi~ EunsooWelcome to ModuLABS~~~!!'

(Pdb) quit
```

종료

- 적용 예시

In [*]:

```
1  def modulabs_hi(x):
2      import pdb; pdb.set_trace()
3      answer = "Hi~ "
4      answer += x
5      return answer
6
7  print(modulabs_hi("Eunsoo"))
```

```
> <ipython-input-14-441c0ae19d05>(3)modulabs_hi()
-> answer = "Hi~ "
(Pdb) b 4
Breakpoint 2 at <ipython-input-14-441c0ae19d05>:4
(Pdb) b
Num Type         Disp Enb   Where
1   breakpoint   keep yes   at <ipython-input-3-d127aa185ba4>:6
2   breakpoint   keep yes   at <ipython-input-14-441c0ae19d05>:4
(Pdb) c
> <ipython-input-14-441c0ae19d05>(4)modulabs_hi()
-> answer += x

(Pdb)
```

1) 4번째 줄에 break point

2) Break point 보기

3) Break point까지 수행

- 적용 예시

```
In [*]:
1  def modulabs_hi(x):
2      import pdb; pdb.set_trace()
3      answer = "Hi~ "
4      answer += x
5      return answer
6
7  print(modulabs_hi("Eunsoo"))
```

```
> <ipython-input-14-441c0ae19d05>(3)modulabs_hi()
-> answer = "Hi~ "
(Pdb) b 4
Breakpoint 2 at <ipython-input-14-441c0ae19d05>:4
(Pdb) b
Num Type         Disp Enb   Where
1   breakpoint   keep yes   at <ipython-input-3-d127aa185ba4>:6
2   breakpoint   keep yes   at <ipython-input-14-441c0ae19d05>:4
(Pdb) c
> <ipython-input-14-441c0ae19d05>(4)modulabs_hi()
-> answer += x
(Pdb)
```

1) 4번째 줄에 break point

2) Break point 보기

3) Break point까지 수행

4) 4번째 줄로 이동 됨

이것만 잘 알면 됨

Python Debugger Cheatsheet

Getting started

start pdb from within a script:
import pdb;pdb.set_trace()
start pdb from the commandline:
python -m pdb <file.py>

Basics

h(elp)	print available commands
h(elp) ***command***	print help about *command*
q(uit)	quit debugger

Examine

p(rint) ***expr***	print the value of *expr*
pp ***expr***	pretty-print the value of *expr*
w(here)	print current position (including stack trace)
l(ist)	list 11 lines of code around the current line
l(ist) ***first*****, *last***	list from *first* to *last* line number
a(rgs)	print the args of the current function

Movement

<ENTER>	repeat the last command
n(ext)	execute the current statement (step over)
s(tep)	execute and step into function
r(eturn)	continue execution until the current function returns
c(ontinue)	continue execution until a breakpoint is encountered
u(p)	move one level up in the stack trace
d(own)	move one level down in the stack trace

Breakpoints

b(reak)	show all breakpoints
b(reak) ***lineno***	set a breakpoint at *lineno*
b(reak) ***func***	set a breakpoint at the first line of a *func*

Manipulation

!*stmt* treat *stmt* as a Python statement instead of a pdb command

IPython.core.debugger.set_trace

- 사용 방법 :
  - 먼저 임포트

```
from IPython.core.debugger import set_trace
```

  - 멈추길 원하는 위치에서

```
set_trace()
```

pdb와 동일

Syntax Highlight 지원

In [15]:
```
from IPython.core.debugger import set_trace
```

In [*]:
```
def modulabs_hi(x):
    answer = "Hi~ "
    set_trace()
    answer += x
    return answer

print(modulabs_hi("Eunsoo"))
```

```
> <ipython-input-17-178787abc98d>(4)modulabs_hi()
      2     answer = "Hi~ "
      3     set_trace()
----> 4     answer += x
      5     return answer
      6

ipdb>
```

특히
이것만 잘
알면 됨

Python Debugger Cheatsheet

Getting started

start pdb from within a script:
import pdb;pdb.set_trace()
start pdb from the commandline:
python -m pdb <file.py>

Basics

h(elp)	print available commands
h(elp) ***command***	print help about *command*
q(uit)	quit debugger

Examine

p(rint) ***expr***	print the value of *expr*
pp ***expr***	pretty-print the value of *expr*
w(here)	print current position (including stack trace)
l(ist)	list 11 lines of code around the current line
l(ist) ***first*****, *last***	list from *first* to *last* line number
a(rgs)	print the args of the current function

Movement

<ENTER>	repeat the last command
n(ext)	execute the current statement (step over)
s(tep)	execute and step into function
r(eturn)	continue execution until the current function returns
c(ontinue)	continue execution until a breakpoint is encountered
u(p)	move one level up in the stack trace
d(own)	move one level down in the stack trace

Breakpoints

b(reak)	show all breakpoints
b(reak) ***lineno***	set a breakpoint at *lineno*
b(reak) ***func***	set a breakpoint at the first line of a *func*

Manipulation

!*stmt* treat *stmt* as a Python statement instead of a pdb command

pdb와 동일

Syntax Highlight 지원

```
In [15]:
1 from IPython.core.debugger import set_trace

In [*]:
1 def modulabs_hi(x):
2     answer = "Hi~ "
3     set_trace()
4     answer += x
5     return answer
6
7 print(modulabs_hi("Eunsoo"))

> <ipython-input-17-178787abc98d>(4)modulabs_hi()
      2     answer = "Hi~ "
      3     set_trace()
----> 4     answer += x
      5     return answer
      6

ipdb>
```

%pdb 로 사용하기

- 예시코드

In [31]:
```
1  def welcomeModulabs(name):
2      hi = "Hi "
3      welcome = ", Welcome to Modulabs"
4      print(hi + name + welcome)
```

In [32]:
```
1  welcomeModulabs('Eunsoo')
```

```
Hi Eunsoo, Welcome to Modulabs
```

%pdb 로 사용하기

- %pdb 적용 : 에러가 발생할 때만 동작 함

```
In [40]:
1  %pdb
2  def welcomeModulabs(name):
3      hi = "Hi "
4      welcome = ", Welcome to Modulabs"
5      print(hi + name + welcome)

Automatic pdb calling has been turned ON
```

Automatic pdb calling이 ON 상태라고 나옴

%pdb 로 사용하기

- %pdb 적용 : 에러가 발생할 때만 동작 함

```
In [33]:
1  %pdb
2  def welcomeModulabs(name):
3      hi = "Hi "
4      welcome = ", Welcome to Modulabs"
5      print(hi + name + welcome)

Automatic pdb calling has been turned OFF

In [34]:
1  welcomeModulabs('Eunsoo')

Hi Eunsoo, Welcome to Modulabs
```

에러가 없으면
자동으로 OFF
상태가 됨

%pdb 로 사용하기

- %pdb 적용 : 에러가 발생할 때만 동작 함

```
In [43]:
1  %pdb
2  def welcomeModulabs(name):
3      hi = "Hi "
4      welcome = ", Welcome to Modulabs"
5      print(hi + name + welcome)

Automatic pdb calling has been turned ON
```

다시 실행하여 ON 상태로 둔 뒤

%pdb 로 사용하기

- 에러가 발생 하는 줄에서 멈춤
- 그 이후로는 ipdb 임

```
In [43]:
1 %pdb
2 def welcomeModulabs(name):
3     hi = "Hi "
4     welcome = ", Welcome to Modulabs"
5     print(hi + name + welcome)

Automatic pdb calling has been turned ON

In [*]:
1 welcomeModulabs(10)

---------------------------------------------------------------
TypeError                                 Traceback (most
<ipython-input-44-3a03407de066> in <module>()
----> 1 welcomeModulabs(10)

<ipython-input-43-f8524d43c25a> in welcomeModulabs(name)
      3     hi = "Hi "
      4     welcome = ", Welcome to Modulabs"
----> 5     print(hi + name + welcome)

TypeError: must be str, not int

ipdb>
```

%%debug 로 사용하기

- 셀 시작부터 디버그가 시작 됨
- 그 이후로는 ipdb 임

```
In [62]:
1 def welcomeModulabs(name):
2     hi = "Hi "
3     welcome = ", Welcome to Modulabs"
4     print(hi + name + welcome)

In [*]:
1 %%debug
2 name = 'Eunsoo'
3 name += ' Park'
4 welcomeModulabs(name)
None
> <string>(2)<module>()                    <-- 2번째 줄

ipdb> n                                    <-- next
None
> <string>(3)<module>()                    <-- 3번째 줄

ipdb> n                                    <-- next
None
> <string>(4)<module>()                    <-- 4번째 줄

ipdb> s                                    <-- Step in
--Call--
> <ipython-input-62-1751971e3559>(1)welcomeModulabs()
----> 1 def welcomeModulabs(name):         <-- 함수 진입
      2     hi = "Hi "
      3     welcome = ", Welcome to Modulabs"
      4     print(hi + name + welcome)

ipdb>
```

Python 3.7 부터는 breakpoint() 명령어로 pdb를 실행 시킬 수 있다고 합니다

```
def bad_function(var):
    breakpoint()
    return var + 0

bad_function("Mike")
```

Visual Debugger : PixieDebugger

- 설치

  pip install pixiedust

- 시작할 때

```
In [1]: 1 import pixiedust

        Pixiedust database opened successfully
```

Debugging

In [2]:
```
%%pixie_debugger
import random
def find_max (values):
    max = 0
    for val in values:
        if val > max:
            max = val
    return max
find_max(random.sample(range(100), 10))
```

디버그 할 셀에 추가

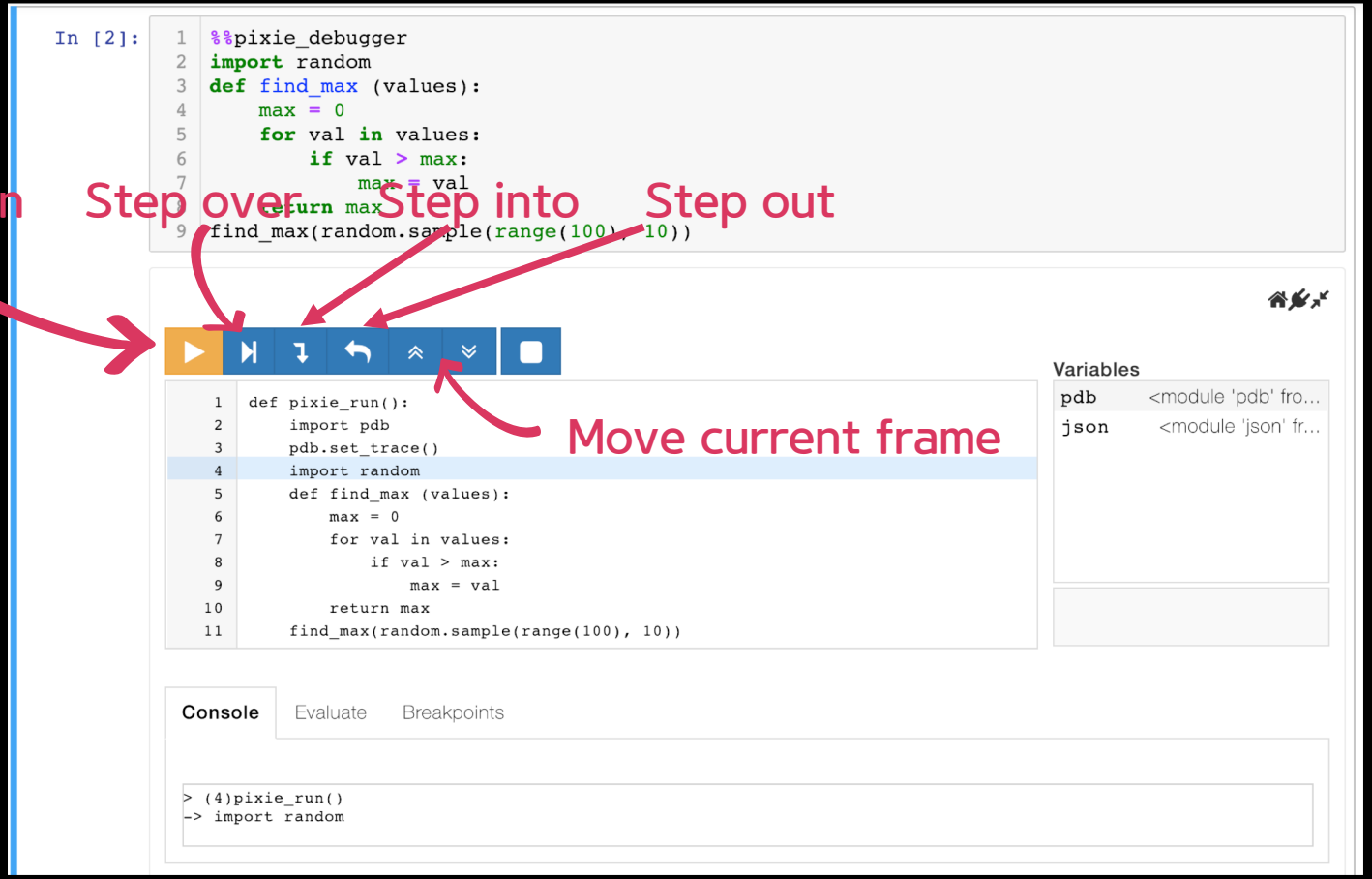
Debugging

In [2]:
```
%%pixie_debugger
import random
def find_max (values):
    max = 0
    for val in values:
        if val > max:
            max = val
    return max
find_max(random.sample(range(100), 10))
```

디버그 할 셀에 추가

실행하면

Debugging


In [2]:
%%pixie_debugger
import random
def find_max (values):
max = 0
for val in values:
if val > max:
max = val
return max
find_max(random.sample(range(100), 10))
Resume execution
Step over
Step into
Step out
Move current frame
def pixie_run():
import pdb
pdb.set_trace()
import random
def find_max (values):
max = 0
for val in values:
if val > max:
max = val
return max
find_max(random.sample(range(100), 10))
Variables
pdb <module 'pdb' fro...
json <module 'json' fr...
Console
Evaluate
Breakpoints
> (4)pixie_run()
-> import random

Debugging

In [2]:

```
1  %%pixie_debugger
2  import random
3  def find_max (values):
4      max = 0
5      for val in values:
6          if val > max:
7              max = val
8      return max
9  find_max(random.sample(range(100), 10))
```

Variables

```
1  def pixie_run():
2      import pdb
3      pdb.set_trace()
4      import random
5      def find_max (values):
6          max = 0
7          for val in values:
8              if val > max:
9                  max = val
10         return max
11     find_max(random.sample(range(100), 10))
```

Variables

pdb	<module 'pdb' fro...
json	<module 'json' fr...

Console | Evaluate | Breakpoints

```
> (4)pixie_run()
-> import random
```

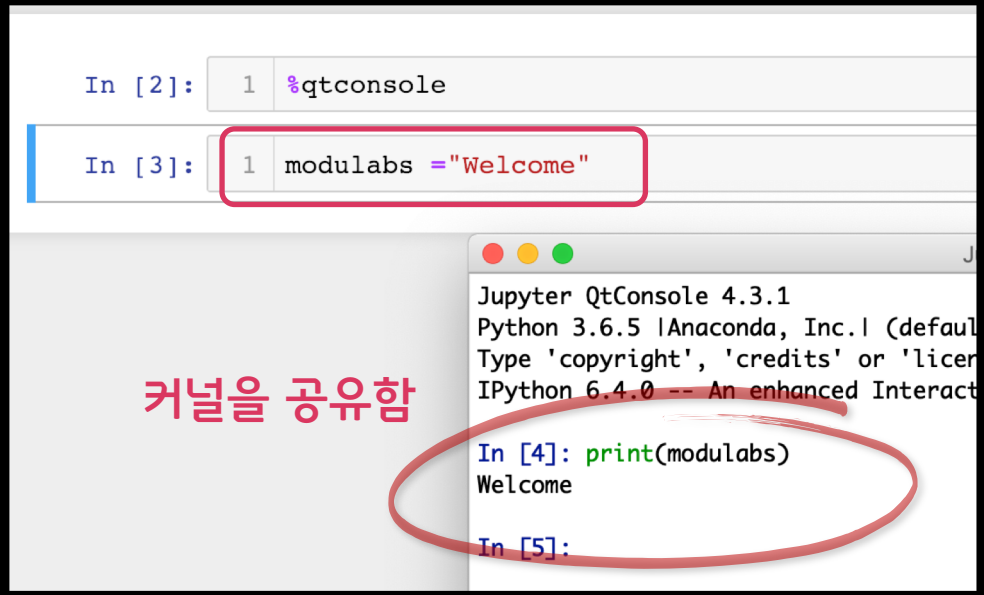
Tip

- Jupyter notebook과 qtconsole연결하기

%qtconsole

Tip

- Jupyter notebook과 qtconsole연결하기

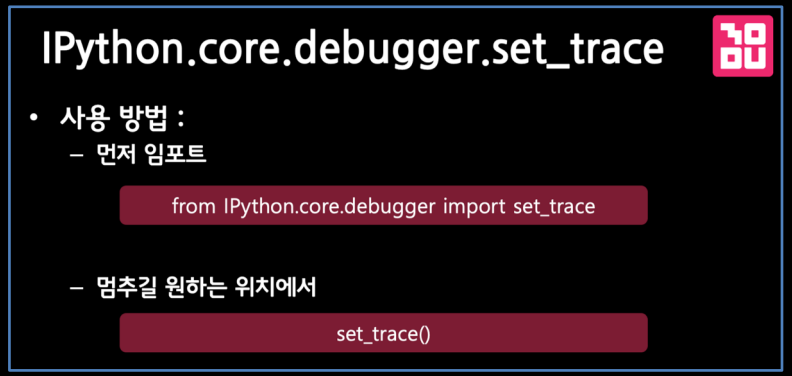


디버깅 혹은 코딩 시 편리

종종 꺼내 쓰면 편리 (개인적으로 맘에 듦)

TLDR

- Visual Debugger는 아직 버그가 좀 있는 것 같습니다
- Jupyter notebook 에서 debugging은 용도에 따라 쓰임이 다름
- 그냥 set_trace() 방식이 가장 일반적인 것 같음



참고

- [1] Jupyter Notebook Debugging : https://www.blog.pythonlibrary.org/2018/10/17/jupyter-notebook-debugging/
- [2] **Debugging Jupyter notebooks :** https://davidhamann.de/2017/04/22/debugging-jupyter-notebooks/
- **[3] Python Debugger Cheatsheet** https://appletree.or.kr/quick_reference_cards/Python/Python%20Debugger%20Cheatsheet.pdf
- **[4]** The Visual Python Debugger for Jupyter Notebooks You've Always Wanted : https://medium.com/ibm-watson-data-lab/the-visual-python-debugger-for-jupyter-notebooks-youve-always-wanted-761713babc62

박 은 수 Research Director

E-mail : es.park@modulabs.co.kr