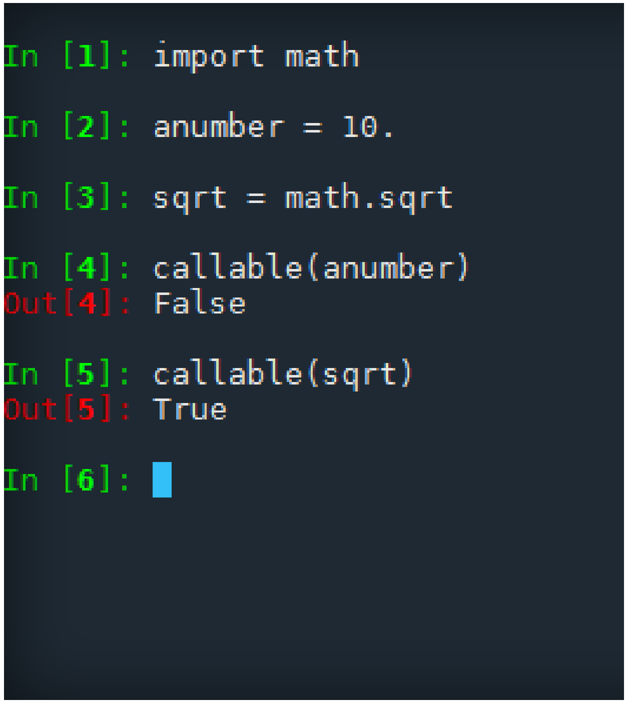
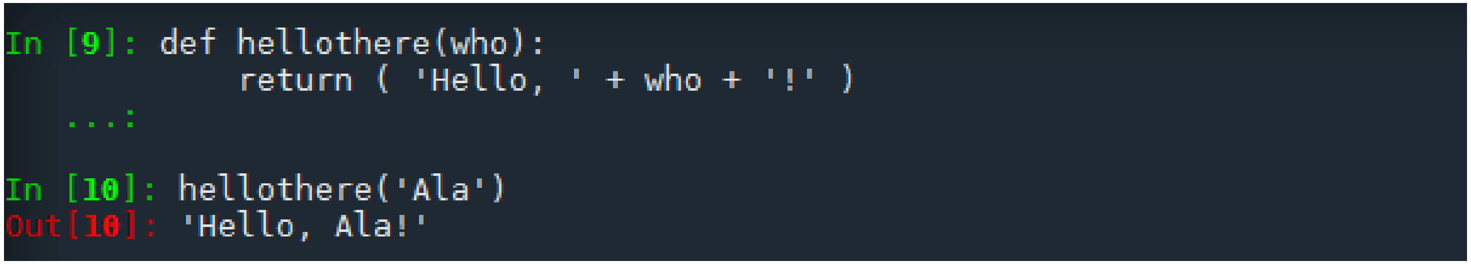
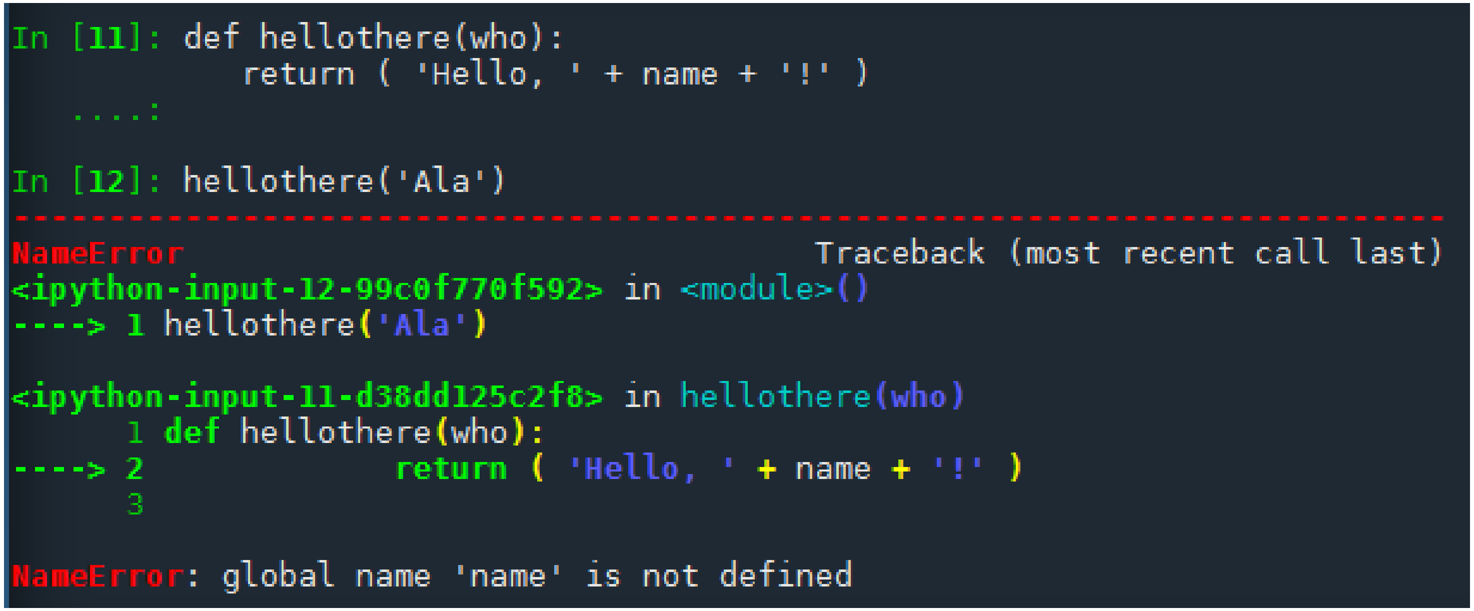


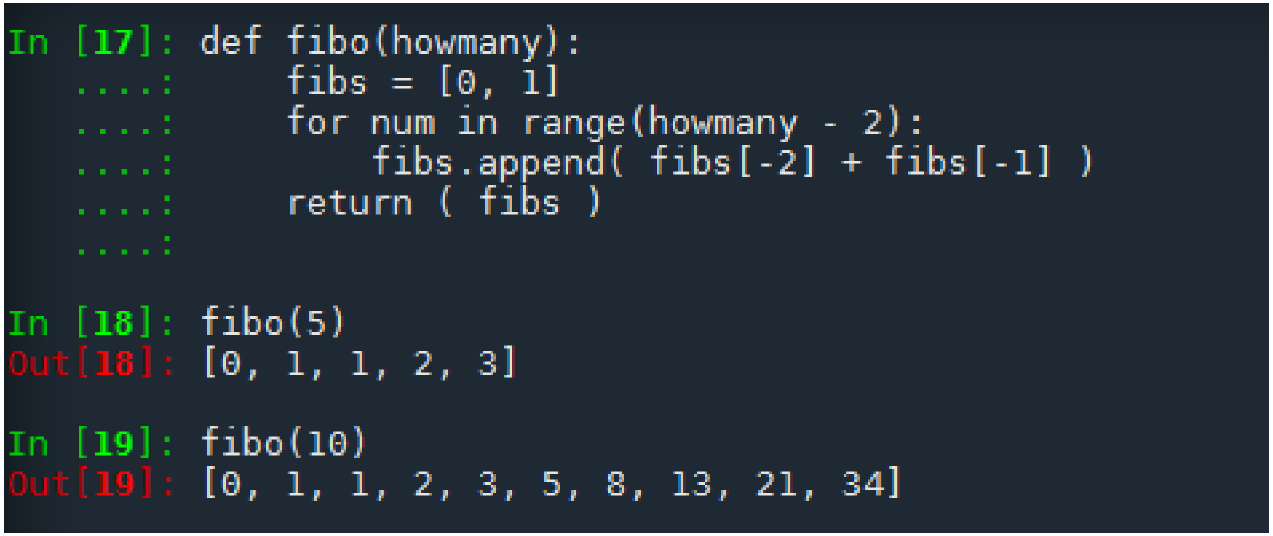
```
In [1]: import math

In [2]: anumber = 10.

In [3]: sqrt = math.sqrt

In [4]: callable(anumber)
Out[4]: False

In [5]: callable(sqrt)
Out[5]: True

In [6]: 
```



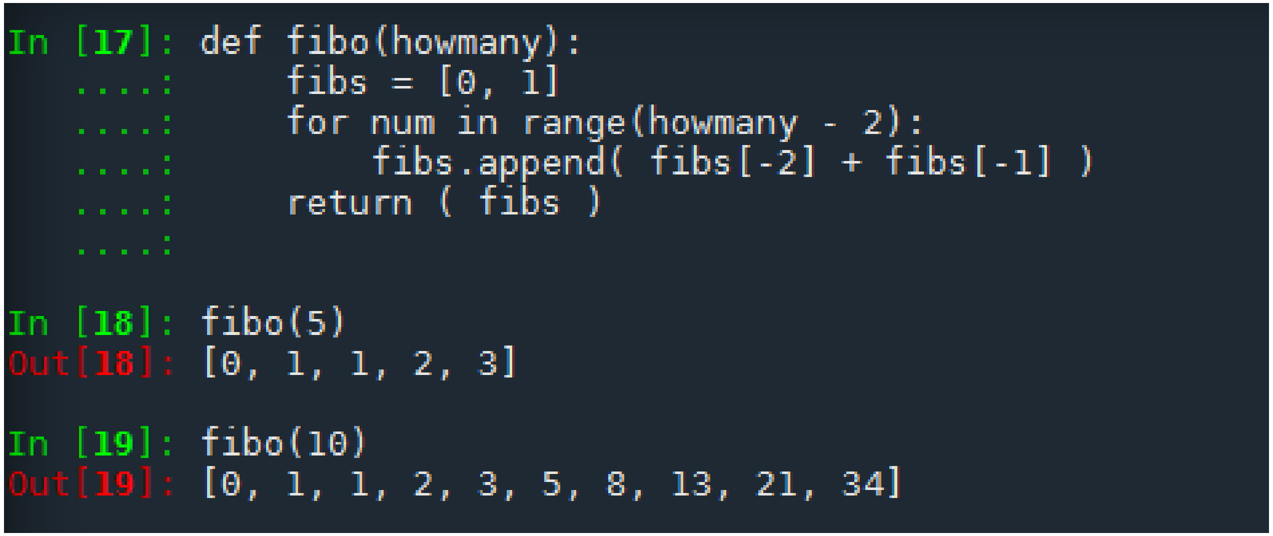
```
In [9]: def hellothere(who):
            return ( 'Hello, ' + who + '!' )
   ...:

In [10]: hellothere('Ala')
Out[10]: 'Hello, Ala!'
```



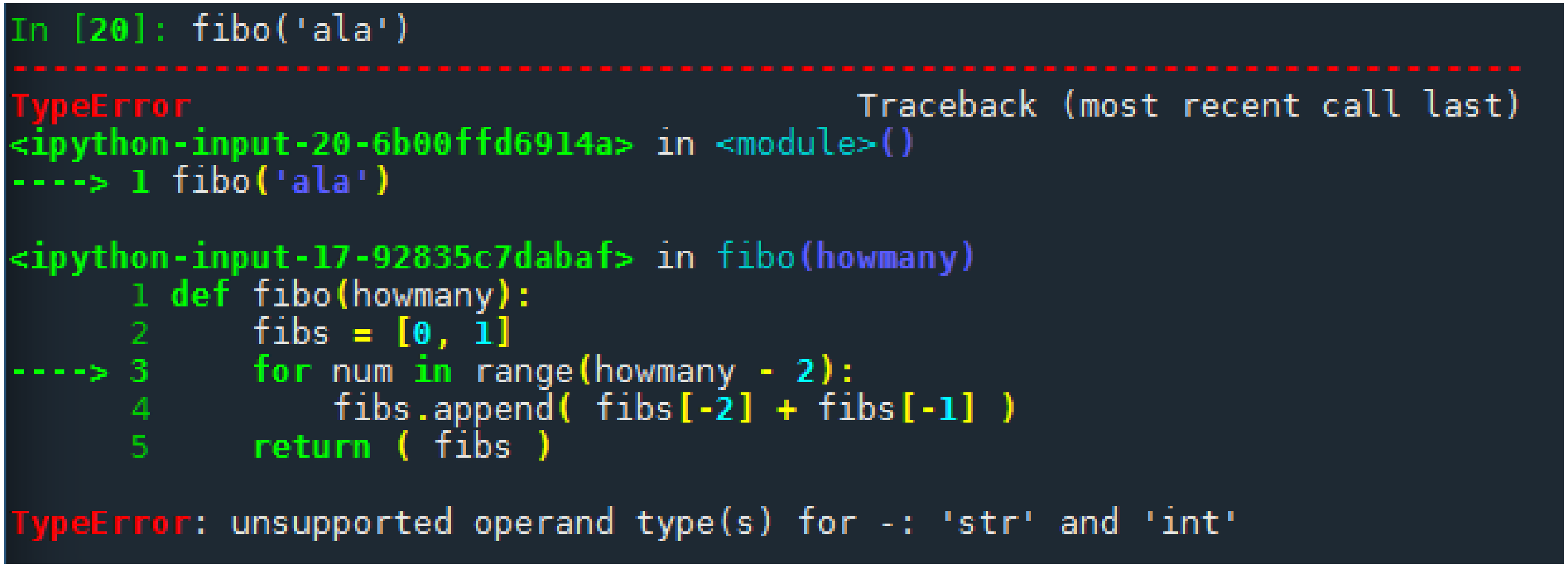
```
In [11]: def hellothere(who):
             return ( 'Hello, ' + name + '!' )
    ....:

In [12]: hellothere('Ala')
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
<ipython-input-12-99c0f770f592> in <module>()
----> 1 hellothere('Ala')

<ipython-input-11-d38dd125c2f8> in hellothere(who)
      1 def hellothere(who):
----> 2             return ( 'Hello, ' + name + '!' )
      3

NameError: global name 'name' is not defined
```



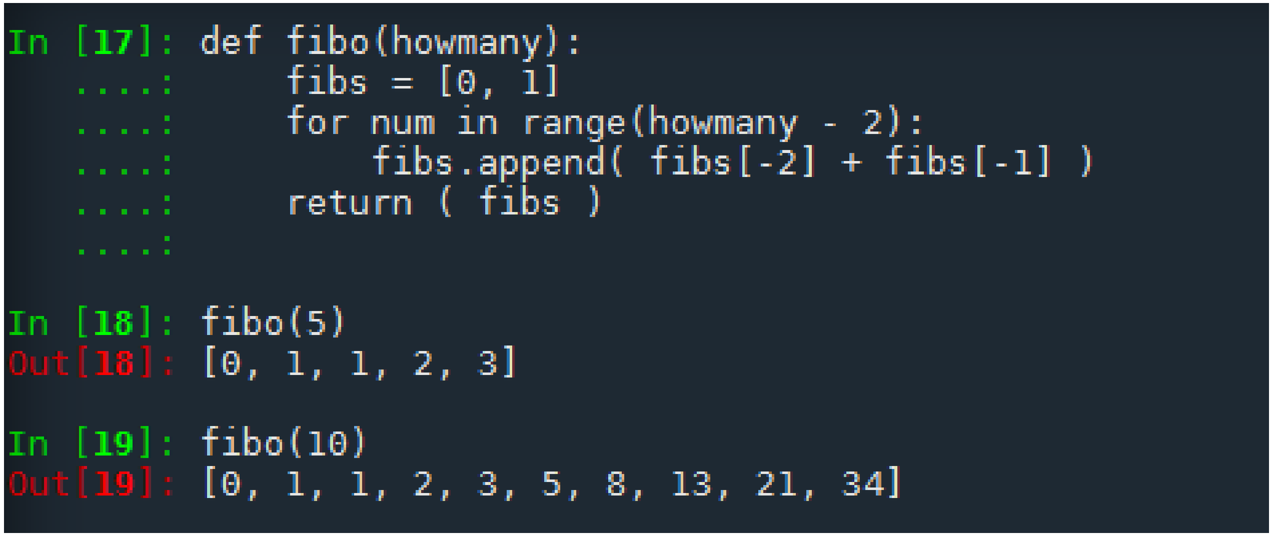
```
In [17]: def fibo(howmany):
    ....:     fibs = [0, 1]
    ....:     for num in range(howmany - 2):
    ....:         fibs.append( fibs[-2] + fibs[-1] )
    ....:     return ( fibs )
    ....:

In [18]: fibo(5)
Out[18]: [0, 1, 1, 2, 3]

In [19]: fibo(10)
Out[19]: [0, 1, 1, 2, 3, 5, 8, 13, 21, 34]
```



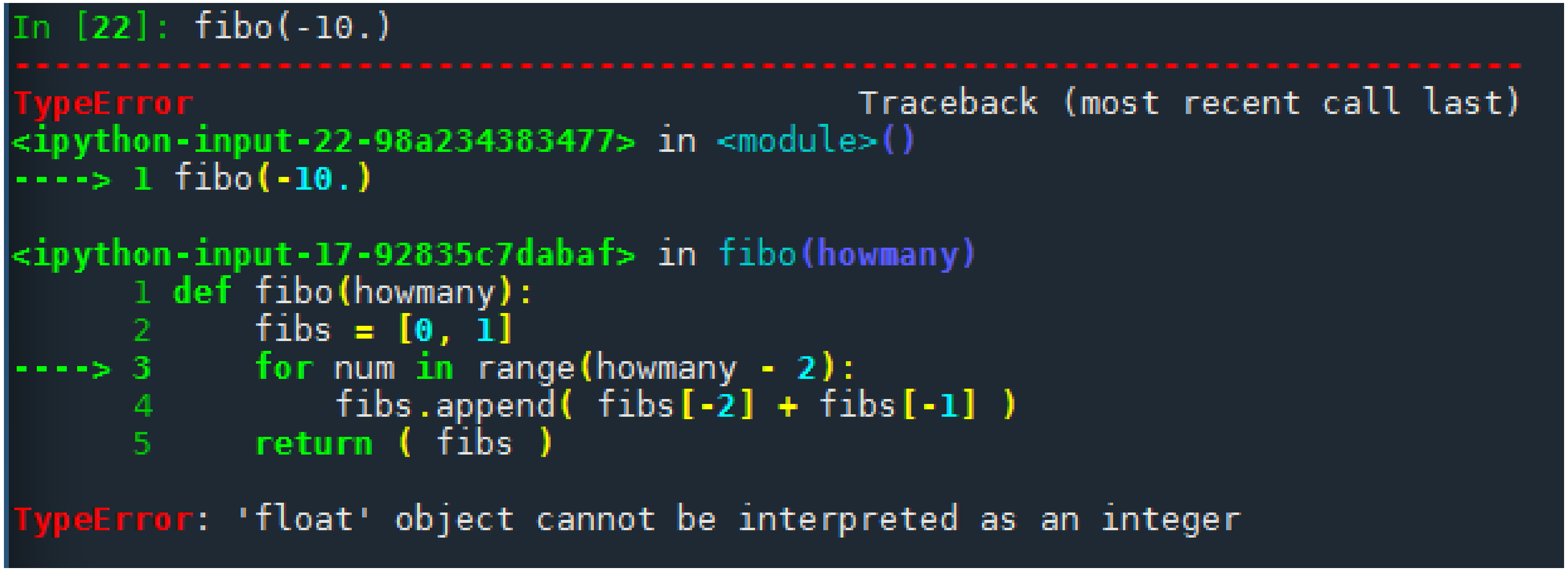
```
In [17]: def fibo(howmany):
    ....:     fibs = [0, 1]
    ....:     for num in range(howmany - 2):
    ....:         fibs.append( fibs[-2] + fibs[-1] )
    ....:     return ( fibs )
    ....:

In [18]: fibo(5)
Out[18]: [0, 1, 1, 2, 3]

In [19]: fibo(10)
Out[19]: [0, 1, 1, 2, 3, 5, 8, 13, 21, 34]
```



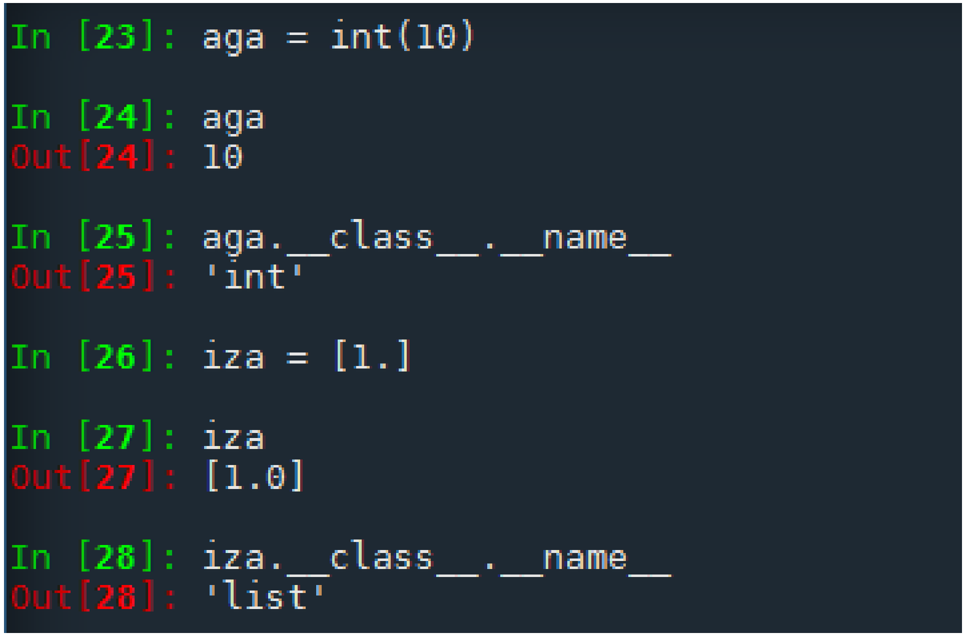
```
In [20]: fibo('ala')
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
<ipython-input-20-6b00ffd6914a> in <module>()
----> 1 fibo('ala')

<ipython-input-17-92835c7dabaf> in fibo(howmany)
      1 def fibo(howmany):
      2     fibs = [0, 1]
----> 3     for num in range(howmany - 2):
      4         fibs.append( fibs[-2] + fibs[-1] )
      5     return ( fibs )

TypeError: unsupported operand type(s) for -: 'str' and 'int'
```



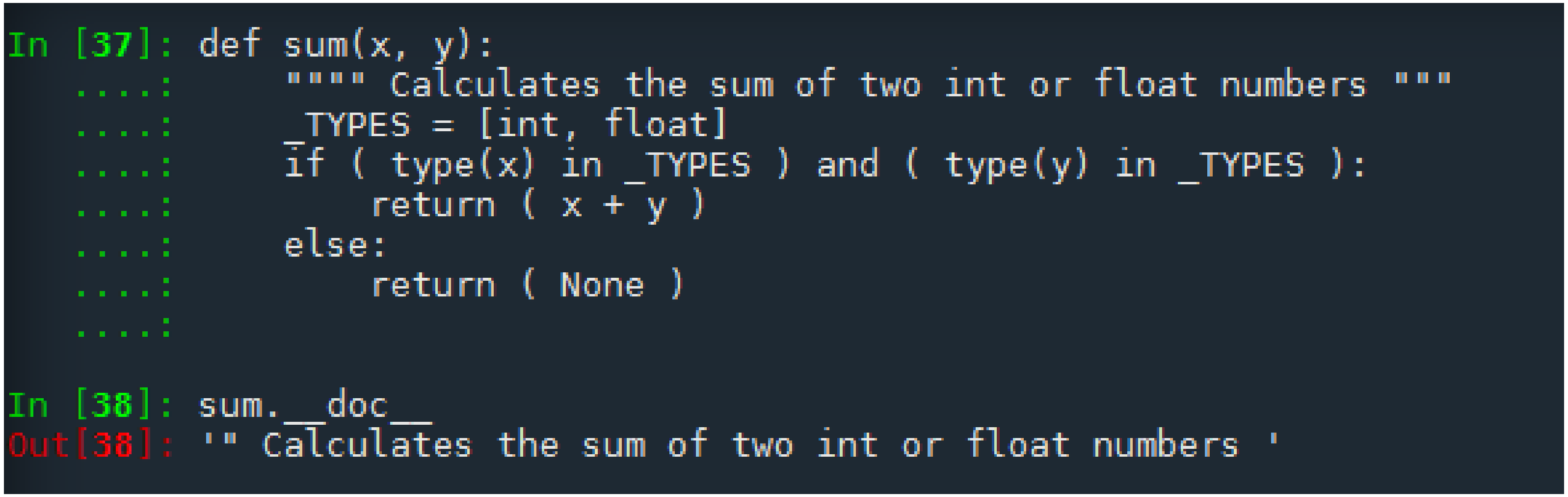
```
In [17]: def fibo(howmany):
    ....:     fibs = [0, 1]
    ....:     for num in range(howmany - 2):
    ....:         fibs.append( fibs[-2] + fibs[-1] )
    ....:     return ( fibs )
    ....:

In [18]: fibo(5)
Out[18]: [0, 1, 1, 2, 3]

In [19]: fibo(10)
Out[19]: [0, 1, 1, 2, 3, 5, 8, 13, 21, 34]
```



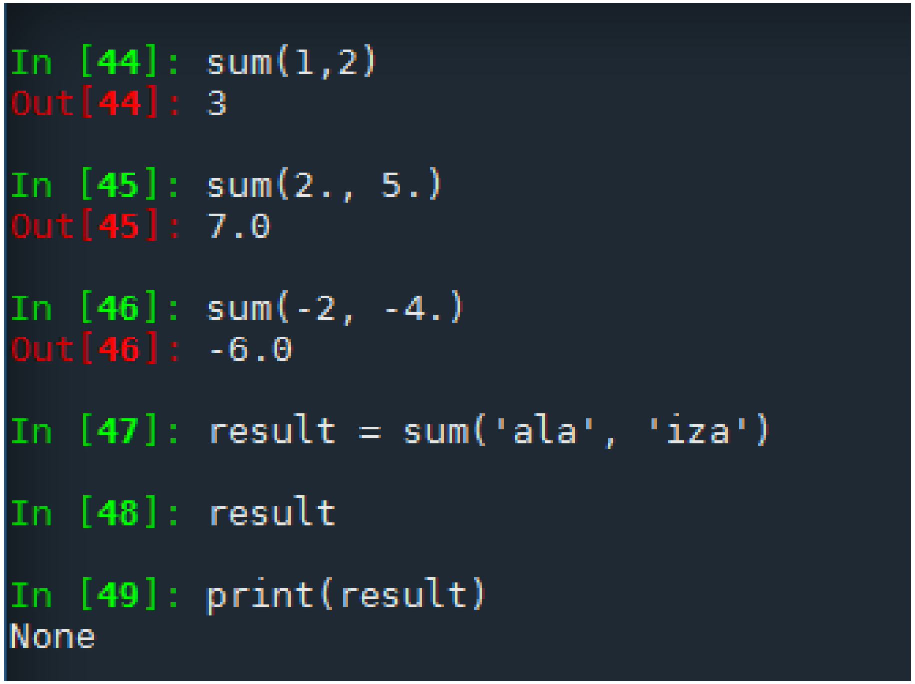
```
In [22]: fibo(-10.)
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
<ipython-input-22-98a234383477> in <module>()
----> 1 fibo(-10.)

<ipython-input-17-92835c7dabaf> in fibo(howmany)
      1 def fibo(howmany):
      2     fibs = [0, 1]
----> 3     for num in range(howmany - 2):
      4         fibs.append( fibs[-2] + fibs[-1] )
      5     return ( fibs )

TypeError: 'float' object cannot be interpreted as an integer
```



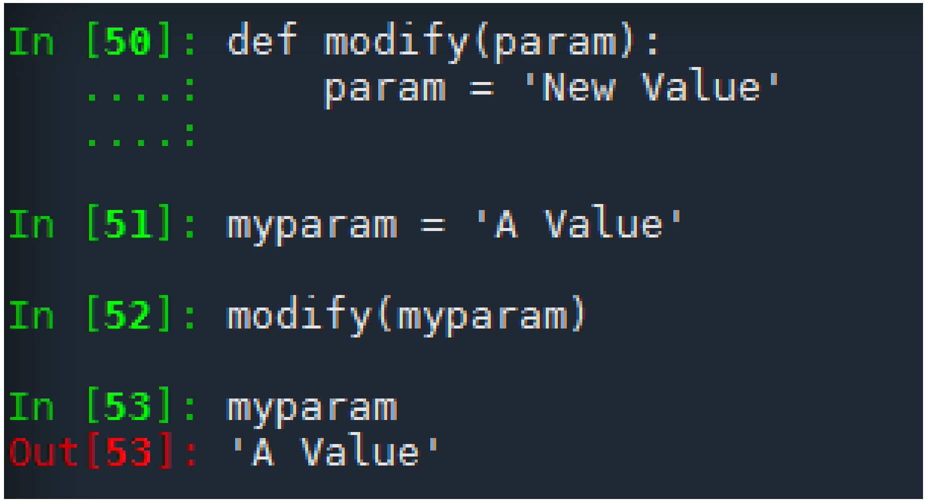
```
In [23]: aga = int(10)

In [24]: aga
Out[24]: 10

In [25]: aga.__class__.__name__
Out[25]: 'int'

In [26]: iza = [1.]

In [27]: iza
Out[27]: [1.0]

In [28]: iza.__class__.__name__
Out[28]: 'list'
```



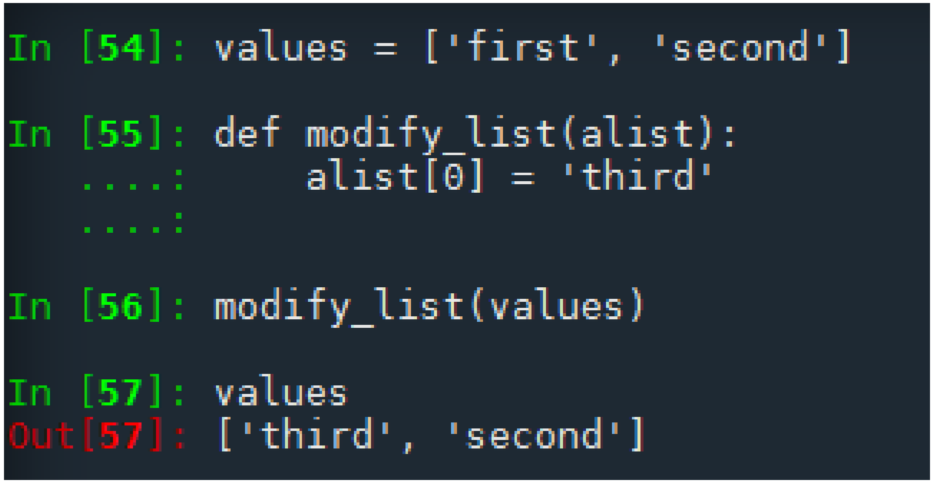
```
In [37]: def sum(x, y):
    ....:     """" Calculates the sum of two int or float numbers """
    ....:     _TYPES = [int, float]
    ....:     if ( type(x) in _TYPES ) and ( type(y) in _TYPES ):
    ....:         return ( x + y )
    ....:     else:
    ....:         return ( None )
    ....:

In [38]: sum.__doc__
Out[38]: '" Calculates the sum of two int or float numbers '
```



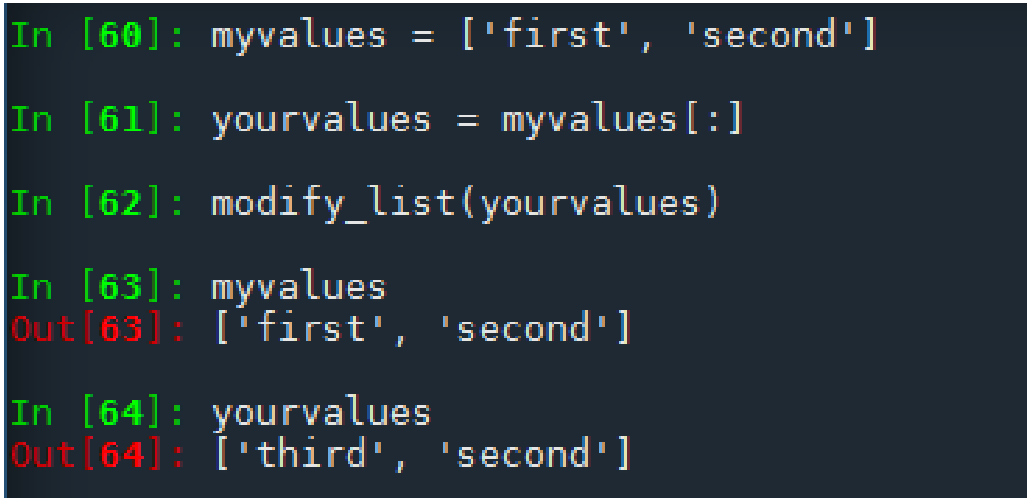
```
In [44]: sum(1,2)
Out[44]: 3

In [45]: sum(2., 5.)
Out[45]: 7.0

In [46]: sum(-2, -4.)
Out[46]: -6.0

In [47]: result = sum('ala', 'iza')

In [48]: result

In [49]: print(result)
None
```



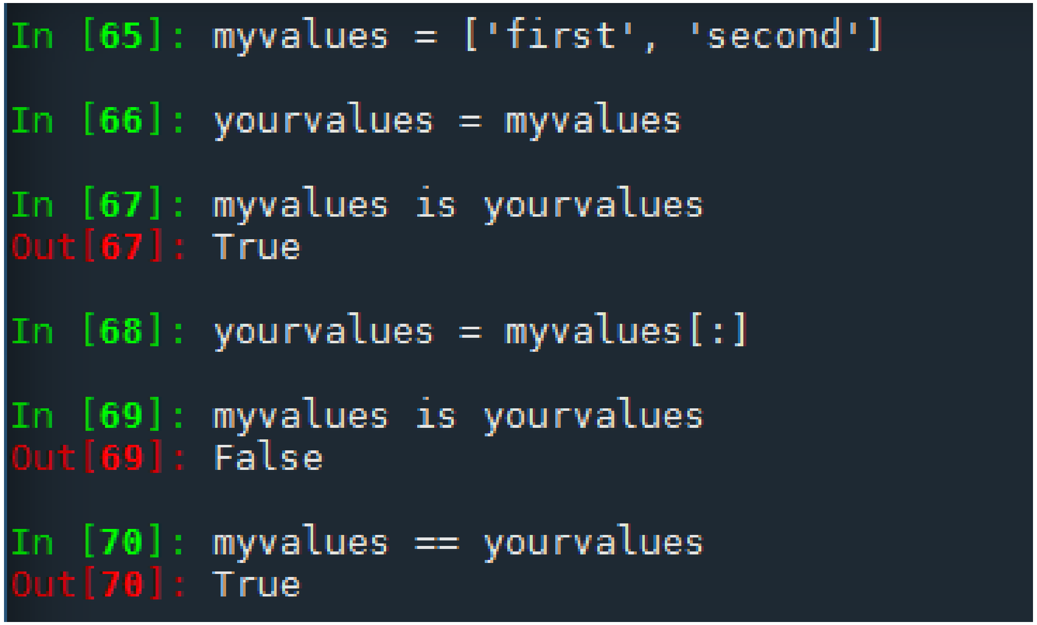
```
In [50]: def modify(param):
    ....:     param = 'New Value'
    ....:

In [51]: myparam = 'A Value'

In [52]: modify(myparam)

In [53]: myparam
Out[53]: 'A Value'
```



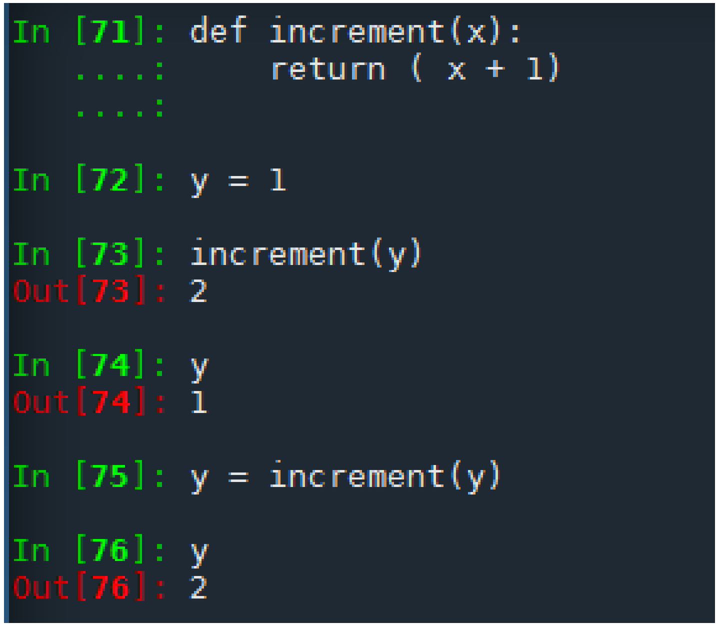
```
In [54]: values = ['first', 'second']

In [55]: def modify_list(alist):
    ....:     alist[0] = 'third'
    ....:

In [56]: modify_list(values)

In [57]: values
Out[57]: ['third', 'second']
```



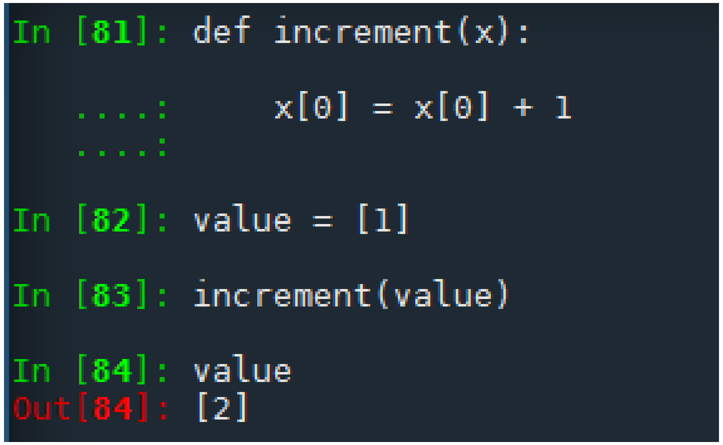
```
In [60]: myvalues = ['first', 'second']

In [61]: yourvalues = myvalues[:]

In [62]: modify_list(yourvalues)

In [63]: myvalues
Out[63]: ['first', 'second']

In [64]: yourvalues
Out[64]: ['third', 'second']
```



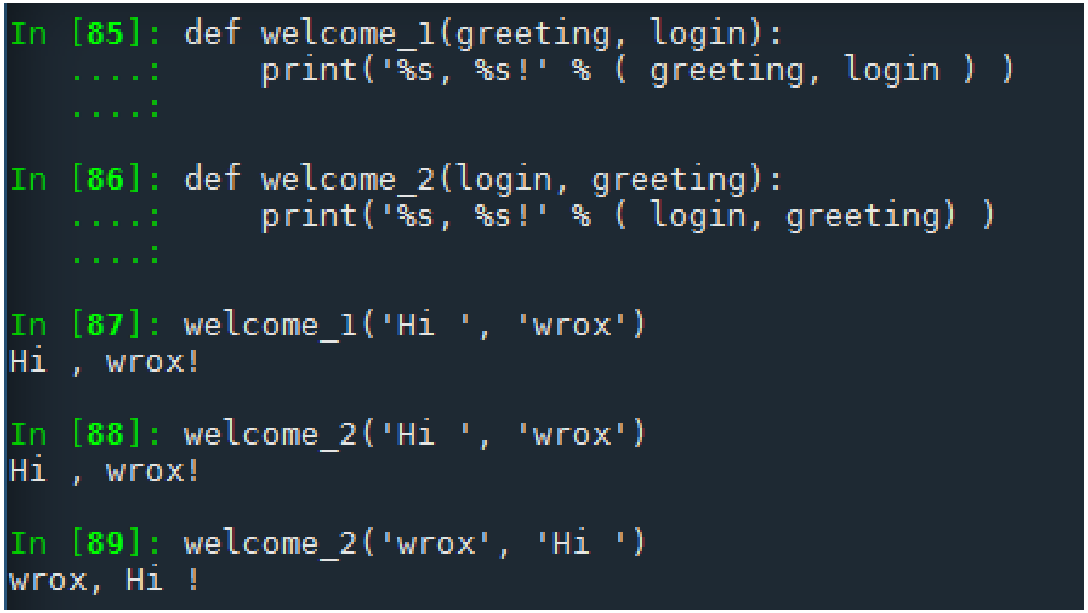
```
In [65]: myvalues = ['first', 'second']

In [66]: yourvalues = myvalues

In [67]: myvalues is yourvalues
Out[67]: True

In [68]: yourvalues = myvalues[:]

In [69]: myvalues is yourvalues
Out[69]: False

In [70]: myvalues == yourvalues
Out[70]: True
```



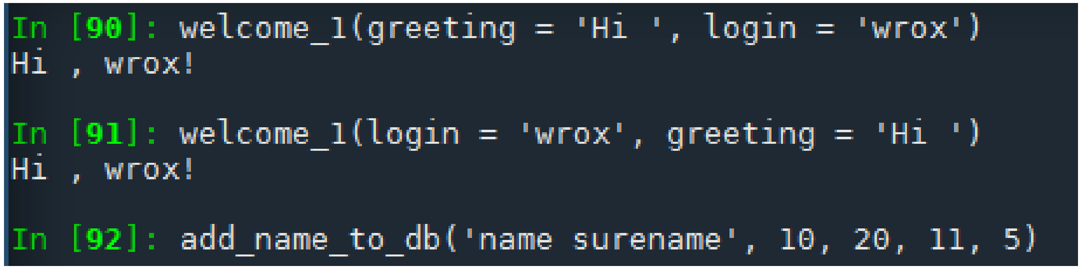
```
In [71]: def increment(x):
    ....:     return ( x + 1)
    ....:

In [72]: y = 1

In [73]: increment(y)
Out[73]: 2

In [74]: y
Out[74]: 1

In [75]: y = increment(y)

In [76]: y
Out[76]: 2
```



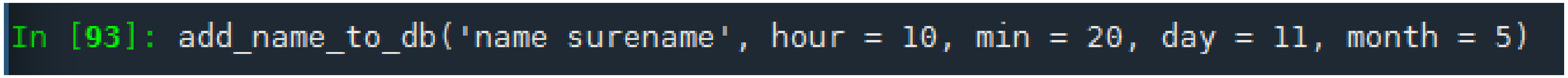
```
In [81]: def increment(x):

    ....:     x[0] = x[0] + 1
    ....:

In [82]: value = [1]

In [83]: increment(value)

In [84]: value
Out[84]: [2]
```



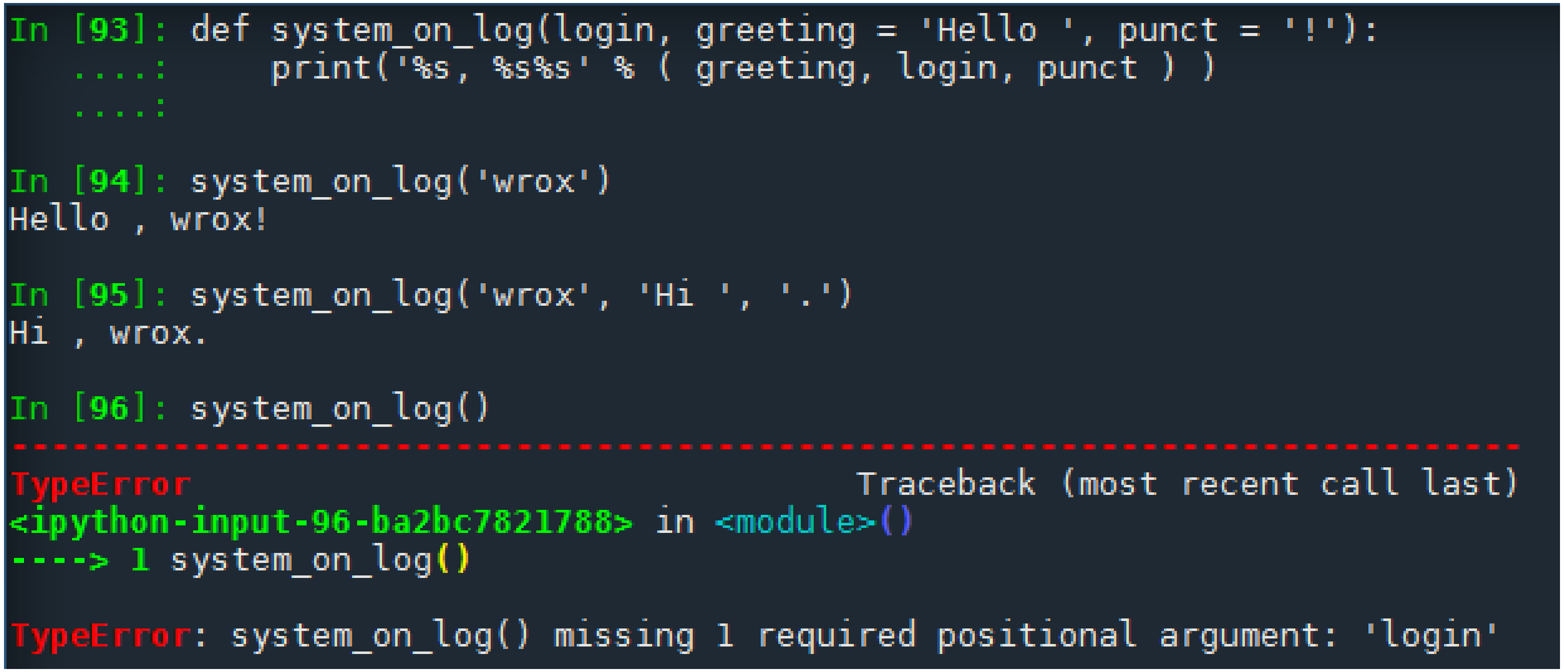
```
In [85]: def welcome_1(greeting, login):
    ....:     print('%s, %s!' % ( greeting, login ) )
    ....:

In [86]: def welcome_2(login, greeting):
    ....:     print('%s, %s!' % ( login, greeting) )
    ....:

In [87]: welcome_1('Hi ', 'wrox')
Hi , wrox!

In [88]: welcome_2('Hi ', 'wrox')
Hi , wrox!

In [89]: welcome_2('wrox', 'Hi ')
wrox, Hi !
```



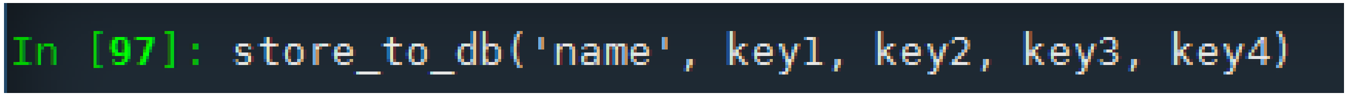
```
In [90]: welcome_1(greeting = 'Hi ', login = 'wrox')
Hi , wrox!

In [91]: welcome_1(login = 'wrox', greeting = 'Hi ')
Hi , wrox!

In [92]: add_name_to_db('name surename', 10, 20, 11, 5)
```



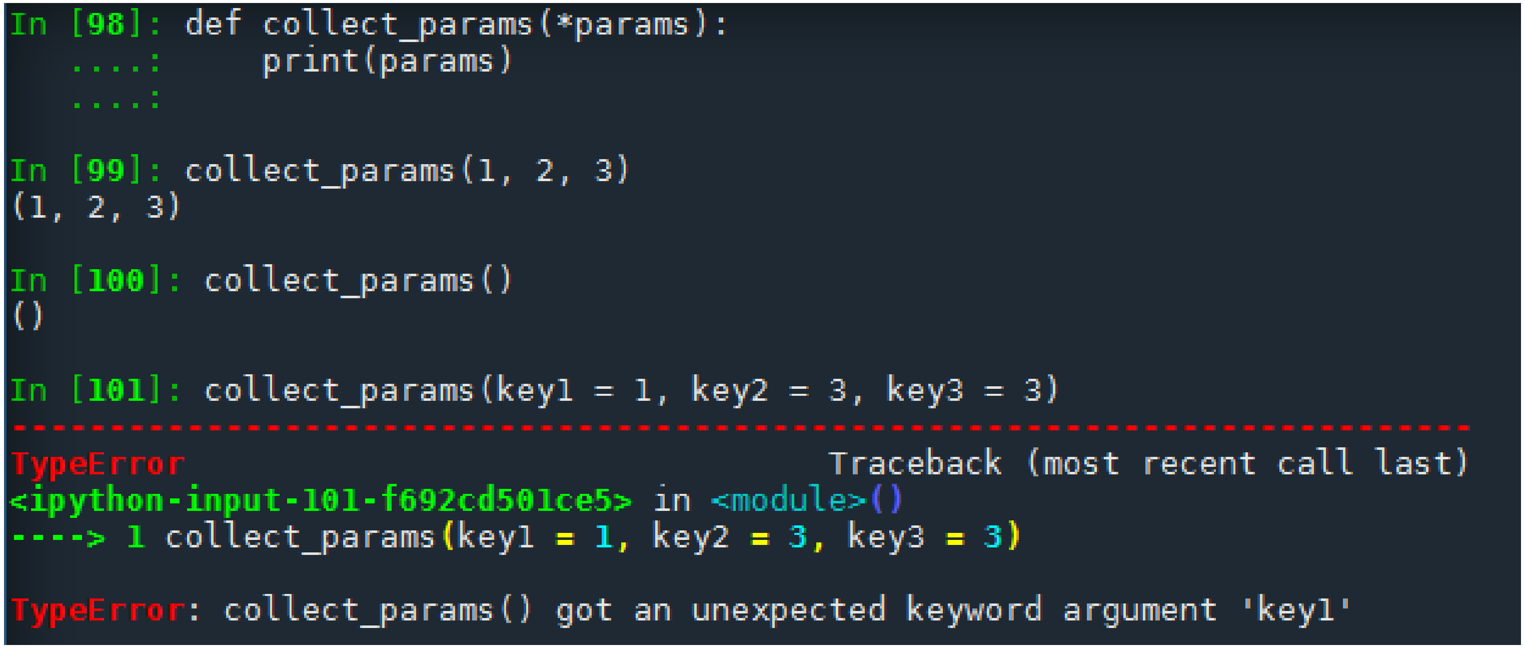
```
In [93]: add_name_to_db('name surename', hour = 10, min = 20, day = 11, month = 5)
```



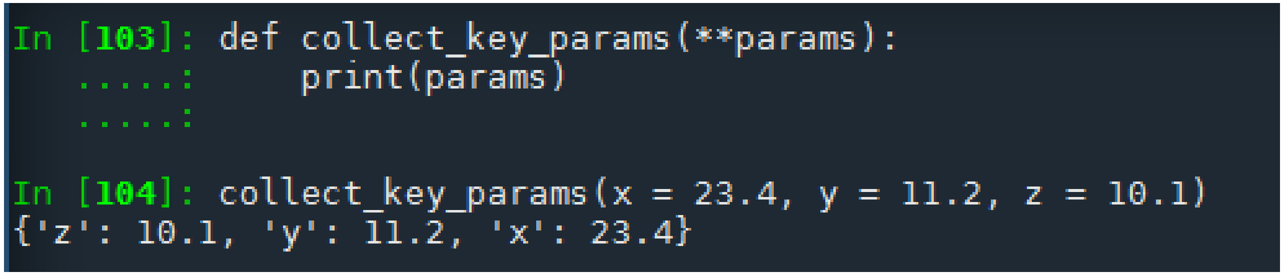
```
In [93]: def system_on_log(login, greeting = 'Hello ', punct = '!'):
   ....:     print('%s, %s%s' % ( greeting, login, punct ) )
   ....:

In [94]: system_on_log('wrox')
Hello , wrox!

In [95]: system_on_log('wrox', 'Hi ', '.')
Hi , wrox.

In [96]: system_on_log()
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
<ipython-input-96-ba2bc7821788> in <module>()
----> 1 system_on_log()

TypeError: system_on_log() missing 1 required positional argument: 'login'
```



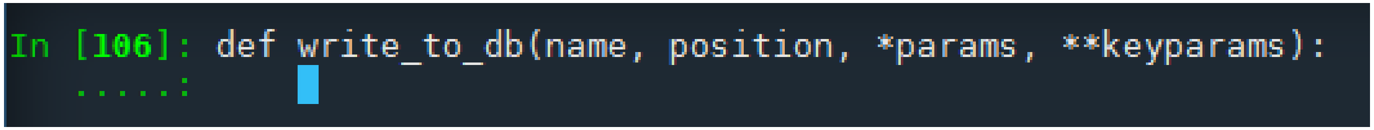
```
In [97]: store_to_db('name', key1, key2, key3, key4)
```



```
In [98]: def collect_params(*params):
   ....:     print(params)
   ....:

In [99]: collect_params(1, 2, 3)
(1, 2, 3)

In [100]: collect_params()
()

In [101]: collect_params(key1 = 1, key2 = 3, key3 = 3)
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
<ipython-input-101-f692cd501ce5> in <module>()
----> 1 collect_params(key1 = 1, key2 = 3, key3 = 3)

TypeError: collect_params() got an unexpected keyword argument 'key1'
```

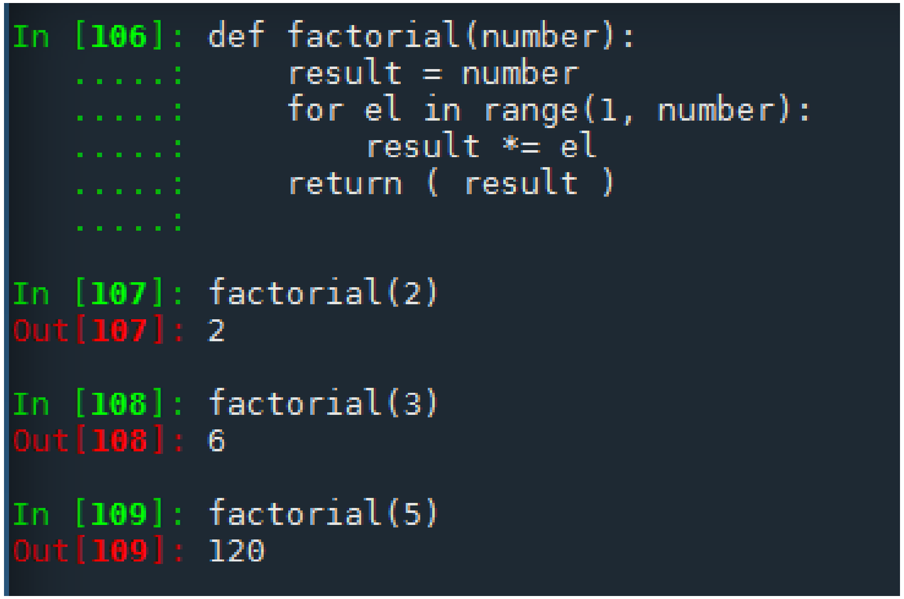


```
In [103]: def collect_key_params(**params):
   .....:     print(params)
   .....:

In [104]: collect_key_params(x = 23.4, y = 11.2, z = 10.1)
{'z': 10.1, 'y': 11.2, 'x': 23.4}
```

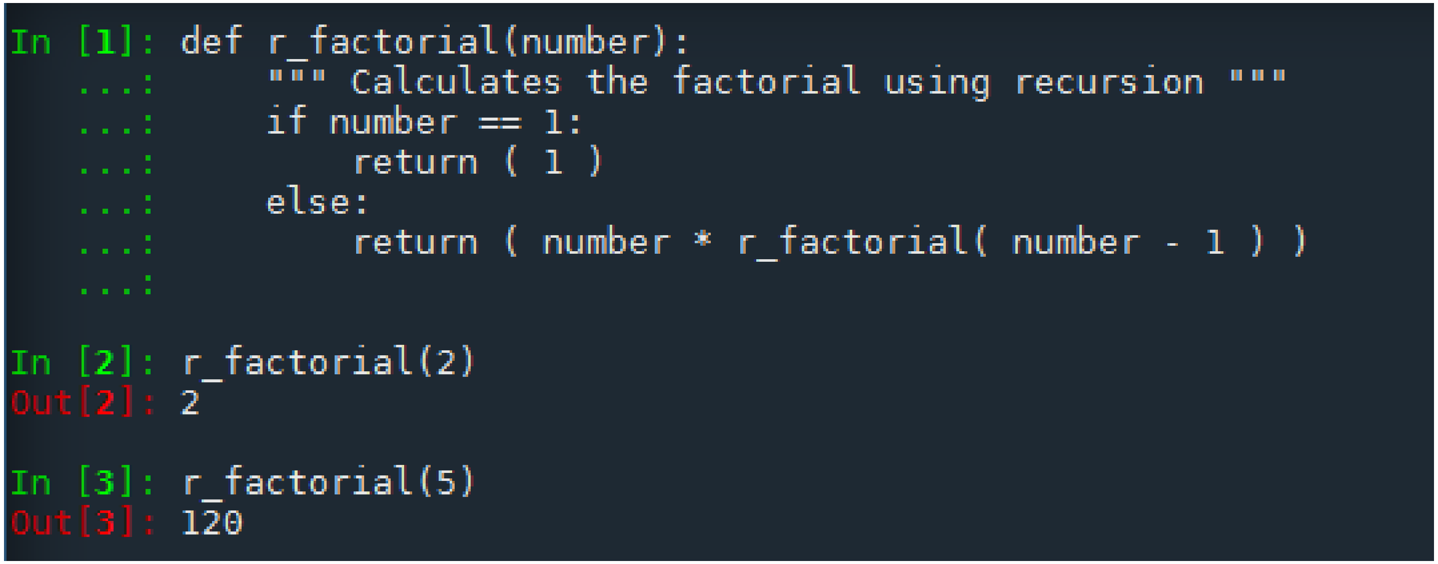


```
In [106]: def write_to_db(name, position, *params, **keyparams):
   ......:
```

Now, try to be creative here…



```
In [106]: def factorial(number):
     ...:     result = number
     ...:     for el in range(1, number):
     ...:         result *= el
     ...:     return ( result )
     ...:

In [107]: factorial(2)
Out[107]: 2

In [108]: factorial(3)
Out[108]: 6

In [109]: factorial(5)
Out[109]: 120
```



```
In [1]: def r_factorial(number):
   ...:     """ Calculates the factorial using recursion """
   ...:     if number == 1:
   ...:         return ( 1 )
   ...:     else:
   ...:         return ( number * r_factorial( number - 1 ) )
   ...:

In [2]: r_factorial(2)
Out[2]: 2

In [3]: r_factorial(5)
Out[3]: 120
```