

17.Reursion_SpecialFunction

May 8, 2020

0.0.1 What is Scope ? answer in chat box

scope --> lifetime of region where we store statements or variables

1 names space

```
[2]: # global scope  
name = 'sachin' # global variable
```

```
[5]: print(name)
```

sachin

```
[6]: print(__name__)
```

__main__

1.0.1 global scope

```
[7]: globals # is dictionary which store global data (variable, etc)
```

```
[7]: <function globals()>
```

```
[8]: globals()
```

```
[8]: {'__name__': '__main__',  
      '__doc__': 'Automatically created module for IPython interactive environment',  
      '__package__': None,  
      '__loader__': None,  
      '__spec__': None,  
      '__builtin__': <module 'builtins' (built-in)>,  
      '__builtins__': <module 'builtins' (built-in)>,  
      '_ih': ['',  
              'socpe --> lifetime of region',  
              '"# global scope\\nname = 'sachin"',  
              'print(name)',  
              'print(naem)',  
              'print(name)',
```

```

'print(__name__)',
'globals',
'globals()'],
'_oh': {7: <function globals()>},
'_dh': ['C:\\Batches\\Batch_7pm_online'],
'_In': ['',
'socpe --> lifetime of region',
"# global scope\\nname = 'sachin'",
'print(name)',
'print(naem)',
'print(name)',
'print(__name__)',
'globals',
'globals()'],
'_Out': {7: <function globals()>},
'_get_ipython': <bound method InteractiveShell.get_ipython of
<ipykernel.zmqshell.ZMQInteractiveShell object at 0x000002310FB3EEC8>>,
'_exit': <IPython.core.autocall.ZMQExitAutocall at 0x2310fb6f1c8>,
'_quit': <IPython.core.autocall.ZMQExitAutocall at 0x2310fb6f1c8>,
'_': <function globals()>,
'__': '',
'___': '',
'_i': 'globals',
'_ii': 'print(__name__)',
'_iii': 'print(name)',
'_i1': 'socpe --> lifetime of region',
'_i2': "# global scope\\nname = 'sachin'",
'_name': 'sachin',
'_i3': 'print(name)',
'_i4': 'print(naem)',
'_i5': 'print(name)',
'_i6': 'print(__name__)',
'_i7': 'globals',
'_7': <function globals()>,
'_i8': 'globals()'}

```

```
[9]: def hello(name):
      print("Hello ", name)
```

```
[10]: h = hello
```

```
[11]: h('sachin')
```

Hello sachin

```
[13]: globals()['h']
      # h is an identifier or reference
```

```
[13]: <function __main__.hello(name)>
```

```
[14]: globals()['hello']
```

```
[14]: <function __main__.hello(name)>
```

```
[15]: globals() == locals()
```

```
[15]: True
```

```
[16]: globals()['name']
```

```
[16]: 'sachin'
```

```
[17]: locals()['name']
```

```
[17]: 'sachin'
```

```
[20]: globals()['hello']('sachin')
```

```
Hello sachin
```

```
[21]: x = hello('sachin')
```

```
Hello sachin
```

```
[22]: print(x)
```

```
None
```

```
[23]: name = 'sachin' # global variable
def hello(name):
    """
        name is a local variable
    """
    print("hello, ", name)

x = hello('yadav')
print(x)
print(name)
```

```
hello, yadav
```

```
None
```

```
sachin
```

```
[24]: def check_local():
        print(globals() == locals())
```

```
[25]: print(globals() == locals())
      check_local()
      print(globals() == locals())
```

```
True
False
True
```

```
[1]: name
```

```

      □
↳ -----

      NameError                                Traceback (most recent call↳
↳ last)

      <ipython-input-1-9bc0cb2ed6de> in <module>
      ----> 1 name

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

```
[2]: x = 5
      def func(x):
          x += 10
          print("inside x is ", x)

      print("before x is ", x)
      func(x)
      print("after x is ", x)
```

```
before x is 5
inside x is 15
after x is 5
```

```
[3]: x = 5
      def func():
          print("inside value of x is: ", x)
      print('before x is: ', x)
      func()
      print('after x is: ', x)
```

```
before x is: 5
inside value of x is: 5
after x is: 5
```

```
[4]: x = 5
def func():
    x += 10
    print("inside value of x is: ", x)
print('before x is: ', x)
func()
print('after x is: ', x)
```

before x is: 5

```

      □
↳ -----

UnboundLocalError                                Traceback (most recent call↳
↳last)

  <ipython-input-4-a718be7c3c72> in <module>
      4     print("inside value of x is: ", x)
      5 print('before x is: ', x)
----> 6 func()
      7 print('after x is: ', x)

  <ipython-input-4-a718be7c3c72> in func()
      1 x = 5
      2 def func():
----> 3     x += 10
      4     print("inside value of x is: ", x)
      5 print('before x is: ', x)

UnboundLocalError: local variable 'x' referenced before assignment
```

```
[6]: x = 10
def func():
    x = 20
    print("global variable x is", globals()['x'], id(globals()['x']))
    print("local variable x is ", locals()['x'], id(locals()['x']))
func()
```

global variable x is 10 140732206654128
local variable x is 20 140732206654448

```
[9]: x = 10 # global
def func():
    print(locals()['x'])
```

```
[10]: func()
```

```

      □
↳-----

      KeyError                                Traceback (most recent call↳
↳last)

      <ipython-input-10-bd1982955a12> in <module>
      ----> 1 func()

      <ipython-input-9-7d9dd1461d12> in func()
          1 x = 10 # global
          2 def func():
      ----> 3     print(locals()['x'])

      KeyError: 'x'
```

```
[13]: x = 10 # global
      def func():
          globals()['x'] += 10 # local
```

```
[14]: func()
```

```
[15]: print(x)
```

20

```
[17]: x = 5
      def func():
          global x # permission to change global variable
          x += 10
          print('inside x : ', x)
      print("before x is ", x)
      func()
      print("after x is ", x)
```

```
before x is 5
inside x : 15
after x is 15
```

Recursion

when a function calls itself

```
[20]: def hello():  
      x = 5  
      print("local variable x: ", x)  
      hello()
```

```
[22]: #hello()
```

```
[23]: def hello():  
      hello()
```

```
[24]: hello()
```

```
↳ -----  
  
      RecursionError                                Traceback (most recent call↳  
↳last)  
  
      <ipython-input-24-a75d7781aaeb> in <module>  
----> 1 hello()  
  
      <ipython-input-23-b3f7a3b5d46b> in hello()  
      1 def hello():  
----> 2     hello()  
  
      ... last 1 frames repeated, from the frame below ...  
  
      <ipython-input-23-b3f7a3b5d46b> in hello()  
      1 def hello():  
----> 2     hello()  
  
      RecursionError: maximum recursion depth exceeded
```

```
[25]: import sys # python utilities
```

```
[26]: sys.getrecursionlimit()
```

```
[26]: 3000
```

```
[29]: c = 1
def hello():
    global c
    c += 1
    hello()
hello()
```

```

└─
↪-----

RecursionError                                Traceback (most recent call
↪last)

  <ipython-input-29-c6e397de18b4> in <module>
      4      c += 1
      5      hello()
----> 6 hello()

  <ipython-input-29-c6e397de18b4> in hello()
      3      global c
      4      c += 1
----> 5      hello()
      6 hello()

... last 1 frames repeated, from the frame below ...

  <ipython-input-29-c6e397de18b4> in hello()
      3      global c
      4      c += 1
----> 5      hello()
      6 hello()

RecursionError: maximum recursion depth exceeded
```

```
[30]: c
```

```
[30]: 2962
```

```
[31]: sys.setrecursionlimit(10000)
```

```
[32]: c = 1
```



```
[ ]: hello()
```

```
[1]: def hello(c):  
      if c <= 10:  
          print(c)  
          hello(c+1)
```

```
[2]: hello(1)
```

```
1  
2  
3  
4  
5  
6  
7  
8  
9  
10
```

```
[6]: def hello(c):  
      if c <= 5:  
          value = c + hello(c+1)  
          print(value)  
          return value  
      return 0
```

```
[7]: ans = hello(1)  
      print("Answer is : ", ans)
```

```
5  
9  
12  
14  
15  
Answer is : 15
```

```
[8]: def rev_string(string, i=0):  
      if i < len(string):  
          rev_string(string, i+1)  
          print(string[i], end='')
```

```
[9]: rev_string('sachin')
```

```
nihcas
```

iterative
prime
fabonacci
loop
...
tower of hanoi
Special Function

lambda

map

filter

reduce

closures

decorators

generators

[]: