20.Decorators_Generators

June 2, 2020

Visible ? Closures

a local scope accessable outside local scope

[1]:
```
def hello(name):
    def hi():
        print("Hello My name is ", name)
    return hi
```

[2]:
```
func = hello("Sachin Yadav")
print(func)

func()
```

```
<function hello.<locals>.hi at 0x000002428154E828>
Hello My name is  Sachin Yadav
```

Decorators

a python funciton which has a inner function in its body, also it's argument is also a function and it also returns a function

[6]:
```
def outer(func):
    def inner():
        print("I just Hacked You")
    return inner
```

[7]:
```
def hi():
    print('Hello World!!')
hi()
```

```
Hello World!!
```

[8]:
```
new_hi = outer(hi)
new_hi()
```

```
I just Hacked You
```

[19]:
```python
def outer(func):
    def inner(*args, **kwargs):
        print("_"*60)
        print("_"*60)
        print()
        func(*args, **kwargs)
        print()
        print("_"*60)
        print("_"*60)
    return inner
```

[20]:
```python
# @outer --> outer()
@outer
def hello(name):
    print(f"{name.upper()}".center(60))

hello('Sachin Yadav')
```

```
------------------------------------------------------------
------------------------------------------------------------

                        SACHIN YADAV

------------------------------------------------------------
------------------------------------------------------------
```

[21]:
```python
@outer
def add(x, y):
    print(f"x = {x}".center(60))
    print(f"y = {y}".center(60))
    print(f"x + y = {x+y}".center(60))
```

[22]:
```python
add(5, 4)
```

```
------------------------------------------------------------
------------------------------------------------------------

                          x = 5
                          y = 4
                        x + y = 9

------------------------------------------------------------
------------------------------------------------------------
```

[13]:
```python
def hello(name):
    print(f"Hello world my name is {name}.".center(60))
```

```
hello = outer(hello) # we are wasting a line here

hello("Sachin Yadav")
```

```
------------------------------------------------------------
------------------------------------------------------------

            Hello world my name is Sachin Yadav.

------------------------------------------------------------
------------------------------------------------------------
```

[11]:
```
print(id(outer))
```

```
2484661625624
```

[23]:
```
@outer
@outer
def awesome():
    print("!!!Mind Blowing!!! Kuch Smjh Nh aaya !!Bouncer!!!")
```

[24]:
```
awesome()
```

```
------------------------------------------------------------
------------------------------------------------------------

------------------------------------------------------------
------------------------------------------------------------

!!!Mind Blowing!!! Kuch Smjh Nh aaya !!Bouncer!!!

------------------------------------------------------------
------------------------------------------------------------

------------------------------------------------------------
------------------------------------------------------------
```

[33]:
```
def decorator(old_func):
    def new_func(*args, **kwargs):
        print("Yeoo!! you got some exiting features throgh decorator")
        old_func(*args, **kwargs)
    return new_func
```

[36]:
```
@decorator
@decorator
def hi(name):
    print(f"name: {name}")
```

```
#hi = decorator(hi)
#hi = decorator(hi)
```

[37]:
```
hi("Sachin Yadav")
```

```
Yeoo!! you got some exiting features throgh decorator
Yeoo!! you got some exiting features throgh decorator
name: Sachin Yadav
```

Flask, Django --> Frame works works decorator

[38]:
```
import time
http_response = f"""Http/1.1 200 OK
Encoding: utf-8
Content-type: text/html
Set-Cookies: username=sachin;password=nihcas
Date: {time.ctime}

<!Doctype html>
<html>
    <body>
        <h1 style='color:red'> Hello World!!</h1>
    </body>
</html>
"""
```

[40]:
```
print(repr(http_response))
```

```
"Http/1.1 200 OK\nEncoding: utf-8\nContent-type: text/html\nSet-Cookies:
username=sachin;password=nihcas\nDate: <built-in function ctime>\n\n<!Doctype
html>\n<html>\n    <body>\n        <h1 style='color:red'> Hello World!!</h1>\n
</body>\n</html>\n"
```

[60]:
```
def http(func):
    header = f"Http/1.1 200 OK\nEncoding: utf-8\nContent-type: text/
 ↪html\nSet-Cookies: username=sachin;password=nihcas\nDate: {time.
 ↪ctime()}>\n\n"
    def inner():
        html = func()
        response = header + html
        return response
    return inner
```

[62]:
```
@http
def home():
    page = """<!Doctype html>
    <html>
```

```
    <body>
        <h1>Welcome to home page</h1>
    </body>
    </html>
    """
    return page
```

[63]:
```
print(home())
```

```
Http/1.1 200 OK
Encoding: utf-8
Content-type: text/html
Set-Cookies: username=sachin;password=nihcas
Date: Tue Jun  2 20:30:23 2020>

<!Doctype html>
    <html>
    <body>
        <h1>Welcome to home page</h1>
    </body>
    </html>
```

Data Science

class link -> 12:30 - 03:00pm

decorators, clousures

0.0.1 Generators

[64]:
```
r = range(13, 131, 13)
```

[65]:
```
print(r)
```

```
range(13, 131, 13)
```

[66]:
```
print(*r)
```

```
13 26 39 52 65 78 91 104 117 130
```

[67]:
```
#map()# generator
```

[68]:
```
mgen = map(int, [ '1', '2', '3', '4', '5'])
```

[69]:
```
print(mgen)
```

```
<map object at 0x000002428332A8C8>
```

[70]:
```
print(next(mgen))
```

```
1
```

[71]:
```
print(next(mgen))
```

```
2
```

[72]:
```
while True:
    print(next(mgen))
```

```
3
4
5
```

```
␣
↪---------------------------------------------------------------------------

        StopIteration                                 Traceback (most recent call␣
↪last)

        <ipython-input-72-d605c72151f9> in <module>
          1 while True:
    ----> 2     print(next(mgen))


        StopIteration:
```

[73]:
```
def mygen():
    print("pause")
    yield 1
    print("resume")
    yield 2
    yield 3
```

yield keyword is used to create generators

[74]:
```
gen = mygen()
```

[75]:
```
print(gen)
```

```
<generator object mygen at 0x0000024283D139C8>
```

[76]:
```
def mygen():
    print("pause")
    yield 1
    print("resume")
    yield 2
    yield 3
gen = mygen() # prime condition creating a generator
print(next(gen))
print(next(gen))
print(next(gen))
print(next(gen))
```

```
pause
1
resume
2
3
```

```
␣
↪---------------------------------------------------------------------------

StopIteration                             Traceback (most recent call␣
↪last)

<ipython-input-76-69bfbaeebb40> in <module>
  9 print(next(gen))
 10 print(next(gen))
---> 11 print(next(gen))


StopIteration:
```

[77]:
```
def mygen(start, end, jump=1):
    while start <= end:
        yield start
        start += jump
```

[78]:
```
for var in mygen(1, 10, 2):
    print(var)
```

```
1
3
5
7
9
```

[79]:
```
def mygen():
    c = 1
    while True:
        yield c
        c += 1
```

[ ]:
```
for var in mygen():
    print(var)
```

[88]:
```
def mymap(fun, seq):
    for value in seq:
        yield fun(value)

l = list( mymap( lambda x: x**2, [1, 2, 3, 4, 5]) )
print(l)
```

```
[1, 4, 9, 16, 25]
```

[82]:
```
s = [ '1', '2', '3']
```

[83]:
```
m = mymap(int, s)
```

[84]:
```
next(m)
```

[84]:
```
1
```

[85]:
```
next(m)
```

[85]:
```
2
```

[86]:
```
next(m)
```

[86]:
```
3
```

[87]:
```
next(m)
```

```
␣
↪---------------------------------------------------------------------------

       StopIteration                             Traceback (most recent call␣
↪last)

       <ipython-input-87-8efa10874b95> in <module>
   ----> 1 next(m)
```

```
        StopIteration:
```

File Handling

[ ]: