

Python Decorators - Assignment

a)

Create a decorator `do_twice` which will perform a function twice.

Sample Output:

```
Hello Singapore
Hello Singapore
```

In [1]:

```
# YOUR CODE HERE
import functools

def do_twice(func):
    @functools.wraps(func)
    def inner(*args, **kwargs):
        func(*args, **kwargs)
        func(*args, **kwargs)

    return inner

@do_twice
def greet(whom):
    print('Hello', whom)

greet('Singapore')
```

```
Hello Singapore
Hello Singapore
```

b)

Create a decorator `log` which will save messages to a log file `log.txt` before and after running a function.

- Name of function `f()` can be obtained with `f.__name__`

Sample Output:

```
Start function greet
End function greet
```

In [2]:

```
# YOUR CODE HERE
import functools

def log(func):

    @functools.wraps(func)
    def inner(*args, **kwargs):
        with open('log.txt', 'a') as f:
            f.write('Start function {}\n'.format(func.__name__))
        func(*args, **kwargs)
        with open('log.txt', 'a') as f:
            f.write('End function {}\n'.format(func.__name__))

    return inner

@log
def greet(whom):
    print('Hello', whom)

#!del Log.txt
greet('Singapore')
```

Hello Singapore

In [3]:

```
!more log.txt
```

```
Start function greet
End function greet
Start function greet
End function greet
Start function greet
End function greet
```