Exercise 10 – Modules and packages

Objective

To write and call our own user-written modules, and to continue practising Python.

Questions

1. In this exercise, we will take two functions you turn functions into a module.

   In **Ex10_1.py** ,there are two timing functions: **start_timer()** and **end_timer()**.Use that file as a basis of this exercise, or your own solution if you wish.

   Create a module called **mytimer**, which contains these functions (and any other supporting variables). Test the module by importing it and calling the functions before and after a lengthy operation, as before.

   Note: There is a module called **timeit** in the Python Standard Library. If you look in the documentation, you will find it is rather more complex than ours. On Windows, there is also a module bundled with Python called **timer**. So please do not use either of those module names.

2. Now test your module's docstring using IDLE.

   To test under IDLE, first **import mytimer**. Did that work? If IDLE did not find your module, then maybe you should tell it where it is (**hint:** sys.path)? The easiest way to grab the path is to copy it from the Address bar in Windows Explorer and paste it into IDLE (use a 'raw' string).

   Once you have managed to import the module, type:

   ```
   >>> help(mytimer)
   ```

3. Our module is not complete without some tests. Add a simple test to the docstring: call start_timer() immediately followed by end_timer(), so that the result is predictable. Do not forget to add the expected output. Then add the test for__ **main__**, with the call to **doctest.testmod()**.

   Test by running **timer.py –v** from the Windows command-line (cmd.exe).

Solutions

Here are our versions of these exercises, remember that yours can be different to these, but still correct. If in doubt, ask your instructor.

The test script looks like this:

```
import mytimer


mytimer.start_timer()
lines = 0
for row in open("words"):
   lines += 1
mytimer.end_timer()
print("Number of lines:", lines)
```

Here is our final module:

```
""" This user written module contains a simple mechanism for timing
operations from Python. It contains two functions, start_timer(), which
must be called first to initialise the present time, and end_timer()which
calculates the elapsed CPU time and displays it.


>>> start_timer()
>>> end_timer()
End time    : 0.000 seconds
"""
import os


start_time = None


# TIMER FUNCTIONS
def start_timer():
```

```
    """ The start_timer() function marks the start of a
    Timed interval, to be completed by end_timer().
    This function requires no parameters.
    """
    global start_time
    start_time = os.times()[:2]
    return

def end_timer(txt='End time'):
    """ The end_timer() function completes a timed interval
    started by start_timer.  It prints an optional text
    message (default 'End time') followed by the CPU time
    used in seconds.
    This function has one optional parameter, the text to
    be displayed.
    """
    end_time = os.times()[:2]
    print ("{0:<12}: {1:01.3f} seconds".
       format(txt, end_time - start_time))
    return

if __name__ == "__main__":
    import doctest
    doctest.testmod()
```