Chap16_TimeExercises

May 15, 2020

```
In [ ]: # Chapter 16. Classes and Functions
```

```
In [1]: #Exercise 16.1
        class Time(object):
            """ represents the time of day."""
        time=Time()
        time.hour = 10
        time.minute = 58
        time.second = 40

        def print_time(t):
            print "%.2d:%.2d:%.2d" % (t.hour, t.minute, t.second)

        print_time(time)
```

```
10:58:40
```

```
In [2]: #Exercise 16.2
        class Time(object):
            """ Represents the time of day."""
        t1=Time()
        t2=Time()
        t1.hour=9
        t1.minute=59
        t1.second=30
        t2.hour=10
        t2.minute=57
        t2.second=30
        def is_after(t1,t2):
            a=3600*t1.hour+60*t1.minute+t1.second
            b=3600*t2.hour+60*t2.minute+t2.second
            return a>b

        # THINK ABOUT... the next definition of the same function...
        def is_after_T(t1,t2):
            return (t1.hour, t1.minute, t1.second) > (t2.hour, t2.minute, t2.second)
        # What does the "T" from the name of the function stand for?
        print is_after(t1, t2)
```

```
False
```

```
In [1]: #Exercise 16.3
        #increment - without any loop --> MODIFIER
        class Time(object):
            """ represents the time of day. """

        time = Time()
        time.hour = 10
        time.minute = 37
        time.second = 50

        def increment(time, seconds):
            print ("Timpul original: %.2d:%.2d:%.2d"% (time.hour, time.minute, time.second))

            time.second += seconds
            if time.second > 59:
                cat, rest = divmod(time.second, 60)
                time.minute += cat
                time.second = rest
            if time.minute > 59:
                cat, rest = divmod(time.minute, 60)
                time.hour += cat
                time.minute = rest
            print "Secunde in plus %g " % (seconds)
            print "Noul timp: %.2d:%.2d:%.2d" % (time.hour, time.minute, time.second)

        increment(time,450)
```

```
Timpul original: 10:37:50
Secunde in plus 450
Noul timp: 10:45:20
```

```
In [25]: #Exercise 16.4
         #increment --> PURE FUNCTION
         class Time(object):
             """ represents the time of day. """

         def increment(t,seconds):

             tOut = Time()
             minutes, tOut.second = divmod(t.second+seconds, 60)
             tOut.hour, tOut.minute = divmod(t.minute+minutes, 60)
             return tOut

         time = Time()
         time.hour = 10
         time.minute = 37
         time.second = 50

         print "Timpul initial", time.hour, time.minute, time.second
         print "Secunde in plus", 1350
         timeOut = increment(time,1350)
         print "Noul timp:", timeOut.hour, timeOut.minute, timeOut.second
```

```
Timpul initial 10 37 50
Secunde in plus 1350
Noul timp: 1 0 20
```

```
In [34]: #Exercise 16.5

         class Time(object):
             """ represents the time of day. """

         time = Time()
         time.hour = 10
         time.minute = 55
         time.second = 30

         def time_to_int(t):
             minutes = t.hour * 60 + t.minute
             seconds = minutes * 60 + t.second
             return seconds

         def int_to_time(seconds):
             t = Time()
             minutes, t.second = divmod(seconds, 60)
             t.hour, t.minute = divmod(minutes, 60)
             return t

         def increment(t, seconds):
             return int_to_time(time_to_int(t) + seconds)

         print ("Timpul initial: %.2d:%.2d:%.2d"% (time.hour, time.minute, time.second))
         new_time = increment(time, 350)
         print ("Noul timp : %.2d:%.2d:%.2d"% (new_time.hour, new_time.minute, new_time.second))
```

```
Timpul initial: 10:55:30
Noul timp : 11:01:20
```

```
In [35]: #Exercise 16.6
         class Time(object):
             """ represents the time of day."""

         def time_to_int(time):
             minutes = time.hour * 60 + time.minute
             seconds = minutes * 60 + time.second
             return seconds

         def int_to_time(seconds):
             t = Time()
             minutes, t.second = divmod(seconds, 60)
             t.hour, t.minute = divmod(minutes, 60)
             return t

         def mul_time(time, multiply):
             return int_to_time(time_to_int(time) * multiply)

         time = Time()
         time.hour = 5
         time.minute = 30
         time.second = 20

         print ("Timpul initial: %.2d:%.2d:%.2d" % (time.hour, time.minute, time.second))

         new_time = mul_time(time, 4)
         print ("Noul timp: %.2d:%.2d:%.2d" % (new_time.hour, new_time.minute, new_time.second))
```

```
Timpul initial: 05:30:20
Noul timp: 22:01:20
```

```
In [36]: #Exercise 16.7.
         #TO DO
```