T: Time

Gareth McCaughan Revision 1.8, May 14, 2001

Credits

For the LaTeX source of this sheet, and for more information on LiveWires and on this course, see the LiveWires web site at http://www.livewires.org.uk/python/

Introduction

Time is important. So you might want to know how long someone has been using your program, or what time of day it is; you might want to make something happen exactly 10 times per second; in any case, you need to know what Python can do about time. This sheet tells you about that.

The 'time' module

If you say `import time` then after that you can use a number of functions for working with times. If you're curious about what "`import`" means, see Sheet M (*Modules*).

Telling the time

`time.time()` gives the number of seconds since the very beginning of the year 1970. You may think this is a strange way to represent time. You'd be right too, but fortunately Python provides ways of turning this sort of time into something more useful.

`time.localtime(t)`, if `t` is a time value produced by `time.time()`, is an object made up of 9 numbers. Here's what it produced for me using the time right now:

```
>>> time.localtime(time.time())
(1999, 8, 10, 17, 21, 16, 1, 222, 1)
```

Those 9 numbers are, in order:

0. the year
1. the month of the year (January is 1, December is 12)
2. The day of the month
3. The hour of the day (in the 24-hour clock: so `17` means 5pm
4. The minute of the hour
5. The number of seconds past the minute
6. The day of the *week* (Monday is 0, Sunday is 6)
7. The day number within the year (1 January is 1)
8. 1 if "daylight saving time" is in force, 0 otherwise

So, you can use this to make a simple clock.

```
import time
while 1:
  t = time.localtime(time.time())
  print 'The time is', t[3], ':', t[4], 'and', t[5], 'sec'.
  time.sleep(1)                                    I'll explain this in a moment.
```

Describing the time

There's a complicated function called `time.strftime` which lets you print times more neatly. If you want to know the gruesome details, ask a leader. Here's a little example.

```
>>> import time
>>> time.strftime('%A, %d %B %Y, at %I:%M%p', time.localtime(time.time()))
'Tuesday, 10 August 1999, at 05:41PM'
```

Waiting

`time.sleep(0.1234)` does absolutely nothing for 0.1234 seconds (or as close to that as the machine can manage).