

# Rob's Raspberry Pi blog

I post the things I have learned about the Raspberry PI in a format for others to use. I hope this information helps you with your projects

Monday, 8 February 2016

## Raspberry Pi - All things Date and Time in Python

When writing python code you often want to compare times dates and add delays etc. Here are a few things I have learnt that may help you in your projects.

### Time delays

These are easy to use and are very useful all you need is to define time then use the sleep function.

```
import time
print "Waiting 10 seconds"
time.sleep(10)
print "Running again"
```

The sleep function can also do delays less than a second for example `time.sleep(0.01)` this is sometimes used for some GPIO that may need a moment to complete in another thread.

### Formatting Date time

You can convert datetime to various parts that you want to display

Here I create a date time to format:-

```
import datetime
MyDateTime = datetime.datetime(2016,1,3,8,30,20)
```

#### Simple date and time extractions

```
print "Day of month :", MyDateTime.day
print "Month :", MyDateTime.month
print "year :", MyDateTime.year
print "Weekday Monday to Sunday (0 to 6) :", MyDateTime.weekday()
print "Weekday Monday to Sunday (1 to 7) :", MyDateTime.isoweekday()
```

Program output:-

```
Day of month : 3
Month : 1
year : 2016
Weekday Monday to Sunday (0 to 6) : 6
Weekday Monday to Sunday (1 to 7) : 7
```

#### Formatted date outputs

```
print "ISO Formatted date      :", MyDateTime.isoformat()
print "Long format date time  :", MyDateTime.ctime()
```

Program output:-

```
ISO Formatted date      : 2016-01-03T08:30:20
Long format date time  : Sun Jan  3 08:30:20 2016
```

#### String formatted dates

```
print MyDateTime.strftime("Time      : %H:%M:%S")

print MyDateTime.strftime("Short Date : %d/%m/%Y")

print MyDateTime.strftime("Long Date  : %d/%m/%Y")

print MyDateTime.strftime("D/Month/Y   : %d-%b-%Y")
print MyDateTime.strftime("Full Date   : %A %d %B %Y")
```

Program output:-

```
Time      : 08:30:20
Short Date : 03/01/16
Long Date  : 03/01/2016
D-Month-Y   : 03-Jan-16
Full Date   : Sunday 03 January 2016
```

For the complete reference of `strftime` click this link <http://strftime.org/>

Search



Blog



About



## How to compare dates and times and just the time

To compare first you need to input a date, time or both. To input a full date and time use the datetime function with year,day,month,hour,minute,second. Use the if statement with greater than (>) or less than (<) operators (never use == as the time is measured in microseconds so its unlikely match exactly).

```
import datetime
MyDateTime = datetime.datetime(2016,1,3,8,30,20)
print "Entered Date time is ", MyDateTime

if MyDateTime > datetime.datetime.now():
    print "Current datetime is bigger"
else:
    print "My datetime is bigger"
```

To compare time only use the time function and compare the time only. can be used to do some operation the same time every day.

```
MyTimeOnly = datetime.time(8,31,53)
print "Entered time is ", MyTimeOnly
```

```
if MyTimeOnly > datetime.time():
    print "Current time is bigger"
else:
    print "My time is bigger"
```

Program output:-

```
Entered Date time is  2016-01-03 08:30:20
My datetime is bigger
Entered time is  08:31:53
Current time is bigger
```

## What is UTC time and what does it mean?

UTC means Coordinated Universal Time (This used to be known as Greenwich Mean Time, or GMT). UTC is the common time standard across the world.

So why UTC and not CUT?

**The Official Abbreviation**

The official abbreviation for Coordinated Universal Time is UTC. It came about as a compromise between English and French speakers.

- **Coordinated Universal Time** in English would normally be abbreviated CUT.
- **Temps Universel Coordonné** in French would normally be abbreviated TUC.

How to display the current time and UTC time

```
import datetime
print "The current local date time is ",datetime.datetime.now()

print "The current UTC date time is  ",datetime.datetime.utcnow()
```

Program output:-

```
The current local date time is  2016-02-07 21:59:32.597914
The current UTC date time is   2016-02-07 10:59:32.599402
```

The resolution of time is to the microsecond. The date is the February 2nd 2016 (My time zone is Sydney Australia so UTC -11 hours on that date).

To get the current time zone UTC offset do the following

```
import time
print "The UTC offset is",time.altzone/3600,"Hours"
```

Program output:-

```
The UTC offset is -11 Hours
```

The function `time.altzone` returns seconds so divide by 3600 to get hours offset

Rate Page:    Awesome (1)    Great (0)    OK (0)    Poor (0)    Dreadful (0)

 +1 Recommend this on Google

## 2 comments:



**Johnny Marketer** 10 July 2016 at 00:58

Awesome! Very helpful. Thank you

[Reply](#)

[Replies](#)



**Robert Lorenzo** 26 July 2016 at 06:24

Your most welcome :)

[Reply](#)

Enter your comment...