MOONHACK 2022

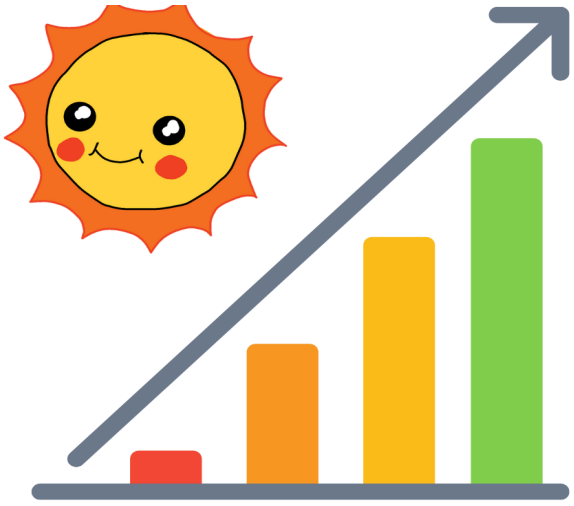
Light

Use data to help you calculate the precise time of day to turn your lights on and off for maximum efficiency.

INTRODUCTION

What you will make

You will write a program that will tell its user morning and evening times for astronomical twilight, nautical twilight and civil twilight, specific to their location and elevation.



What you will learn

- How to import third party packages
- How to use data for purpose

What you will need

HARDWARE

A computer

SOFTWARE

Mu

TO DOWNLOAD MU

Follow the instructions on this page

Getting started with Mu

Code Club Australia acknowledges the Traditional Owners of Country throughout Australia and their continuing connection to land, sea and community. We pay our respects to First Nations people and their living cultures and to Elders past and present.

MOONHACK 2022

Light

ACKNOWLEDGEMENT

This project was made with the collaboration of Dr Brad Tucker, an astrophysicist and cosmologist who is currently a Fellow at Mt Stromlo Observatory at the Australian National University. Dr Tucker has a passion for astronomy and frequently gives talks to school groups and the public, as well as regularly releasing Space News and hot topics on his YouTube channel.

The Project

Creating an energy efficient home is a common goal of families. By utilising information from online databases we can develop a schedule for the lighting in our homes. Newer homes who run their lights on a programmable timer can use the Python project to determine the best time for lights to be turned on and off. This program could also be utilised by larger office buildings.

Additional notes for Educators

Astronomical Twilight - when there is absolutely no light from the Sun and astronomers call it dark

Nautical Twilight - when it is dark enough you can not see the ocean

Civil Twilight - the time right before sunrise and right after sunset

Ephem - The word ephem is short for Ephemeris which is the term for a table giving the positions of a planet, asteroid or comet for a series of dates.

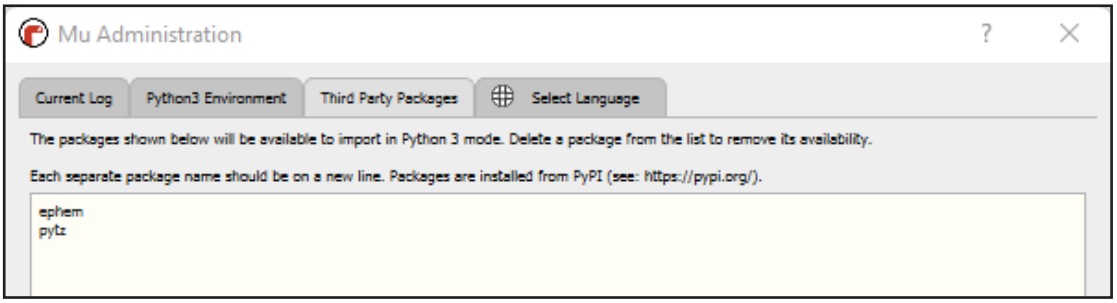
Pytz - Pytz is a module that determines the time of any region using the local timezone name. It uses your machine's time to make its calculations.

STEP 1 - IMPORTING THE LIBRARIES

To get started, we need to import the libraries that will be used for the calculations.

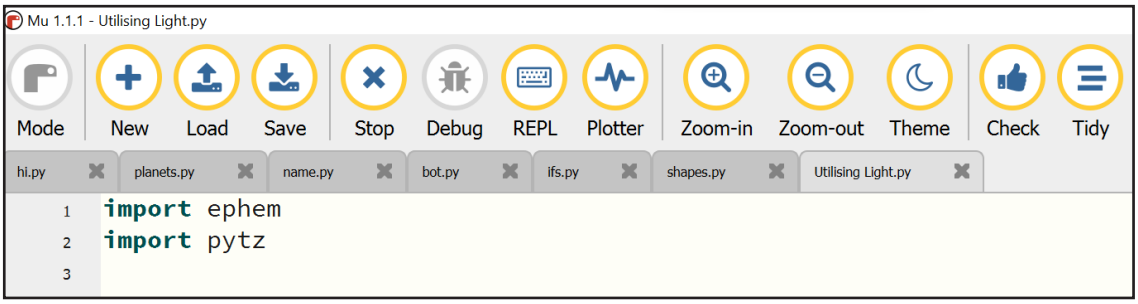
Open Mu and create a new project. In the bottom left corner of your screen click on the gear icon. Select the Third Party Packages tab.

Type ephem on the first line, hit enter, and type pytz on the second line. Then click OK. This will import the two libraries needed for this projet.

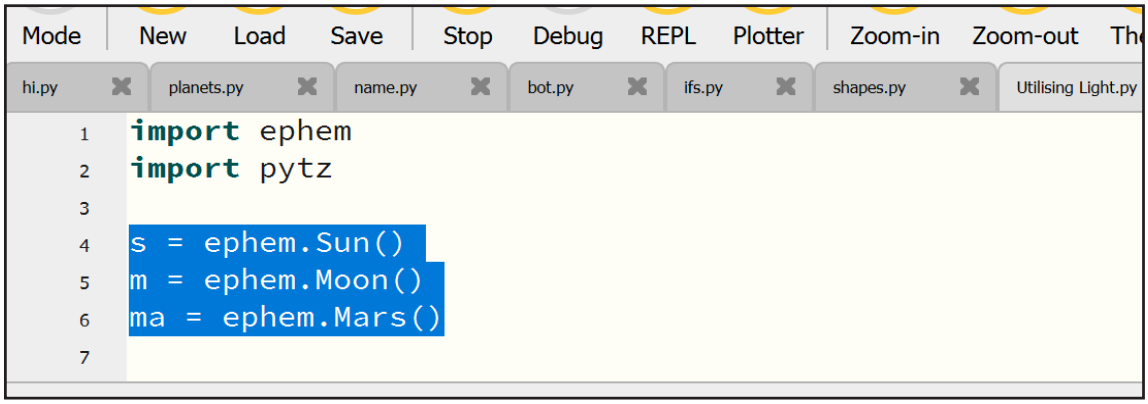


STEP 2 - PROGRAM BEGINNINGS

Import the 2 libraries, ephem and pytz.



Create variables to load the sun, moon and planet data from ephem.



This code works by knowing your exact location.

Add code to calculate your exact position in the world. To find your location information use a website such as Maps. Enter your location and it will provide you with the latitude, longitude and elevation

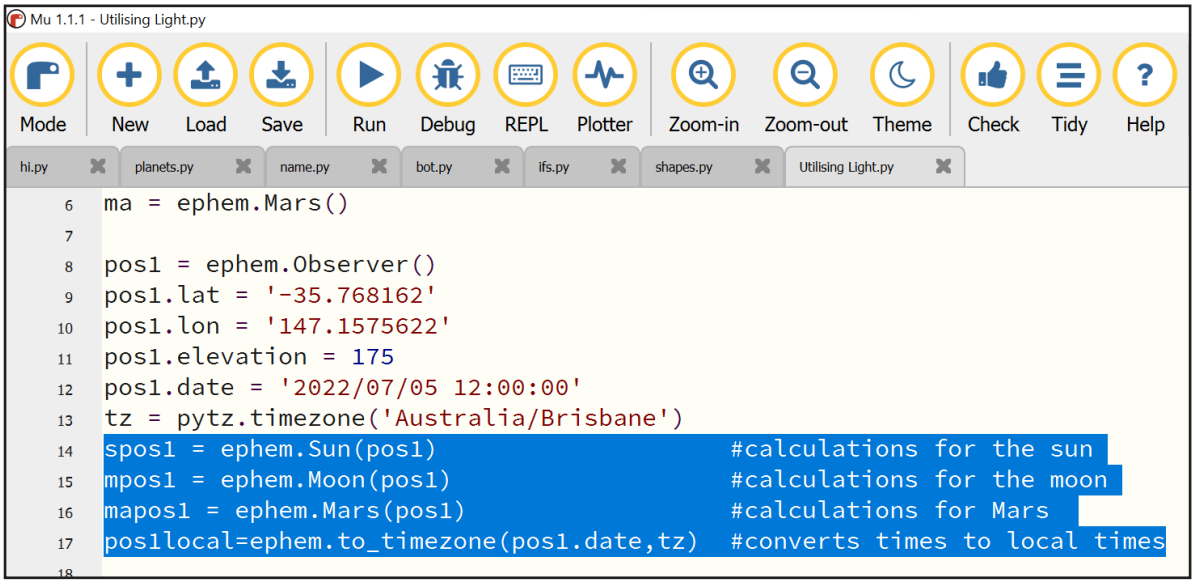
```
s = ephem.Sun()
m = ephem.Moon()
ma = ephem.Mars()

pos1 = ephem.Observer()
pos1.lat = '-35.768162'   #add your latitude here
pos1.lon = '147.1575622'  #add your longitude here
pos1.elevation = 175      #add your elevation here
```

Add code to set up the date you want to make calculations for and your timezone.

```
s = ephem.Sun()
m = ephem.Moon()
ma = ephem.Mars()

pos1 = ephem.Observer()
pos1.lat = '-35.768162'
pos1.lon = '147.1575622'
pos1.elevation = 175
pos1.date = '2022/07/05 12:00:00'  #enter the date, leave time at noon
tz = pytz.timezone('Australia/Brisbane')  #enter your state's capital city
```

```
Running: Utilising Light.py
>>>
```

This code will create the calculations for the variables and convert them to local times.



Time to start putting it all together. The next code will calculate the sunrise and sunset time at your location.

Use the print function for sunrise and sunset.

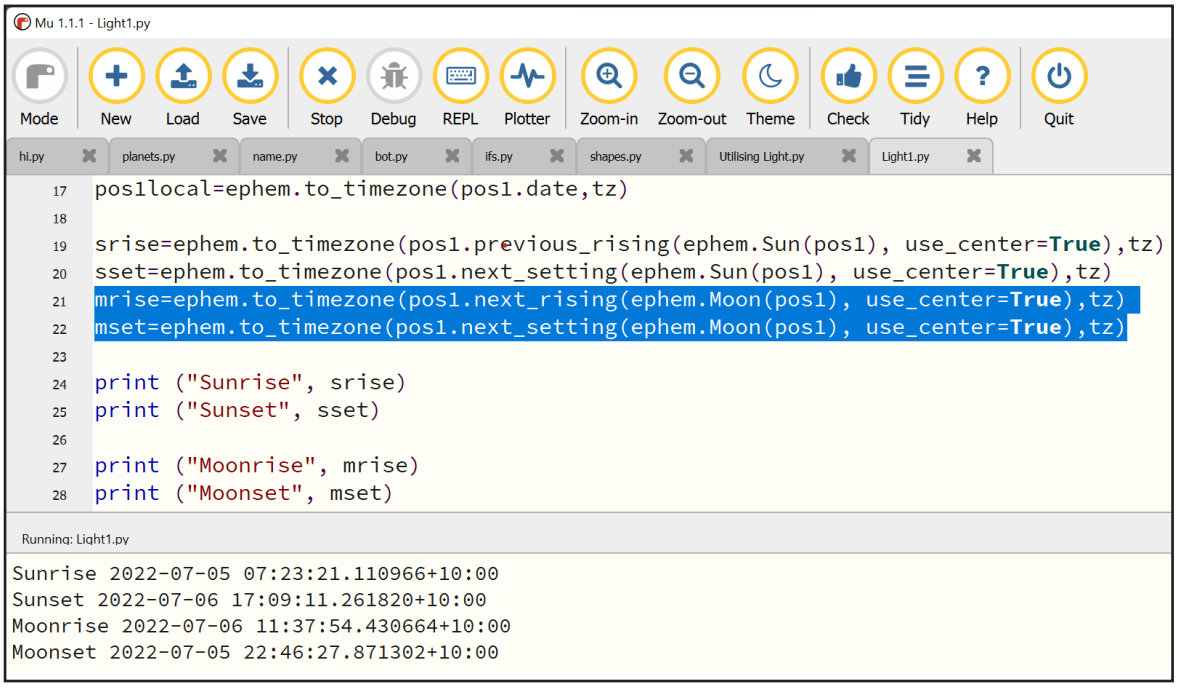
```
mpos1 = ephem.Moon(pos1)                    #calculations for the moon
mapos1 = ephem.Mars(pos1)
pos1local=ephem.to_timezone(pos1.date,tz)

srise=ephem.to_timezone(pos1.previous_rising(ephem.Sun(pos1), use_center=True),tz)
sset=ephem.to_timezone(pos1.next_setting(ephem.Sun(pos1), use_center=True),tz)

print ("Sunrise", srise)
print ("Sunset", sset)
```

Run your program. You should see the date and exact time for sunset and sunrise in your location.

STEP 3 - MORE DATA

Next we will retrieve data that will give precise times for the moon rising and setting.

Add code to calculate Moonrise and Moonset and add in the print function for these.



You can also add code to display what fraction of the moon will be illuminated. Don't forget the print function.

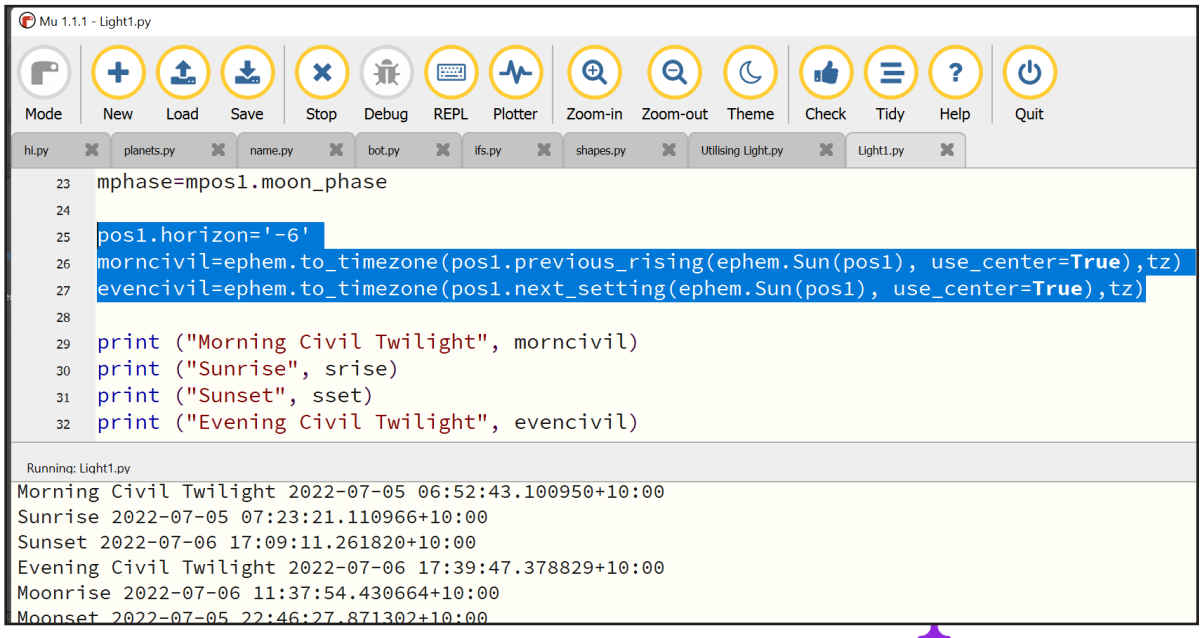
```
mrise=ephem.to_timezone(pos1.next_rising(ephem.Moon(pos1), use_center=True),tz)
mset=ephem.to_timezone(pos1.next_setting(ephem.Moon(pos1), use_center=True),tz)
mphase=mpos1.moon_phase

print ("Sunrise", srise)
print ("Sunset", sset)

print ("Moonrise", mrise)
print ("Moonset", mset)
print ("Moon fraction", mphase)

Running: Light1.py
Sunset 2022-07-06 17:09:11.261820+10:00
Moonrise 2022-07-06 11:37:54.430664+10:00
Moonset 2022-07-05 22:46:27.871302+10:00
```

STEP 4 - COMPLEX LIGHT DATA

To best calculate the times for lights going on and off we need more data than just sunrise and sunset. Let's add to the program so we can also get the exact time for civil twilight, nautical twilight and astronomical twilight.

Define morning and evening civil twilight in your program. NB - Civil Twilight refers to the time just before sunrise and just after sunset when the difference between the horizon and the sun is 6 degrees.

Don't forget to add your print function at the bottom and then run your program to test it.



Define morning and evening nautical twilight. Nautical Twilight takes place when the difference between the horizon and sun is 12 degrees.

```
27 evencivil=ephem.to_timezone(pos1.next_setting(ephem.Sun(pos1), use_center=True),tz)
28
29 pos1.horizon='-12'
30 mornnaut=ephem.to_timezone(pos1.previous_rising(ephem.Sun(pos1), use_center=True),tz)
31 evennaut=ephem.to_timezone(pos1.next_setting(ephem.Sun(pos1), use_center=True),tz)
32
33 print ("Morning Nautircal Twilight", mornnaut)
34 print ("Morning Civil Twilight", morncivil)
35 print ("Sunrise", srise)
```

Add the print function at the bottom. Take note of the order these are being created. This makes the data read out in the order it happens during a day.

```
33 print ("Morning Nautical Twilight", mornnaut)
34 print ("Morning Civil Twilight", morncivil)
35 print ("Sunrise", srise)
36 print ("Sunset", sset)
37 print ("Evening Civil Twilight", evencivil)
38 print ("Evening Nautical Twilight", evennaut)
39
40 print ("Moonrise", mrise)
41 print ("Moonset", mset)
42 print ("Moon fraction", mphase)
```

Run your program to test it.

Define morning and evening astronomical twilight. Astronomical Twilight takes place when the difference between the horizon and sun is 18 degrees.

Don't forget to add your print function at the bottom in the correct order.

```
30 mornnaut=ephem.to_timezone(pos1.previous_rising(ephem.Sun(pos1), use_center=True),tz)
31 evennaut=ephem.to_timezone(pos1.next_setting(ephem.Sun(pos1), use_center=True),tz)
32
33 pos1.horizon='-18'
34 mornastro=ephem.to_timezone(pos1.previous_rising(ephem.Sun(pos1), use_center=True),tz)
35 evenastro=ephem.to_timezone(pos1.next_setting(ephem.Sun(pos1), use_center=True),tz)
36
37
38 print ("Morning Astronomical Twilight", mornastro)
39 print ("Morning Nautical Twilight", mornnaut)
40 print ("Morning Civil Twilight", morncivil)
41 print ("Sunrise", srise)
42 print ("Sunset", sset)
43 print ("Evening Civil Twilight", evencivil)
44 print ("Evening Nautical Twilight", evennaut)
45 print ("Evening Astronomical Twilight", evenastro)
46
```

Run your program. In the result window you should see a list of the exact times for each Twilight in your area.

Challenges:

Information Presentation

How could you display this data for a home or business owner?

Swap planets

In your program Mars was chosen as the variable planet. What happens if you use other planets?

Different twilights

Any person who is looking at your data may not understand, or have knowledge of, the 3 different types of twilights.

What could you add to your program to convey the meaning?

Congratulations you're a Moonhack changemaker! You can try one of our other projects or try one of the challenges.

Don't forget to talk to an adult about registering your participation and downloading your certificate at moonhack.com