

## **PYTHON ~ The alternative to JAVA, and easier to learn**

**June 12, 2013**

<http://www.python.org/>

Download and install Python from here.

1. Python itself

<http://www.python.org/download/releases/2.5.1/>  
use Quick Links, Windows Installer

2. Python Scripter

<http://www.mmm-experts.com/>

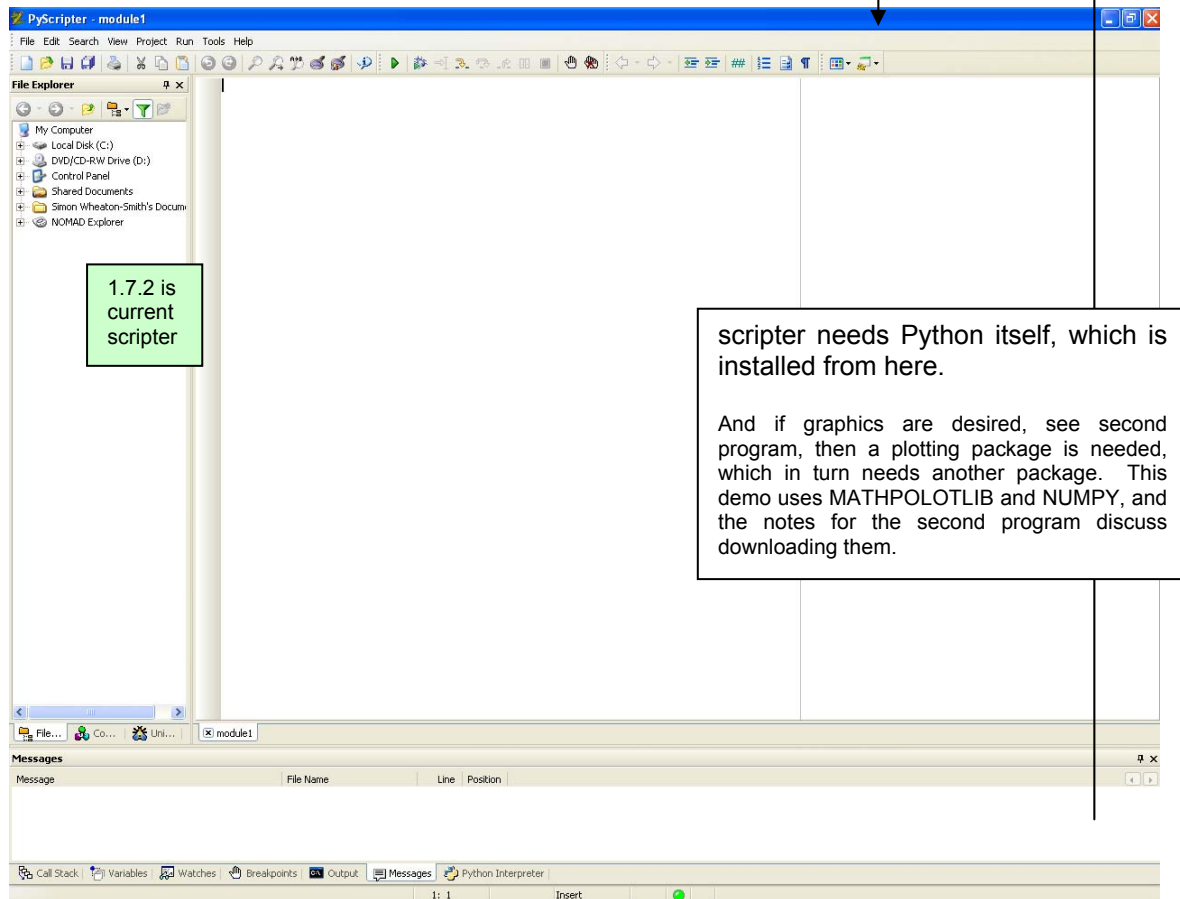
<http://www.python.org/doc/>

Documentation  
Useful introduction

<http://wiki.python.org/moin/BeginnersGuide>

an IDE for Python

2.7.1 is  
current  
stable  
version



When Python itself is installed, and then the PyScripter, you now have an IDE (Interactive Development Environment).

**NOTE:** The text versions work on Windows XP sp1 and 2, the graphical versions may not work on SP 1.

March 2011 updates in green. Under Windows Vista 64 bit, while python runs, pyScripter fails when you RUN anything.

January 23, 2008 ~ this may be distributed freely provided the web site credit  
and this notice are retained ~ updated June 12, 2013, February 7, 2017

## **NON GRAPHICAL (TEXT OR CONSOLE) APPLICATIONS**

Here is the Pascal program for the h-dial converted to Python, as a console application.

```
# program HorizontalDial;

# This program runs using the DOS feature of Windows, rather than the more modern
# GUI or graphical user interface. This is to make the progrsm clear and simple.

# NOTE: The body of the loop is indented: indentation groups statements.

import math                # this import is needed for the ... math.sin(xx) etc

print "*----- Horizontal Dial -----*"
print "www.illustratingshadows.com  hdialPyConsole.py"
print "*----- January 12, 2008 -----*"

lat  = float(raw_input('Latitude: '))
lng  = float(raw_input('Longitude: '))
ref  = float(raw_input('Legal Meridian: '))
print "Design latitude:      ", lat
print "Design longitude:     ", lng
print "Reference meridian:   ", ref

# display the corrections
corm = 4*(lng - ref)
print "Correction to shadow ",
print corm,
print " minutes."
print "Correction to shadow ",
print corm/60,
print " hours."
print " "

# do this one time
slat = math.sin(math.radians(lat))

print ( 'Corrected hour line angles')
print ( 'morning hours first' )
i = -6
while i < 7:
    if i==0:
        print "noon"
    if i==1:
        print "noon"
    # get the hour angle of the sun
    ii = -i
    j  = 15 * (ii+(corm/60))
    # get the resulting hour line angle ~ atan(sin(lat)*tan(hr*15)
    hlat= slat * math.tan(math.radians(j))
    # get the hour line angle back to degrees
    hla = math.degrees( math.atan(hlat) )
    print "Hour:  ", i, " Hour angle:  ", j , " Hour line angle:  ", hla
    i = i + 1
print ( 'afternoon hours last' )
print " "
print '*** END ***'
okok = raw_input('Any key to end. ')
# END
```

This program as "hdialPyConsole.py" can be double clicked and (through Python itself) a DOS window will be generated and the program will run.

It can also be "RUN" from the IDE.

## GRAPHICAL ADDITIONS

**NOTE:** The graphical versions may not work on Windows XP service pack 1, SP 2 may be required.

A plotting package is needed, the most popular is MATPLOTLIB and can be downloaded free of charge from:-

<http://sourceforge.net/projects/matplotlib>

This is called MATPLOTLIB and of course it needs installing. It installs as long as Python itself is installed. And it in turn needs "numpy", even for these simple programs, which is available from:-

<http://www.scipy.org/>

And, NUMPY installs as long as Python itself is installed.

Once these are installed, then graphics are possible. However, the code should be run by double clicking the program in its folder rather than from the IDE, because the IDE has a good chance of crashing.

Some notes on graphics are found at:-

<http://matplotlib.sourceforge.net/tutorial.html>  
<http://matplotlib.sourceforge.net/matplotlib.pyplot.html#-axis>

**NOTE:** The graphical depictions should be double checked for aspect ratio. Code exists to ensure the correct aspect ration, however, the screen may display a slightly different aspect ratio.

The graphical program that runs, in this case under Windows XP, is shown below.

```
# HorizontalDial: hdialPyGraphical.py : January 13, 2008 1006 mst
# Simon Wheaton-Smith FRI, MBCS, CITP
# www.illustratingshadows.com

# This program runs using a graphical display.
# This should not be run from an IDE as the IDE may crash,
#     instead locate the folder and program, and double click it.

# NOTE: The body of ifs and loop must be indented: indentation groups statements.

# REGULAR math library mandatory import
import math          # this import is needed for the ... math.sin(xx) etc

# SPECIAL library for graphics using MATPLOTLIB and it in turn uses NUMPY
from pylab import *   # these are needed for graphics displays
from matplotlib.lines import Line2D
from matplotlib.patches import Rectangle

# GENERAL introductory text
print "*----- Horizontal Dial -----*"
print "www.illustratingshadows.com  hdialPyConsole.py"
print "NOTE: run by double clicking in a folder, and"
print "      not from an IDE, as the IDE may crash."
#print "NOTE: the graphical display area may not have"
#print "      good aspect ratio, you may resize it."
print "*----- January 13, 2008 -----*"

# READ parameters for the dial plate
lat  = float(raw_input('Latitude: '))
lng  = float(raw_input('Longitude: '))
ref  = float(raw_input('Legal Meridian: '))
print "Design latitude:      ", lat
print "Design longitude:      ", lng
print "Reference meridian:     ", ref
```

```
# DISPLAY the corrections for the dial
corm = 4*(lng - ref)
print "Correction to shadow ",
print corm,
print " minutes."
print "Correction to shadow ",
print corm/60,
print " hours."
print " "

# ONE TIME math calculation
slat = math.sin(math.radians(lat))

# LIST the table of hours and hour angles
print ( 'Corrected hour line angles')
print ( 'morning hours first' )
i = -6
while i < 7:
    if i==0:
        print "noon"
    if i==1:
        print "noon"
    # get the hour angle of the sun
    ii = -i # ii is now the hour with the sign reversed
    j = 15 * (ii+(corm/60)) # j is now the local hour angle for the time
    # get the resulting hour line angle ~ atan(sin(lat)*tan(hr*15)
    hlat= slat * math.tan(math.radians(j))
    # # hlat is the tan of the hour line angle
    # # hla is the hour line angle in degrees
    hla = math.degrees( math.atan(hlat) )
    print "Hour: ", i, " Hour angle: " , j , " Hour line angle: ", hla
    i = i + 1 # go to the next hour
print ( 'afternoon hours last' )
print " "
print '*** To quit, close the graphical display ***'

# START the graphical depiction
ax = gca() # get the current axes "instance"
# # and you can set parms if you like
# read: http://matplotlib.sourceforge.net/matplotlib.pyplot.html#-gca
i = -5 # eg: -3 is 12-3 is 0900,
# eg: +3 is 12+3 is 15 is 3pm
while i < 7:
    # get the hour angle of the sun
    ii = -i # ii is now the hour with the sign reversed
    j = 15 * (ii+(corm/60)) # j is now the local hour angle for the time
    # get the resulting hour line angle ~ atan(sin(lat)*tan(hr*15)
    hlat= slat * math.tan(math.radians(j))
    # # hlat is the tan of the hour line angle
    # # hla is the hour line angle in degrees
    hla = math.degrees( math.atan(hlat) )
    # define dial center
    xc = 2.45 # determine dial center by trial and
    yc = 0.8 # error with the graph area x,y=2.5,0.8

    # derive line data - 9am to 3pm has y scale of 4.0

    if i < -3 and hla >0:
        # QUADRANT 1 is A M or morning hours
        # before 0900 and with a positive hour line angle
        # left hand side of dial plate
        # has an x scale of 70% of 4.0
        xs = -4.0*0.6 # define dial plate limits
        xe = xs # change the x scale to keep in sight
        ye = xs / hlat # 1/tan is same as cotan
        plot ( [xc, xc+xe], [yc, (yc-ye)] )
        # the plot goes by default to the current instance
        ax.text(xc+0.1+xe, (yc-ye), 'am' )
        # this sends text to "ax" the current graphical "instance"
        ax.text(xc+0.1+xe, (yc-0.15-ye), 12+i )
        ax.text(xc+0.1+xe, (yc-0.3-ye), int(hla) )
    elif i > 3 and hla <0:
        # QUADRANT 3 is P M or afternoon hours
        # after 1500 and with a negative hour line angle
        # right hand side of dial plate
        xs = 4.0*0.5 # define dial plate limits
        ys = xs
        xe = xs
        ye = -ys / hlat
        plot ( [xc, xc+xe], [yc, yc+ye] )
        ax.text(xc+xe+0.1, yc+ye, 'pm' )
        ax.text(xc+xe+0.1, yc+ye-0.15, 12+i )
```

```

        ax.text(xc+xe+0.1, (yc+ye-0.3), int(hla) )
    elif i >=-3 and i <= 3:
        # M I D D A Y
        # on or after 0900 but less than or equal to 1500
        # top if dial plate
        xs = 4.0          # define dial plate limits
        ys = xs
        ye = ys
        xe = -ys * hlat
        plot ( [xc, xc+xe], [yc, yc+ye])
        ax.text(xc+xe, yc+ye, '24Hr' )
        ax.text(xc+xe+0.1, yc+ye-0.15, 12+i )
        ax.text(xc+xe+0.1, yc+ye-0.3, int(hla) )
    else:
        i = i              # do nothing
        # and bump the hour
        i = i + 1

#xc = 2.5
#yc = 0.8
plot ([xc-2.5, xc-2.5], [yc, yc+4.2], 'b') # left side of a box
plot ([xc+2.4, xc+2.4], [yc, yc+4.2], 'b') # right side of a box
plot ([xc-2.5, xc+2.4], [yc, yc ], 'b')    # bottom side of a box
plot ([xc-2.5, xc+2.4], [yc+4.2, yc+4.2 ], 'b') # top of a box
#ax.text(xc-2.5, yc-0.2, 'Graph may not be correct aspect ratio, resize panel.')
ax.text(xc-2.5, yc-0.4, 'H-dial file name: hdialPyGraphical.py')
ax.text(xc-2.5, yc-0.6, 'www.illustratingshadows.com Simon Wheaton-Smith')
ax.text(xc-2.5, yc-0.8, 'Lat:                      Long:                      Ref:')
ax.text(xc-2.0, yc-0.8, lat)
ax.text(xc-1 , yc-0.8, lng)
ax.text(xc+0.02,yc-0.8, ref)

axis([0, 5, 0, 5])          # manages the graphics box
axis('equal')               # makes angles correct aspect ratio
axis('off')
show()

okok = raw_input('Any key to end. ')
# END

```

The textual output is shown below.

```

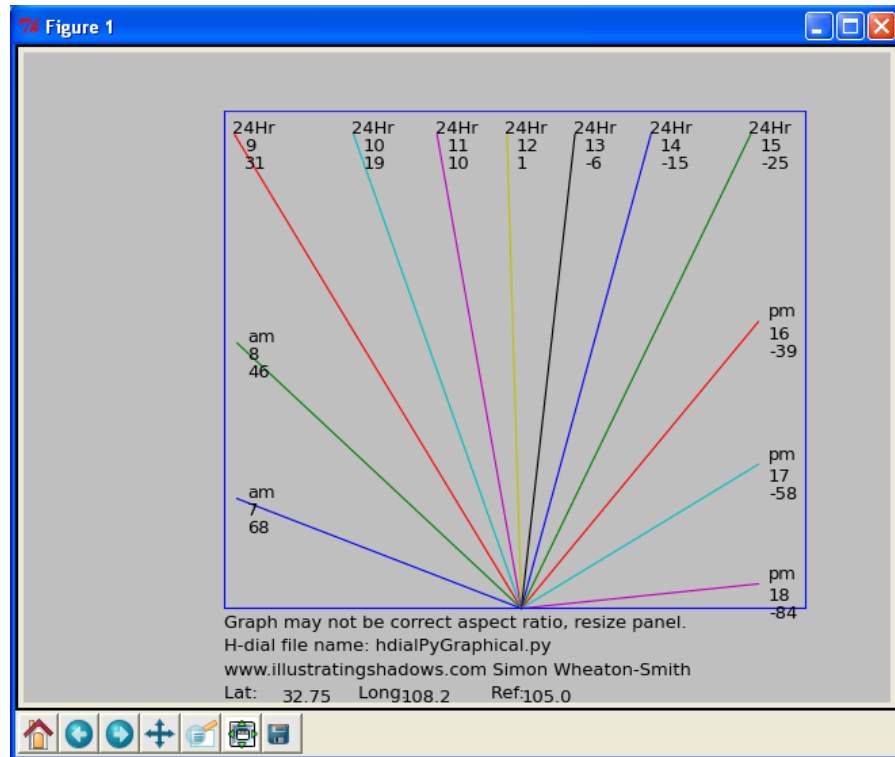
*----- Horizontal Dial -----*
www.illustratingshadows.com  hdialPyConsole.py
NOTE: run by double clicking in a folder.  and
      not from an IDE.  as the IDE may crash.
NOTE: the graphical display area may not have
      good aspect ration.  you may resize it.
*----- January 13, 2008 -----*
Latitude: 32.75
Longitude: 108.2
Legal Meridian: 105
Design latitude: 32.75
Design longitude: 108.2
Reference meridian: 105.0
Correction to shadow 12.8 minutes.
Correction to shadow 0.213333333333 hours.

Corrected hour line angles
morning hours first
Hour: -6 Hour angle: 93.2 Hour line angle: -84.0995382402
Hour: -5 Hour angle: 78.2 Hour line angle: 68.8846794286
Hour: -4 Hour angle: 63.2 Hour line angle: 46.9620965859
Hour: -3 Hour angle: 48.2 Hour line angle: 31.1759103378
Hour: -2 Hour angle: 33.2 Hour line angle: 19.4941552534
Hour: -1 Hour angle: 18.2 Hour line angle: 10.0853509826
noon
Hour: 0 Hour angle: 3.2 Hour line angle: 1.73239238779
noon
Hour: 1 Hour angle: -11.8 Hour line angle: -6.4479493558
Hour: 2 Hour angle: -26.8 Hour line angle: -15.2838376483
Hour: 3 Hour angle: -41.8 Hour line angle: -25.8124505256
Hour: 4 Hour angle: -56.8 Hour line angle: -39.5803910089
Hour: 5 Hour angle: -71.8 Hour line angle: -58.7103909709
Hour: 6 Hour angle: -86.8 Hour line angle: -84.0995382402
afternoon hours last

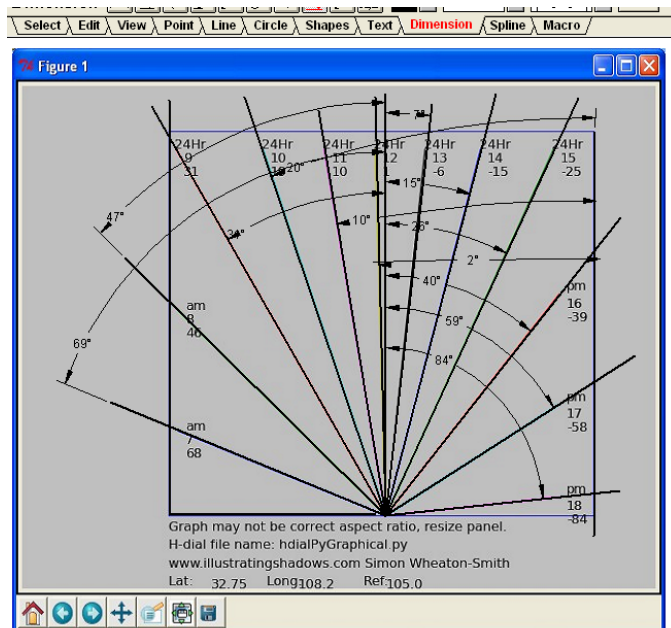
*** To quit, close the graphical display ***
Any key to end.

```

The graphical display is shown below.

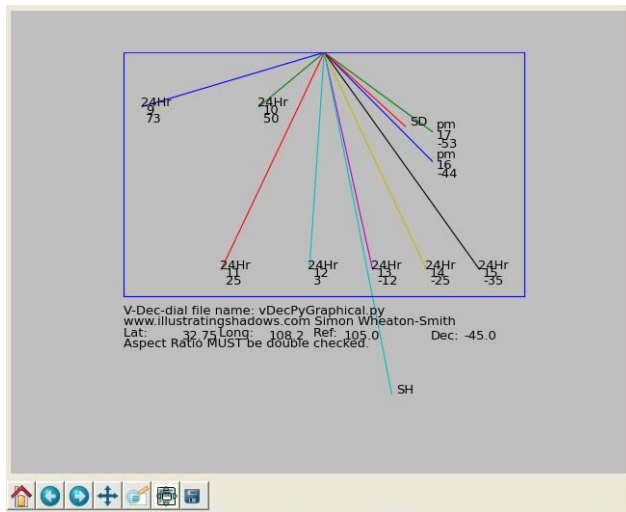


The angles were validated with DeltaCAD. shown below.



NOTE: Python has much in common with Java, however the learning curve was much faster. Python, like Java, can be used for web applications and the like.

There is code for the vertical decliner also. This works for declinations other than 0. For declinations of 0, there is the usual divide by 0 problem, which I chose not to test for. Simple, and done in the other programs, however the point here was to have simple coding.



```

C:\python\python.exe
----- Vertical Declining Dial -----
FROM: www.illustratingshadows.com vDecPyConsole.py
NOTE: run by double clicking in a folder, and not in
an IDE, as the IDE may crash.
----- January 13, 2008 -----
Latitude: 32.75
Longitude: 108.2
Legal Meridian: 105
Wall declination [-SW][+SE]: -45
Design latitude: 32.75
Design longitude: 108.2
Reference meridian: 105.0
Wall declination : -45.0
Faces SW
Correction to shadow 12.8 minutes.
Correction to shadow 0.213333333333 hours.
Corrected hour line angles
morning hours first
Hour: -5 Hour angle: 78.2 Hour line angle: -74.4011311017
Hour: -4 Hour angle: 63.2 Hour line angle: -88.27414075
Hour: -3 Hour angle: 48.2 Hour line angle: 73.4641444426
Hour: -2 Hour angle: 33.2 Hour line angle: 50.3077993858
Hour: -1 Hour angle: 18.2 Hour line angle: 25.4385749997
noon
Hour: 0 Hour angle: 3.2 Hour line angle: 3.9227599594
noon
Hour: 1 Hour angle: -11.8 Hour line angle: -12.5848988631
Hour: 2 Hour angle: -26.8 Hour line angle: -25.2611319958
Hour: 3 Hour angle: -41.8 Hour line angle: -35.631610805
Hour: 4 Hour angle: -56.8 Hour line angle: -44.8571105645
Hour: 5 Hour angle: -71.8 Hour line angle: -53.8237811805
afternoon hours last
*** To quit, close the graphical display ***
SD <style distance> : -47.7086915458
SH <style height> : 36.4915615522

```

If a declination of 0 is required, then the horizontal dial can be used, reversing the longitude and reference longitude, and thusly the hours. Alternatively, the vdial program can be used. The aspect ratio must be double checked.

## **PYSCRIPTER**

PyScripter works on XP SP1 and SP2 however graphics only work on SP2

Under Windows Vista 64 bit, PyScripter will crash.

Not tested under Windows 8.

Other IDE systems for Python may exist.

## **OTHER SYSTEMS**

Other IDE systems

ACTIVE STATE

<http://www.activestate.com/Products/activepython/>

this package runs on Windows XP SP1 in text (non graphical mode).

## **OTHER SYSTEMS USING PYTHON**

The free FreeCAD CAD program uses Python in its scripting.

FreeCAD is a full CAD system and is thus very appropriate for sundial design.

The free Blender Modeling program uses Python in its scripting.

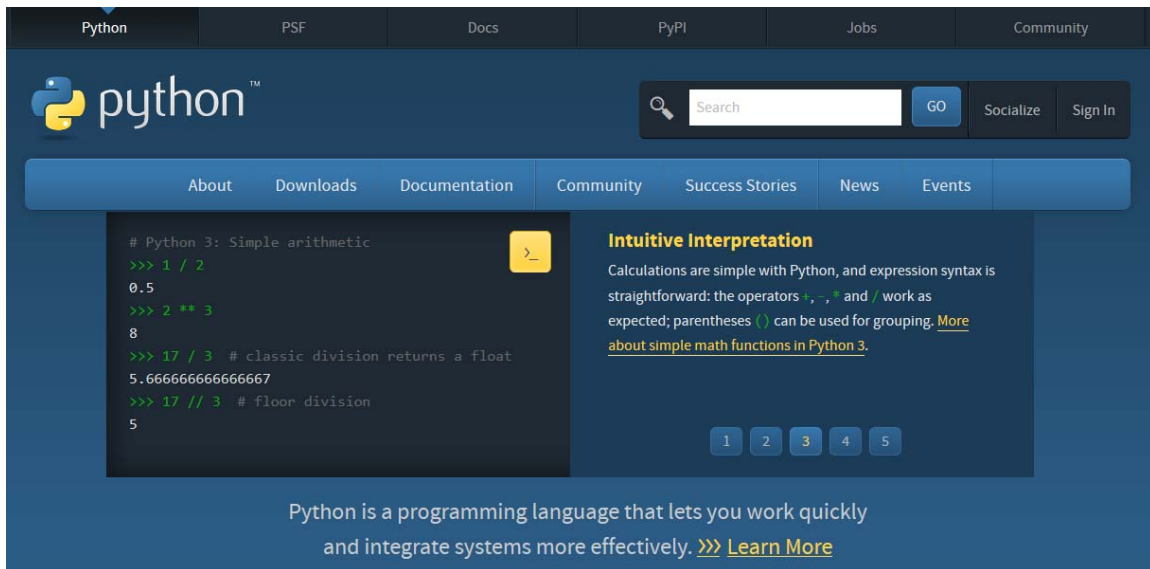
Blender has many of the CAD attributes, however, it is more of a modeling system and this, while fun, is less appropriate for sundial design.



## PYTHON ~ Same as in earlier pages, just a different IDE

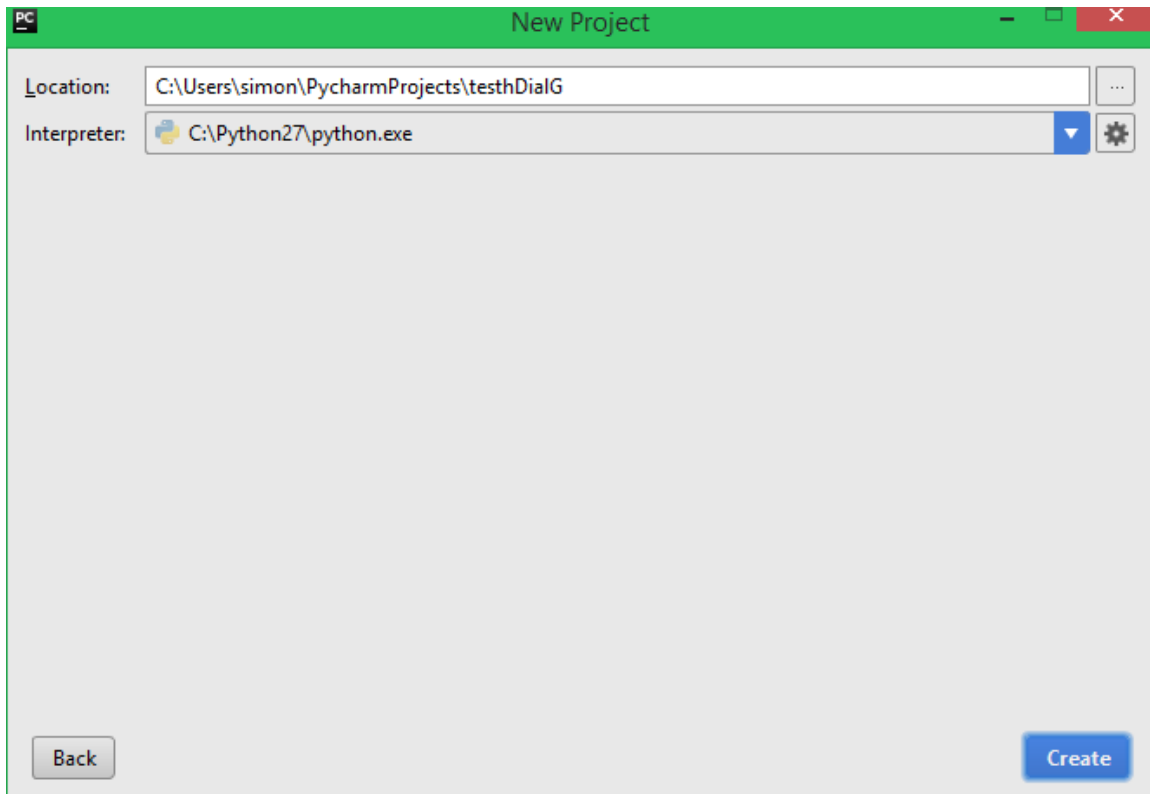
February 7, 2017

<https://www.python.org/>



<http://www.jetbrains.com/pycharm>





This IDE used Turtle graphics, quite different from the previous pages.

```
n Wheaton-Smith FRI, MBCS, CITP
# www.illustratingshadows.com

# This program runs using a graphical display.
# This should not be run from an IDE as the IDE may crash,
#     instead locate the folder and program, and double click it.

# NOTE: The body of ifs and loop must be indented: indentation groups statements.

# REGULAR math library mandatory import
import math # this import is needed for the ... math.sin(xx) etc
import turtle

# GENERAL introductory text
print "*----- Horizontal Dial -----*"
print "python, pycharm . . . using Turtle graphics"
print "NOTE: run by double clicking in a folder, and"
print "    not from an IDE, as the IDE may crash."
# print "NOTE: the graphical display area may not have"
# print "    good aspect ratio, you may resize it."
print "*----- January 13, 2008 -----*"

# READ parameters for the dial plate
lat = float(raw_input('Latitude: '))
lng = float(raw_input('Longitude: '))
ref = float(raw_input('Legal Meridian: '))
print "Design latitude:    ", lat
print "Design longitude:    ", lng
print "Reference meridian:   ", ref
```

January 23, 2008 ~ this may be distributed freely provided the web site credit  
and this notice are retained ~ updated June 12, 2013, February 7, 2017

```
# DISPLAY the corrections for the dial
corm = 4 * (lng - ref)
print "Correction to shadow ",
print corm,
print " minutes."
print "Correction to shadow ",
print corm / 60,
print " hours."
print " "

# ONE TIME math calculation
slat = math.sin(math.radians(lat))

# LIST the table of hours and hour angles
print ('Corrected hour line angles')
print ('morning hours first')
i = -6
while i < 7:
    if i == 0:
        print "noon"
    if i == 1:
        print "noon"
    # get the hour angle of the sun
    ii = -i # ii is now the hour with the sign reversed
    j = 15 * (ii + (corm / 60)) # j is now the local hour angle for the time
    # get the resulting hour line angle ~ atan(sin(lat)*tan(hr*15)
    hlat = slat * math.tan(math.radians(j))
    # # hlat is the tan of the hour line angle
    # # hla is the hour line angle in degrees
    hla = math.degrees(math.atan(hlat))
    print "Hour: ", i, " Hour angle: ", j, " Hour line angle: ", hla
    i = i + 1 # go to the next hour
print ('afternoon hours last')
print " "
print '*** To quit, close the graphical display ***'

# START the graphical depiction

i = -5 # eg: -3 is 12-3 is 0900,
while i < 6: # eg: +3 is 12+3 is 15 is 3pm
    # get the hour angle of the sun
    ii = -i # ii is now the hour with the sign reversed
    j = 15 * (ii + (corm / 60)) # j is now the local hour angle for the time
    # get the resulting hour line angle ~ atan(sin(lat)*tan(hr*15)
    hlat = slat * math.tan(math.radians(j))
    # # hlat is the tan of the hour line angle
    # # hla is the hour line angle in degrees
    hla = math.degrees(math.atan(hlat))

    # http://www.eg.bucknell.edu/~hyde/Python3/TurtleDirections.html
    turtle.goto(0, 0)
    turtle.degrees()

    turtle.width(1)
    if i > 0:
        turtle.width(2)
    if i == 0:
        turtle.width(5)

    #turtle.setheading(hla)
    #turtle.forward(150)

    # or ...

    turtle.left(hla+90)
    turtle.forward(300)
    turtle.goto(0, 0)
    turtle.right(hla+90)
```

```
# and bump the hour
i = i + 1

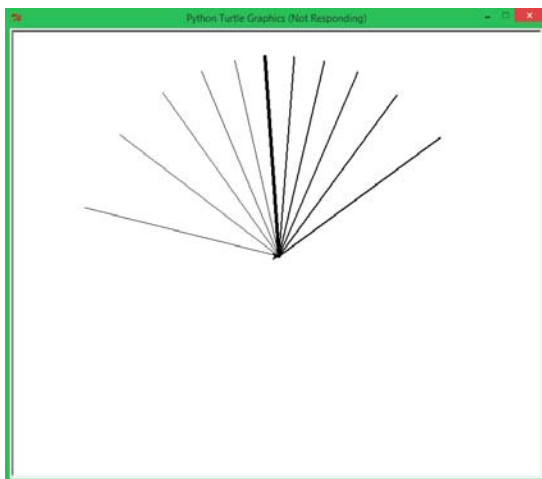
okok = raw_input('Any key to end. ')
# END
```

The console is shown below:-

```
C:\Python27\python.exe C:/Users/simon/PycharmProjects/testhDialG/testhdialg2017.py
*----- Horizontal Dial -----*
python, pycharm . . . using Turtle graphics
NOTE: run by double clicking in a folder, and
      not from an IDE, as the IDE may crash.
*----- January 13, 2008 -----*
Latitude: 33.5
Longitude: 112.1
Legal Meridian: 105
Design latitude:      33.5
Design longitude:     112.1
Reference meridian:   105.0
Correction to shadow  28.4 minutes.
Correction to shadow  0.473333333333 hours.

Corrected hour line angles
morning hours first
Hour:  -6  Hour angle:  97.1  Hour line angle: -77.2829880301
Hour:  -5  Hour angle:  82.1  Hour line angle:  75.8878456001
Hour:  -4  Hour angle:  67.1  Hour line angle:  52.5719153294
Hour:  -3  Hour angle:  52.1  Hour line angle:  35.3364225578
Hour:  -2  Hour angle:  37.1  Hour line angle:  22.6569580349
Hour:  -1  Hour angle:  22.1  Hour line angle:  12.6322947239
noon
Hour:   0  Hour angle:   7.1  Hour line angle:   3.93274718712
noon
Hour:   1  Hour angle: -7.9  Hour line angle: -4.37959570538
Hour:   2  Hour angle: -22.9  Hour line angle: -13.1239139018
Hour:   3  Hour angle: -37.9  Hour line angle: -23.2518047762
Hour:   4  Hour angle: -52.9  Hour line angle: -36.1216506216
Hour:   5  Hour angle: -67.9  Hour line angle: -53.6582271933
Hour:   6  Hour angle: -82.9  Hour line angle: -77.2829880301
afternoon hours last

*** To quit, close the graphical display ***
Any key to end.
```



To the left is the turtle graphics depiction. I have not as yet been able to simply display text in that area, so I cannot show the hours.