A. Why I wrote this program

After seeing the digital planetarium exhibit at the California Academy of Sciences, I decided that I wanted to paint the night sky on my ceiling and walls using glow-in-the-dark paint. To do this, I wrote a Python program to chart each star's position and magnitude relative to a single point in space. I plan to complete the project come summer 2013.

B. Concerning the apparent magnitude and shape of each star

In 1856, the English astronomer Sir Norman Pogson suggested to standardize the measurement of the apparent magnitude of stars. He suggested that a star of magnitude 1 be 100 times as bright as a star of magnitude 6 (which represent the very faintest of stars). Therefore, a star of magnitude 1 is about 2.512 (the 5th root of 100) times brighter than a star of magnitude 2. Currently, the star Vega represents a magnitude of 0. While I use this scale in my program, Polaris (the North Star) is used to set the standard, at a magnitude of about 2. Because magnitudes are represented by area, it is virtually impossible to paint a star that is 100 time the area of the faintest star in the sky. So, I allowed the upper and lower bound of magnitudes to be variable. Instead of Polaris being about 40 times brighter than a star of magnitude 6, the relative radii of Polaris and a star of magnitude 6 can be adjusted. I also included a linear scale to keep stars of negative magnitude (extremely bright stars) from being blown out of proportion if need be.

The shapes of all the stars are ellipses, something that is necessarily to keep the stars in perspective.
----for more information see http://en.wikipedia.org/wiki/Apparent_magnitude

C. What is included in the folder

-coorlist.txt
-this is a list of 9,111 visible stars downloaded from http://www-kpno.kpno.noao.edu/Info/Caches/Catalogs/BSC5/catalog5.html

-parameters.txt, parameters2.txt
-these include information that will be discussed later

-planetarium.py
-this is the program itself

-sky.html, sky2.html
-these are vector files of the stars. they can be modified

-ursamaj.txt
-contains the coordinates of the Ursa Major constellation

-ursamajmin.txt
-contains the coordinates of both the Ursa Major and Ursa Minor constellations

-ursamin.txt
-contains the coordinates of the Ursa Minor constellation

D. How to use this program

(quotation marks are not to be included when typing)

Macs have Python already. To download Python for windows go to: http://www.python.org/download/

1. drag folder into your desktop (call it "night_sky")
2. open the application "Terminal" if your using a Mac, "Command Prompt" for Windows

3. from now on, hit the "ENTER" key after typing something into Terminal

4. type "cd desktop"
5. type "cd night_sky"

[all this cd business is just going through directories, aka folders]

6. type "python planetarium.py"
7. it will ask "Which file would you like to write to? > "
8. type in "sky.html" or "sky2.html"

[this is the file that will contain the vectorized image of all the stars]

9. it will ask "Run default parameters, run a parameters file, or input parameters? (D, F, or IP) >"

You have encountered a fork in the road

a. if you enter "D" it will write the corresponding vector images of each star into the file you specified above using the default settings built into the program

b. if you enter "F"

it will ask "Which parameter file? > "
-enter "parameters.txt" or "parameters2.txt"

it will then write the stars to whatever file you specified earlier using the parameters specified in the file "parameters.txt" or "parameters2.txt"

for example:

```
filename = ursamajmin.txt
e = 450
f = 450
g = 450
h = 450
i = 450
j = 450
ra_rotation = 10
dec_rotation = 10
radius_of_polaris = 8
```

```
min_radius = 1
lim_mag = 5
kind =   log
stroke_width = 1
fill =  #3874C2
s = 1
allowed_domain = 5
```

coorlist.txt is the list of all the coordinates of the visible stars that you might see on a moonless cloudless night.

To understand what e, f, g, h, i, and j are, take a look at the attached photo called "domains.jpg" Each letter e to j represents a length from the center point to either a wall ceiling or floor along the x, y, and z axes.

The next two lines refer to the "celestial sphere" or the an imaginary sphere on which all the stars lie.

The way that a star is located is similar to longitude and latitude, with the right ascension (ra) equivalent to longitude and declination (dec) equivalent to latitude. While right ascension is generally measured in hours, minutes, and seconds, declination is in degrees, minutes and seconds. But you don't have to worry about that. All you have to know is that whatever value you put after "ra_rotation" is in degrees and rotates the celestial sphere around its z-axis "eastward" where the North Star marks the North Pole. The declination rotates the celestial sphere in the opposite direction. See image "sphere.jpg"

the "radius_of_polaris" can be changed to what ever you want.
the same goes for "min_radius" (which indicates the radius of a star of apparent magnitude of 6)

"lim_mag" limits the stars projected based on magnitude. For example, if 5 is the limiting magnitude, only stars that are of a magnitude of 5 or lower are projected.

"kind" indicates the type of scale used.
entering "log" keeps the traditional logarithmic scale.
entering "k" makes the relationship linear.

"stroke_width" refers to the width of the ellipse's circumference.

"fill" refers to the color. You can enter a hexadecimal if you wish

Don't worry about what "s" does

"allowed_domain" is basically which wall or ceiling you want to project the stars onto.

c. if you enter "IP" you will be able to input custom commands on the fly
The only cool thing is that it can create a hexadecimal for you if you want.
enter "yes" or "no" when you get to the "Any fill? > " part

I would recommend just choosing "F" and modifying the file.

After all this, open "sky.html" or "sky2.html" using the browser of your

```
choice.

Special thanks to Pino Trogu, Mr. Cohen, and Grant Paul.
(and the online resource, "Learn Python the Hard Way")

Francesco Trogu, December 2012
```