Project: MP2
Course: ITM-513
Author: Brian T. Bailey

Project Description:

The objective of this project is to create a Python application that models the planets of our solar system and do some analytics on those objects. The data from each of the nine planets needs to be parsed from separate text files for each planet into Planet objects. The Planet class needs to have attributes for the following data points: mass, diameter, escape velocity, revolution period, and mean surface temperature. The Class also needs to supply a full argument constructor, display method to print planet attributes, and getter and setter methods.

Once the planet data is parsed into Planet objects, they need to be stored in a dictionary where the key is the planet name and the value is the Planet object. The data analytics consist of creating five sorted lists of the planet names in ascending order based on one data point for each list. And finally, all of the results need to be output to the console and written to a file called mp2out.txt.

Installation, Compile and Run-Time Requirements:

This project was written in Python using version 2.7.1. The scripts were written in BBEdit version 10.1.2 on the Macintosh platform. The computer used was a 2.7 GHz dual-core Intel Core i7 13" MacBook Pro with 8GB of RAM running OS X Lion 10.7.4.

This application can be executed by running the main.py Python file in the project directory.

Insights and Expected Results:

One thing I noticed in this project was the majority of my code in my driver consisted of comments and code to output data to the console or write to a file. The parsing of the data, creating the dictionary and sorting the lists took a minimal amount of code. This would have required much more code in another language like Java.

Throughout my projects source files I tried to follow Google's Python Style Guide and the PEP8 style guide as closely as possible. The place where this came up the most was line length. Both guides recommend line lengths of less then 80 characters. I also followed their recommend indenting for multi-line statements.

To parse the data from files I created a function in the planet_utils module called parsePlanetDatafile. I created each data file as a csv file with the first line being a header describing each field. The parsePlanetDatafile method takes the path to a data file as a string argument and returns a Planet object representing that data.

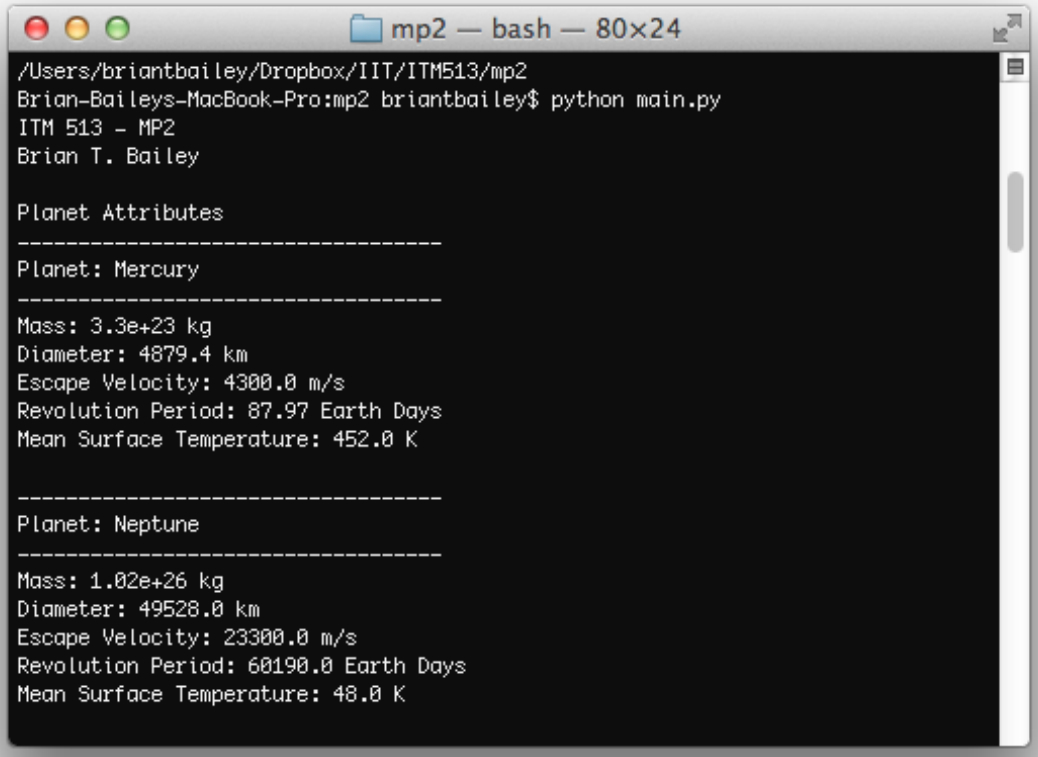
In the driver I defined a list of all the paths to the nine data files. Then in a single line I used a dictionary comprehension with an nested list comprehension, which used that parsePlanetDatafile function, to create the SolarSystem dictionary.

For each of the sorted lists I used basically the same process. I used a list comprehension to build the list. Inside each comprehension I used the sorted function to create a list of the SolarSystem dictionary's values sorted using a lambda function as the key to the sort. The lambda function basically looked at each Planet object's respective attributes for the sort. The outer list comprehension then took the name attribute from the sorted list.

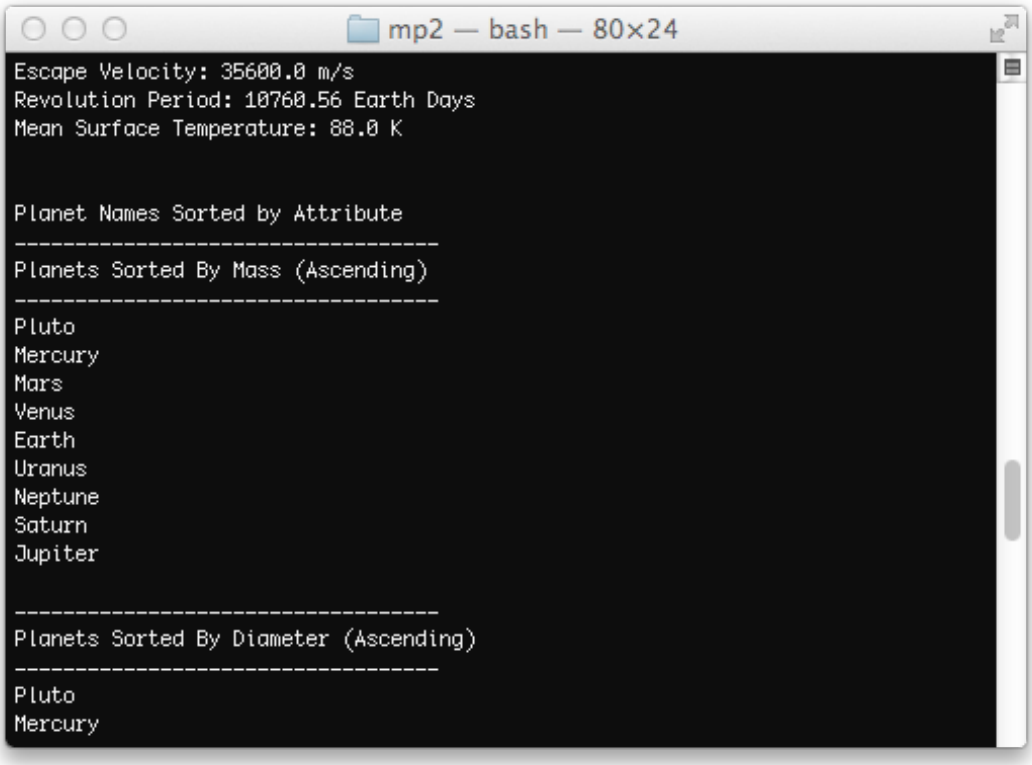
The rest of the functions defined in the planet_utils module are for various outputs. I tried to combine the output to the console and the write to a file in the same function when possible. Most of these functions take the file object as the first argument and the data that will output as the second one.

Screenshots Demonstrating Application:

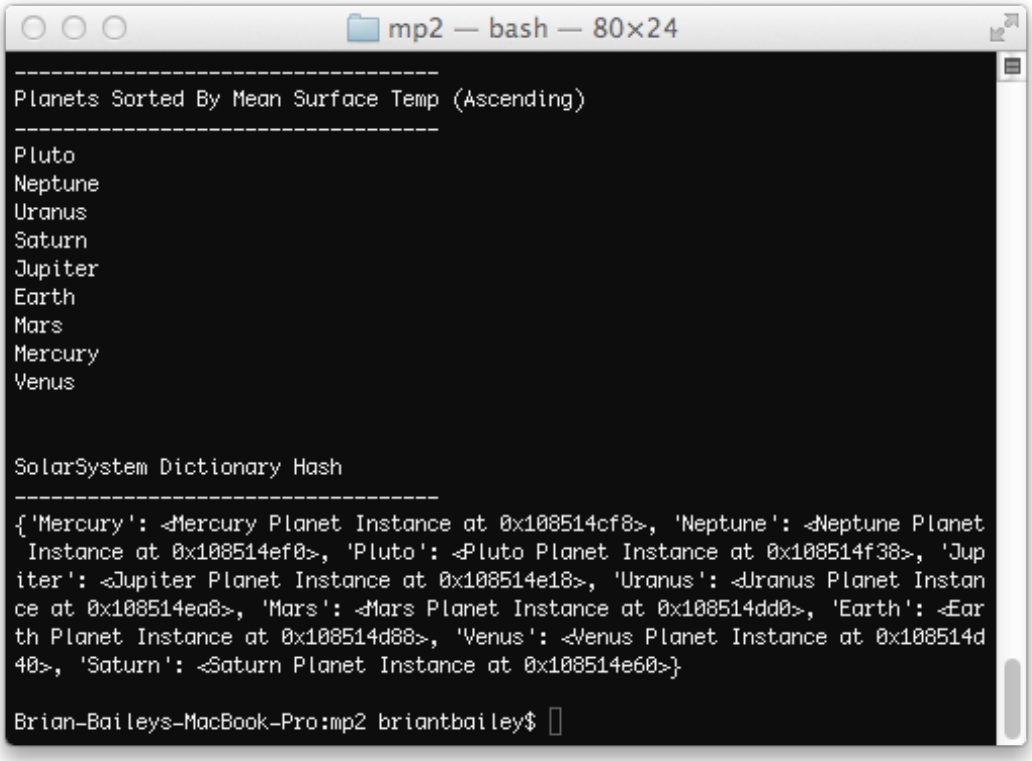
Screenshot showing the application launching and starting to output the Planet attributes to the console.



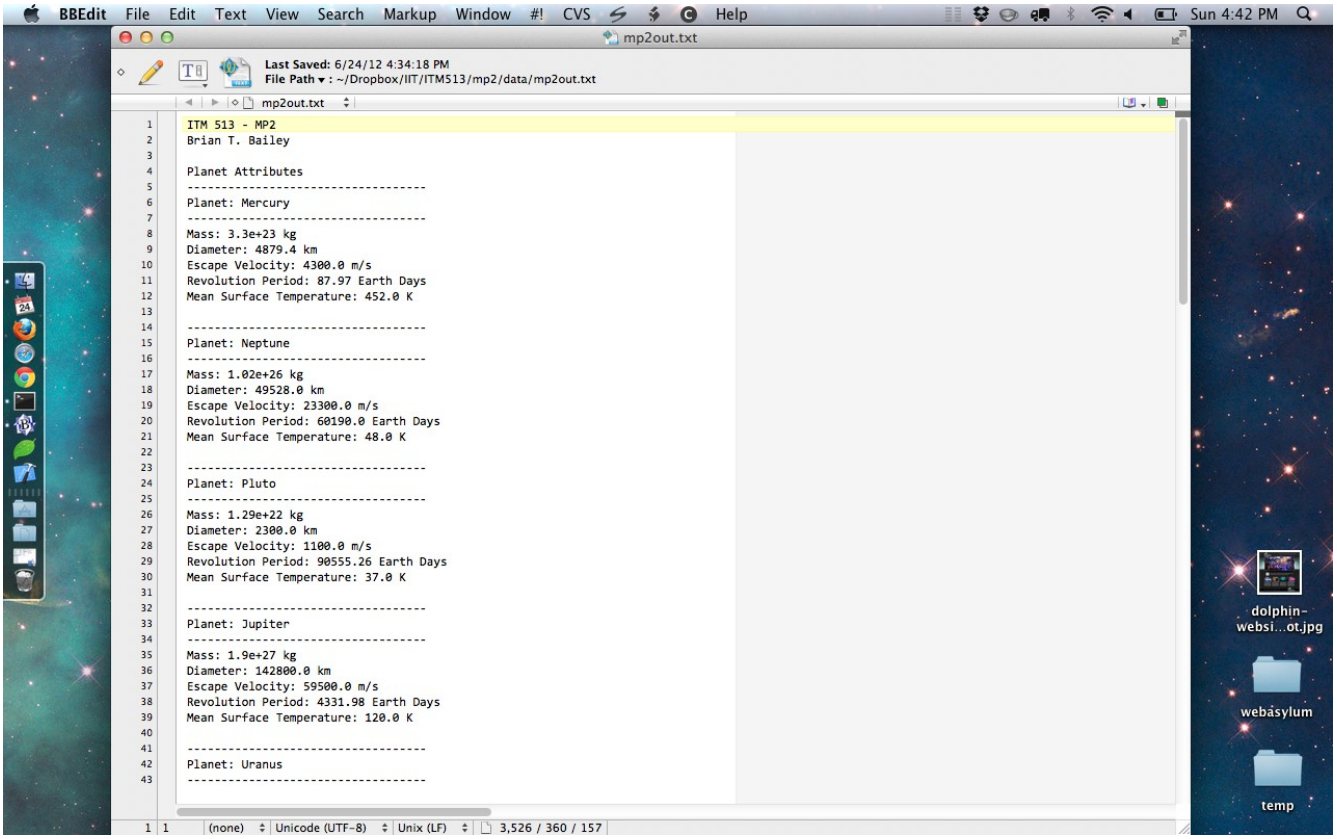
This Screenshot shows the application continuing to run and now is starting to output the sorted lists of Planet names based on the various attributes.

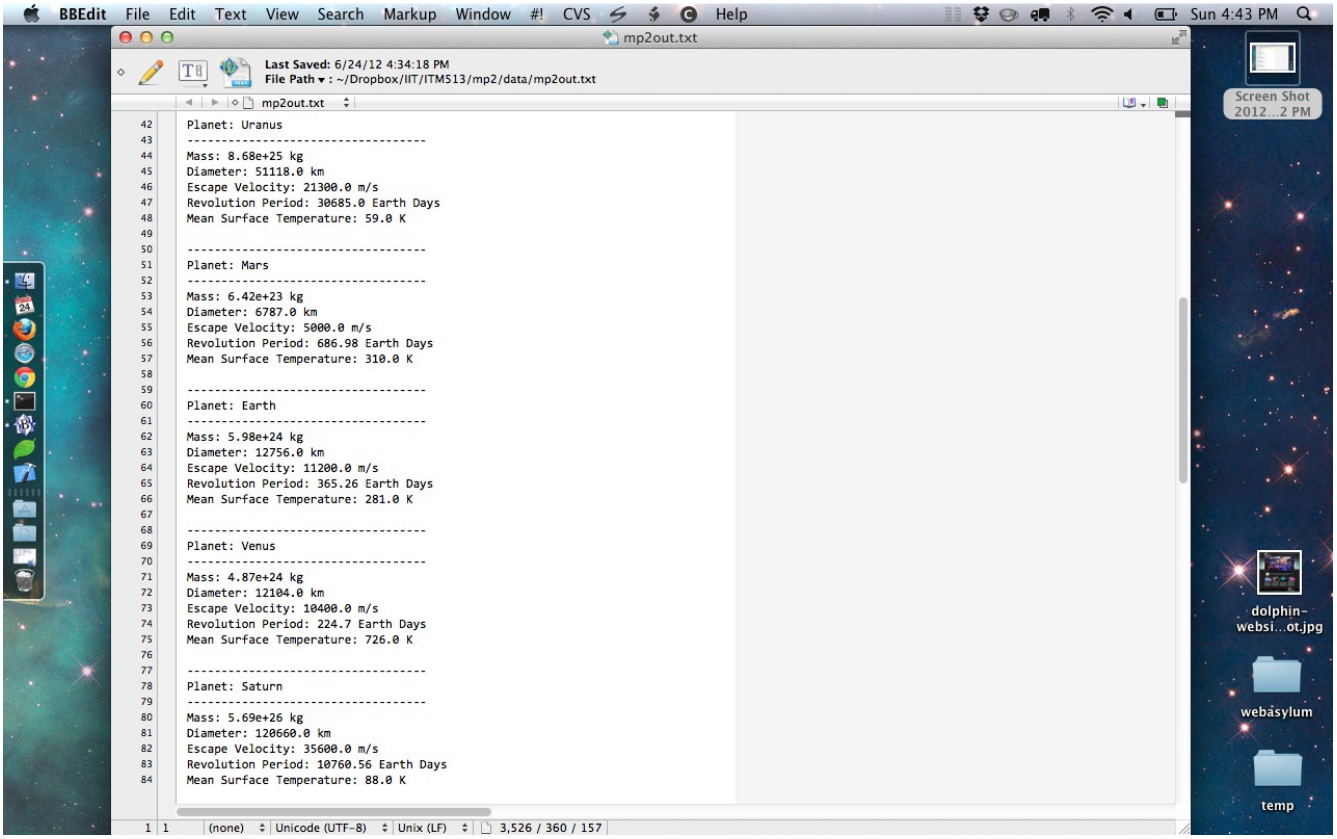


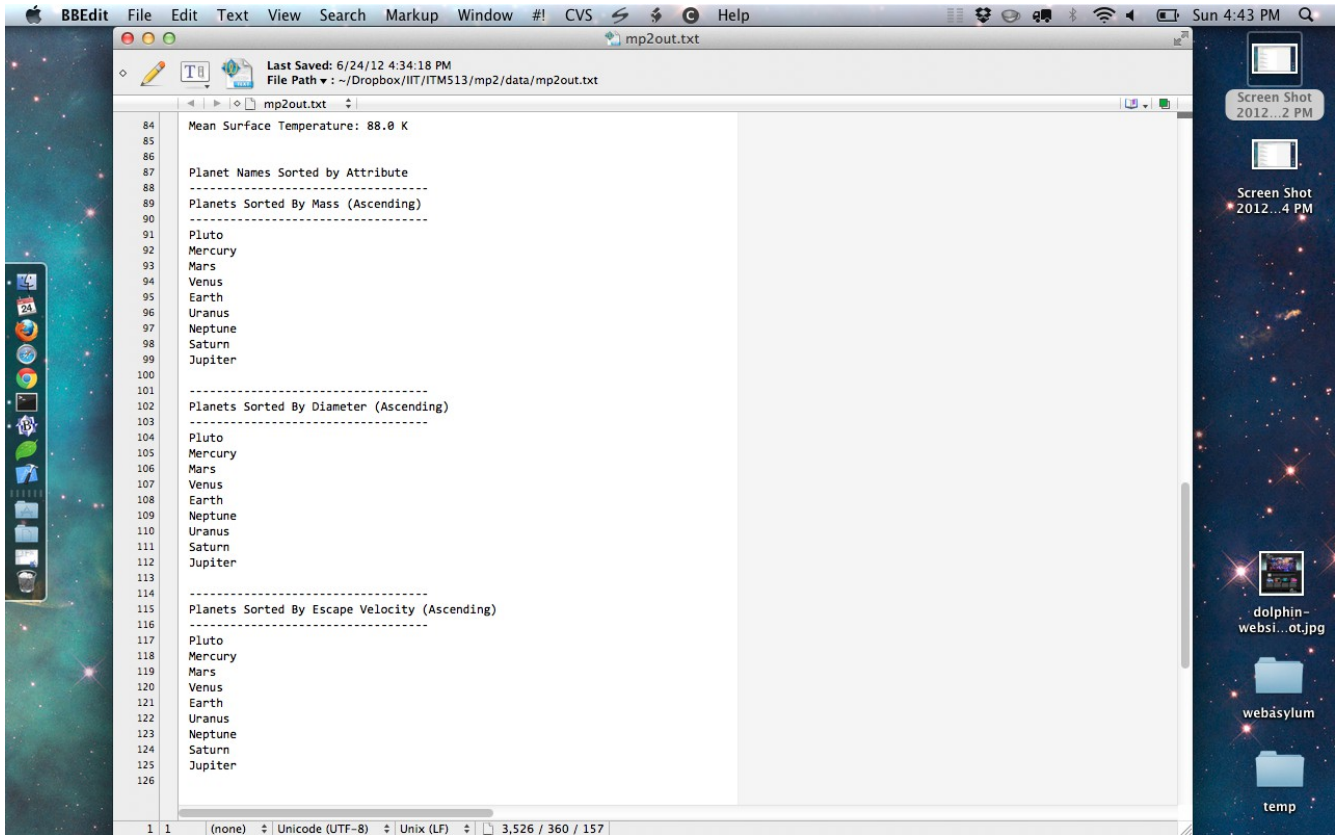
This screenshot shows the application finishing displaying the sorted lists and then outputting the SolarSystem dictionary. Then the application is finished running.

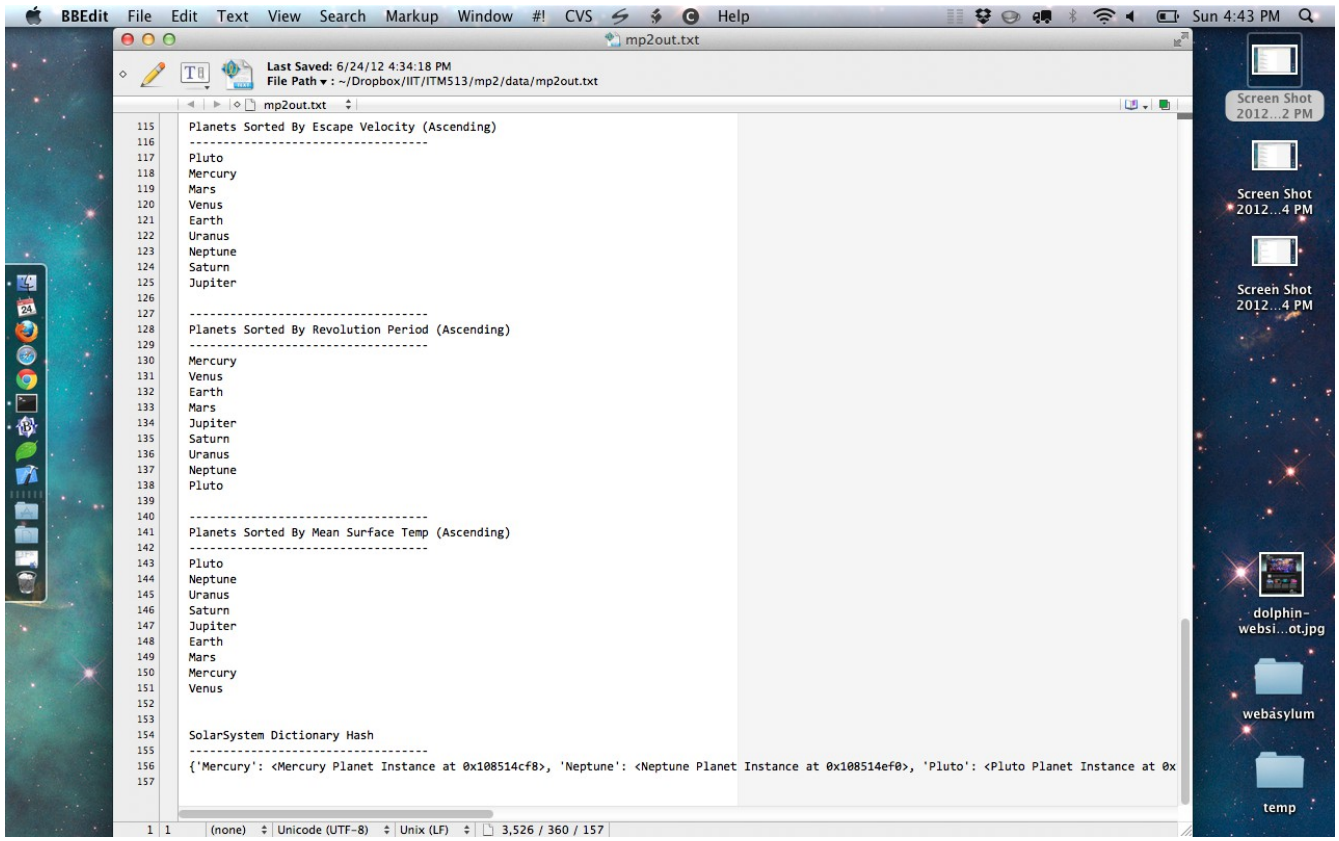


The following 4 screenshots show the mp2out.txt file and its contents.






BBEdit File Edit Text View Search Markup Window #! CVS Help Sun 4:43 PM
mp2out.txt
Last Saved: 6/24/12 4:34:18 PM
File Path : ~/Dropbox/IIT/ITM513/mp2/data/mp2out.txt
mp2out.txt
Mean Surface Temperature: 88.0 K
Planet Names Sorted by Attribute
Planets Sorted By Mass (Ascending)
Pluto
Mercury
Mars
Venus
Earth
Uranus
Neptune
Saturn
Jupiter
Planets Sorted By Diameter (Ascending)
Pluto
Mercury
Mars
Venus
Earth
Neptune
Uranus
Saturn
Jupiter
Planets Sorted By Escape Velocity (Ascending)
Pluto
Mercury
Mars
Venus
Earth
Uranus
Neptune
Saturn
Jupiter
1 1 (none) Unicode (UTF-8) Unix (LF) 3,526 / 360 / 157
Screen Shot 2012...2 PM
Screen Shot 2012...4 PM
dolphin-websi...ot.jpg
webasylum
temp


BBEdit File Edit Text View Search Markup Window #! CVS Help Sun 4:43 PM
mp2out.txt
Last Saved: 6/24/12 4:34:18 PM
File Path : ~/Dropbox/IIT/ITM513/mp2/data/mp2out.txt
mp2out.txt
Planets Sorted By Escape Velocity (Ascending)
Pluto
Mercury
Mars
Venus
Earth
Uranus
Neptune
Saturn
Jupiter
Planets Sorted By Revolution Period (Ascending)
Mercury
Venus
Earth
Mars
Jupiter
Saturn
Uranus
Neptune
Pluto
Planets Sorted By Mean Surface Temp (Ascending)
Pluto
Neptune
Uranus
Saturn
Jupiter
Earth
Mars
Mercury
Venus
SolarSystem Dictionary Hash
{'Mercury': <Mercury Planet Instance at 0x108514cf8>, 'Neptune': <Neptune Planet Instance at 0x108514ef0>, 'Pluto': <Pluto Planet Instance at 0x
1 1 (none) Unicode (UTF-8) Unix (LF) 3,526 / 360 / 157
Screen Shot 2012...2 PM
Screen Shot 2012...4 PM
Screen Shot 2012...4 PM
dolphin-websi...ot.jpg
webasylum
temp