Lec07-lists-RyanSponzilli

September 17, 2024

1 ASTR 310 Lecture 07 - lists

1.0.1 1. List mayhem

- Create a list L containing 1000 ones.
- Set every third and every fifth entry to zero.
- Count the number of ones remaining. [You should get 533.]
- Extend L with a second list containing the first 500 multiples of 4.
- Print the sum of the values in the resulting list. [You should get 499533.]

[6 pts total]

You must use lists, not numpy arrays! We will get to numpy arrays before long.

[62]:
```
L = [1]*1000
for i in range(len(L)):
    if i % 3 == 0:
        L[i] = 0
    if i % 5 == 0:
        L[i] = 0
sum(L)
```

[62]:
```
533
```

[63]:
```
L2 = list(range(0, 500*4, 4))
L.extend(L2)
sum(L)
```

[63]:
```
499533
```

1.0.2 2. Lists of magnitudes

- Create three lists called stars, absmags, and distances containing
  - stars: Vega, Deneb, Rigel, Sirius, Arcturus
  - absmags: 0.582, -8.38, -7.84, 1.42, -0.30
  - distances: 7.68, 802, 260, 2.64, 11.26 (these values are in parsecs)
- Now create a new list, appmags, containing apparent magnitudes m calculated using

$$m = M + 5(\log d - 1)$$

where M is the absolute magnitude and d is the distance in parsecs. Note that the logarithm is base-10, so use the log10 function from the math module. *Again, no numpy yet.*
- Iterate over the stars, printing for each star "The apparent magnitude of (star) is (appmag)."

[7 pts total]

[64]:
```
from math import *
```

[65]:
```
stars = ["Vega", "Deneb", "Rigel", "Sirius", "Arcturus"]
absmags  = [0.582, -8.38, -7.83, 1.42, -0.30]
distances = [7.68, 802, 260, 2.64, 11.26]

appmags = [ M + 5 * (log10(d) - 1) for M in absmags for d in distances ]
appmags

for i in range(len(stars)):
    print(f"The apparent magnitude of {stars[i]} is {appmags[i]}")
```

```
The apparent magnitude of Vega is 0.008806100157559782
The apparent magnitude of Deneb is 10.102871841420818
The apparent magnitude of Rigel is 7.656866739854089
The apparent magnitude of Sirius is -2.309980365650845
The apparent magnitude of Arcturus is 0.8396919525766374
```

1.0.3 3. Dictionaries of magnitudes

- Practice with dictionaries by creating a dictionary for one of the stars. The keys should be 'm', 'M', and 'd', and the values should be the apparent magnitude and so on as appropriate.
- Now use the data in the four lists to create a nested dictionary called 'stardict'. Each dictionary entry should have as its key the name of a star, and the value should be itself a dictionary like the one you created above.
- Print stardict['Rigel']['m'].

[7 pts total]

[66]:
```
dict = [{'m': appmags[i], 'M': absmags[i], 'd': distances[i]} for i in␣
 ↪range(len(stars))]
dict
```

[66]:
```
[{'m': 0.008806100157559782, 'M': 0.582, 'd': 7.68},
 {'m': 10.102871841420818, 'M': -8.38, 'd': 802},
 {'m': 7.656866739854089, 'M': -7.83, 'd': 260},
 {'m': -2.309980365650845, 'M': 1.42, 'd': 2.64},
 {'m': 0.8396919525766374, 'M': -0.3, 'd': 11.26}]
```

[67]:
```
stardict = {s : d for s, d in zip(stars, dict)}
```

[68]:
```
print(stardict['Rigel']['m'])
```

```
7.656866739854089
```