The importance of keeping logs & notes

- Keeping track of your work
- Increases productivity

logs

- Very helpful when someone else needs to work on the same project
- Extremely helpful when you need to reuse your own work years later

The importance of keeping logs & notes

In [ ]:

```
tt = dats[:,0]
yy = dats[:,1]


tq = np.arange(140)/10.

params, params_covariance = optimize.curve_fit(test_func, tt, yy,
                                               p0=[60,4,.4])


plt.figure(figsize=(18,10))
plt.plot(tt, yy, "ok", label="data")
#plt.plot(tt, test_func(tt, params[0], params[1], params[2]),
#         label='Fitted function')
plt.plot(tq, test_func(tq, params[0], params[1], params[2]),
         label='Fitted function')
#plt.plot(tq,60*np.sin(2*np.pi*tq/4+.4))

plt.xlabel('Time [hrs]',fontsize=24)
plt.ylabel('Amplitude [m/s]',fontsize=24)
plt.xticks(fontsize=24)
plt.yticks(fontsize=24)
plt.legend(loc="best",fontsize=24)
plt.show()


a = (params[1]**2/365**2)**(1/3.)

print("Approximate a ",a, "AU")


mstar = 1.1
K = abs(params[0])
m = K *np.sqrt(mstar) *np.sqrt(a)/28.43

print("approximate m" , m, "M$_{Jup}$")
```

The importance of keeping logs & notes

In [ ]:
```
tt = dats[:,0]
yy = dats[:,1]


tq = np.arange(140)/10.

params, params_covariance = optimize.curve_fit(test_func, tt, yy
                                               p0=[60,4,.4])



plt.figure(figsize=(18,10))
plt.plot(tt, yy, "ok", label="data")
#plt.plot(tt, test_func(tt, params[0], params[1], params[2]),
#          label='Fitted function')
plt.plot(tq, test_func(tq, params[0], params[1], params[2]),
         label='Fitted function')
#plt.plot(tq,60*np.sin(2*np.pi*tq/4+.4))

plt.xlabel('Time [hrs]',fontsize=24)
plt.ylabel('Amplitude [m/s]',fontsize=24)
plt.xticks(fontsize=24)
plt.yticks(fontsize=24)
plt.legend(loc="best",fontsize=24)
plt.show()


a = (params[1]**2/365**2)**(1/3.)

print("Approximate a ",a, "AU")


mstar = 1.1
K = abs(params[0])
m = K *np.sqrt(mstar) *np.sqrt(a)/28.43

print("approximate m" , m, "M$_{Jup}$")
```

In [ ]:
```
#assign data to variables
tt = dats[:,0]
yy = dats[:,1]

#perform a curve fit to data to find best fit period and amplitude
tq = np.arange(140)/10.

params, params_covariance = optimize.curve_fit(test_func, tt, yy,
                                               p0=[60,4,.4])


plt.figure(figsize=(18,10))
plt.plot(tt, yy, "ok", label="data")
#plt.plot(tt, test_func(tt, params[0], params[1], params[2]),
#          label='Fitted function')
plt.plot(tq, test_func(tq, params[0], params[1], params[2]),
         label='Fitted function')
#plt.plot(tq,60*np.sin(2*np.pi*tq/4+.4))

plt.xlabel('Time [hrs]',fontsize=24)
plt.ylabel('Amplitude [m/s]',fontsize=24)
plt.xticks(fontsize=24)
plt.yticks(fontsize=24)
plt.legend(loc="best",fontsize=24)
plt.show()

#use period to retrieve a:
#from Kepler: 4np.pi^2/p^2 = G (m1+m2)/a^3   --> assume 1.1Mo comprable to Sun -->
#p2^~a^3
a = (params[1]**2/365**2)**(1/3.)

print("Approximate a ",a, "AU")

#use amplitude to retrieve mass:
#: K=\frac{28.43\left[m/s\right]}{\sqrt{(1)-e^2)}\frac{m_2sin{i}}{M_{Jup}}m1+m2[Mo]-1/2α1 [AU]-1/2
#assume you see the system edge-on and it has a circularized orbit:
#K = 28.43 * m [Mjup] /sqrt( [mstar+m] [Msun] /sqrt[a [AU]] )
#simplify mstar+m ~ mstar
mstar = 1.1
K = abs(params[0])
m = K *np.sqrt(mstar) *np.sqrt(a)/28.43

print("approximate m" , m, "M$_{Jup}$")
```

Organizing your work

- Organization, naming and logging are important

➢ Find things later

➢ See what you have on a topic

➢ Possibly you will get fewer questions when someone needs to take over/needs to use your project

- Organize projects and their files so that...

➢ Files you get later will fit naturally into that structure

➢ Things list in the order you want them to

➢ One look reveals their purpose

➢ You'll scarcely want to move them

➢ Example: Directory (folder) for this course

Organizing your work

We will keep class logs, just like a research log

➢ Dated entries for all classroom and work sessions
➢ Notes and summaries of what you do
➢ Links to online resources
➢ Snippets of code, screen shots and plots

All these enable you to:

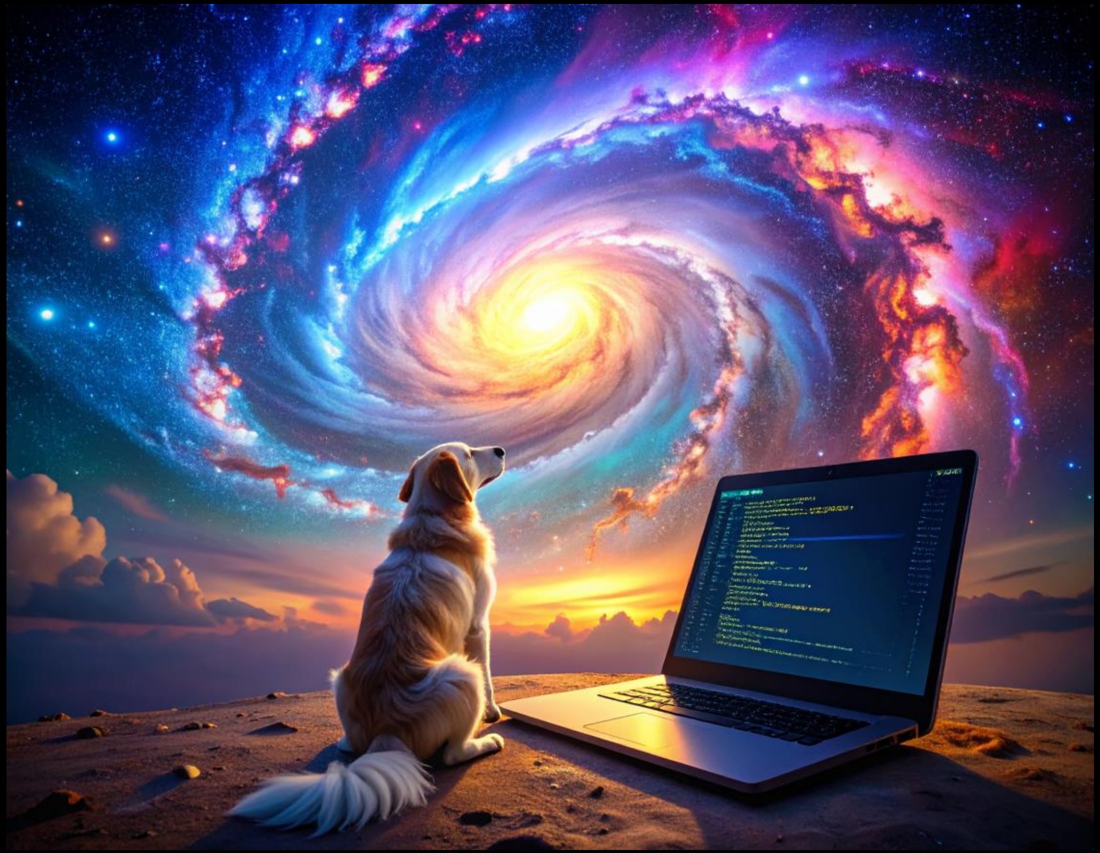
1. Go back later and restart work
2. Figure out what went wrong and fix it
3. Manage a large number of trials
4. Prove/report your activities to others
5. Track time

Logs & notes

The log

- ➢ Use a text editor (not a word processor), see HW0
- ➢ Start new entry with “*****IN: ” and output of ’date’
- ➢ Leave blank line above and below
- ➢ Never (EVER) change above current date line
- ➢ If tracking time, end with “*****OUT: ” and output of ’date’

HW0



- Start HW0 please!
- Due at 3:40pm

Peer evaluation

- 10% of your grade!

- Do them well
  - ‘nice job!’
  - ‘well done!’
  - ‘all parts there and working’

Peer evaluation

- 10% of your grade!

- Do them well
  - ‘nice job!’
  - ‘well done!’
  - ‘all parts there and working’

Peer evaluation

- 10% of your grade!

- Do them well
  - ‘nice job!’
  - ‘well done!’
  - ‘all parts there and working’

  - ‘nice job! I liked how well written and clear your log is and how well documented the code is….’

Peer evaluation

- not grading (we will do the grading)

- Why?
  - Going through the rubric, solutions, and another's code in detail leads to a much deeper understanding and a longer retention of the material
  - Seeing how others code something might help you learn how to do something better/faster (or what to avoid)
  - We aim to discuss multiple ways it could have been coded and the benefits and disadvantages of each in class

Let’s see an example

```
--------saved as 0-ast4762-tkar.log.dat ---------

Theodora Karalidi
Log for AST 4762, Fall 2024

*****IN: Wed Aul 22 12:47:07 EST 2024
Opened TextEdit, started this log, saved as:
/home/jdoe/Desktop/0-phz3150-janedoe.log

     Name: Theodora Karalidi
     Username: tkar

     What astronomy classes have you taken? All of them!

     Are you generally comfortable with calculus and physics?
Not much.

     Have you ever programmed a computer? Yes

     In what language(s)? FORTRAN, Python, IDL, Origin, Pascal,
bit of Perl

…..

TO DO: add last parts of HW and submit it

*****OUT: Wed Aul 22 12:54:07 EST 2024

Right clicked on file, went to compress and zipped file.
Submitted to Webcourses
```

- Let’s check an example out