Lecture 11 - File I/O & Debugging

* prayer / spiritual thought

* announcements

Joke:

["hip", "hip"]

(hip hip array!)

* Unit of the Day:

1 atm = 760 mm Hg ↪ aka Torr

↑ mercury.

(weight of Hg in a column)

I. Debugging

* Four stages of programming

iterative

- Design - what should the code do?
- Coding - writing code
- Testing - check if it works as intended, i.e. find errors
- Debugging - fix errors.

* Types of bugs

- Syntax errors - "spelling" / formatting
  - easiest, interpreter finds
- execution errors - something bad happens
  - e.g. divide by 0, bad variable assignment
  - medium → easy to locate, tricky to fix.

- Logical errors – code runs, but gives wrong answer
  - computer is stupid.
  - You told it something different than you thought you did.
  - Hardest to find & fix.

* How find / fix bugs?

- the scientific method!
  (1) Hypothesis: what do you think is wrong?
  (2) Experiment: How can you isolate & test this hypothesis?
  (3) Analyze: was that the problem? Yes (wahoo!), No (:, repeat)

- other tips
  - Test early & test often. Don't write whole code & then test.
  - Go to a point where something works. Then add lines of code
  - Test modules (i.e. pieces) of code.
  - Read the interpreter's output. what is it saying? what line?
  - Test your assumptions. check your answers.
  - Use spyder's debugging tools
  - Be careful & detail oriented. If you haven't tested it, assume it's wrong.

- write out what you expect the code to do on a piece of paper. (*) Exam problem! Good
- Programming is really an exercise in debugging

* Debugging is a valuable skill you will use the rest of your life. It is maybe the most important lecture in this class.

Activity

- Debug some code together.
- Practice problem debugging some code.

II. File I/O

- np.savetxt, np.loadtxt
- open, close, write, readline

Activity

- Practice some basic file I/O