COMPUTER SCIENCE

LESSON 28+29, TUESDAY JAN 9TH + THURSDAY JAN 11TH

WWW.CITA.UTORONTO.CA/~WOODFORD

WOODFORD@CITA.UTORONTO.CA

CANADIAN COMPUTING COMPETITION

- Need to register soon!! We register as a school since it's combined with a math challenge.
- Feb 14th is when the actual challenge is
- Individual competition: no teams

- http://www.cemc.uwaterloo.ca/contests/computing.html

THIS WEEK IN CS AND STEM

- The rise of cryptocurrency and e-Residency – Estonia going digital
  - https://futurism.com/estonia-revealed-three-uses-proposed-national-cryptocurrency/
- What DIDN'T happen??
  - https://futurism.com/images/this-week-in-science-dec-16-22-2017/
  - ✓Using STEM cells to cure paralysis?
  - ✓Finding the earliest evidence for life on Earth?
  - ✓UFO investigations?
  - ✓The first CRISPR clinical trials?
  - ✓A baby born from a 24 year old frozen embryo???
  - Science calm down.

INPUT AND OUTPUT FILES: SUMMARY

- A lot of the data we've been using thus far has been input from the user on the command line or hard-coded into the file. This isn't reasonable for large data sets or complicated entry methods.
- It's typically easier for users to create an input file that contains all the data needed by a function.
- f. = open.("filename.txt","r"), f.read(), for line in f: line.split()
- f = open("filename.txt","w"), f.write()
- f.close()

ARGPARSE

- A bit more formalism for this one. Take a look at the reference material for this on the python docs page for full details.

```
Import argparse

parser = argparse.ArgumentParser(description = “Get an input file name”)

parser.add_argument(“—i_filename”, required = True)

args.parser.parse_args()
```

- Lets take a look in Lecture24_25_argparse.py

REFRESHER

- Open Lecture28_29.py, there are 2 tasks:
  1. First modify to open input and output files assuming you know the names
  2. Then alter to use argparse so the filenames are general

PLOTTING: THE BASICS

- So far we've been making all kinds of data and output and working on the first 4 steps of our coding practise
- Let's dive a little into aesthetics. Usually the best way to represent data is visually, ie. by plotting. What are some kinds of plotting that you're familiar with?

MATPLOTLIB

- This is the most popular and intuitive plotting package for python. To use, “import matplotlib.pyplot as plt” is the typically import line

- There’s nearly everything you’ll ever need in this package, let’s get started with Lecture28_29_plotting.ipynb

EXTENDING TO SCRIPTS

- Now that we've had some practise, let's see what we can do with plotting in Lecture28_29.py
  - Try outputting using plt.show() first. You shouldn't have a plot for every timestep (that's a lot!), but every few time steps. How many do you think?
  - Now try saving the plots. You'll need to think of a way to save the plots separately as opposed to overwriting them.

"ASSIGNMENT"

- Depending on the exam grades…
  - If you've done well and you're happy with your exam grade, then you don't have any assignment work this week.
  - If you're not pleased with your mark and wish to improve it, submit corrections to all but the last question for part marks. It will either be 1/3 or ½ depending on the class average.
- Due Jan 18th, next Thursday, at the beginning of class.

REFERENCES

- https://docs.python.org/3/library/filesys.html
- https://docs.python.org/3/library/functions.html#open
- https://docs.python.org/3/library/argparse.html
- https://docs.python.org/3/tutorial/inputoutput.html
- https://matplotlib.org/