

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

- Tesla makes good on their deals – solar panel roofing tiles available for installation
 - <https://futurism.com/teslas-solar-roofing-tiles-begun-production-buffalo-factory/>
- University of Cambridge produces a BPV that's 5x more efficient than current bio-solar cells
 - <https://futurism.com/genetically-modified-algae-key-tomorrows-bio-solar-cells/>
 - Not suited for large scale productions, but ideal for small scale use (ie. homes, car-charging, etc)

ASSIGNMENT 10

- Upgrade Lecture28_29.py to include all the i/o we discussed (feel free to use Lecture28_29_updated.py).
- Modify so that a plot of the projectile is saved in the same directory every 5 timesteps. Your plots should include some sort of identification numbers.
- Submit the script, and a sample input file. NOT THE PLOTS.
- Due Sunday Jan 20, 2018 by 11:59pm via email to woodford@cita.utoronto.ca

MODULARIZATION

- We've had some practise with modularization and packages. Now we'll start building larger codes that combine these techniques.
- Essentially, we can group similar functions together in files, and then call on specific functions by loading those files as packages into a current script.

STARTING SMALL

- One of the more modularized codes we've used so far are those for finding the tan of a number (ie. the scripts for assignment 4).
- If you don't have the files, or if you've lost them, you can find the solution on the website!
- Break the code up into different files, grouping like functions together. Then use the main file to import the packages and call the functions.

LOOKING AT LARGER CODES

- Now that we know HOW to modularize and call functions from other files, let's take a look at how larger packages do this.
- Let's look at the Pyglet and Scrapy packages:
<https://bitbucket.org/pyglet/pyglet/wiki/Home>
<https://github.com/scrapy/scrapy/blob/1.5/docs/index.rst>
 - What do you notice? How are functions called, how can you tell, and where are they?
 - One is hosted on btibucket, the other on github; what are these? Did you find one easier to navigate than the other?

REFERENCES

- https://www.python-course.eu/modules_and_modular_programming.php
- <https://docs.python.org/3/tutorial/modules.html>
- <https://pythontips.com/2013/07/30/20-python-libraries-you-cant-live-without/>