Computer Science — Python — HW #16

Assigned on Day, YYYY-MM-DD. Complete by Day, YYYY-MM-DD.

1. Read Appendices A & B of ThinkPython, 2nd edition. Celebrate having completed the entire book!
2. **[Optional]** Watch a video of Grace Hopper (the first compiler writer) being awesome on the David Letterman show.

<https://www.youtube.com/watch?v=1-vcErOPofQ>

In this video, Hopper mentions The Mark I computer. It was built at Harvard by Harvard and IBM from 1939 to 1944. It was 8 feet tall and 51ft long! Transistors, which led to smaller computers, hadn't been invented yet.

1. **Call by reference vs. call by value (append7 revisited).** Recall that HW #3 defined a function called **append7**.

def append7(items):

items.append(7)

We saw that when the main part of the program called this function several times, it had a cumulative effect.

my\_list = [ ]

append7(my\_list) # Now my\_list = [7]

append7(my\_list) # Now my\_list = [7, 7]

So, when **append7** is called, it doesn't just pass append7 a copy of **my\_list**. In fact, it passes something that *refers* to the main copy of **my\_list** (which is defined outside of the function). This allows **append7** to change the value of **my\_list**. This something is called a "*reference*". In Python, when any objects (including any list) is passed as an argument to a function, it is passed as a *reference*. This behavior can result in subtle bugs if not understood properly. For a helpful warning, read about "Mutable Default Arguments" at <http://docs.python-guide.org/en/latest/writing/gotchas/>

1. **[Turn in]**  The Smithsonian is more than dusty paintings and fossilized dinosaur bones.

Provide brief responses to the following questions, in complete sentences, and in your own words.

* 1. Given that four grand challenges in the Smithsonian Institution's (SI's) 2017 Strategic Plan (<https://www.si.edu/Content/Pdf/About/SI_Strategic_Plan_2017.pdf>) don't mention software or digitization, why does the document go ahead to put forward a "Digital Action Agenda"?
  2. The Smithsonian has well over 100 million items in its collection. What are the objectives for its digitization program? (See <https://www.si.edu/content/pdf/about/2010_si_digitization_plan.pdf>)
  3. Give an example of a case where Python has been used in the Smithsonian's digitization work.

**Note:** You can restrict a Google search to a the domain **si.edu** by adding **site:si.edu** to your search.

* 1. Pick an internship available at the Smithsonian that sounds interesting, provide a link to a web page about it, and describe the internship.
  2. The Smithsonian is an educational institution, so what's the difference between the Smithsonian Center for Learning and Digital Access (CSLDA) and the Office of the Chief Information Officer (OCIO)
  3. If you have a smartphone, install the Smithsonian Mobile app, and start exploring!

1. **[Optional]** **Northwest Yeshiva Hidden Strings (NYHS).** Remember the script **adventure.py** from HW #12? It established links between different rooms in a floor plan. This is an example of a data structure called a (network) **graph**, where "nodes" are linked together by "edges".

|  |  |
| --- | --- |
| Take a moment to look for words in the grid shown to the right, where letters linked to each other in the grid by an edge are allowed to follow each other in the word. For example, the word "horse" can be spelled using the shaded disks, which are all linked. You should be able to find several words just by looking. To find them all, you could use Python!  The script **grid\_words.py**, at the course website, can find all of them — at least all that are in the file wordlist.txt, which comes with the script in **grid\_words.py.zip** (together with wordlist\_short.txt, for testing).  This is just one example where the notion of a (network) graph is useful. There are many, many more. |  |