Richard Hayes Crowley

06/28/2021

CSC\_242\_Lab\_06\_QA

**(1)** Is a multi - dimensional list created here? Explain your answer.

|  |
| --- |
| **string = "this is a phrase"**  **result1 = [line.strip().split() for line in string.split('\n')]**  **string = "this is another phrase"**  **result2 = [line.strip().split() for line in string.split('\n')]**  **result = [result1,result2]**  **print (result)** |

**Yes! This is actually a 3D list, constructed using list comprehension. The line.split() operation produces an array of words for each line created by the string.split() method, which would create a new item for each new line (in this instance, there is only one line).**

**The end result of result1 looks like a 2D list:** [['this', 'is', 'a', 'phrase']]

**Adding this to the end result of result, which performs the exact same operations, is another 2D list** [['this', 'is', 'another', 'phrase']]

**The final result of putting both of these 2D lists together in a new list is a 3D list, with dimensions 1x2x4 (this can be visualized as row, column, depth):**

[[['this', 'is', 'a', 'phrase']], [['this', 'is', 'another', 'phrase']]]

**(2)** Would it make sense to perform a matrix transposition of either of your multi - dimensional grids that comprise your program? Explain your answer.

**It would make sense perhaps if we wanted to shuffle the items around the library, however, to keep categories aligned with their quantities, both matrices would need to be transposed in order to preserve 1:1 correlation between cells in the category matrix and the quantity matrix.   
  
However, in my program I’ve combined these two matrices into one 3D matrix with a [categoryNumber, quantity] list in each cell. So we could transpose only this matrix if we wanted to shuffle the locations of items in the library (ignoring the floor column and room row, of course.)**

**(3)** Does Python’s**pickle** library provide for multi - language support, as does JSON ( JavaScript Object Notation ) ? Is Python **pickle** suitable for human reading devices ?

**No and no. Pickle streams Python objects into a binary format (bytes) and reads binary pickle objects back into Python objects. It’s not suitable for use outside of the Python ecosystem and is not very secure, since arbitrary code can be executed during unpickling.**

**Pickling seems to have its place however for Python developers who want to e.g., read/write entire Class objects to a database. The JSON module only enables developers to access a subset of Python types (primitives, dicts, Lists, etc.), however most interactions with a server or database would likely involve these more universal data types anyhow, and as JSON is largely available to most systems (e.g. Java, JavaScript, Swift, Obj-C etc. all have JSON support), JSON is probably the better tool for reading and writing data to/from a database.**

**(4)** Explain how atwo - dimensional array list can be used to model the available seats in a movie cinema complex.

**A movie theater could be visualized as a rectangular grid, wherein each row in an array of rows has an array of available and non-available seats in it.**

**A 2D list, or an array of arrays one level deep, could represent available and non-available seats as e.g., “X” and “O”, representing closed and open.  
  
It would look like the following:**

availableSeats = [["x", "x", "o"], ["o", "o", "x"], ["x", "x", "x"]]

**(5)** After the **genres** dictionary was created, explain how you would supplement it with additional genre categories.

**I could prompt the user to add more genre categories to the dictionary, and based on the length of items in the dictionary, add the next n + 1 key integer to the dictionary (along with the genre value). We’d then need to create a new Library instantiation with the new genres, since all of my setup logic is within the initializer. However, one could move those initializing functions (e.g., build library) into their own methods, and then call those methods again to rebuild the library whenever the genres get updated**