Discrete Structures!

CMPSC 102

Data Containers

ALLEGHENY COLLEGE

Key Questions and Learning Objectives

- How do I use the mathematical concepts of ordered pairs, n-tuples, lists and dictionaries to implement functions with a clearly specified behaviors?

- To remember and understand some discrete mathematics and Python programming concepts, enabling the investigation of practical applications

Combining Dictionaries and Lists - create a list of data

```
# define Alice's list
detailsAlice=["555-8181", "Alice@...", "Paris"]
print(f" email: {detailsAlice[1]}")

# define Mike's list
detailsMike=["555-1234", "michael@...", "Meadville"]
print(f" email: {detailsMike[1]}")

# create dictionary
contacts = {}
#add details as key, value assignment
contacts["Alice"] = detailsAlice
contacts["Mike"] = detailsMike

for i in contacts: # extract details
                print(f"{i} -> {contacts[i]}")
```

More with Dictionaries and Lists - Part 1

```
multsOfTwo = []
for i in range(10):
        multsOfTwo.append(i**2)
print(f"multsOfTwo : {multsOfTwo}")
# multsOfTwo : [0, 1, 4, 9, 16, 25, 36, 49, 64, 81]

multsOfTwo = []
multsOfThree = []
multsOfFour = []
for i in range(10):
        multsOfTwo.append(i**2)
        multsOfThree.append(i**3)
        multsOfFour.append(i**4)

print(f"multsOfTwo : {multsOfTwo}") # : [0,1,4,...,81]
print(f"multsOfThree : {multsOfThree}") # : [0,1,8,...,729]
print(f"multsOfFour : {multsOfFour}") # : [0,1,16,...,6561]
```

More with Dictionaries and Lists - Part 2

- Add all lists to a dictionary

```
# assign dictionary
multiples = {}
multiples["twos"] = multsOfTwo
multiples["three"] = multsOfThree
multiples["four"] = multsOfFour

for i in multiples:
        print(f" multiples of {i} -> {multiples[i]}")
```

Data in the Form of Tuples

- Comma separate value (CSV) are frequently used in business and science!
- How can we input this file of n-tuples into a Python program?
- How do we parse each line based on a delimiter?
- How can the program handle multiple-word content with commas?

CSV Data - Files in Directories Can Store n-Tuples

- Suppose you had some data in a CSV format?
- How to do something with the data?!

- CSV data: sandbox/contacts.csv

```
tylernelson@gmail.com,Careers adviser
gregory02@medina-mayer.com,"Accountant, management"
jonesmiguel@hotmail.com,Health and safety inspector
rsanchez@yahoo.com,"Surveyor, planning and development"
hillfrank@ward-wood.com,"Scientist, physiological"
aaronhunter@gmail.com,"Surveyor, insurance"
kylebarnes@hotmail.com,Records manager
joe70@yahoo.com,Network engineer
torresjames@white.info,Electrical engineer
shawkins@watson.com,Science writer
```

Functions that Manage Tuples - File: csvreader.py

```
from os.path import exists
from logging import exception

def openCSVFile(fname_str: str) -> str:
        """loads a file, returns csv string"""
        # print("openCSVFile()")
        if not exists(fname_str): # no file found?
                print(f"\t [-] No file by that name: {fname_str}")
                exit() # end program if no file has been found.
        try:
                data_str = open(fname_str, "r").read()
        except exception:
                print("\t [-] Using correct filename?")
                return None
        # commas in this loaded file?
        if len(data_str) > 0 and "," in data_str:
                return data_str
        return None
```

Functions that Manage Tuples - iterateData

```
def iterateData(in_str: str) -> dict:
        """Function to output the data in tidy lines. Place data into dictionary."""
        contact_dict = {}
        for line in in_str.splitlines():
                # print(line)
                # get the name, located before first comma
                name_str = line[: line.find(",")]
                service_str = line[line.find(",") + 1 :].replace("'", "")
                contact_dict[name_str] = service_str
        return contact_dict
```

Functions that Manage Tuples - main()

```
def main() -> None:
        """driver function"""
        prompt_str = "\t Enter the CSV filename : "
        myFile_str = input(prompt_str)
        # print(f"\t [+] You entered file : {myFile_str}")
        myCSV_str = openCSVFile(myFile_str)
        # print(f"Main() {myCSV_str}")
        # print out in tidy lines
        myContact_dict = iterateData(myCSV_str)
        # print(f"Dictionary of names: {myContact_dict}")
        for i in myContact_dict:
                print(f"\t [+] {i} : {myContact_dict[i]}")
```