On your way in...(on the side table)

Pick-Up:
1. HW 02, graded
2. POGIL Activity 22 – List Comprehensions
CS THESIS PROPOSALS THIS FRIDAY
2:35P IN TCL 123 (WEGE)

• Will Burford
• Tongyu Zhou*
• Adly Templeton
• Josh Kang

Come see what CS seniors are up to!
If you’re interested in majoring in Computer Science, you’ll need 20 colloquium to graduate, and any you sign-in for now will count!
Welcome to CS 134!
Introduction to Computer Science
Iris Howley
-Files & List Comprehensions-
Continuing from Wednesday

- How to take any string and insert a comma after the 5th element?

```python
>>> str = "hello there!"
>>> newstr = str[:5] + ',' + str[5:]
  >>> str[:5]
    'hello'
  >>> str[5:]
    ' there!'
>>> newstr
'hello, there!'
```

- How to take any tuple and insert a 2020 after the 5th element?

```python
>>> tup = (1,2,3,4,5,6,7,8,9)
>>> newtup = tup[:5] + (2020,) + tup[5:]
  >>> tup[:5]
    (1, 2, 3, 4, 5)
  >>> tup[5:]
    (6, 7, 8, 9)
>>> newtup
(1, 2, 3, 4, 5, 2020, 6, 7, 8, 9)
```

Strings & Tuples are immutable, so we must make a new one to modify them!
TODAY’S LESSON
Files

(A persistent store between python sessions)
Reading Files

Opens the file
Filename as a string
Opens filename calls it fin

• `with open('prideprejudi.txt') as fin:
• `for line in fin:
  • `pass
  • `# file is implicitly closed

For each line in our file, fin
Does nothing, why’s it here? What should be here?

Once we leave the “with” indentation, the file is closed!

FILES MUST BE OPENED, READ, AND THEN CLOSED
Writing Files

Opens the file

\[
\text{with open(‘newFile.txt’, ‘w’) as fout:}
\]

\[
\text{fout.write(“Hello!!”)}
\]

\[
\text{for item in mylist:}
\]

\[
\text{fout.write(item)}
\]

Once we leave the “with” indentation, the file is closed!

\[
\text{# file is implicitly closed}
\]

If unable to use the ‘with’ keyword, can also use \text{fout.close()} to explicitly close file

FILES MUST BE OPENED, WRITTEN, AND THEN CLOSED

Specifies mode. \text{w} means?

What if we had ‘r’ here?

Opens filename calls it \text{fout}

Filename as a string

Opens the file

\text{with open(‘newFile.txt’, ‘w’) as fout:}

\text{fout.write(“Hello!!”)}

\text{for item in mylist:}

\text{fout.write(item)}

Once we leave the “with” indentation, the file is closed!

\text{# file is implicitly closed}

If unable to use the ‘with’ keyword, can also use \text{fout.close()} to explicitly close file

What if we had ‘r’ here?

Opens filename calls it \text{fout}

Filename as a string

Opens the file

\text{with open(‘newFile.txt’, ‘w’) as fout:}

\text{fout.write(“Hello!!”)}

\text{for item in mylist:}

\text{fout.write(item)}

Once we leave the “with” indentation, the file is closed!

\text{# file is implicitly closed}

If unable to use the ‘with’ keyword, can also use \text{fout.close()} to explicitly close file
TODAY’S LESSON
Reading CSV Files

(A common format for text data files)
Reading CSV Files

• What is a CSV file?
  • Comma Separated Values

It’s a raw text format way to save data that is typically organized into rows & columns
Reading CSV Files

Adds the csv module to our list of possible tools to use

```python
import csv

def printCSVfile(filename):
    with open(filename, 'r') as f:
        csvf = csv.reader(f)
        for row in csvf:
            print(row, end="\n\n")
```

Filename as a string

Opens filename calls it f

Create a CSV reader object to parse the commas in f for us

Print the row and place some extra newlines at the end for readability

For each ‘row’ (ended with a newline) in our CSV file reader...

Once we leave the “with” indentation, the file is closed!
Reading CSV Files

What is this?

c Department', '2001, B.A., Skidmore College', '2003, M.A., Boston College', '2014, Ph.D., Boston University']

['Yu,Li', 'Chair and Associate Professor of Chinese', 'Asian Studies Department', '1994, B.A., East China Normal University', '1997, M.A., Ohio State University', '2003, Ph.D., Ohio State University']


['Zimmerberg,Betty', 'Howard B. Schow '50 and Nan W. Schow Professor of Neuroscience', 'Psychology Department', '1971, B.A., Harvard University', '', '1976, Ph.D., City University of New York']

['Zimmerman,David J.', 'Professor of Economics and Orrin Sage Professor of Political Economy', 'Economics Department', '1985, B.A., University of Toronto', '1987, M.A., Princeton University', '1992, Ph.D., Princeton University']


['van de Stadt, Juana M.', 'Professor of Russian', 'German and Russian Department', '1988, B.A., Amherst College', '1994, M.A., University of Wisconsin, Madison', '2000, Ph.D., University of Wisconsin, Madison']
Reading CSV Files

• `python3 readFile.py`

```python
import csv

def printCSVfile(filename):
    with open(filename, 'r') as f:
        csvf = csv.reader(f)
        for row in csvf:
            # row[0] --> name
            # row[1] --> title
            # row[3] --> undergrad degree
            print(row[0], row[1], row[3], sep="\t", end="\n\n")
```

Wood, James B.  Charles R. Keller Professor of History  1968, B.A., Eckerd College

Wootters, William K.  Barclay Jermain Professor of Natural Philosophy  1973, B.S., Stanford University

Yamamoto, Kasumi  Professor of Japanese  1980, B.A., Aoyama Gakuin University

Yesnowitz, Joshua C.  Visiting Assistant Professor of Political Science  2001, B.A., Simon’s Rock College

Yu, Li  Chair and Associate Professor of Chinese  1994, B.A., East China Normal University

Zaki, Safa R.  Professor of Psychology  1989, B.A., American University

Zimmerberg, Betty  Howard B. Schow ’50 and Nan W. Schow Professor of Neuroscience  1971, B.A., Harvard University

Zimmerman, David J.  Professor of Economics and Orrin Sage Professor of Political Economy;  1985, B.A., University of Toronto

van Ginhoven, Kristen M.  Visiting Lecturer in Theatre  2003, B.Ed., Queen’s University

van de Stadt, Juana M.  Professor of Russian  1988, B.A., Amherst College
Splitting Strings on Commas

• >>> str = "hello, world! i'm a programmer!"
• >>> str.split("",")
• ['hello’, ’ world! i'm a programmer!’]

.split('item') will create a list of strings, split into elements based on item

>>> name = 'van de Stadt,Juana M.'
>>> name.split(',',')
['van de Stadt', 'Juana M.]
>>> name.split(' ')
['van', 'de', 'Stadt,Juana', 'M.']
TODAY’S LESSON
List Comprehensions

(ways to iterate over items more efficiently)
POGIL Activity 22- List Comprehensions

• A briefer way to iterate over sequences

• Look at Python Activity 22, Question 1-6

• Find a partner and talk through the questions together
POGIL – Activity 22: Questions 1 & 2

These two code samples do the same thing, how?

a. What part of code initializes the list `gbp`?

b. What part of the code iterates through each element of the `monies` list?

c. What part of the code converts the values of `monies` from USD to GBP?
List Comprehensions

• `words = ['aye', 'matey', 'yohoho']`
  
  It’s going to be a list

• `pirate = [wd+'arrr' for wd in words]`
  
  Variable to store results in  What item to store in the variable

  A for...loop that refers to each element in words (list) as wd

• `print(pirate)`
  
  • ['ayearrr', 'mateyarrr', 'yohohoarrr']
POGIL – Activity 22: Question 3

0 # Assume each element of the list words is a line from
0 # /usr/share/dict/words (the unix dictionary)
1 longer = [ wd for wd in words if len(wd) > 5 ]

a. What differs in this list comprehension that we did not have in the previous USD/GBP example?

```python
0 monies = [1.22, 5.50, 3]
1 gbp = [usd*0.77 for usd in monies]
```
POGIL – Activity 22: Question 3

0 # Assume each element of the list words is a line from
0 # /usr/share/dict/words (the unix dictionary)
1 longer = [ wd for wd in words if len(wd) > 5 ]

b. What does the variable \texttt{wd} represent in this code?

c. What does the code \texttt{if len(wd) > 5} do?

d. Why is this line of code enclosed in square brackets?

e. When this code completes execution, describe what is stored in the \texttt{longer} variable:

f. Write code to create a list that contains only words that begin with the letter ‘w’. Use a list comprehension:
POGIL – Activity 22: Question 4

```python
0 testStr = "Hello 12345 World"
1 newList = []
2 for x in testStr:
3    if x.isdigit():
4        newList.append(x)
```

a. What does the code on line 3 do?

b. What will `newList` contain when this code completes execution?

```python
["1", "2", "3", "4", "5"]
```

c. Construct a list comprehension that accomplishes the same tasks as this example code:
POGIL – Activity 22: Question 5

```python
>>> def hasSub(word, substring):
...     return substring in word

>>> names = ['pixel', 'tally', 'wally', 'linus', 'annie']

>>> similar = [dog for dog in names if hasSub(dog, 'lly')]

>>> similar
['tally', 'wally']
```

a. If we call `hasSub(dog, 'lly')`, what does the function return?

b. What might `substring in word` do?

c. Construct a list comprehension that accomplishes the same tasks as this example code, but without the function `hasSub(...)`: 
POGIL – Activity 22: Question 6

```
combined = [ x+y for x in wds for y in wds if x+y in words ]
```

a. Rewrite the above list comprehension as a multi-line statement:

b. What does this list comprehension do?
YOU SHOULD COMPLETE THE REST OF ALL POGILS OUTSIDE OF CLASS.

BEST DONE WITH A PARTNER OR STUDY GROUP.

CHECK YOUR ANSWERS ON A COMPUTER!
List Comprehensions

\[ w_4 = \{\text{word for word in words if } \text{len(word)} == 4\} \]

• Is the same as:

\[ w_4 = [\ ] \]

• for word in words:
  ▪ if len(word) == 4:
    □ \( w_4 \cdot \text{append}(\text{word}) \)
List Comprehensions

- Can use nested for loops, too

- Find compound words made up of 2 four-letter words

- \[ \text{ww8} = [w1+w2 \text{ for } w1 \text{ in } w4 \text{ for } w2 \text{ in } w4 \text{ if } w1+w2 \text{ in words}] \]

5000 X 5000 combinations of words
See if it’s in a list of several 100,000
...this will take awhile...
TODAY’S LESSON

More Lists

(a few more handy things we can do with lists)
Appending/Extending

• \( l = [5, 16, 18] \)
• \( m = ['pixel', 'pup'] \)
• \( l.append(m) \)

What happened here?

• \( l = [5, 16, 18] \)
• \( m = ['pixel', 'pup'] \)
• \( l.extend(m) \)

What happened here?
Other List Functions

- \( x = ['new']*3 + ['old']*8 \)
- \( x \)
  - ['new', 'new', 'new', 'old', 'old', 'old', 'old', 'old', 'old', 'old', 'old', 'old']

**What happened here?**

- \( x\.count('new') \)
  - 3
- \( x\.index('old') \)
  - 3
QUESTIONS?
Leftover Slides
List Comprehensions

```
words = [line.strip() for line in open('/usr/share/dict/words')]
```

• Is the same as:

```
words = []
for line in open('/usr/share/dict/words'):
    words.append(line.strip())
```
POGIL Activity 15- Writing to Files

• More data structures for sequences of objects

• Look at Python Activity 15, Question 1
• Find a partner and talk through the questions together
def studentInfo():
    lastName = input(“Last name: “)
    firstName = input(“First name: “)
    studentID = input(“Enter ID: “)

    inFile = open(“studentInfo.txt”, “a”)
    inFile.write(“Name: “ + firstName + “ “ + lastName)
    inFile.write(“\nStudentID: “ + studentID)
    inFile.write(“\n”)
    inFile.close()
    print(“Done! Data is saved in file: studentInfo.txt”)

a. What output appears on the screen? __________________________________________

b. What does the program do? __________________________________________________

__________________________________________

c. Locate the file studentInfo.txt on your computer. The file is stored in the same folder as
the program. What is stored in the file? ____________________________________________

d. Change the argument ‘a’ to ‘w’ in the call to the open function. What is the purpose of
the “w” in the following line of code?

    inFile = open(“studentInfo.txt”, ‘w’)
def studentInfo():
    lastName = input("Last name: ")
    firstName = input("First name: ")
    studentID = input("Enter ID: ")

    inFile = open("studentInfo.txt", "a")
    inFile.write("Name: " + firstName + " " + lastName)
    inFile.write("\nStudentID: " + studentID)
    inFile.write("\n")
    inFile.close()
    print("Done! Data is saved in file: studentInfo.txt")

e. Did you create the file: studentInfo.txt separately from the program code? ____________

f. Execute the program again using different input. Open the studentInfo.txt file. What is in the file? Is the data from the first program execution still there? ________________________________
POGIL – Activity 15: Question 1

def studentInfo():
    lastName = input("Last name: ")
    firstName = input("First name: ")
    studentID = input("Enter ID: ")

    inFile = open("studentInfo.txt", "a")
    inFile.write("Name: " + firstName + " " + lastName)
    inFile.write("\nStudentID: " + studentID)
    inFile.write("\n")
    inFile.close()
    print("Done! Data is saved in file: studentInfo.txt")

g. Change the ‘w’ to ‘a’ in the open() function. Execute the program again with different input. Examine the studentInfo.txt file. What did ‘a’ as an argument in the open() function do?

h. Notice the function – write(). How many arguments does the write function have?

i. How does the write() function know what file to write to?
def studentInfo():
    lastName = input("Last name:")
    firstName = input("First name:")
    studentID = input("Enter ID:")

    inFile = open("studentInfo.txt","a")
    inFile.write("Name: "+ firstName + " " + lastName)
    inFile.write("\nStudentID: "+ studentID)
    inFile.write("\n")
    inFile.close()
    print("Done! Data is saved in file: studentInfo.txt")

j. What line of code closes the file? Where is the line of code positioned in the program?

k. Rewrite the program so that the user can enter three names during one execution of the program. You may need to change the order of some of the code.
YOU SHOULD COMPLETE THE REST OF ALL POGILS OUTSIDE OF CLASS.

BEST DONE WITH A PARTNER OR STUDY GROUP.

CHECK YOUR ANSWERS ON A COMPUTER!
More About the Built-in Format Function

```python
print('{:4}'.format(a*b), end=' ', file=out)
```

- `print('Num is {}'.format(3))`
  - ‘Num is 3’
- `print('Num is {:10}'.format(3))`
  - ‘Num is 3’ → Gives 10 total characters to the {} (fills in with spaces)
- `print('Num is {:^10}'.format(3))`
  - ‘Num is 3’ → Center-aligns {} within those 10 spaces
More About the Built-in Print Function

```python
print('{:4}'.format(a*b), end='', file=out)
```

- **end=''**
  - print() usually ends with a ‘\n’ by default
  - Can change that!

- **file=out**
  - Print() usually prints to std out (the terminal) by default
  - Can change that!
Writing Files

You can also specify a file to print to using the `print(..)` function

```python
with open('table.dat', 'w') as out:
    for a in range(1, 10+1):
        for b in range(1, 10+1):
            print('{:4}'.format(a*b), end='', file=out)
    print(file = out)
# out.close() happens here, implicitly
```

See `printwrite.py` and `filewrite.py` examples