CS 111: Program Design I

Lecture Last: Review

Robert H. Sloan & Richard Warner
University of Illinois at Chicago
December 5, 2019

UIC

Working with real data?

- Better demonstrates how we can use tools we're learning on real-world data
- Is harder—more funny issues of special characters, data cleaning, etc.
- Is it worth it?
  A. Great experience; don't change a thing
  B. Pain in the neck but worth it
  C. Reduce but don't eliminate
  D. Probably not worth it
  E. Give us easier datasets even if "toy" data

Questions on chloropleth map assignment

Final Exam

- 3:30 pm – 5:30 pm Wednesday here in LC B
  - our regular classroom
- Comprehensive, but weighted towards topics at end
- Expect only minimal knowledge of pandas, geopandas, and networkx

Some important topics: Fast list

- Variables, expressions, operators, types
- Modules, their use, some we used
- functions, parameters, use
  - Local variables in functions
- Control flow: if (and else, elif), for, while
- Strings, methods, slices, in, looping over
- Lists, methods, slices, in, looping over
  - *Nested lists*

Topics, cont.

- Files, opening, reading, writing
- *Dictionaries* methods, use, lookup
- Software design, hierarchical decomposition, abstraction
- Debugging
- Basics of computers and their components
- Notion of a network, nodes, edges, degrees
  - Social network analysis: benefits and possible problems

Topics continued

- Basics of plotting with matplotlib
- Data science/analytics
- CFFA
- Cryptography
- The Supreme Court
- Fourth and Fifth Amendments

Please complete the post-class survey (now)

- https://oberlin.qualtrics.com/jfe/form/SV_cwOAbPB3KsUuoyF

Link also available from Welcome page of course Blackboard site

CS PRACTICE QUESTIONS, ESPECIALLY DICTIONARIES

d = {12:35, 35:35, 1:12}

- d[35]

A. 15
B. 35
C. 12
D. 1
E. Error

d = {12:35, 35:35, 1:12}

- len(d)

A. 0
B. 2
C. 3
D. 6
E. error

d = {'a':15, 'b':35, 'g':12}

- d['c']

A. -1

B. 0

C. Error

d = {'a':15, 'b':35, 'g':12}

- 'g' in d

A. True
B. False
C. 12
D. Error

d = {'a':15, 'b':35, 'g':12}

- 12 in d

A. True

B. False

C. g

D. Error

How useful have you found the question and discuss peer instruction method?

A. Very useful
B. Somewhat useful
C. Neutral
D. Not so useful
E. Big waste of time

To create a column of data to use geopandas plotting on the Chicago Community map we have to work with the area_num_1 (Community number) entry of the map. Why?

A. To provide a double check data is ok

B. Community geodata frame rows are *not* ordered by community number

C. It handles conversion of floats to ints

D. Funsies

E. I don't know

What is printed?

```
ls = [[4, [True, False], 6, 8], [28, 9]]
if ls[0][1][0]:
    print(ls[1][0])
else:
    print(ls[1][1])
```

A. 6
B. 8
C. 28
D. 9
E. I don't know

Is [5, 'squid' ,7] a valid list? How about [4]?

A. Both are

B. Neither is

C. [4] is, but not [5, 'squid', 7]

D. [5, 'squid', 7] is, but not [4]

E. I don’t know

Suppose A, B and C are boolean variables. Write a boolean expression that evaluates to true if and only if one or more of these variables are False.

A. (not A) and (not B) and (not C)
B. (not A) or (not B) or (not C)
C. not (A and B and C)
D. More than one of the above
E. I don't know

```
grade = 98
if (grade >= 90):
    print('You got an A!')
if (grade >= 80 and grade < 90):
    print('You got a B!')
if (grade < 80):
    print ( 'You got something else ')
```

Rewrite this code using Elif

```
grade = 98
if (grade >= 90):
    print('You got an A!')
elif (grade < 90):
    print('You got a B!')
elif (grade >= 80):
    print('You got a B!')
elif (grade < 80):
    print ( 'You got somethin')
```

A

```
grade = 98
elif (grade >= 90):
    print('You got an A!')
elif (grade >= 80):
    print('You got a B!')
elif (grade < 80):
    print ( 'You got somethin')
```

C

```
grade = 98
if (grade >= 90):
    print('You got an A!')
elif (grade >= 80):
    print('You got a B!')
else:
    print ( 'You got somethin')
```

B

```
grade = 98
if (grade >= 90):
    print('You got an A!')
elif (grade >= 80) and (grade < 90):
    print('You got a B!')
else:
    print ( 'You got somethin')
```

D

E. I don't know

Write a Python function called min(a,b,c) that takes parameters integers a,b,c and returns the smallest. (You *may not use built-in Python functions min or max*)

Write a Python function called min(a,b,c) that takes parameters integers a,b,c and returns the smallest.

```
def min(a,b,c):
    smallest = a
    if (b < smallest):
        smallest = b
    if (c < smallest):
        smallest = c
    return smallest
```

A

```
def min(a,b,c):
    smallest = a
    if (b < smallest):
        smallest = b
    if (c < smallest):
        smallest = c
min()
```

C

```
def min(a,b,c):
    min = a
    if (b < min):
        min = b
    if (c < smallest):
        min = c
```

B

```
def min(a,b,c):
    smallest = a
    if (b > smallest):
        smallest = b
    if (c > smallest):
        smallest = c
    return smallest
```

D

E. I don't know

```
first = [1, 2, 3]
second = [3, 4, 5]
x = first.append(second)
```

What is first?

A. [1, 2, 3, 3, 4, 5]
B. [1, 2, 3, [3, 4, 5]]
C. [1, 2, 3, 4, 5]
D. None
E. [1, 2, 3]

```
first = [1, 2, 3]
second = [3, 4, 5]
x = first.append(second)
```

What is x?

A. [1, 2, 3, 3, 4, 5]
B. [1, 2, 3, [3, 4, 5]]
C. [1, 2, 3, 4, 5]
D. None
E. [1, 2, 3]

```
s = 'pirate ship'
i=0
while(len(s) > i ):
    s=s[i:]
    i=i+1
    print (s , i )
```

What does this print?

A

```
pirate ship 1
irate ship 2
rate ship 3
ate ship 4
te ship 5
e ship 6
```

B

```
pirate ship 1
irate ship 2
ate ship 3
 ship 4
p 5
```

C

```
pirate ship 1
irate ship 2
rate ship 3
ate ship 4
te ship 5
e ship 6
 ship 7
ship 8
hip 9
ip 10
p 11
```

D. None of the above

E. I don't know