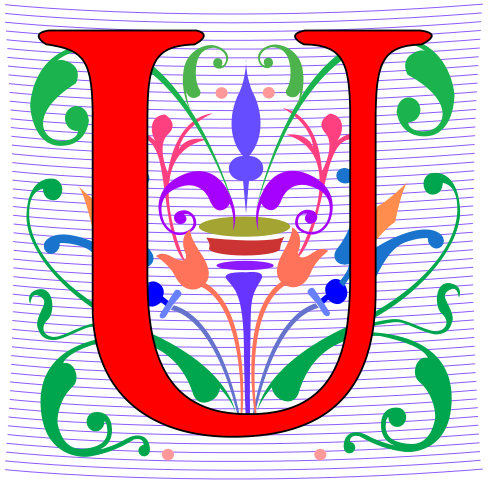
Compsci 101, Midterm Review

Owen Astrachan
Kristin Stephens-Martinez
April 5, 2018

U is for …

- **URL and URI**
  - io.org to artist:5XenQ7XfcvQdflbpLEFaKQ
- **Usenet**
  - Original source of FAQ, Flam, Spam, more
- **UI and UX**
  - User is front and center



Plan for the Day

- Reminder of modules os and pathlib
  - Not for midterm, more with next assignment

- Midterm review
  - Let's solve some problems

Processing all files in a folder

- Using os and os.path works
  - Across Python versions
  - Uses strings to identify paths to files/folders

- Using pathlib and pathlib.Path works
  - In Python3 and only with work in older versions
  - Uses Path objects to identify paths

- More later, details don't matter for midterm

List Comprehensions

- Given a list of words

```
words = ["ant", "bat", "orb", "fox", "ton"]
```

- What is

```
z = [w for w in words if w[-1] == 't']
```

- You create: list of all words that contain an 'o'
  - `["orb", "fox", "ton"]` works for any list

WOTO

http://bit.ly/101spring18-mid2-1

Set Examples

- We have two lists: w1 and w2
  - We want a single, sorted list of the strings in either w1 or w2. Store in variable **`all`** which contains every string in either w1 or w2 (elements in **`all`** are unique)

- Shorthand for **`s.intersection(t)`** is **`s & t`**
  - What about union?

Nested Loops

- Understanding indexing and looping

```
>>> for x in range(1,4):
...     for y in range(1,3):
...         print(x,y)
...
1 1
1 2
2 1
2 2
3 1
3 2
```

WOTO

http://bit.ly/101spring18-mid2-2

Motivating Dictionaries

- What work is done repeatedly in code below?
  - How many times is words made into a set?
  - How many times is w counted in words?

```
7 def most_frequent(fname):
8     words = open(fname).read().split()
9     tups = [(words.count(w),w) for w in set(words)]
10    return sorted(tups)[-1][1]
```

Dictionaries

- Recalculating: most frequent word in one pass

```
12 def most_frequent2(fname):
13     d = {}
14     for w in open(fname).read().split():
15         if w not in d.keys():
16             d[w] = 0
17         d[w] += 1
18     return [x for x in d if d[x] == max(d.values())]
```

Reversing Keys and Values

- How do go from A to B?

```
8 data = {"ola":["compsci101", "neuro101", "bio101"],
9         "rcd":["compsci308", "psych101", "neuro101"],
10        "ksm":["compsci101", "econ101","bio101"],
11        "nsf":["bio101","econ101","compsci101"]}

12 revd = {'compsci101': ['ola', 'ksm', 'nsf'],
13         'neuro101': ['ola', 'rcd'],
14         'bio101': ['ola', 'ksm', 'nsf'],
15         'compsci308': ['rcd'],
16         'psych101': ['rcd'],
17         'econ101': ['ksm', 'nsf']}
```

Developing RevDictionary

- We will need keys and values: types?
  - How will we initialize d[key]?
  - How will we update d[key]?

- How will we access the students (keys or values)?
- How will access the classes (keys or values)?

Tuples organized by class size

- We want a list of tuples ordered by class-size
  - Tuples in form (class-name, roster)
  - Ties broken by class-name alphabetically

```
('bio101', ['ola', 'ksm', 'nsf'])
('compsci101', ['ola', 'ksm', 'nsf'])
('econ101', ['ksm', 'nsf'])
('neuro101', ['ola', 'rcd'])
('compsci308', ['rcd'])
('psych101', ['rcd'])
```

Generating Ideas

- We have (str, list[strings]) tuples
  - We want to sort by length of list

- Can we access second element of tuple?
  - **`operator.itemgetter(2)`**
- Can we sort in reverse order? **`reverse=True`**

- Make temporary lisf of tuples: **`(str,list,int)`**
  - Where **`t[2`**] is the length of the list

Coping with Unfamiliar

- Can work to understand the unfamiliar
  - How do we sort by length of list? We haven't done that before?
  - What have we done before?

- Map to something familiar: sorting tuples
  - Sort first to break ties
  - Syntax of **`key=operator.itemgetter(1)`**

Questions