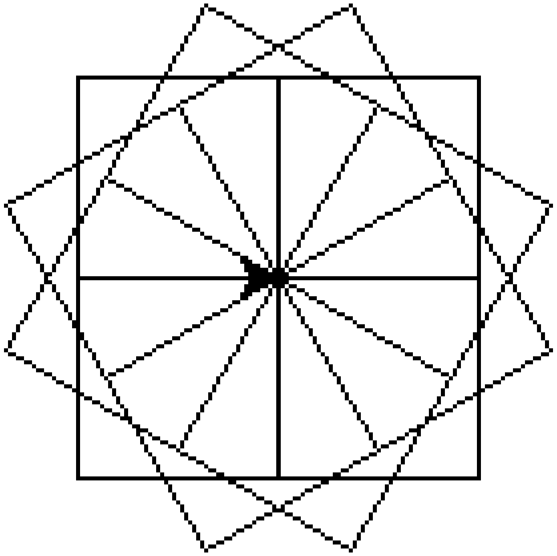
CompSci 101
Introduction to Computer Science



Sep 14, 2017

Prof. Rodger

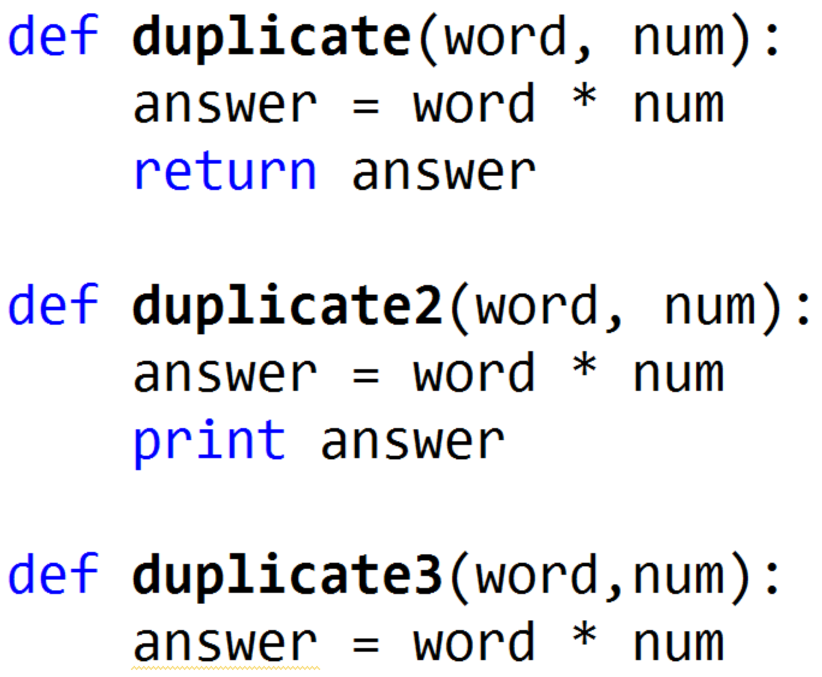
Announcements

- Reading and RQ6 due next time
- Assignment 2 due today, Assignment 3 out
- APT 2 due on Tuesday

- Today:
  - Problem solving: Strings, Lists
  - Looping over structures (characters, words) and building something

Review Functions

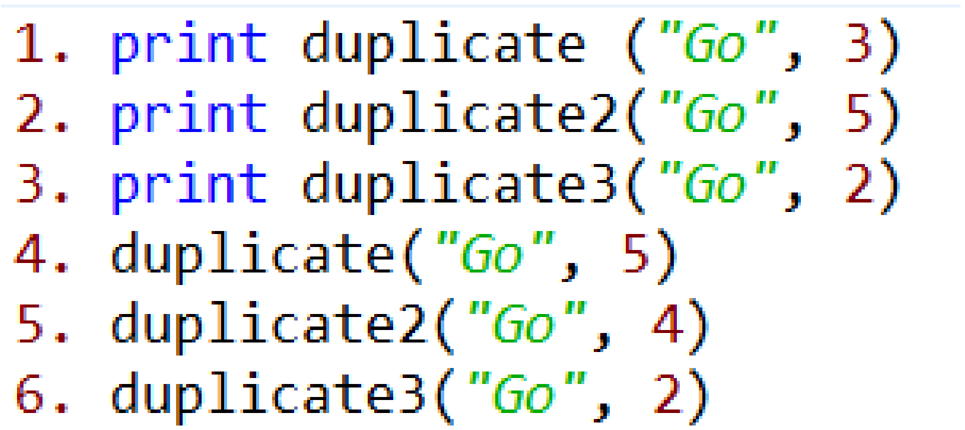
www.bit.ly/101f17-0912-3



```
def duplicate(word, num):
    answer = word * num
    return answer

def duplicate2(word, num):
    answer = word * num
    print answer

def duplicate3(word,num):
    answer = word * num
```



```
1. print duplicate ("Go", 3)
2. print duplicate2("Go", 5)
3. print duplicate3("Go", 2)
4. duplicate("Go", 5)
5. duplicate2("Go", 4)
6. duplicate3("Go", 2)
```

How many ways can I run Python in this course?

- Eclipse
  - Complete program
  - Interactive Console
  - APT
- Online textbook
  - We are using Python 2.7

    ‘/’ (2.7)   vs   ‘//’ (3)
- Python Tutor

Use Python Tutor

- Debug/trace your code
- Doesn’t work with input files

More on Strings

- Strings are indexed starting at 0
- Example: `‘word’`

w	o	r	d
0	1	2	3

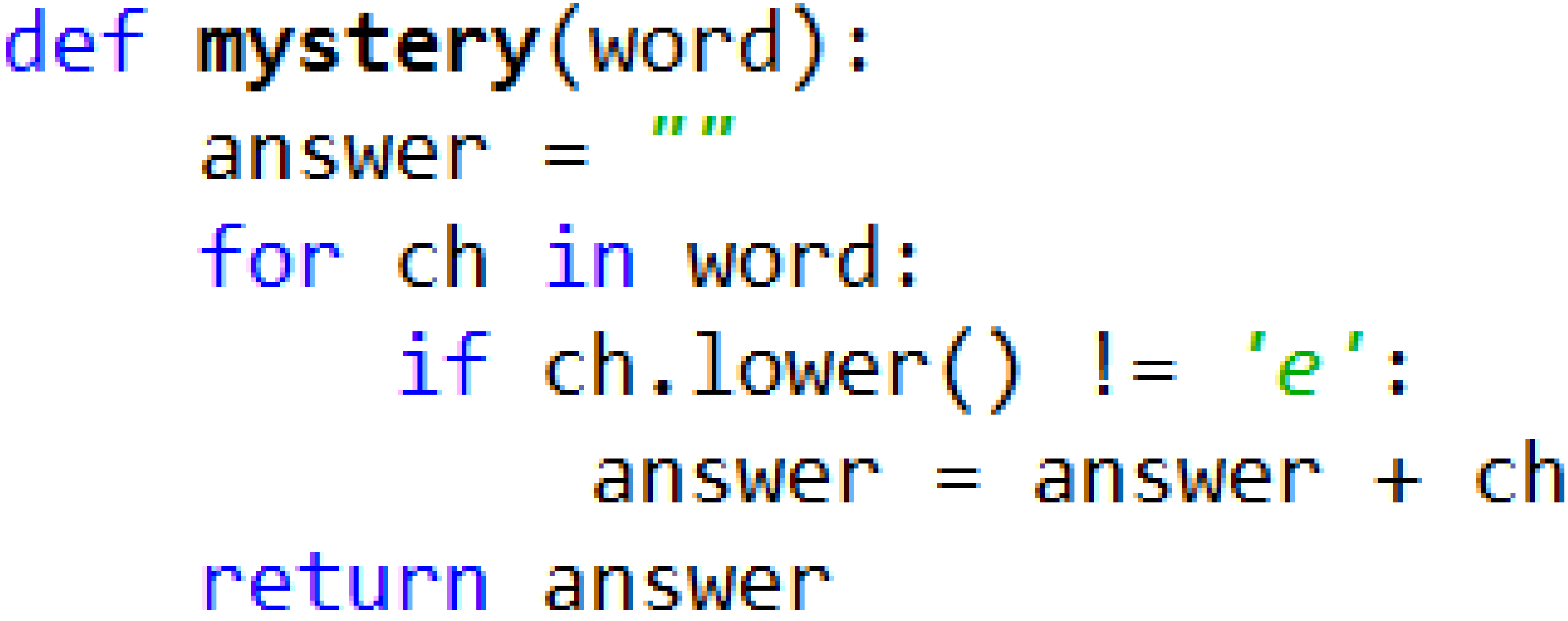
- Use [x] – to refer to a particular character in word
- Use [x:y] to refer to a slice of the string starting at position x and up to but not including position y. Can leave out x or y.

Examples
bit.ly/101f17-0914-1

```
phrase = "Duke Blue Devils"
1) phrase[0] + phrase[-3] + phrase[-2]*2
2) phrase[5:10] + phrase[:4]
3)(phrase[phrase.find('ev'):]).upper()
4) phrase[-5::2] + phrase[:4:-1]
```

String fun
Crazy import

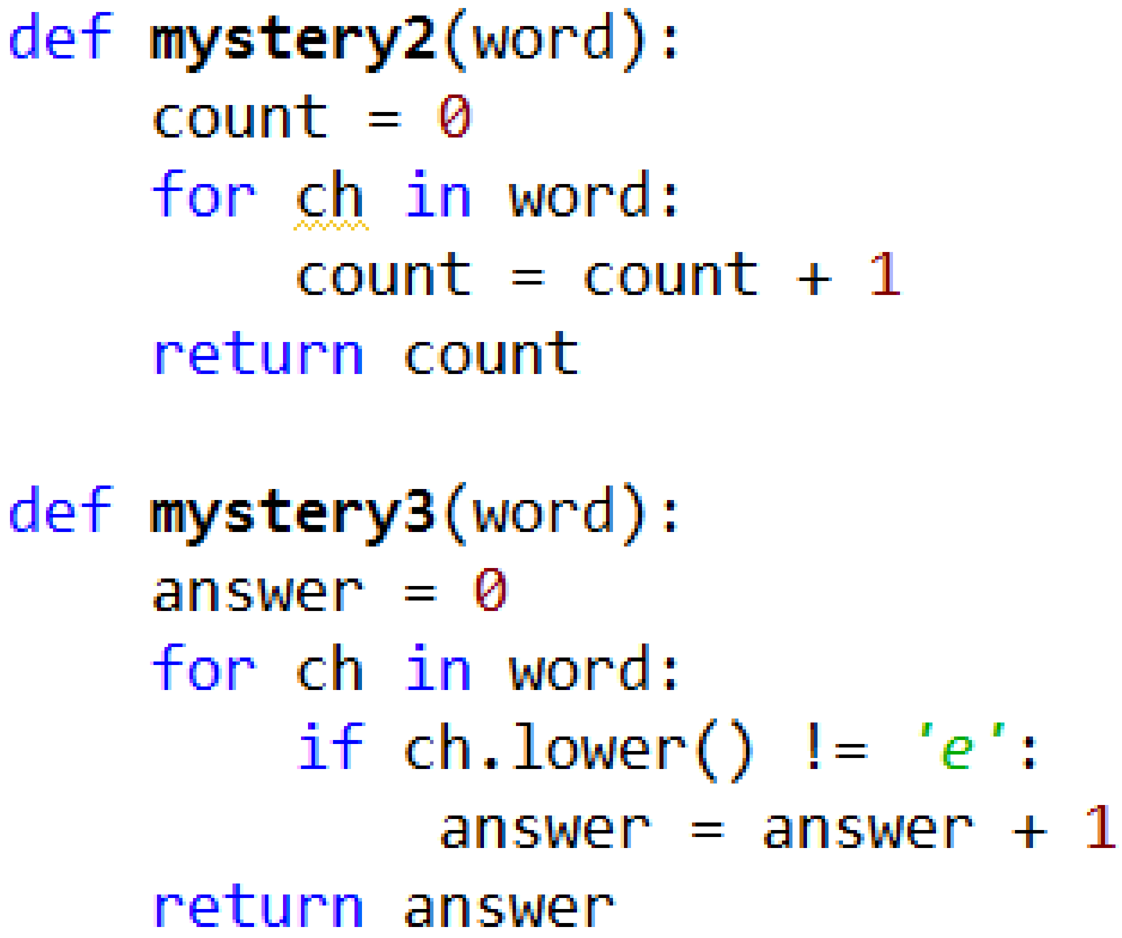
Loop over all characters in a String



```python
def mystery(word):
    answer = ""
    for ch in word:
        if ch.lower() != 'e':
            answer = answer + ch
    return answer
```

Loop over string

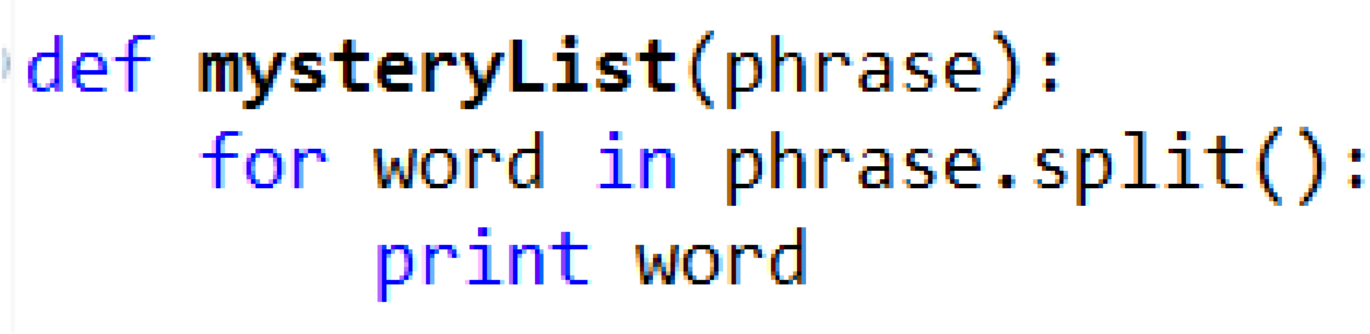
- www.bit.ly/101f17-0914-2



```
def mystery2(word):
    count = 0
    for ch in word:
        count = count + 1
    return count

def mystery3(word):
    answer = 0
    for ch in word:
        if ch.lower() != 'e':
            answer = answer + 1
    return answer
```

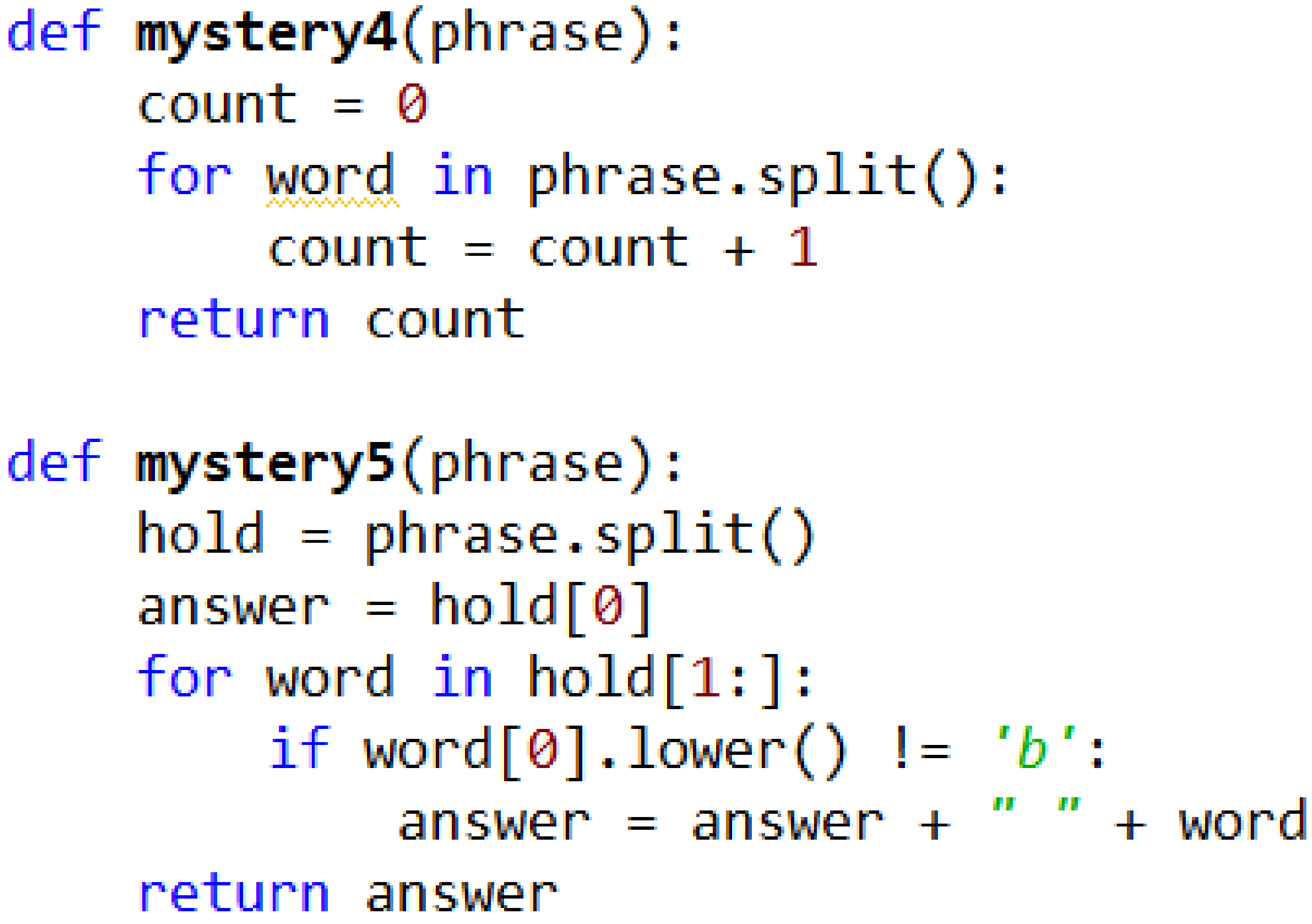
Loop over all words in a list



```
def mysteryList(phrase):
    for word in phrase.split():
        print word
```

Loop over words

- www.bit.ly/101f17-0914-3



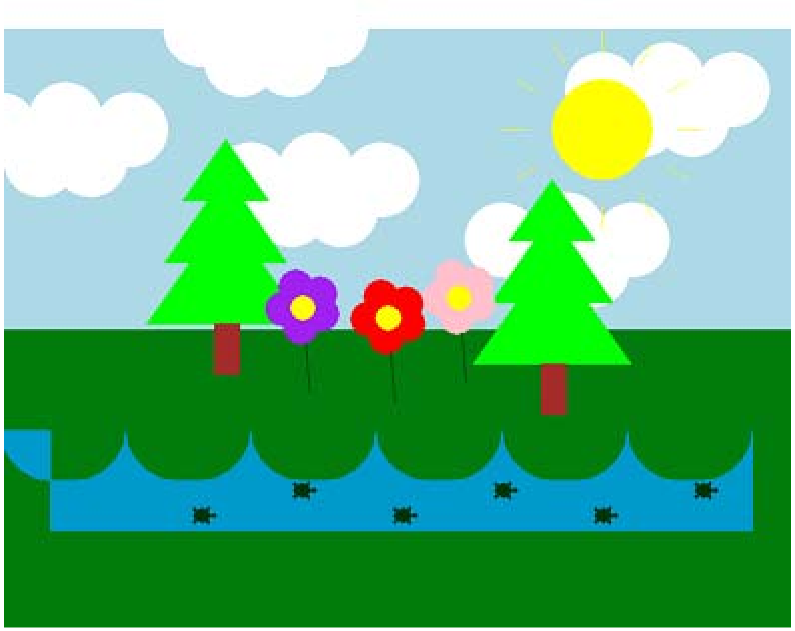
```python
def mystery4(phrase):
    count = 0
    for word in phrase.split():
        count = count + 1
    return count

def mystery5(phrase):
    hold = phrase.split()
    answer = hold[0]
    for word in hold[1:]:
        if word[0].lower() != 'b':
            answer = answer + " " + word
    return answer
```

Computer Science Alum

- Biology and CS
- Undergraduate Research - JFLAP
- Epic
- Now in Med School at Vanderbilt

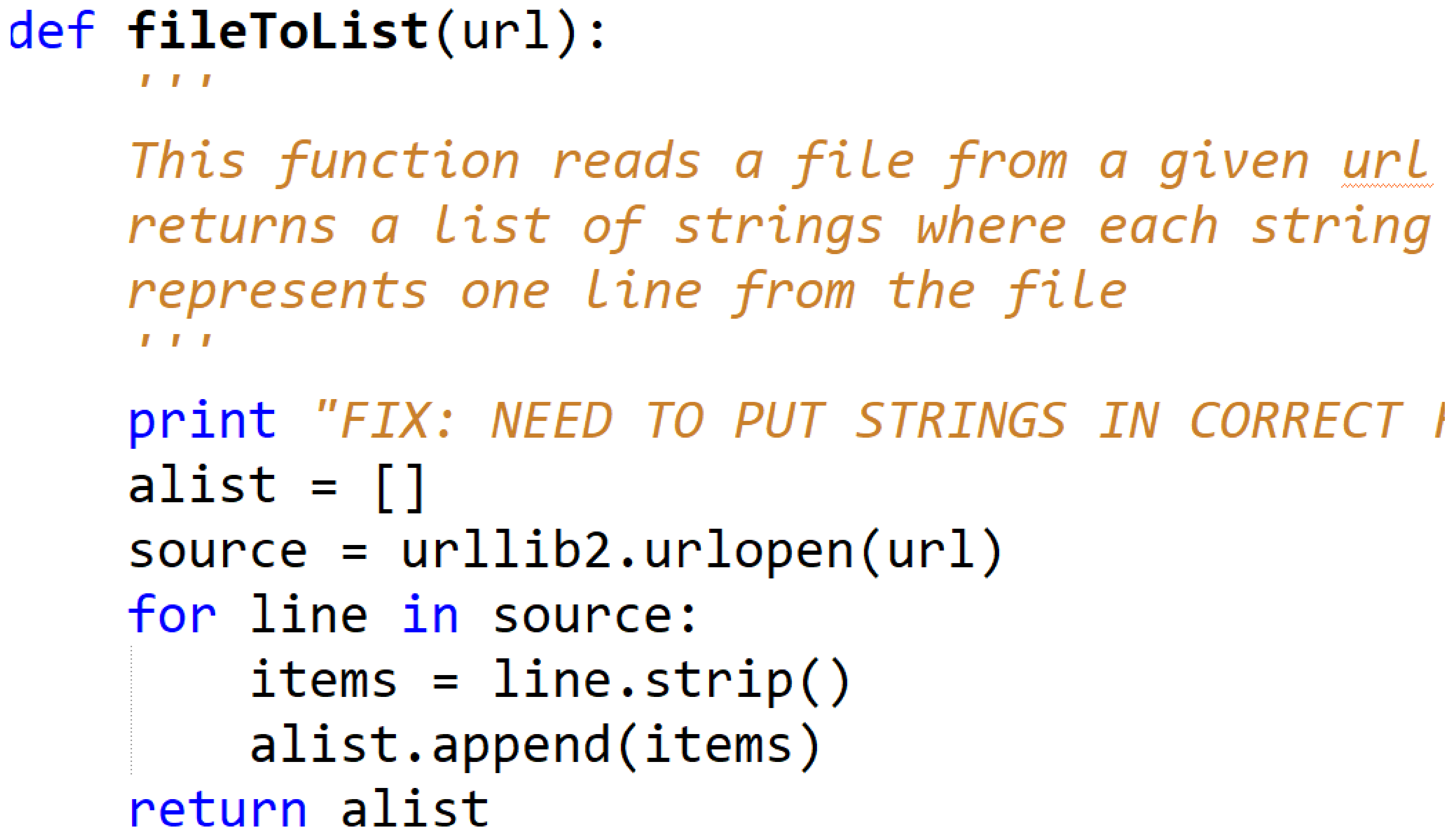
Assignment 3
snarf to use starter files



- Turtles
  - Creative
- Earthquakes
  - Data from last 30 days around the world
  - Example - Find the largest earthquake

Getting Started with Earthquake part

- Read lines of data into a list of strings



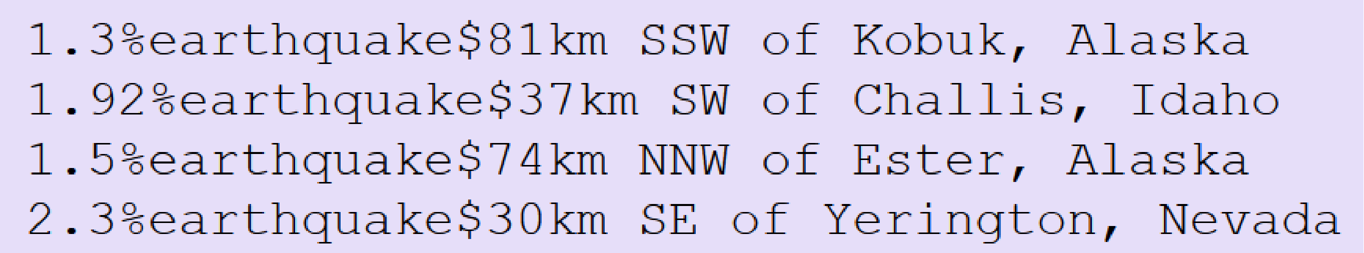
```
def fileToList(url):
    '''

    This function reads a file from a given url
    returns a list of strings where each string
    represents one line from the file
    '''

    print "FIX: NEED TO PUT STRINGS IN CORRECT
    alist = []
    source = urllib2.urlopen(url)
    for line in source:
        items = line.strip()
        alist.append(items)
    return alist
```

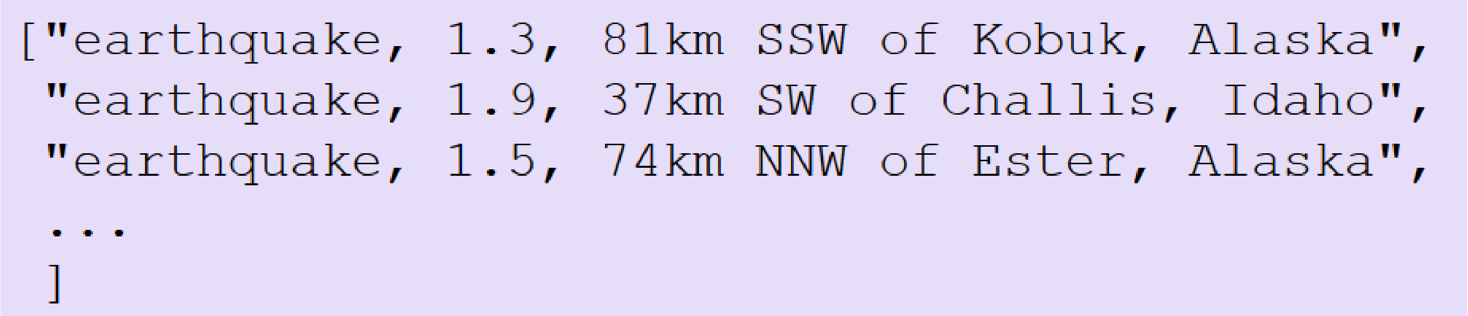
Getting Started with Earthquake part

- Here is first few lines of the small file:



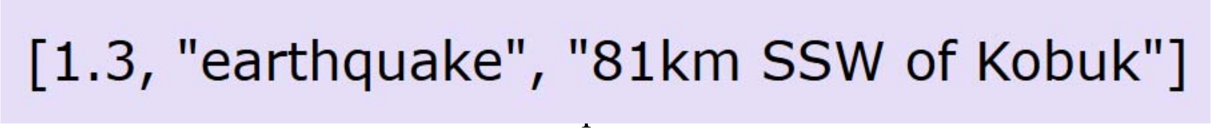
```
1.3%earthquake$81km SSW of Kobuk, Alaska
1.92%earthquake$37km SW of Challis, Idaho
1.5%earthquake$74km NNW of Ester, Alaska
2.3%earthquake$30km SE of Yerington, Nevada
```

- Read into a list, reformatting the lines



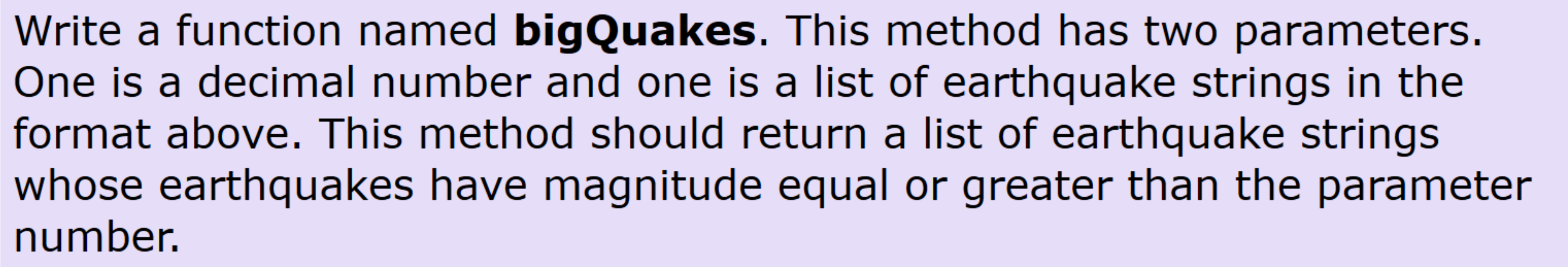
```
["earthquake, 1.3, 81km SSW of Kobuk, Alaska",
 "earthquake, 1.9, 37km SW of Challis, Idaho",
 "earthquake, 1.5, 74km NNW of Ester, Alaska",
 ...
 ]
```

- Write function getParts to get parts of a line

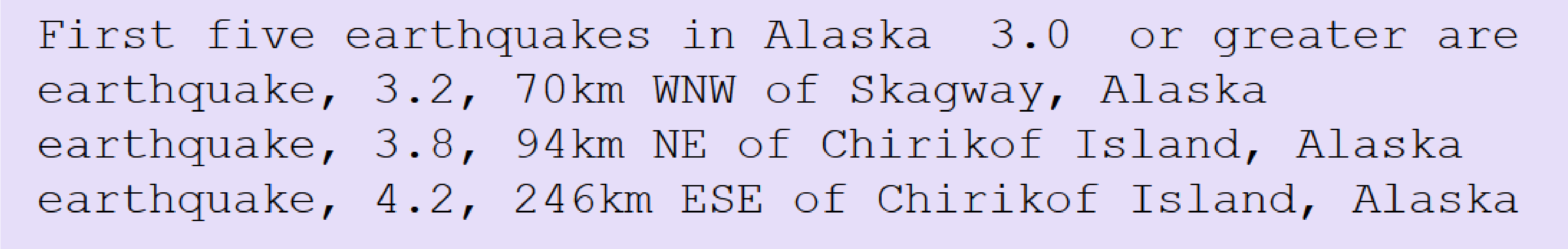


```
[1.3, "earthquake", "81km SSW of Kobuk"]
```

Use the list to calculate facts about earthquakes

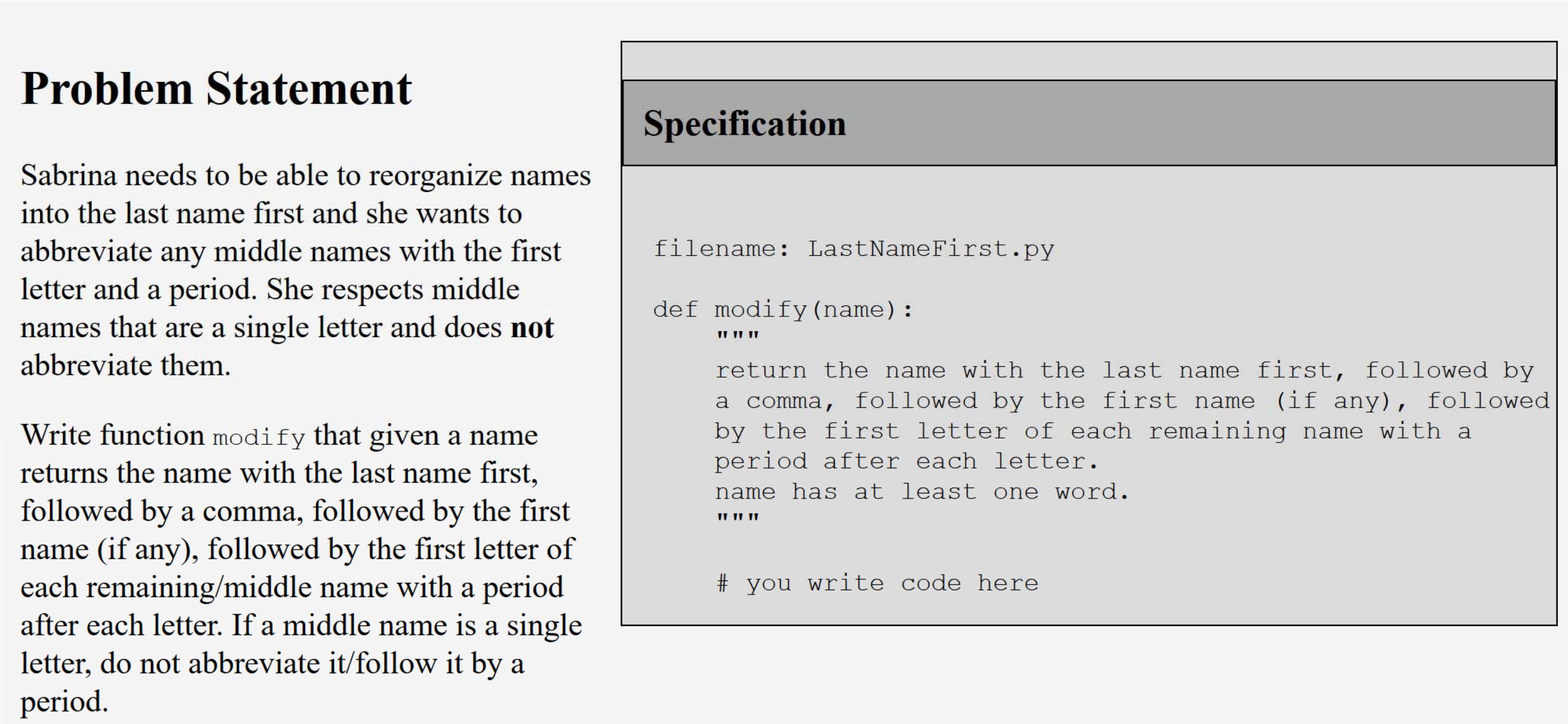


Write a function named **bigQuakes**. This method has two parameters. One is a decimal number and one is a list of earthquake strings in the format above. This method should return a list of earthquake strings whose earthquakes have magnitude equal or greater than the parameter number.



```
First five earthquakes in Alaska  3.0  or greater are
earthquake, 3.2, 70km WNW of Skagway, Alaska
earthquake, 3.8, 94km NE of Chirikof Island, Alaska
earthquake, 4.2, 246km ESE of Chirikof Island, Alaska
```

APT: Last Name First



Problem Statement

Sabrina needs to be able to reorganize names into the last name first and she wants to abbreviate any middle names with the first letter and a period. She respects middle names that are a single letter and does **not** abbreviate them.

Write function `modify` that given a name returns the name with the last name first, followed by a comma, followed by the first name (if any), followed by the first letter of each remaining/middle name with a period after each letter. If a middle name is a single letter, do not abbreviate it/follow it by a period.

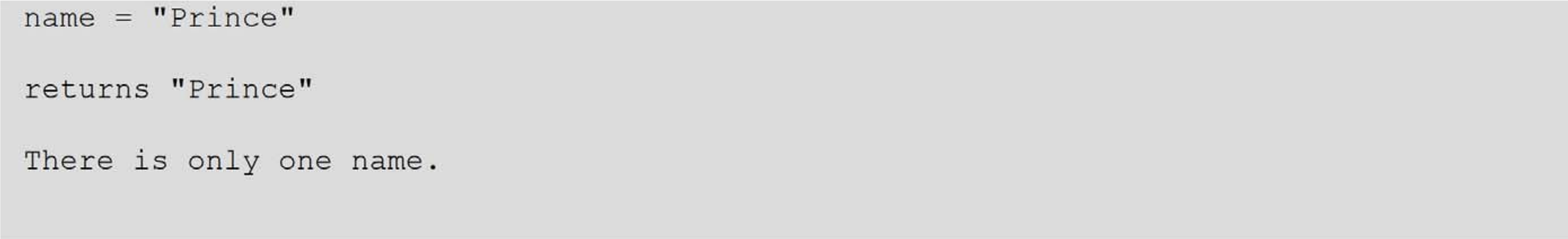
Specification

```
filename: LastNameFirst.py

def modify(name):
    """
    return the name with the last name first, followed by
    a comma, followed by the first name (if any), followed
    by the first letter of each remaining name with a
    period after each letter.
    name has at least one word.
    """

    # you write code here
```

2.

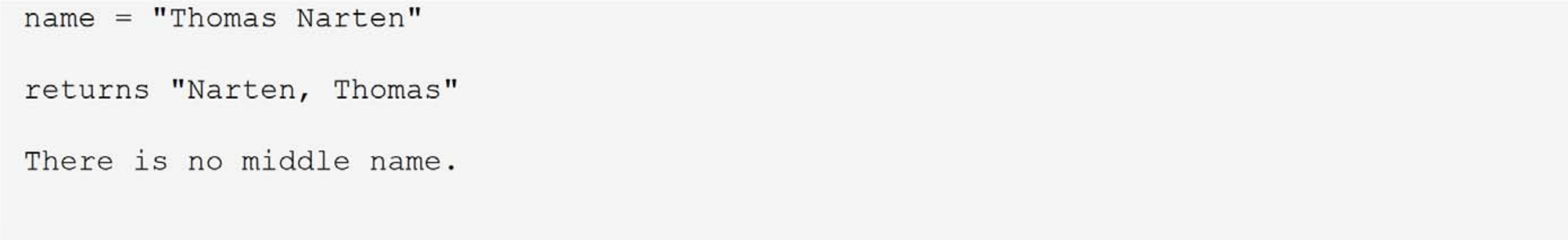


```
name = "Prince"

returns "Prince"

There is only one name.
```

3.

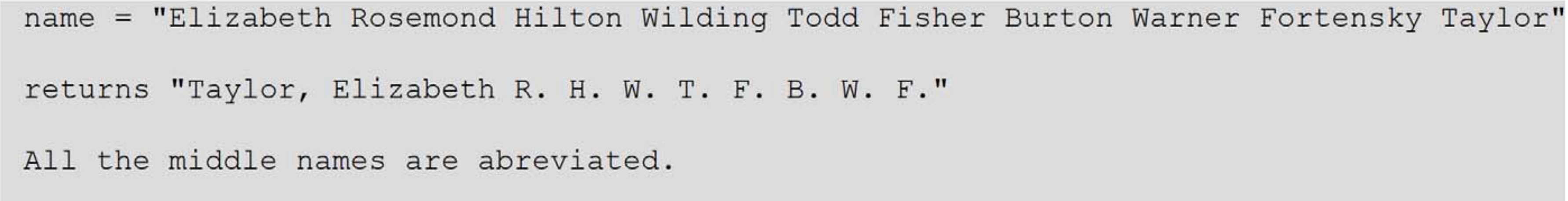


```
name = "Thomas Narten"

returns "Narten, Thomas"

There is no middle name.
```

4.



```
name = "Elizabeth Rosemond Hilton Wilding Todd Fisher Burton Warner Fortensky Taylor"

returns "Taylor, Elizabeth R. H. W. T. F. B. W. F."

All the middle names are abreviated.
```

LastNameFirst APT

http://www.cs.duke.edu/csed/pythonapt/lastnamefirst.html

Answer Questions here:

bit.ly/101f17-0914-4

Problem Solving to Code
7 Step Process

1. Work small examples by hand
2. Write down what you did in words (algorithm)
3. Find Patterns (generalize algorithm)
4. Work another example by hand (does your algorithm work? If not, go back to 2)
5. Translate to code
6. Test several cases
7. Debug failed test cases

Use 7 step process to solve LastName First