

ENGL 516X:
Methods of Formal Linguistic Analysis
Semester: Spring '18

Instructor: Sowmya Vajjala

Iowa State University, USA

20 Feb 2018

Class outline

- ▶ Assignment 3 Discussion
- ▶ Assignment 4 Description
- ▶ Lists recap questions
- ▶ Tips for using lists
- ▶ Reading and Writing files in Python
- ▶ Resource for visualizing programs: <http://pythontutor.com>

Assignment 3 Discussion

- ▶ Volunteer for discussing Q3

Assignment 3 Discussion

- ▶ Volunteer for discussing Q3
- ▶ String method `.count()` - why it won't work here:
`"baabaaaaa".count("aa")`

Assignment 4 Description

- ▶ Deadline: 10th March (changed from 3rd March)
- ▶ 10% of your grade (last individual assignment)
- ▶ 2 questions: This covers parts of what you learnt before (regular expressions) and what we will cover in this week
- ▶ This is actually quite difficult compared to Assignments 1–3. So, start early.
- ▶ Note: Deadline is 10th March, but I am unavailable (even on email) after 8th.

Lists recap-1

What will the following print?

```
def funct1(lst1):  
    lst1.sort()  
  
def funct2(lst1):  
    lst2 = lst1[:]  
    lst2.sort(reverse=True)  
    return lst2  
  
lst1 = [9,8,7,12,1,2]  
print(funct1(lst1))  
print(lst1)  
print(funct2(lst1))  
print(lst1)
```

Lists Recap-2

let us analyse this code

```
string = input("Enter a paragraph of text:\n")
splitString = string.split(" ")
varN = 5
result = ""
for loopvar in range(0,len(splitString)):
    if loopvar == varN and varN < len(splitString):
        result = result + " " + "____"
        varN = varN +5
    else:
        result = result + " " + splitString[loopvar]
#Question: What does result have now?
print(result)
```

Use a test string: "What is happening here? What is this code doing?"

Tips for using Lists - 1

1. Be aware that some of the list methods manipulate the list and do not return anything `lst1.sort()` sorts the list `lst1` directly..and not save the result as a new list!

Tips for using Lists - 1

1. Be aware that some of the list methods manipulate the list and do not return anything `lst1.sort()` sorts the list `lst1` directly..and not save the result as a new list!
2. There are multiple ways of doing the same thing with minor differences (`+` vs `append`, `l1.sort()` vs `sorted(l1)` etc.). Pick one idiom and stick to that to avoid confusion.

Tips for using Lists - 1

1. Be aware that some of the list methods manipulate the list and do not return anything `lst1.sort()` sorts the list `lst1` directly..and not save the result as a new list!
2. There are multiple ways of doing the same thing with minor differences (+ vs `append`, `l1.sort()` vs `sorted(l1)` etc.). Pick one idiom and stick to that to avoid confusion.
3. Make a copy of the list before you use functions like `sort()`, to avoid aliasing.

Tips for using Lists - 1

1. Be aware that some of the list methods manipulate the list and do not return anything `lst1.sort()` sorts the list `lst1` directly..and not save the result as a new list!
2. There are multiple ways of doing the same thing with minor differences (+ vs `append`, `l1.sort()` vs `sorted(l1)` etc.). Pick one idiom and stick to that to avoid confusion.
3. Make a copy of the list before you use functions like `sort()`, to avoid aliasing.
4. It is easy to fall to write a code that will break easily, especially when we are using regular expressions, `split()` etc. So, be careful (and patient). Handle possible exceptions, use print statements to do some debugging.

Tips for using lists - 2

What is a potential problem with this code?
(Ignore the first two lines. My question is related to the rest of the code.)

```
fhand = open('mbox-short.txt')
for line in fhand:
    words = line.split()
    if words[0] != 'From':
        continue
    print(words[2])
```

File processing in Python

- ▶ There are three basic operations involved: opening a file, reading its contents, and writing content into a new or existing file.
- ▶ We will be talking primarily about files with text content in this course.
- ▶ Why should we know file processing?: Several advantages. We can do corpus analysis easily, for example.
- ▶ Knowing how to use files along with regular expressions is sufficient to do a lot of text processing even if there are no additional any additional language tools (like parsers, taggers etc).

Opening a file

- ▶ You open a file with `open()` function.
`open("/home/Desktop/a.txt")` creates a "handle" for the file if the file really exists on your computer. Otherwise, it throws an error.
- ▶ Assuming that I have `mbox-short.txt`, what will happen if I type this?

```
fhand = open('mbox-short.txt')  
#this looks for the file in your current folder.  
print(fhand)
```

Opening a file

- ▶ You open a file with `open()` function.
`open("/home/Desktop/a.txt")` creates a "handle" for the file if the file really exists on your computer. Otherwise, it throws an error.
- ▶ Assuming that I have `mbox-short.txt`, what will happen if I type this?

```
fhand = open('mbox-short.txt')  
#this looks for the file in your current folder.  
print(fhand)
```

- ▶ Assuming that I don't have a file that i am trying to open, what will happen?

```
fhand = open('some_random_name.txt')  
print(fhand)
```

Opening a file

- ▶ You open a file with `open()` function.
`open("/home/Desktop/a.txt")` creates a "handle" for the file if the file really exists on your computer. Otherwise, it throws an error.
- ▶ Assuming that I have `mbox-short.txt`, what will happen if I type this?

```
fhand = open('mbox-short.txt')  
#this looks for the file in your current folder.  
print(fhand)
```

- ▶ Assuming that I don't have a file that i am trying to open, what will happen?

```
fhand = open('some_random_name.txt')  
print(fhand)
```

- ▶ One way to avoid such errors is to use try-except loops while trying to read data from files.

Reading a file

- ▶ There are two ways to read a file in python. Read line by line (makes more sense for a text file). Read as a whole (useful for text files, and also other formats).
- ▶ Reading line by line:

```
fhand = open('mbox-short.txt')  
content = ""  
for line in fhand:  
    content = content + line
```

- ▶ Reading the whole file at once:

```
fhand = open('mbox-short.txt')  
content = fhand.read()
```

Checking if a file really exists

- ▶ As mentioned earlier, one way is to use try-except loop and see if there is a "FileNotFound" error when you try to read the file.
- ▶ Other way is to use a built in module `os` and its `isfile()` function.
- ▶ example at: `CheckFilePath.py`

Searching through a file

One example:

```
fhand = open('mbox-short.txt')
for line in fhand:
    line = line.rstrip()
    if line.startswith('From:') :
        print(line)
```

Searching through a file

One more example:

```
fhand = open('mbox-short.txt')
for line in fhand:
    line = line.rstrip()
    if line.find('@uct.ac.za') == -1 :
        continue
    print(line)
```

Searching through a file

One more example:

```
fhand = open('mbox-short.txt')
for line in fhand:
    line = line.rstrip()
    if line.find('@uct.ac.za') == -1 :
        continue
    print(line)
```

This will print only lines that have that string "@uct.ac.za".

More examples of searching

.. in regular expressions chapter, lot of practice while doing Assignment 4.

Writing into a file

- ▶ You still have to "open" a file, but add an extra parameter called "w"

```
writeHandle = open('output.txt', 'w')
```

- ▶ You then use the write() method of this handle to write content into the file.

```
writeHandle.write("this is a line\n")
```

- ▶ The handle will not do a "enter" keypress. So, you need to put your own newlines.

- ▶ Once you are done, close the handle.

```
writeHandle.close()
```

- ▶ Caution: If the file already exists, doing this will delete the old data! so be careful! If the file doesn't exist, a new one is created.

Getting a list of all files in the directory

- ▶ Sometimes, we want to work with a bunch of files instead of a single file.
- ▶ It is useful to have a way to list all files (or just .txt files or .py files etc.).
- ▶ ...and later we can loop through the directory, process one file after another iteratively.
- ▶ Go to: ListFilesInDir.py

Time for a small exercise

Try to write a small code to read a file given by the user, and give: a pdf file as input, and see what happens. Give it one of your .py files as input and see what happens.

Post solution on the forum

Try to write a small code that will take a folder/directory path, and lists the number of lines of all files in that directory (if it can actually read it!).

Next Class

- ▶ Two useful data structures in Python: Dictionaries and Tuples (chapters: 9–10)
- ▶ Next Tuesday: Recap, Practice. So, post any questions you have online in the forum with "Recap: 27 Feb" as the title.