ENGL 516X: Methods of Formal Linguistic Analysis

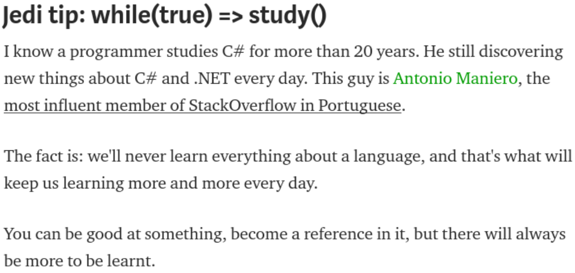
Semester: Spring '18

Topic: Lists in Python

Instructor: Sowmya Vajjala

Iowa State University, USA

15 Feb 2018



Jedi tip: while(true) => study()

I know a programmer studies C# for more than 20 years. He still discovering new things about C# and .NET every day. This guy is Antonio Maniero, the most influent member of StackOverflow in Portuguese.

The fact is: we'll never learn everything about a language, and that's what will keep us learning more and more every day.

You can be good at something, become a reference in it, but there will always be more to be learnt.

source: https://hackernoon.com/how-to-learn-a-new-programming-language-faster-dc31ec8367c

5 points to improve your programming logic

- Think, split a problem into sub problems
- practice
- learn about different data structures, some common algorithms
- learn general programming paradigms in that language
- look at other people's code

source: `https://hackernoon.com/5-points-to-improve-your-programming-logic-23c8bbafe8d2`

Class Outline

- Lists in Python
- Practice with Lists
- Note: We will get back to regular expressions on Tuesday again.
- Reminder: Assignment 3 is due this weekend.

What is a List?

- A list is a sequence of values. A string can be seen as a list of characters. There can be a list of strings as well.
- These values in a list are called ”elements” or ”items”.
- Lists in Python are identified by the presence of square brackets.
- Examples:
  1. [516,515,520,540]- is a list of integers
  2. [”Python”,”Java”,”Perl”,”R”,”Ruby”]- is a list of strings.
  3. [’spam’, 2.0, 5, [10, 20]]- is list with elements of various data types. There is a list inside a list too!

An Example list and its use

Look at this code:

```
str = "Look at this code"
demoList = str.split(" ")
#Splits the string wherever there is a space.
print(demoList)
[’Look’, ’at’, ’this’, ’code’]
print(len(demoList)
# prints the number of items in a demoList object. 4 here.
print(demoList[1])
# prints "at".
```

Lists and their components: An Exercise

Type the following on the console and see what happens.

```
complexList = [1, "complex list", ["nested", "item"]]
print(complexList[1])
print(complexList[2])
print(complexList[3])
print(complexList[2][0])
print(complexList[2][1])
print(len(complexList))
print(len(complexList[2]))
anotherList = [1, 2, 3]
print(complexList, anotherList)
```

Lists are Mutable

- What is mutability? Are strings mutable?

Lists are Mutable

- What is mutability? Are strings mutable?
- Strings are immutable. But lists are mutable.
- Try to guess what each of these lines do, and then type them on console and see what is happening with print statements.

  ```
  numbers = [1, 2, 3, 123]
  print(numbers[1])
  newNum = numbers[0]
  numbers[0] = 2
  print(numbers)
  ```

Traversing a List

… is the same as traversing a string.

```
numbers = [1, 2, 3, 123]
#One way:
for item in numbers:
  print(item)
#Another way:
for index in range(0,len(numbers)):
  print(numbers[i])
```

Why traverse using an index?

What will this code do?

```
numbers = [1, 2, 3, 123]
print(numbers)
for index in range(0,len(numbers)):
  numbers[index] = numbers[index]*numbers[index]
print(numbers)
```

List operations

- + operator concatenates lists.

  ```
  a = [1,2,3]
  b = [4,5,6]
  c = a+b
  print(c)
  #this gives you:
  [1,2,3,4,5,6]
  ```

- * operator repeats a list given number of times.

  ```
  print(a*3)
  [1, 2, 3, 1, 2, 3, 1, 2, 3]
  ```

List slicing

Some part of the usage is the same as in strings.

```
c = [1,2,3,4,5,6]
print(c[1:3])
[2,3]
```

List slicing

Some part of the usage is the same as in strings.

```
c = [1,2,3,4,5,6]
print(c[1:3])
[2,3]
```

A slice operator, for lists, however can be used on the left hand side, to change list contents.

```
c[1:3] = [8,9]
print(c)
[1, 8, 9, 4, 5, 6]
```

List "Methods"

1. append(): takes one argument and adds it as a new element to the end of the list.
2. extend(): takes a list as argument and appends all items in the list to the current list.
3. sort(): sorts list elements from low to high.

Important note: All these three methods change the list. They do not return the value into a new variable.

List "Methods": Exercise

Go to python console and try the following things:

1. a = [1,2,3]
2. a.append(4)
3. print(a)
4. print(a.append(4,5))
5. print(a)
6. print(a.append([1,2,3]))
7. print(a)
8. print(a.extend([1,2,3]))
9. print(a)
10. b = [1,4,5,8,0,3]
11. b.sort()
12. print(b)

extend vs +

What is the difference between extend and +? Figure this out yourself later, by trying what you know so far, by searching online etc.

Deleting elements in a list

Different ways:

1. if you know the index of the element, use pop(index) or del operator
2. If you know the element you want to remove (but not the index), you can use remove()

Deleting elements in a list: exercise

Try the following in the Python console.

1. a = [1, 2, 3, 4, 5, 6]
2. a.pop()
3. print(a)
4. a.pop(1)
5. print(a)
6. del(a[1])
7. print(a)
8. a.remove(1)
9. print(a)

Lists and Functions

Type the following in the Python console and observe the output:

1. nums = [3, 41, 12, 9, 74, 15]
2. print(len(nums))
3. print(max(nums))
4. print(min(nums))
5. print(sum(nums))

Lists and Strings

Type the following in the Python console and observe the output:

1. string1 = ”python”
2. alist = list(string1) #converts string to a list
3. print(alist)
4. string2 = ”this is a longer string”
5. blist = string2.split()
6. print(blist)
7. string3 = ”there are strings, lists, int, and float”
8. clist = string3.split(”,”) #, is called the delimiter.
9. print(clist)
10. comma = ”,”
11. string4 = comma.join(clist)
12. print(string4)

”Aliasing”

Let us say you want to copy the contents of one list into another. How do you go about that?

”Aliasing”

Let us say you want to copy the contents of one list into another. How do you go about that? Let a = [1, 2, 3] be a list. Will it be sufficient if I say b = a in Python?

"Aliasing"

Let us say you want to copy the contents of one list into another. How do you go about that? Let a = [1, 2, 3] be a list. Will it be sufficient if I say b = a in Python? Try these things on the console:

- a = [1,2,3]
- b = a
- b[0] = 9
- print(a)

Did you see what you expected to see?

So, how should I copy from one list to another, then?

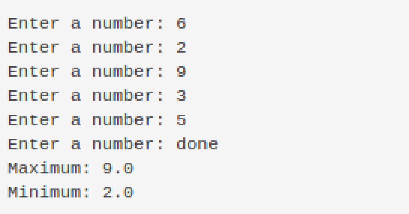
- a = [1,2,3]
- firstcopy = list(a)
- secondcopy = a[:]

There are also other ways, by using ”copy” module in python. Find out yourselves!

Practice Exercise: 1

Exercise 6 in Chapter 8

Write a program that interacts with user like this, using lists and list functions:



```
Enter a number: 6
Enter a number: 2
Enter a number: 9
Enter a number: 3
Enter a number: 5
Enter a number: done
Maximum: 9.0
Minimum: 2.0
```

Practice Exercise: 2

Modify the user input in the above program such that you take all input in one line (space separated integers). Write any code for exception handling. i.e., example interaction should be:

```
Enter your numbers separated by space: 1 2 3 4 5 6
Maximum: 6
Minimum: 1
```

(Use lists)

Additional Practice Exercises

- Create a list containing 100 random integers between 0 and 1000 (use a loop, append function, and the random module). Write a function called average that will take the list as a parameter and return the average.
- Write a function sum_of_squares(xs) that computes the sum of the squares of the numbers in the list xs, and use it in main().
- Count how many words in a list have length 5.
- Sum all the elements in a list up to but not including the first even number.

More in the 2nd textbook: `http://interactivepython.org/runestone/static/thinkcspy/Lists/Exercises.html`

Next Week

- Topics: use of Tuples and Dictionaries in Python; Reading and Writing content from Files
- Readings: Chapters 7, 9 and 10 in the text book
- For Tuesday: Read - Chapter 7 (Files)
- Do: Exercises at the end of lists chapter.
- Assignment 3 is due this weekend!