

Lab 3: Lists and Strings

CSE/IT 107

NMT Computer Science

“Each decision we make, each action we take, is born out of an intention.”

— Sharon Salzberg

“Programming is learned by writing programs.”

— Brian Kernighan

“The purpose of computing is insight, not numbers.”

— Richard Hamming, 1962

1 Introduction

In the first two labs, we showed you how to generally use Python and how to have your programs make decisions based on what the user entered. With this lab, we will be starting to show you how to do some useful things in Python: how to make lists and manipulate them and how to deal with strings. You have seen strings before, but we will be showing you how to do neat things to them.

2 Lists

2.1 Summary

2.2 Exercises

3 Strings

You have seen strings in Python before. They are sequences of characters enclosed by either double quotes or single quotes; for example:

```
1 >>> print(s)
2 I'm a string.
3 >>> r = 'I am also a string.'
4 >>> print(r)
5 I'm also a string.
```

Notice how we did not use a single quote in the second string because it was enclosed (*delimited*) by single quotes. The proper way to use a single quote in a single quoted string or a double quote in a double quoted string goes like this:

```
1 >>> s = "Previously, we said \"I'm a string.\"."
2 >>> print(s)
3 Previously, we said "I'm a string.".
4 >>> r = 'I\'m also a string.'
5 >>> print(r)
6 I'm also a string.
```

This is called *escaping* a character. We *escaped* the double quotes and single quote respectively so that Python did not think it was the end of the string.

We also saw indirectly and previously that we can concatenate strings together using the addition operator +:

```
1 >>> s = "The cat"
2 >>> r = " in the hat"
3 >>> t = s+r
4 >>> print(t)
5 The cat in the hat
```

In addition to that, we can repeat strings using the multiplication operator *:

```
1 >>> s = "Hi"
2 >>> r = 5*s
3 >>> print(r)
4 HiHiHiHiHi
```

You previously saw *slicing* in the section on lists. Slicing works on strings, too!

```
1 >>> s = "The cat in the hat"
2 >>> print(s[4:7])
3 cat
4 >>> print(s[15:18])
5 hat
6 >>> print(s[14:18])
7  hat
8 >>> print(s[17:14:-1])
9 tah
10 >>> print(s[17::-1])
11 tah eht ni tac ehT
```

If you want strings to go on for multiple lines, you have to use three double quotes:

```
1 weizsaecker = """We in the older generation owe to young people not the
2 fulfillment of dreams but honesty. We must help younger people to
3 understand why it is vital to keep memories alive. We want to help them
4 to accept historical truth soberly, not one-sidedly, without taking
5 refuge in utopian doctrines, but also without moral arrogance. From our
6 own history we learn what man is capable of. For that reason we must not
7 imagine that we are quite different and have become better. There is no
8 ultimately achievable moral perfection. We have learned as human beings,
9 and as human beings we remain in danger. But we have the strength to
10 overcome such danger again and again."""
```

3.1 Summary

3.2 Exercises

4 For Loops Again

4.1 Summary

4.2 Exercises

5 Submitting

Files to submit:

- .py (see Section)

You may submit your code as either a tarball (instructions below) or as a .zip file. Either one should contain all files used in the exercises for this lab. The submitted file should be named either `cse107_firstname_lastname_lab3.zip` or `cse107_firstname_lastname_lab3.tar.gz` depending on which method you used.

For Windows, use a tool you like to create a .zip file. The TCC computers should have 7z installed. For Linux, look at lab 1 for instructions on how to create a tarball or use the “Archive Manager” graphical tool.

Upload your tarball or .zip file to Canvas.