Prelab 6

Before starting this Pre Lab please read through chapters 6 and 8 in your textbook.

Basic List Operations

List is one of the powerful datatypes available in Python. Creating a list is very simple:

```
List1 = ['CS', 'Math', 'Bio']
List2 = [1, 2, 3, 4]
List3 = ['Fall', 2012, 'Spring', 2011]
```

Create and populate a List using range

```
>>> myList = list(range(5))
>>> print (myList)
[0, 1, 2, 3, 4]
```

List Operations

Python provides a number of basic list operations for manipulating lists. Let us first review the + and * operations:

```
>>> myList = [0,1,2,3]
>>> newList = myList + myList
>>> print(newList)
[0, 1, 2, 3, 0, 1, 2, 3]
```

The + operator allows us to concatenate ("merge") two list together. In the above example we concatenated the list [0,1,2,3] with [0,1,2,3]. We could have accomplished this in another way:

```
>>> myList = [0,1,2,3]
>>> newList = myList*2
>>> print(newList)
[0, 1, 2, 3, 0, 1, 2, 3]
```

Just like in math, we know that multiplication can be expressed in terms of addition. This is why these two examples produce the same result in Python. Let us now consider what the following example produces:

```
>>> myList = [0,1,2,3]
>>> newList = myList + [myList]
>>> print(newList)
```

Computer Science Courses - http://courses.cs.purdue.edu/
In the code above we have modified our first example by placing brackets ([ ]) around the variable myList. What does this do? It creates a list that has one element. That element is the value that is stored in the variable myList. In this case myList contains the list [0,1,2,3]. Therefore, the result of this operation is the list: [ [0,1,2,3] ]. We now perform the following addition: [0,1,2,3] + [ [0,1,2,3] ]. This operation concatenates these two lists together yielding the new list: [0,1,2,3, [0,1,2,3]]. Notice that the last element of this list is the sublist that was stored in the variable myList. The difference between this piece of code and the example at the beginning of the prelab is whether or not we want [0,1,2,3] as a sublist or not.

**Basic String Operations**

A String is a list of characters. One way to create a string is by surrounding your word or phrase with quotation marks and assign it to a variable

```python
>>> myString = "Hello Strings"
>>> print (myString)
Hello Strings
```

Strings can concatenated using the plus (+) operator:

```python
>>> print ("Hello" + " Strings")
Hello Strings
```

**Substrings**

You can get a substring, or slice, from a string by using a range expression. A range expressions uses a colon to indicate the range:

```python
>>> myString = "hello"
>>> print (myString[1:4])
ell
```

**String Methods**

Every variable in Python is an object that may have methods. Methods are like functions except they apply to functions. In order to execute a method on an object, we use the dot notation. The dot notation takes the form: object.method

```python
>>> myString = "Hello Strings"
>>> print(myString.upper())
HELLO STRINGS
```

Explore the following methods by yourself:
Plotting

In order to create 2D plots in python, you need to install the following libraries:

numpy: http://www.lfd.uci.edu/~gohlke/pythonlibs/#numpy

pylab (part of matplotlib): http://www.lfd.uci.edu/~gohlke/pythonlibs/#matplotlib

Important Note to Mac users: until this moment, these libraries are not supported in Mac OS. You need to use the machines in Windows Lab in LWSN to work with these libraries.

Plot Single Graph

```python
import numpy
import pylab # part of matplotlib

def graphIt(xList, yList):
    pylab.plot(xList, yList, 'b')
    pylab.show()

def main():
    x = list(range(10))
    y = list()
    for k in range(10):
        y.append(k*k)
    graphIt(x, y)
```
Plot Multiple Graphs

```python
import numpy
import pylab # part of matplotlib

def graphIt2(xList, yList, zList):
    pylab.plot(xList, yList, 'b')
    pylab.plot(xList, zList, 'r')
    pylab.show()

def main():
    x = list(range(10))
    y = list()
    z = list()
    for k in range(10):
        y.append(k*k)
        z.append(100-k*k)
    graphIt2(x, y, z)
```

Figure 1
From: http://courses.cs.purdue.edu/ - Computer Science Courses

Permanent link: http://courses.cs.purdue.edu/cs17700:prelab6

Last update: 2012/09/21 10:46