CS 177

Lists & Plotting

Week 6
Announcements

- Project 2 due next Monday
- Next Tuesday is review session
- Midterm 1 on Wed., EE 129, 8:00 – 9:30pm
ANY QUESTIONS?
Table of Contents

- Loops
- Lists
- String Operations
- Plotting functions
Two Types of Loops

- For loops
  - Loops have more controlled structure and cannot run infinitely
  - Commonly used with range function

- While loops
  - Expresses an execution condition
    - Only executes until that condition no longer holds
For Loop

- Used when you know the number of iterations beforehand

```python
n = eval(input("# you want factorial of?"))
factorial = 1
for i in range(n):
    factorial = factorial * i
print("Factorial of", n, "is", factorial)
```
For Loop

- Used when you know the number of iterations beforehand

```python
n = eval(input("# you want factorial of?"))
factorial = 1
for i in range(1, n+1):
    factorial = factorial*i
print("Factorial of", n, "is", factorial)
```
While Loop

- Used when you have to have a condition met
- If you have a menu like project 2, where you want to continue to ask the user what to do until they want to exit the game

```python
x = 0

while(x != 8):
    x = eval(input("What to do? Enter 8 to exit"))
```
Lists

- Multiple elements are stored consecutively in memory
- Lists can be nested
  - \([[[1,2], 4, [6, 10, [11, 12]], 5]\]
- Lists can also be empty
  - \([]\)
- Lists elements can be of different types
  - \([1, 2, ["ABCD", "EFG"], 3, 4]\)
Indexing Lists

- Indexing provides us with a quick mechanism for accessing a given element within the list.
- The index starts at 0 NOT 1 so the last element is the length of the list – 1.
- $L[i]$ will give you the $i$th element of the list.

Example
- $L = [0, 2, 4, 6, 8]$
- What does $L[3]$ point to?
- What does $L[5]$ point to?
- What does $L[-4]$ point to?
Indexing Lists

- Indexing provides us with a quick mechanism for accessing a given element within the list.
- The index starts at 0 NOT 1 so the last element is the length of the list – 1.
- $L[i]$ will give you the $i$th element of the list.
- Example
  - $L = [0, 2, 4, 6, 8]$
  - What does $L[3]$ point to? 6
  - What does $L[5]$ point to?
  - What does $L[-4]$ point to?
Indexing Lists

- Indexing provides us with a quick mechanism for accessing a given element within the list.
- The index starts at 0 NOT 1 so the last element is the length of the list – 1.
- \( L[i] \) will give you the \( i \)th element of the list.

Example
- \( L = [0, 2, 4, 6, 8] \)
- What does \( L[3] \) point to? 6
- What does \( L[5] \) point to? ERROR
- What does \( L[-4] \) point to?
Indexing Lists

- Indexing provides us with a quick mechanism for accessing a given element within the list.
- The index starts at 0 NOT 1 so the last element is the length of the list – 1.
- \( L[i] \) will give you the \( i \)th element of the list.

Example

- \( L = [0, 2, 4, 6, 8] \)
- What does \( L[3] \) point to? 6
- What does \( L[5] \) point to? ERROR
- What does \( L[-4] \) point to? 2
Appending to a List

- Adding an element to the end of a list

```python
l = list(range(0, 10))
l.append(10)
l.append(11)
l.append(12)
print(l)
```
Appending to a List

- Adding an element to the end of a list

```python
l = list(range(0, 10))
l.append(10)
l.append(11)
l.append(12)
print(l)
```

0 1 2 3 4 5 6 7 8 9 10 11 12
Nested Lists

- You can have lists within a list
- You can also index the list within a list

Let's say:
\[ L = [2, [4, 6, 8], "ABC", ["DE", "FGHI"], 10] \]

- \( L[1] \) would print out [4, 6, 8]
- \( L[1][2] \) would print out 8
- \( L[3][0] \) would print out DE
- \( L[3][1][2] \) would print out H
Exercise

l = list(range(-5, 5, 2))
l.append(7)
l.append(-7)
print(l)
l[2] = 0
l[5] = 11
print(l)
Exercise

```python
l = list(range(-5, 5, 2))
l.append(7)
l.append(-7)
print(l)
l[2] = 0
l[5] = 11
print(l)
```
Exercise

```python
l = list(range(-5, 5, 2))
l.append(7)
l.append(-7)
print(l)
l[2] = 0
l[5] = 11
print(l)
```
Objects and Methods

- Methods are similar to functions
  - We can think of a Method as a function or an *action* performed on an Object

- An Object can be anything
  - More concretely, any python value
    - We will revisit this definition
    - For now we will consider strings as Objects
  - Different types support different methods

- `Object.Method(arguments)`
Advanced String Operations

- `print` ("hello world".capitalize())
  - Hello world

- `print` ("Hello World".endswith("Hello"))
  - False

- `print` ("HeLIO WoRlD".lower())
  - hello world

- `print` ("Hello World".upper())
  - HELLO WORLD

- `print` ("Hello World".find("World"))
  - 6
Exercise

b = “Hey There”
a = b.lower()
d = b.endswith(“there”)  
c = a.capitalize()
print(b)
print(a)
print(d)
print(d)
print(c)
Exercise

```python
b = "Hey There"
a = b.lower()
d = b.endswith("there")
c = a.capitalize()
print(b)  # Hey There
print(a)
print(d)
print(d)
print(c)
```
Exercise

b = “Hey There”
a = b.lower()
d = b.endswith(“there”)
c = a.capitalize()
print(b)
print(a)
print(d)
print(d)
print(c)
Exercise

b = “Hey There”
a = b.lower()
d = b.endswith(“there”)  
c = a.capitalize()
print(b)
print(b)
print(a)
print(d)
print(d)
print(c)
print(c)
Exercise

b = "Hey There"
a = b.lower()
d = b.endswith("there")
c = a.capitalize()
print(b)
print(a)
print(d)
print(c)

Hey There
hey there
False
Hey there

NOTE: b.lower does not change the value in b at all it only performs the method on the string and stores it in a.
Plotting

- Requires libraries.
  - numpy
  - matplotlib

- Make sure you get the right version!

- Handy link:
  - http://www.lfd.uci.edu/~gohlke/pythonlibs/
Basic plotting

• Single graph: xList and yList needed; must have same length

Example:
```python
import numpy
import pylab

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

x = list(range(-5, 6))
y = list()
for k in range(-5, 6):
    y.append(k**2)
graphIt(x, y)
```

NOTE: The b does nothing except draws the plot in blue.
Graph
ANY QUESTIONS?