Prelab 7

We encourage you to work together on the Pre Lab. The Pre Lab is not graded but will help you prepare for your lab session. If you have any questions on the material of the Pre lab, first check the book and recitation slides. If you continue to have any doubt about it, please email your recitation TA or the course instructor.

This prelab has two parts:

- The first one explains how to access (that is how to read and/or write) elements in a list of lists
- A very interesting example of lists of lists are digital images. Actually images are encoded in Python as lists of lists. The second part of this pre-lab explains how to manage digital images, that is the functions provided by the Python Image Library.

Review of List

A list is a collection of elements. A list can contain elements of the same type or it can contain elements of different types.

```python
# Example 1
list0 = ['CS', 'MATH', 'MUSIC', 'BIO']
list1 = [10, 20, 30, 40]
list2 = ['Freshman', "Sophomore", 'Junior', "Senior"]
list3 = ['lab1', 80.52]
list4 = ['lab2', 90]
list5 = ['lab3', 75]
list6 = ['lab4', 91]
```

In the example above, list0-list2 contain elements of the same type, while list3 contains elements of different types:

- list0 contains four elements. Each element is of character string type
- list1 contains four elements. Each element is of type integer
- list2, like list0, contains four elements, each of character string type
- list3 to list6 contains two elements. The first one is a string of characters; the second one is a floating point number

list3 to list6 may represent the grade you got in the first four labs of, say, CS177.

All the examples above show how to build a list. You simply assign to a variable name the sequence of its elements enclosed in [] brackets.

You can identify each of the list elements by using an index. Remember that the index starts from 0, not from 1. Why is this useful? Suppose you want to extract the grade of project1 from list3 above and assign it to a variable named myLab1Grade. Well, the value of the grade is the element at position 1 in list3, so all you need to do is to write:
# Example 2

```python
list3 = ['lab1', 80.52]
myLab1Grade = list3[1]
```

```python
>>> print(myLab1Grade)
80.52
```

## List of lists

Since a list can contain elements of any type, an element of a list can be another list. Suppose you want to build a list containing the grade you got in the first four labs of CS177. You can do that simply using the code below:

```python
# Example 3
myLabGrades = [['lab1', 80.52], ['lab2', 90], ['lab3', 75], ['lab4', 91]]
```

But, if you already created the list3,...,list6 (see Example 1), you can create the list myLabGrades in a smarter way:

```python
# Example 4
myLabGrades = [list3, list4, list5, list6]
```

```python
>>> print(myLabGrades)
[['lab1', 80.52], ['lab2', 90], ['lab3', 75], ['lab4', 91]]
```

```python
>>> print(len(myLabGrades))
4
```

You also saw in the prelab6 how to use the `+` operator to concatenate two (or more lists). Note that concatenating list3,...,list6 would have produced a very different result:

```python
# Example 5-a
myLabGrades = list3 + list4 + list5 + list6
```

```python
>>> print(myLabGrades)
['lab1', 80.52, 'lab2', 90, 'lab3', 75, 'lab4', 91]
```

```python
>>> print(len(myLabGrades))
1
```

In example 5 myLabGrades has length of 1, this list has only one element that contains 8 values (strings and floats). In example 4 myLabGrades has length of 4, it contains four elements (lists) an each one with 2 values(strings and floats).

We can access each value of the new list using a for loop:

```python
# Example 5-b
# for-loop to index all the values
for i in range(len(myLabGrades)):
    print(myLabGrades[i])
```

#output:
Indexing into a List which contains Lists

You already know how to access an element of a list using an index. Now we will review how to index nested lists.

# Example 6

# Creating a List using other Lists
students = [list0, list1, list2]

>>> print (students)
[['CS', 'MATH', 'MUSIC', 'BIO'], [10, 20, 30, 40], ['Freshman', 'Sophomore', 'Junior', 'Senior']]

# display the first element of list students (that is list0):
>>> print (students[0])
['CS', 'MATH', 'MUSIC', 'BIO']

# display the first element of sub-list list1 in students list:
>>> print (students[1][0])
0

# display the first element of sub-list list2 in students list:
>>> print (students[2][0])
'Freshman'

Using for-loops with Lists

If we want to display each element in a specific List we can use a “for” loop to traverse each index of the List:

# Example 7-a

# a for-loop to access and print the first(0) element from each sub-list in the students list:
for i in range(len(students)):
    print (students[i][0])

#output:
CS
10
Freshman

Or we can use the for-loop to access all the second(1) elements in each sub-list:

```python
# Example 7-b
for i in range(len(students)):
    print (students[i][1])

#output:
MATH
20
Sophomore
```

If we want to print all the lists[i] and all the values from each sub-list we need to use two for loops:

```python
# Example 8
for i in range(len(students)):
    print ("List: "+str(i))
    for n in range(len(students[i])):
        print (students[i][n])

#output:
list: 0
CS
MATH
MUSIC
BIO
list: 1
10
20
30
40
list: 2
Freshman
Sophomore
Junior
Senior
```

Let's review the statements in the for loop:

First we need to know what are the values of len(students) and len(students[i])

```python
>>> print (len(students))   #number of elements in students-List
3
>>> print (len(students[1])) #number of values in the first element(list)
4
```

The first loop will go over all the elements in the students list. Remember that each element of the list students is a list.
We can store the values of different elements in a new list. The following example shows how to create a new list and append to it the new values. The “newlist” works like an accounting variable which keeps track of multiple values every time the loop iterates.

```
# Example 9
students = [list0, list1, list2]

for i in range(len(students)):
    # this is called outer loop
    # initialize a new list. Everytime the loop runs the newlist is empty
    newlist = []
    print("List: "+str(i))
    for n in range(len(students[i])):  # this is called inner loop
        newlist.append(students[i][n])  # append the values of each list
    print(newlist)

#output:
List: 0
['CS', 'MATH', 'MUSIC', 'BIO']
List: 1
[1, 2, 3, 4]
List: 2
['Freshman', 'Sophomore', 'Junior', 'Senior']
```

The first values of the newlist are:

```
newlist.append(students[i][n])
students[i][n] --> point to the first list0
students[i][n] --> point to the first list0 and value0
n0
['CS']
Second value:
   n0  n1
['CS', 'MATH']
Third value
   n0  n1  n2
['CS', 'MATH', 'MUSIC']
Fourth value
   n0  n1  n2  n3
['CS', 'MATH', 'MUSIC', 'BIO']
```

The same process is repeated for the other lists.

**Dealing with images**
Software you need

In order to read images in python, you need to install the following Python Imaging Library:

- PIL: http://www.pythonware.com/products/pil/

Important Note to Mac users: these libraries are not yet supported for Python 3.2 in Mac OS. You need to use the machines in Windows Lab (LWSN B160 or HAAS G056) to work with these libraries.

Open an image

To load an image from a file, use the open(file) function from the Image module. Be careful to write the correct “namefile” and the extension (.jpg or .gif, etc).

```python
from PIL import Image
myImg = Image.open("namefile.gif")
```

The 'myImg' is now an object that is converted into RGB and it has attributes to examine the file-image content.

Convert image to RGB

The RGB color model represents the red, green, and blue colors which are added together in multiple ways to reproduce an array of other colors.

```python
#To convert an image into RGB, use the convert(mode) function from the Image module
myImg = Image.convert('RGB ')
```

Get the size of the Image

To get the size of the image in pixels, use the size function with the object you created (myImg)

```python
Width, height = myImg.size
```

The size function returns 2-tuple attributes containing width and height.

Get the pixels

To get the pixel at a given position, use the getpixel(x,y) function with the object you created (myImg)

```python
r,g,b = myImg.getpixel(x,y)
```
The `getpixel` function returns a value or a tuple containing the pixels of the image. Depending of the image it requires to have at least 3 variables (r,g,b) to unpack all the pixels.

**Put the pixels**

To modify the pixel at a given position, use the `putpixel(xy, colour)` function with the object you created (`myImg`)

```python
myImg.putpixel((x, y), (r, g, b))
```

The `putpixel` function modifies the color of a pixel at a given position. The colour is given as a single or tuple numerical values (r,g,b)

**Show an image**

To show an image, use the `show()` function from the `Image` module.

```python
myImg.show()
```

**Save an image**

To save an image to a file, use the `save(file)` function from the `Image` module.

```python
myImg.save("namefile.gif")
```

Be careful to write the correct “namefile” and extension (.jpg or .gif, etc).

**Image-Example**

```python
# Example 10
from PIL import Image
def displayImg():
    myImg = Image.open("purdue.gif")
    myImg = myImg.convert('RGB')
    width, height = myImg.size

    for x in range (width):
        for y in range (height):
            r, g, b = myImg.getpixel((x, y))

            myImg.putpixel((x, y), (0, 0, b))
```
myImg.save("blue.gif")
myImg.show()