**Class 2: Introduction to Python**

#### **Review from class 1**

Data types:

* Numbers
  + 1, 4, 23
  + 2 + 89, 4 \*\* 8
* Strings
  + “bob”, “banana”, “cake”
  + “hello “ + “world”
  + “hello”[3], “hello”[2:3]
  + “hello”.capitalize(), “hEllo”.swapcase()
  + len(“hello”), print(“hello”)
  + A complete list of built in string functions is at http://docs.python.org/2/library/stdtypes.html#string-methods
* Lists
  + [2, 5, 100], [True, False, True], [“bob”, “banana”, “elephant”], [[1, 3], [4, 5]]
  + list = [2, 5, 100]
  + list [0], list[1]
  + len(list), print(list)
  + 5 in list
  + A complete list of built in functions is at http://docs.python.org/2/library/stdtypes.html#sequence-types-str-unicode-list-tuple-bytearray-buffer-xrange

Programming concepts

* Variables
  + word = “banana”
  + print(word + “ split”)
* Types of functions
  + Operators: Symbols that can be used to perform operations
    - 3 + 5, “banana” \* 5, “banana”[3],
    - List of python operators is at http://docs.python.org/2/library/operator.html#mapping-operators-to-functions
  + Methods: functions on a particular object
    - “banana”.capitalize()
  + Functions: functions with no particular object these will generally be applicable to more than one data type
    - len(“banana”), print(“banana”)

#### **Functions**

Sometimes it’s useful to re-use code in multiple places. This can be done by writing and calling a function.

*Remember, you can create a file that contains the program by selecting File->New Window in IDLE. Save the file and select Run->Python Shell to run it.*

*Hint: pay attention to indentation!*

#### Example 1

#### Write a function that will sing birthday song:

def sing\_happy\_birthday(name):

happy = "Happy Birthday to "

print happy + "you"

print happy + "you"

print happy + name

print happy + "you"

sing\_happy\_birthday("Molly")

#### Exercises

1. Write a function called \_sum that takes in two numbers and returns their sum. [BONUS] What happens when you give this function two strings or two lists?
2. Write a function called first that returns the first letter of a string. What happens when you pass the function the empty string (“”)?
3. Write a function that takes a word and prints out the word five times.
4. Write a function that takes a string, *word* and a number *repeats* and prints out *word* *repeats* times with a space between each time word comes out. (Hint you probably will print out one more space than you will need.)
5. Write some code that calls that function, and asks for user input to select the word.
6. Write a function that takes a temperature and converts it from Fahrenheit to Celsius. (*hint:* C = 5/9 (*F*-32))

#### **Loops**

Sometimes you might want to do something more than once in your program. What do you think this code does?

*Example 1:*

for number in [1, 2, 3, 4, 5]:

print number

You can loop through any type of list. Here’s an example with strings.

*Example 2:*

for color in [“green”, “red”, “blue”, “banana”]:

print “this is a “ + color + “ banana.”

Things that aren’t indented will not be done in the loop, things that are intended will be done in the loop, so will be repeated for each element in the loop:

*Example 3:*

for color in [“green”, “red”, “blue”, “banana”]:

print color.capitalize() + “ is the color of my true love’s hair”

print “La di da da”

This code will produce:

Green is the color of my true love's hair

Red is the color of my true love's hair

Blue is the color of my true love's hair

Banana is the color of my true love's hair

La di da da

#### Exercises:

1. Use a for loop to get the computer to print:

1

little monkeys

2

little monkeys

3

little monkeys

(use separate print statements for the number and the text)

1. Create a new copy of the program (File->New Window) and modify it to use words (one, two, three) instead of numbers.
2. Find out how the range function works. Use it to create a list.
3. Use a loop to write a function that takes a list of numbers and prints out the sum of all those numbers. [BONUS] Write a function that computes the average of all the numbers. What happens when you give these functions a list of strings?
4. [Challenge] Take in a list of words and print out a numbered list. (Hint: You will have to use len() and range())
5. [Challenge] Write a function to print out a right justified Mario staircase of a height specified in the function argument. A staircase of height 4 is below:

# #

# # #

# # # #

# # # # #

#### **Python Imaging Library (PIL) example**

http://ywla.bigomega.org/

|  |
| --- |
| from PIL import Image  # Identify the image specified in the call of the program  inputImage = "Pictures/cupcake.gif"  print "Manipulating the image " + inputImage  # Open the Image  im = Image.open(inputImage).convert('RGB')  print "\nThis is the origional image data ..."  imageValues = list(im.getdata())  print imageValues[:10]  # Main Image Processing  # \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  newImageValues = [] # Initialize new values  for pixel in imageValues:  red = pixel[0]  green = pixel[1]  blue = pixel[2]  # Remember that the pizel order is red green blue  newPixel = (green, blue, red)  newImageValues.append(newPixel)  # \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  print "\nThis is the manipulated image data ..."  print newImageValues[:10]  # Output the newImageValues  outIm = Image.new('RGB', im.size)  outIm.putdata(newImageValues)  outputImage = "out.png"  print "Outputting " + outputImage  outIm.save(outputImage) |

**Things we tried:**

To flip the image upside down replace the # Main Image Processing section with:

# Main Image Processing

# \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

imageValues.reverse()

newImageValues = imageValues

# \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

To change the contrast of the image replace the # Main Image Processing section with:

# Main Image Processing

# \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

newImageValues = [] # Initialize new values

for pixel in imageValues:

red = pixel[0]

green = pixel[1]

blue = pixel[2]

# Remember that the pizel order is red green blue

newPixel = (255 - red, 255 - green, 255 - blue)

newImageValues.append(newPixel)

# \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

#### Exercises:

1. Make the output image the same name as the original image, but prefixed with “out\_”.
2. Ask the user what they would like their output image called.
3. Combine the two things we did to the images in class to create an upside down picture with reversed contrast.
4. Change the code so that it will take in a different image (Our code is setup to handle gifs so make sure that’s what you’re giving the code). [BONUS] Make the whole code into a function that takes the image in as an argument, and then ask the user to provide the image name they would like to use. (Hint: Remember you will have to have the image in the same folder as the python program. For a challenge you can try to use the image path so this no longer is a requirement)

### Homework:

1. Make sure you’ve done all the exercises from within class.
2. Write a function that takes in a list of old pixel values and returns a new list with pixels with the opposite contrast ratio (Hint: Use the trick we did in class by taking new\_pixel = 255 - old\_pixel).
3. Write a function that takes in a list of old pixel values and returns a new list of pixel values where each pixel is 100 points more blue than it was before. [BONUS] Have your function also take in a pixel value you’d like to add to every pixel. For example if you would like each pixel to be slightly more orange you’d pass in the pixel (255, 100, 0).
4. Change the image manipulation code so that it prints out an image that’s entirely red.