CIS 1051 Introduction to Programming in Python **Summer II 2014**

Lecture 3: July 14

Lecturer: Anwar Mamat

Disclaimer: *These notes may be distributed outside this class only with the permission of the Instructor.*

3.1 Draw a Calendar with Turtle

Listing 1: Turtle Calendar Example

```
1  import turtle
2  t = turtle.Turtle()
3
4  def line(x1,y1,x2,y2):
5      t.up()
6      t.goto(x1,y1)
7      t.down()
8      t.goto(x2,y2)
9
10 def table():
11     for i in range(7):
12         line(-200,200-i*40,220,200-i*40)
13     for i in range(8):
14         line(-200+i*60,200,-200+i*60,-40)
15
16 def tex():
17     t.up()
18     t.goto(-20,220)
19     t.color('blue')
20     t.write('January 2014', False, "center","Arial 24")
21     t.goto(-170,170)
22     t.color('red')
23     t.write('S', False, "center","Arial 24")
24     t.goto(-110,170)
25     #t.color('red')
26     t.write('M', False, "center","Arial 24")
27     t.goto(-50,170)
28     #t.color('red')
29     t.write('T', False, "center","Arial 24")
30     t.goto(10,170)
31     #t.color('red')
32     t.write('W', False, "center","Arial 24")
33     t.goto(70,170)
34     #t.color('red')
35     t.write('T', False, "center","Arial 24")
36     t.goto(130,170)
```

```
37      #t.color('red')
38      t.write('F', False, "center","Arial 24")
39      t.goto(190,170)
40      #t.color('red')
41      t.write('S', False, "center","Arial 24")
42
43      x = -170
44      y = 130
45      pos = 1
46      t.color('black')
47      for day in range(1,32):
48          t.goto(x,y)
49          t.write(day, False, "center", "Arial 24")
50          x += 60
51          pos += 1
52          if(pos > 7):
53              x = -170
54              y -= 40
55              pos = 1
56  def main():
57      t.speed(10)
58      table()
59      tex()
60      turtle.exitonclick()
61  main()
```

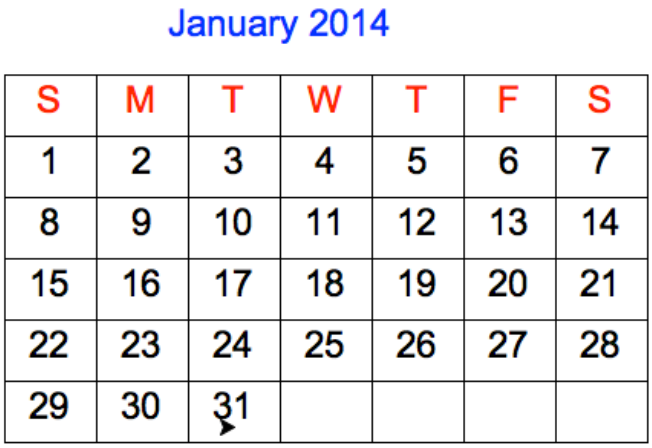

Figure 3.1: Draw a calendar with Turtle

3.2 Calculate PI using Monte Carlo Method

Consider a circle inscribed in a unit square. Given that the circle and the square have a ratio of areas that is $\pi/4$, the value of π can be approximated using a Monte Carlo method.

- Draw a square on the ground, then inscribe a circle within it.
- Uniformly scatter some objects of uniform size (grains of rice or sand) over the square.
- Count the number of objects inside the circle and the total number of objects.
- The ratio of the two counts is an estimate of the ratio of the two areas, which is $\pi/4$. Multiply the result by 4 to estimate π.

Listing 2: Calculate PI by simulation

```
1  import random
2  import math
3  import turtle
4
5  def line(t,x1,y1,x2,y2):
6      t.up()
7      t.goto(x1,y1)
8      t.down()
9      t.goto(x2,y2)
10
11 def showMontePi(numDarts):
12     wn = turtle.Screen()
13     t = turtle.Turtle()
14     t.speed(10)
15     wn.setworldcoordinates(-2,-2,2,2)
16
17     line(t,-1,0,1,0)
18     line(t,0,1,0,-1)
19     circle = 0
20     t.up()
21     for i in range(numDarts):
22         x = random.random()
23         y = random.random()
24
25         d = (math.sqrt(x**2 + y**2))
26         t.goto(x,y)
27         if(d <= 1):
28             circle += 1
29             t.color('red')
30         else:
31             t.color('blue')
32         t.dot()
33     pi = circle / numDarts * 4
34     t.goto(0.2,-0.5)
35     t.write(pi, 'center', font=("Arial", 20, "normal"))
36     turtle.exitonclick()
```

```
37      return pi
38  def main():
39      showMontePi(1000)
40  main()
```

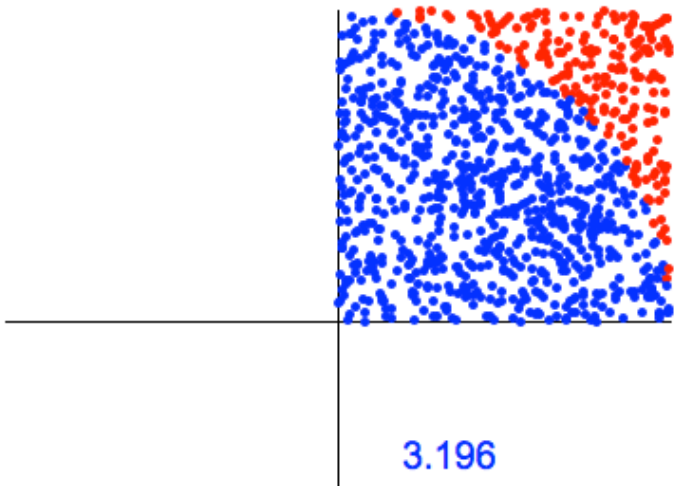

Figure 3.2: Calculate PI using Monte Carlo Method

3.3 Strings

Listing 3: Iterate each character in a string

```
1  str="hello world"
2  for c in str:
3      print(c)
```

3.3.1 Count the frequency of letters

In this example, we count the frequencies of letters [a-z] in the book "War and Peace" by Leo Tolstoy, and draw a bar graph based on the frequency.

Listing 4: Letter Frequency Example

```
1  import turtle
2  def letterFreq(filename):
3      freq = [0 for i in range(128)]
4      instr = open(filename,"r")
5      count = 0
6      for s in instr:
7          s = s.lower()
8          #print(s)
```

```
9           for c in s:
10              freq[ord(c) ] += 1
11          count += 1
12      print(count)
13      #print(freq)
14      instr.close()
15      return freq
16  def drawRect(t,left,top,right,bottom):
17      t.up()
18      t.goto(left,top)
19      t.down()
20      t.pensize(10)
21      t.goto(right,bottom)
22  def drawBar(fr, t):
23      t.up()
24      x = -250
25      y = 0
26      for i in range(97,97+26):
27          y = fr[i] / 1000
28          drawRect(t,x,y,x,0)
29          t.up()
30          t.goto(x,-40)
31          t.write(chr(i), 'left', font=("Arial", 20, "normal"))
32          x += 20
33
34  def main():
35      t = turtle.Turtle()
36      #change the path on your computer
37      file = "/Users/anwar/Desktop/war_peace.txt"
38      fr = letterFreq(file)
39      #for i in range(32,128):
40      #     if(i >=97 and i <= 97+26):
41      #         print(chr(i),end='\t')
42      #         print(fr[i])
43      drawBar(fr,t)
44      turtle.exitonclick()
45  main()
```

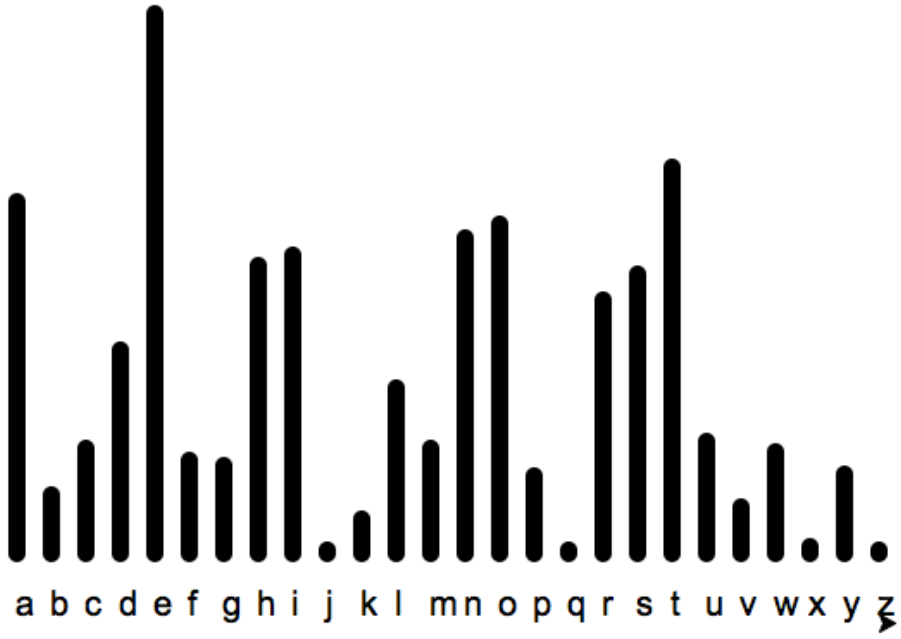

Figure 3.3: Letter frequency

3.3.2 Count the frequency of words

In this example, we count the frequencies of words in the book "War and Peace" by Leo Tolstoy, and display most frequent words.

Listing 5: Word Frequency Example

```
import re
def letterFreq(filename):
    freq=dict()
    #print(freq)
    instr = open(filename,"r")
    count = 0
    for s in instr:
        #print(s)
        s = s.lower()
        #words = s.split()
        words = re.sub(r'[.?/!$\%*,"]',' ',s).split()
        for word in words:
            if(freq.get(word) is None):
                freq[word] = 1
            else:
                freq[word] += 1

        count +=1
    instr.close()
    #print(freq)
```

```
21      return freq
22
23  def main():
24
25      file="/Users/anwar/Desktop/war+peace.txt"
26      fr = letterFreq(file)
27      sorted_list = sorted(fr, key=lambda x:fr[x],reverse=True)
28      for key in range(10): #display most frequent 10 words
29          print(sorted_list[key],end="\t")
30          print(fr.get(sorted_list[key]))
31  main()
```

Listing 6: Word Frequency Output

```
1   the     34488
2   and     22199
3   to      16661
4   of      14861
5   a       10512
6   he      9796
7   in      8884
8   his     7979
9   that    7859
10  was     7342
```

3.3.3 Anagrams

An anagram is a type of word play, the result of rearranging the letters of a word or phrase to produce a new word or phrase, using all the original letters exactly once; for example Doctor Who can be rearranged into Torchwood. This program receives an input from user, and lists all words from a dictionary that can be constructed using the user the letters of the user input. For example:"Lord Voldemort" can be constructed from "Tom Marvolo Riddle".

Listing 7: Word Frequency Output

```
1   def contains(user_input, word):
2       for c in word:
3           pos = user_input.find(c)
4           if(pos == -1):
5               return False
6           else:
7               word = word[:pos] + word[pos+1:]
8       return True
9   def readWord(file, user_input):
10      fin = open(file,"r")
11      for word in fin:
12          word = word.strip()
13          if(len(word) >= 3 and contains(user_input, word)):
14              print(word)
15  def main():
```

```
16      filename = "/Users/anwar/Desktop/wordlist.txt" #dictionary file
17      user_input = input("Enter a word:")
18      readWord(filename, user_input)
19  main()
```

Listing 8: Word Frequency Output

```
1  Enter a word:temple
2  eel
3  elm
4  lee
5  let
6  lpm
7  meet
8  melt
9  met
10 mete
11 pee
12 peel
13 pelt
14 pet
15 tee
16 teem
17 tem
18 temp
19 temple
```