Vanier College

Faculty of Careers and Technical Programs

Department of Computer Science Technology

Advanced UNIX

Lab #3B

Title: Python Tutorial (Part II)

Student Name: Marissa Gonçalves

Submitted to Florin Pilat

September 25, 2020

Please type into your computer the code for each of the following programs listed below and submit your screenshots with the source code and the corresponding outcomes for each program, individually.

1. <suma2.py>

suma2.py - /home/marissa/Documents/Python_Programs/suma2.py (2.7.15)

File Edit Format Run Options Window Help

```
#!/usr/bin/env python
#
# The variable "sys.argv[]" - containing the list of
# command line options passed to a program
# Please type: python suma2.py arg1 arg2, where arg1
# and arg2 are two positive integers, arg1 < arg2
#
import sys

n1 = int(sys.argv[1])
n2 = int(sys.argv[2])
print "The entered numbers are " + str(n1) + " " + str(n2)
def ss(numbers):
    total = 0
    for n in numbers:
        total += n
    return total
m = ss(range(n1,n2))
print "The sum of numbers starting with ", str(n1), " up to ", \
      str(n2), " is ", str(m)
print "The program name is " + sys.argv[0]
print "Number of arguments: ", len(sys.argv), " arguments."
print "Argument List: ", str(sys.argv)
```

```
[marissa@server13 Python_Programs]$ python suma2.py 6 10
The entered numbers are 6 10
The sum of numbers starting with  6  up to  10  is  30
The program name is suma2.py
Number of arguments:  3  arguments.
Argument List:  ['suma2.py', '6', '10']
```

```
[marissa@server13 Python_Programs]$ python suma2.py 3 15
The entered numbers are 3 15
The sum of numbers starting with  3  up to  15  is  102
The program name is suma2.py
Number of arguments:  3  arguments.
Argument List:  ['suma2.py', '3', '15']
```

2. <mySplitText.py>

mySplitText.py - /home/marissa/Documents/Python_Programs/mySplitText.py (2.7.15)

File Edit Format Run Options Window Help

```
#!/usr/bin/env python
"""
Write a program to count the number of words found
in a single sentence and determine the average word
length. Print the sentence you entered.
"""

sentence = raw_input("Please enter a sentence: ")
# Python includes a set of string operations
# called methods such as split() and join()
listOfWords = sentence.split()
print "There are ", len(listOfWords), " words in the sentence."
noTotalChar = 0
for word in listOfWords:
    noTotalChar += len(word)
print "The average word length is ", "%5.3f" % (float(noTotalChar) / len(listOfWords))
print listOfWords
```

```
====== RESTART: /home/marissa/Documents/Python_Programs/mySplitText.py ======
Please enter a sentence: Programming languages like Python are fun.
There are  6  words in the sentence.
The average word length is  6.167
['Programming', 'languages', 'like', 'Python', 'are', 'fun.']
>>>


====== RESTART: /home/marissa/Documents/Python_Programs/mySplitText.py ======
Please enter a sentence: I have typed a sentence.
There are  5  words in the sentence.
The average word length is  4.000
['I', 'have', 'typed', 'a', 'sentence.']
>>>
```

3. <subprocess1.py>

subprocess1.py - /home/marissa/Documents/Python_Programs/subprocess1.py (2.7.15)

File Edit Format Run Options Window Help

```
#!/usr/bin/env python
import subprocess
#
# Note that Python is much more flexible with equal signs.
# There can be spaces around equal signs.
# Run this program as administrator: #python subprocess1.py
#
DEVICES = "lsblk -l"
SPACE = "df -h"

# Places variables into a list/array.
cmds = [DEVICES, SPACE]

# Iterates over list, running statements for each item in the list
# Note that the whitespace is absolutely critical and that a consistent
# indent must be maintained for the code to work properly
count = 0
for cmd in cmds:
    count += 1
    print "Running Command Number %s" % count
    subprocess.call(cmd, shell=True)
```

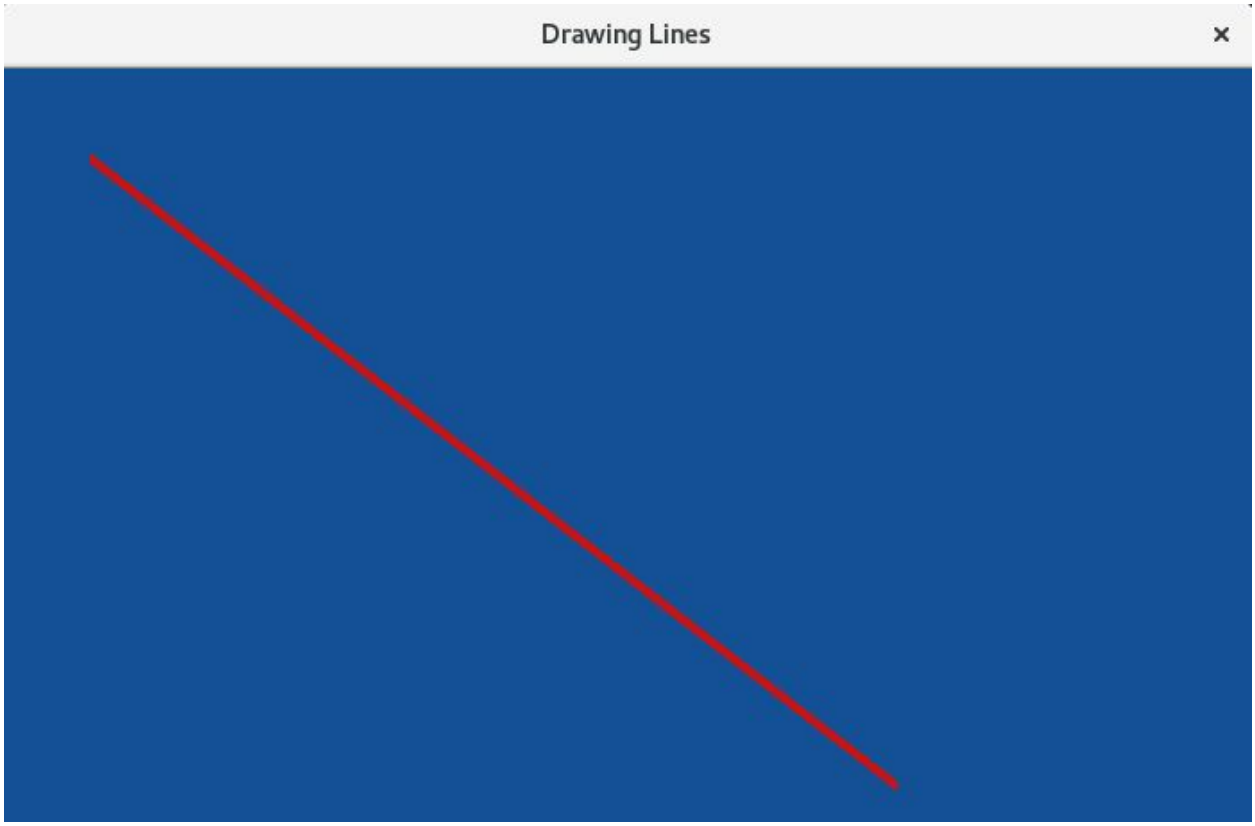
```
[root@server13 Python_Programs]# python subprocess1.py
Running Command Number 1
NAME        MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
sda           8:0    0   16G  0 disk
sda1          8:1    0    1G  0 part /boot
sda2          8:2    0   15G  0 part
sr0          11:0    1 1024M  0 rom
fedora-root 253:0    0 13.4G  0 lvm  /
fedora-swap 253:1    0  1.6G  0 lvm  [SWAP]
Running Command Number 2
Filesystem               Size  Used Avail Use% Mounted on
devtmpfs                 2.5G     0  2.5G   0% /dev
tmpfs                    2.5G     0  2.5G   0% /dev/shm
tmpfs                    2.5G  1.6M  2.5G   1% /run
tmpfs                    2.5G     0  2.5G   0% /sys/fs/cgroup
/dev/mapper/fedora-root   14G  5.3G  7.2G  43% /
/dev/sda1                976M  125M  784M  14% /boot
tmpfs                    2.5G  148K  2.5G   1% /tmp
tmpfs                    496M   28K  496M   1% /run/user/42
tmpfs                    496M  4.6M  491M   1% /run/user/1000
```

4. <line.py>

line.py - /home/marissa/Documents/Python_Programs/line.py (2.7.15)

File Edit Format Run Options Window Help

```
#!/usr/bin/env python
import pygame
import sys
"""
You have to install pygame for having the graphic support
for python. Please run the following line with administrative
rights: yum -y install pygame
"""

from pygame.locals import *
pygame.init()
screen = pygame.display.set_mode((700, 600))
#
pygame.display.set_caption("Drawing Lines")

while True:
    for event in pygame.event.get():
        if event.type in (QUIT, KEYDOWN):
            sys.exit()

    screen.fill((20,80,150))

    #Draw the line.

    color = 200,20,20
    width = 6
    pygame.draw.line(screen, color, (50,50), (500,400), width)

    pygame.display.update()
```


Drawing Lines

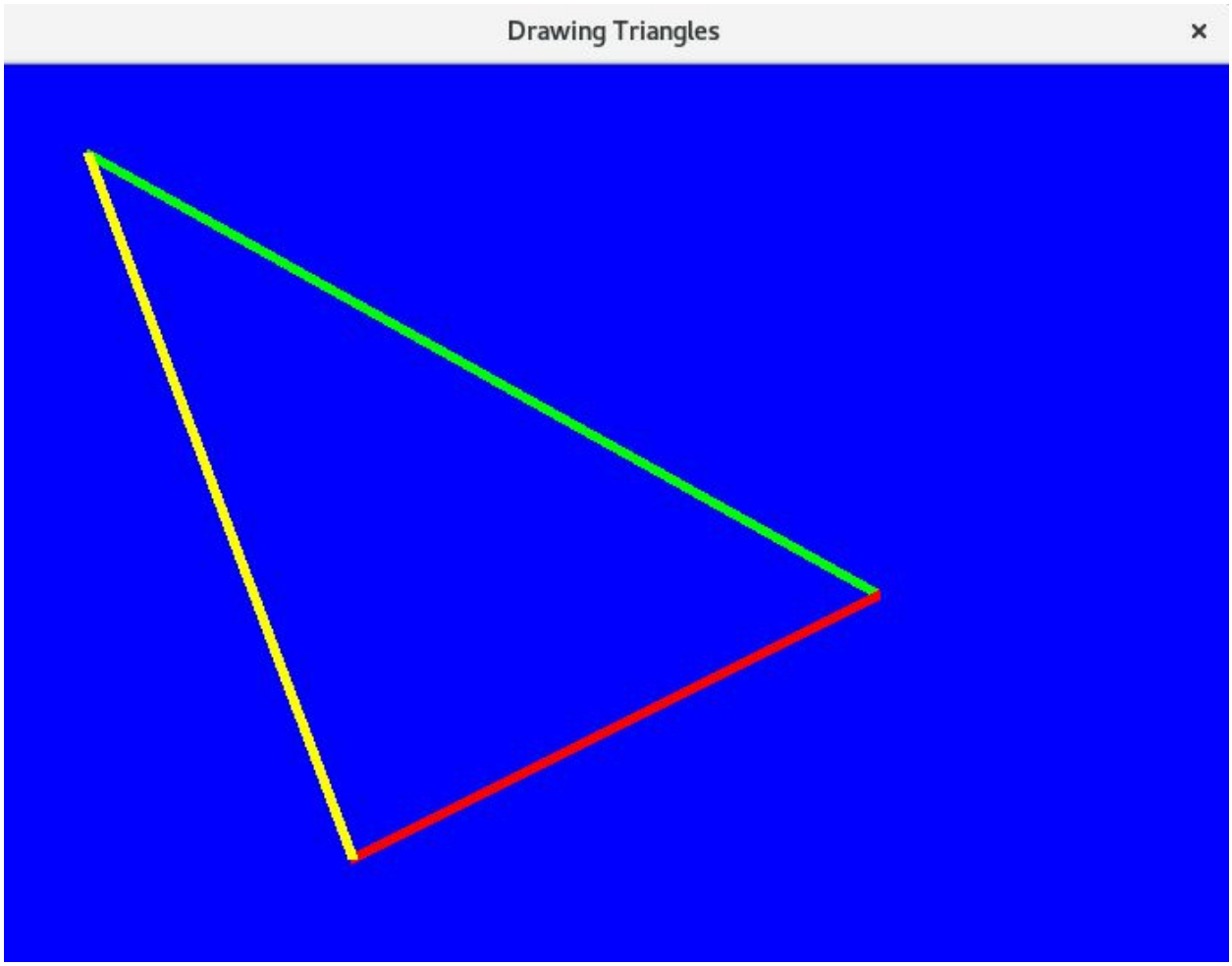
5. <triangle.py>

triangle.py - /home/marissa/Documents/Python_Programs/triangle.py (2.7.15)

File Edit Format Run Options Window Help

```
#!/usr/bin/env python
import pygame
import sys

from pygame.locals import *
pygame.init()
screen = pygame.display.set_mode((700,700))
pygame.display.set_caption("Drawing Triangles")

while True:
    for event in pygame.event.get():
        if event.type in (QUIT, KEYDOWN):
            sys.exit()

    blue = (0,0,255)
    green = (0,255,0)
    red = (255,0,0)
    yellow = (255,255,0)
    width = 6

    screen.fill(blue)

    pygame.draw.line(screen, green, (50,50), (500,300), width)
    pygame.draw.line(screen, red, (500,300), (200,450), width)
    pygame.draw.line(screen, yellow, (200,450), (50,50), width)

    pygame.display.update()
```

```
[marissa@server13 Python_Programs]$ python triangle.py
```


Drawing Triangles

6. <income_tax_report.py>