

ANEXO II - TUTORIALES Y BUENAS PRÁCTICAS - PYTHON**TUTORIAL DEL EJECICIO II - Trinket**

Con el objeto de poder clasificar la basura, identificando los materiales reciclables y no reciclables, tendrán que representar los desechos en la pantalla a través de diferentes formas geométricas y colores.

Contenidos: tipos de variables; manejo de input output; manejo de gráficos e imágenes; sentencias de decisión; bucles; manejo de datos; módulos; manejo de archivos; uso de librerías.

Actividad 1

Crean una pantalla de 400x400 píxeles con fondo blanco, agreguen en tiempo de ejecución formas aleatorias de la librería "turtle" con cantidad, colores ubicaciones aleatorias. Similarmente, agreguen rectángulos de tamaños, colores y ubicaciones aleatorias. Lo mismo con círculos. Cantidad límite de objetos: 25 de cada tipo.

Crean "Arte en tiempo de ejecución"

PC Arte

El objetivo de la carrera es:

- Usar ciclos de control "For"
- Usar funciones.
- Usar la librería "random".
- Usar colores almacenados o en formato RGB.
- Crear arte mediante código Python.

Ciclo de control For:

Este ciclo de control permite que se ejecute repetidamente una porción de código una cierta cantidad de código. A continuación se presenta un código con su correcta sintaxis:

```
j=0
```

```
for i in range(0,10):
```

```
    j=2*i+j
```

La variable "i" inicia con el valor $i=0$, $J=2*0+0=0$. Luego: $i=1$, $J=2*1+0=2$. Se repetirá sucesivamente. La última iteración se da en $i=9$.

Esta forma de incrementar variables se denomina contador.

El módulo “randint” de la librería “random”:

“randint” selecciona un número entero aleatorio dentro de un rango que se ingresa como argumento. El siguiente ejemplo se muestra con su sintaxis:

```
from random import randint  
  
numero=randint(0,100)
```

Luego de la ejecución de esta porción de código, la variable “número” tendrá un valor cualquiera entre 0 y 100.

Desarrollo de la actividad 1:

Tipeen el código que figura abajo. Llámelo “PC_arte”. Modifíquelo los colores, las ubicaciones y los estilos. Si usan Trinket, comenten la línea donde dice “version IDLE”.

El código de PC_arte:

```
# librerias  
  
from turtle import *  
  
from random import randint  
  
from random import choice  
  
# seteos iniciales  
  
limite=25  
  
screen=Screen()  
  
screen.colormode(255) # version IDLE  
  
screen.setup(400,400)  
  
formas=['arrow', 'turtle', 'circle', 'square', 'triangle', 'classic']  
  
# Funciones  
  
def cualquierColor():  
  
rojo=randint(0,255)  
  
verde=randint(0,255)
```



```
azul=randint(0,255)
color(rojo,verde,azul)
return color
def cualquierLugar():
penup()
x=randint(-180,180)
y=randint(-180,180)
goto(x,y)
return
def dibujarRectangulo():
right(randint(0,360))
cualquierColor()
cualquierLugar()
hideturtle()
lenght=randint(10,100)
height=randint(10,100)
begin_fill()
forward(lenght)
right(90)
forward(height)
right(90)
forward(lenght)
right(90)
forward(height)
right(90)
end_fill()
return
# Tortugas
shape('turtle')
```



```
#randomcolor()

stamp()

for i in range (randint(0,limite)):

tortuga=Turtle()

shape(choice(formas))

cualquierColor()

cualquierLugar()

right(randint(0,360))

stamp()

for i in range (randint(0,limite)):

penup()

cualquierColor()

cualquierLugar()

dot(randint(0,50))

for i in range (randint(0,limite)):

penup()

dibujarRectangulo()
```

Salida de Pantalla



Actividad 2

Los camiones de basura tienen que recorrer toda la ciudad levantando los desechos. Para optimizar los tiempos de recolección, deben identificar cada camión con una figura geométrica y el recorrido que realiza cada uno. Deberán cumplir todo el recorrido en el menor tiempo posible.

Presenten en pantalla una carrera entre cuatro objetos de forma aleatoria. Dibujen una grilla con 10 columnas. Gana el primer objeto que toque la llegada. El objetivo de la carrera es:

- Usar la librería “*random*”.
- Realizar animación automática de objetos

Desarrollo de la actividad 2:

Tipeen el siguiente código y ejecútenlo. Llámenlo “carrera de objetos”.

Sugerencia: Usen la librería “*turtle*” para dibujar la grilla, los objetos y su animación. Primero armen la grilla, luego presenten los objetos y finalmente hagan la carrera.

El código `carrera_de_objetos`:

```
from turtle import *  
  
from random import randint  
  
from random import choice
```

```
formas=['arrow', 'turtle', 'circle', 'square', 'triangle', 'classic']

# Armado de la pista

speed(10)

penup()

x0=-100

y0=100

goto(x0,y0)

for step in range(11):

write(step, align='center')

penup()

right(90)

forward(10)

pendown()

forward(150)

penup()

backward(160)

left(90)

forward(20)

penup()

# Inicio de formas

def iniciar(self,kolour,posY):

self.color(kolour)

self.shape(choice(formas))

self.penup()

self.goto(x0,posY)

self.pendown()

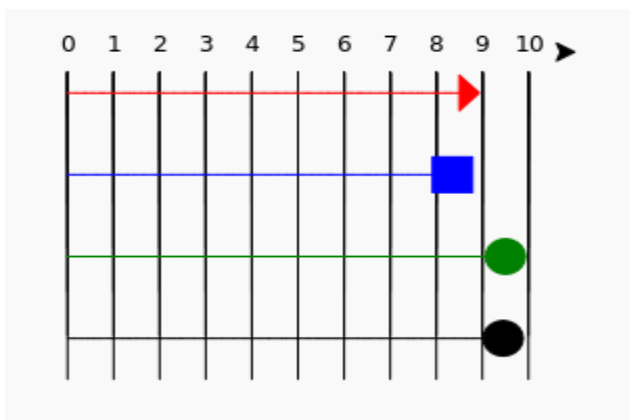
return

roja=Turtle()

iniciar(roja,'red',y0-20)
```



```
azul=Turtle()
iniciar(azul,'blue',y0-60)
verde=Turtle()
iniciar(verde,'green',y0-100)
negra=Turtle()
iniciar(negra,'black',y0-140)
# A correr !!!
for turn in range(60):
roja.forward(randint(1,5))
verde.forward(randint(1,5))
azul.forward(randint(1,5))
negra.forward(randint(1,5))
```

Salida de pantalla (Trinket):**Actividad 3:****Piedra, Papel o Tijera**

Cada camión de recolección de basura, deberá cumplir un circuito de recolección por la ciudad. Para la distribución de dicho circuito, usarán un juego de azar donde el ganador elige el recorrido.

Programen un juego de “Piedra, papel o tijera”, donde se invite al usuario a elegir una de las opciones. El programa elegirá su opción, y abajo se expresará al ganador. Además se debe preguntar si desea seguir jugando o no.

- Usar el módulo “choice” de la librería “random”.

- Introducción al uso de “Listas”.
- Usar estructuras de control: “While”.
- Usar condicional “if”.

La librería “random” y el módulo “choice”:

Random es un generador de aleatoriedad. Para llamar al módulo “choice” de la librería “random”:

```
from random import choice
```

“choice” elige aleatoriamente un elemento dentro de una lista.

```
PC=choice(lista)
```

En este ejemplo, a la variable “PC” se le asigna un elemento cualquiera de la lista de elementos llamada “lista”.

Listas:

Son estructuras de datos que contienen elementos. Los elementos pueden repetirse. La sintaxis es: nombre_de_la_lista=[elemento_1,elemento_2,...,elemento_n]. Ejemplo:

```
lista=['piedra','papel','tijera']
```

La lista llamada “lista” está compuesta por las cadenas: ‘piedra’, ‘papel’ y ‘tijera’.

Ciclo de control While:

Se ejecutará el código indentado a continuación, mientras se cumpla la condición previa. Podemos pensar que “mientras tenga combustible, el auto marcha”. Observar el siguiente pseudo-código:

```
while(combustible > 0):
```

```
    el auto marcha
```

```
    leer nivel de combustible
```

La condición es “mientras tenga combustible” y el código que sigue es “el auto marcha” y “leer nivel de combustible”. Este código se irá repitiendo hasta que aparezca una lectura nula de combustible. En ese caso, al no cumplir la condición, no ejecuta el código (y el auto no marcha).

La sintaxis es como en el ejemplo: while (condición), luego dos puntos, y el código del ciclo estará indentado (un “tab”).

Condicional “if”:

Frente a una condición, esta se cumplirá o no (if). Si se cumple, se ejecutará el código a continuación. El siguiente ejemplo se muestra con la sintaxis correcta:

```
if (a==2):  
    print 'a vale 2'
```

Notar que en la comparación “a==2” se ponen dos signos “=” y no uno.

Estructura Condicional “if” - “elif” - “else”:

Es una estructura condicional de salida múltiple. Se encabeza con un condicional “if”: si la condición se cumple, se ejecuta el código que sigue. Pero si no se da la primera condición puede darse una segunda condición (elif) y así sucesivamente. Esta es una estrategia de programación más compleja que el condicional que permite la selección de múltiples opciones. También puede ejecutarse código si no se da ninguna de las condiciones anteriores (else). El siguiente ejemplo muestra todos los casos, con sintaxis correcta:

```
if(puesto==1):  
    print 'Ganó medalla de oro'  
  
elif(puesto==2):  
    print 'Ganó medalla de plata'  
  
elif(puesto==3):  
    print 'Ganó medalla de bronce'  
  
else:  
    print 'No ganó ninguna medalla'
```

Notar que en las comparaciones se usa doble signo igual.

Desarrollo del enunciado actividad 3:

Tipeen el siguiente código y ejecútenlo. Llámenlo piedra_papel_o_tijera.

Sugerencia: Usar un While para seguir jugando o no (iniciar con “seguir” = “si”), y dentro del mismo codificar el resto del juego.

Código piedra_papel_o_tijera:

```
from random import choice
```

```

lista=['piedra','papel','tijera']

denuevo='s' # es la variable que indica seguir jugando o no

while(denuevo=='s'):

jugador="" # es la elección del jugador.

while(jugador==""):

entrada=input('¿piedra, papel o tijera?')

# el siguiente condicional es para asegurar que el jugador tipee una de las
# opciones en forma correcta

if(entrada=='piedra' or entrada=='papel' or entrada=='tijera'):

jugador=entrada

else:

pass # pass no hace nada

print 'Elegiste: '+str(jugador)

PC=choice(lista)

print 'PC eligió: '+str(PC)

# el siguiente condicional presenta las reglas del juego

if((jugador=='papel' and PC=='piedra')or(jugador=='piedra' and PC=='tijera')or(jugador=='tijera' and
PC=='papel')):

res='GANASTE!!!'

elif((jugador=='papel' and PC=='tijera')or(jugador=='piedra' and PC=='papel')or(jugador=='tijera' and
PC=='piedra')):

res='PC GANÓ!!!'

else:

res='EMPATE'

print res

denuevo=input('¿jugar de nuevo? s/n: ')

```

Salida de pantalla:

```

Powered by trinket
¿piedra, papel o tijera? papel
Elegiste: papel
PC eligió: tijera
PC GANÓ!!!
¿jugar de nuevo? s/n: s
¿piedra, papel o tijera? piedra
Elegiste: piedra
PC eligió: papel
PC GANÓ!!!
¿jugar de nuevo? s/n: n
    
```

Guardá el trabajo.

Actividad 4:

Reciclabilidad de residuos

La planta de reciclado de desechos está buscando incorporar personal. Para ello, se seleccionará personal que pueda aprobar un test de clasificación de residuos. Ustedes deberán colaborar en armar el mismo.

En un archivo de texto se presenta un listado de residuos, y si su reciclabilidad: son reciclables (r) o bien basura (b). El residuo de su reciclabilidad se encuentra separado por “;”. Plantear un desafío al usuario acerca de cuánto sabe de la reciclabilidad o no de ciertos residuos. Al finalizar indicar su puntuación (de 0 a 10). El objetivo de la carrera es:

- Manejo de Listas.
- Manejo de archivos de datos.

Manejo de listas:

Antes de llenarse, una lista debe crearse, vacía. Luego se la puede cargar con elementos. El método para agregar elementos es “append”: entre paréntesis se pone lo que se agregará a la lista. Para eliminar un elemento de la lista, el método correspondiente es “remove”: entre paréntesis se especifica el elemento o bien entre corchetes el índice del mismo.

A continuación se presenta un ejemplo de creación, carga y eliminación de elementos en una lista.

```

lista = []                # Creación de la lista
lista.append('elemento_1') # Agrego elemento_1
lista.append('elemento_2') # Agrego elemento_2
lista.remove('elemento_1') # Elimino elemento_1
    
```

La lista tiene en este momento solo un elemento: elemento_2.

Manejo de archivos de datos:

Datos pueden almacenarse en archivos de texto para su uso en tiempo de ejecución. Para extraerle información, deben abrirse y leerse línea a línea. Cada línea se captura en una variable. A continuación se presenta un ejemplo de código que abre un archivo de texto y utiliza sus datos.

```
archivo = open('misdatos.txt', 'r')           # "r": se abre para Solo Lectura
for linea in archivo.readlines():             # lee el archivo por línea
print linea                                   # imprime cada línea en pantalla
```

Desarrollo del enunciado actividad 4

Tipeen el código presentado a continuación y ejecútenlo. Llámelo "reciclabilidad_de_residuos.py". La cantidad predefinida de preguntas es 4, pero podés cambiarla vos.

Sugerencia: Lean el archivo de la respuestas por línea: "residuos.txt". Cada línea está separada por "enter" y cada campo por ";" y presenten la primer parte en la pregunta. Capturen la respuesta del usuario y compárenla con la respuesta válida (la segunda parte de la línea) e incrementen o no la puntuación.

El archivo "residuos.txt":

```
radiografia;r
tapita plástica;r
envase plástico;r
papel;r
caja de cartón;r
envase de tetrabrick;r
botella de vidrio;r
frasco de vidrio;r
lata;r
envase de acero;r
papel sucio;b
papel metalizado;b
```



envase plástico con solvente;b

CD;b

pañal;b

lámpara de bajo consumo;b

lámpara halógena;b

espejo;b

cáscara de banana;b

restos de comida;b

Código de reciclabilidad_de_residuos:

```
from random import choice
```

```
residuos=[]
```

```
# se carga la lista con el contenido del archivo "residuos.txt" y se lo pasa a una coleccion
```

```
archivo=open('residuos.txt','r') # la 'r' indica que se abre el archivo para lectura
```

```
residuos=archivo.read().splitlines()
```

```
for i in range(len(residuos)):
```

```
linea=residuos[i].split(';')
```

```
print '¿Cuánto sabés acerca del reciclado de basura?'
```

```
cant_preguntas=4
```

```
cuestionario=[]
```

```
# Eleccion de residuos:
```

```
respuestas_correctas=0
```

```
for i in range(cant_preguntas):
```

```
pregunta=choice(residuos)
```

```
residuos.remove(pregunta)
```

```
residuo=str(pregunta).split(';')
```

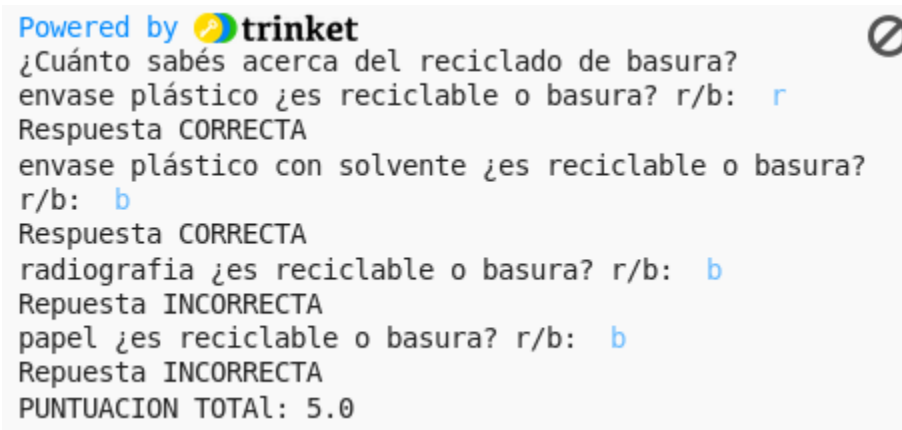
```
text=""
```

```
while(text==""):
```

```
res=input(str(residuo[0])+ ' ¿es reciclable o basura? r/b: ')
```



```
if (res=='r' or res=='b'):
text='text'
if(res==str(residuo[1])):
respuestas_correctas=respuestas_correctas+1
print 'Respuesta CORRECTA'
else:
print 'Repuesta INCORRECTA'
print 'PUNTUACION TOTAL: '+str(float(float(respuestas_correctas)/float(cant_preguntas))*10)
```

Salida de pantalla:

```
Powered by trinket
¿Cuánto sabés acerca del reciclado de basura?
envase plástico ¿es reciclable o basura? r/b: r
Respuesta CORRECTA
envase plástico con solvente ¿es reciclable o basura?
r/b: b
Respuesta CORRECTA
radiografia ¿es reciclable o basura? r/b: b
Repuesta INCORRECTA
papel ¿es reciclable o basura? r/b: b
Repuesta INCORRECTA
PUNTUACION TOTAL: 5.0
```

Guarden el trabajo.

Actividad 5

Por la planta de reciclado de basura pasan diariamente gran cantidad de camiones. Deben mejorar la circulación a través de un sistema de semáforos.

Dada la imagen del semáforo apagado, presenten un programa que encienda la luz si se hace click dentro de la luz apagada. Una vez logrado, reproduzcan un sonido. El objetivo de la carrera es:

- Usar un IDE local
- Usar la librería PIL para tratamiento de imágenes
- Usar la librería OS
- Usar la librería Pygame

- Reproducir sonidos

Uso de la librería OS:

Puede verificarse si la librería está instalada o no. Para ello tipeen: 'import os'. Si no da mensaje de error, la librería se encuentra instalada. Para instalar la librería os usar "pip":

```
pip install os
```

El módulo OS de Python provee funcionalidades para interactuar directamente con el sistema operativo del equipo. Algunas de ellas son:

```
os.getcwd()          # Devuelve el directorio actual de trabajo
os.uname()           # Devuelve el sistema operativo.
os.listdir(path)     # Lista los directorios en ese path
os.mkdir(path)       # Crea un directorio en ese path
os.remove(path)      # borra el archivo del path
os.rmdir(path)       # borra un directorio
```

PIL es la librería de manejo de imágenes de Python: Python Image Library (desde hace unos días se la llama "Pillow"). Un módulo muy útil para el tratamiento de imágenes es Image. Se obtiene:

Uso de la librería PIL:

PIL es la librería de manejo de imágenes de Python: Python Image Library. Un módulo muy útil para el tratamiento de imágenes es Image. Se obtiene:

```
from PIL import Image
```

Resumen del uso de Image:

Abrir el archivo imagen:

```
imagen=Image.open(nombre_de_archivo)    # abre el archivo imagen
```

Información de la Imagen:

```
formato=imagen.format                   # formato: PNG, TIFF, JPG, etc.
```

```
ancho=imagen.size[0]                   # ancho
```

```
alto=imagen.size[1] # alto
```

Tratamiento de píxeles:

```
rojo= imagen.getpixel((x,y))[0] # se obtiene valor de rojo del píxel (x,y)
verde= imagen.getpixel((x,y))[1] # se obtiene valor de verde del píxel (x,y)
azul= imagen.getpixel((x,y))[2] # se obtiene valor de azul del píxel (x,y)
imagen.putpixel((x,y),(0,0,0)) # se escribe un color RGB en un píxel (x,y)
```

Uso de la librería Pygame:

```
pygame.init() # Iniciar Pygame
screen = pygame.display.set_mode(x,y) # Setear tamaño de la ventana
pygame.display.set_caption(titulo) # Poner título a la ventana
```

Mostrar una imagen de fondo de la ventana:

```
img=pygame.image.load('nombre_de_archivo')
screen.blit(img,(0,0))
pygame.display.flip()
```

Obtención de coordenadas del click del mouse:

```
if event.type == pygame.MOUSEBUTTONDOWN:
pos = pygame.mouse.get_pos() # pos: coordenadas (x,y)
```

Reproducir sonido:

```
pygame.mixer.music.load('archivo_sonido.wav')
pygame.mixer.music.play()
```

Desarrollo del enunciado actividad 5

Abran el IDLE. Copien y ejecuten el código que se presenta a continuación. Llámelo “semaforo.py”

Sugerencia: Para pintar las luces del semáforo, tener en cuenta el uso de colores plenos: Rojo=(255,0,0), Amarillo=(255,255,0), Verde=(0,255,0). Para apagar una luz, pintarla de color gris=(125,125,125)

Código de 'semaforo.py':

```
import os

from PIL import Image

import pygame

import time

# Inicializacion

path=os.getcwd()

nombre='semaforo_G.png'

archivo=str(path)+'/imagenes/'+str(nombre)

im=Image.open(archivo)

ancho=im.size[0]

alto=im.size[1]

punto1Rojo=(10,25)

punto2Rojo=(120,125)

punto1Am=(15,130)

punto2Am=(105,210)

punto1Verde=(20,210)

punto2Verde=(100,290)

rojo=(255, 0, 0)

am=(255, 255, 0)

verde=(0, 255, 0)

gris=(125,125,125)

# Metodos

def encender_rojo(punto1Rojo,punto2Rojo):

    for i in range(punto1Rojo[0],punto2Rojo[0]):

        for j in range(punto1Rojo[1],punto2Rojo[1]):

            punto=(i,j)
```



```

        if((im.getpixel(punto)[0]>=125) and (im.getpixel(punto)[1]>=125) and
(im.getpixel(punto)[2]>=125)):
            im.putpixel(punto,rojo)
    for i in range(punto1Am[0],punto2Am[0]):
        for j in range(punto1Am[1],punto2Am[1]):
            punto=(i,j)
            if((im.getpixel(punto)[0]>=125) or (im.getpixel(punto)[1]>=125) or
(im.getpixel(punto)[2]>=125)):
                im.putpixel(punto,gris)
    for i in range(punto1Verde[0],punto2Verde[0]):
        for j in range(punto1Verde[1],punto2Verde[1]):
            punto=(i,j)
            if((im.getpixel(punto)[0]>=125) or (im.getpixel(punto)[1]>=125) or
(im.getpixel(punto)[2]>=125)):
                im.putpixel(punto,gris)
    im.save('imagenes/semaforo_R.png')
    return
def encender_amarillo(punto1Am,punto2Am):
    for i in range(punto1Am[0],punto2Am[0]):
        for j in range(punto1Am[1],punto2Am[1]):
            punto=(i,j)
            if((im.getpixel(punto)[0]>=125) and (im.getpixel(punto)[1]>=125) and
(im.getpixel(punto)[2]>=125)):
                im.putpixel(punto,am)
    for i in range(punto1Rojo[0],punto2Rojo[0]):
        for j in range(punto1Rojo[1],punto2Rojo[1]):
            punto=(i,j)
            if((im.getpixel(punto)[0]>=125) or (im.getpixel(punto)[1]>=125) or
(im.getpixel(punto)[2]>=125)):
                im.putpixel(punto,gris)
    for i in range(punto1Verde[0],punto2Verde[0]):
        for j in range(punto1Verde[1],punto2Verde[1]):

```

```

    punto=(i,j)

    if((im.getpixel(punto)[0]>=125) or (im.getpixel(punto)[1]>=125) or
(im.getpixel(punto)[2]>=125)):

        im.putpixel(punto,gris)

    im.save('imagenes/semaforo_A.png')

    return

def encender_verde(punto1Verde,punto2Verde):

    for i in range(punto1Verde[0],punto2Verde[0]):

        for j in range(punto1Verde[1],punto2Verde[1]):

            punto=(i,j)

            if((im.getpixel(punto)[0]>=125) and (im.getpixel(punto)[1]>=125) and
(im.getpixel(punto)[2]>=125)):

                im.putpixel(punto,verde)

            for i in range(punto1Am[0],punto2Am[0]):

                for j in range(punto1Am[1],punto2Am[1]):

                    punto=(i,j)

                    if((im.getpixel(punto)[0]>=125) or (im.getpixel(punto)[1]>=125) or
(im.getpixel(punto)[2]>=125)):

                        im.putpixel(punto,gris)

            for i in range(punto1Rojo[0],punto2Rojo[0]):

                for j in range(punto1Rojo[1],punto2Rojo[1]):

                    punto=(i,j)

                    if((im.getpixel(punto)[0]>=125) or (im.getpixel(punto)[1]>=125) or
(im.getpixel(punto)[2]>=125)):

                        im.putpixel(punto,gris)

    im.save('imagenes/semaforo_V.png')

    return

def mostrar_imagen(archivo):

    img=pygame.image.load(archivo)

    screen.blit(img,(0,0))

    pygame.display.flip()

```

```

return

def sonido():

    pygame.mixer.music.load('sonidos/b24.wav')

    pygame.mixer.music.play()

    return

pygame.init()

tamano = [ancho, alto]

screen = pygame.display.set_mode(tamano)

pygame.display.set_caption('controlando el Semaforo')

clock = pygame.time.Clock()

mostrar_imagen(archivo)

ln=1

c = pygame.time.Clock()

while True:

    c.tick(3)

    ln += 1

    for event in pygame.event.get(): # Usuario hace algo

        if event.type == pygame.QUIT:

            done = True

            quit()

        elif event.type == pygame.MOUSEBUTTONDOWN:

            pos = pygame.mouse.get_pos()

            x = pos[0]

            y = pos[1]

            if(((x>punto1Rojo[0]) and (x<punto2Rojo[0]))and((y>punto1Rojo[1]) and (y<punto2Rojo[1]]))):

                encender_rojo(punto1Rojo,punto2Rojo)

                archivo='imagenes/semaforo_R.png'

                mostrar_imagen(archivo)

                sonido()

```

```
if(((x>punto1Am[0]) and (x<punto2Am[0]))and((y>punto1Am[1]) and (y<punto2Am[1]]))):
```

```
    encender_amarillo(punto1Am,punto2Am)
```

```
    archivo='imagenes/semaforo_A.png'
```

```
    mostrar_imagen(archivo)
```

```
    sonido()
```

```
if(((x>punto1Verde[0]) and (x<punto2Verde[0]))and((y>punto1Verde[1]) and (y<punto2Verde[1]]))):
```

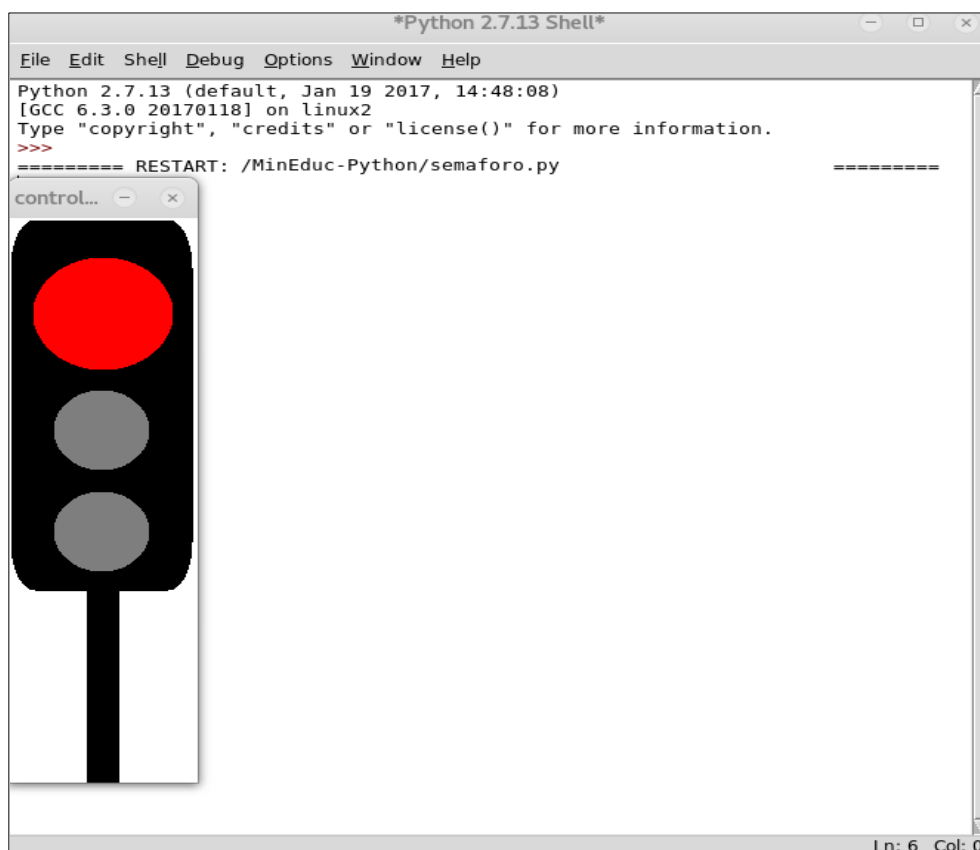
```
    encender_verde(punto1Verde,punto2Verde)
```

```
    archivo='imagenes/semaforo_V.png'
```

```
    mostrar_imagen(archivo)
```

```
    sonido()
```

```
mostrar_imagen(archivo)
```



Guarden el trabajo.