

Guía para la utilización de la librería introcs

Módulo	Descripción
stdio.py	Funciones para leer y escribir números y texto
stddraw.py	Funciones que permiten al usuario crear un dibujo. El dibujo aparece en el “canvas” en una ventana emergente. El módulo también importa los colores definidos en el módulo color.
stdaudio.py	Funciones que permiten reproducir sonido.
stdrandom.py	Funciones que permiten obtener números aleatorios que pertenezcan o no a una distribución (Por ejemplo: Un número dentro de una función gaussiana).
stdarray.py	Funciones para crear, leer y escribir arreglos de 1 o 2 dimensiones.
stdstats.py	Funciones relacionadas al análisis estadístico y la visualización gráfica de los datos.
color.py	Define la clase y objetos Color. Entrega el tipo de dato para generar colores.
picture.py	Define la clase Picture. Entrega el tipo de dato para procesar imágenes digitales.
instream.py	Define la clase InStream. Entrega el tipo de dato para leer números y texto desde archivos y URL's.
outstream.py	Define la clase OutStream. Entrega el tipo de dato para escribir números y texto en archivos.

Fuente: *Introduction to Programming in Python*
(<https://introcs.cs.princeton.edu/python/code/>)

Ejemplos

1. Dibujar figuras en una ventana emergente.

- Círculo

```
import stddraw #importar el módulo

stddraw.setXscale(-1.0, 1.0) #Escala de la ventana emergente
stddraw.setYscale(-1.0, 1.0) #Escala de la ventana emergente
stddraw.setPenColor(stddraw.PINK) #Color del dibujo
stddraw.filledCircle(0,0,.1) #dibuja un círculo con centro en (0,0) y radio 0.1

stddraw.show() #muestra el dibujo en la ventana emergente
```

- Rectángulo

```
import stddraw #importar el módulo

stdraw.setXscale(-1.0, 1.0)#Escala de la ventana
emergente
stdraw.setYscale(-1.0, 1.0)#Escala de la ventana
emergente
stdraw.setPenColor(stdraw.PINK)#Color del dibujo
stdraw.filledRectangle(0,0,.1,.2) #dibuja un rectángulo
con esquina inferior izquierda en (0,0), ancho 0.1 y alto
0.2

stdraw.show()#muestra el dibujo en la ventana emergente
```

- Cuadrado

```
import stddraw #importar el módulo

stdraw.setXscale(-1.0, 1.0)#Escala de la ventana
emergente
stdraw.setYscale(-1.0, 1.0)#Escala de la ventana
emergente
stdraw.setPenColor(stdraw.PINK)#Color del dibujo
stdraw.filledSquare (0,0,.2) #dibuja un cuadrado con
centro en (0,0) y lado 2*0.2
stdraw.show()#muestra el dibujo en la ventana emergente
```

- Triángulo

```
import stddraw #importar el módulo

stdraw.setXscale(-1.0, 1.0)#Escala de la ventana
emergente
stdraw.setYscale(-1.0, 1.0)#Escala de la ventana
emergente
stdraw.setPenColor(stdraw.PINK)#Color del dibujo
stdraw.filledPolygon([-0.2,.5,.6],[0,-.1,-.5])#dibuja un
triángulo con las coordenadas en x(-0.2,0.5,0.6) y
coordenadas en y(0,-0.1,-0.5)

stdraw.show()#muestra el dibujo en la ventana emergente
```

Nota: Este código sirve para crear polígonos con las coordenadas x e y que ingrese el usuario.

- Línea

```
import stddraw #importar el módulo

stdraw.setXscale(-1.0, 1.0)#Escala de la ventana emergente
stdraw.setYscale(-1.0, 1.0)#Escala de la ventana emergente
stdraw.setPenColor(stddraw.PINK)#Color del dibujo
stdraw.line(-.2,.5,0,-.1)#dibuja una línea desde
(x1,y1)=(-0.2,0.5) hasta (x2,y2)=(0,-0.1)

stdraw.show()#muestra el dibujo en la ventana emergente
```

2. Escribir texto en la ventana emergente

```
import stddraw #importar el módulo

stdraw.setXscale(-1.0, 1.0)#Escala de la ventana emergente
stdraw.setYscale(-1.0, 1.0)#Escala de la ventana emergente
stdraw.setFontFamily('Comic Sans')#Fuente del texto
stdraw.setFontSize(40)#Tamaño del texto

stdraw.setPenColor(stddraw.BLUE)#Color del texto

stdraw.text(0,0,"Hola mundo") #Se escribe el texto "Hola mundo" en las
coordenadas (x,y) = (0,0)

stdraw.show()#mostrar el texto
```

3. Mostrar una imagen

```
import stddraw #importar el módulo
from picture import Picture #importar la clase Picture desde el módulo
picture

pic = Picture('C:/Users/Francisca/AppData/Local/Programs/Python/Python37-
32/mario.jpg') #copiar la dirección de la imagen, en este ejemplo es una
imagen de mario

stdraw.setCanvasSize(pic.width(), pic.height())#escala de la ventana
stdraw.picture(pic)#dibujar la imagen
stdraw.show()#mostrar la imagen
```

Nota: Este código no funciona en IDLE, se debe ejecutar desde el terminal digitando: python C:\Users\Francisca\AppData\Local\Programs\Python\Python37-32\image.py (siendo esta última la dirección del archivo que contiene el código)

4. Reproducir sonido como archivo

```
import stdaudio #importar el módulo

stdaudio.playFile('C:/Users/Francisca/Downloads/cartoon001') #abrir el
archivo, en este ejemplo es un archivo en formato wav (cartoon001.wav) que
reproduce el sonido de una caricatura.
```

Fuente sonidos: <http://www.grsites.com/archive/sounds/category/23/?offset=0>

5. Utilizar el teclado para mover un objeto en la pantalla

```
import stddraw #importar el módulo

RADIUS = .1 #definir radio del círculo
stddraw.setXscale(-1.0, 1.0) #definir escala
stddraw.setYscale(-1.0, 1.0) #definir escala
stddraw.setPenColor(stddraw.YELLOW) #definir color del dibujo
rx = 0 #coordenada en x
ry = 0 #coordenada en y

while True: #se genera un loop infinito, donde cada vez que se presione una tecla se
realiza una acción
    if stddraw.hasNextKeyTyped(): #condición: se presionó una tecla?
        k=stddraw.nextKeyTyped() #cuál es la tecla presionada?
        if k==stddraw.K_UP: #si la tecla es la flecha hacia arriba:
            if ry<.8: #si la coordenada en y es menor a 0.8
                ry+=.1 #aumentar en 0.1 la coordenada en y
            else: #si no es menor a 0.8
                ry=ry #no cambia (esto se hace para que no traspase los bordes)
        elif k==stddraw.K_DOWN: #si la tecla es la flecha hacia abajo:
            if ry>-.8:
                ry-=.1
            else:
                ry=ry
        elif k==stddraw.K_RIGHT: #si la tecla es la flecha hacia la derecha:
            if rx<.8:
                rx+=.1
            else:
                rx=rx
        elif k==stddraw.K_LEFT: #si la tecla es la flecha hacia la izquierda:
            if rx>-.8:
                rx-=.1
            else:
                rx=rx
    stddraw.clear(stddraw.BLACK) #fondo negro
    stddraw.filledCircle(rx, ry, RADIUS) #círculo con centro rx,ry y radio =RADIUS
    stddraw.show(0) #mostrar el dibujo
```

Los comandos de teclado los pueden encontrar en:

<https://github.com/josiest/pygtails/blob/master/docs/pygstants.rst>