**Guía de proyectos y uso de stddraw**

La presente guía es para complementar la solución y uso de librerías de los proyectos PY2 y PY3.

1. Cuando en un problema le planteen ingresar la probabilidad, o que se genere información considerando cierto valor de probabilidad.
   1. Considere usar una variable que lea esa probabilidad en formato flotante tal como p = float(input())

Usted debe generar un valor r = random.random() y comparar ese valor r con p.

En la relación r <= p

Lo que significará que los datos generados cumplen con la probabilidad solicitada. Solo si r es <= p usted hará lo que le piden en el problema.

1. Cuando usted crea una matriz de listas tal como:
   1. Lista = [[1, 2, 3], [1, 2, 3], [2, 3, 5]]

Usted puede acceder a los valores de filas y columnas de la siguiente manera:

Lista[i][j] donde i corresponde a los elementos de la lista mayor y j a los elementos de la sub-lista.

1. Si le solicitan generar imágenes en posiciones aleatorias debe utilizar la función random.random() de la biblioteca import random. Esta genera un valor aleatorio entre 0 y 1.
2. La biblioteca stddraw.py tiene un área de dibujo por default desde el punto (0,0) a (1,1).
3. Usted puede cambiar la escala de dibujo con las funciones stddraw.setXSscale(min, max), stddraw.setYSscale(min, max). Así por ejemplo puede configurar un área de x de 0, 100 y de y de 0, 100. Dibujando figuras en las coordenadas de (0,0) a (100,100).
4. Para utilizar los colores en sus programas incluya la biblioteca import color. Los colores se encuentran como constantes: ejemplo color.BLACK, color.YELLOW
5. Para dibujar con un color debe primero setear el color de esta manera: stddraw.setPenColor(color.BLUE)

Para volver al color por default utilice stddraw.setPenColor(color.WHITE)

1. La función stddraw.square o filledSquare tiene como parámetros (x, y, r) donde r es el radio del cuadrado.
2. Para dibujar círculos use stddraw.circle(x, y, r)
3. Para dibujar otras figuras poligonales use Polygon(x,y) o filledPolygon(x,y) donde x, y son listas de coordenadas de cada punto de la figura. Ejemplo: x = [10, 20, 30], y = [20, 40, 60]
4. Para dibujar funciones utilice un esquema de líneas entre puntos de la gráfica. Es decir dibuje líneas entre puntos generados en un plano cartesiano. Si es una ecuación linear es una línea desde la posición (x,y) a (x1,y1).
5. Para una ecuación con curvas, genere los valores en listas y luego dibuje líneas entre estos puntos. Observe este ejemplo:

**import** stddraw **as** d

x = []  
y = []  
  
**for** i **in** range(-10, 11):  
 x.append(i)  
 y.append(i\*i)  
  
  
d.setXscale(-10, 10)  
d.setYscale(-20, 100)  
  
**for** i **in** range(1,21):  
 d.line(x[i-1], y[i-1], x[i], y[i])

1. Para dibujar texto encima de figuras utilice el siguiente código como referencia:

**import** stddraw **as** d  
**import** color **as** c  
  
*# print cuadrados en filas con texto*d.setXscale(0, 150)  
d.setYscale(0, 150)  
  
d.filledSquare(50, 90, 5)  
d.setPenColor(c.WHITE)  
d.text(50, 90, **'1'**)  
d.setPenColor(c.BLACK)  
  
d.filledSquare(40, 75, 5)  
d.setPenColor(c.WHITE)  
d.text(50, 90, **'1'**)  
  
d.setPenColor(c.BLACK)  
d.filledSquare(60, 75, 5)  
d.setPenColor(c.WHITE)  
d.text(50, 90, **'1'**)  
  
d.show()