

# Práctica 10. Gráficos con Python: Pygame

---

```
#!/usr/bin/python

# Ejercicio 01:
# 1.- Dibuje 20 círculos aleatorios.
# 2.- Dibuje un solo círculo en mitad de la pantalla con un radio igual a la
# mitad del tamaño
# de la ventana.
#
# Victor Manuel Andreu Felipe 2025

import pygame
import sys
import random

# dimensiones de la ventana
window_x = 640
window_y = 480

pygame.init()

# preparar la ventana
window = pygame.display.set_mode((window_x, window_y))
window.fill((255, 255, 255))

# dibujar 20 círculos aleatorios
for _ in range(20):
    x = random.randint(0, window_x)
    y = random.randint(0, window_y)
    radio = random.randint(10, 50)
    color = (random.randint(0, 255), random.randint(0, 255), random.randint(0,
255))
    pygame.draw.circle(window, color, (x, y), radio)

center_radio = min(window_x, window_y) // 2

# dibujar un círculo negro en el centro con borde 2 pixeles(, 2) y sin relleno
pygame.draw.circle(window, [ 0, 0, 0 ], [ window_x / 2, window_y / 2 ],
center_radio, 2)

pygame.display.flip()

while True:
    for event in pygame.event.get():
        if event.type == pygame.QUIT:
            pygame.quit()
            sys.exit()
```

```
#!/usr/bin/python

# Ejercicio 02:
# Modifica el programa para que dibuje rectángulos aleatorios:
# - De un mismo color.
# - De colores aleatorios.
#
# Victor Manuel Andreu Felipe 2025

import pygame
import sys
import random

# dimensiones de la ventana
window_x = 640
window_y = 480

pygame.init()

# preparar la ventana
window = pygame.display.set_mode((window_x, window_y))
window.fill((255, 255, 255))

# dibujar rectángulos aleatorios de un mismo color (verde)
for _ in range(10):
    x = random.randint(0, window_x)
    y = random.randint(0, window_y)
    base = random.randint(20, 100)
    altura = random.randint(20, 100)
    color = (0, 255, 0)
    pygame.draw.rect(window, color, (x, y, base, altura))

# dibujar rectángulos aleatorios de colores aleatorios
for _ in range(10):
    x = random.randint(0, window_x)
    y = random.randint(0, window_y)
    base = random.randint(20, 100)
    altura = random.randint(20, 100)
    color = (random.randint(0, 255), random.randint(0, 255), random.randint(0, 255))
    pygame.draw.rect(window, color, (x, y, base, altura))

pygame.display.flip()

while True:
    for event in pygame.event.get():
        if event.type == pygame.QUIT:
            pygame.quit()
            sys.exit()
```

```

#!/usr/bin/python

# Ejercicio 03:
# Cambia el icono de la pantalla por una foto de tu cara, tomada con el móvil o
# con una webcam.
# Haz que tu cara vaya moviéndose en pantalla de izquierda a derecha, cuando
# llegue a uno de los
# extremos salga rebotada hacia el extremo contrario.
#
# Victor Manuel Andreu Felipe 2025

import pygame
import sys
import random

# dimensiones de la ventana
window_x = 640
window_y = 480

pygame.init()

# preparar la ventana
window = pygame.display.set_mode((window_x, window_y))
window.fill((255, 255, 255))

# cargar la imagen
cara = pygame.image.load("cara.jpg")
cara_rect = cara.get_rect() # crea un marco

# posición inicial y velocidad
x = random.randint(0, window_x - cara_rect.width)
y = random.randint(0, window_y - cara_rect.height)
speed_x = 5

running = True
while running:
    pygame.time.delay(40) # segundo control de velocidad

    for event in pygame.event.get():
        if event.type == pygame.QUIT:
            running = False

    window.fill((255, 255, 255)) # limpiamos porque si no deja surco
    x += speed_x

    # condicional para que cuando toca un borde de la vuelta
    if x + cara_rect.width > window_x or x < 0:
        speed_x = -speed_x

    window.blit(cara, (x, y))
    pygame.display.flip()

```

```
pygame.quit()  
sys.exit()
```

```
#!/usr/bin/python

# Ejercicio 04:
# Modifica el programa anterior para que la foto se pueda mover en todas
# direcciones y que vaya
# rebotando por toda la pantalla.
#
# Victor Manuel Andreu Felipe 2025

import pygame
import sys
import random

# dimensiones de la ventana
window_x = 800
window_y = 600

pygame.init()

# Configurar la ventana
window = pygame.display.set_mode((window_x, window_y))
window.fill((255, 255, 255))

# cargar la imagen
cara = pygame.image.load("cara.jpg")
cara_rect = cara.get_rect()

# posición inicial y velocidad
x = random.randint(0, window_x - cara_rect.width)
y = random.randint(0, window_y - cara_rect.height)
speed_x = 4
speed_y = 3

# Bucle principal
running = True
while running:
    pygame.time.delay(30) # segundo control de velocidad

    for event in pygame.event.get():
        if event.type == pygame.QUIT:
            running = False

    window.fill((255, 255, 255))
    x += speed_x
    y += speed_y

    if x + cara_rect.width > window_x or x < 0:
        speed_x = -speed_x

    if y + cara_rect.height > window_y or y < 0:
        speed_y = -speed_y
```

```
    window.blit(cara, (x, y))
    pygame.display.flip()

pygame.quit()
sys.exit()
```

```

#!/usr/bin/python

# Ejercicio 05:
# Modifica el anterior programa para que en vez de una cara rebotando por la
# pantalla sean dos
# caras rebotando por la pantalla.
#
# Victor Manuel Andreu Felipe 2025

import pygame
import sys
import random

# dimensiones de la ventana
window_x = 800
window_y = 600

pygame.init()

# preparar la ventana
window = pygame.display.set_mode((window_x, window_y))
window.fill((255, 255, 255)) # Fondo blanco

# cargar la imagen
cara = pygame.image.load("cara.jpg")
cara_rect = cara.get_rect()

# posiciones iniciales y velocidades
caras = [
    {"x": random.randint(0, window_x - cara_rect.width),
     "y": random.randint(0, window_y - cara_rect.height),
     "speed_x": 4,
     "speed_y": 3},
    {"x": random.randint(0, window_x - cara_rect.width),
     "y": random.randint(0, window_y - cara_rect.height),
     "speed_x": 4,
     "speed_y": 5}
]

running = True
while running:
    pygame.time.delay(30) # segundo control de velocidad

    for event in pygame.event.get():
        if event.type == pygame.QUIT:
            running = False

    window.fill((255, 255, 255))

    for cara_obj in caras:
        cara_obj["x"] += cara_obj["speed_x"]
        cara_obj["y"] += cara_obj["speed_y"]

```

```
# condicionales para rebotar
if cara_obj["x"] + cara_rect.width > window_x or cara_obj["x"] < 0:
    cara_obj["speed_x"] = -cara_obj["speed_x"]

if cara_obj["y"] + cara_rect.height > window_y or cara_obj["y"] < 0:
    cara_obj["speed_y"] = -cara_obj["speed_y"]

window.blit(cara, (cara_obj["x"], cara_obj["y"]))

pygame.display.flip()

pygame.quit()
sys.exit()
```