

Py5

La librería **py5** es una herramienta poderosa para crear animaciones y gráficos en Python, inspirada en la popular librería Processing. A continuación te presento una guía básica para empezar a crear animaciones con **py5**:

1. Instalación de py5

Primero, necesitas instalar la librería **py5**. Puedes hacerlo fácilmente usando pip:

```
pip install py5
```

2. Configuración del Entorno

Asegúrate de tener configurado un entorno de desarrollo adecuado, como VSCode, PyCharm, o Jupyter Notebooks. También puedes ejecutar scripts directamente desde la terminal.

3. Estructura Básica de un Script py5

Un script básico en **py5** generalmente incluye las funciones **setup()** y **draw()**. La función **setup()** se ejecuta una vez al inicio, mientras que **draw()** se ejecuta en bucle, permitiendo la animación.

```
import py5

def setup():
    py5.size(400, 400) # Define el tamaño de la ventana

def draw():
    py5.background(255) # Establece el color de fondo en blanco
    py5.circle(py5.mouse_x, py5.mouse_y, 50) # Dibuja un círculo siguiendo el
mouse

py5.run_sketch()
```

4. Creación de Animaciones

Para crear animaciones, necesitas manipular las variables a lo largo del tiempo dentro de la función `draw()`.

Ejemplo 1: Movimiento de un Círculo

```
import py5

x = 0

def setup():
    py5.size(400, 400)
    py5.background(255)

def draw():
    global x
    py5.background(255)
    py5.fill(150, 0, 150)
    py5.circle(x, 200, 50)
    x += 2 # Mueve el círculo hacia la derecha

    if x > py5.width:
        x = 0 # Reinicia la posición si se sale de la ventana

py5.run_sketch()
```

Ejemplo 2: Animación de una Espiral

```
import py5
import math

angle = 0

def setup():
    py5.size(400, 400)
    py5.background(255)

def draw():
    global angle
    py5.background(255)

    for i in range(50):
```

```
x = 200 + math.cos(angle + i * 0.1) * i * 5
y = 200 + math.sin(angle + i * 0.1) * i * 5
py5.fill(150, i * 5, 200 - i * 5)
py5.circle(x, y, 10)

angle += 0.05 # Incrementa el ángulo para animar la espiral

py5.run_sketch()
```

5. Interacción con el Usuario

Puedes capturar eventos del mouse y del teclado para hacer la animación interactiva.

```
import py5

color = (150, 0, 150)

def setup():
    py5.size(400, 400)

def draw():
    py5.background(255)
    py5.fill(color)
    py5.circle(py5.mouse_x, py5.mouse_y, 50)

def mouse_pressed():
    global color
    color = (py5.random(255), py5.random(255), py5.random(255))

py5.run_sketch()
```

6. Guardar la Animación como GIF

Para guardar la animación como un GIF, puedes usar un bucle que guarde cada frame como una imagen, y luego unirlos usando una herramienta como [imageio](#).

```
import py5
import imageio

frames = []

def setup():
    py5.size(400, 400)
    py5.frame_rate(10) # Establece la velocidad de fotogramas

def draw():
    global frames
    py5.background(255)
    py5.fill(150, 0, 150)
    py5.circle(py5.width / 2, py5.height / 2, py5.frame_count % 100 + 10)

    frame = py5.get_image() # Captura el fotograma actual
    frames.append(frame)

    if py5.frame_count > 30: # Guarda 30 fotogramas y luego detén la animación
        py5.no_loop()
        save_gif()

def save_gif():
    gif_path = "animation.gif"
    imageio.mimsave(gif_path, frames, duration=0.1)
    print(f"GIF guardado como {gif_path}")

py5.run_sketch()
```

7. Recursos Adicionales

- **Documentación oficial:** [py5 documentation](#)
- **Ejemplos:** Explora ejemplos en línea para inspirarte y aprender técnicas más avanzadas.