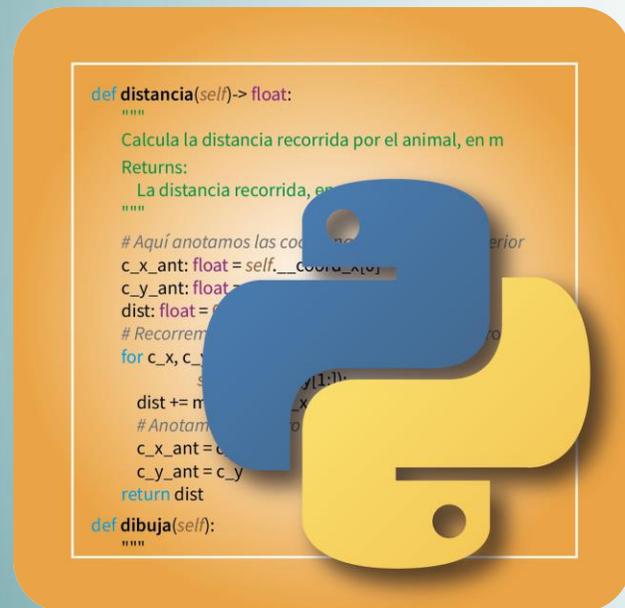


Programación

Práctica 4. Creación de objetos



Michael González Harbour
José Javier Gutiérrez García
José Carlos Palencia Gutiérrez
José Ignacio Espeso Martínez
Adolfo Garandal Martín

Departamento de Ingeniería
Informática y Electrónica

Este material se publica con licencia:
[Creative Commons BY-NC-SA 4.0](https://creativecommons.org/licenses/by-nc-sa/4.0/)

Práctica 4

Objetivo: Practicar con la creación de objetos y la invocación de sus métodos, para hacer un dibujo animado

Software suministrado: Un módulo llamado `dibujo.py` que contiene clases para dibujar figuras en una ventana, como estas:

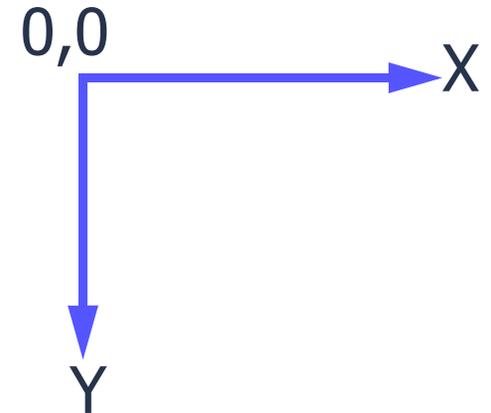
- **Dibujo:** representa el lienzo de dibujo
- **Ovalo:** representa una figura ovalada
- **Rectángulo:** representa un rectángulo
- **Línea:** representa una línea
- **Imagen:** representa una imagen obtenida de un archivo en formato **gif** o **png** (No valen imágenes **.jpg**)

Pertenece al paquete `fundamentos`. Las instrucciones de instalación y descarga están en la sección de recursos del moodle

Sistema de coordenadas

Coordenadas: las coordenadas de las figuras se describen en **píxeles** (puntos de pantalla)

- el sistema de coordenadas se sitúa en la esquina superior izquierda de la ventana
 - las coordenadas x avanzan hacia la **derecha** en el dibujo
 - las coordenadas y avanzan hacia **abajo** en el dibujo



Documentación del paquete fundamentos

Para comprender el uso de las clases suministradas puede consultarse la documentación desde un navegador de Internet

- está en la carpeta `docs`, dentro del paquete `fundamentos`
- al principio del módulo `dibujo.py` viene un ejemplo de cómo hacer un dibujo animado

Una vez importada la clase que se vaya a usar, se puede también acceder a su documentación con el comando `help` desde la consola

- p.e., `help(Dibujo)`, `help(Imagen.mueve)`, etc.

Repaso: Crear objetos

Crear objetos de una clase:

- Sintaxis

```
nombre_objeto = Clase(parámetros)  
# los parámetros son los del constructor
```

- Ejemplo

```
ventana=Rectangulo(dib, 10, 10, 30, 30, "green")  
# crea un rectángulo en la figura dib  
# con coordenadas (10,10) (30,30)  
# y de color verde
```

Repaso: Usar objetos

Invocar un método del objeto

- Sintaxis

```
nombre_objeto.nombre_método(parámetros)  
# los parámetros son los que requiera el método
```

- Ejemplo

```
ventana.mueve(20,30)  
# mueve la ventana 20 píxeles a la derecha  
# y 30 hacia abajo
```

Descripción de la práctica

Se pide hacer un módulo llamado `disparo.py` con una función `main()` que haga una animación del disparo de un proyectil, con los siguientes pasos:

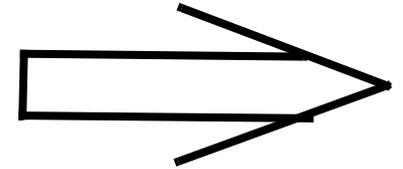
- Crear un objeto de la clase `Dibujo` de tamaño 900x600 píxeles
 - usaremos este objeto posteriormente al crear las figuras
- Crear un objeto de la clase `Imagen` hacia el centro-derecha del dibujo
 - esto será el blanco
 - puede descargarse una foto de Internet
 - pulsando el botón derecho del ratón sobre la imagen, elegir "**Guardar imagen como**", o similar, y guardarla en la carpeta del proyecto
 - elegir una imagen de 500x500 píxeles como máximo
- Crear un objeto de la clase `Ovalo` situado hacia la izquierda del dibujo, para pintar un círculo
 - esto será el proyectil
- Pintar el dibujo usando el método `pinta()`

Descripción (cont.)

- Esperar un segundo (ver apéndice)
- Mover el círculo 2 píxeles o más hacia la derecha y pintar el dibujo usando el método `pinta()`
- Mediante un bucle, reproducir las acciones del paso anterior tantas veces como sea necesario para que el círculo quede aproximadamente en el centro de la imagen
- Usando el fichero `boom.png` crear un objeto de la clase `Imagen` colocado de modo que tape el círculo y volver a pintar el dibujo
 - esto representará el impacto del proyectil
 - esta imagen mide 64x64 píxeles
- Esperar mediante el método `espera()` a que el usuario pulse el botón `OK`
- Finalmente, destruir el dibujo

Parte avanzada

Queremos crear un proyectil compuesto por varias líneas o figuras (al menos 3), como en el ejemplo



Para ello crearemos en nuestro módulo `disparo.py` una nueva clase, llamada `Proyectil` con:

- atributos:
 - las figuras que forman el proyectil
- métodos:
 - Constructor que crea en el dibujo las figuras del proyectil; el objeto de la clase `Dibujo` y la posición central inicial del proyectil se le pasarán al constructor como parámetros
 - método `mueve()` que desplaza el proyectil en la cantidad de píxeles horizontales y verticales que se indican como parámetros; lo hace a base de desplazar cada figura individualmente

Parte avanzada (cont.)

Crear en nuestro módulo `disparo.py` una nueva función principal llamada `main2()`, copia de la anterior, que haga una animación similar a la de la parte obligatoria pero con el nuevo proyectil en lugar del círculo original

- el código debe ser *idéntico* al del `main()` de la parte obligatoria, pero creando un `Proyectil` en lugar de un `Ovalo`

Entregar dos ficheros

1. Código fuente, incluyendo la parte obligatoria y la parte avanzada si se ha hecho
2. Informe en pdf conteniendo:
 - Parte obligatoria
 - Una captura de pantalla del dibujo tal como queda hacia el principio de la ejecución del programa
 - Una captura de pantalla del dibujo tal como queda al final de la ejecución
 - Parte avanzada (si se ha hecho)
 - Una captura de pantalla con el dibujo que queda hacia el principio de la ejecución

Apéndice

Función para esperar un tiempo especificado en segundos

Por ejemplo, 50 ms:

```
from time import sleep
```

```
sleep(0.05)
```