

# Módulo turtle — Decimonovena sesión Sui generis

Ramiro de Maeztu

Febrero 2024

Yo os enseño el superhombre. El hombre es algo que debe ser superado. ¿Qué habéis hecho vosotros para superarlo?... Yo os enseño al superhombre: ¡él es este rayo y esta demencia!

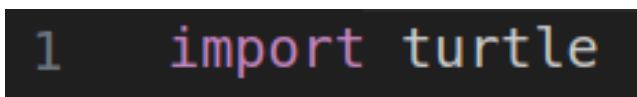
Zaratustra

## 1 Introducción

El módulo turtle puede usarse para graficar de forma simple figuras. En mi opinión es una basura comparado con el módulo pygame, pero es una buena forma de empezar a representar fenómenos matemáticos o similares.

Python tiene una serie de módulos que vienen preinstalados y que puedes importar directamente. Las librerías de OS (operative system), itertools (herramientas para iteradores, listas, tuplas, diccionarios), time (para medir el tiempo), math (herramientas matemáticas) y turtle vienen preinstaladas de forma que nosotros, los usuarios, no tenemos que hacer nada más que enunciar que queremos utilizar las librerías al principio del archivo para disponer de sus funcionalidades.

En la figura siguiente se ve como se usa la keyword *import* para importar un módulo que es en este caso turtle, pero podría ser cualquier otro de los mencionados previamente.



```
1 import turtle
```

Así se importa un módulo (no tiene mucho misterio)

## 2 Marco teórico

### 2.1 Control del objeto e inicialización del programa

Presta atención al siguiente código e intenta entender que hace cada cosa.

```
1  import turtle
2
3  wn = turtle.Screen()
4  t = turtle.Turtle()
5
6  t.forward(100)
7  t.left(90)
8  t.forward(50)
9
10 wn.mainloop()
```

Tenemos que definir una variable para nuestra pantalla. En este caso la he llamado "wn", pero la puedes nombrar lo que quieras. Acuérdate del nombre porque al final del programa vas a tener que escribir "tu nombre".mainloop() para que lo que has dibujado permanezca en la pantalla.

Luego tienes que definir otra variable que sea tu tortuga. La he llamado t, de turtle, pero la puedes llamar lo que quieras. Para conseguir que se mueva y deje un rastro debes utilizar los métodos: forward, backward, left y right con un argumento que tienes que especificar.

La tortuga está mirando hacia una dirección. Inicialmente, está mirando hacia el este. Si nosotros le decimos t.forward(100) se moverá hacia 100 píxeles hacia el este. Sin embargo, si ponemos t.left(90), haremos que la tortuga se gire 90º a la izquierda. Si ahora le decimos que se mueva t.forward(50), se moverá hacia el norte. Similarmente, si nosotros pusiéramos t.right(90), volvería a mirar hacia el este.

Es importante que aprendas la sintaxis bien, porque de otro modo el código te devolverá un error. Prueba el código mostrado en pantalla y varía algunas cosas para entender como funciona.

También hemos obviado algunas cosas. El tamaño de la ventana que abre turtle viene predefinido, y el objeto tortuga que hemos definido empieza en el centro de coordenadas.

## 2.2 Customización

Aunque ya tengas la capacidad de dibujar lo que quieras, hay mucho más que puedes hacer con tu tortuga, o la variable que hayas definido para nombrarla.

Aquí hay más métodos para customizar su movimiento.

**penup()**: Levanta el lápiz, evitando que la tortuga dibuje al moverse.

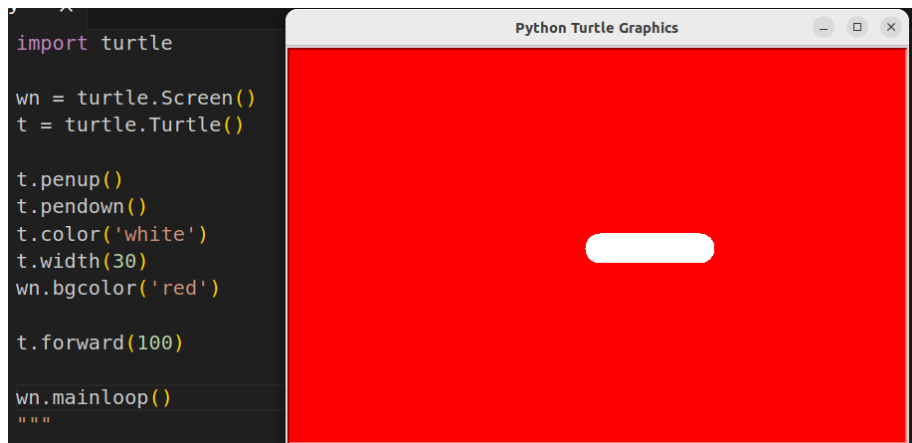
**pendown()**: Baja el lápiz para que la tortuga pueda dibujar al moverse.

**pensize(width)**: Establece el ancho de la línea de dibujo.

**color(color)**: Establece el color de la línea.

**bgcolor(color)**: Establece el color de fondo de la ventana.

Por ejemplo:



## 2.3 Otros métodos útiles

Aquí se pueden observar otros métodos para aplicar a la tortuga.

**goto(x, y):** Mueve la tortuga a las coordenadas (x, y) Mucho más conveniente para figuras específicas(nombres por ejemplo) que el método forward, porque no tienes que especificar la dirección. Importante, utiliza coordenadas cartesianas.

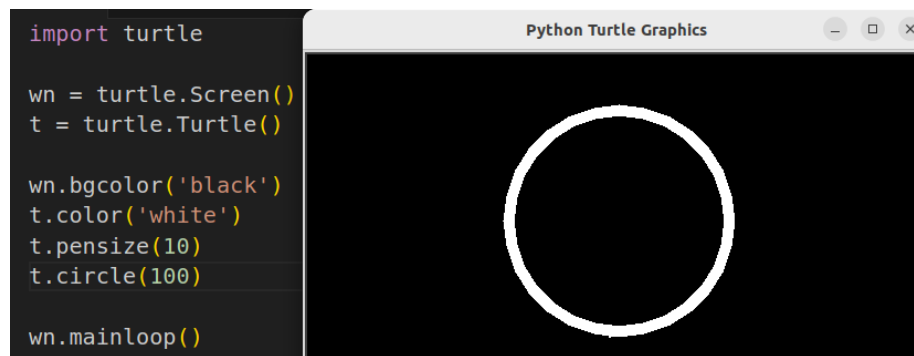
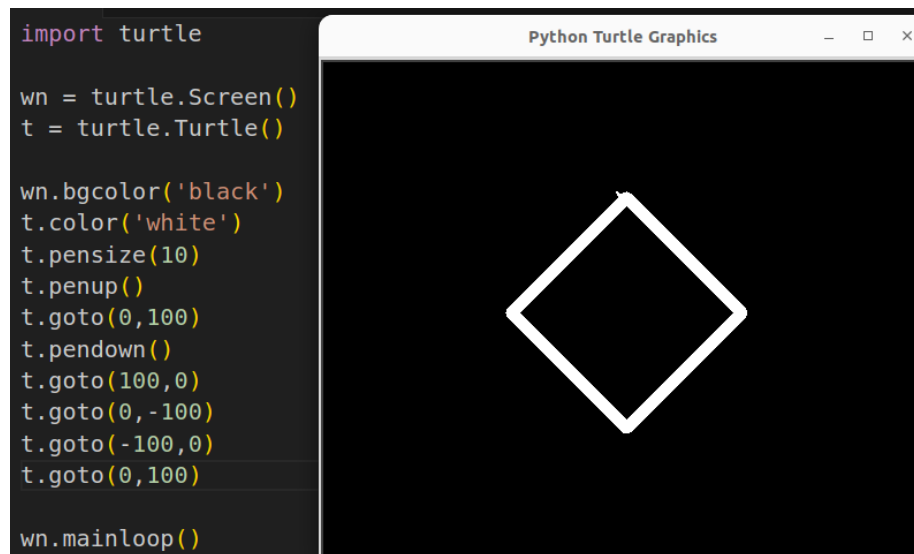
**circle(radio):** Dibuja un círculo con el radio. Se puede especificar `circle(radio, steps = n)` para crear un polígono regular con n lados. `circle(radio, n)` creará un círculo completo solo con n grados.

**clear():** Borra la pantalla y reinicia la posición de la tortuga.

**reset():** Borra todo y restablece la tortuga a su estado inicial.

**speed(velocidad):** Establece la velocidad de la tortuga (1 a 10, 1 es el más lento).

Ejemplos:

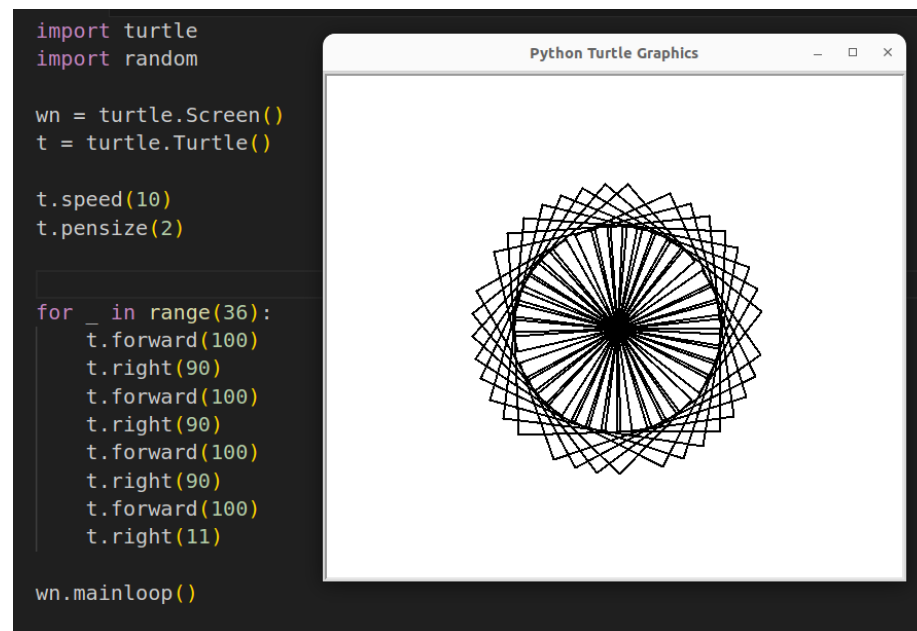


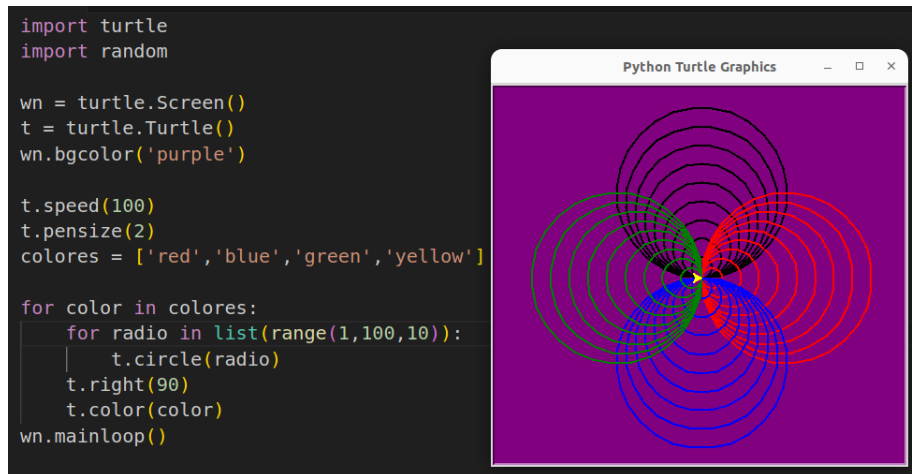
El resto de funcionalidades se explican solas.

## 2.4 Ejemplos de figuras bonitas

Para sacarle el máximo provecho al módulo, es pertinente ver cómo se hacen las figuras bien. Copiar a otros es esencial para convertirse en un buen programador.

Lo que hace el programa es crear un cuadrado, y luego rotar ligeramente dirección en la última línea del bucle para crear el patrón tan bonito que se aprecia.





Nadie espera que seas capaz de hacer figuras bonitas habiéndote leído este documento. Si te interesa el tema pregúntale a chatgpt o mira en youtube.

### 3 Ejercicios

1. Haz un cuadrado.
2. Crea un romboide.
3. Escribe la primera letra de tu nombre.
4. Crea un corazón, tienes que usar el método circle.
5. Haz un programa que haga cuadrados que se vayan haciendo más pequeños.
6. Utiliza el código de la penúltima figura para variar el ángulo de rotación.
7. Cacharrea.