

Dibujar con código: **la tortuga de Python**

1. ¿Qué es la tortuga?

Turtle quiere decir tortuga en inglés. Viene de Logo, un lenguaje que unos maestros inventaron para enseñar a programar. La idea detrás es que aprendemos mejor un concepto abstracto cuando lo podemos mover con el cuerpo. Tú no piensas "un ángulo de 170 grados": piensas "yo camino, yo giro". La tortuga es tu cuerpo puesto en la pantalla.

Por eso no dibujamos nosotros. Le decimos a la tortuga: "avanza 80", "gira 170 grados"; así el dibujo aparece por donde pasó la tortuga.

La tortuga les salió tan bien que otros lenguajes la fueron adoptando. Uno de ellos es Python: un idioma para darle instrucciones a la computadora, hoy de los más usados en el mundo, y el que vamos a usar aquí.

Python trae guardados un montón de paquetes de herramientas. A esos paquetes se les dice módulos, y la tortuga es uno de ellos.

Python no los carga todos de un jalón, porque serían miles y nomás estorbarían. Se los tienes que pedir, y por eso la primera línea de tus programas es `import turtle`, que quiere decir "tráeme el módulo de la tortuga".

Cómo leer los colores de este cuadernillo

El código de aquí viene pintado. Cada color quiere decir algo:

- **morado** · palabras que Python ya conoce. Se escriben siempre igual. Si les cambias una letra, truena.
- **verde** · números y palabras entre comillas. Esto es lo que puedes cambiar sin miedo. Aquí es donde juegas.
- **negro** · lo que le decimos a la tortuga. Va con punto y paréntesis.
- **# notas en cursiva** · lo que va después del gato es un recado para humanos. Python lo ignora. No tienes que copiarlo.

Antes de empezar

- Abre Thonny, que es el programa donde se escribe Python.
- Guarda tu archivo con un nombre tuyo, por ejemplo `espiral.py`.
- Nunca le pongas `turtle.py`. Python se confunde y cree que tu archivo es la tortuga.
- Corre el programa con el botón Run o con la tecla F5.

2. Las tres líneas que siempre van

```
import turtle           # trae la tortuga a tu programa
t = turtle.Turtle()     # nace tu tortuga y se llama t
t.speed(0)              # que camine lo más rápido posible
```

Qué hace cada una

<code>import turtle</code>	Trae la caja de herramientas de la tortuga. Sin esta línea, Python no sabe qué es una tortuga.
<code>t = turtle.Turtle()</code>	Crea una tortuga y la guarda en una caja llamada <code>t</code> . Cada vez que escribas <code>t</code> , te refieres a ella. Le podrías poner <code>tortuga</code> o <code>chelo</code> , mientras uses el mismo nombre siempre.
<code>t.speed(0)</code>	La velocidad. Del 1, que es lentísima, al 10, que es rápida. El 0 quiere decir sin animación, o sea lo más rápido de todo. Mientras aprendes, ponle 1 y mírala caminar.

Y hasta abajo de todo programa va esta línea:

```
turtle.done()          # deja la ventana abierta
```

Sin ella, la ventana se cierra en un parpadeo y no alcanzas a ver nada.

3. Diccionario de comandos

<code>t.forward(80)</code>	Avanza 80 pasos hacia donde esté viendo.
<code>t.backward(40)</code>	Retrocede 40 pasos, sin voltearse.
<code>t.right(90)</code>	Gira 90 grados a la derecha, en su lugar.
<code>t.left(45)</code>	Gira 45 grados a la izquierda, en su lugar.
<code>t.penup()</code>	Levanta la pluma. Camina pero ya no deja rastro.
<code>t.pendown()</code>	Baja la pluma. Vuelve a dibujar.
<code>t.pencolor("purple")</code>	Cambia el color de la línea.
<code>t.pensize(4)</code>	Cambia el grosor de la línea.
<code>t.circle(30)</code>	Dibuja un círculo de radio 30.
<code>t.goto(0, 0)</code>	Se va directo a una coordenada. El (0, 0) es el centro.
<code>t.home()</code>	Regresa al centro viendo a la derecha.
<code>t.hideturtle()</code>	Esconde la flechita para que solo se vea el dibujo.

Colores que puedes escribir tal cual, siempre entre comillas: "purple", "green", "red", "blue", "orange", "brown", "black", "pink", "gold", "turquoise".

4. Ejemplo 1: la estrella que se cierra sola

```
import turtle

t = turtle.Turtle()
t.speed(0)

for i in range(36):      # repite 36 veces lo de abajo
    t.forward(80)        # avanza 80
    t.right(170)         # gira 170 grados a la derecha

turtle.done()
```

Línea por línea

<code>for i in range(36):</code>	Esto es un ciclo: repite 36 veces lo que viene abajo. La <code>i</code> es un contador que lleva la cuenta: 0, 1, 2, hasta 35. Los dos puntos del final avisan que ahí empieza lo que se repite.
<code>t.forward(80)</code>	Avanza 80. Está adentro del ciclo, así que pasa 36 veces.
<code>t.right(170)</code>	Gira 170 grados. También pasa 36 veces.

¿Por qué sale una estrella y no un cuadrado?

Aquí está lo bonito. Al girar 170 grados casi te regresas por donde venías, pero no del todo: te faltan 10 grados. Ese pedacito que sobra hace que la figura no cierre luego luego, sino que se vaya abriendo en picos.

Haz la cuenta: 36 giros por 170 grados son 6120 grados. Y 6120 entre 360 da 17 vueltas exactas. Como sale número entero, la tortuga termina justo donde empezó y la figura cierra perfecto. Si no diera entero, el dibujo quedaría abierto.

Experimenta con este

170 por 144	Sale la estrella de cinco picos de toda la vida.
170 por 90	Sale un cuadrado, repetido encima de sí mismo.
170 por 175	Los picos se hacen más finos. Vas a necesitar más repeticiones.
170 por 91	Un cuadrado que se va girando poquito. Sube el 36 a 200 y verás.
80 por 200	La misma figura, más grande. Tal vez se salga de la pantalla.

5. Ejemplo 2: cuatro plantas en fila

```
import turtle

t = turtle.Turtle()
t.speed(0)

for planta in range(4):           # cuatro plantas
    for hoja in range(6):         # seis hojas en cada planta
        t.forward(40)             # sale la hoja
        t.backward(40)            # y se regresa al centro
        t.right(60)               # gira para la siguiente hoja
    t.penup()                     # levanta la pluma
    t.forward(60)                  # brinca al lugar de la otra planta
    t.pendown()                   # baja la pluma otra vez

turtle.done()
```

Un ciclo adentro de otro ciclo

Este ejemplo tiene dos ciclos, uno metido dentro del otro. El de afuera cuenta plantas. El de adentro cuenta hojas. Por cada planta se dibujan seis hojas. Cuatro por seis dan 24 hojas en total.

<code>for planta in range(4):</code>	Ciclo de afuera. Se repite cuatro veces, una por planta.
<code>for hoja in range(6):</code>	Ciclo de adentro. Por cada planta, esto corre seis veces.
<code>t.forward(40)</code> <code>t.backward(40)</code>	Sale y regresa por la misma línea. Así se dibuja un rayo que sale del centro, como una hoja.
<code>t.right(60)</code>	Gira 60 grados para poner la siguiente hoja. Seis hojas por 60 grados dan 360, la vuelta completa. Por eso la planta queda como abanico cerrado.
<code>t.penup()</code> <code>t.forward(60)</code> <code>t.pendown()</code>	Estas tres ya salieron del ciclo de las hojas. Levantan la pluma, se mueven 60 de lado sin dibujar y la vuelven a bajar. Es el brinco al lugar de la siguiente planta. Sin el <code>penup</code> , quedaría una raya fea uniendo todo.

La sangría manda

En Python los espacios de la izquierda no son adorno. Son los que dicen quién está adentro de quién. Van cuatro espacios por cada nivel.

- Pegado a la orilla: pasa una sola vez.
- Cuatro espacios: está adentro del ciclo de las plantas.
- Ocho espacios: está adentro del ciclo de las hojas.

Mueve `t.penup()` a ocho espacios en vez de cuatro y el dibujo cambia por completo.
Hazlo a propósito para verlo.

6. Comandos para subirle

Estos no venían en los ejemplos, pero te van a servir para lo tuyo.

Pintar el relleno

```
t.fillcolor("green")      # el color de adentro
t.begin_fill()           # aquí empieza el relleno

for lado in range(6):
    t.forward(60)
    t.right(60)

t.end_fill()             # aquí se pinta
```

Todo lo que dibujes entre esas dos líneas se rellena de color cuando la figura cierra.

Cambiar el fondo

```
turtle.bgcolor("black")  # el fondo de la ventana
t.pencolor("purple")     # el color de la línea
```

Colores que van cambiando solos

```
colores = ["purple", "green", "blue", "orange", "red", "gold"]

for i in range(36):
    t.pencolor(colores[i % 6])  # agarra el siguiente color
    t.forward(80)
    t.right(170)
```

La lista guarda seis colores. El `i % 6` saca el residuo de dividir entre 6, así que va dando 0, 1, 2, 3, 4, 5, y de vuelta al 0. Cada repetición agarra el que sigue.

Que crezca en cada vuelta

```
for i in range(60):
    t.forward(i * 3)          # cada paso más largo que el anterior
    t.right(91)
```

Como `i` va creciendo, cada paso es más largo. Sale una espiral cuadrada.

Ajustar la ventana a la pantalla chiquita

```
turtle.setup(width=460, height=300)
```

Ponla justo después del `import`. Sirve cuando la pantalla es pequeña y el dibujo no cabe.

7. Si algo falla

IndentationError	Los espacios de la izquierda no cuadran. Revisa que todo lo del ciclo tenga los mismos cuatro espacios.
SyntaxError	Falta algo chiquito: los dos puntos al final del <code>for</code> , un paréntesis sin cerrar, una comilla suelta.
NameError	Usaste un nombre que no existe. Casi siempre es una <code>t</code> escrita de otro modo, o un color sin comillas.
AttributeError	Tu archivo se llama <code>turtle.py</code> . Cámbiale el nombre.
La ventana se cierra sola	Falta <code>turtle.done()</code> hasta abajo.
No se ve nada	La pluma quedó levantada, o el dibujo se salió de la pantalla. Baja los números.

La regla de oro

Cambia un solo número a la vez y vuelve a correr el programa. Si cambias tres cosas juntas y sale algo raro, ya no sabes cuál fue. Así se depura, y así trabaja quien programa todos los días.

8. Tu turno: diseña el tuyo

Antes de escribir código, dibújalo en papel y contesta las preguntas. Es más rápido pensarlo aquí que borrar quince veces en la compu.

¿Qué quiero dibujar?

¿Qué se repite? ¿Cuántas veces?

¿De cuánto es cada paso y cada giro?

¿Hay algo adentro de algo? ¿Qué va en el ciclo de afuera?

Escribe aquí tu código

Retos por si te atoras

- Una flor con doce pétalos usando `t.circle()` adentro de un ciclo.
- Tu inicial hecha con `goto()` y `penup()`.
- Una milpa: tres o cuatro plantas del ejemplo 2, pero cada una de distinto tamaño.
- Una greca al estilo de las de barro, puro `forward` y giros de 90 grados.
- La espiral cuadrada, pero cambiando de color en cada vuelta.

La tortuga sabe hacer mucho más de lo que cabe aquí. La lista completa de comandos vive en:

docs.python.org/es/3/library/turtle.html

Notas
