

## PRÁCTICA GUIADA: OBJETO RECTÁNGULO

Pretendemos definir una clase Rectángulo que nos permita crear esos objetos geométricos; además usaremos *turtle*: un módulo estándar para dibujar en una ventana gráfica usando una “tortuga” (sospechosamente parecida a un triángulo) que se mueve siguiendo las instrucciones que le demos. Este módulo hay que importarlo al comienzo del programa.

```
import turtle
class Rectangulo:
    def __init__(self, ancho, alto, color):
        self.ancho = ancho
        self.alto = alto
        self.color = color
        self.t = turtle.Turtle()
        self.t.color(self.color)

    def dibujar(self):
        for _ in range(2):
            self.t.forward(self.ancho)
            self.t.right(90)
            self.t.forward(self.alto)
            self.t.right(90)

rectangulo = Rectangulo(100,50, "red")
rectangulo.dibujar()
turtle.done()
```

Explicación del código:

Primero importamos la clase *turtle* (sin ella no podremos usar los gráficos).

A continuación definimos la clase *Rectángulo* que tiene un método inicializador (también se le llama constructor) con los siguientes parámetros:

- self: para hacer referencia al objeto.
- ancho: longitud horizontal del rectángulo, medido en píxels.
- alto: longitud vertical del rectángulo, medido en píxels.
- color: del borde; hay varios colores disponibles.

Las posibilidades de color en *turtle* de Python son:

Básicos	Extendidos	Tonos claros y oscuros	Tipo web (css)	RGB
black, white, red, green, blue, yellow, orange, purple, pink, brown, gray	cyan, magenta, violet, lime, gold, silver, navy, teal, marron, olive	lightblue, lightgreen, lightgray, darkblue, darkgreen, darkred, darkorange, darkviolet	tomato, coral, salmon, chocolate, khaki, plum, orchid, turquoise, skyblue	Valores RGB entre 0-255. turtle.colormode(255) t.color(255,0,0) rojo t.color(0,255,0) verde t.color(0,0,255) azul

## PRÁCTICA GUIADA: OBJETO RECTÁNGULO

Sabemos ya que cuando se cree un objeto rectángulo el método `__init__` (constructor) internamente hace algo así:

```
__init__(rectángulo, 100, 50, "red")
```

Creamos el objeto con esta línea:

```
rectangulo = Rectangulo(100,50, "red")
```

Donde indicamos el ancho, el alto y el color.

Para empezar a dibujar tenemos que crear un objeto llamado tortuga gráfica; lo hacemos así:

```
self.t = turtle.Turtle()
```

A cada rectángulo que creamos se asignará su propia tortuga gráfica.

Para modificar el color por defecto usamos:

```
self.t.color(self.color)
```

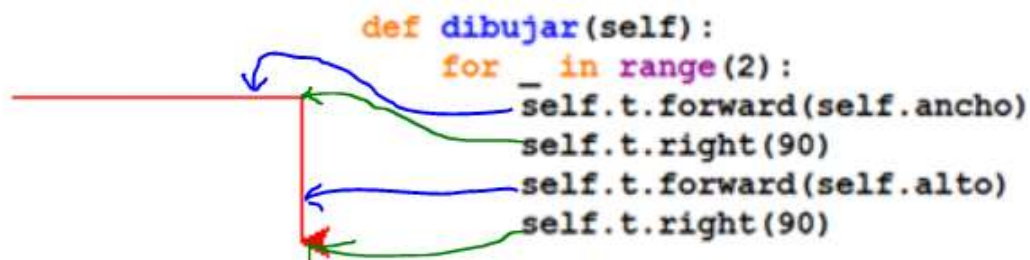
El método para dibujar el rectángulo es `rectangulo.dibujar()`

El bucle `for` del interior del método que dibuja el rectángulo hace lo siguiente:

(aclaración: se puede hacer un bucle sin una variable cuando no nos interesa usar el valor de esa variable en el interior del bucle, por eso este bucle no es `for i in range(2)`, sino `for _ in range(2)`).

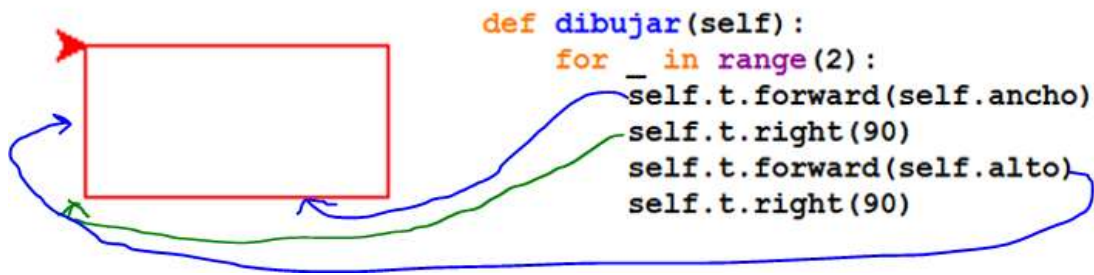
El bucle va del rango 0 a 1 porque el rectángulo se dibuja de esta manera: en la primera iteración se dibuja primero el ancho (línea superior), a continuación se hace un giro de noventa grados y se traza la línea del lateral derecho; por último, se hace un giro de noventa grados; en la segunda iteración del bucle, y partiendo de la posición en que se quedó la tortuga después de la primera iteración, se dibuja un nuevo ancho (la línea horizontal inferior), se hace un giro de noventa grados y se dibuja la línea lateral izquierda. Para que la tortuga quede en la posición inicial después de dibujar la línea vertical derecha se hace un giro de noventa grados.

Primera iteración del bucle:



## PRÁCTICA GUIADA: OBJETO RECTÁNGULO

Segunda iteración del bucle:



En el cuerpo del programa (instancia) se crea un objeto de la clase Rectángulo con un ancho de 100, un alto de 50 y color rojo:

```
rectangulo = Rectangulo(100,50, "red")
```

A continuación se llama al método que dibuja el rectángulo con los valores que hemos usado para crearlo:

```
rectangulo.dibujar()
```

Por último: se usa un truco para que la ventana donde se ha dibujado el rectángulo no se cierre:

```
turtle.done()
```

La ventana donde se ha hecho el dibujo, dependiendo del entorno en que se haya hecho el programa puede cerrarse; con esta instrucción le decimos a Python que quede a la espera de algún evento y que no se cierre.

Algo más:

La tortuga gráfica se mueve, por defecto, hacia la derecha y con un ángulo de cero grados, partiendo de la posición  $x=0$ ,  $y=0$ ; es decir: (0,0); se considera que esta posición se encuentra en el centro de la ventana que se abre para hacer el dibujo. Si queremos la cambiar la posición de inicio de la tortuga tendremos que usar las siguientes instrucciones:

<b>self.t.penup()</b>	Levantamos el lapicero de la tortuga
<b>self.t.goto(-50,50)</b>	Desplazamos la tortuga
<b>self.t.pendown()</b>	Bajamos el lapicero de la tortuga

Es importante tener en cuenta que las dimensiones del lienzo sobre el que se dibuja son de 800x600 píxels:

-400 a 400 para el eje horizontal (X)

-300 a 300 para el eje vertical (Y)

## PRÁCTICA GUIADA: OBJETO RECTÁNGULO

Aunque podemos controlar las dimensiones de la ventana en la que se hace el dibujo con:

```
screen = turtle.Screen()  
screen.setup(width=800, height=600)
```

Reto: hacer un dibujo como el siguiente:

