



El futuro digital
es de todos

MinTIC



Gráficos con Turtle

OPERADO POR:



Mision
TIC 2022

ruta de aprendizaje 1



Gráficos con Turtle

Turtle es una biblioteca de Python preinstalada que permite a los usuarios crear imágenes y formas proporcionándoles un lienzo virtual. El lápiz en pantalla que usa para dibujar se llama tortuga y esto es lo que le da a la biblioteca su nombre. En resumen, la biblioteca de tortugas de Python ayuda a los nuevos programadores a tener una idea de cómo es la programación con Python de una manera divertida e interactiva.

Turtle es usado principalmente, para mostrar las capacidades que tiene Python para apoyarnos en el proceso de solución de problemas.





La tortuga o **turtle** se utiliza principalmente para introducir a los niños en el mundo de las computadoras. Es una forma sencilla pero versátil de comprender los conceptos de Python.



Esto lo convierte en una gran vía para que los niños den sus primeros pasos en la programación de Python. Dicho esto, ¡la biblioteca de tortugas Python no está restringida solo a los más pequeños!

También ha demostrado ser extremadamente útil para los adultos que están probando Python, lo que lo hace ideal para los principiantes de Python.

Con la biblioteca de tortugas Python, puede dibujar y crear varios tipos de formas e imágenes. La mayoría de los desarrolladores usan tortugas para dibujar formas, crear diseños y hacer imágenes. Otros usan la tortuga para crear minijuegos y animaciones, como el que viste arriba.



Empezando con turtle

Entorno Python: asegúrese de estar familiarizado con su entorno de programación. Puede utilizar aplicaciones como IDLE o Jupyter Notebook para programar con turtle. Sin embargo, si no se siente cómodo con ellos, puede programar con REPL, que utilizará en este tutorial.

Versión de Python: asegúrese de tener la versión 3 de Python en su computadora. Si no es así, puede descargarlo del sitio web de Python. Para obtener ayuda con la configuración, consulte la Guía de instalación y configuración de Python 3.



Lo bueno de **turtle** es que es una biblioteca incorporada, por lo que no es necesario instalar ningún paquete nuevo. Todo lo que necesita hacer es importar la biblioteca a su entorno de **Python**. Antes de comenzar con la programación de Python, debe comprender qué es una biblioteca.



En el mundo no informático, una biblioteca es un lugar donde se almacenan diferentes tipos de libros. Puede acceder a estos libros en cualquier momento, tomar de ellos la información que necesite y devolverlos al mismo lugar.

En el mundo de la informática, una biblioteca funciona de manera similar. Por definición, una biblioteca es un conjunto de funciones y métodos importantes a los que puede acceder para facilitar su programación. La biblioteca de **turtle** de Python contiene todos los métodos y funciones que necesitará para crear sus imágenes.



En algunos casos es necesario instalar la librería, para realizar la instalación debemos escribir **pip install PythonTurtle** en la ventana de consola, y oprimir la tecla **Enter**, con esto se instalará la librería si no viene con los paquetes que descargamos con Anaconda.



```
In [6]: pip install PythonTurtle
```



Para acceder a una biblioteca de Python, debe importarla a su entorno de Python, así: `import turtle` debe ir al inicio del código, de igual manera, `turtle.done()` debe ir al final para decirle al código que debe terminar el proceso una vez siga todas las instrucciones.



```
8  import turtle
9
10  tortuga = turtle.Turtle()
11
12
13
14  turtle.done()
```




Ahora que tenemos **turtle** en el entorno de Python, podemos comenzar a programar con él. **Turtle** es una biblioteca gráfica, lo que significa que tendrás que crear una ventana separada (llamada pantalla) para ejecutar cada comando de dibujo. Puede crear esta pantalla inicializando una variable para ella.

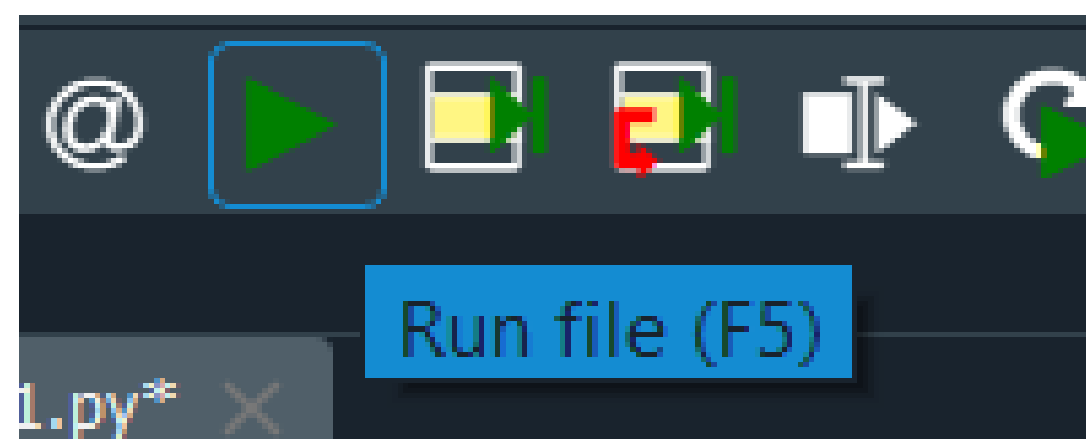
En Python, usa variables para almacenar información que usará más adelante en su programa. Inicializa una variable cuando le asigna un valor inicial. Dado que el valor de la variable no es constante, puede cambiar varias veces durante la ejecución de su programa.

Ahora, para abrir la pantalla de la tortuga, inicializamos una variable, la cual tendrá un nombre cualquiera, para este ejemplo vamos a llamar al cursor, “tortuga” y debemos escribirlo de la siguiente manera:

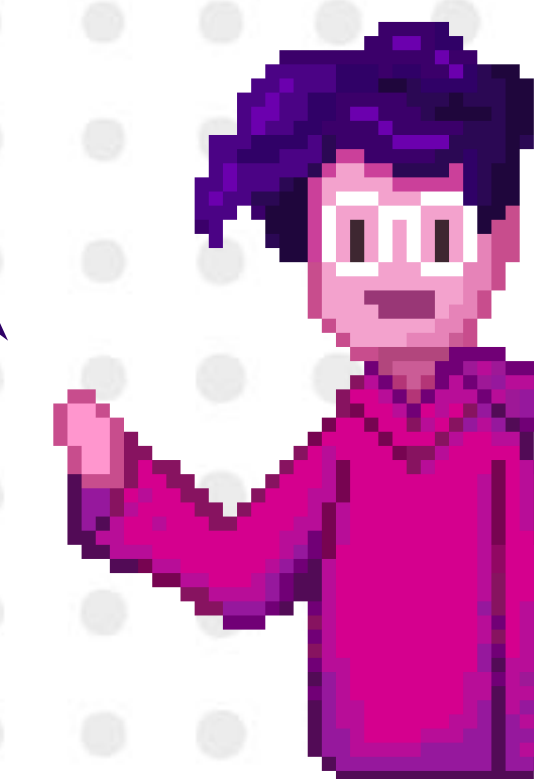
```
tortuga = turtle.Turtle()
```




Al escribir estas 3 líneas de código y hacer click en Run file, “flecha de color verde” o hacer click en F5:

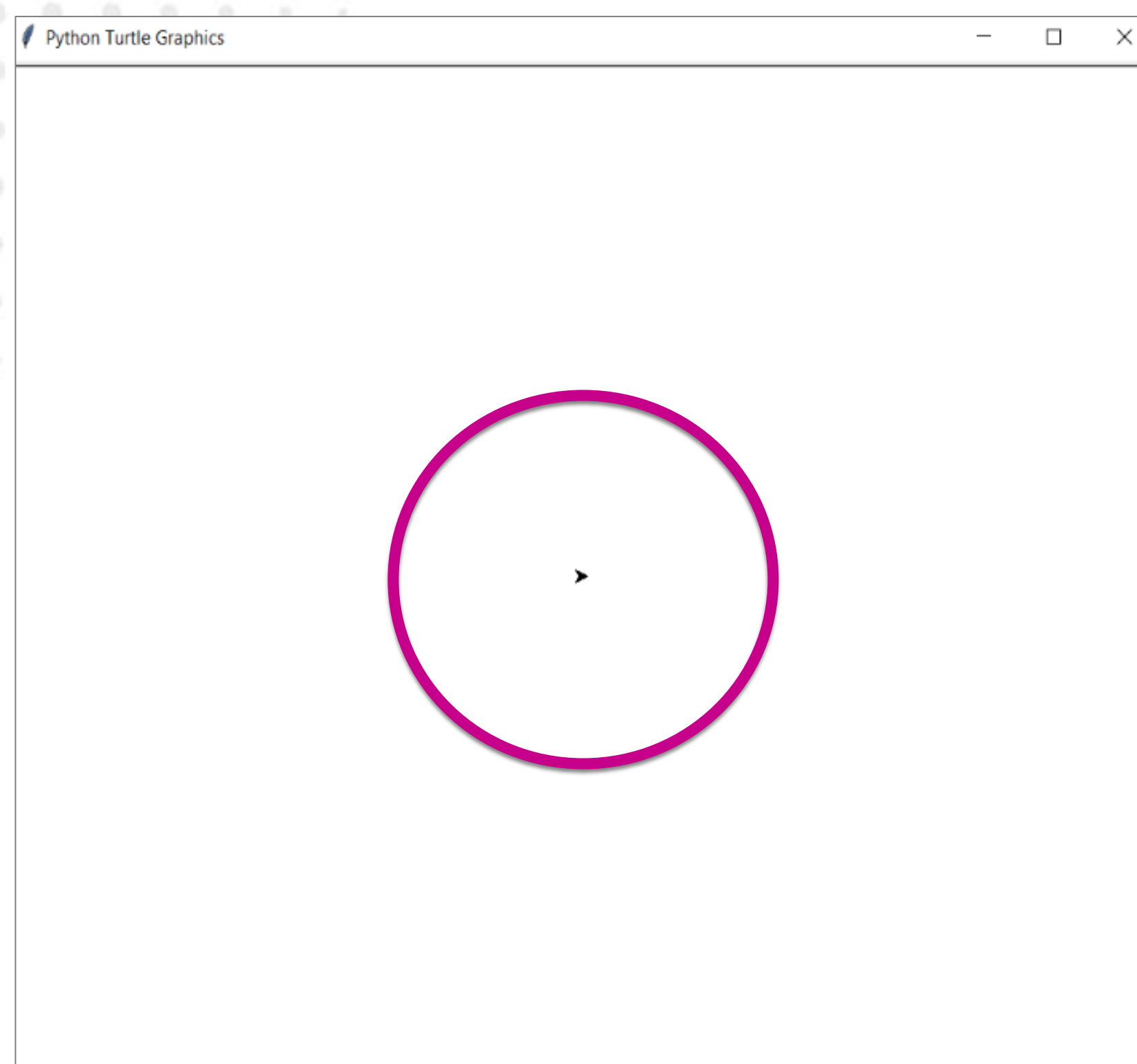


Podemos ejecutar nuestro código en Spyder y observar cual es la respuesta de que nos entregan nuestras instrucciones en la ventana de consola.





Debe salir una ventana nueva que tiene la siguiente forma:



Ahora tenemos nuestra ventana, donde podremos realizar las graficas que necesitemos, el cursor o la flecha pequeña de color negro seria nuestra **“tortuga”** y debemos indicarle como moverse para que haga las gráficas que queremos.



Vamos a indicarle ahora como debe moverse para que comencemos a dibujar, para eso vamos a escribir la siguiente línea de código o instrucción:

```
tortuga.forward(100)
```

Le estamos diciendo al código: “tortuga” muévete hacia adelante “forward” 100 pixeles (100), con esa instrucción ahora tenemos que detener el proceso anterior, algo que ocurre en muchas ocasiones, para eso debemos hacer click en “**Stop the current command**” o el “cuadrado de color rojo” para detener el “kernel” o el proceso que esta haciendo Python, y poder ejecutar nuestro código con la nueva instrucción:

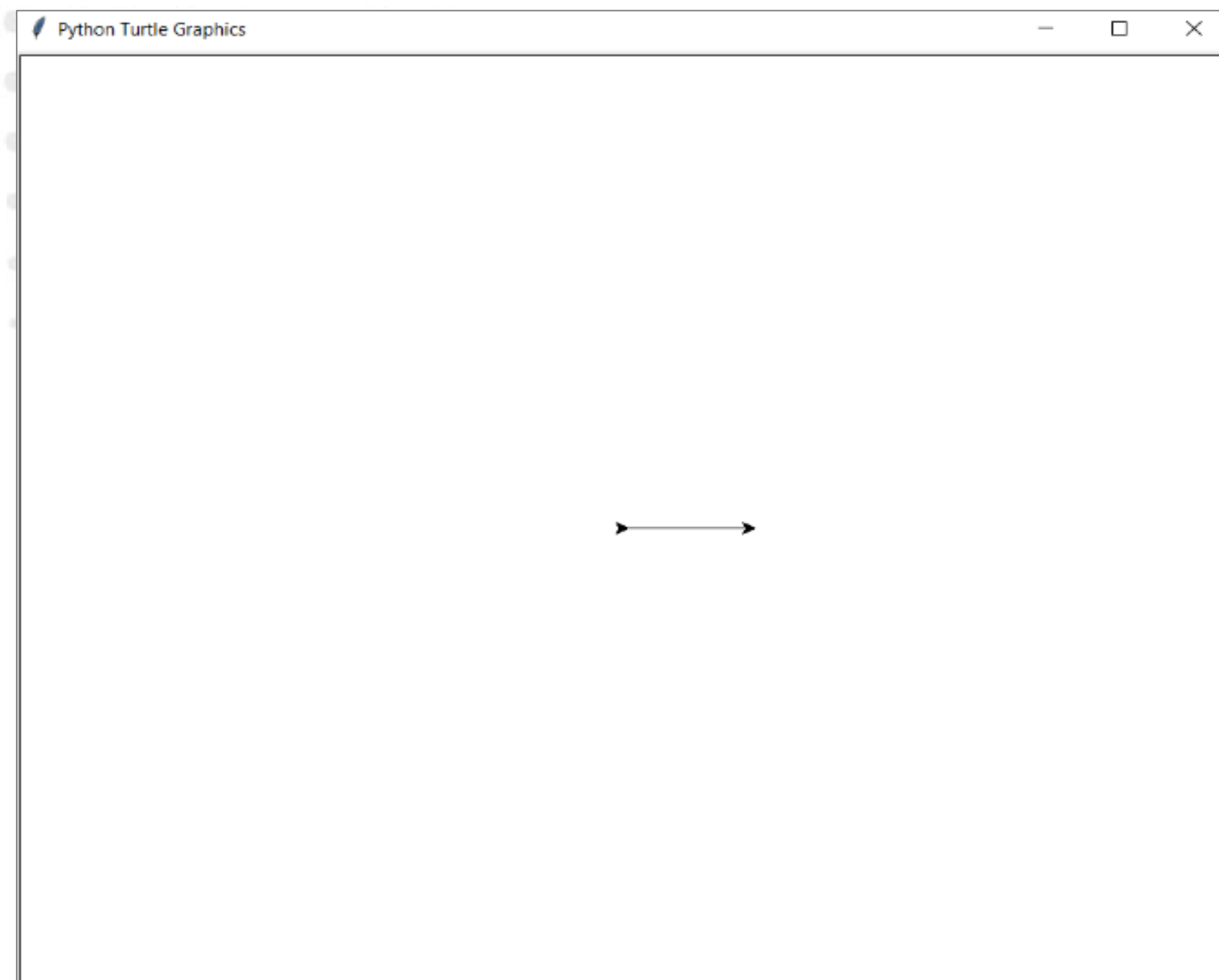
```
Console 1/A X
```

Stop the current command

```
In [3]: runfile('C:/Users/LauraM/untitled1.py', wdir='C:/Users/LauraM')
Traceback (most recent call last):
```



Una vez detenemos el proceso y hacemos click nuevamente en Run para ejecutar el código, nos debe aparecer la siguiente imagen en la ventana que abrimos anteriormente:



Ahora vamos a decirle a la “tortuga” que gire 45 grados usando la instrucción:

```
tortuga.left(45)
```

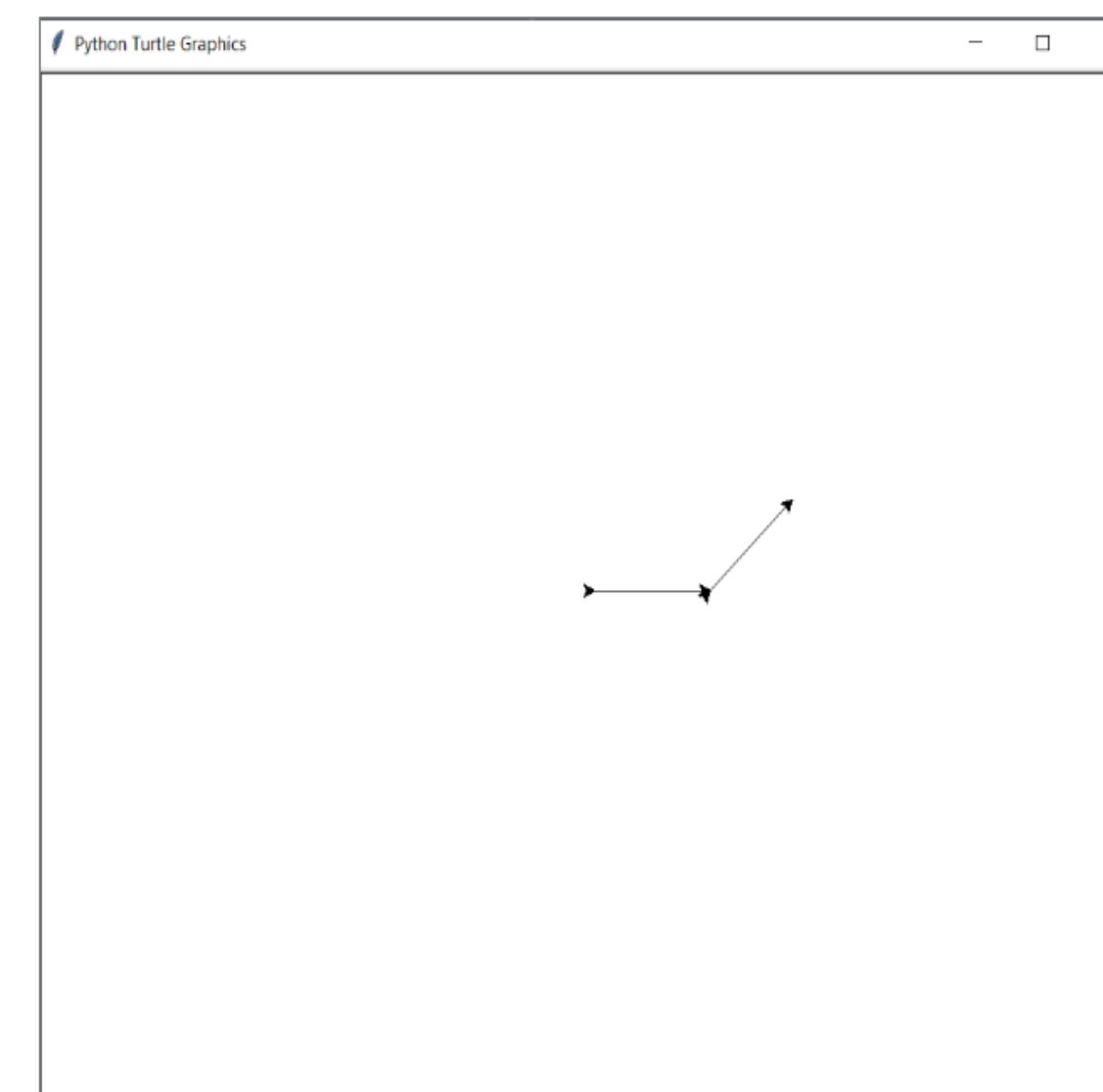
Y vamos a decirle nuevamente que camine hacia adelante 100 pixeles:

```
tortuga.forward(100)
```




Todas las instrucciones que turtle tiene disponible para hacer dibujos o diagramas, se pueden encontrar en la página de documentación oficial:
<https://docs.python.org/3.3/library/turtle.html>

Podemos de igual manera pedirle que haga formas típicas, como cuadrados, triángulos, etc. Para eso vamos a comentar las líneas que escribimos anteriormente usando el símbolo #. Y vamos a usar las mismas instrucciones, pero esta vez, vamos a darle las instrucciones necesarias para que dibuje un cuadrado:



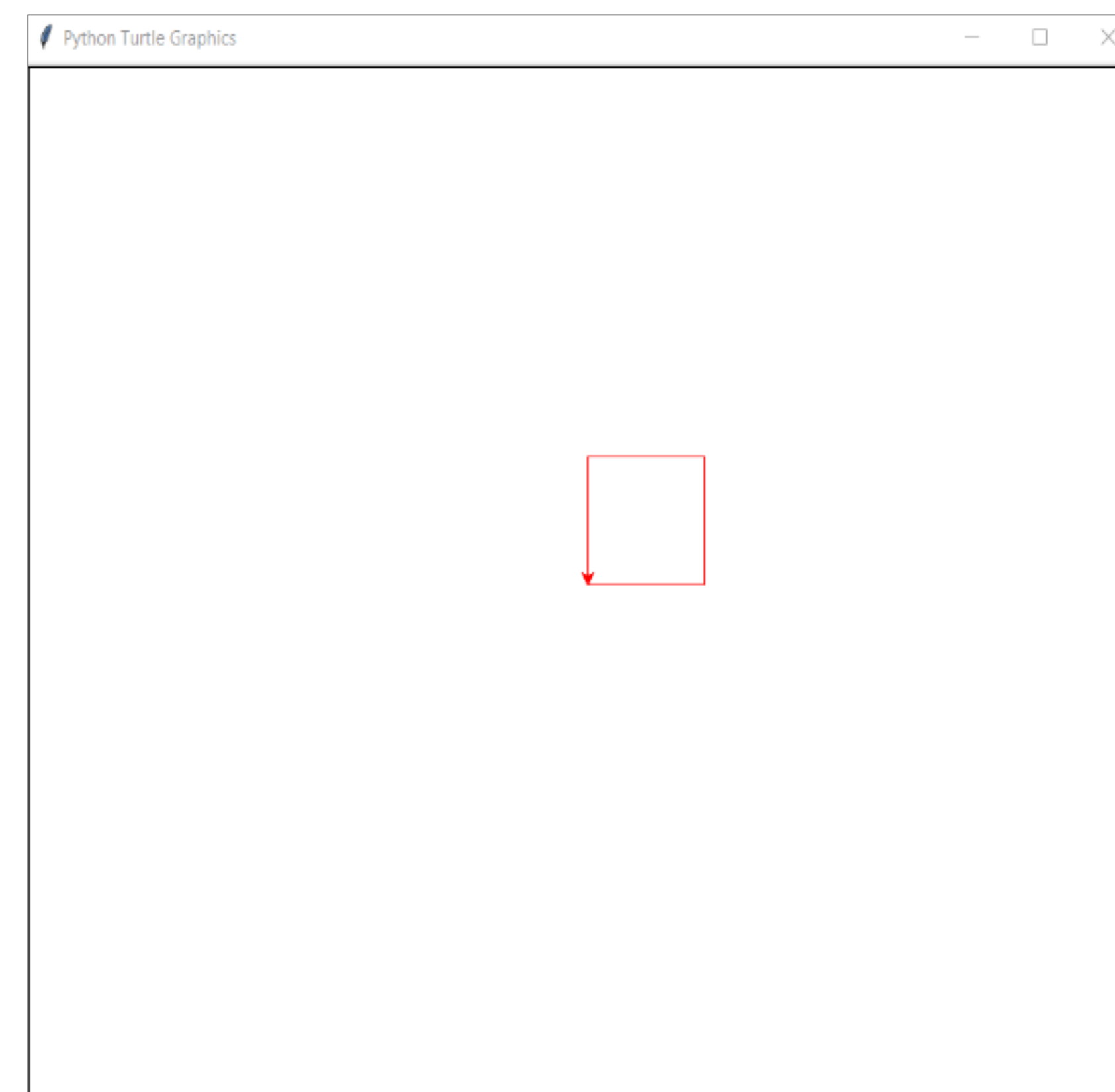


Ahora vamos a ver otras instrucciones simples antes de realizar diagramas más complejos:

Vamos a indicarle a turtle que además de seguir las instrucciones para dibujar un cuadrado, necesitamos que el color del cuadrado sea rojo, para eso vamos a usar la instrucción:

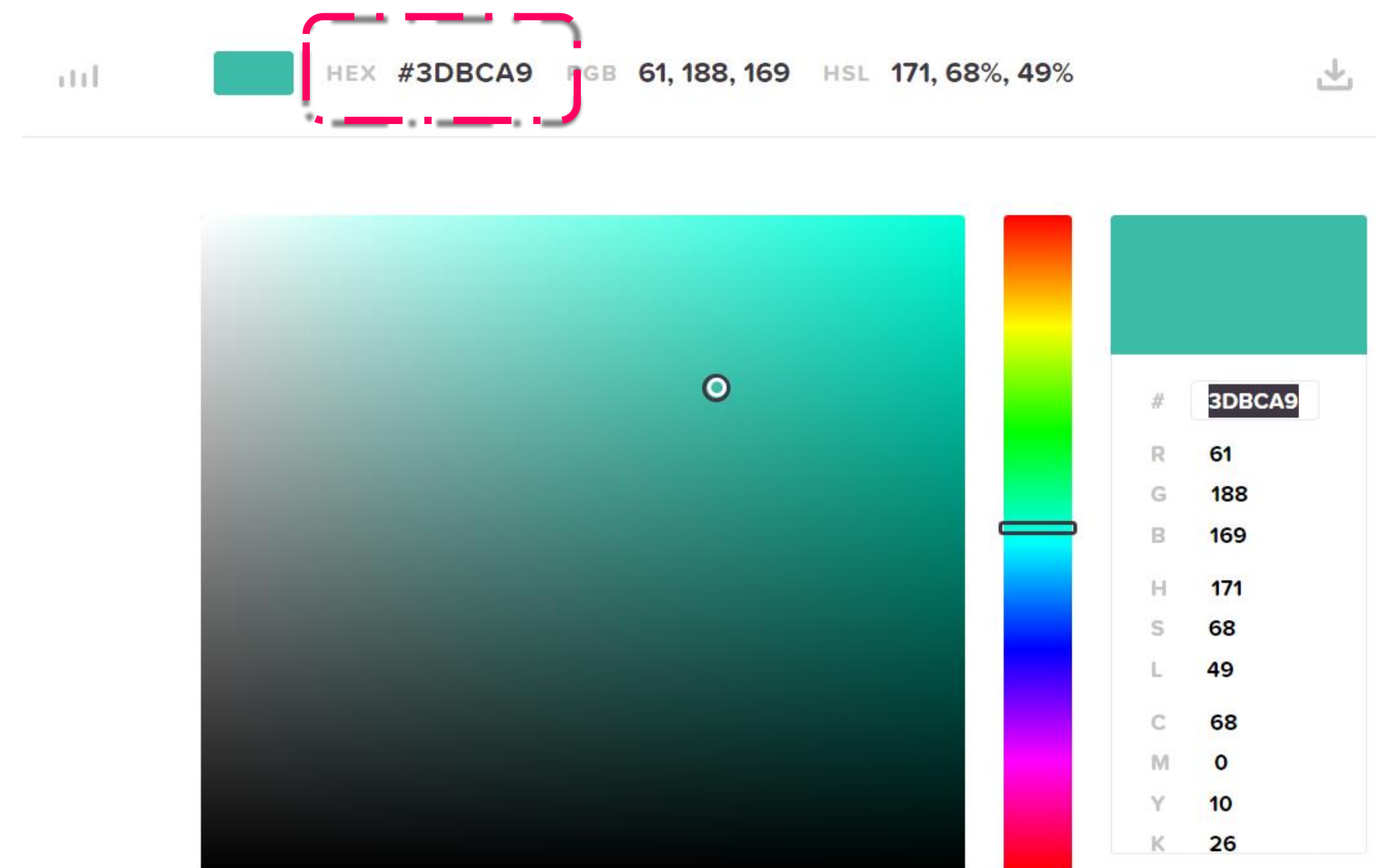
```
Tortuga.color("red")
```

La palabra color le indica de que color es la línea con la que dibuja nuestra tortuga, y el nombre del color debe ir entre comillas, en este caso rojo en inglés.



También podemos pedirle que use un color que no esta definido, para eso podemos utilizar paginas especiales como: <https://htmlcolorcodes.com/es/selector-de-color/> donde podemos elegir el color que queramos, y saber que código representa ese color.

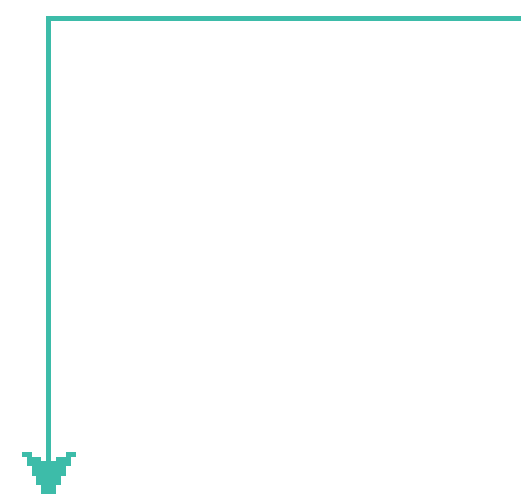
Por ejemplo vamos a seleccionar un color cualquiera, que no esta definido por turtle con nombres básicos, para poder utilizarlo, necesitamos copiar su código en este caso **HEX #3DBCA9** donde solo necesitamos copiar la parte que viene acompañada del numeral y ponerla en nuestro código en Python.



```
19  
20     tortuga.color("#3DBCA9")  
21
```



Así tendremos nuestro cuadrado delineado con un color especial:



El cuadrado queda delineado con el color que le especificamos. Es importante recordar que debe ser con el código correcto para que no tengamos un error.





También podemos decirle que rellene la forma que hemos echo de un color especifico, para eso necesitamos estas dos instrucciones: **tortuga.begin_fill()** al inicio de las instrucciones y **tortuga.end_fill()** al final como se muestra a continuación:

```
19  
20  tortuga.color("blue")  
21  
22  
23  tortuga.begin_fill()  
24  
25  tortuga.forward(100)  
26  
27  tortuga.left(90)  
28  
29  tortuga.forward(100)  
30  
31  tortuga.left(90)  
32  
33  tortuga.forward(100)  
34  
35  tortuga.left(90)  
36  
37  tortuga.forward(100)  
38  
39  tortuga.end_fill()  
40
```



Ahora tenemos el cuadrado definido como en el caso anterior, pero con un color azul de fondo.



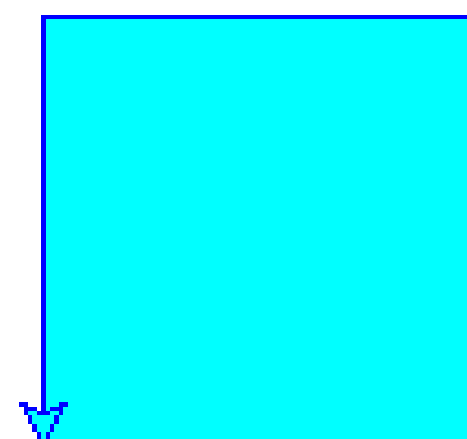


De igual manera, podemos diferenciar entre el color de línea y el color de fondo de la figura que estamos graficando. Para eso debemos definir ambos valores por separado:

```
tortuga.color("blue", "cyan")
```



En este ejemplo, el color de línea es azul oscuro, y el color cyan sería el color de fondo.





El futuro digital
es de todos

MinTIC

GRACIAS

OPERADO POR:

