UNIVERSIDAD LUTERANA SALVADOREÑA

carnet	estudiante
PH01137391	Diana Beatriz Perez Hernandez
GP01137393	José Salathiell García Pérez
MJ01137252	Noemy Marisol Mejia Jacinto
RP01137145	**José Saul Raymundo Pérez**

DOCENTE: LIC.DAVID CLIMACO

Tema: Python Turtle

Algoritmo

INTRODUCCION

¿Qué es turtle?

La "tortuga" de Logo es un cursor al que se le pueden dar órdenes de movimiento (avance, retroceso o giro) y que puede ir dejando un rastro sobre la pantalla. Moviendo adecuadamente la tortuga se pueden conseguir dibujar todo tipo de figuras. Python fue concebido a finales de la década de 1980 por Guido van Rossum en Centrum Wiskunde & Informática (CWI) en los Países Bajos como sucesor del lenguaje de programación ABC, que se inspiró en SETL, capaz de manejar excepciones e interactuar con el sistema operativo Amoeba. Su aplicación comenzó en diciembre de 1989. Van Rossum asumió la responsabilidad exclusiva del proyecto, como desarrollador principal, hasta el 12 de julio de 2018, cuando anunció sus "vacaciones permanentes" de sus responsabilidades como "dictador benevolente de por vida" de Python, un título que la comunidad de Python le otorgó para reflejar su compromiso a largo plazo como principal responsable de la toma de decisiones del proyecto. En enero de 2019, los desarrolladores activos del núcleo de Python eligieron un Consejo Directivo de cinco miembros para dirigir el proyecto.

El creador de Python: Conoce a Guido Van Rossum Su origen se remonta a finales de 1980 Guido Van Rossum es el creador y responsable de que Python exista. Se trata de un informático de origen holandés que fue el encargado de diseñar Python y de pensar y definir todas las vías posibles de evolución de este popular lenguaje de programación

turtle era parte original del lenguaje de programación logo, desarrollado por Wally Feurzeig ,Seymour Papert y Cynthia Solomon en 1967
El módulo turtle es una re implementación extendida del mismo módulo de la distribución estándar Python hasta la versión 2.5.
turtle provee las primitivas gráficas, tanto en orientación procedimental como orientada a objetos
Movimiento de Turtle
turtle.forward(distance)
turtle.fd(distance)
Movimiento de Turtle
turtle.right(angle)
turtle.rt(angle)
turtle.left(angle)
turtle.lt(angle)
El comando turtle dot() se utiliza para dibujar un punto circular con un tamaño particular en la posición actual, con algo de color.
El comando turtle window_width() se utiliza para devolver el ancho de la ventana actual en píxeles.
El comando turtle fillcolor() se utiliza para devolver o establecer el fillcolor. Si la forma de tortuga es polígono,

El comando turtle end_fill() se utiliza para rellenar la forma dibujada después de la última llamada a begin_fill().

el comando turtle begin_fill() se usa para llamar justo antes de dibujar una forma para rellenar.
El comando Python turtle right() se utiliza para cambiar la dirección de la tortuga por el valor dado del argumento.

El comando turtle left() se utiliza para cambiar la dirección de la tortuga por el valor dado
El comando turtle color() se utiliza para cambiar el color de la tortuga, y el color predeterminado de la tortuga es "negro".
El comando turtle forward() se utiliza para mover a la tortuga hacia adelante por la distancia especificada
El comando turtle down() se utiliza para tirar hacia atrás del lápiz hacia abajo en la pantalla y comienza a dibujar cuando se mueve.
El comando turtle up() detiene todo dibujo. Tira del bolígrafo hacia arriba y no se atraerá nada a la pantalla, hasta que se llame hacia abajo.
El comando turtle shape() se utiliza para establecer la forma de tortuga con un nombre dado

Primero, uno necesita crear la tortuga y luego usar las funcionalidades de posición en la tortuga. La tortuga admite cuatro movimientos básicos: adelante, atrás, izquierda y derecha. La función adelante y atrás necesita la distancia como parámetro, por otro lado, la función izquierda y derecha necesitan un ángulo de giro como parámetro.
Trazado con Turtle

Para hacer uso de los métodos y funcionalidades de las tortugas, necesitamos importar tortugas". turtle" viene empaquetado con el paquete estándar de Python y no necesita ser instalado externamente.

INDICE

Objetivo general

Nuestro objetivo consiste en dibujar diferentes figuras desde algo básico como una línea un circulo una estrella hasta cosas más avanzadas como un cuadro estrellas de colotes etc. de las medidas que consideres oportunas utilizando estructuras de control for,while,entre otras para ahorrar instrucciones. Se valorará el cambio de color en cada trazo usando listas. Para ello fíjate bien en la práctica realizada en muestro proyecto y configúrate una lista con los colores que quieres ir usando, nuestro objetivo principal es que el usuario pueda ir comprendiendo la estructura turtle y ponerla en practicar al mismo aprender por medio de ejemplos presentados en nuestro trabajo, hacer que el usuario interactúe.

Una tortuga creada en la consola o una ventana de visualización (similar a un lienzo) que se utiliza para dibujar, es en realidad un bolígrafo (tipo virtual). La función de tortuga hacia atrás y la tortuga hacia arriba, hacia abajo, hacia la izquierda hacia la derecha y otras funciones se utilizan para mover a la tortuga.

Objetivo especifico

1-puede dibujar figuras intrincadas usando programas que repiten movimientos simples

2-El módulo turtle es una re implementación extendida del mismo módulo de la distribución estándar Python

3-Al combinar estos comandos y otros similares, se pueden dibujar figuras intrincadas y formas

- Marco teórico.

Python 2.0 fue lanzado el 16 de octubre de 2000, con muchas características nuevas importantes. Python 3.0, lanzado el 3 de diciembre de 2008, con muchas de sus principales características transferidas a Python 2.6.x y 2.7.x. Las versiones de Python 3 incluyen la utilidad, que automatiza la traducción del código de Python 2 a Python 3.

Python incluye un módulo llamado "turtle" que permite crear gráficos de tortuga.

```
import turtle
```

Turtle es una característica de Python como un tablero de dibujo, ¡que nos permite ordenar a una tortuga que dibuje sobre ella, En Python es similar porque cuando usamos turtle en el código le decimos que queremos un movimiento o figura. Podemos usar funciones como turtle.forward(…)y turtle.right(…)que pueden mover la tortuga. Turtle es una forma amigable para principiantes de aprender Python ejecutando algunos comandos básicos y viendo cómo la tortuga lo hace gráficamente. Es como un tablero de dibujo que te permite dibujar sobre él. El módulo tortuga se puede utilizar tanto de forma orientada a objetos como orientada a procedimientos.

- **¿Como se Utiliza?**

En primer lugar, deberemos importar el módulo TURTLE y su conjunto de comandos (a modo de librería) con la función:
import turtle
Después es importante escribir la siguiente función para que cada una de las instrucciones del módulo TURTLE no tengamos que escribir la palabra turtle.lafunciónquesea() Sería esta:

t = turtle.Turtle ()

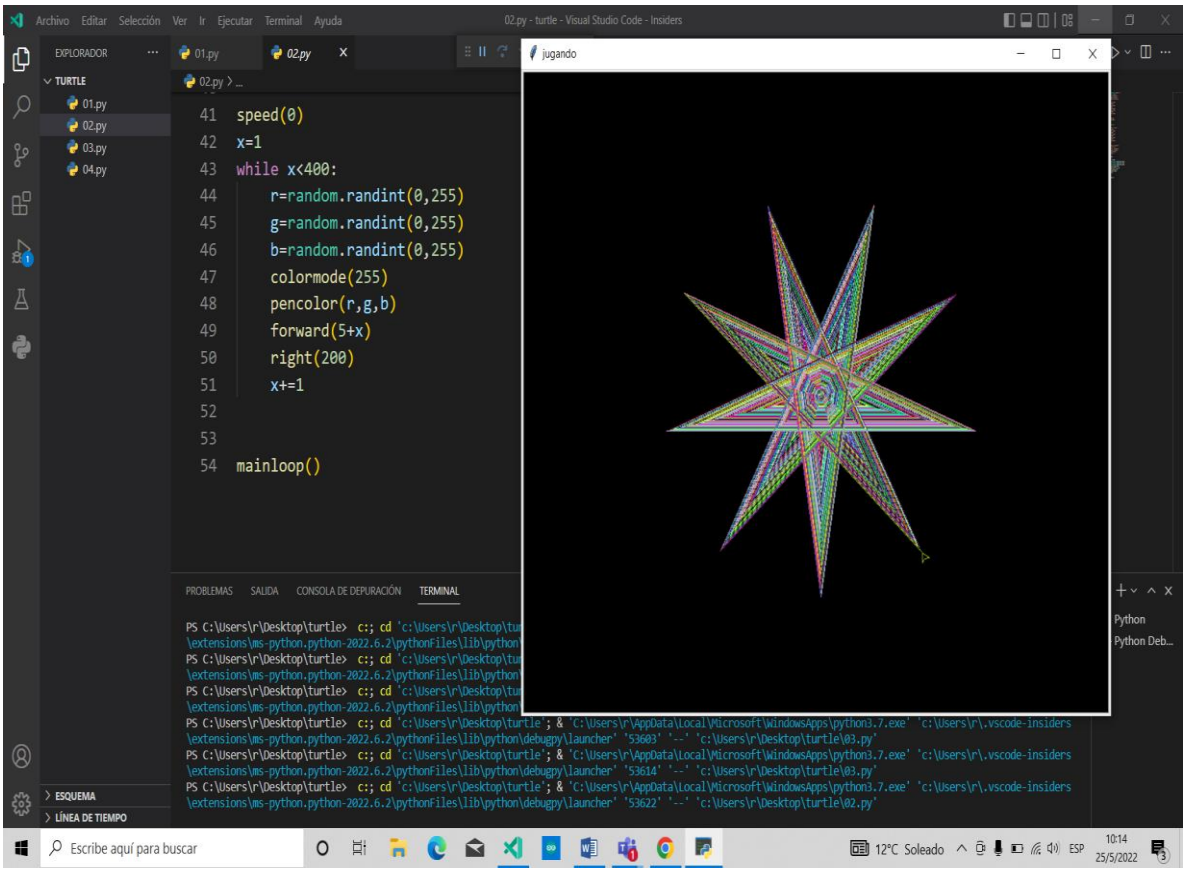
- ¿Para qué nos sirve turtle?

Turtle nos sirve para poder dibujar diferentes figuras usando programas que repiten movimientos simples al combinar comandos y otros similares en este caso puedes dibujar todo lo que este en tu imaginación con los comandos correctos aquí te enseñaremos algunos ejemplos de cómo poder hacer tus mejores dibujos

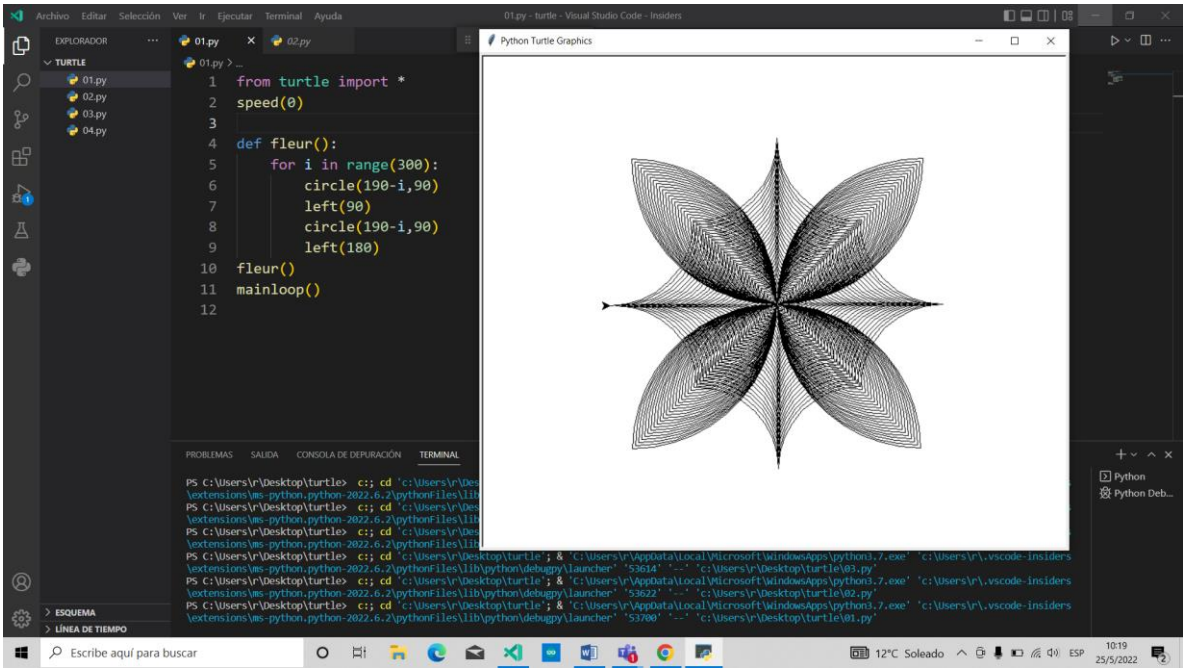
shape(f): Cambia la forma del cursor. f puede ser "arrow", "turtle", "circle", "square", "triangle" o "classic".

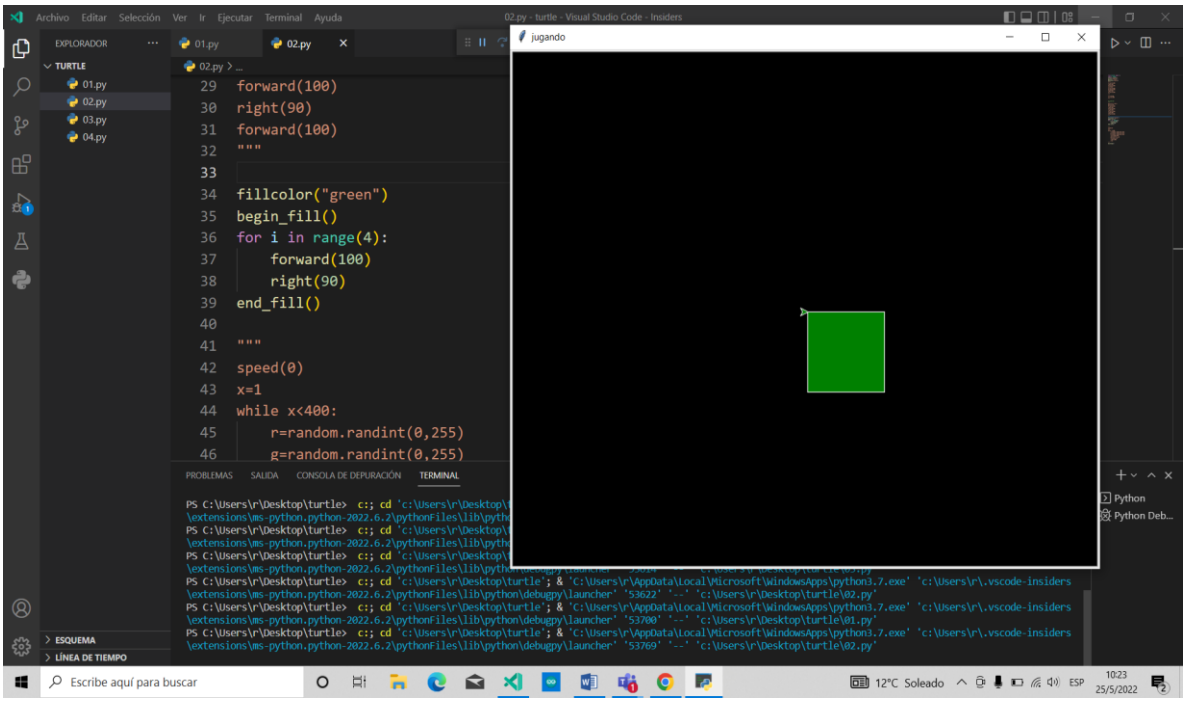
- forward(d): Mueve el cursor hacia delante la distancia d.
- backward(d): Mueve el cursor hacia atrás la distancia d.
- right(a): Gira el cursor hacia la derecha a grados.
- left(a): Gira el cursor hacia la izquierda a grados.
- goto(x,y): Mueve al cursor a la posición x y y.
- up(): Levanta el lápiz de dibujo, si el lápiz está levantado el cursor no dibujará cuando se mueva.
- down(): Baja el lápiz de dibujo, por defecto el lápiz estará bajado. Cuando el lápiz está bajado y movamos el cursor, se dibujará el recorrido.
- color(c): Cambiamos el color con el que el cursor pintará al color c.
- dot(): Dibuja un punto en la posición donde está el cursor.
- fillcolor(c): Cambia el color de relleno de polígonos al color f.
- begin_fill(): Marca el punto inicial para rellenar un polígono de color.
- end_fill(): Marca el punto final para rellenar un polígono con color.
- circle(r,g): Dibuja un círculo de radio r, si se indica g dibujará g grados del círculo, si no se indica nada dibujará el círculo entero.
- pensize(w): Cambia el grosor del lápiz a w puntos.

```python
speed(0)
x=1
while x<400:
    r=random.randint(0,255)
    g=random.randint(0,255)
    b=random.randint(0,255)
    colormode(255)
    pencolor(r,g,b)
    forward(5+x)
    right(200)
    x+=1


mainloop()
```



```python
from turtle import *
speed(0)

def fleur():
    for i in range(300):
        circle(190-i,90)
        left(90)
        circle(190-i,90)
        left(180)
fleur()
mainloop()
```






```
fillcolor("green")
begin_fill()
for i in range(4):
    forward(100)
    right(90)
end_fill()
```

- Personalizaciones que podemos realizar con turtle
    - Color de la línea. Lo configuramos con la siguiente instrucción:
    t.color('blue')
- configurar el icono del puntero y darle diferentes formas con la función, así como esconderlo:
    t.shape(«turtle»)
    Otras formas de puntero: 'arrow', 'circle', 'square', 'triangle', 'classic'
    - configurar la velocidad del trazo con la función
    t.speed(5)
    - configurar el ancho del trazo:
    t.pensize(4)
- cambiar el color de fondo. Primero deberemos programarlo con la siguiente instrucción:
    canvas = turtle.Screen()
    - luego con la instrucción con el color que queramos:
    canvas.bgcolor («purple»)

- Para avanzar usaremos la siguiente función, teniendo en cuenta que las unidades son pixeles:
    t.forward(50)

    también se puede utilizar:

    t.backward(120)

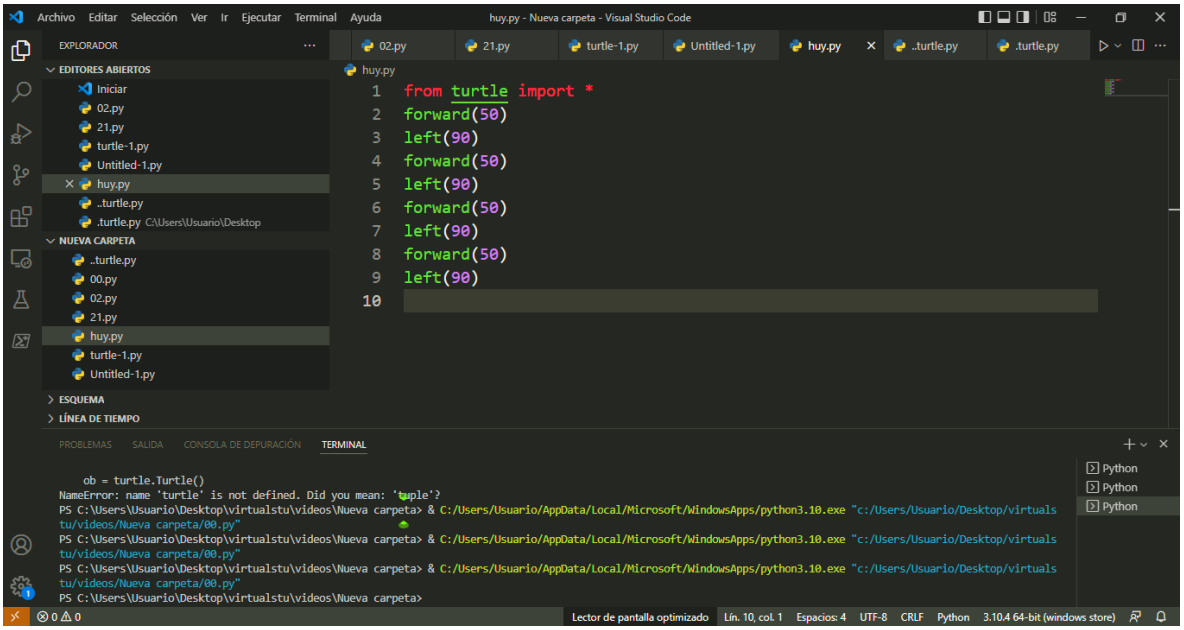
    - Para girar, como os podréis haber imaginado… y con unidades en grados sexagesimales.
    t.left(90)
    t.right(120)

las funciones que nos permiten levantar el puntero del lienzo para desplazarnos a otra parte y continuar dibujando. Aunque el lienzo es infinito es preferible trabajar dentro de las coordenadas del cuadrante visible, el cual tiene unas dimensiones de 600 x 600 pxls.

Ejemplos:

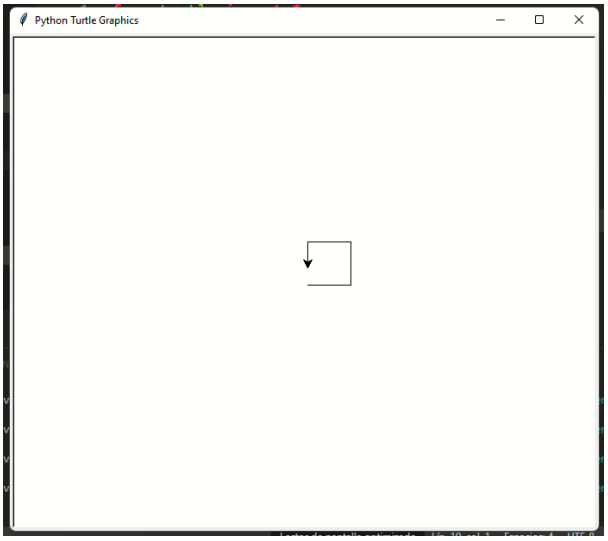
t.penup()
t.goto(-300, -300)
t.pendown()

```
from turtle import *
forward(50)
left(90)
forward(50)
left(90)
forward(50)
left(90)
forward(50)
left(90)
```

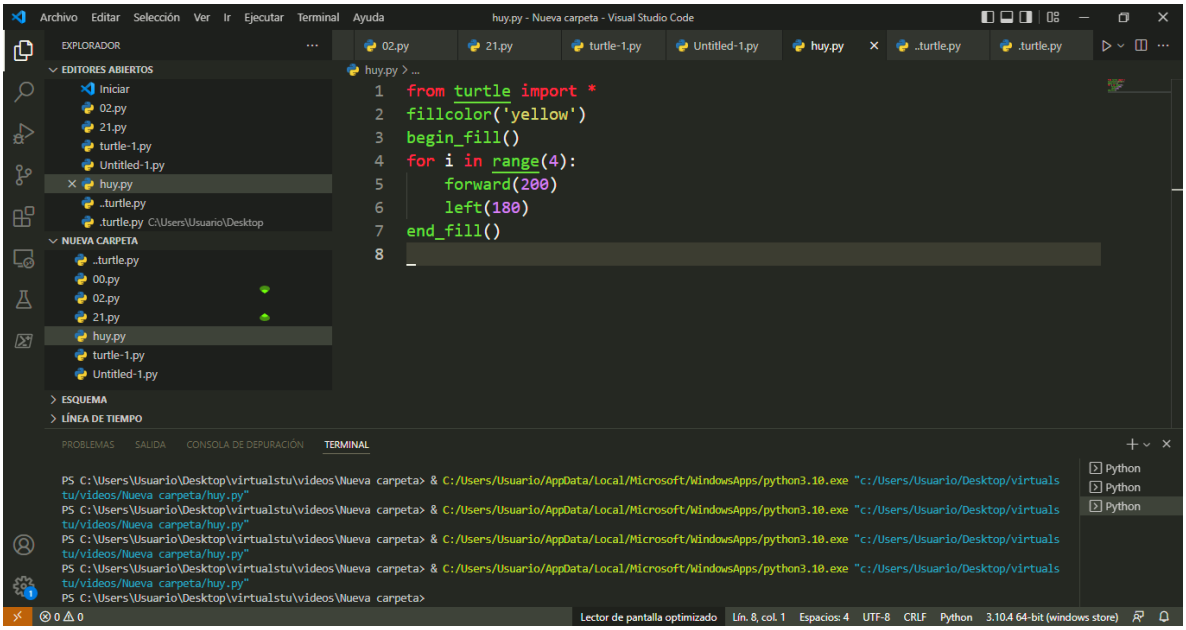


```
from turtle import *
forward(50)
left(90)
forward(50)
left(90)
forward(50)
left(90)
forward(50)
left(90)
```

```
    ob = turtle.Turtle()
NameError: name 'turtle' is not defined. Did you mean: 'tuple'?
PS C:\Users\Usuario\Desktop\virtualstu\videos\Nueva carpeta> & C:/Users/Usuario/AppData/Local/Microsoft/WindowsApps/python3.10.exe "c:/Users/Usuario/Desktop/virtuals
tu/videos/Nueva carpeta/00.py"
PS C:\Users\Usuario\Desktop\virtualstu\videos\Nueva carpeta> & C:/Users/Usuario/AppData/Local/Microsoft/WindowsApps/python3.10.exe "c:/Users/Usuario/Desktop/virtuals
tu/videos/Nueva carpeta/00.py"
PS C:\Users\Usuario\Desktop\virtualstu\videos\Nueva carpeta> & C:/Users/Usuario/AppData/Local/Microsoft/WindowsApps/python3.10.exe "c:/Users/Usuario/Desktop/virtuals
tu/videos/Nueva carpeta/00.py"
PS C:\Users\Usuario\Desktop\virtualstu\videos\Nueva carpeta>
```

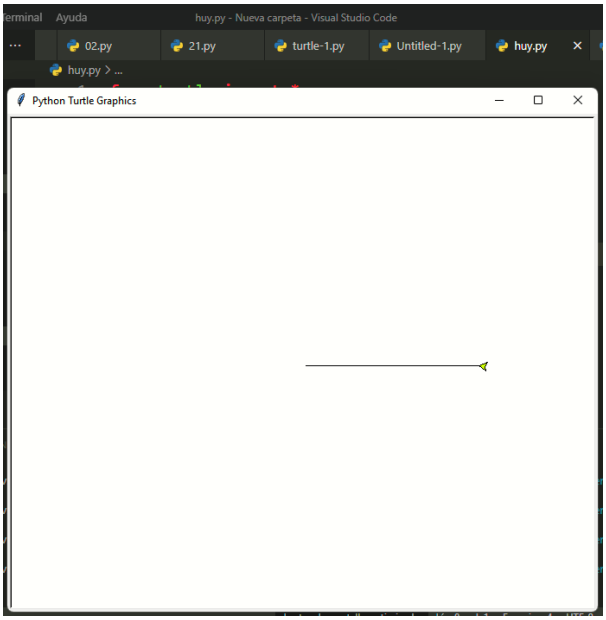


```
from turtle import *
fillcolor('yellow')
begin_fill()
for i in range(4):
   forward(200)
   left(180)
end_fill()
```

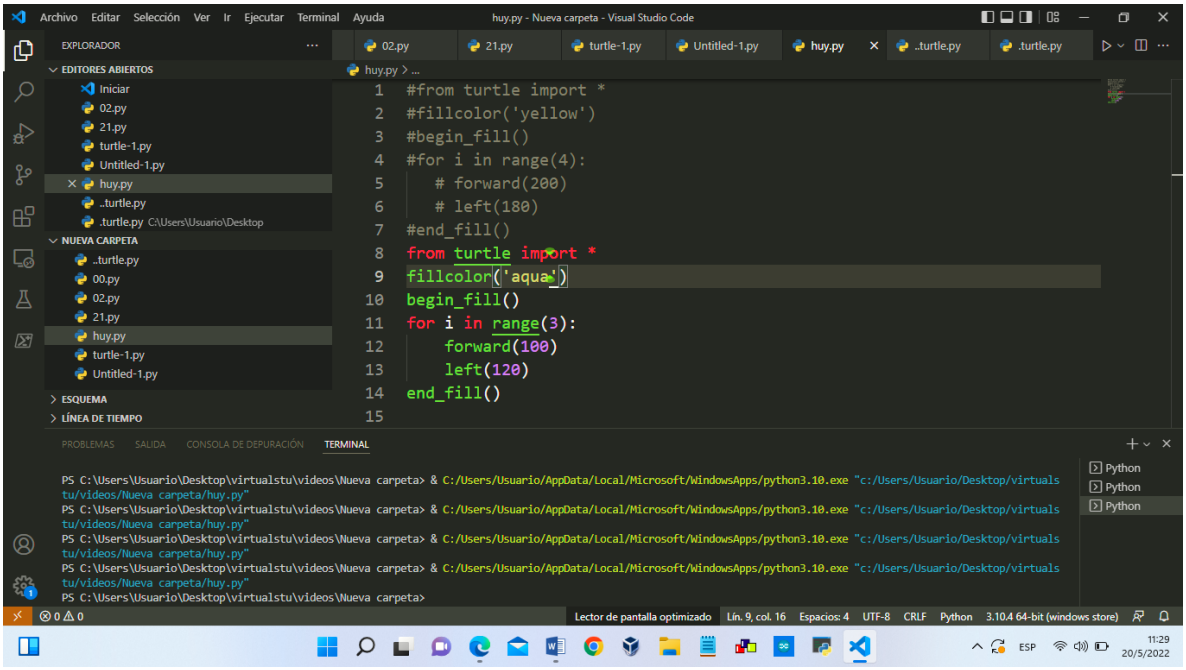


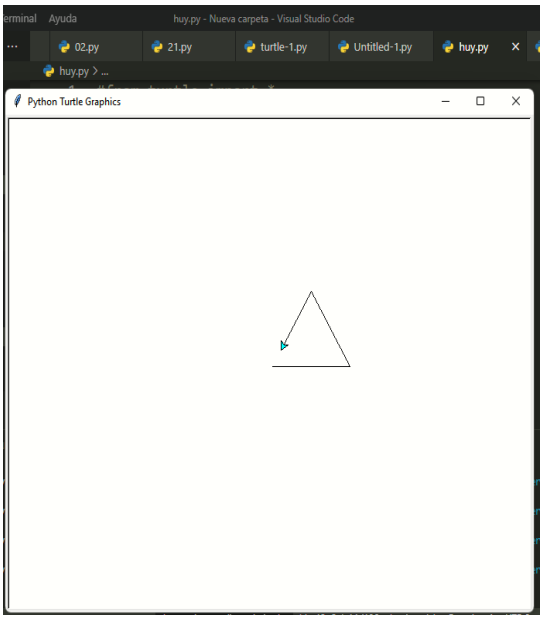
```
from turtle import *
fillcolor('yellow')
begin_fill()
for i in range(4):
    forward(200)
    left(180)
end_fill()
```

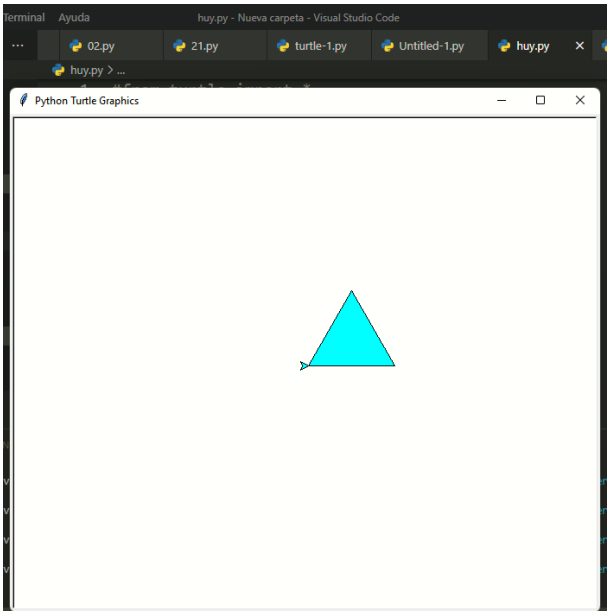
```
PS C:\Users\Usuario\Desktop\virtualstu\videos\Nueva carpeta> & C:/Users/Usuario/AppData/Local/Microsoft/WindowsApps/python3.10.exe "c:/Users/Usuario/Desktop/virtualstu/videos/Nueva carpeta/huy.py"
PS C:\Users\Usuario\Desktop\virtualstu\videos\Nueva carpeta> & C:/Users/Usuario/AppData/Local/Microsoft/WindowsApps/python3.10.exe "c:/Users/Usuario/Desktop/virtualstu/videos/Nueva carpeta/huy.py"
PS C:\Users\Usuario\Desktop\virtualstu\videos\Nueva carpeta> & C:/Users/Usuario/AppData/Local/Microsoft/WindowsApps/python3.10.exe "c:/Users/Usuario/Desktop/virtualstu/videos/Nueva carpeta/huy.py"
PS C:\Users\Usuario\Desktop\virtualstu\videos\Nueva carpeta> & C:/Users/Usuario/AppData/Local/Microsoft/WindowsApps/python3.10.exe "c:/Users/Usuario/Desktop/virtualstu/videos/Nueva carpeta/huy.py"
PS C:\Users\Usuario\Desktop\virtualstu\videos\Nueva carpeta>
```



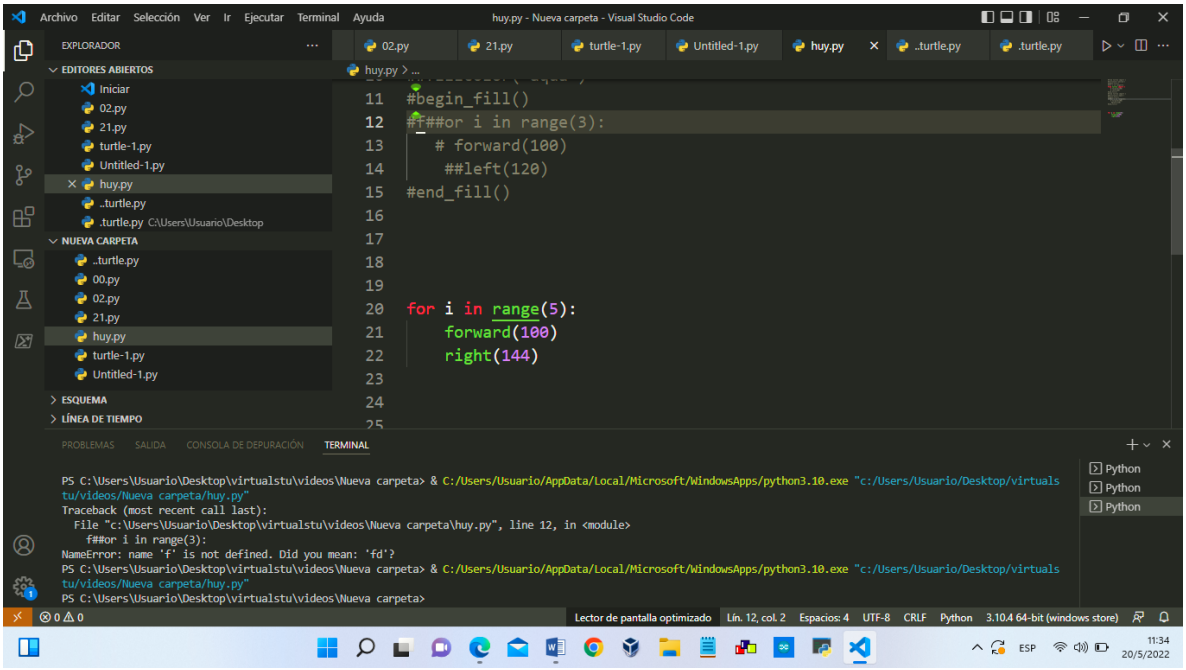
```
from turtle import *
fillcolor('aqua')
begin_fill()
for i in range(3):
    forward(100)
    left(120)
```

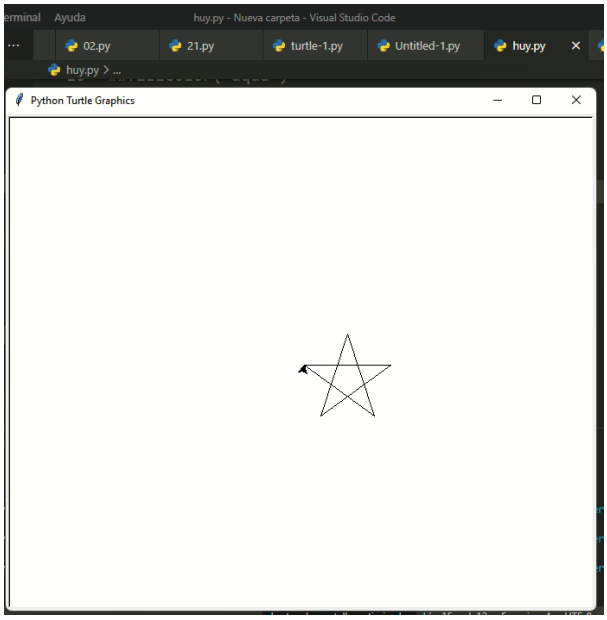




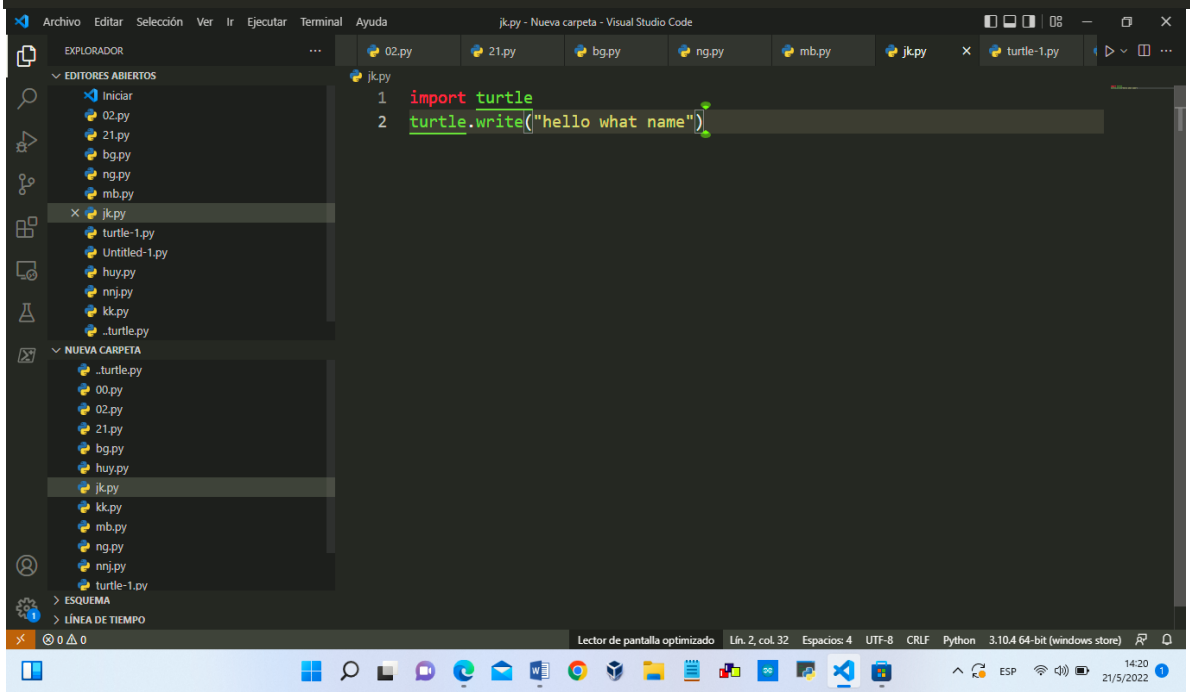


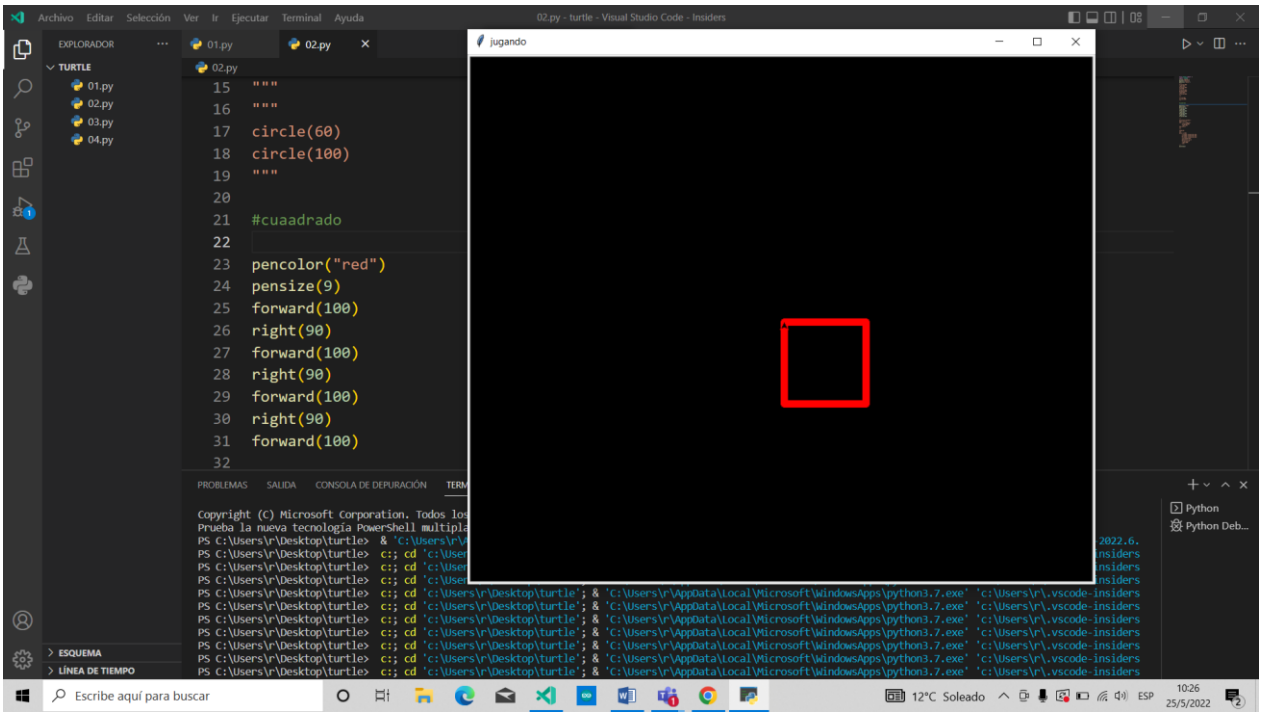
```
for i in range(5):
    forward(100)
    right(144)
```



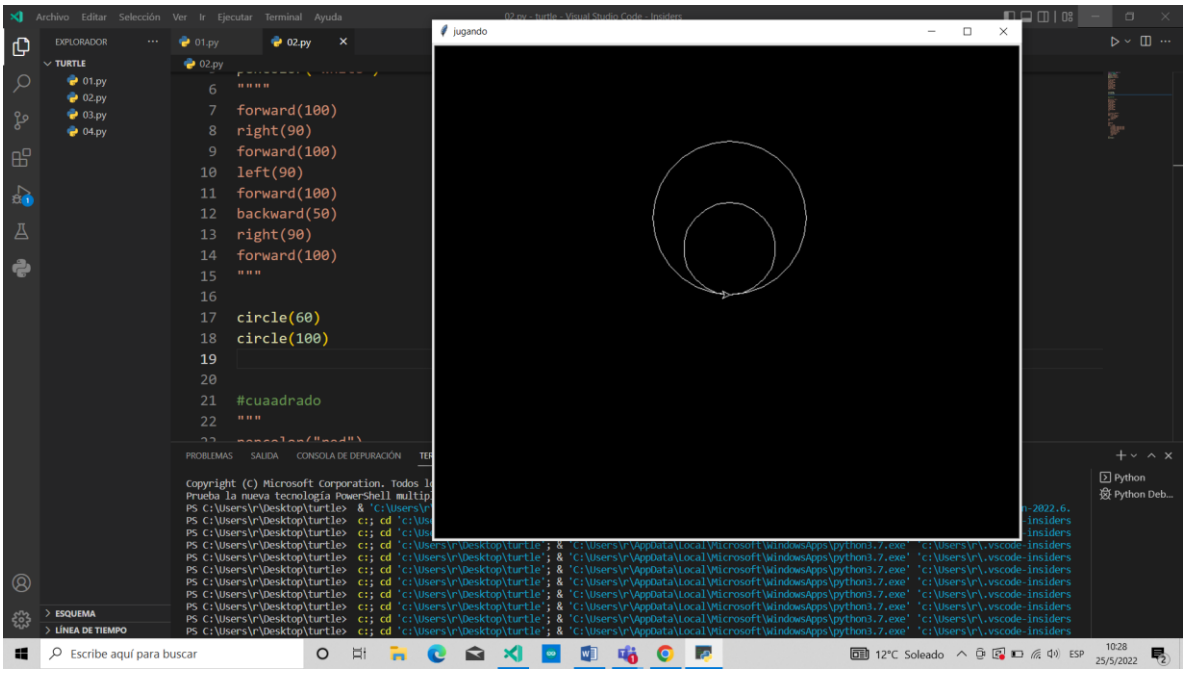


```
import turtle
turtle.write("hello what name")
```

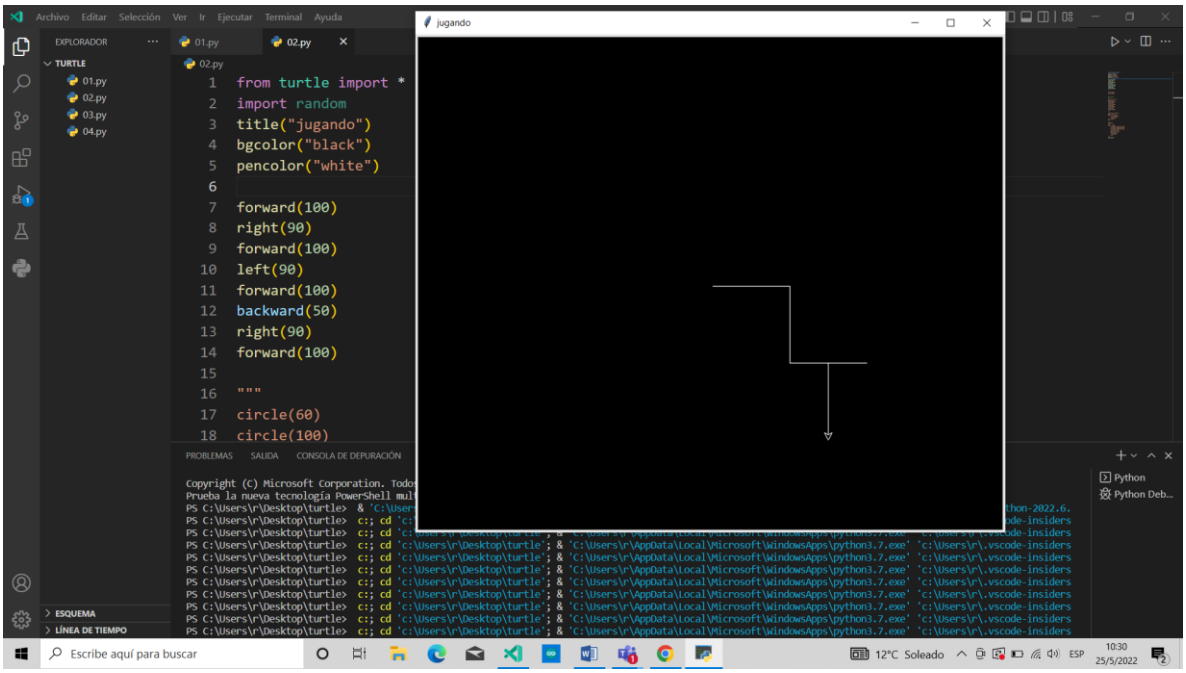





```
#cuaadrado

pencolor("red")
pensize(9)
forward(100)
right(90)
forward(100)
right(90)
forward(100)
right(90)
forward(100)
```




```
circle(60)
circle(100)
```




```
01.py    02.py
02.py
1   from turtle import *
2   import random
3   title("jugando")
4   bgcolor("black")
5   pencolor("white")
6
7   forward(100)
8   right(90)
9   forward(100)
10  left(90)
11  forward(100)
12  backward(50)
13  right(90)
14  forward(100)
```

- ## Recomendaciones

Pero hay que tener en cuenta que para crear este tipo de figuras deberás tener instalados Python o visual estudio code y leer diferentes tipos de códigos para saber cómo crear una de tus figuras hay que tener conocimiento sobre lo que nosotros queremos crear en este tipo de programas

Como recomendaciones y nosotros hemos leído varios manuales sobre turtle hemos visto tutoriales en YouTube sobre cómo podemos no ser este tipo de programa hemos visto diferentes tipos de manuales y programas desde los más básico hasta la más complicado en general nosotros ahorita hemos aprendido solo lo más básico sobre turtle aunque hay gráficos del programa muy bonitos muy extensos pero todavía no nos hemos aprendido este tipo de programación porque apenas estamos aprendiendo de qué se trata turtle

Con este programa hemos aprendido a cómo dibujar una estrella hacer un cuadro y hacer líneas rectas, aunque no es un gran comando, pero estamos aprendiendo sobre este tema que seguimos leyendo sobre cómo utilizar cada 1 de los comandos.

Hay comandos que con solo cambiar el ángulo ya cambia su función es decir puede dibujar una figura distinta partiendo del mismo comando utilizado.

Por lo cual nuestras recomendaciones son aprender se los comandos básicos y luego pasar a utilizar a usar unos más avanzados.

- ## Conclusión

- En nuestro trabajo describimos turtle y su importancia en Python, ya que vemos que hoy en día los programadores y los estudiantes están trabajando con Python por sus herramientas y funciones, Es un lenguaje de programación intérprete desarrollado como un proyecto de código abierto. Compatibilidad con Python programación orientada a objetos, programación procedimental y también funcional un beneficio de Python es que hay numeras bibliotecas disponibles y gratis, las bibliotecas más importantes puede ser Tkinter biblioteca gráfica, el Aleatorio para generar valores aleatorios, y el biblioteca de gráficos de tortuga llamada Tortuga Python es uno de los lenguajes de programación más fáciles y populares para que los estudiante

- ## Bibliografía

clic, p. (25 de May de Python Basic Library Turtle). Tutorial sencillo de Python-Turtle. Obtenido de Un breve resumen de las funciones de uso común: https://programmerclick.com/article/86981614026/

principiantes), L. t. (18 de Aug de 2015 » python). Programación desde cero Recursos educativos sobre programación y temas relacionados. Obtenido de Algunas órdenes básicas de Turtle » python: http://patriciaemiguel.com/python/2015/08/18/tortuga-python-principiantes.html

Wally Feurzeig, S. P. (1967.). turtle — Gráficos con Turtle. Obtenido de turtle — Gráficos con Turtle: https://docs.python.org/es/dev/library/turtle.html#introduction