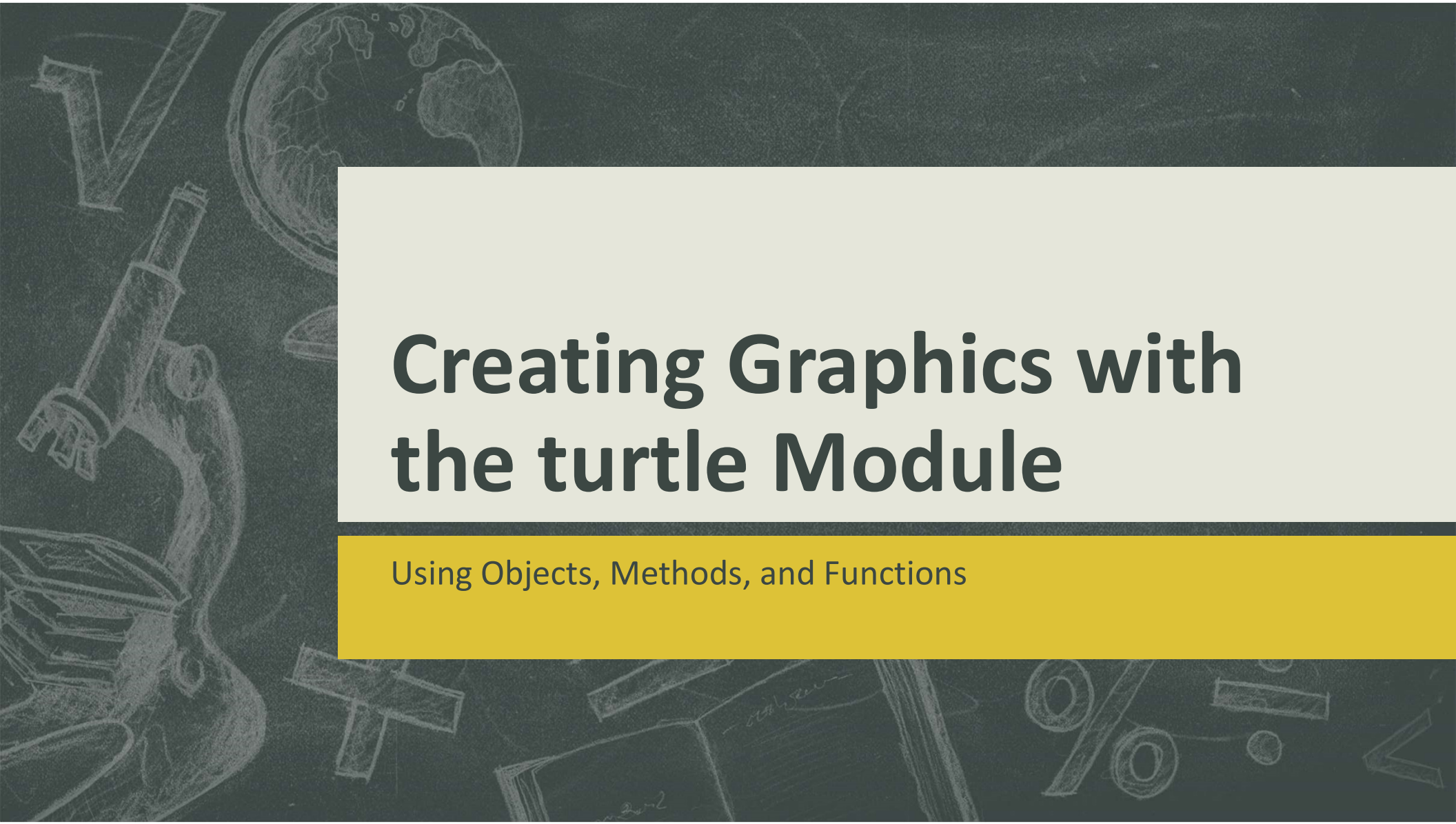


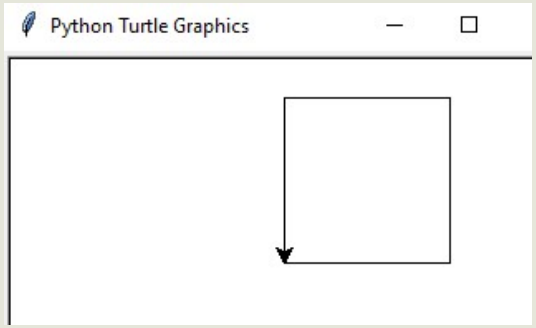
Creating Graphics with the turtle Module

Using Objects, Methods, and Functions

Using turtle module Functions

```
from turtle import forward, left
forward(100)
left(90)
forward(100)
left(90)
forward(100)
left(90)
forward(100)
```

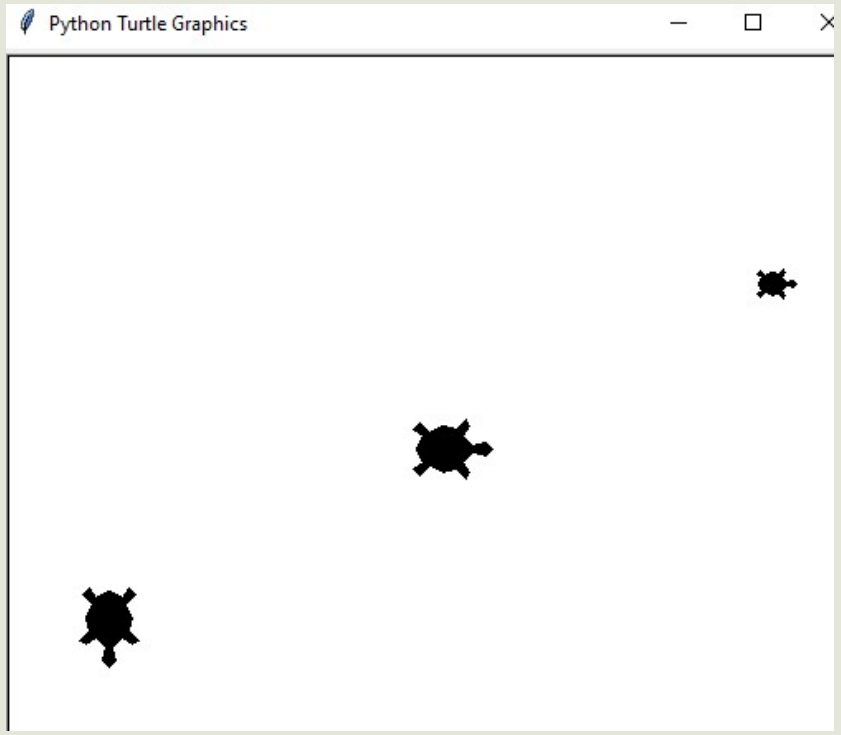
Output:



Using turtle module Functions – Importing “star”

```
from turtle import *
shape('turtle')
penup()
goto(200, 100)
stamp()
turtlesize(2)
goto(-200, -100)
setheading(270)
stamp()
goto(0, 0)
setheading(0)
```

Output:



Classes and Methods

- Classes are special groups of code to bind together properties and functions into a single *object*.
- Functions contained in classes are called *methods*.
- The turtle module contains a Turtle class. Turtle objects have all same functions as the *generic* turtle functions, but now they are methods.
- Using Turtle objects allows us greater control over our programs and Turtles.
- Like other classes, notice that the Turtle class is spelled with a CAPTIAL T

Classes and Methods

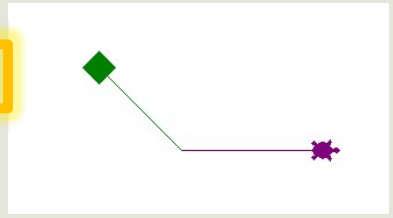
```
from turtle import Turtle

hank = Turtle()
hank.shape('turtle')
hank.color('purple')
hank.forward(120)

jill = Turtle()
jill.shape('square')
jill.color('green')
jill.left(135)
jill.forward(100)
```

All we need is to import the Turtle class from turtle module

Create an object using the *constructor* method.

Methods are called *on* the object using the dot operator.

Output:



Classes and Properties

- Objects have properties and methods that are unique to themselves.
- The Turtle class has many methods and properties to access.

```
print(hank.xcor(), hank.ycor())
print(hank.isdown())
hank.penup()
print(hank.isdown())
```

Output:
120.0 0.0
True
False

Classes and Properties

- Some classes, including the Turtle class, allow you to assign additional properties to an object from the Class.
- Objects made by a class are called an *instance of the class*.

```
hank = Turtle()
hank.shape('turtle')
hank.name = 'Hank'
hank.is_turtle = True
print(hank.name, hank.is_turtle)

jill = Turtle()
jill.shape('square')
jill.name = 'Jill'
jill.is_turtle = False
print(jill.name, jill.is_turtle)
```

Here we add two additional properties to our Turtle instance: name and is_turtle

Output:
Hank True
Jill False

Methods vs Functions

Functions

- Perform actions on arguments.
- Examples:
  - The max function returns the bigger of two number arguments.

```
>>> max(1, 3)
3
```

  - The forward function moves a *default* turtle forward the amount given in the argument it cannot move a specific turtle

```
>>> forward(100)
```

Methods

- Perform actions on a target object.
- Result depends on status of the object the method is called on.
- Example: After creating 2 Turtle objects named hank and jill, the forward method called on hank moves only the hank object forward.

```
hank = Turtle()
jill = Turtle()
hank.forward(30)
```

Common Turtle Methods/Functions

showturtle()	Makes the turtle visible.
hideturtle()	Makes the turtle invisible.
forward(distance)	Moves the turtle forward the amount entered as distance.
right(angle)	Turns the turtle right the amount of degrees entered.
left(angle)	Turns the turtle left the amount of degrees entered
goto(x, y)	Moves the turtle to the point x, y. 0, 0 is the origin located in the center of the graphics window.
setx(x)	Moves the turtle to the x-coordinate x. The y position is not changed.
sety(y)	Moves the turtle to the y-coordinate y. The x position is not changed.
circle(radius)	The turtle moves in a circle with radius provided.
dot()	Makes a dot in the current location of the turtle.
stamp()	Stamps an image of the turtle at the current location
speed(setting)	Changes the turtle speed. Setting can be an integer from 1 – 10. 1 is slowest and 10 is faster. The fastest setting is reserved for the value 0.
pendown()	Puts the turtle's pen down - begin writing.
penup()	Picks up the turtle's pen – stop writing.
color(color)	Changes the color of the turtle. Acceptable color parameters are found at: https://www.tcl.tk/man/tcl8.4/TkCmd/colors.htm
begin_fill()	Enter fill mode.
end_fill()	Exit fill mode.
shape(name)	Change the shape of the turtle. Choose from "arrow", "turtle", "circle", "square", "triangle", or "classic".
clear()	Clears the turtle's drawings from the screen.
turtlesize(size)	Sets the size of the turtle to the integer value size.
setheading(heading)	Points the turtle in the direction heading, where 0 is facing right, 90 is up, 180 left, 270 down.
write(text)	Write the text to the screen. The text must be entered inside quotes. Words in quotes are called strings.

Making an Animation

- Check out turtle_animation.py for examples on how to tell a story using Turtle objects
- Then, try to make your own!