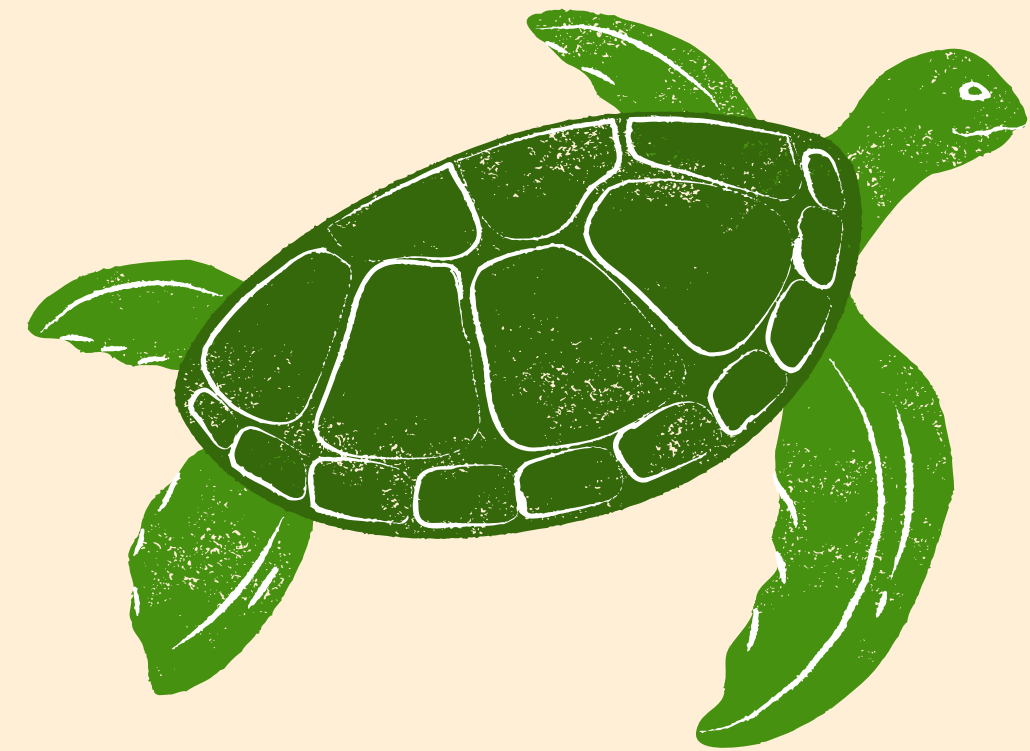


Intro to working with Objects in Python



Understanding the basics of Objects

What is Turtle?

Turtle is a Python Module, Which allows us to draw and create art



Why are we doing this Module?

Although this seems like something for kids, **Turtle Module is simply a hidden gem** for learning OOP.

This Module will give you a **strong understanding of how Objects work** in Programming and how we can use them throughout our code

What is Turtle?



**We have 9 objects, all
of which are Turtles**

Every Turtle has its own properties, such as
Color or Speed.

They all have **different actions** that can do
different tasks

**What are Methods &
Properties?**

Methods and Properties:

A **Method** is a Function, specifically it is a **function in a Class**

A **Property** is a Variable, specifically it is a **variable in a Class**







Properties (Variables)	Methods (Functions)
owl.color = "maroon"	owl.fly()
parrot.speed = 5	parrot.eat()
canary.width = 2	canary.speak()

A Property & Method **must be linked** to an object to work

Methods and Properties:

A **Method** is a Function, specifically it is a **function in a Class**

A **Property** is a Variable, specifically it is a **variable in a Class**

Properties (Variables)	Methods (Functions)
owl.color = "maroon" 	owl.fly() 
parrot.speed = 5 	parrot.eat() 
canary.width = 2 	canary.speak() 

A Property & Method **must be linked** to an object to work

Setting up your code:

```
from turtle import *
```

```
t1 = Turtle()
```

```
t2 = Turtle()
```

```
t1.color("orange")
```

```
t2.color("purple")
```

```
t1.forward(50)
```

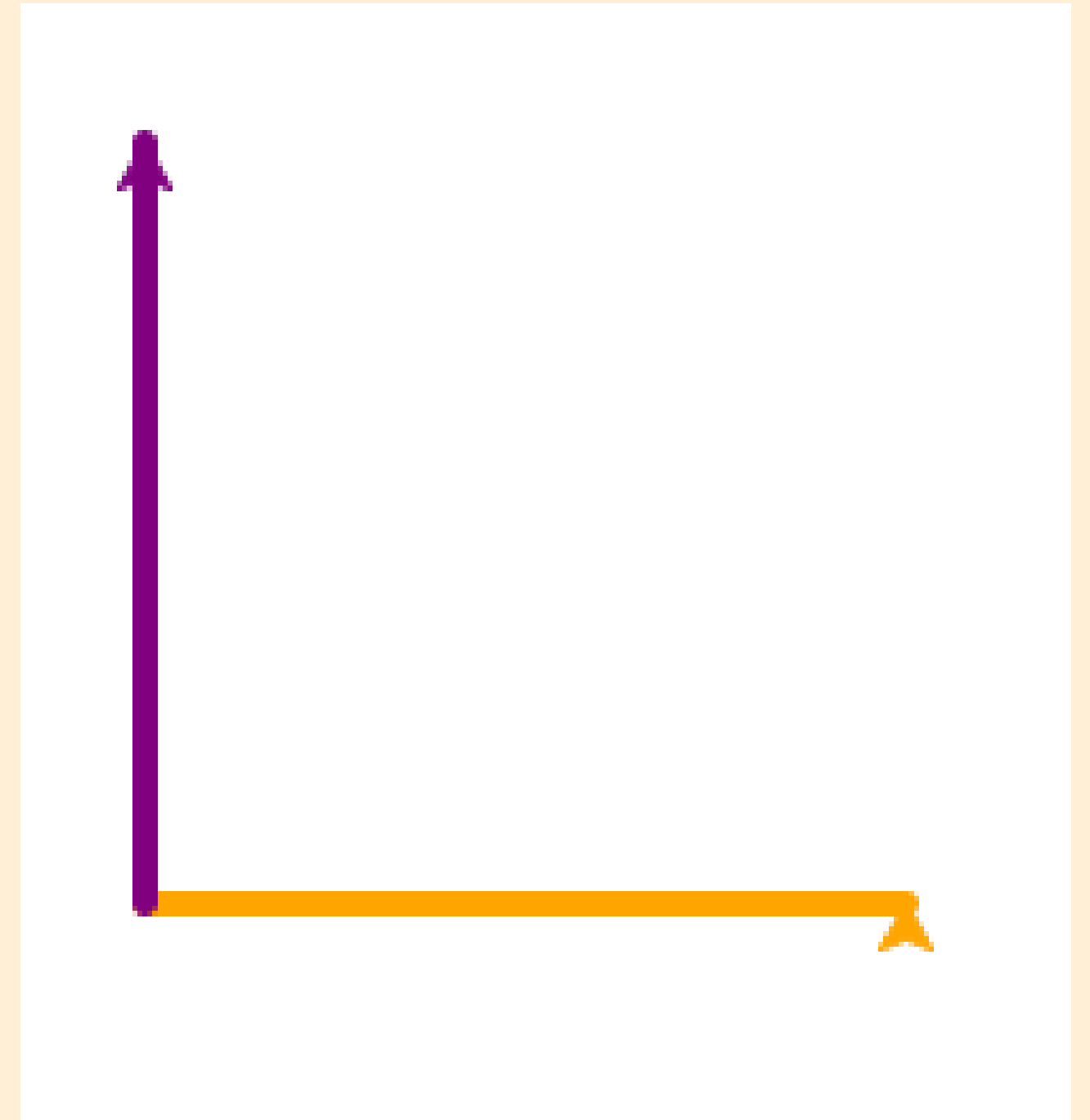
```
t1.left(90)
```

```
t2.left(90)
```

```
t2.forward(50)
```

```
done()
```

Final Output



Setting up your code:

```
from turtle import *
```

Importing everything from
the Turtle Module

```
t1 = Turtle()  
t2 = Turtle()
```

Creating two Objects

```
t1.color("orange")  
t2.color("purple")
```

Assigning the a Color
Property to each object

```
t1.forward(50)  
t1.left(90)
```

Linking two methods to
object one to move

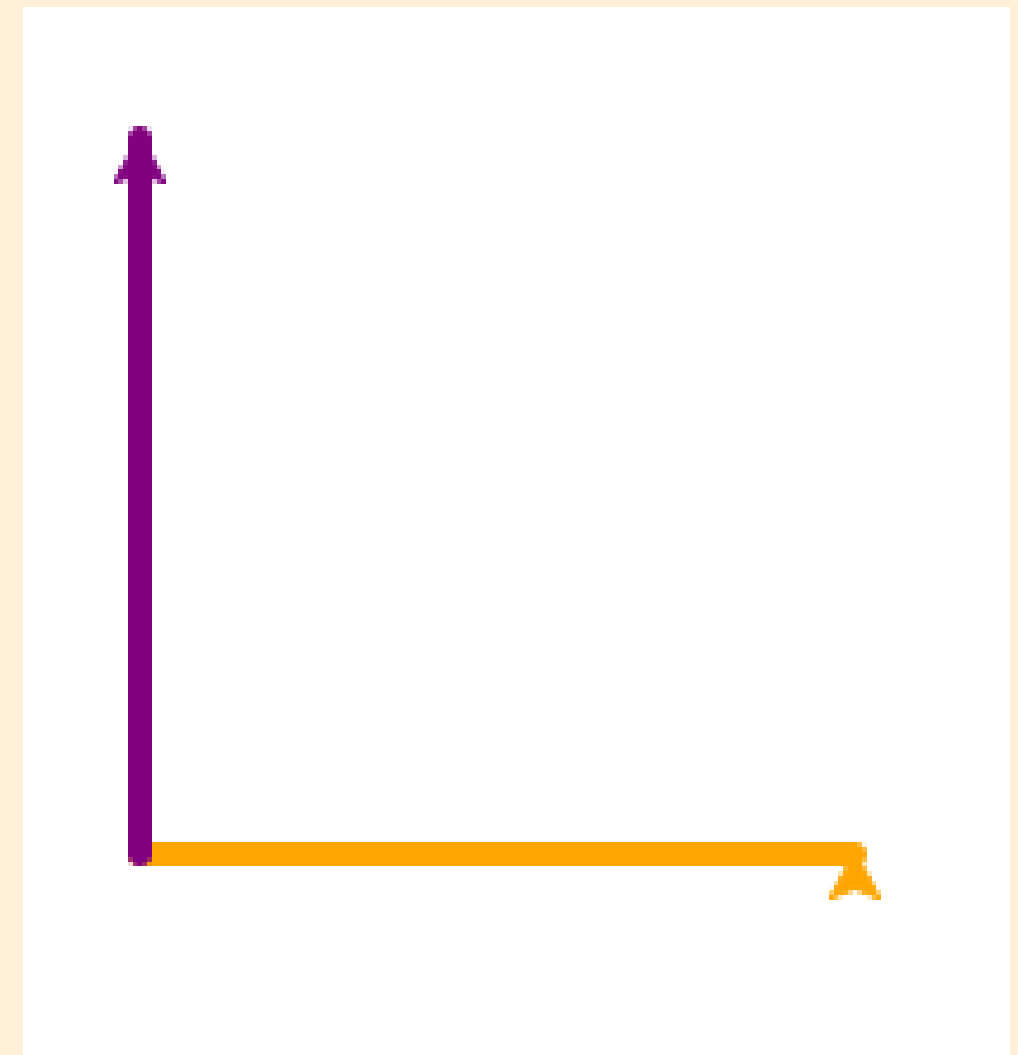
```
t2.left(90)  
t2.forward(50)
```

Linking two methods to
object two to move

```
done()
```

Allows the drawing to stay
on the screen once done

Final Output



Methods and properties of the Turtle Module:

Property / Method	What they do
<code>.color("green")</code>	Assigns a Color to your object, the color is a string
<code>.width(5)</code>	Assigns a width to the Line
<code>.shape("circle")</code>	Assigns a shape to the object (arrow, circle, square, triangle, turtle)
<code>.speed(10)</code>	Assigns a Speed to the object
<code>.fd(100)</code>	A method to move your object forward a number of pixels
<code>.left(90)</code> or <code>.right(45)</code>	A method to turn your object , a number of degrees

Methods and properties of the Turtle Module:

Property / Method	What they do
.up()	Lifts your object off the screen
.goto(x, y)	Moves your object to a new location (x , y)
.down()	Places your object back down once moved
.circle(10)	Used to draw a circle (takes a radius)
.begin_fill() or .end_fill()	Allows you to fill in your drawing with a solid color
.done()	Used to keep the drawing displayed once complete

Use Turtle with the basics of Python:

```
from turtle import *
```

```
t1 = Turtle()  
t1.color("orange")  
t1.width(5)
```

```
t1.begin_fill()  
for i in range(5):  
    t1.forward(150)  
    t1.left(144)  
t1.end_fill()
```

```
done()
```

You can use your previous knowledge to expand with turtle, this includes:

- Using **For Loops** to repeat
- Using **Conditional Statements**
- **Creating Functions**



Use Turtle with the basics of Python:

```
from turtle import *
```

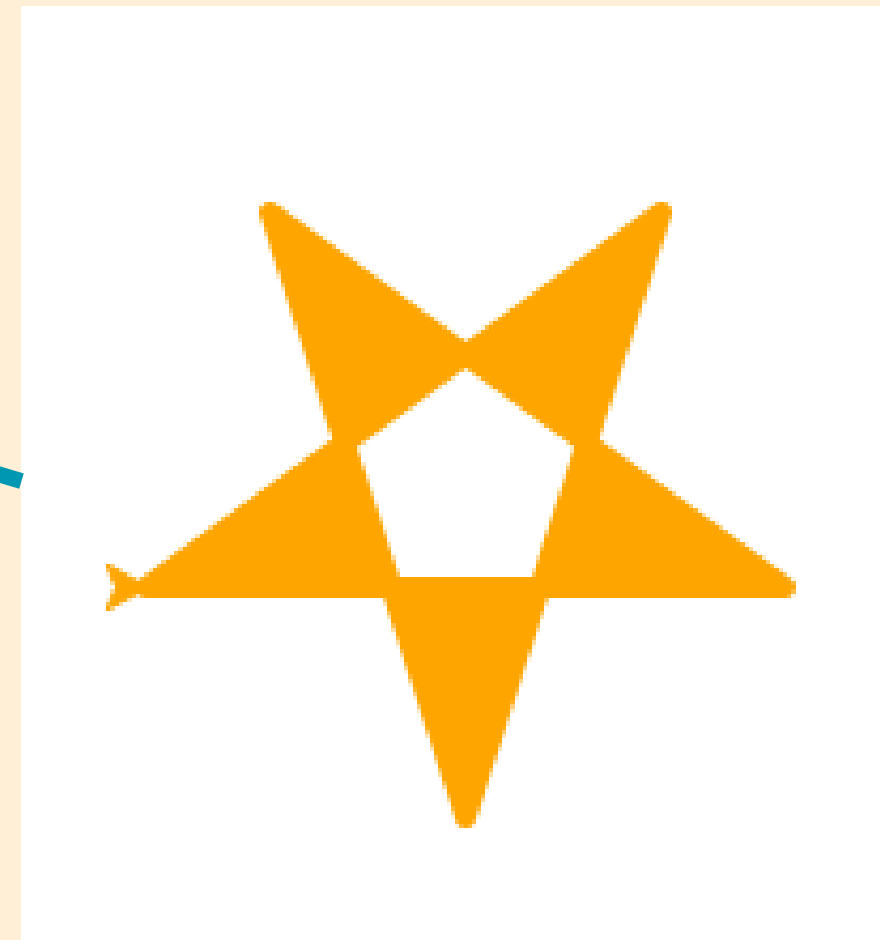
```
t1 = Turtle()  
t1.color("orange")  
t1.width(5)
```

```
t1.begin_fill()  
for i in range(5):  
    t1.forward(150)  
    t1.left(144)  
t1.end_fill()
```

```
done()
```

You can use your previous knowledge to expand with turtle, this includes:

- Using **For Loops** to repeat
- Using **Conditional Statements**
- **Creating Functions**



Additional Code Examples:

```
from turtle import *
```

```
def star(t,width,size,color):
```

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

```
    t.width(width)
```

```
    t.begin_fill()
```

```
    for i in range(5):
```

```
        t.forward(size)
```

```
        t.left(144)
```

```
    t.end_fill()
```

```
def circle(t,radius,color):
```

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

```
    t.begin_fill()
```

```
    t.circle(radius)
```

```
    t.end_fill()
```

```
t = Turtle()
```

```
ask = input("Enter shape: ")
```

```
while ask != "done":
```

```
    if ask == 'star':
```

```
        width = int(input("Enter width: "))
```

```
        col = input("Enter a color: ")
```

```
        size = int(input("Enter a length: "))
```

```
        star(t,width,size,col)
```

```
    elif ask == 'circle':
```

```
        radius = int(input("Enter a radius: "))
```

```
        col = input("Enter a color: ")
```

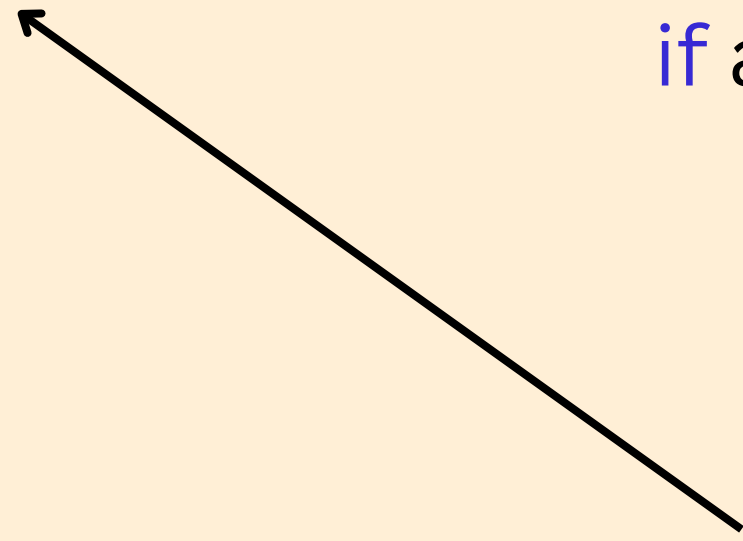
```
        circle(t,radius,col)
```

```
    else:
```

```
        print("No shape entered")
```

```
    ask = input("Enter shape: ")
```

```
done()
```



Additional Code Examples:

```
from turtle import *
```

```
def star(t,width,size,color):
```

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

```
    t.width(width)
```

```
    t.begin_fill()
```

```
    for i in range(5):
```

```
        t.forward(size)
```

```
        t.left(144)
```

```
    t.end_fill()
```

```
def circle(t,radius,color):
```

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

```
    t.begin_fill()
```

```
    t.circle(radius)
```

```
    t.end_fill()
```

```
t = Turtle()
```

```
ask = input("Enter shape: ")
```

```
while ask != "done":
```

```
    if ask == 'star':
```

```
        width = int(input("Enter width: "))
```

```
        col = input("Enter a color: ")
```

```
        size = int(input("Enter a length: "))
```

```
        star(t,width,size,col)
```

```
    elif ask == 'circle':
```

```
        radius = int(input("Enter a radius: "))
```

```
        col = input("Enter a color: ")
```

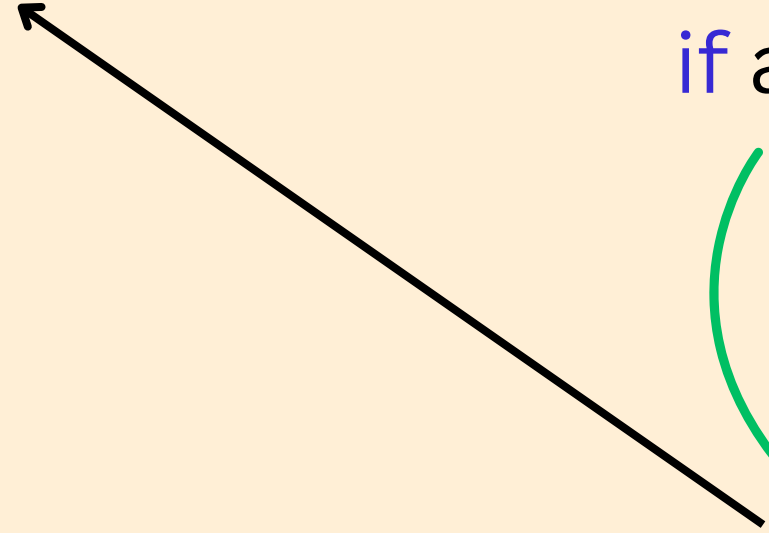
```
        circle(t,radius,col)
```

```
    else:
```

```
        print("No shape entered")
```

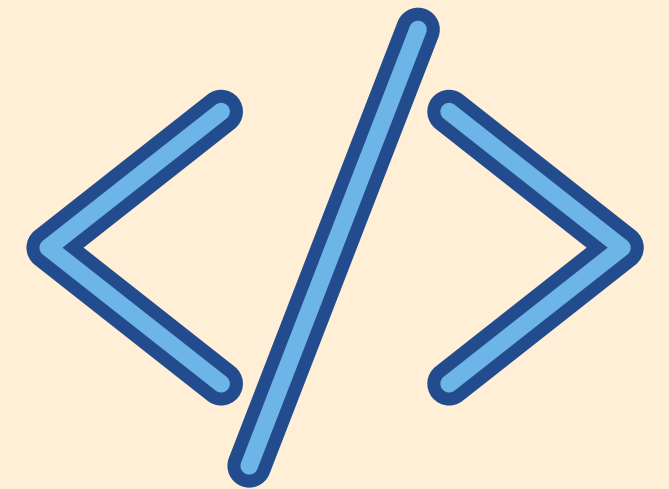
```
    ask = input("Enter shape: ")
```

```
done()
```



Code Challenges:

1. Draw a hollow pentagon, each side a different color
2. Draw 4 solid squares all different colors in different locations
3. Create 3 functions that'll draw something different when called (similar to the previous example)
4. Draw 10 circles, each at a **random** location with a **random** radius
(hint -> Use the Python random module, randint())
5. Create 3 objects, that'll draw 3 triangles in a column, all a different color



[Turtle Graphics Module Documentation](#)