**Worksheet: Introduction to Turtle Graphics in Python**

**Objective**: Introduce students to the turtle module, enabling them to create basic shapes and understand how turtle graphics work in Python.

**Estimated Time**: 1 hour

 **What is Turtle Graphics?**

* The turtle module provides an environment for creating drawings with a "turtle" that moves on the screen.
* The turtle can move in different directions and draw shapes using various commands.

 **Basic Setup**:

import turtle

# Create a turtle object

t = turtle.Turtle()

# Move the turtle forward

t.forward(100)

# Close the drawing window

turtle.done()

**Activity 1: Move and Turn the Turtle**:

* Move the turtle forward and backward using t.forward() and t.backward().
* Turn the turtle left and right using t.left() and t.right().

**Challenge**: Write a script that moves the turtle in a square by combining forward movement and right turns.

**Part 2: Drawing Shapes (20 minutes)**

1. **Activity 2: Drawing a Triangle**:
   * Using what you’ve learned, write code to draw a triangle.
   * **Hint**: A triangle has three sides, and the sum of the angles in any triangle is 180 degrees.

**Activity 3: Drawing a Circle**:

* Use the t.circle(radius) function to draw a circle.

import turtle

t = turtle.Turtle()

t.circle(50) # Draw a circle with a radius of 50

turtle.done()

**Changing the Turtle’s Color and Pen Size**:

* Set the turtle’s pen color and pen size using the t.pencolor() and t.pensize() functions.

import turtle

t = turtle.Turtle()

t.pencolor("blue") # Change pen color to blue

t.pensize(3) # Change pen size to 3

for \_ in range(4):

t.forward(100)

t.right(90)

turtle.done()

**Changing the Turtle’s Speed**:

* You can adjust the speed of the turtle’s movement using t.speed(). The range is 1 (slow) to 10 (fast).

t.speed(5) # Medium speed

1. **Activity 4: Customize Your Drawing**:
   * Draw a shape of your choice, changing the pen color, size, and speed.

**Challenge**: Try drawing a star or a polygon (e.g., a pentagon or hexagon).

**Part 4: Turtle Functions and Loops (10 minutes)**

1. **Creating Functions to Reuse Code**:
   * You can wrap turtle commands inside functions for reusable code.

**Example**: Function to draw a square:

import turtle

def draw\_square():

for \_ in range(4):

t.forward(100)

t.right(90)

t = turtle.Turtle()

draw\_square()

turtle.done()

**Activity 5: Draw Multiple Shapes Using Loops**:

* Use loops to create multiple shapes without rewriting the drawing code.

**Challenge**: Write code to draw five squares, each rotated slightly, creating a cool pattern.

import turtle

t = turtle.Turtle()

for \_ in range(5):

for \_ in range(4):

t.forward(100)

t.right(90)

t.right(72) # Rotate for the next square

turtle.done()

**Reflection (10 minutes)**

* **What did you learn about using loops to create patterns in turtle graphics?**
* **What are the advantages of using functions to organize your turtle drawing code?**