TurtleSim Python Practice Worksheet

This notebook will guide you through various exercises to control a turtle using ROS2 commands.

In [ ]:
```python
import rclpy
import sys
import os


from turtlemove import TurtleController  # Import TurtleController class
import time

# Initialize ROS2 node
rclpy.init()
turtle = TurtleController()

# Function to send movement commands to the turtle
def send_command(turtle, linear, angular, duration):
    """Helper function to move the turtle."""
    turtle.move_turtle(linear, angular, duration)
```

Task 1: Move Backward

Make the turtle move backward at 1.5 m/s for 4 seconds. Fill in the missing parameters.

In [ ]:
```python
# send_command(turtle, ?, ?, ?)
```

Task 2: Rotate Clockwise

Make the turtle rotate clockwise at 2 rad/s for 3 seconds.

In [ ]:
```python
# send_command(turtle, ?, ?, ?)
```

Task 3: Move in a Circle

Make the turtle move in a circle with a linear velocity of 1 m/s and an angular velocity of 0.5 rad/s for 6 seconds.

In [ ]:
```
# send_command(turtle, ?, ?, ?)
```

Task 4: Move in a Square Pattern

Complete the loop below to move the turtle in a square pattern.

In [ ]:
```
# for _ in range(4):
#     send_command(turtle, ?, ?, ?)
#     send_command(turtle, ?, ?, ?)
```

Task 5: Zigzag Pattern

Complete the loop to make the turtle move in a zigzag pattern.

In [ ]:
```
# for _ in range(4):
#     send_command(turtle, ?, ?, ?)
#     send_command(turtle, ?, ?, ?)
```

Task 6: Triangle Pattern

Complete the loop to make the turtle move in a triangular pattern.

In [ ]:
```
# for _ in range(3):
#     send_command(turtle, ?, ?, ?)
#     send_command(turtle, ?, ?, ?)
```

Cleanup

Shutdown the ROS2 node after completing tasks.

In [ ]:
```
turtle.destroy_node()
rclpy.shutdown()
```