Python Script to control the TrutleBot

Up until now we have learned to move the TurtleBot with teleoperation and sending commands through the CCS. Let's now assume we want the robot to move in a more complex pattern. For that writing a script is convenient since we can exactly specify what sequence of steps the robot will take.

Writing your first publisher

We actually want to write a small program to *publish* messages to a specific topic. This program is called a *Publisher*. A Publisher is a node that keeps publishing a message into a topic with a specified rate.

Remember that a topic is not a message itself. You can see it as a channel or a pipe. ROS nodes can then send messages through this pipe to each other and others can read those messages.

Naming your nodes correctly

Before we actually get into writing the code let's think on how to name it. Remember, we need a convenient name for our Publisher. It's very bad practice to name your nodes like: 'my_node' or 'publisher_test'.

Try to give descriptive names to your nodes!

Question 1

1 point possible (ungraded)

Which of the following node names you think is best, for a publisher that moves the turtlebot in circles?

- ○ /node_turtlebot
- ○ /drive_turtlebot_circle
- ○ /walking

Submit

Question 2

1 point possible (ungraded)

Next we need to find out to which topic we are going to publish. Remember that it will be the same topic we used to send commands through the CCS.

What is the name of the topic we should publish to?

- ○ /odom
- ○ /cmd_vel_mux/input/teleop
- ○ /turtle_move_base

Submit

Now that we have our *Publisher node* named correctly, and we know the topic we want to publish to, it's time to find out what type of messages we need to use. We simply can look up info about the topic using the rostopic info command, just replace the '<answer-to-Q2>' by the correct topic name.

```
$ rostopic info <answer-to-Q2>
```

Question 3

1 point possible (ungraded)

One of the important piece of information is the message type. From theNext we need to find out to which topic we are going to publish. Remember that it will be the same topic we used to send commands through the CCS.

What is the message type of the topic we're using?

- ◯ /nav_msgs/Odometry
- ◯ /geometry_msgs/Twist
- ◯ /rosgraph_msgs/Clock

Submit

To find information about a specific message type, you can use the rosmsg show command.

Just replace the '<answer-to-Q3>' by the correct message Type, and run the following command:

```
$ rosmsg show <message-type>
```

Question 4

1 point possible (ungraded)

How many fields does our message type have?

Submit

Now you are ready to write the full script!

Using your favorite editor, create the `<answer-to-Q1>.py` script on the `$HOME/hrwros_ws/src/hrwros_week3/scripts` folder.

If you do not have this folder structure yet, please make sure you have already downloaded Week 3 files from the Weekly contents:

Copy and paste the below template code to the script and try to complete the script.

There are 3 spaces you have to fill:

[answer-to-Q2] is the topic name, we find out on Question 2

[linear-axis] is the axis for linear movement of the turtlebot.

[angular-axis] is the axis for rotation of the turtlebot.

Template Code

```
#!/usr/bin/env python3
## Node to drive the turtlebot in circles.


import rospy
from geometry_msgs.msg import Twist

rospy.init_node('drive_turtlebot_circle')
pub = rospy.Publisher([answer-to-Q1], Twist, queue_size=1)
rate = rospy.Rate(2)
move = Twist()
move.linear.[linear-axis] = 0.2 #Move the robot with a linear
velocity in the [] axis
move.angular.[angular-axis] = 0.5 #Move the with an angular
velocity in the [] axis

while not rospy.is_shutdown():
  pub.publish(move)
  rate.sleep()
```

Once the script to control the turtlebot is completed, you need to make it executable, **Inside the CCS** go to the corresponding folder and change its executing permission

```
$ roscd hrwros_week3/scripts            // move to the script
location
$ chmod a+x <answer-to-Q1>.py           // make the script
executable
```

Now you are ready to run the full python script to control the turtlebot!