Ros Notes

EDXIBOS|

→ add source /opt/ros/noetic/setup.bash // This will 'source' Ros files or some shit.
[All changes will be in same shell / not subshell]

1. Catkin Workspace

$ mkdir -p ~/catkin_ws/src // makes folder src inside catkin_ws [which also is generated if not present already]

$ cd ~/catkin_ws/ ← current directory

$ catkin_make ← creates a CMakeLists.txt file inside 'src' → Text describing how to build code / where to install it

// now we've made a workspace [Set of directories w/ all the tools / related Ros code needed] called 'catkin_ws'

// devel contains setup files → running it tells system to use workspace & its code

Creating Packages

→ All package files inside /catkin_ws/src/

→ $ catkin_create_pkg ultimatrix std_msgs rospy roscpp
(dependencies)

Writing Publisher & Subscriber nodes

→ Change directory to package

$ roscd ultimatrix // roscd is Ros command → knows about packages
→ can go to package without typing path

Lets deconstruct the code

→ Publishing a topic

```
1  #!/usr/bin/env python → // Lets OS know this is a python file
                          ↳ passes to python interpreter
```

chmod - command which modifies access permission for files / directories

```
3  import rospy  - // every ROS node needs this
4  from std_msgs.msg import String → The messages are of type 'string'
                                   → We're importing the message type → a string container

7  pub = rospy.Publisher("chatter", String, queue_size=10)   // This line declares that node is publishing to so&so topic and so&so msg type
                                                             limits the message queue size to 10 if subscriber is not receiving fast enough
8  rospy.init_node('talker', anonymous=True)
                   ↑ name of node   ↑ makes sure name is unique (adds bs numbers at end of name)

9  rate = rospy.Rate(10) # 10 Hz → Code loops 10 times per second
                                 → Message sending frequency
   ↑ object of type 'Rate'

10. while not rospy.is_shutdown():   ← condition
        custom_str = "Access master Control %s" %. rospy.get_time()
        rospy.loginfo(custom_str) → prints message on screen + writes message to Node log file + writes message to rosout
        pub.publish(custom_str) → "Work" (publisher node will publish string in topic 'chatter'
        rate.sleep()  ← maintains desired rate of 10 Hz. by sleep function
    ↑ the object !! Remember?
```

callback - function that is called by another function, which takes first function as parameter

- A function which is 'called' when something happens.

My publisher node:

```
#!/usr/bin/env python

import rospy

from std_msgs.msg import String

pub = rospy.Publisher("team_abhiyaan", String, queue_size=10)
rospy.init_node("publisher_node", Anonymous=True)
rate = rospy.Rate(10)

while not rospy.is_shutdown():
    str = "Team Abhiyaan Rocks"
```