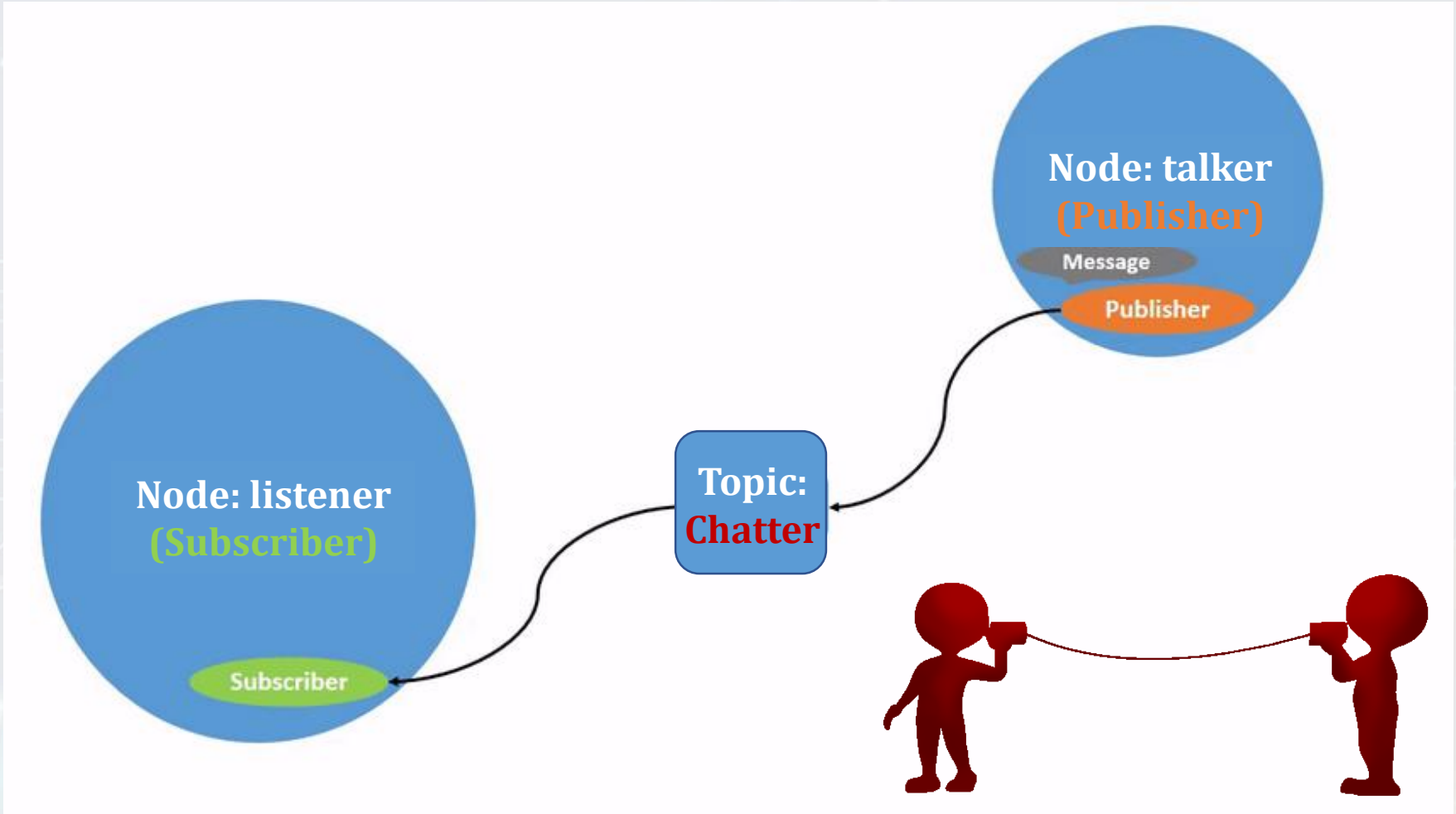
ROS 2 Python Chatter
By TESR

ROS2

Thai Embedded

Chatter concept



Create the workspace and package

- Before we start, Install a useful tool using:

```
sudo pip3 install gdown -y
```

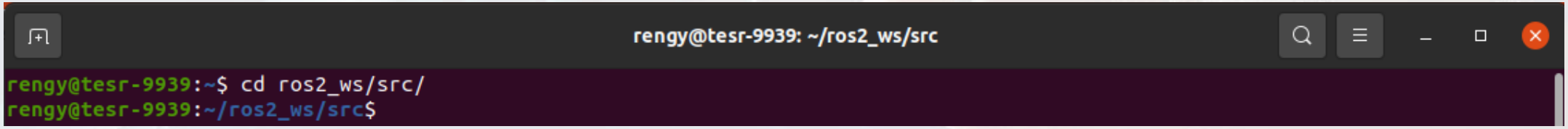
- And then, To create a new ros2 **workspace** open new terminal and type:

```
mkdir -p ros2_ws/src
```

```
rengy@tesr-9939:~$ ls
arduino                          catkin_ws   grafana                NodeRed          sketchbook  untitled1
build-test1-Desktop-Debug        Downloads   model_editor_models    python_chatter   turtlebot3
cartographer_ws                  foxy_ws     Music                  ros2_ws          ui
```

- Change the current working directory to **ros2_ws/src** using:

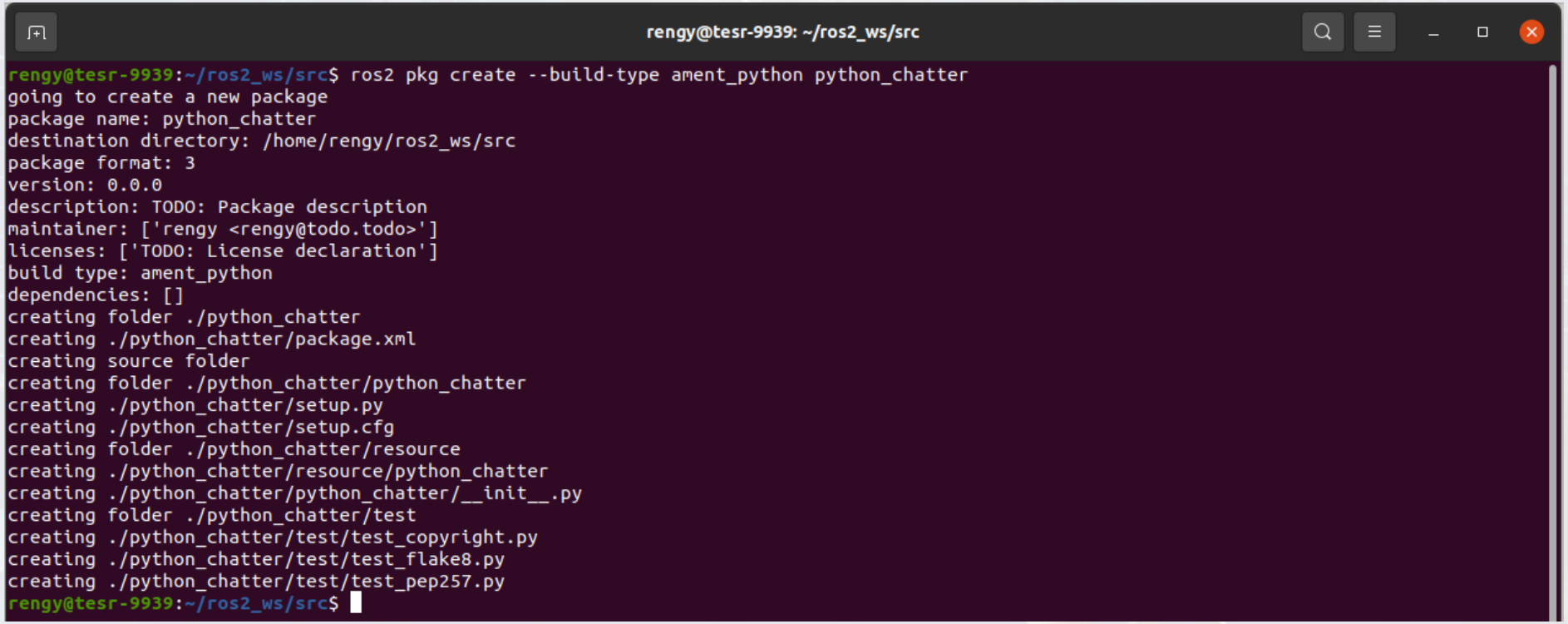
```
cd ros2_ws/src
```



Create the workspace and package

- Create the **package** in src of workspace using:

```
ros2 pkg create --build-type ament_python python_chatter
```



Create the workspace and package

- And then, you have 2 files that need to edit are:
  - **package.xml**
  - **setup.py**

```
rengy@tesr-9939:~/ros2_ws/src$ ls
python_chatter
rengy@tesr-9939:~/ros2_ws/src$ cd python_chatter/
rengy@tesr-9939:~/ros2_ws/src/python_chatter$ ls
package.xml  python_chatter  resource  setup.cfg  setup.py  test
```

Create the workspace and package

- We must add following dependencies corresponding to **package.xml**

```
<depend>rclpy</depend>
<depend>std_msgs</depend>
```

```
1  <?xml version="1.0"?>
2  <?xml-model href="http://download.ros.org/schema/package_format3.xsd"
   schematypens="http://www.w3.org/2001/XMLSchema"?>
3  <package format="3">
4    <name>python_chatter</name>
5    <version>0.0.0</version>
6    <description>TODO: Package description</description>
7    <maintainer email="pi@todo.todo">pi</maintainer>
8    <license>TODO: License declaration</license>
9
10   <buildtool_depend>ament_cmake_python</buildtool_depend>
11   <depend>rclpy</depend>
12   <depend>std_msgs</depend>
13
14   <test_depend>ament_copyright</test_depend>
15   <test_depend>ament_flake8</test_depend>
16   <test_depend>ament_pep257</test_depend>
17   <test_depend>python3-pytest</test_depend>
18
19   <export>
20     <build_type>ament_python</build_type>
21   </export>
22 </package>
```

Create the workspace and package

- Add following **entry_points** to your node import statement in **setup.py**

```
entry_points={
   'console_scripts': [
      'talker = python_chatter.py_talker:main',
      'listener = python_chatter.py_listener:main',
      ],
},
```

```python
from setuptools import setup

package_name = 'python_chatter'

setup(
    name=package_name,
    version='0.0.0',
    packages=[package_name],
    data_files=[
        ('share/ament_index/resource_index/packages',
            ['resource/' + package_name]),
        ('share/' + package_name, ['package.xml']),
    ],
    install_requires=['setuptools'],
    zip_safe=True,
    maintainer='pi',
    maintainer_email='pi@todo.todo',
    description='TODO: Package description',
    license='TODO: License declaration',
    tests_require=['pytest'],
    entry_points={
        'console_scripts': [
            'talker = python_chatter.py_talker:main',
            'listener = python_chatter.py_listener:main',
        ],
    },
)
```

Build package

- Check for missing dependencies before building:

```
cd ~/ros2_ws/
rosdep install -i --from-path src --rosdistro foxy -y
```

- As a result, show **"All required rosdeps installed successfully"**.

```
rengy@tesr-9939:~/ros2_ws$ rosdep install -i --from-path src --rosdistro foxy -y
#All required rosdeps installed successfully
```

Build package

- So, you are ready to build package using:

```
colcon build
```

```
rengy@tesr-9939:~/ros2_ws$ colcon build
Starting >>> python_chatter
Finished <<< python_chatter [0.82s]

Summary: 1 package finished [0.96s]
```

- After, build is completed source environment and echo source script to

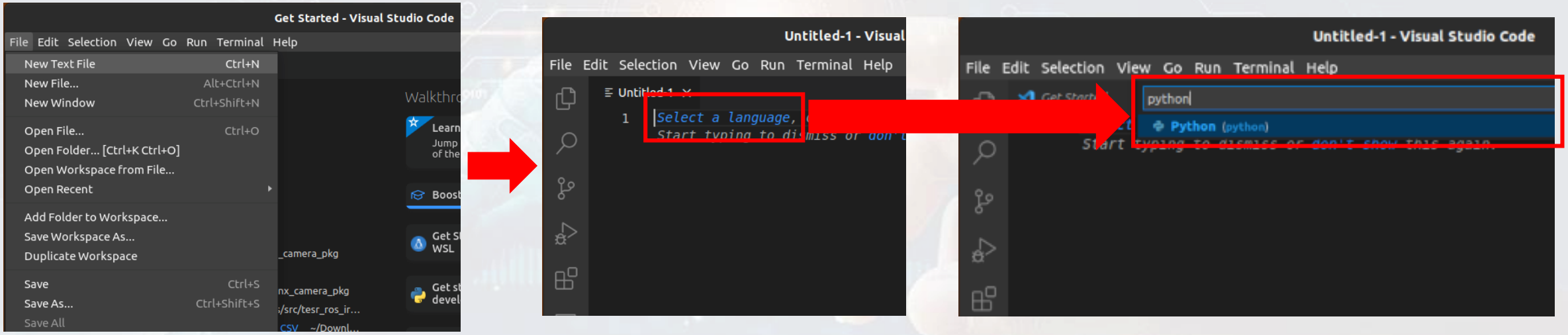
```
echo "source ~/ros2_ws/install/setup.bash" >> ~/.bashrc
source ~/.bashrc
```

Create the Python scripts

- Open the Visual Studio Code using:

```
code
```

- Create a new file by select **"File > New Text File"** and select Python language.



Thai Embedded

Create Node Talker(Publisher)

- **Example Python code: (name as py_talker.py)**

```python
#!/usr/bin/python3
import rclpy
from rclpy.node import Node
from std_msgs.msg import String

class talker(Node):
    def __init__(self):
        super().__init__('talker_node')
        self.topic = "chatter"
        self.publishers_ = self.create_publisher(String,self.topic,10)
        timer_peroid = 0.5 #seconds
        self.timer = self.create_timer(timer_peroid, self.timer_callback)
        self.i = 0

    def timer_callback(self):
        msg = String()
        msg.data = "message:" +str(self.i)
        self.publishers_.publish(msg)
        self.get_logger().info('Publishing: "%s"'%msg.data)
        self.i += 1

def main(args=None):
    rclpy.init(args=args)

    talker_pub = talker()
    rclpy.spin(talker_pub)

    talker_pub.destroy_node()
    rclpy.shutdown()

if __name__ == '__main__':
    main()
```

Create Node Listener(Subscriber)

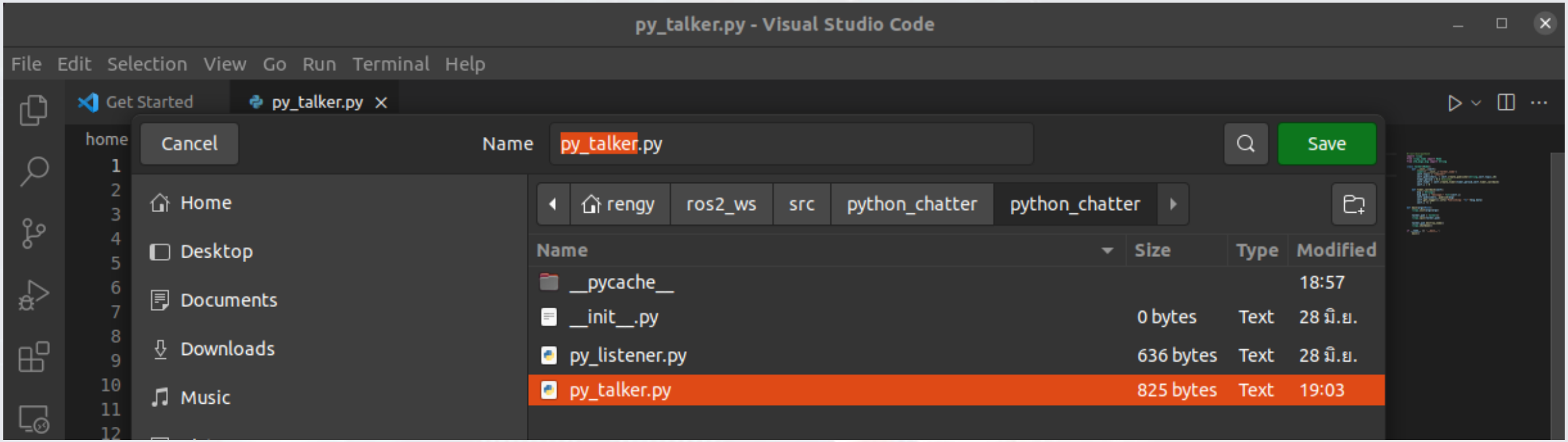
- **Example Python code: (name as py_listener.py)**

```
#!/usr/bin/python3
import rclpy
from rclpy.node import Node
from std_msgs.msg import String

class listener(Node):
    def __init__(self):
        super().__init__('listener_node')
        self.topic = "chatter"
        self.subscription = self.create_subscription(String, self.topic, self.listener_callback, 10)
        self.subscription

    def listener_callback(self,msg):
        self.get_logger().info('I heard: "%s"' %msg.data)
```

```
def main(args=None):
    rclpy.init(args=args)

    listener_sub = listener()
    rclpy.spin(listener_sub)

    listener_sub.destroy_node()
    rclpy.shutdown()

if __name__ == '__main__':
    main()
```

Save file to package's script folder

- Save your code at **"ros2_ws/src/python_chatter/python_chatter"**



Save file to package's script folder

- And then, give the permission to execute to the files using:

```
cd ~/ros2_ws/src/python_chatter/python_chatter/
sudo chmod +x *
```

```
rengy@tesr-9939:~$ cd ~/ros2_ws/src/python_chatter/python_chatter/
rengy@tesr-9939:~/ros2_ws/src/python_chatter/python_chatter$ sudo chmod +x *
[sudo] password for rengy:
rengy@tesr-9939:~/ros2_ws/src/python_chatter/python_chatter$ ls
__init__.py  __pycache__  py_listener.py  py_talker.py
```

- After that, build the package again using:

```
cd ~/ros2_ws
colcon build
```

Run Talker using Python

- On first terminal, run the talker by type:

```
ros2 run python_chatter talker
```

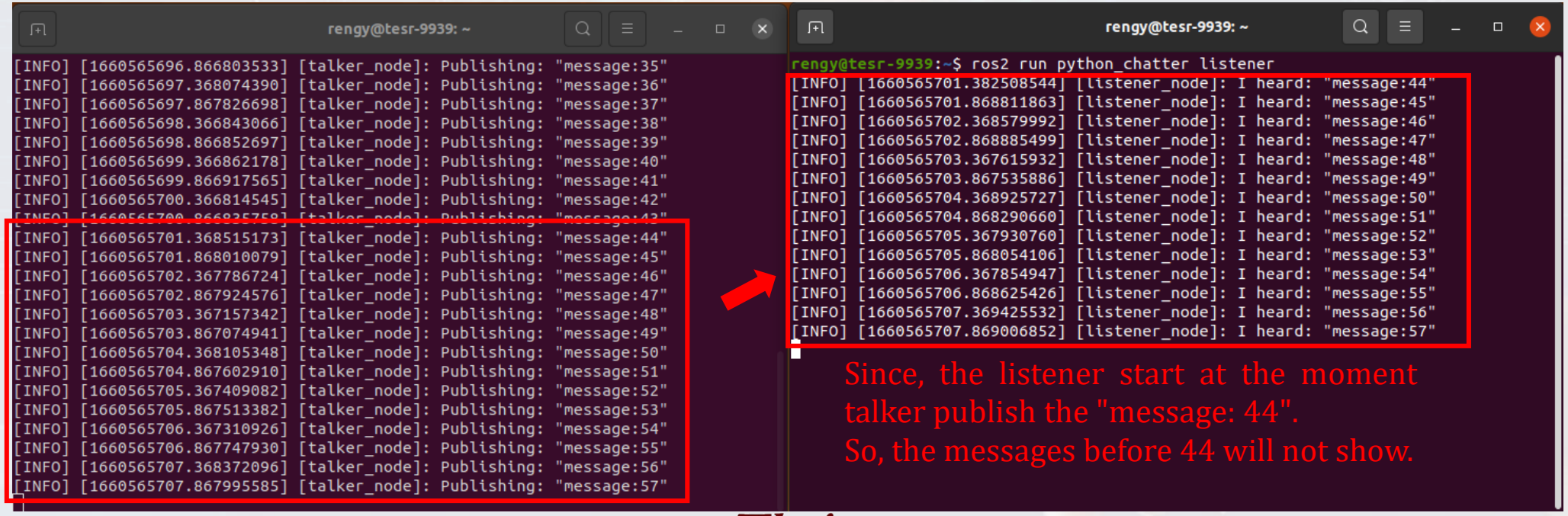
- As a result, this program is keep publish the **"message: increasing number"**

Run Listener using Python

- Open the second terminal and then run the listener by type:

```
ros2 run python_chatter listener
```

- As a result, the second terminal will show from talker:



ROS 2 element in Chatter

- Check Chatter's ROS 2 node

```
ros2 node list
```

- Check Chatter's ROS 2 topic

```
ros2 topic list
```

- Check Chatter's ROS 2 service

```
ros2 service list
```

- Check Chatter's ROS 2 param

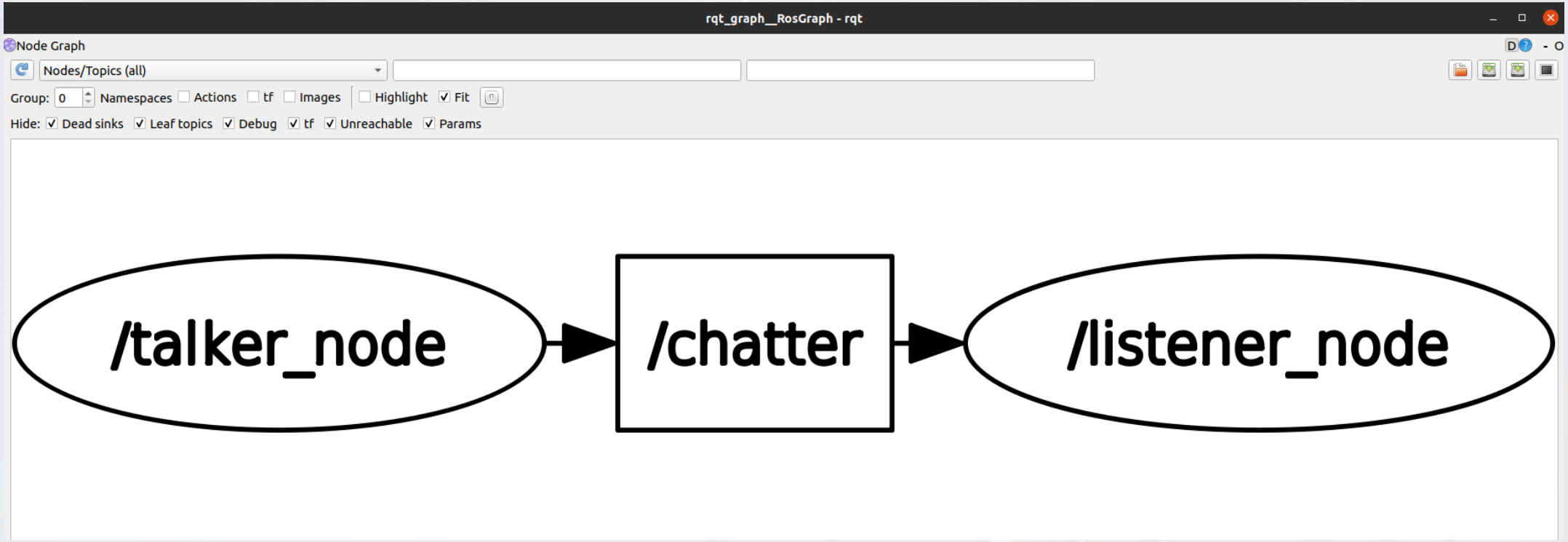
```
ros2 param list
```

```
rengy@tesr-9939:~$ ros2 node list
/listener_node
/talker_node
rengy@tesr-9939:~$ ros2 topic list
/chatter
/parameter_events
/rosout
rengy@tesr-9939:~$ ros2 service list
/listener_node/describe_parameters
/listener_node/get_parameter_types
/listener_node/get_parameters
/listener_node/list_parameters
/listener_node/set_parameters
/listener_node/set_parameters_atomically
/talker_node/describe_parameters
/talker_node/get_parameter_types
/talker_node/get_parameters
/talker_node/list_parameters
/talker_node/set_parameters
/talker_node/set_parameters_atomically
rengy@tesr-9939:~$ ros2 param list
/listener_node:
  use_sim_time
/talker_node:
  use_sim_time
```

Chatter's RosGraph by rqt_graph

- Open new terminal and type:

```
rqt_graph
```



Contact Us

Email: tesrshop@gmail.com
Line official Account: @ion1900z
Facebook fanpage: TESR
Tel. 082-983-7768

Scan here

More

Thai Embedded

TESR Co., LTD

112/296 หมู่บ้าน เพอร์เฟค มาสเตอร์พีซ
หมู่ที่ 2 ตำบลไทรม้า อำเภอเมืองนนทบุรี
จังหวัดนนทบุรี 11000