



## ROS2: Writing a simple publisher and subscriber (Python)

In this section, we will learn how to write a simple publisher and subscriber program in python using an example.

### Introduction

Before we redirect you to the example, allow us to give a brief introduction about publishers and subscribers.

#### Publisher

A ROS publisher is a ROS node that publishes a specific type of ROS message over a given ROS topic.

The publisher creates a particular type of ROS message. In the given example, it's `std_msgs/String`. How do we know which message is published in our topic? There's a command given in [ROS2 Concepts](#), do you remember which one? 😞

#### Subscriber

A ROS Subscriber is a ROS node which is essentially a process or executable program, written to 'obtain from' or 'subscribe to' the messages and information being published on a ROS Topic.

Similar to publishers, we can use the same command given in [ROS2 Concepts](#) to find out which message is being received by the subscriber.

---

**Note:** In Task 1A, we'll need to create a **single** node that subscribes from "*pose*" and publishes to "*cmd\_vel*" topics.

---

### THE EXAMPLE: Writing a simple publisher and subscriber (Python)

We're all ready for some hands on learning now! Given below is the link to official ROS2 Wiki website. Let's head over straight to the example given on their webpage and try to perform the same.

*(Note: It is very important that you perform this example to strengthen your understanding of ROS especially for Task 1)*

### ROS 2 Wiki Example: Writing a simple publisher and subscriber (Python)

---

## ROS2: Writing a Service Client (Python)

We have already discussed ROS services in [ROS2 Concepts](#). Now it's time to try making a service client on our own. Making a server is not needed for task 1 and therefore not done here, but if you are interested feel free to explore:

Given below is the syntax for a service function to change the colour of the pen in turtlesim.

```
from turtlesim.srv import SetPen

def call_set_pen_service(self, r, g, b, width, off):
    client = self.create_client(SetPen, '/turtle2/set_pen')
    while not client.wait_for_service(1.0):
        self.get_logger().warn("Waiting for service...")

    request = SetPen.Request()
    request.r = r
    request.g = g
    request.b = b
    request.width = width
    request.off = off

    future = client.call_async(request)
```



In order to call the service, you can simply call the defined function wherever you want to run the service in your code. Note that we're not using the done callback in this service function for the sake of simplicity, i.e. this service will not provide a textual feedback when initiated successfully.

Also, making a server is not needed for task 1 and therefore not discussed here.

If you wish to learn more about services, you can visit [ROS2 Wiki: Writing a simple service and client \(Python\)](#)

---

