

## Tips and Suggestions

### Use Terminator:

You will have multiple Terminal windows running at once while implementing this theme. Using the standard Terminal can start being unwieldy, Terminator offers a nice solution to neatly manage the multiple Terminal windows. Visit this [link](#) for more details.

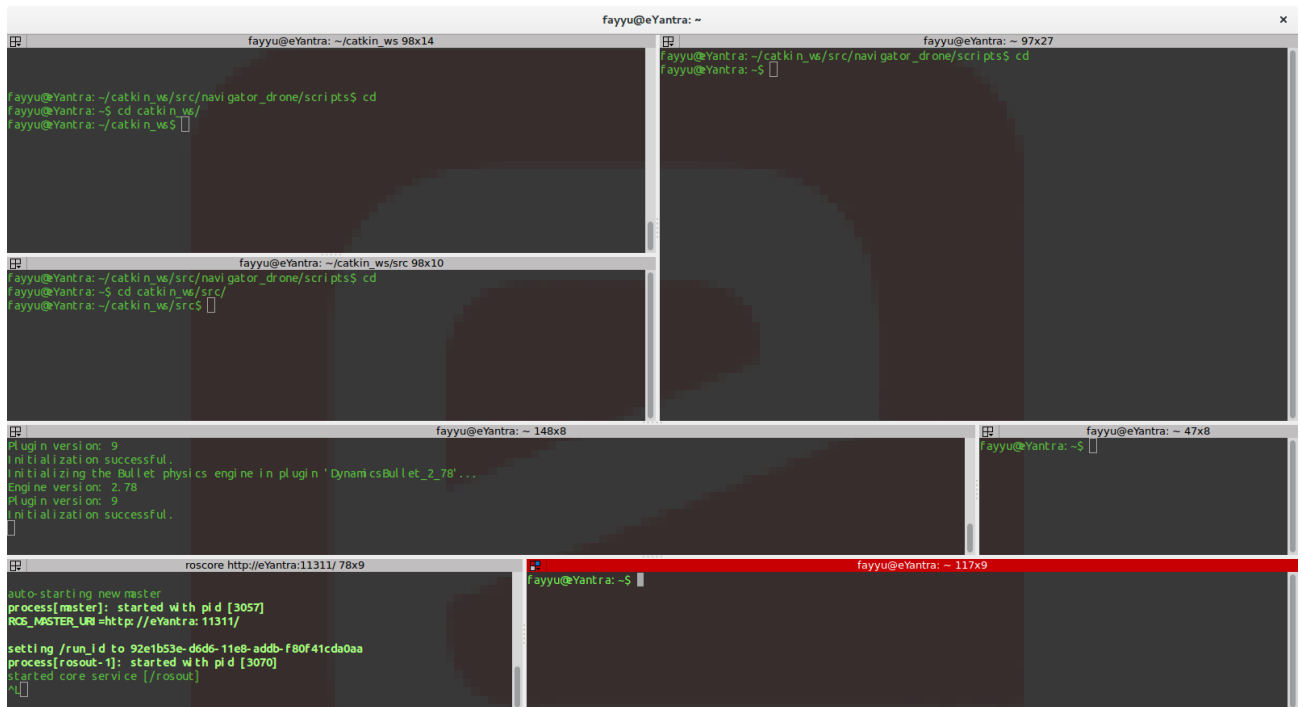


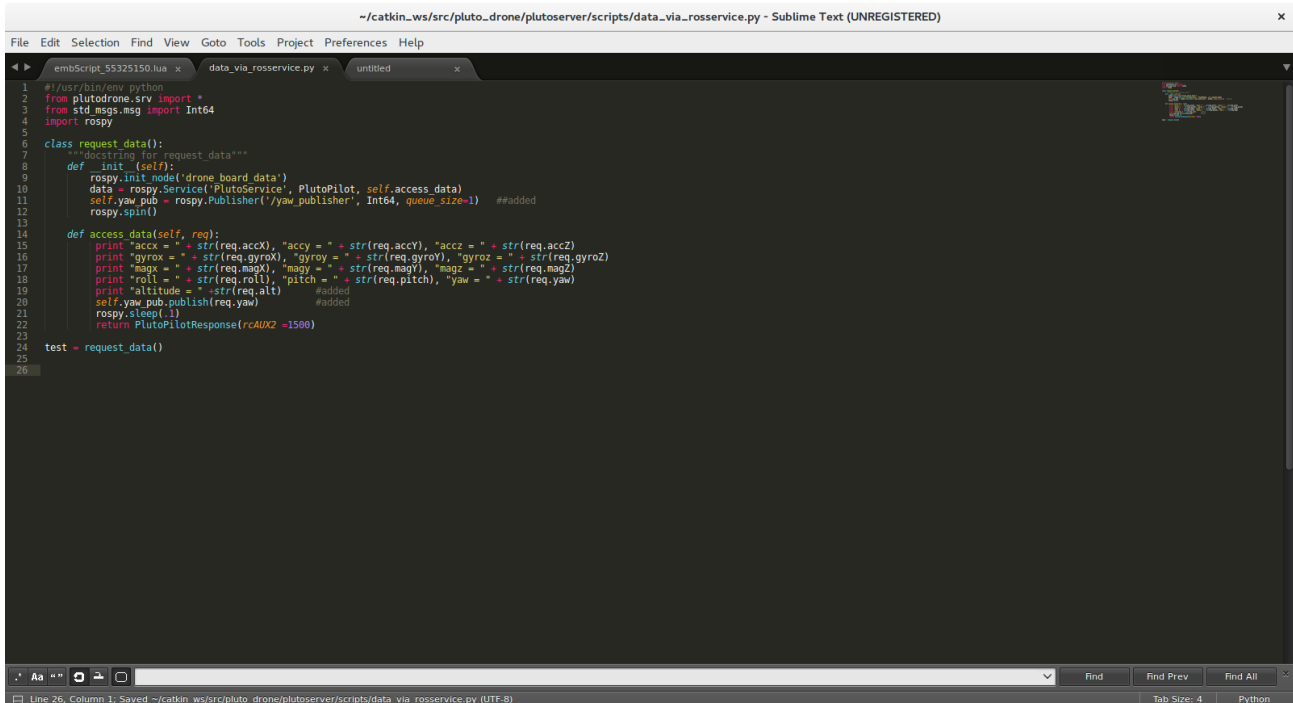
Figure 1: Terminator windows

### Use Workspaces:

We highly recommend that you use multiple workspaces while developing a solution. They will help keep your desktop uncluttered. Here is a [link](#) introducing workspaces in Ubuntu.

## Use Sublime Text:

A good text editor goes a long way in making your work-flow smoother, we find Sublime Text 3 to be well-suited for this purpose and recommend you use it.



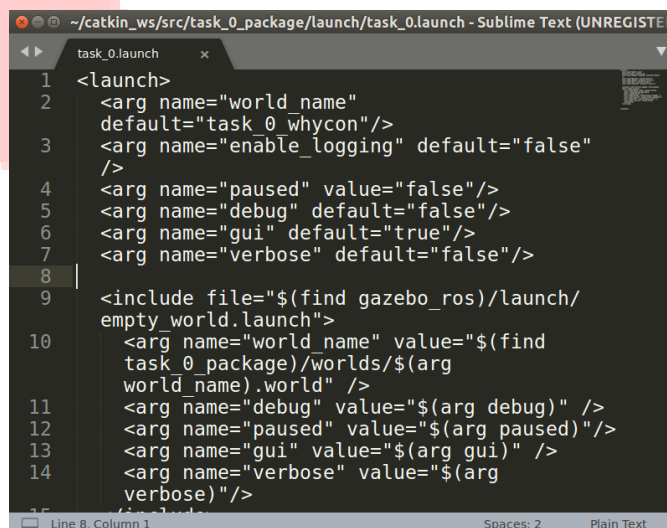
```

1 #!/usr/bin/env python
2 from plutodrone.srv import *
3 from std_msgs.msg import Int64
4 import rospy
5
6 class request_data():
7     """docstring for request_data"""
8     def __init__(self):
9         rospy.init_node('drone_board_data')
10        data = rospy.Service('PlutoService', PlutoPilot, self.access_data)
11        self.yaw_pub = rospy.Publisher('/yaw_publisher', Int64, queue_size=1) #added
12        rospy.spin()
13
14    def access_data(self, req):
15        print "accx = " + str(req.accx), "accy = " + str(req.accy), "accz = " + str(req.accz)
16        print "gyrox = " + str(req.gyrox), "gyroy = " + str(req.gyroy), "gyroz = " + str(req.gyroz)
17        print "magx = " + str(req.magx), "magy = " + str(req.magy), "magz = " + str(req.magz)
18        print "roll = " + str(req.roll), "pitch = " + str(req.pitch), "yaw = " + str(req.yaw)
19        print "altitude = " + str(req.alt) #added
20        self.yaw_pub.publish(req.yaw) #added
21        rospy.sleep(1)
22        return PlutoPilotResponse(rcAUX2 = 1500)
23
24 test = request_data()
25
26

```

Figure 2: The Sublime Text 3 Editor

Note: While opening ‘.launch’ or ‘.world’ file via sublime or any text editors, it might preview you the scripts in plaintext format , which you may find difficult to distinguish in between comment or uncomment line, as shown in figure3.



```

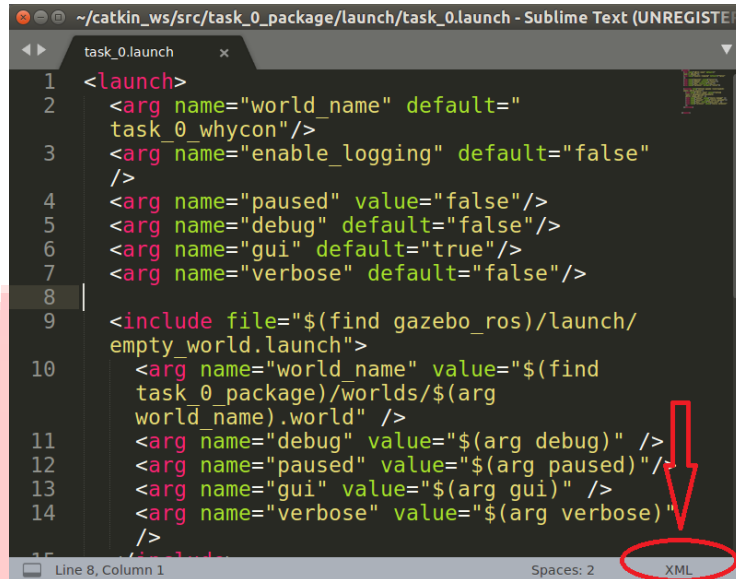
1 <launch>
2   <arg name="world name"
3     default="task_0 whycon"/>
4   <arg name="enable_logging" default="false"
5     />
6   <arg name="paused" value="false"/>
7   <arg name="debug" default="false"/>
8   <arg name="gui" default="true"/>
9   <arg name="verbose" default="false"/>
10
11   <include file="$(find gazebo_ros)/launch/
12     empty_world.launch">
13     <arg name="world name" value="$(find
14       task_0_package)/worlds/$(arg
15       world_name).world" />
16     <arg name="debug" value="$(arg debug)" />
17     <arg name="paused" value="$(arg paused)" />
18     <arg name="gui" value="$(arg gui)" />
19     <arg name="verbose" value="$(arg
20       verbose)" />
21   </include>

```

Figure 3: Plain text format

So, for making it more lively and colourful, on the right lower corner of the sublime screen. Click on “Plain Text” . A whole list of different formatting will appear. Scroll down and select “XML”.

As shown in Figure 4.



The screenshot shows the Sublime Text editor with a file named `task_0.launch` open. The file contains XML code for a ROS launch file. The status bar at the bottom indicates 'Line 8, Column 1' and 'Spaces: 2'. A red arrow points to the 'XML' option in the status bar, which is circled in red.

```
1 <launch>
2   <arg name="world_name" default="
  task_0_whycon"/>
3   <arg name="enable_logging" default="false"
  />
4   <arg name="paused" value="false"/>
5   <arg name="debug" default="false"/>
6   <arg name="gui" default="true"/>
7   <arg name="verbose" default="false"/>
8
9   <include file="$(find gazebo_ros)/launch/
  empty_world.launch">
10    <arg name="world_name" value="$(find
  task_0_package)/worlds/$(arg
  world_name).world" />
11    <arg name="debug" value="$(arg debug)" />
12    <arg name="paused" value="$(arg paused)" />
13    <arg name="gui" value="$(arg gui)" />
14    <arg name="verbose" value="$(arg verbose)"
  />
15  </include>
```

Figure 4: XML format