Starting Out

Before you try anything in this document it is **highly recommended** that you run through the **ROS** tutorials at http://wiki.ros.org/ROS/Tutorials and then create your own package to use to house and run your scripts and nodes.

Basics

To write a node or script in python first you must change the file type to python script by clicking Plain Text at the bottom of the window and selecting Python from the drop down menu. Then to tell ROS that this is a python script you must copy and paste the following into the very first line of the script:

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

To make a node or script executable you must open up a terminal and set your current directory to the folder the node or script is in and then use the following command where script_name is the name of your script:

```
chmod +x script_name
```

Then set current directory back to your main working directory and use the following command:

```
catkin_make
```

Movement

To write a movement node or script look at the scripts in the movement folder.
Note: The Euler to Quaternion script contains a subroutine which you must move when giving Baxter orientations in movement commands as it converts Euler angles (rotation about x,y and z axes) to the Quaternion coordinate system which Baxter must be given orientations in. The explanation of how this conversion is done can be found at this website http://www.euclideanspace.com/maths/geometry/rotations/conversions/eulerToQuaternion/.
Note: You will need the ik_solver .py file in your scripts folder as well as it is used by the movement examples to calculate the relevant joint positions for the arms from a position and orientation in Cartesian space.

Vision

To write a vision node or script look at the script in the vision folder which shows how a camera subscriber script works for Baxter.
Note: This script is Copyright 2014 Active Robots Ltd and for Educational Use Only.

Running a Script or Node

To run a script use the following command in a terminal where script_name is the name of your script and package_name is the name of the package you've created that your script is contained within:

```
rosrun package_name script_name
```