


ROS Tutorials

The  [official ROS tutorials](#) are very comprehensive, however I found the way they present ROS to a beginner results in a very steep learning curve. Here is our own set of tutorials, which are a work in progress. I may reference tutorials from other sites if I find them useful.

Pre-requisites

Basic linux command-line knowledge (specifically BASH) is assumed. If you are completely unfamiliar with the command line, check out this [codecademy tutorial](#), or [this series of videos](#).

High Level Concepts

ROS is many things, it is primarily a communication framework, but it also has a huge set of useful tools, including viewing data, navigating the filesystem, and starting up processes.

Each process is called a **node**. Nodes are designed to do a specific task. We can create complex functionality (such as controlling a robot) by building multiple nodes and having them talk together.

There are several ways for nodes to talk to each other, the most common one is a **topic**. The basic idea is that a single node can “publish” data to a topic, and then one or more nodes can “subscribe” to the topic and receive the data.

Nodes are typically grouped together into a **package**. All of our software is in a single package called “robosub”. When you install ROS, several other packages are installed that provide some useful functionality, such as `rqt_plot`.

Command Line Tools

ROS provides many useful command-line tools. Your BASH environment gets access to these when you have the following line in your `~/.bashrc` file:

```
source /opt/ros/indigo/setup.bash
```

Most ROS commands support tab autocompletion, so take advantage of it!

Navigating the filesystem

ROS has a few different commands that are useful for moving and looking around. They are typically in the form:

```
$ <command> <package_name> <...>
```

Most are based on the typical navigation functions:

- roscd
- rosls

Starting nodes

Starting a single node

While you can manually find the executable (or python script) and run it manually, ROS provides a tool for quickly finding and starting a node, the `roslaunch` command:

```
$ roslaunch <package_name> <node_name>
```

Starting multiple nodes

The `roslaunch` command allows you to launch multiple nodes at the same time. The nodes are specified in a special file called a launch file.

```
$ roslaunch <package_name> <launchfile_name>
```

More information about launch files can be found [here](#)

Viewing active nodes and topics

The `rostopic` command shows information about topics. You can view all currently running nodes by running

```
$ rostopic list
```

Likewise, the `rostopic` command shows information about topics. You can view all currently active topics by running

```
$ rostopic list
```

How to Write a ROS Node

Check out these tutorials for whatever language you're using: [C++](#) [Python](#)

From:
<https://robosub.eecs.wsu.edu/wiki/> - **Palouse RoboSub Technical Documentation**

Permanent link:
<https://robosub.eecs.wsu.edu/wiki/cs/ros/tutorials/start>

Last update: **2016/09/23 21:34**



