ROS Tutorial #7:

Note, this tutorial assumes that you have just finished tutorial #6. But, the tutorial #7 initial instructions are not quite correct, as tutorial #6 assumes you are running two copies of the turtle, while tutorial #7 assumes a single turtle. So, to set up #7 properly, do this:

- Open a new terminal
- cd ~/catkin_ws
- source devel/setup.bash
- roscore & % start the ross master, the
- rosrun turtlesim turtlesim_node & % start the turtlesim node, popping up a window
- rosrun turtlesim turtle_teleop_key % start the keyboard logger

Now start the steps of the beginner tutorial #6.

- 2.1: rosservice type /clear % what arguments does the /clear service require?
- 2.2: rosservice call /clear % call turtlesim /clear, which erases any paths
- 2.3: rosservice type /spawn | rossrv show
  "This is “linux speak” to string together two commands
  rosservice type /spawn returns “turtlesim/spawn”
  the “| rossrv show” bit in effect calls: rossrv show turtlesim/spawn
  this returns the message structure of the spawn service in the turtlesim node
- 2.4: rosservice call /spawn 2 2 0.2 ""
  Spawn a new robot in the window at location (x=2,y=2,theta=0.2) with no name

The next part of tutorial #7, part 3, deals with parameters in ROS, using the rosparm function

- 3.1: rosparm list % list the available parameters
- 3.2: rosparm set /turtlesim/background_r 150
  Note, the tutorial is wrong. The tutorial does properly prepend the “/turtlesim” portion of the name. This probably comes from the way that ROS melodic works versus
- 3.2 (continued): rosparm get /turtlesim/background_g
  Again, the tutorial is wrong, forgetting to prepend the “/turtlesim” portion
- 3.2 (continued): rosparm get /
  The tutorial function call is correct, but ROS melodic returns a different format that that described in the tutorial
- 3.3: rosparm dump params.yaml
- 3.3 (continued): rosparm load params.yaml copy
- 3.3 (continued): rosparm get /copy/turtlesim/background_b
  The tutorial is wrong. The turtlesim node needs to be specified.