

ROS python

Juan Casado Ballesteros

Setup python

We need to have a python stable version binded with the corresponding pip to manage dependencies. For the example we will work with *Python 3.5*.

```
sudo apt-get install python3-pip python3-yaml
sudo pip3 install rospkg catkin_pkg
```

Setup workspace

First we need to tell ROS which version of python it needs to execute. We can do this when we create the package.

```
mkdir -p ~/catkin_ws/src;
cd ~/catkin_ws/;
catkin config -DPYTHON_EXECUTABLE=/usr/bin/python3 \
-DPYTHON_INCLUDE_DIR=/usr/include/python3.5m \
-DPYTHON_LIBRARY=/usr/lib/x86_64-linux-gnu/libpython3.5m.so;
catkin config --install;
catkin_make;
source devel/setup.bash;
```

Setup the file architecture

The convenient way to do this is to have a /scripts directory where we will write our *headers* and then have another /src directory where to write the actual code.

To make a friendly readable cli for our package it's recommended that we write our /scripts/files without extension so to run a bit of code we would be doing *roslaunch* instead of *roslaunch .py* which is way more convenient.

Tell ROS about our file architecture

We need to declare where our code is so ROS can find it. To do so we will need to modify our CmakeList.txt, and also to create a setup.py.

CmakeList.txt

```
catkin_python_setup()
install(PROGRAMS
  scripts/myscript1
  DESTINATION ${CATKIN_PACKAGE_BIN_DESTINATION}
)
```

setup.py

```
#!/usr/bin/env python3.5
from distutils.core import setup
from catkin_pkg.python_setup import generate_distutils_setup
setup_args = generate_distutils_setup(
    packages=['mypackage'],
    package_dir={'': 'src'}
)
setup(**setup_args)
```

Writing a new python node

To write a new python node we will need to create at least two files one that will be located in /scripts and another that will be located in the corresponding directory inside /src.

A proper example would be

```
/ws
  /package
    /src
      /scripts
        script1
      /src
        /script1
          main.py
        CMakeLists.txt
        package.xml
        setup.py
    ...
```

Which will be run with *roslaunch package script1*

script1

```
#!/usr/bin/env python3.5
from script1.main import main
if __name__ == '__main__':
    main()
```

main.py

```
def main():
    print('Hello ROS!!!')
```

Setup to run

After ROS can detect our script we need to make it executable with *chmod +x* and then we need to *catkin_make* the package.

IT IS IMPORTANT TO MAKE THE SCRIPT FILES EXECUTABLE !!!