1. Introduction

Official documentation

- Packages are container for ROS code.
- They facilitate to release certain code work and allow others to build and use it easily
- Packages creation in ROS uses Ament as its build system and Colcon as its build tool. But one can create a package using either CMake or Python.

2. Creation

2.1 Independent Python or C++ package

1. Move to the src directory of the workspace
2. Create the package indicating the build type for python, the name and the dependencies

```
ros2 pkg create <my_package> --build-type ament_<> --
dependencies <> <>
```

Option	Python	C++
build	ament_python	ament_cmake
dependencies	rclpy	rclcpp

3. Build the package

```
colcon build
colcon build --packages-select <my_package>
```

- Important files are automatically created:
  - For Python:
    - setup.py: Drives the compilation process indicatind what files, where to install, how to link dependencies, ets.
    - setup.cfg: Indicates where the scripts will be installed.
    - package.xml (manifiesto): Metadata to indicate the package dependencies and developer information
  - For C++:
    - CMakeLists: Compilation file
    - package.xml (manifiesto): Metadata to indicate the package dependencies and developer information

4. Customize the manifesto and compilation files following the statements
   - description
   - version
   - maintainer
   - license: As option use Apache License 2.0
   - **Dependencies**

2.3 Both C++ and Python

- Non-official documentation
- Create an standard C++ package
- For C++ work normally
- For Python:
  1. Create an import module `__init__.py`. Create it inside the same folder as the custom modules is recommended.
  2. When creating a node **add the shebang** line and make **excecutable**:

```
#!/usr/bin/env python3
```

3. Configure the package file, adding the client and compilation dependencies for C++ and Python.
4. Configure the compilation file: