Coding style

FA296

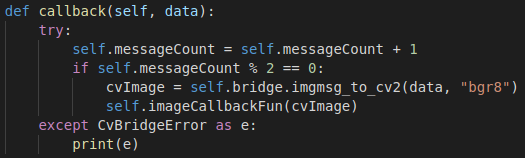
This document will detail the code style rules we have decided to follow. Any amendments to the style document will be included with the reason for the change.

# Which Rules are we following?

Code should be self-explanatory and require as few comments as possible to communicate its functionality. This document is for all code written in Python for the duration of this project.

## Spacing and formatting

* Indentations and tabs should be 4 spaces
* Lines should not exceed 160 characters



## Naming convention

Variable names should be short, concise and descriptive. If a variable name contains multiple words they should be written in camelCase.

Method names follow the same convention and should sufficiently cover the purpose of the method, potentially including the type of data it returns if possible. Method names should also be written in camelCase.

## Exception handling and logging

We will use four different levels of logging:

1. Debug: Debug level log messages should contain technical information that will assist the developer when trying to debug a running system. **Example**: “Module is adding two ints, x and y result: z”.
2. Info: Info level log messages should contain general information about what is happening in the system. **Example**: “Process x started at time: y, because reason: r.
3. Warning: Warning level messages should be used when a potential issue is detected. **Example**: “Field is missing from data, retrying…”
4. Error: Error level messages should be used when an actual issue has occurred. **Example** “Queue subscriber has crashed due to unexpected message. reason: r, message: m, stacktrace: s.”

**Note:** Logging section may only be relevant to ROS sections of code.

## Comments

A comment should be included at the head of each module, this comment should quickly describe the module’s functionality or its purpose in the system.

https://scontent-lht6-1.xx.fbcdn.net/v/t1.15752-9/46525944_922321111309596_6763291954563252224_n.png?_nc_cat=109&_nc_ht=scontent-lht6-1.xx&oh=8d4ef95a8a0e14958b6d7d685193799b&oe=5C745F87

Comments should only be included inline if there is a particularly complicated piece of logic that is not immediately obvious.

## Tests

Any code that contains any form of logic must be covered by a unit test, and any calls that leave the module should have their results mocked by the appropriate mocking tool.