

Ver 1.0

Date 2019/11/5

Ordering:

- All imports go to the very top of the file
- Attributes should be at the beginning of the class, before functions
- Constructors should be the first functions in the class

Indentation:

- Using 4 spaces as 1 indent
- No hard limitation on maximum line length, but if one wants to do so, make sure to align the indentation of the new lines with the opening delimiter or parenthesis, for example

```
def someFunction(variable1, variable2,
                 variable3, variable4):
    dictionary = {
        Dictionary_key: variable1 +
                        variable2
    }
```

Spacing:

- There should be 1 space after every comma in the context of list
- No trailing commas
- Other operands should be surrounded by 1 and only 1 space, with the exception of
  - ++, —
  - : when used in index and only one side has value
  - - when used as negative sign
  - . when used as a decimal point
  - = when used in keyword arguments
- There should be a blank line at the end of every function definition
- No one-line if statements regardless of how short they are, i.e.
  - if poop: doStuff()
- Free to insert blank lines into function body to separate code into blocks if they clearly do different things, i.e. declarations statements and calculation

Naming:

- Classes: Capitalize the first letter of every word
- Functions: Capitalize the first letter of every word except the first word, be as descriptive as possible
- Variables: Capitalize the first letter of every word except the first word, if it's a constant aka the value should never be changed, capitalize all and use underscore to separate words

- Threads: Name all your threads, not the variable, and make sure the name ends with “thread”

#### Comments:

- Comments regarding functions should make it clear
  - What’s this function for
  - What are the parameters if any
  - What if any condition does the function assume about the caller satisfies
  - What does the function return
- Comments regarding variables should make it clear what datatype is it and what is it used for
- Comments regarding a class should make it clear what’s the class for
- Put comments if you think this part of code would be used by someone else and the naming doesn’t make it immediately clear or there’s some specification to take note of before using
- Otherwise do them at your own will

#### Others:

- Try to break long functions into smaller functions if possible
- Be extra clear when naming parameters
- Generalize the function if it can be used for other things
- Reuse code
- Name wrapper function accordingly, i.e.
  - `joinChannel(channel)`
  - `joinChannelWithSocket(channel, socket)`
- No method overloading
- Make sure if you use any thread, the thread is set to daemon, and name your threads