**Code Review**

Code review – an activity in which people other than the author of a software deliverable examine it for defects and improvement opportunities – is one of the most powerful software quality tools available.

Code reviews help novice developers to learn from the senior developers on best practices of writing code.

**Code Review best practices**

* Keep the changes small and focused
* Ensure logical coherence of changes
* Be positive, polite and respectful

**Yard Sticks I used when doing my Code Review**

* Does the new feature add value or is it a sign of feature-creep?
* Is the code properly planned and designed?
* Is the code well organized in terms of placement of components?
* Does the code keep with the idioms and code patterns of the language?
* Does the code make use of the language features and standard libraries?
* Is the code clear and concise?
* Does it comply with PEP-8?
* Are all language and project conventions followed?
* Are identifiers given meaningful and style guide-compliant names?
* Is the code self-documenting or well-documented?
* Is the code free of obfuscation and unnecessary complexity?
* Is the control flow and component relationship clear to understand?
* Is the code free of implementation bugs that could be exploited?
* Have all the new dependencies been audited for vulnerabilities?
* Is the code optimized for in terms of time and space complexity?
* Does it scale as per the need?
* Does it have instrumentation like reporting for metrics and alerting for failures?

**Performing Code Review**

I performed a code review on a github repository (link: <https://github.com/miltonpereira/Scrapper> ). I was able to identify some of the issues with the code and I made suggestions on how the code can be improved following the best coding practices so that people reading the code can easily understand it.

I went further to implement these suggested changes in the code after cloning the github repository. My implemented changes followed an automated python tool called flake8 which enabled me to perform the code tests and follow the PEP8 standards. I finally made a merge request for the suggested changes to be merged into the current repository and make it a better project since it is a public repo.

**Code review changes suggested for the repo**

1. **README** File cannot be empty

Add a short description to help those interested in the project

This can help on directing them on how to run the project.

2. **License** Information missing

Add Licensing used in generation of your code

3. **Naming Convention Structure**

Instead of using *main\_script\_file.py*, it would be conventional to name this file after the project name i.e. *scrapper.py*

4. **Reviewing *requirements.txt***

* + There are lots of packages that you don’t need, out of the 25, the program is using only 3 packages so other are misleading.
  + Considered having a clean and minimal requirements file.

5. **Project Structure**

* The project is not well structured the way python developers are familiar with
* Put ReadME, Requirements
* It would be look familiar if all the project script are put in one folder to separate them from other files in the project.