**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.

**6. Review on Functions**

**1. main\_script\_file.py**

* + The script is well structured into functions
  + The naming is inconsistent with conventional naming of python functions
  + The newlines in the functions make the code hard to read, remove them
  + Inconsistent spacing between functions
  + Using inconsistent tabs in the functions
  + Convert all tabs into spaces as recommended by PEP8
  + Split up the imports splitting the standard python imports from those that are installed(third party)
  + Formatting of the code, spaces between the arguments
  + Inconsistent on the tense used on the naming of the functions
  + Inconsistent assignment(spaces between equal signs)
  + Not clear how these functions work together

**. For Generate\_html function ()**

* + Remove redundant code(*return r.content* instead of html\_*doc = r.content*)
  + Change the name of the function from generate\_html to fetch\_html()
  + Add a brief description of what the function is doing(“””**Fetch base\_url and return Html content**”””)
  + Rename base\_url to url, because base\_url in python convention stands for a prefixed url.

**3. For parse\_and\_scrape\_data()**

* + The name is fine okay as it describes what the system is actually doing.
  + Splitting it into two functions one for parsing and the other for scraping but since it is a short function so let it remain as it is.
  + Remove the redundant variable name and use the loop direct on soup elements.
  + Change the loop iterator variable to singular because we are looking at one item
  + Since we are not creating a dictionary ahead of time, put the title and link into the product\_details{} dictionary. This makes more sense and not declare an empty dictionary which is doing nothing.

4. **Write\_on\_csv() function**

* + Instead of using “data” as the argument for consistent use “product\_details”
  + Make the ‘googleesults.csv’ a default filename to catch scenarios when there is an error in the project operation when it scales up. Alternatively, it could be made a constant in the function.
  + Create constants for the repeated variables that are being reused by more than one function. i.e. FIELD\_TITLE=‘Title’ FIELD\_LINKS=‘Link’ to avoid inconsistency in using these variables throughout the program.

**5. Top level if statement**

* + Since you have a main script, it would be better to add a main function so that if someone runs the code directly on command line using “*python main\_script\_file.py*”, they get to know where really they are.

**6. Recommendations**

* Use formatting tool like flake8 to help in formatting your code and identify inconsistencies that would make your code unreadable to other developers and you yourself after some time lapse.
* Continue with the best practice of making the functions short but also ensure to write python code following the basic python conventions specified in PEP8 standards.
* I have refactored the repository cloned with all the suggested changes and I have made git pull request, if you find it worthy, you can integrate them into your repo.