# Code review for Pooya

The review was conducted on July 28, 2021, using a bottom-up approach, starting from the file, and moving towards the code.

## The File

To begin, the function’s file name does not follow the naming convention found in PEP8; the agreed-on standard for this project. Granted, a simple file naming error may not reflect one’s code quality, however it is important to conform to the project standards.



Figure : File name

Beside the point above, the file naming perfectly describes what the module is expected to carry. It is straight to the point which makes referencing and finding the file very easy.

## The Code

One concern is the file header, as seen in *Figure 2*. When creating or using comment headers, its best to have them contain useful information pertaining to the project or the module; in this case, the header seems to be an auto-generated stub from the IDE. This header does not contribute significant about either the project or code and can be removed or edited to contain pertinent information.

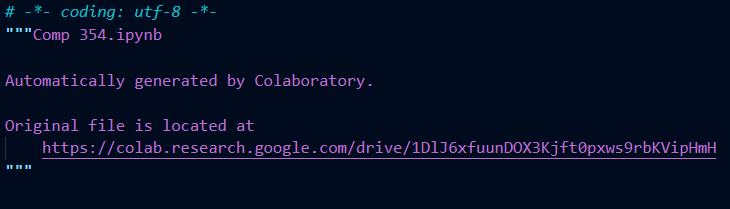


Figure : File header

When writing a function header, it is best to keep the description as concise, but as informative as possible so that anyone who isn’t part of the original development team can understand the function’s capabilities. It is a quick way for a developer to look at the definition, understand what they need to pass the function and what they should expect from it. There are ways to improve upon the function header shown in *Figure 3*. For instance, a description of imported modules usage does not need to be in the header unless it is a parameter for the function. Refer to *Figure 4* for an example of a function header.

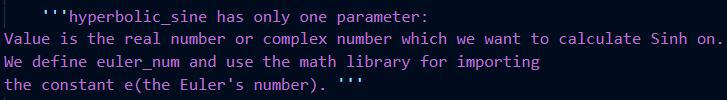


Figure : Function header

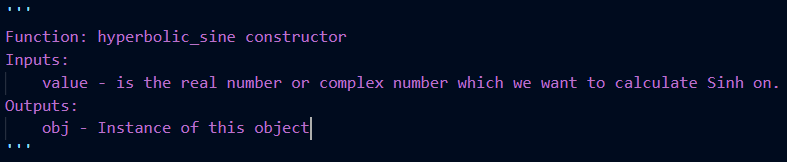


Figure : Updated function header

There are also function headers that are small (see *Figure 5*), yet these more than sufficiently describe their function and therefore serve their purpose.



Figure : Concise function header

As for the implementation of the function and all its ‘helper’ functions, they are concise and serve a useful purpose. The code is well structured, it is very easy to read through and understand and follows the PEP8 naming convention. The implementation of setters/getters is a nice addition in case the user wishes to reuse the object, therefore reducing the amount of program waste. In short, the code is very well written.