|  |
| --- |
| University of Sunderland |
| Python Linting: Cleaning Code with PEP8 Standards |
| An overview of how I’ve used “linters” within my application to improve overall code quality. |

|  |
| --- |
| Heslop, Alan John  11-1-2020 |

### Importance of Code Quality

Code quality is an important aspect in any developer’s scenario, as code can be reused it’s a good principle to clean the quality of the code before the application is in a production environment. There are many plugins and packages that can allow a developer to automatically analyse their code and match to the PEP8 coding standards. One plugin is known as a “linter”

### Linting

Linting is an automated check of the source-code you provide, it will scan for stylistic and other errors.

Below is a screenshot of the WeekOne code, and this is the first time it’s received a linting analysis on the code. It returned 77 errors, and also attached is a



### Text Description automatically generated

### Next Steps to Resolve Errors

In order to fix these known errors and notifications, I had to understand what the issues were, I began by creating a copy of the original python file and renaming it – so I had a local copy on which I could fix. Examples of some issues I encountered are listed below and are not exhaustive.

### Issue examples

### Issue 1:

### Problem:

schoolApp.py:1:0: C0103: Module name "schoolApp" doesn't conform to snake\_case naming style (invalid-name)

### Fix:

Change the name of the python file from schoolApp to schoolapp, this solves the issue.

### Issue 2:

### Problem:

schoolapp.py:31:4: C0116: Missing function or method docstring (missing-function-docstring)

### Fix:

As you see below, I added a docstring using three double quotation marks (“””) to open the docstring, ensuring it was correctly indented, adding a string to explain the function. Then ending the docstring with (“””).

1. def getschoolname(self):  # Defining the method schoolname
2. """Return the sum of x and y."""
3. return self.schoolname

### Issue 3:

### Problem:

schoolapp.py:66:4: R0913: Too many arguments (6/5) (too-many-arguments)

### Fix:

The below example shows I was able to fix the issue; however I used an inbuilt service in which I ask the linter to ignore the “too many arguments” message.

class Child(School):  # Child class is using inheritance from the School Class

    """Child class stores data about the child"""

    # pylint: disable-msg=too-many-arguments

    def \_\_init\_\_(self, name, age, yeargroup, schoolname, location):

### Code after cleaned

The below screenshot shows the result after I had cleaned the code and aligned to the PEP8 Standards.

Note: I changed the name of the python file to separate it from the previous file.

A screenshot of a computer screen

Description automatically generated