PYTHON STANDARED CODING CONVENTIONS

1.Indentation

In python the Use of 4 spaces per indentation level. its a basic convention while writing the code. When programming in Python, indentation is something that you will definitely use. However, you should be careful with it, as it can lead to syntax errors. The recommendation is therefore to use 4 spaces for indentation.

2. Comments

Comments are used for in-code documentation in Python. They add to the understanding of the code. There are lots of tools that you can use to generate documentation, such as comments and docstrings, for your own module. Comments should be more verbose so that when someone reads the code, the person would get the proper understanding of the code and how it is being used with other pieces of the code.

Block Comments

Block comments generally apply to some (or all) code that follows them, and are indented to the same level as that code. Each line of a block comment starts with a # and a single space (unless it is indented text inside the comment).Paragraphs inside a block comment are separated by a line containing a single #.

Inline Comments

Use inline comments sparingly.

An inline comment is a comment on the same line as a statement. Inline comments should be separated by at least two spaces from the statement. They should start with a # and a single space.

3. Blank Lines

In Python scripts, top-level function and classes are separated by two blank lines. Method definitions inside classes should be separated by one blank line

4. **Maximum Line Length**

Limit all lines to a maximum of 79 characters.For flowing long blocks of text with fewer structural restrictions (docstrings or comments), the line length should be limited to 72 characters.

5. Imports

- Imports should usually be on separate lines.
- Imports are always put at the top of the file, just after any module comments and docstrings, and before module globals and constants.

- Imports should be grouped in the following order:

  Standard library imports.

  Related third party imports.

  Local application/library specific imports.

  You should put a blank line between each group of imports.

Absolute imports are recommended, as they are usually more readable and tend to be better behaved (or at least give better error messages) if the import system is incorrectly configured (such as when a directory inside a package ends up on sys.path):

White Space

- immediately inside parentheses, brackets or braces
- Between a trailing comma and a following close parenthesis.
- Immediately before a comma, semicolon, or colon
- Immediately before the open parenthesis that starts the argument list of a function call.
- Immediately before the open parenthesis that starts an indexing or slicing
- More than one space around an assignment (or other) operator to align it with another

Formatting

• Use double quotes(") around strings that are used for interpolation or that intended for the end-user to see, otherwise use single quotes(').

• To format strings use format function or if you using Python≥3.6 use f-strings

• Always start a new block on a new line

PYTHON STANDARED NAMING CONVENTIONS

1. General

- Avoid using names that are too general. Strike a good balance between the two.
- When using CamelCase names, capitalize all letters of an abbreviation (e.g. HTTPServer)

2. Packages

- Package names should be all lower case
- When multiple words are needed, an underscore should separate them
- It is usually preferable to stick to 1 word names

3. Modules

- Module names should be all lower case
- When multiple words are needed, an underscore should separate them
- It is usually preferable to stick to 1 word names

4. Classes

- Class names should follow the UpperCase, CamelCase convention
- Python's built-in classes, however are typically lowercase words
- Exception classes should end in "Error"

5. Global Variables

- Global variables should be all lowercase
- Words in a global variable name should be separated by an underscore

6. Instance Variables

- Instance variable names should be all lower case
- Words in an instance variable name should be separated by an underscore

7. Methods

- Method names should be all lower case
- Words in an method name should be separated by an underscore

8. Method Arguments

- Instance methods should have their first argument named 'self'.

9. Functions

- Function names should be all lower case
- Words in a function name should be separated by an underscore

10. Constants

- Constant names must be fully capitalized.
- if there are words need to be sepreated by underscore