A vital part of coding projects is maintaining clean, understandable, and consistent code. Two important elements of this are how we name our variables and functions, and how we document them. The PEP 8 style guide for Python code gives good recommendations in these areas. A couple of examples of this include the use of snake\_case for identifiers and triple-double-quoted docstrings for functions. There are solid reasons to follow these conventions.

PEP 8 strongly recommends that all variable names and function names use lowercase words separated by underscores, also known as snake\_case. This style is beneficial because it naturally separates words, making code much easier to read and understand at a glance. When you see a function named process\_order\_details, its purpose is immediately clear. Furthermore, applying the same naming scheme across variables and functions promotes consistency and reduces confusion, especially in larger codebases. This uniformity is particularly important when working in teams because it helps everyone follow the same structure, allowing code reviews to focus on logic and functionality rather than style differences. While snake\_case is preferred, PEP 8 does recognize that some legacy modules may use mixedCase. In such cases, it is best to maintain the existing style to preserve compatibility while still striving for clarity.

Documenting code behavior, usage, and intent is just as important as writing the code itself. PEP 8 provides specific guidelines for function documentation through docstrings. A docstring should appear immediately after the def statement as the first indented line within a function. It should be enclosed in triple double-quotes (""" ... """) whether it is a single line or multiple lines. This guideline is important for many reasons. Docstrings are accessible at runtime through the function’s \_\_doc\_\_ attribute, meaning they can be used by tools such as help() and pydoc to generate dynamic documentation. Using a consistent quoting style makes it easier for documentation tools to parse and process the code automatically. Well-structured docstrings also enhance readability by providing a clear explanation of what a function does, what parameters it takes, and what it returns. This level of documentation is crucial for both current and future developers who need to understand and maintain the code.

Adopting snake\_case for naming variables and functions ensures clarity and consistency, making Python code more intuitive and maintainable. Complementing this with triple-double-quoted docstrings creates a strong foundation for in-code documentation that supports both developers and automated tools. Together, these practices align with PEP 8 recommendations and foster professional, maintainable, and team-friendly Python development. By following these guidelines, developers can create code that is not only functional but also easy to read, share, and improve over time.

https://peps.python.org/pep-0008/#introduction