Python Coding Standards

Occupi - Office Capacity Predictor

Name	Student Number
Rethakgetse Manaka	u22491032
Kamogelo Moeketse	u22623478
Michael Chinyama	u21546551
Tinashe Austin	u21564176
Carey Mokou	u21631532

September 01, 2024

Contents

Introduction

This document outlines the Python coding standards to ensure consistent, readable, and maintainable code. Following these standards will help all contributors write clean, efficient, and bug-free Python code.

General Code Structure

All code should be clean, readable, and maintainable. Key practices include:

- Using meaningful names for variables, functions, and classes.
- Follow consistent indentation with 4 spaces per level.
- Organize imports at the top of each file, grouping standard library imports, third-party imports, and local imports separately.

PEP 8 Compliance

All Python code must follow the PEP 8 style guide:

- Line length should not exceed 79 characters.
- Use spaces around operators (e.g., ‘x = 5‘, not ‘x=5‘).
- Include a blank line between function definitions and after class declarations.
- Avoid unnecessary blank lines inside functions.

Error Handling and Logging

Ensure code is resilient and provides meaningful error messages:

- Use try-except blocks for any potentially unsafe operations.
- Always log errors using the ‘logging‘ module rather than ‘print‘.
- Catch specific exceptions, not generic ones (e.g., ‘except ValueError‘ instead of ‘except‘).

Functions and Methods

Functions should be small, focused, and easy to understand:

- Use descriptive names for functions and methods.
- Include type hints for function arguments and return values where appropriate.
- Each function should accomplish one specific task.
- Comment functions and methods with a brief description of what they do.

Code Comments and Documentation

All public functions, classes, and methods must have docstrings:

- Use multi-line docstrings for functions and classes.
- Use inline comments sparingly and only when the code is non-obvious.
- Document any external libraries or APIs used.
- Write descriptive comments that explain the why, not the how.

Code Example

Example of a well-documented Python function:

```python
def get_predictions(data: Dict[str, Any]) -> List[float]:
    """
    Fetch predictions from the model.

    Args:
        data (dict): Input data for the model.

    Returns:
        list: Model predictions as a list of floats.
    """
    return model.predict(data)
```

Testing and Debugging

Ensure that all critical functions are tested and debugged properly:

- Write unit tests using ‘unittest‘ or ‘pytest‘ for each function.
- Use mock data where necessary, especially when handling external services.
- Use assertions in tests to validate function outputs.

References

For more details on Python coding standards, please refer to:

- PEP 8: Python Enhancement Proposal 8 - Style Guide for Python Code
- Python Documentation: Official Python Documentation