■ Python Style Guide

1. File Header

Every Python source file must begin with a standardized header comment containing key metadata.

```
"""
File Name:    example_module.py
Author(s):    Ju-ve Chankasemporn
Copyright:    (c) 2025 DigiPen Institute of Technology. All rights reserved.
"""
```

■ Place this header before all import statements.

2. Imports

Follow this import order:
- Standard library imports
- Third-party imports
- Local application imports

```
import os
import sys


import numpy as np
import cv2


from my_project.utils import helper
```

3. Naming Conventions

```
Modules / Files  : lowercase_with_underscores   → data_loader.py
Classes          : PascalCase                   → class PlayerController:
Functions        : lowercase_with_underscores   → def calculate_damage():
Variables        : lowercase_with_underscores   → player_health = 100
Constants        : ALL_CAPS_WITH_UNDERSCORES     → MAX_SPEED = 10
Private Members  : prefix with _                → _internal_cache
```

4. Code Layout

- Indentation: 4 spaces per indentation level (no tabs).
- Line Length: Limit lines to 79 characters.
- Blank Lines: Use 2 between top-level functions/classes, 1 between methods in a class.

5. Whitespace Rules

Use a single space after commas, colons, and around operators.

```
result = a + b
my_list = [1, 2, 3]
```

6. Functions and Classes

Always use docstrings and type hints for all public functions and classes.

```
def compute_distance(x, y):
    """Compute Euclidean distance between two points."""
    return (x ** 2 + y ** 2) ** 0.5
```

```
def add(x: int, y: int) -> int:
    return x + y
```

7. Comments

Use '#' for inline or block comments. Keep them short, clear, and up to date.

```
# Initialize player position at origin
player_pos = (0, 0)
```

8. Error Handling

Use exceptions rather than return codes.

```
try:
    process_data()
except ValueError as e:
    print(f"Invalid input: {e}")
```

9. Logging and Debugging

Prefer 'logging' over 'print()' for debug or info output.

```
import logging
logging.basicConfig(level=logging.INFO)
logging.info("Game started successfully")
```

10. Version Control and Licensing

All code files must include the file header (see Section 1). Commit messages should be clear and imperative.

- "Add A* pathfinding algorithm"
- "Fixed stuff"