Day_flow’s Python Language Coding Standards

Last update: 10 November 2023

Comment Rules

Comment Rules

1. Every source file (.c and .h files) must include a comment at the beginning of the file that briefly describes the purpose of the source file and which lists the author and the date. For example:

```
“””
main_ui.py
The main user interface program for the day_flow project
including these features
-Search for events by their title – show all matches with date, time
-Display month, week, and day display
-Edit and delete event
-Export events for a specified date range in CSV format
most functionalities involve a lot on user interfaces

Created by Roong 19 October 2023
“””
```

2. Every function, except the main function, must be preceded by a block comment that explains
a) the purpose of the function
b) the purpose of each argument to the function except for the self argument
c) what the function returns (if anything). For example:

```
“”” Calculate and print the change to be returned for
a supermarket order.
Arguments:
        totalBill – the total the customer must pay
        paidAmount - the amount the customer gave the cashier
Assume the calling program has already checked that the
paid amount is sufficient to cover the bill.
“”””
calculate_change(total_bill, paid amount)
{
...
}
```

3.Use comments within a function only when you feel that the code alone is not self-explanatory. More comments does not usually mean better style. For example:

```
# we will stop when the user enters
While ((value >= 0) and (count < MAXVALS))
```

4.Use comments all variables

Naming Rules

Naming conventions for variables, functions, and classes.
-Variables and functions: **lowercase_with_underscores** (snake_case).
-Constants: **UPPERCASE_WITH_UNDERSCORES**.
-Classes: **CapWords** (CamelCase).

CamelCase
Class SumOfSquares
or Class sumOfSquares

snake_case
sum_of_squares = 0
final_velocity

Constants
ROW = 30
COLUMN = 25

-Avoid Using Single Letters for Variables:
Use descriptive variable names. Except for i, j, and k in the loop.
Poor Style
n = 10

Better Style
item_count = 10

-Avoid really long variable names, which are also difficult to understand. For example:
Poor style
int residualIncomeAfterDeductions; // better: residual_income
double bahtToDollarExchangeRate; // better: baht_dollar

Maximum Line Length

Limit lines to **79** characters for code and **72** for docstrings/comments. You can go up to **120** characters if necessary, but try to keep it readable.

Multiple Imports

Multiple Imports should usually be on separate lines.

For example:

```
import sys
import datetime
import calendar
```

Always on the Top

Imports are always put at the top of the file i.e.

After any module comments and docstrings.

Before module globals and constants.

For example:

```
import sys
import datetime
import calendar


if __name__ == "__main__":
    pass
```