

# Coding Standards – No-file-found

Last Update: 28 September 2023

## Comment Rules

1. Every Source file (.py, .c, and .h) 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:

```
'''
findMin.py

Demonstration program for Intro to Programming
Asks for three numbers x,y and z, then figures out
which one is smallest and prints the result.

Created by Erling Haaland, 23 September 2029
'''
```

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; 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.
'''
def calculateChange(totalBill, paidAmount):
```

3. Variables that are crucial to the program should have a comment explaining the usage, or variables that are not clearly defined. For example:

```
# list storing all search history queries
searchHistory = []
```

4. 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:

```
# Print numbers from 0 to 9
for i in range(0, 10):
    print(i)
```

## Naming Rules

1. Names for variables (as well as functions and other identifiers) should be meaningful, i help the reader understand the purpose of the variable. For example:

```
fileCount = 0
filePath = ""
```

2. Variable names should start with lowercase letters.
3. Class names should start with uppercase letters.
4. Constant and global variables should be all uppercase letters.
5. For variable names that consist of multiple words, you have to use Camel Case. For example:

```
sumOfNumbers = 0
listOfName = []
```

6. Folder and file names must all be lowercase with underscores to separate words and also must have a meaningful name.

For example:

```
File_list_indexer.py
```

7. Avoid using abbreviations for variable names unless it is commonly understood.

For example:

**Bad style:**

```
dob = "12/01/12"
```

**Good style:**

```
dateOfBirth = "12/01/12"
```

## Coding Considerations

1. When using try Exception you must have an exception type.

```
try:
    with open("file not found", w) as file:
        .....
except FileNotFoundError as e:
    .....
```

2. When you are writing a function that must work with specific variable types ie: int, str, dict etc... you must set the type of the variable

For example:

```
def addNumber(a: int, b: int):
    return a + b
```