Zach Harvey
11/22/2023
IT FDN 110 A
Assignment 6
https://github.com/harv-uw/python110_spring2023

Assignment 6 - Classes and Functions

Introduction

In this module, we started to organize our program with classes and functions. These provide better encapsulation, reusability, and readability of code. Classes and functions are essential in creating well-constructed scripts.

Classes

In this assignment, we defined two classes, IO and FileProcessing. These classes describe the broad functionality of the program and divide it into separate parts. Each class with its functions is defined before being invoked in the program. Additionally, it is best practice to include a DocumentString to describe the functionality of the class and when it was created or updated. These work well with IDEs like PyCharm as it will present the information when you hover over an invocation. The image below shows how the FileProcessor Class was defined.

```
# Processing
class FileProcessor:
    """
    A collection of processing layer functions that work with JSON files
    ChangeLog:
    Zach Harvey,11,22,2023,Create Class
    """
```

Image 1: Definition of the FileProcessor class with docstring

Functions

Functions are reusable blocks of code that perform a specific thing. They can be grouped in two ways, standalone functions or those that are defined on a class which are called methods. Functions can accept parameters as inputs which are used in the function's code. These can be of specific types which are defined in the signature of the function. Similar to classes, it is also best practice to use a docstring when defining a function in order to provide information when using it. Additionally, functions can be annotated with decorators that modify how the function is used. In this module, we use the "@staticmethod" decorator so we don't have to instantiate the class to call the method. The image below shows a method that was defined on the FileProcessor class.

```python
@staticmethod
def read_data_from_file(file_name: str, student_list: list):
    """
    This function reads student data from a given file
    ChangeLog:
    Zach Harvey,11,22,2023,Create Function
    :return: None
    """
    try:
        file = open(file_name, "r")
        student_list = json.load(file)
        file.close()
    except FileNotFoundError as e:
        IO.output_error_messages( message: "JSON file must exist before running this script!", e)
    except Exception as e:
        IO.output_error_messages( message: "A non-specific error occurred", e)
    return student_list
```

Image 2: Definition of a method on the FileProcessor class

As you can see, this method takes two parameters in order to use them in the function body. We can also call other functions within functions themselves as we do for error handling here. Finally, we return a value at the end of the function to be captured where the function is called. This can be seen in the image below.

```python
students = FileProcessor.read_data_from_file(FILE_NAME, students)
```

Image 3: Using the read_data_from_file method in the program

As you can see, the returned value of the method is captured as the global variable “students”. Additionally, “FILE_NAME” and “students” are passed into the function in order to be used by it. In this case, the variable “students” is overwritten when the function returns.

Summary

Classes and functions are powerful tools when writing programs. They not only provide a means of organizing your code but result in cleaner and easier-to-read programs. Also, because functions can be reused, there are fewer places for human error or bugs.