Jordan Sellers

2/21/2024

Foundations of Python: Programming

Assignment 06

https://github.com/jordan-aloysius

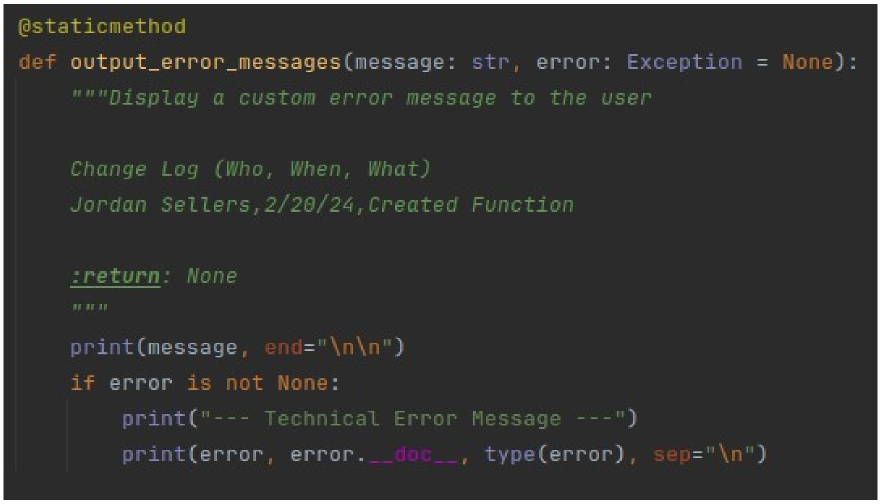
# Module 6: Functions

## Introduction

Functions are a way to organize code that saves sections of code into a single unit. They can include parameters, which specify the kinds of variables that will be used. A function can be used multiple times throughout a program, making the program more concise. Functions can further be organized into classes, categorizing them by type (Separation of Data).

## Functions

A function contains a block of code within a program. The function is given a name, which can be used, or *called*, when needed. This eliminates the need to write the same code multiple times if the function appears multiple times within a program. The function is defined as the block of code it represents. This begins as *def*, followed by the name of the function, then parentheses, and at the end a colon. The rest of the code appears indented below [figure 1].



```
@staticmethod
def output_error_messages(message: str, error: Exception = None):
    """Display a custom error message to the user

    Change Log (Who, When, What)
    Jordan Sellers,2/20/24,Created Function

    :return: None
    """
    print(message, end="\n\n")
    if error is not None:
        print("--- Technical Error Message ---")
        print(error, error.__doc__, type(error), sep="\n")
```

*Figure 1: A function is defined at the start of the program. It includes a* docstring, *which has explanatory notes for others reading/editing the script, as well as a change log. Note the use of decorator* @staticmethod *before the name of the function. This indicates that the function is* static, *meaning it never changes.*

## Parameters and Arguments

The parentheses are used for parameters. A parameter stands in for variables that will be used within the function, and appear as a name (which resembles a variable) and the type of data the variable will be (string, integer, etc) separated by a colon. Functions can contain more than one parameter, with each separated by a comma. The specific variables that will be used, called *arguments*, are specified when the function is called.

## Classes and Separation of Data

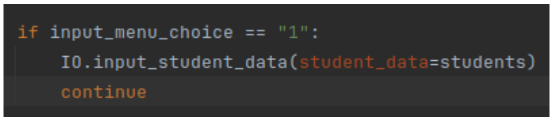
Functions are defined within a program before the main body of script and after the constants and variables are defined. Defining functions at the beginning means they do not need to be written as they are used, or in the order that they appear within the main body of the script. This allows them to be organized in a way that is logical and consistent, making the code much easier for others to read. This is called Separation of Data (SoD).

In SoD, *classes* are used to group functions according to the roll they play in the program. Classes include *FileProcessor* (functions used in manipulating files) and IO (Input/Output—functions used in inputting or outputting data). Each class is listed as a heading before its respected functions are defined, and is used when calling functions within the main body of the script.

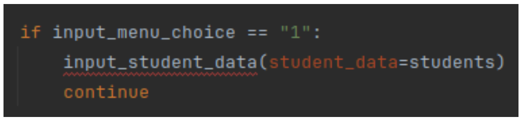
## Calling Functions

Using a function in the main body of the script is referred to as *calling the function*. The name of the function is written out, and includes the variables (*arguments*) used in the function. The arguments come in parentheses after the name of the function, and are written after their respected parameters with an = sign in between. If there are multiple parameters/arguments, they are separated by commas.

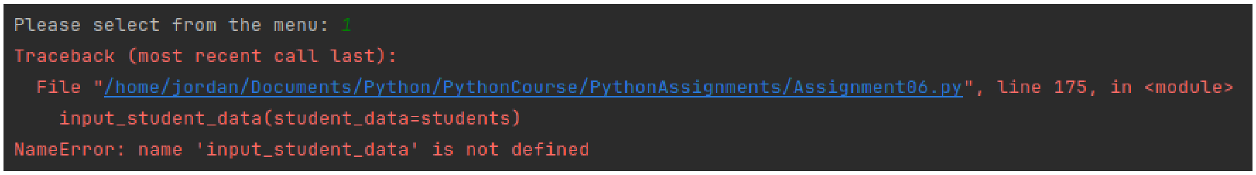
An important thing to remember is to include the class when calling the function. The class comes first, followed by a period and the rest of the function. The class tells the program where in the script the function is referenced. Failing to specify the class results in and error [Figure X], because Python regards the function as being undefined.



```
if input_menu_choice == "1":
    IO.input_student_data(student_data=students)
    continue
```



```
if input_menu_choice == "1":
    input_student_data(student_data=students)
    continue
```



```
Please select from the menu: 1
Traceback (most recent call last):
  File "/home/jordan/Documents/Python/PythonCourse/PythonAssignments/Assignment06.py", line 175, in <module>
    input_student_data(student_data=students)
NameError: name 'input_student_data' is not defined
```

*Clockwise from upper left: The class always needs to be indicated when the function is called (figure 2). Calling the function without the class (figure 3) results in an error (figure 4).*

## Summary

Functions reuse code used more than once in a program. They are grouped in to classes, which organizes the function according to what they do. Separation of Data makes the script more concise.