Sun-Hae Kim
November 21 2023

IT FDN 110 A Au 23 - Foundations Of Programming: Python
Assignment 06 - Functions

# Module 06 - Functions

In my programming practice, I heavily rely on functions as reusable blocks of code, emphasizing their role in enhancing modularity and promoting code reuse. Breaking down a large program into manageable functions allows me to concentrate on specific tasks, leading to code that is more understandable, maintainable, and debuggable.

The use of parameters in functions has become a central aspect of my coding approach. Moving away from global variables, I now declare parameters within functions to encapsulate necessary data. This practice not only enhances code organization but also fosters flexibility and predictability. Encapsulation ensures clarity on data requirements, reducing reliance on external variables and mitigating potential errors and side effects. Parameters make functions more adaptable to various contexts, supporting better code reuse.

Understanding the distinction between mutable and immutable objects when passing arguments has been crucial. This awareness enables me to handle objects effectively within functions, considering factors such as object mutability and the impact of modifications on original data.

In addition to functions, I've embraced the organizational benefits of classes in Python. Classes provide a structured way to group functions, variables, and constants, facilitating a modular code structure. The separation of concerns pattern, dividing code into distinct layers (Data, Processing, and Presentation), has been instrumental in managing complexity in larger projects. Objects created from classes act as independent copies, and the use of "self" in class functions signifies their autonomy.

Static methods in classes have proven efficient for functions that don't require object instances, enhancing code clarity and execution. Adopting docstrings for classes and functions has become a best practice, offering valuable insights into their purpose and usage. This documentation practice is particularly beneficial as code complexity increases.

In summary, my coding approach revolves around the effective use of functions, parameters, and classes. Functions promote modularity and reusability, parameters enhance organization and flexibility, and classes provide a structured framework for code organization. Embracing these concepts, along with the separation of concerns pattern, contributes to writing more maintainable, scalable, and well-documented code.