Alyanna Marie N. Bilazon
CS3C

Functions

1. **Defining a Function -** In a Python, a function is defines using the **def** keyboard followed by the function name and parentheses **()**. Inside the parentheses, you can include parameters that the function will use, the function's code block starts with the colon **:** and is indented.
2. **Reasons of Using Functions -** Functions assist in segmenting our program into more manageable and modular parts. Functions help our program become easier to use and controllable as it gets bigger and bigger. They make the code reusable and prevent repetition.
3. **Types of Functions in Python -** Built-in Functions: Pre-defined functions provided by Python, like print(), len(), etc.
    - User-defined Functions: Functions defined by the user to perform specific tasks.
4. **Advantages of User – Defined Function -** They assist in breaking up large programs into manageable chunks, which facilitates comprehension, upkeep, and debugging of the program. If a program contains repetitive code, the function can be used to include that code and call it when necessary.
5. **Rules in Declaring a Function in Python -** Function blocks begin with the keyword **def** followed by the function name and parentheses **()**. Any input parameters or arguments should be placed within these parentheses.
6. **Python Function Syntax -**

```
def function_name(parameters):
    """docstring"""
    # Function body
    return value(s)
```

7. **Function Argument and Parameter -** Parameters are the variables that are defined or used inside parentheses while defining a function, whereas arguments are the value passed for these parameters while calling a function.
8. **The Return Statement -** The **return** statement is used to exit a function and go back to the place from where it was called. It can include an expression which gets evaluated and the value is returned. If there is no expression in the statement or the **return** statement itself is not present inside a function, then the function will return the **None** object.