Function

User-defined functions in Python are blocks of code that you create to perform specific tasks. They help you organize your code, make it more reusable, and improve readability. Here are some key points to remember about user-defined functions in Python:

- **Definition:** You define a function using the `def` keyword followed by the function name and parentheses. Optionally, you can specify parameters (arguments) within the parentheses. The function body, which contains the code to be executed, is indented after the colon.
- **Parameters:** Parameters are variables that act as inputs to the function. They allow you to pass data to the function when you call it.
- **Return Values:** Functions can optionally return a value using the `return` statement. The returned value becomes the output of the function call.
- **Calling a Function:** You call a function using its name followed by parentheses. You can pass arguments to the function within the parentheses.

Here's a basic example of a user-defined function in Python:

Python
```python
def greet(name):
  """This function greets the user by name.

  Args:
      name: The name of the user.

  Returns:
      A string greeting the user.
  """
  print("Hello,", name + "!")

greet("Alice")  # Output: Hello, Alice!
```

In this example, the `greet` function takes a parameter `name` and prints a greeting message.

Here are some advantages of using user-defined functions:

- **Reusability:** You can call the same function multiple times with different arguments, avoiding code duplication.

- **Readability:** Functions improve code readability by encapsulating specific tasks within named blocks.
- **Modularity:** Functions break down complex programs into smaller, manageable units.

Exercise:

1. Write a Python program that defines a function `print_current_date` that displays the current date on the screen.
2. Write a Python program that defines a function `play_greeting_message` that plays a pre-recorded greeting message.
3. Write a Python program that defines a function `display_user_options` that presents a menu of options to the user on the screen.
4. Write a Python program that defines a function `simulate_dice_roll` that simulates a dice roll and prints the result (between 1 and 6) on the screen.

Class work:

1. Write a Python function that takes no parameters and prints the Fibonacci sequence up to a specified limit.

2. Create a Python function that generates a random password of a given length.

3. Implement a Python function to check if a given number is prime without using any built-in functions except for basic arithmetic operations.

4. Write a Python function that accepts a list of integers and returns the sum of all even numbers in the list without using any built-in functions like `sum()` or list comprehensions.

5. Develop a Python function that prints the frequency of each character in the string as a dictionary, excluding spaces and punctuation.