Python Functions

In Python, a function is a block of organized, reusable code that performs a specific task. Functions help in modularizing code, making it more readable, and promoting code reuse.

1. **Defining a Function:** We can define a function using the ‘ def ‘ keyword, followed by the function name and parameters. The function block is indented.

```
def my_function(parameter1, parameter2):

    # Function body

    result = parameter1 + parameter2

    return result.
```

2. **Calling a Function:** To call a function, use the function name followed by parentheses and provide any required arguments.

```
result_value = my_function(10, 20)

print(result_value)
```

3.Parameters and Arguments:

- **Parameters**: These are variables listed in the function definition. They act as placeholders for the actual values that will be passed into the function.

- **Arguments:** These are the values passed into the function when it is called.

4. Return Statement:

A function can return a value using the return statement. If no return statement is present, the function returns None by default.

5. Default Parameters:

We can assign default values to parameters, making them optional during function calls.

```
  def greet(name, greeting="Hello"):

print(f"{greeting}, {name}!")

greet("John")                              # Prints "Hello, John!"

greet("John", "Good morning")              # Prints "Good morning, John!"
```

6. Variable-Length Arguments:

We can use *args and **kwargs to handle variable-length arguments (arbitrary numbers of positional and keyword arguments).

```
def variable_args(*args, **kwargs):

    print("Positional arguments:", args)

    print("Keyword arguments:", kwargs)

variable_args(1, 2, 3, a='apple', b='banana').
```

7. Lambda Functions:

We can create anonymous functions using the `lambda` keyword.

```
multiply = lambda x, y: x * y

print(multiply(2, 3))           # Prints 6.
```

8.**Docstrings:**

Document your functions using docstrings to provide information about the function's purpose, parameters, and return values.

```
def square(number):

    """

    This function squares the input number.

    Parameters:

    - number (int): The input number.

    Returns:

    - int: The square of the input number.

    """

    return number ** 2
```