In Python, a function is a block of reusable code that performs a specific task. Functions help organize code, avoid repetition, and improve readability.

Types of Functions in Python

1. Built-in Functions

   Already provided by Python.
   Examples: `print()`, `len()`, `max()`, `sum()`, `type()`, `input()`.

```
print(len("Python"))  # 6
print(sum([1, 2, 3]))  # 6
```

2. User-defined Functions
   Created by the programmer using the `def` keyword.

```
def greet(name):
    return f"Hello, {name}!"

print(greet("Athar"))  # Hello, Athar!
```

3. Anonymous (Lambda) Functions

   Defined using the `lambda` keyword (small, one-line functions).

```
square = lambda x: x * x
print(square(5))  # 25
```

Defining a Function in Python

Syntax:

```
def function_name(parameters):
    # function body
    return value
```

Example:

```
def add(a, b):
    """This function returns the sum of two numbers."""
    return a + b
print(add(5, 3))  # 8
```

Function Arguments in Python

Python functions can accept different types of arguments:

1. Positional Arguments

```
def multiply(a, b):
       return a * b

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

2. Default Arguments

```
def greet(name="Guest"):
       return f"Hello, {name}"

  print(greet())         # Hello, Guest
  print(greet("Athar"))  # Hello, Athar
```

3. Keyword Arguments

```
   def divide(a, b):
       return a / b

print(divide(b=2, a=10))  # 5.0
```

4. Arbitrary Arguments (`*args`) → for multiple values (tuple)

```
 def total(*numbers):
       return sum(numbers)

   print(total(1, 2, 3, 4))  # 10
```

5. Arbitrary Keyword Arguments (` kwargs`) → for key-value pairs (dictionary)

```
   def person_info( details):
       return details

   print(person_info(name="Athar", age=35, country="Qatar"))
   # {'name': 'Athar', 'age': 35, 'country': 'Qatar'}
```

Built-in functions (ready to use, e.g., `len`, `print`)
User-defined functions (`def myfunc():`)
Lambda functions (one-liners using `lambda`)
Functions can take **positional, default, keyword, *args, and kwargs arguments.