

Programming and Problem Solving through Python Language

O Level / A Level

Chapter - 6 : Functions

Arguments or Parameters

- Information can be passed into functions as arguments.
- Arguments are specified after the function name, inside the parentheses.
- We can add many arguments separating them with a comma.

Function Arguments

A function can be called by using the following types of formal arguments –

- Required arguments
- Keyword arguments(**kwargs**)
- Default arguments
- Variable-length arguments or Arbitrary Arguments(***args**)
- Arbitrary Keyword Arguments (****kwargs**)

Required arguments

- Required arguments are the arguments passed to a function in correct positional order.
- The number of arguments in the function call should match exactly with the function definition.

```
# Function definition is here
def printme( name, age ):
    "This prints a passed string into this function"
    print (name , age)
```

```
# Now you can call printme function
printme("Ajay",30)
```

To call the function **printme()**, it is definitely need to pass one argument, otherwise it gives a syntax error.

Keyword Arguments (*kwargs*)

- Keyword arguments in a function call, allows to identifies the arguments by the parameter name.
- This allows us to skip arguments or place them out of order. The interpreter is able to use the keywords provided to match the values with parameters.

```
# Function definition is here
def printme( name, age ):
    "This prints a passed string into this function"
    print (name , age)

# Now you can call printme function
printme(age=30, name="Ajay")
```

Default Arguments

- A default argument is an argument that assumes a default value if a value is not provided in the function call for that argument.

```
# Function definition is here
def printme( name, age=40 ):
    "This prints a passed string into this function"
    print (name , age)

# Now you can call printme function
printme(age=30, name="Ajay")
printme(name="Ajay")
```

Output

```
Ajay 30
Ajay 40
```

Return Values

- If a function return a value, use the return statement.

```
def my_function(x):
    return 5 * x

print(my_function(3))
print(my_function(5))
```

The pass Statement

A function definitions cannot be empty, but if you for some reason have a function definition with no content, put in the pass statement to avoid getting an error.

```
def myfunction():
    pass

a=10
print("Value=", a)
```