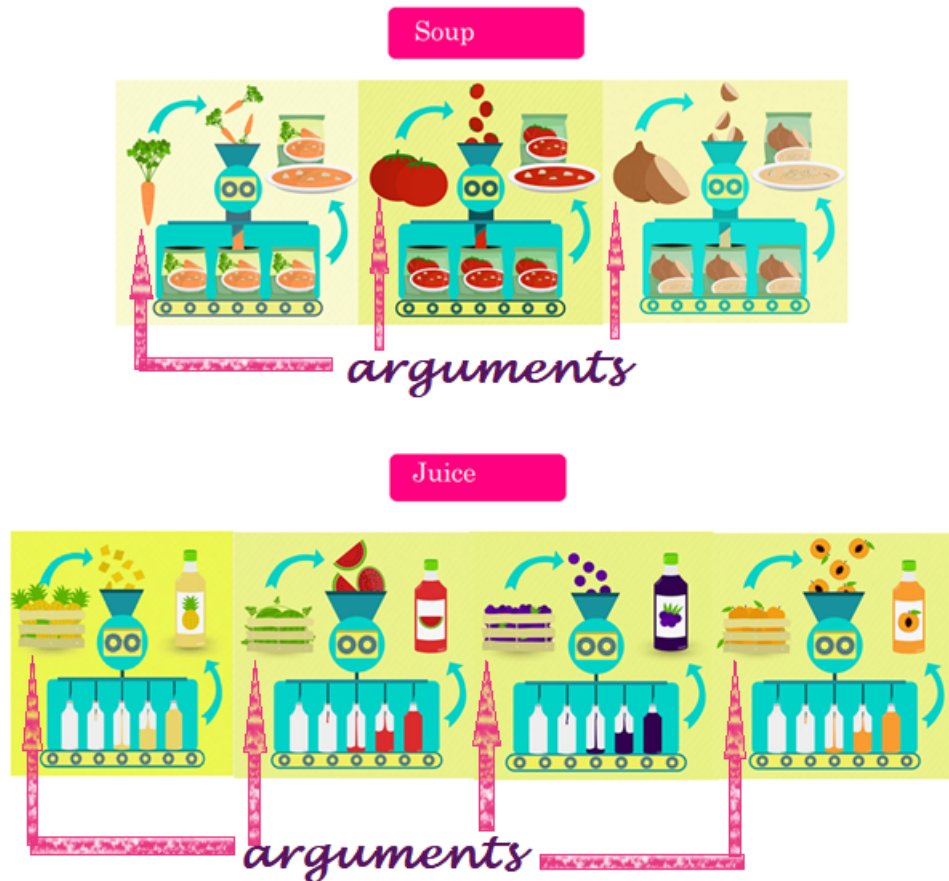


PARAMETER AND ARGUMENTS

Arguments appear in **function call** statements and are used to **send value** from **calling function to called function**. It is also known as actual parameter or actual argument

Parameter appears in **function header** and used to **receive value** in called function. It is also known as formal parameter or formal argument.



```
def square(x):
```

```
    a = x ** 2
```

```
    return a
```

```
n = 5
```

```
print( square(n))
```

output:

25

here x is parameter and n is argument

TYPE OF ARGUMENTS /PARAMETER

- Positional arguments
- Keyword argument
- **Default arguments**
- variable length argument

POSITIONAL ARGUMENT

Positional arguments are arguments to be passed in **correct positional order**. It is also known as a **required or mandatory** argument as **no value can be skipped** or cannot change the order.

When the function call statement must **match the number and order of arguments** as defined in the function definition, this is called positional argument matching

```
def fun(x,y):
```

```
    s=x-y
```

```
    print(s)
```

```
a=10
```

```
b=20
```

```
fun(a,b)
```

output:

-10

Note:- here the value of a that is **10** will be assigned to x and the value of b that is **20** will be assigned to y.

a and b are example of positional argument

KEYWORD ARGUMENT

Keyword arguments are **provided by using name** instead of position (order). The **order of argument is not important** but the number of arguments must be matched.

```
def fun(x,y,z):
```

```
    s=x-y+z
```

```
    print(s)
```

```
a=10
```

```
b=20
```

```
c=30
```

```
fun(c,z=b,y=a)
```

output:

40

Note:- here the **value of c** that is 30 will **be assigned to x** and the **value of b** that is 20 will be **assigned to z** and the value of a that is 10 will be assigned to y.

c is passed as a **positional argument**. **a and b** passed as a **keyword argument**

DEFAULT ARGUMENT

A parameter having **default value in the function header** is known as default parameter / argument. The default values for parameters are considered only if no value is provided for that parameter in the function call statement.

```
def fun(x,y=10):
```

```
    s=x-y
```

```
    print(s)
```

```
fun(30,15)
```

```
fun(30)
```

output:

15

20

Note:- here y is an example of a default parameter. if the value of y is not passed in the function call statement, it will take the default value. In the first function call value for y is passed that is 15 but in the second function call value of y is not passed so it will take the default value.

VARIABLE LENGTH ARGUMENT

In certain situations, we need to pass a variable number of arguments. This type of argument is called variable length argument. This can be done by declaring variable with * (asterisk) for positional argument and declaring variable with ** (two asterisk) for keyword argument.

```
def fun(*x):  
    sum=0  
    for i in x:  
        sum=sum+i  
    print(sum)
```

```
fun(10)  
fun(10,20)  
fun(10,50,60)
```

output:

10

30

120

Note:- here x is variable length positional argument. Datatype of x is tuple.

```
def fun(**x):  
    for i,v in x.items():  
        print("value at ",i,"=",v)  
fun(x=10,y=30)
```

output:

value at x = 10

value at y = 30

Note:- here x is variable length keyword argument. Datatype of x is dictionary.

EXAMPLE 1:-

#program to show different type of function

```
def simple(p, r=10, t=2, *details, name, age=50, **per_details ):    # function  
header
```

```

'''calculate simple interest'''
si=(p*r*t)/100
a=p+si
print("*****")
print("Simple interest on ",p," at ",r,"% for ",t,"years=",si)
print("Amount=",a)
print("*****other info*****")
for i in details:
    print(i)
print("*****")
print(name, " Please check your details \n age=", age)
for vname, value in per_details.items():
    print(vname,"=", value)
print("-----fun call with only two argument-----")
simple(1000,name="Ravi")          #function call 1
print("-----fun call 2 -----")
simple(5000,15,5, "2018-2020","SBI Bailey Road", age=40, name="Ajay",
address="Sheikhpura more Patna", phone="7772530633")    #function call 2

```

OUTPUT:-

```
-----fun call with only two argument-----
```

```
*****
```

```
Simple interest on 1000 at 10 % for 2 years= 200.0
```

```
Amount= 1200.0
```

```
*****other info*****
```

```
*****
```

```
Ravi Please check your details
```

```
age= 50
```

```
-----fun call 2 -----
```

```
*****
```

```
Simple interest on 5000 at 15 % for 5 years= 3750.0
```

```
Amount= 8750.0
```

```
*****other info*****
```

```
2018-2020
```

```
SBI Bailey Road
```

```
*****
```

```
Ajay Please check your details
```

```
age= 40
```

```
address = Sheikhpura more Patna
```

phone =7772530633

NOTE :- In above example `p` positional argument, `r` and `t` are default arguments that can be skipped but if provided then provided as per position. `details` is variable length argument that will be provided after providing all positional arguments. all values provided after all positional arguments will go to `details`. it will not produce errors if no value is provided for `details`. `name` is keyword argument that must be provided with parameter name `age` is the default parameter that can be skipped but if provided then provided with parameter name. `per_details` is a variable length keyword argument that will be provided after providing all keyword arguments and it should be provided with a parameter name. In definition position argument come first after that variable length argument (*,positional) after that keyword argument after that variable length argument(**,keyword)

IMPORTANT POINT:-

- i) Positional argument should be followed by keyword argument
- ii) Keyword in argument list should be from the list of parameter list
- iii) No parameter should receive value twice.

ERROR:-

1) `simple(1000, 12)`

TypeError: simple() missing 1 required keyword-only argument: 'name'

here 1000 will go in `p` and 12 will go in `r` and take default value of `t` `details` and `per_details` is variable length parameter so if it is not provide it will not produce error

it will take default value for `age` but it will produce error if `name` is not provided

2) `simple(1000, r=10, name="ana")` execute perfectly positional argument can be passed as keyword argument but keyword cannot be passed as positional argument

3) `simple(1000, r=10, 5, name="ana")`

Produce error as position argument (5) follow `r` that is provided as keyword argument

4) `simple(10, p=10100, name="ana")`

positional argument must be given in particular position

TypeError: simple() got multiple values for argument 'p'

