Function -: Input --- Calling --- Output

In [2]:
```
# def h(formal parameters):
#     return x
# h(actual parameters)


# Taking nothing, Giving nothing

def hi():
    print("hi")

# Taking something, Giving nothing

def hello(name):
    print(f"Hello {name}!")

# Taking something, Giving something

def square(n):
    return n*n

# Taking nothing, Giving something

def vowels():
    return "aeiou"
```

In [3]:
```
hi()
hello("Nikhil")
s=square(5)
print(s)
v=vowels()
print(f"{v} : {list(v)}")
```

```
hi
Hello Nikhil!
25
aeiou : ['a', 'e', 'i', 'o', 'u']
```

In [4]:
```
def add(a,b,c):
    return a+b+c
x=add(5)
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[4], line 3
      1 def add(a,b,c):
      2     return a+b+c
----> 3 x=add(5)

TypeError: add() missing 2 required positional arguments: 'b' and 'c'
```

In [5]:
```
x=add(1,5)
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[5], line 1
----> 1 x=add(1,5)

TypeError: add() missing 1 required positional argument: 'c'
```

In [6]:
```
x=add(1,2,5)
print(x)
```

```
8
```

In [14]:
```
# write a python f() to calculate volume of cuboid using Taking nothing, Giving
def vol_cuboid():
    l,b,h=[int(i) for i in input("Enter length,breadth,height of cuboid by giv
    print(f"Volume of Cuboid is : {l*b*h} unit cube")
vol_cuboid()
```

```
Enter length,breadth,height of cuboid by giving space10 12 15
Volume of Cuboid is : 1800 unit cube
```

In [15]:
```
# write a python f() to calculate volume of cuboid using Taking something, Giv
def vol_cuboid(l,b,h):
    print(f"Volume of Cuboid is : {l*b*h} unit cube")
l,b,h=[int(i) for i in input("Enter length,breadth,height of cuboid by giving
vol_cuboid(l,b,h)
```

```
Enter length,breadth,height of cuboid by giving space14 15 12
Volume of Cuboid is : 2520 unit cube
```

In [16]:
```
# write a python f() to calculate volume of cuboid using Taking something, Giv
def vol_cuboid(l,b,h):
    return l*b*h
l,b,h=[int(i) for i in input("Enter length,breadth,height of cuboid by giving
print(f"Volume of Cuboid is : {vol_cuboid(l,b,h)} unit cube")
```

```
Enter length,breadth,height of cuboid by giving space12 45 23
Volume of Cuboid is : 12420 unit cube
```

In [17]:
```
# write a python f() to calculate volume of cuboid using Taking nothing, Giving
def vol_cuboid():
    l,b,h=[int(i) for i in input("Enter length,breadth,height of cuboid by giv
    return l*b*h
print(f"Volume of Cuboid is : {vol_cuboid()} unit cube")
```

```
Enter length,breadth,height of cuboid by giving space25 46 12
Volume of Cuboid is : 13800 unit cube
```

Types of Argument

--: Default Arguments

In [7]:
```
# default argument in f()
def jodo(a=1,b=2,c=3):
    return a+b+c
jodo()
```

Out[7]: 6

In [8]:
```
jodo(10)  # it takes value of a = 10
```

Out[8]: 15

In [9]:
```
def jodo_1(a,b=2,c=3):
    return a+b+c
jodo_1(6)
```

Out[9]: 11

In [10]:
```
jodo_1()  # throws error
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[10], line 1
----> 1 jodo_1()  # throws error

TypeError: jodo_1() missing 1 required positional argument: 'a'
```

In [12]:
```
def jodo_2(a=1,b=2,c): # Not allowed in python bcz non default argument is give
    return a+b+c       #  on giving the default argument
jodo_2(6,7,8)
jodo_2(6)
```

```
  Cell In[12], line 1
    def jodo_2(a=1,b=2,c): # Not allowed in python
                           ^
SyntaxError: non-default argument follows default argument
```

In [29]:
```
# case 1:
def are_of_Circle(r=1,p=3.14):
    return r*r*p
are_of_Circle(7)
```

Out[29]: 153.86

In [26]:
```
# case 2:
def area_of_Circle(r,p=3.14):
    return r*r*p
area_of_Circle(7)
```

Out[26]:
```
153.86
```

In [28]:
```
# case 3:
def area_of_Circle(p=3.14,r):  # wrong practice
    return r*r*p
area_of_Circle(7)
```

```
  Cell In[28], line 2
    def area_of_Circle(p=3.14,r):  # wrong practice
                                   ^
SyntaxError: non-default argument follows default argument
```

Positional Argument Vs Keyword Argument

In [30]:
```
# default arguments are positional argument
```

In [31]:
```
# positional Argument

def rectangle(l,b):
    print(f"l: {l}\t b: {b}")
    return l*b

ar=rectangle(5,6)
print(ar)
```

```
l: 5     b: 6
30
```

In [32]:
```
# positional Argument
# case: 1
def rectangle(l,b):    # l & b are keywords
    print(f"l: {l}\t b: {b}")
    return l*b

ar=rectangle(b=5,l=6)
print(ar)
```

```
l: 6     b: 5
30
```

In [33]:
```
# positional Argument
# case: 1
def rectangle(l,b):    # l & b are keywords
    print(f"l: {l}\t b: {b}")
    return l*b

ar=rectangle(5,b=6)  # it is allowed
print(ar)
```

```
l: 5     b: 6
30
```

In [35]:
```
# positional Argument
# case: 2
def rectangle(l,b):    # l & b are keywords
    print(f"l: {l}\t b: {b}")
    return l*b

ar=rectangle(l=5,6)  # it is not allowed
print(ar)
```

```
  Cell In[35], line 7
    ar=rectangle(l=5,6)  # it is not allowed
                       ^
SyntaxError: positional argument follows keyword argument
```

In [37]:
```
# positional Argument
# case: 3
def rectangle(l,b):    # l & b are keywords
    print(f"l: {l}\t b: {b}")
    return l*b

ar=rectangle(5,l=6)  # throws error
print(ar)
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[37], line 7
      4     print(f"l: {l}\t b: {b}")
      5     return l*b
----> 7 ar=rectangle(5,l=6)  # throws error
      8 print(ar)

TypeError: rectangle() got multiple values for argument 'l'
```

Variable Length Arguments

In [39]:
```
# Case: 1
def display(*t):   # it takes arguments as tupple & no. of arguments are not de
    print(t)

display(6,5)
display(4,7,9)
display(4,7,9,6)
```

```
(6, 5)
(4, 7, 9)
(4, 7, 9, 6)
```

In [14]:
```
# average of given arguments
def average(*t):
    s=0
    t=list(t)
    for i in t[0]:
        s+= i
    avg=s/len(t)
    print(f"sum of {t}: {avg}")

s=[int(i) for i in input("Enter no. for sum by giving space:").split()]
s=tuple(s)
print(s)
average(s)
```

```
Enter no. for sum by giving space:1 5 4 2 3 6 9 8 7
(1, 5, 4, 2, 3, 6, 9, 8, 7)
sum of [(1, 5, 4, 2, 3, 6, 9, 8, 7)]: 45.0
```

Keyword Variable Length Argument

In [5]:
```
# Case: 2
def display(**d):   # it takes arguments as dictionary & "kwarg" is general nam
    print(d)
    print(type(d))

display(a=6,b=9,c=5)  #  keyword is used to give values
```

```
{'a': 6, 'b': 9, 'c': 5}
<class 'dict'>
```

In [ ]:
```

```