

```
def func(a,b,c):
    print(a,b,c)
```

Positional Argument

```
func(1,2,3)
```

```
1 2 3
```

```
func(a=10, b=20, c=30) #keyword argument / named argument
```

```
10 20 30
```

```
func(c=100, a=200, b=400)
```

```
200 400 100
```

```
print(1,2,3,end="The End", sep="---")
```

```
1---2---3The End
```

```
def add(a,b,c=0,d=10):# default argument
    print(a+b+c+d)
```

```
add(3,4)
```

```
17
```

```
add(1,2,3)
```

```
16
```

```
add(1,2,3,4)
```

```
10
```

```
def area(l,b=0):
    if b==0:
        return l*l
    else:
        return l*b
```

```
area(4)
```

```
16
```

```
area(2,3)
```

```
6
```

```
def add(*nums): #variable argument / Valriable length argument
    print(tvpe(nums))
```

```

def add(*args):
    print(nums)
    s = 0
    for n in nums:
        s+=n
    return s

```

add(1,2,3,4)

```

<class 'tuple'>
(1, 2, 3, 4)
10

```

add(1,2)

```

<class 'tuple'>
(1, 2)
3

```

add(10,20,30,40,50)

```

<class 'tuple'>
(10, 20, 30, 40, 50)
150

```

print(1)

```

1

```

print(2,3)

```

2 3

```

print(1,2,3,4)

```

1 2 3 4

```

add((1,2,3,4))

```

<class 'tuple'>
((1, 2, 3, 4),)

```

```

-----
TypeError                                Traceback (most recent call last)
<ipython-input-33-10a1739ddc4e> in <module>
----> 1 add((1,2,3,4))

<ipython-input-24-36128a09497a> in add(*nums)
      4     s = 0
      5     for n in nums:
----> 6         s+=n
      7     return s

```

TypeError: unsupported operand type(s) for +=: 'int' and 'tuple'

SEARCH STACK OVERFLOW

n = [1,2,34,5]

add(n)

```

<class 'tuple'>
([1, 2, 34, 5],)
-----
TypeError                                Traceback (most recent call last)
<ipython-input-34-3552ed909899> in <module>
      1 n = [1,2,34,5]
----> 2 add(n)

<ipython-input-24-36128a09497a> in add(*nums)
      4 s = 0
      5 for n in nums:
----> 6     s+=n
      7     return s

```

```

def intro(**var): #named variable argument
    print(var)
    print(type(var))

```

```
intro(name="jhon", age=20, graduated=False)
```

```

{'name': 'jhon', 'age': 20, 'graduated': False}
<class 'dict'>

```

```
intro(name="sam")
```

```

{'name': 'sam'}
<class 'dict'>

```

```
intro(a=10, b=20)
```

```

{'a': 10, 'b': 20}
<class 'dict'>

```

```
print(sep="--",1,2,3)
```

```

File "<ipython-input-40-480451435609>", line 1
    print(sep="--",1,2,3)
          ^

```

SyntaxError: positional argument follows keyword argument

SEARCH STACK OVERFLOW

```
#func(positional, named, default, *args, **args)
```

▼ OOP

```

class Person: # Degfine a class
    name = "Jhon"
    age = 23

```

```
p = Person() # Create an object
```

```
p.name
```

```
'Jhon'
```

```
p.age
```

```
23
```

```
class Person: # Degfine a class
# Class variable
    name = "Jhon"
    age = 23
```

```
def info(self): # Method
    print("Info of a person")
    print(self.name, self.age)
```

```
p = Person()
p.info()
```

```
Info of a person
Jhon 23
```

```
class Person: # Degfine a class
# Class variable
    name = "Jhon"
    age = 23
```

```
def info(self): # Method
    print("Info of a person")
    print(self.name, self.age)
```

```
def setName(self, name):
    self.name = name
```

```
def setData(self, name, age):
    self.name = name
    self.age = age
```

```
p = Person()
# p.setName("Sam")
p.setData("Sara", 20)
```

```
p.info()
```

```
Info of a person
Sara 20
```

```
class Person:
    name = ""
    age = 0
    def __init__(self): # Constructor
        print("Object is created")
```

```
p = Object is createdPerson()
```

```
class Person:
    name = ""
    age = 0
    def __init__(self, name, age): # Constructor
        print("Object is created")
        self.name = name
        self.age = age

    def info(self):
        return f"{self.name} {self.age}"
```

```
p = Person("Ema", 40)
```

```
Object is created
```

```
p.info()
```

```
'Ema 40'
```

```
print(p)
```

```
<__main__.Person object at 0x7f38ea694bb0>
```

```
class Person:
    name = ""
    age = 0
    def __init__(self, name, age): # Constructor
        print("Object is created")
        self.name = name
        self.age = age

    def info(self):
        return f"{self.name} {self.age}"

    def __str__(self):
        return f"{self.name} {self.age}"
```

```
p = Person("Tom", 5)
```

```
Object is created
```

```
print(p)
```

```
Tom 5
```

```
class A:
    a = 10
    def info(self):
        print(self.a)
```

```
class B(A):
    pass
```

```
b = B()
print(b.a)
b.info()
```

```
10
10
```

MRO => Method Resolution Order

```
class A:
    a = 10
    def info(self):
        print(self.a)
    def display(self):
        print("Display of Class A")
```

```
class B(A):
    a = 20
    def info(self):
        print(self.a)
```

```
b = B()
print(b.a)
b.info()
b.display()
```

```
20
20
Display of Class A
```

```
class A:
    a = 10
    def info(self):
        print(self.a)
    def display(self):
        print("Display of Class A")
```

```
class B(A):
    a = 20
```

```
def info(self):
    print(f"Child data ",self.a,"Parent Data", super().a)
    super().info()
```

```
b = B()
print(b.a)
b.info()
b.display()
20
Child data 20 Parent Data 10
20
Display of Class A
```

```
class A:
    a = 0
    def __init__(self, a):
        self.a = a
    def getA(self):
        return self.a
```

```
class B(A):
    b = 0
    def __init__(self, a, b):
        self.b = b
        super().__init__(a)

    def getB(self):
        return self.b
    def sum(self):
        return self.a +self.b
```

```
b = B(10,20)
print(b.sum())
```

30

```
class Student:
    def __init__(self, name, roll):
        # Instance variable
        self.name = name
        self.roll = roll

    def info(self):
        print(self.name, self.roll)
```

```
s = Student("Ajay", 2)
s.info()
```

```
2+3
```

```
5
```

```
'A'+'B'
```

```
'AB'
```

```
s = 'hello'
l = [1,2,3]
len(s)
```

```
5
```

```
len(l)
```

```
3
```

```
t = (10,20,30)
```

```
sum(l)
```

```
6
```

```
sum(t)
```

```
60
```

```
2/0
```

```
-----
ZeroDivisionError                                Traceback (most recent call last)
<ipython-input-94-e8326a161779> in <module>
----> 1 2/0
```

```
ZeroDivisionError: division by zero
```

SEARCH STACK OVERFLOW

```
try:
    a = int(input("Enter first number "))
    b = int(input("Enter second number "))
    c = a/b
    print("Result = ",c)
except ZeroDivisionError:
    print("Second number can not be zero")
except ValueError as ve:
    print("Please enter a valid number")
except Exception as ex:
    print(ex)
finally:
    print("Finally block is executed")
```



```
Enter first number abcd  
Please enter a valid number  
Finally block is executed
```

✓ 5s completed at 11:17 AM

