

# Python - Argumentos Arbitrários

## Argumentos Arbitrários (\*args)

Você pode querer definir uma **função** que seja capaz de aceitar um número **arbitrário** ou **variável de argumentos** . Além disso, o número arbitrário de argumentos pode ser **argumentos posicionais** ou de palavras-chave .

- Um argumento prefixado com um único asterisco \* para argumentos posicionais arbitrários.
- Um argumento prefixado com dois asteriscos \*\* para argumentos de palavras-chave arbitrários.

## Exemplo de argumentos arbitrários

Abaixo está um exemplo de argumentos posicionais de comprimento arbitrário ou variável -

```
# sum of numbers
def add(*args):
    s=0
    for x in args:
        s=s+x
    return s
result = add(10,20,30,40)
print (result)

result = add(1,2,3)
print (result)
```

A variável **args** prefixada com "\*" armazena todos os valores passados para ela. Aqui, args se torna uma tupla. Podemos executar um **loop** sobre seus itens para somar os números.

Ele produzirá a seguinte **saída** -

```
100
6
```

## Argumentos necessários com argumentos arbitrários

Também é possível ter uma função com alguns argumentos obrigatórios antes da sequência do número **variável** de valores.

## Exemplo

O exemplo a seguir tem a função **avg()** . Suponha que um aluno possa fazer qualquer número de testes. O primeiro teste é obrigatório. Ele pode fazer quantos testes quiser para melhorar sua pontuação. A função calcula a média das notas na primeira prova e sua pontuação máxima nas demais provas.

A função tem dois argumentos, o primeiro é o argumento necessário e o segundo para conter qualquer número de valores.

```
#avg of first test and best of following tests
def avg(first, *rest):
    second=max(rest)
    return (first+second)/2

result=avg(40,30,50,25)
print (result)
```

A chamada seguinte para a função avg() passa primeiro o valor para o argumento necessário e os valores restantes para uma tupla chamada rest. Em seguida, encontramos o máximo e o usamos para calcular a média.

Ele produzirá a seguinte **saída** -

```
45.0
```

## Argumentos de palavras-chave arbitrárias (\*\*kwargs)

Se uma variável na lista de argumentos tiver dois asteriscos prefixados, a função poderá aceitar um número arbitrário de argumentos de palavras-chave. A variável se torna um **dicionário** de pares palavra-chave:valor.

## Exemplo

O código a seguir é um exemplo de função com argumentos de palavras-chave arbitrários. A função addr() tem um argumento \*\*kwargs que é capaz de aceitar qualquer número de elementos de endereço como nome, cidade, phno, pin, etc. Dentro da função kwargs, o dicionário de pares kw:value é percorrido usando o método items().

```
def addr(**kwargs):
    for k,v in kwargs.items():
```

```
print ("{}:{}".format(k,v))

print ("pass two keyword args")
addr(Name="John", City="Mumbai")
print ("pass four keyword args")

# pass four keyword args
addr(Name="Raam", City="Mumbai", ph_no="9123134567", PIN="400001")
```

Ele produzirá a seguinte **saída** -

```
pass two keyword args
Name:John
City:Mumbai
pass four keyword args
Name:Raam
City:Mumbai
ph_no:9123134567
PIN:400001
```

DE ANÚNCIOS

## Vários argumentos com argumentos de palavras-chave arbitrárias

Se a função usar tipos mistos de argumentos, os argumentos de palavras-chave arbitrárias deverão estar após os argumentos posicionais, de palavras-chave e de posição arbitrária na lista de argumentos.

### Exemplo

Imagine um caso em que ciências e matemática são disciplinas obrigatórias, além do qual o aluno pode escolher qualquer número de disciplinas optativas.

O código a seguir define uma função **percent()** onde as notas em ciências e as notas são armazenadas em argumentos obrigatórios, e as notas em número variável de disciplinas eletivas em **\*\*argumento opcional**.

```
def percent(math, sci, **optional):
    print ("maths:", math)
    print ("sci:", sci)
    s=math+sci
    for k,v in optional.items():
```

```
print ("{}:{}".format(k,v))
s=s+v
return s/(len(optional)+2)

result=percent(math=80, sci=75, Eng=70, Hist=65, Geo=72)
print ("percentage:", result)
```

Ele produzirá a seguinte **saída** -

```
maths: 80
sci: 75
Eng:70
Hist:65
Geo:72
percentage: 72.4
```