

# Python - argumentos de palavras-chave

## Argumentos de palavras-chave

Python permite passar argumentos de função na forma de palavras-chave, também chamadas de argumentos nomeados. **Variáveis** na **definição da função** são usadas como palavras-chave. Quando a função é chamada, você pode mencionar explicitamente o nome e seu valor.

## Exemplo de argumentos de palavras-chave

```
# Function definition is here
def printinfo( name, age ):
    "This prints a passed info into this function"
    print ("Name: ", name)
    print ("Age ", age)
    return

# Now you can call printinfo function
# by positional arguments
printinfo ("Naveen", 29)

# by keyword arguments
printinfo(name="miki", age = 30)
```

Por padrão, a função atribui os valores aos argumentos na ordem em que aparecem. Na segunda chamada de função, atribuímos o valor a um argumento específico

Ele produzirá a seguinte **saída** -

```
Name: Naveen
Age 29
Name: miki
Age 30
```

Vamos tentar entender mais sobre o argumento de palavra-chave com a ajuda da seguinte definição de função -

```
def division(num, den):
    quotient = num/den
    print ("num:{} den:{} quotient:{}".format(num, den, quotient))
```



```
division(10,5)
division(5,10)
```

Como os valores são atribuídos de acordo com a posição, a saída é a seguinte -

```
num:10 den:5 quotient:2.0
num:5 den:10 quotient:0.5
```

## Função de chamada com argumentos de palavras-chave

Em vez de passar os valores com argumentos posicionais, vamos chamar a função com argumentos de palavras-chave -

### Exemplo

```
division(num=10, den=5)
division(den=5, num=10)
```

Ele produzirá a seguinte **saída** -

```
num:10 den:5 quotient:2.0
num:10 den:5 quotient:2.0
```

## Ordem dos argumentos de palavras-chave

Ao usar argumentos de palavras-chave, não é necessário seguir a ordem dos argumentos formais na definição da função.

O uso de argumentos de palavras-chave é opcional. Você pode usar chamadas mistas. Você pode passar valores para alguns argumentos sem palavras-chave e para outros com palavras-chave.

```
division(10, den=5)
```

No entanto, os argumentos posicionais devem estar antes dos argumentos de palavras-chave ao usar chamadas mistas.

Tente chamar a função `division()` com a seguinte instrução.

```
def division(num, den):
    quotient = num/den
    print ("num:{} den:{} quotient:{}".format(num, den, quotient))
```



```
division(num=5, 10)
```

Como o argumento posicional não pode aparecer após argumentos de palavras-chave, Python gera a seguinte mensagem de erro -

```
division(num=5, 10)
```

^

SyntaxError: non-keyword arg after keyword arg