Functions Arguement

In [1]:

```
def person(name, age):
    print(name)
    print(age)

person('nit',22)
```

```
nit
22
```

In [2]:

```
def person(name, age):
    print(name)
    print(age)

person('nit')
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[2], line 5
      2     print(name)
      3     print(age)
----> 5 person('nit')

TypeError: person() missing 1 required positional argument: 'age'
```

In [3]:

```
def person(name, age):
    print(name)
    print(age)

person(22)
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[3], line 5
      2     print(name)
      3     print(age)
----> 5 person(22)

TypeError: person() missing 1 required positional argument: 'age'
```

In [4]:

```
def person(name, age):
    print(name)
    print(age)

person('nit',22,23,24,45,67)
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[4], line 5
      2     print(name)
      3     print(age)
----> 5 person('nit',22,23,24,45,67)

TypeError: person() takes 2 positional arguments but 6 were given
```

In [5]:

```
def person(name, age):
    print(name)
    print(age)
person(22,'nit')
```

```
22
nit
```

In [9]:

```
def person(name, age):
    print(name)
    print(age-1)

person(22, 'nit')
```

```
22
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[9], line 5
      2     print(name)
      3     print(age-1)
----> 5 person(22, 'nit')

Cell In[9], line 3, in person(name, age)
      1 def person(name, age):
      2     print(name)
----> 3     print(age-1)

TypeError: unsupported operand type(s) for -: 'str' and 'int'
```

In [10]:

```
def person(name, age):
    print(name)
    print(age)

person(name='nit',age= 22)
```

```
nit
22
```

In [12]:

```
def person(name, age):
    print(name)
    print(age-1)

person(age= 22, name='nit')
```

```
nit
21
```

In [15]:

```
def person(name, age, new_age):
    print(name)
    print(age-1)

person(age= 22, name='nit')
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[15], line 5
      2     print(name)
      3     print(age-1)
----> 5 person(age= 22, name='nit')

TypeError: person() missing 1 required positional argument: 'new_age'
```

In [17]:
```
def person(name, age, new_age):
    print(name)
    print(age-1)
    print(new_age)

person(age= 22, name='nit', new_age=23)
```

```
nit
21
23
```

In [18]:
```
# Default argument

def person(name, age=18):
    print(name)
    print(age)
person('nit')
```

```
nit
18
```

In [19]:
```
def person(name, age=18):
    print(name)
    print(age)
person('nit',40)
```

```
nit
40
```

In [20]:
```
# Variable length arguement

def person(name, age):
    print(name)
    print(age)

person('nit', 40, 50, 60, 70, 80 )
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[20], line 7
      4     print(name)
      5     print(age)
----> 7 person('nit', 40, 50, 60, 70, 80 )

TypeError: person() takes 2 positional arguments but 6 were given
```

In [21]:
```
def sum(a,b):
    c = a+b
    print(c)

sum(5,6,7,8)
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[21], line 5
      2     c = a+b
      3     print(c)
----> 5 sum(5,6,7,8)

TypeError: sum() takes 2 positional arguments but 4 were given
```

In [22]:

```
def sum(a,*b):
    c = a+b
    print(c)

sum(5,6,7,8)
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[22], line 5
      2     c = a+b
      3     print(c)
----> 5 sum(5,6,7,8)

Cell In[22], line 2, in sum(a, *b)
      1 def sum(a,*b):
----> 2     c = a+b
      3     print(c)

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

In [23]:

```
def sum(a,*b):
    #c = a+b
    print(type(a))
    print(type(b))
sum(5,6,7,8)
```

```
<class 'int'>
<class 'tuple'>
```

In [24]:

```
def sum(a, *b): # 1st argument is fixed & we fetch each value from the tuple & w
    c = a
    for i in b:
        c = c + i
    print(c)

sum(5,6,7,8)
```

```
26
```

In [25]:

```
def sum(a, *b):
    c = a
    for i in b:
        c = c + i
    print(c)

sum(5,6,7,8,9,10)
```

```
45
```

In [26]:

```
# kwargs

def person():
    person('ALEX', 36, 'JOHN', 9877767)
```

In [27]:

```
def person(name, *data):
    print(name)
    print(data)

person('ALEX', 36, 'JOHN', 9877767)
```

```
ALEX
(36, 'JOHN', 9877767)
```

In [28]:

```
def person(name,*data):
    print(name)
    print(data)

person('ALEX', age = 36, home_place ='southcity', mob =987767)
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[28], line 5
      2     print(name)
      3     print(data)
----> 5 person('ALEX', age = 36, home_place ='southcity', mob =987767)

TypeError: person() got an unexpected keyword argument 'age'
```

In [30]:

```
def person(name,**data):
    print(name)
    print(data)

person('ALEX', age = 36, home_place ='southcity', mob =987767)
```

```
ALEX
{'age': 36, 'home_place': 'southcity', 'mob': 987767}
```

In [33]:

```
def person(name, **data):
    print('name')
    print(data)

person('ALEX', age = 36, home_place ='southcity', mob =987767, slary= 40000, mar
```

```
name
{'age': 36, 'home_place': 'southcity', 'mob': 987767, 'slary': 40000, 'married':
'yes'}
```

In [34]:

```
# Global Variable and Local Variable

a = 10      # global variable

def something():
    b = 15   # local variaable
```

In [35]:

```
a = 10      # global variable

def something():
    b = 15   # local variaable

    print('in function',b)
    print('out function',a)
```

In [37]:

```
a = 10

def something():
    b = 15

    print('in function',b)

print('out function',a)
```

```
out function 10
```

In [39]:

```python
a = 10

def something():
    a = 15

print('in function',a)
print('out function',a)
```

```
in function 10
out function 10
```

In [40]:

```python
a = 10

def something():
    b = 15
    print('in function',b)    # local variable

something()
print('out function',a)
```

```
in function 15
out function 10
```

In [41]:

```python
a = 10

def something():
    #if we remove this variable then can befault it consider as global variable
    print('in function',a)

something()
print('out function',a)
```

```
in function 10
out function 10
```

In [43]:

```python
a = 10
def something():
    global a
    b = 15          # 15 is converted to local when user assigned global a
    print('in function',b)
    print('global variable', a)

something()
print('out function',a)
```

```
in function 15
global variable 10
out function 10
```

In [45]:

```python
a = 20

def something():
    global a
    a = 15
    print('in function',a)

    a=19
something()
print('out function',a)
```

```
in function 15
out function 19
```

In [46]:

```python
x = 10  # Global variable

def update_x():
    global x  # Declare that we are using the global variable x
    x += 5    # Modify the global variable

update_x()
print(x)  # Output: 15
```

```
15
```

In [53]:

```python
 x = 10
def update_x():
    globals()['x'] += 5

update_x()
print(x)
```

```
15
```