

## Day14\_Parameter\_types

1. Arguments or parameters are the parts of a function that makes us to work with the data input into the function.
2. We know that there are formal parameters and actual parameters used at the time of function definition and function call respectively
3. In python we can make the parameters supplied in different forms. They are
  1. Positional Arguments
  2. Keyword Arguments
  3. Default Arguments
  4. Variable Length Arguments
  5. Keyword Based Variable length Arguments

1. Positional Arguments : The arguments which are declared in the function definition are said to be Positional Arguments by default. The function call must follow the position or the order of parameters otherwise there may be some errors or unexpected output

```
def city(first,last):  
    print(first+last)
```

```
city("New ","York")  
#New York
```

```
city("York ","New")  
#York New
```

```
city("New York")  
#TypeError: city() missing 1 required positional argument: 'last'
```

2. Keyword Arguments : The arguments for which we can pass the values using the argument name at the function call irrespective of order

```
def city(first,last):  
    print(first+last)
```

```
city(first="New ",last="Jersey")  
#New Jersey
```

```
city(last="Jersey",first="New ")  
#New Jersey
```

```
city("New ",last="Jersey")  
#New Jersey
```

```
city("Jersey",first="New ")  
#TypeError: city() got multiple values for argument 'first'
```

```
city(last="Jersey","New ")  
#SyntaxError: positional argument follows keyword argument
```

```
city(first="Jersey","New ")  
#SyntaxError: positional argument follows keyword argument
```

3. Default Arguments : These are the types of arguments where we pass the default values into the parameters at the time of function definition itself . The advantage is when we do not pass any value into the argument the default value would be taken

```
def shop(item="Anonymous",price=0.0):  
    print("Item is",item)  
    print("Price is",price)
```

```
shop()  
Item is Anonymous
```

Price is 0.0

```
shop(item="Trouser",price=499.0)
```

Item is Trouser

Price is 499.0

```
shop(item="Trouser")
```

Item is Trouser

Price is 0.0

```
shop(price="499")
```

Item is Anonymous

Price is 499

```
shop("Shirt",599)
```

Item is Shirt

Price is 599

```
shop("Shirt")
```

Item is Shirt

Price is 0.0

```
shop(599)
```

Item is 599

Price is 0.0

```
shop("T-Shirt",price=399)
```

Item is T-Shirt

Price is 399

```
shop(price=399,"T-Shirt")
```

#SyntaxError: positional argument follows keyword argument

```
shop(price=399,item="T-Shirt")
```

Item is T-Shirt

Price is 399

```
shop(item="T-Shirt",399)
```

#SyntaxError: positional argument follows keyword argument

```
def shop(item,price=0.0):
```

```
    print("Item is",item)
```

```
    print("Price is",price)
```

```
shop(item="Trouser",price=499.0)
```

Item is Trouser

Price is 499.0

```
shop(item="Trouser")
```

Item is Trouser

Price is 0.0

```
shop(price="499")
```

#TypeError: shop() missing 1 required positional argument: 'item'

```
shop("Shirt",599)
```

Item is Shirt

Price is 599

```
shop("Shirt")
```

Item is Shirt

Price is 0.0

```
shop(599)
Item is 599
Price is 0.0
```

```
shop("T-Shirt",price=399)
Item is T-Shirt
Price is 399
```

```
shop(price=399,item="T-Shirt")
Item is T-Shirt
Price is 399
```

```
shop(item="T-Shirt",399)
#SyntaxError: positional argument follows keyword argument
```

```
shop(price=399,"T-Shirt")
#SyntaxError: positional argument follows keyword argument
```

```
#SyntaxError: non-default argument follows default argument
def shop(item="Anonymous",price):
    print("Item is",item)
    print("Price is",price)
```

4. Variable length arguments : This kind of arguments allow us to have a collection of values without even passing the values as a predefined collection

```
def varpar(ls):
    for i in ls:
        print(i)
```

```
varpar([4,2,6,8])
```

```
4
2
6
8
```

```
varpar(4,2,6,8)
```

```
#TypeError: varpar() takes 1 positional argument but 4 were given
```

```
def varpar(*ls):
    for i in ls:
        print(i)
```

```
varpar(4,2,6,8)
```

```
4
2
6
8
```

```
def varpar(*ls,s):
    sm=0
    for i in ls:
        sm=sm+i
    print(sm+s)
```

```
varpar(4,2,6,8,9)
```

```
#TypeError: varpar() missing 1 required keyword-only argument: 's'
```

```
def varpar(s,*ls):  
    sm=0  
    for i in ls:  
        sm=sm+i  
    print(sm+s)
```

```
varpar(4,2,6,8,9)
```

5.Keyword Variable arguments : These parameters are defined for taking up a dictionary kind of values from the function call

```
def kvarpar(**kw):  
    print(kw)
```

```
kvarpar(eno=1011,ename='Carl',esal=50000)  
#{'eno': 1011, 'ename': 'Carl', 'esal': 50000}
```