

Mixed positional keyword

If we want a function to take only positional arguments or only keyword can enforce and make it mandatory.

Let's see how some arguments can have positional only or keywords only.

```
Def add(a, b, c, d, e, f):# taking 6 arguments  
return a+ b+ c+ d+ e+ f # adding and returning result
```

if we have integer type of all so, it will add all and return the result of all

add(2, 5, 9, 7, 3, 8)and these arguments will be copied to the corresponding arguments.

In the same position to this formal parameter get copied in actual parameters.

2 goes in a

5 goes in b

.....

Add(f=8, c=9, b=5, c=3, d=7, a=2)

So we can use keyword argument also

Here 8 goes to f

.....

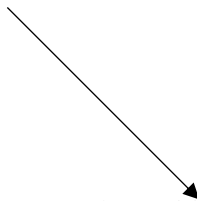
```
Def add(a,b,\ c, d, e, f):
```

```
Return a+b+c+d+e+f
```

Taking 6 arguments adding and returning result but here the slash means we can only give positional arguments before slash and after we can give both positional and keyword arguments both are allowed.

Add(2,5,6,7,9,4) # all positional

Add(2, 5, d=7, f=4, e=9, c=6)



This two must be positional

Where keyword passed order must not be important.

```
def add(a, b, c, d, e, f):  
    return a+b+c+d+e+f  
r=add(2, 5, 7, 4, 8, 9)# updating this to check whether it is working or not
```

```
def add(a, b, /, c, d, e, f):
    print(a, b, c, d, e, f):
    return a+b+c+d+e+f
r=add(f=9,e=8, a=2, c=7, b=5, d=4)
print(r)
```

if given slash it will give an error as we have given a and b as positional argument.

```
Def add(a, b, /, c, d, *, e, f):
```

Positional any keyword

Return a+b+c+d+e+f

Add(2,5,d=8, c=4, 9, 3)

By writing / we say that only positional arguments

By writing * at the beginning of the statement we say keywords are mandatory

Program

```
def add(a, b, /, c, d, *, e, f):
    print(a, b, c, d, e, f)
    return a+b+c+d+e+f
r = add(2, 5, 6, 7, f=9, e=8)
print(r)
```