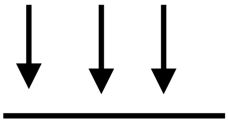
Positional vs keyword Arguments

Def net_sal(basic, allowance, deduction): # for calculating net sal of an employee.

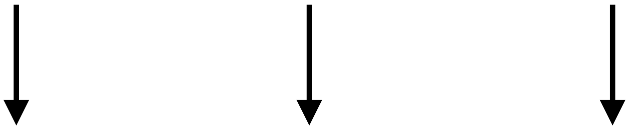
net=basic +allowance_deduction
Return net

n= net_sal(8000, 6000,2000)# way to call function for net sal



Actual parameters

Basic allowance deduction



8000 6000 2000

Actual parameters can be copied

Def net_sal(basic, allowance, deduction): #function for net sal

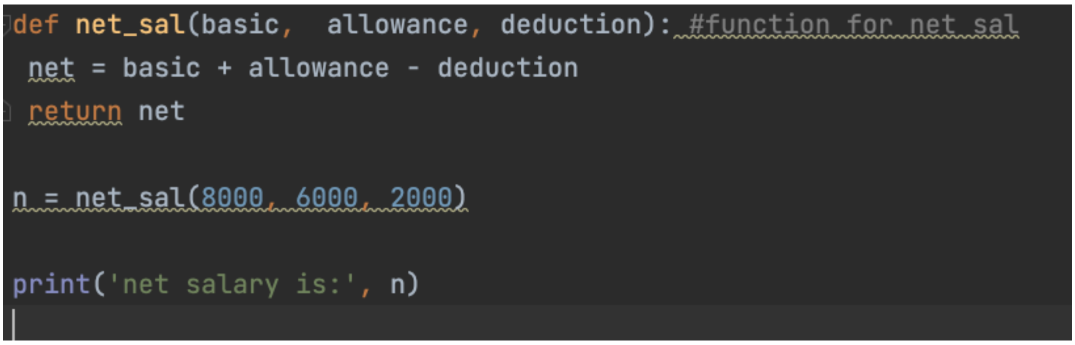
net= basic+allowance-deduction

Return net

n=net_sal(basic, allowance, deduction)

print(‘Net salary is:, n’)

To know which variable has print which value.

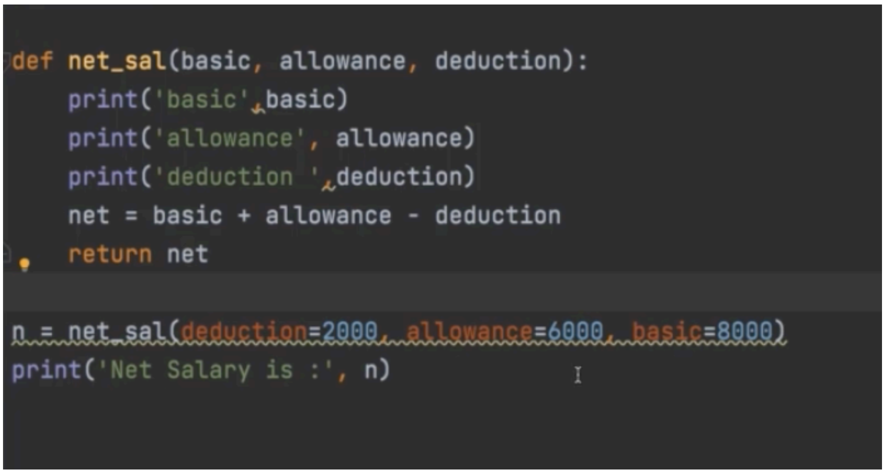


```
def net_sal(basic,  allowance, deduction): #function for net sal
 net = basic + allowance - deduction
 return net

n = net_sal(8000, 6000, 2000)

print('net salary is:', n)
```

Output:
Net salary is : 12000

# so this is called as positional arguments

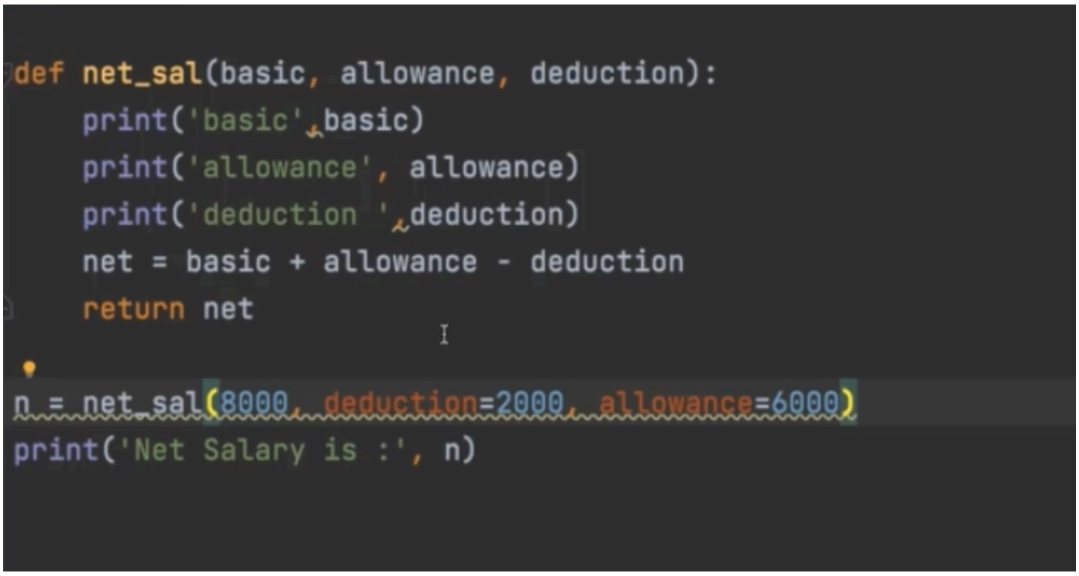
Program:



```python
def net_sal(basic, allowance, deduction):
    print('basic',basic)
    print('allowance', allowance)
    print('deduction ',deduction)
    net = basic + allowance - deduction
    return net

n = net_sal(deduction=2000, allowance=6000, basic=8000)
print('Net Salary is :', n)
```

By writing the names of parameters also we can call the function.

These are called as keyword argument. Without that it is a positional argument



```python
def net_sal(basic, allowance, deduction):
    print('basic',basic)
    print('allowance', allowance)
    print('deduction ',deduction)
    net = basic + allowance - deduction
    return net

n = net_sal(8000, deduction=2000, allowance=6000)
print('Net Salary is :', n)
```

# both positional and keyword arguement