
```
#Variable and string concatenation
```

```
a=20
print("I am ",a,"years old")
```

```
➞ I am 20 years old
```

```
#Arithmetic Operations
```

```
a=int(input("enter first value "))
b=int(input("enter second value "))
print("sum=",a+b)
print("Difference=",a-b)
print("Product=",a*b)
print("Quatient=",a/b)
print("Remainder=",a%b)
print("Power of a to be =",a**b)
```

```
enter first value 3
enter second value 2
sum= 5
Difference= 1
Product= 6
Quatient= 1.5
Remainder= 1
Power of a to be = 9
```

```
#Take salary from the user,if the salary is more then 50000 then tax is 10% and print final salary
```

```
salary=int(input("Enter your salary: "))
if(salary>=50000):
    final_salary=salary-salary*(0.1)
    print("Final salary = ",final_salary)
else:
    print("Your salary is less than 50000, so no tax")
```

```
Enter your salary: 50000
Final salary = 45000.0
```

```
#Get the salary from the user if salary is more than equal to 50K apply 10% tax reduction, if the salary is 50K apply 5% tax ,if the sa
```

```
salary=int(input("Enter your salary: "))
if(salary<50000):
    final_salary=salary-salary*(0.05)
    print("Final salary = ",final_salary)
elif(salary<=70000):
    final_salary=salary-salary*(0.1)
    print("Final salary = ",final_salary)
elif(salary<=100000):
    final_salary=salary-salary*(0.12)
    print("Final salary = ",final_salary)
else:
    print("You are promoted to next level ")
```

```
#For loop
```

```
n=int(input("Enter a number : "))
for i in range(1,11):
    print(n,"x",i,"=",n*i)
```

```
#While loop
```

```
n=int(input("Enter a number : "))
i=1
while(i!=11):
    print(n,"x",i,"=",n*i)
    i=i+1
```

List

```
#List and its operations
pubg=["M416","SCARL","AUG","M249","QBZ","MINI14","FAMOUS","M16A4"]
ammo=["5.56mm","7.32mm","9mm","45ACP","12GUAGE","300MAGNUM"]
pubg.append(5.56)
pubg.pop(2)
pubg.remove("FAMOUS")
pubg.insert(2,"AKM")
print(pubg)
print(pubg+ammo)
print(len(pubg))
```

```
pubg=["M416","SCARL","AUG","M249","QBZ","MINI14","FAMOUS","M16A4"]
pubg.append(5.56)
print(pubg)
```

```
pubg=["M416","SCARL","AUG","M249","QBZ","MINI14","FAMOUS","M16A4"]
pubg.insert(2,"AKM")
print(pubg)
```

```
#Create a list of 10 integers elements and print their squares and save these squares in a new list
l=[1,2,3,4,5,6,7,8,9,10]
l1=[]
for i in l:
    l1.append(i*i)
print(l1)
```

List Comprehension

```
#List comprehension
#newlist=[output for]
l=[1,2,3,4,5,6,7,8,9,10]
l1=[i*i for i in l ]
print(l1)
```

```
#List comprehension
#newlist=[output for if(condition)]
l=[1,2,3,4,5,6,7,8,9,10]
l1=[i*i for i in l if i<=5]
print(l1)
```

```
[1, 4, 9, 16, 25]
```

```
#List comprehension
#newlist=[statements if(condition) else statements]
l=[1,2,3,4,5,6,7,8,9,10]
l1=[i*i if i<=5 else i for i in l]
print(l1)
```

```
[1, 4, 9, 16, 25, 6, 7, 8, 9, 10]
```

```
#Take the salary from the user if salary <50K 10%tax otherwise 15%
salary=[45000,50000,60000,80000]
newlist=[i-i*0.1 if i<=50000 else i-i*0.15 for i in salary]
print(newlist)
```

```
[40500.0, 45000.0, 51000.0, 68000.0]
```

Dictionary

```
#Dictionary
dict1={"Game":'PUBG',"Year":2017,"M416":'5.56mm',"AKM":'7.32mm',"UMP":'45acp'}
print(dict1)
```

```
{'Game': 'PUBG', 'Year': 2017, 'M416': '5.56mm', 'AKM': '7.32mm', 'UMP': '45acp'}
```

```
dict1={"Game":'PUBG',"Year":2017,"M416":'5.56mm',"AKM":'7.32mm',"UMP":'45acp'}
print(dict1.keys())
print(dict1.values())
```

```
dict_keys(['Game', 'Year', 'M416', 'AKM', 'UMP'])
dict_values(['PUBG', 2017, '5.56mm', '7.32mm', '45acp'])
```

```
dict1={"Game":'PUBG',"Year":2017,"M416":'5.56mm',"AKM":'7.32mm',"UMP":'45acp'}
print(dict1['AKM'])
```

```
7.32mm
```

```
dict2={"Dept":'CSE',"details":{"Year":'II',"Sec":'D',"Strength":71}}
print("Year is ",dict2["details"]["Year"])
print("Section is ",dict2["details"]["Sec"])
print("Strength is ",dict2["details"]["Strength"])
```

```
Year is II
Section is D
Strength is 71
```

```
id=[1,2,3,4,5,6]
names=["Shankar","Binnu","Sakul","Dileep","Shiva","Subash"]
new=zip(id,names)
for i in new:
    print(i)
```

```
(1, 'Shankar')
(2, 'Binnu')
(3, 'Sakul')
(4, 'Dileep')
(5, 'Shiva')
(6, 'Subash')
```

```
id=[1,2,3,4,5,6]
names=["Shankar","Binnu","Sakul","Dileep","Shiva","Subash"]
new=dict(zip(id,names))
print(new)
```

```
{1: 'Shankar', 2: 'Binnu', 3: 'Sakul', 4: 'Dileep', 5: 'Shiva', 6: 'Subash'}
```

Numpy

```
import numpy as np
a=np.array([1,2,3,4,5])#1d array
print(a.ndim)
print(a.shape)
print(a)
```

```
1
(5,)
[1 2 3 4 5]
```

```
ar=np.array([[1,2,3],[4,5,6]])#2d array
print(ar.ndim)
print(ar.shape)
print(ar)
```

```
2
(2, 3)
[[1 2 3]
 [4 5 6]]
```

```
arr=np.array([[[1,2,3],[4,5,6]],[[7,8,9],[10,11,12]]])#3d array
print(arr.ndim)
print(arr.shape)
print(arr)
```

```

3
(2, 2, 3)
[[[ 1  2  3]
  [ 4  5  6]]

 [[ 7  8  9]
 [10 11 12]]]

a=np.ones((2,3,3))#g,r,c
print(a)

[[[1.  1.  1.]
  [1.  1.  1.]
  [1.  1.  1.]]

 [[1.  1.  1.]
  [1.  1.  1.]
  [1.  1.  1.]]]

a=np.zeros((3,3))#Inserts all the elements as zeros
print(a)

[[0.  0.  0.]
 [0.  0.  0.]
 [0.  0.  0.]]

a=np.eye(3)#Identity matrix
print(a)

[[1.  0.  0.]
 [0.  1.  0.]
 [0.  0.  1.]]

a=np.arange(1,10,2)#start,stop,stepsize
print(a)
print(type(a))

[1  3  5  7  9]
<class 'numpy.ndarray'>

#Print all the numbers that are number divisible by 8 using arange func b/w 1-1000
a=np.arange(8,1001,8)
print(a)

[  8  16  24  32  40  48  56  64  72  80  88  96 104 112
 120 128 136 144 152 160 168 176 184 192 200 208 216 224
 232 240 248 256 264 272 280 288 296 304 312 320 328 336
 344 352 360 368 376 384 392 400 408 416 424 432 440 448
 456 464 472 480 488 496 504 512 520 528 536 544 552 560
 568 576 584 592 600 608 616 624 632 640 648 656 664 672
 680 688 696 704 712 720 728 736 744 752 760 768 776 784
 792 800 808 816 824 832 840 848 856 864 872 880 888 896
 904 912 920 928 936 944 952 960 968 976 984 992 1000]

a=np.linspace(4,5,10)#linspace splitting the range of numbers into n numbers with common differences
print(a)

[4.          4.11111111 4.22222222 4.33333333 4.44444444 4.55555556
 4.66666667 4.77777778 4.88888889 5.          ]

```

Tuples

```

#Tuples and its operations
tup=(1,2,3,4,5)
print(tup)

(1, 2, 3, 4, 5)

#Length of tuple
tup=(1,2,3,4,5)
print(len(tup))

```

5

```
#Maximum and Minimun of tuple
tup=(1,2,3,4,5)
print(max(tup))
print(min(tup))
```

5

1

```
#Returns the index of the given value
tup=(1,2,3,4,5)
print(tup.index(3))
```

2

```
tup1=(1,2,3,4,5)
tup2=("a","b")
print(tup1+tup2)
```

(1, 2, 3, 4, 5, 'a', 'b')

```
#Updating a tuple
tup=(1,2,3,4,5)
x=list(tup)
x[4]=6
tup=tuple(x)
print(tup)
```

(1, 2, 3, 4, 6)

```
#Append -Coverting into list
tup=("M416","AKM")
x=list(tup)
x.append("UMP")
tup=tuple(x)
print(tup)
```

('M416', 'AKM', 'UMP')

Sets

```
#Sets and its operations
a={1,2,3,4,5,3,2,1}#duplicate values not allowed
print(a)
```

{1, 2, 3, 4, 5}

```
#Adding elemets into a set
a={1,2,3,4,5}
a.add(6)
print(a)
```

{1, 2, 3, 4, 5, 6}

```
a={1,2,3}
b={4,5}
a.update(b)#add elements of b into a
print(a)
```

{1, 2, 3, 4, 5}

```
#Removing an element from the set
```

```
a={1,2,3,4,5}
```

```
a.remove(3)
```

```
print(a)
```

```
{1, 2, 4, 5}
```

```
#Union
```

```
a={1,2,4,5}
```

```
b={3,4,5,6,7}
```

```
c=a.union(b)
```

```
print(c)
```

```
{1, 2, 3, 4, 5, 6, 7}
```

```
#intersection
```

```
a={1,2,4,5}
```

```
b={3,4,5,6,7}
```

```
c=a.intersection(b)
```

```
print(c)
```

```
{4, 5}
```

```
#sort func.
```

```
a=[1,3,5,4,2]
```

```
a.sort()
```

```
print(a)
```

```
[1, 2, 3, 4, 5]
```

```
#Sorted fuction changes and save into another variable
```

```
a=[1,54,23,12,19]
```

```
b=sorted(a)#Doesn't disturbs original list
```

```
print(a)
```

```
print(b)
```

```
[1, 54, 23, 12, 19]
```

```
[1, 12, 19, 23, 54]
```

```
tup=(1,3,5,2,4)
```

```
a=sorted(tup)
```

```
print(a)
```

```
[1, 2, 3, 4, 5]
```