

```
list1=[1,2,3,4]
print(90 not in list1)
```

True

```
a=9
b=10
print(a+b)
```

19

```
def taxcal(S):
    if S>0 and S<10000:
        return (5/100)*S
    elif S>=10000 and S<50000:
        return (10/100)*S
    elif S>=50000 and S<200000:
        return (15/100)*S
    elif S>=200000:
        return (20/100)*S
    else:
        return 0
```

```
taxcal(200000)
```

40000.0

```
taxcal(-30000)
```

0

```
taxcal(90000)
```

13500.0

```
w=[67,45,23,50]
h=[1.60,1.27,1.40,1.87]
for i,j in zip(w,h):
    print(i/(j*j))
```

26.171874999999996

27.900055800111602

11.734693877551022

14.298378563870855

```
for i in range(len(w)):
    print(w[i]/(h[i]*h[i]))
```

26.171874999999996

27.900055800111602

11.734693877551022

14.298378563870855

Numpy

```
import numpy as np
ar1= np.array([90,56,12,34])
ar2=np.array([78,45,55,67])
print(ar1+ar2)

[168 101  67 101]

arr1=np.zeros((2,3))
print(arr1)

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

arr2=np.ones((2,3))
print(arr2)

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

arr3=np.eye(3)
print(arr3)

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

arr4=np.array([[2,3,4],[5,6,8]])
print(arr4)
print(np.ndim(arr4))
print(np.shape(arr4))

[[2 3 4]
 [5 6 8]]
2
(2, 3)

arr5=np.array([[1,2,3,4,5,6,8,2]])
arr5=arr5.reshape(4,2)
arr5

array([[1, 2],
       [3, 4],
       [5, 6],
       [8, 2]])

arr6=np.array([[1,2,3,4,5,6,8,2]])
arr6.resize(4,2)
```

```
arr6
```

```
array([[1, 2],  
       [3, 4],  
       [5, 6],  
       [8, 2]])
```

```
arr7=np.arange(10,50).reshape(8,5)  
print(arr7)  
print(np.shape(arr7))
```

```
[[10 11 12 13 14]  
 [15 16 17 18 19]  
 [20 21 22 23 24]  
 [25 26 27 28 29]  
 [30 31 32 33 34]  
 [35 36 37 38 39]  
 [40 41 42 43 44]  
 [45 46 47 48 49]]  
(8, 5)
```

```
arr8=np.arange(13,1001,13)  
print(arr8)  
print(np.shape(arr8))
```

```
[ 13  26  39  52  65  78  91 104 117 130 143 156 169 182 195 208 221  
234  
 247 260 273 286 299 312 325 338 351 364 377 390 403 416 429 442 455  
468  
 481 494 507 520 533 546 559 572 585 598 611 624 637 650 663 676 689  
702  
 715 728 741 754 767 780 793 806 819 832 845 858 871 884 897 910 923  
936  
 949 962 975 988]  
(76,)
```

```
mult_seven=np.arange(7,701,7)  
print(mult_seven)
```

```
[  7  14  21  28  35  42  49  56  63  70  77  84  91  98 105 112 119  
126  
 133 140 147 154 161 168 175 182 189 196 203 210 217 224 231 238 245  
252  
 259 266 273 280 287 294 301 308 315 322 329 336 343 350 357 364 371  
378  
 385 392 399 406 413 420 427 434 441 448 455 462 469 476 483 490 497  
504  
 511 518 525 532 539 546 553 560 567 574 581 588 595 602 609 616 623  
630  
 637 644 651 658 665 672 679 686 693 700]
```

```

arr9=np.linspace(2,8,6)
print(arr9)

[2.  3.2 4.4 5.6 6.8 8. ]

arr10=np.array( [ [[1,2,3],[5,6,7]],[[2,4,5],[5,6,7]] ] )
print(arr10)
print(np.shape(arr10))
print(np.ndim(arr10))

[[[1 2 3]
  [5 6 7]]

  [[2 4 5]
  [5 6 7]]]
(2, 2, 3)
3

```

Matrix Operations

```

mat1=np.array([2,3,4,5]).reshape(2,2)
mat2=np.array([4,5,3,1]).reshape(2,2)
print("Matrix 1: \n",mat1)
print("Matrix 2: \n",mat2)

Matrix 1:
[[2 3]
 [4 5]]
Matrix 2:
[[4 5]
 [3 1]]

print(mat1*mat2)

[[ 8 15]
 [12  5]]

print(mat1.dot(mat2))

[[17 13]
 [31 25]]

print(mat1@mat2)

[[17 13]
 [31 25]]

print(np.linalg.inv(mat1))

[[-2.5  1.5]
 [ 2.  -1. ]]

```

```
print(np.linalg.inv(mat2))  
[[-0.09090909  0.45454545]  
 [ 0.27272727 -0.36363636]]
```

Statistics

```
arr1=np.array([83,34,53,23,76])  
print(np.mean(arr1))  
53.8  
print(np.std(arr1))  
23.181026724457222  
print(np.median(arr1))  
53.0  
print(np.var(arr1))  
537.36
```

Trigonometry

```
print(np.pi)  
3.141592653589793  
rad = [90,30,45]  
for i in rad:  
    print(np.sin(i))  
0.8939966636005579  
-0.9880316240928618  
0.8509035245341184
```

Arithmetic Operation

```
a=np.array({8,9,1})  
b=np.array([2,5,8])  
print(np.sum((a,b)))
```