```
#find square of each number & store in the list.
```

```
lst=[1,2,3,4,5]
#list comprehensive
result=[i**2 for i in lst]
result
```

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

```
!pip install numpy
```

```
Requirement already satisfied: numpy in /usr/local/lib/python3.11/dist-packages (1.26.4)
```

```
import numpy as np
```

```
arr = np.array([1,2,3,4,5])
arr
```

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

```
lst
```

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

```
lst*2
```

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

```
arr * 2
```

```
array([ 2,  4,  6,  8, 10])
```

```
lst = range(10000)
%timeit [i**2 for i in lst]
```

```
640 µs ± 10.7 µs per loop (mean ± std. dev. of 7 runs, 1000 loops each)
```

```
arr = np.array(range(10000))
%timeit arr**2
```

```
6.46 µs ± 1.14 µs per loop (mean ± std. dev. of 7 runs, 100000 loops each)
```

```
arr=[1,2,3]
arr[0]
```

```
1
```

```
lst=[2,'kantha',15.2]
lst[1]
```

```
#create 1-D array
arr=np.array([1,2,3,4,5])
arr
```

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

```
type(arr)
```

```
numpy.ndarray
```

```
arr.dtype
```

```
dtype('int64')
```

```
arr.ndim
```

```
1
```

```
arr = np.array([[2,2,3],[4,5,6]])
arr
```

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

```
type(arr)
```

```
numpy.ndarray
```

```
type(arr)
```

```
numpy.ndarray
```

```
arr.dtype
```

```
dtype('int64')
```

```
arr.ndim
```

```
2
```

```
arr.shape
```

```
(2, 3)
```

```
a=np.array([1,2,3,4,5,6,7,8])
print(a.ndim, a.shape)
#quiz question
```

```
1 (8,)
```

```
list(range(10))
```

```
[0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
```

```
list(range(1,10,2))
```

```
[1, 3, 5, 7, 9]
```

```
np.arange(1,10,0.5)
```

```
array([1. , 1.5, 2. , 2.5, 3. , 3.5, 4. , 4.5, 5. , 5.5, 6. , 6.5, 7. ,
       7.5, 8. , 8.5, 9. , 9.5])
```

```
arr = np.array([1,2,3,4.0])
arr
```

```
array([1., 2., 3., 4.])
```

```
#indexing
m1=np.arange(12)
m1
```

```
array([ 0,  1,  2,  3,  4,  5,  6,  7,  8,  9, 10, 11])
```

```
m1[11]
```

```
11
```

```
m1[[4,2,1,0,2]]
#access multiple indexes
```

```
array([4, 2, 1, 0, 2])
```

```
m1[[4,2,-1]]
```

```
array([ 4,  2, 11])
```

```
#slicing
m1
```

```
array([ 0,  1,  2,  3,  4,  5,  6,  7,  8,  9, 10, 11])
```

```
m1[2:6]
```

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

```
m1[::-1]
```

```
array([11, 10,  9,  8,  7,  6,  5,  4,  3,  2,  1,  0])
```

```
m1[-5:-1]
```

```
array([ 7,  8,  9, 10])
```

```
m1[-5:-1:-1]
```

```
array([], dtype=int64)
```

```
import numpy as np
a = np.array([0,1,2,3,4,56,7,8,9,10])
a[4:] = 5
print(a)
```

```
[0 1 2 3 5 5 5 5 5 5]
```

```
import numpy as np
Even = np.arange(22,70,2)
print(Even)
#assignments
```

```
[22 24 26 28 30 32 34 36 38 40 42 44 46 48 50 52 54 56 58 60 62 64 66 68]
```

```
odd = np.arange(21,70,2)
print(odd)
```

```
[21 23 25 27 29 31 33 35 37 39 41 43 45 47 49 51 53 55 57 59 61 63 65 67
 69]
```

```
import numpy as np
x = np.array([-5, 9 , 20 , 25, -3, 5, 16, 10,-8])
x[(x >= -5) & (x <= 15)] *= -1
print(x)
```

```
[  5  -9  20  25   3  -5  16 -10  -8]
```

```
import numpy as np
m1 = np.arange(12)
m1
```

```
array([ 0,  1,  2,  3,  4,  5,  6,  7,  8,  9, 10, 11])
```

```
m1 < 6
```

```
array([ True,  True,  True,  True,  True,  True, False, False, False,
       False, False, False])
```

```
m1[m1<6]
```

```
array([0, 1, 2, 3, 4, 5])
```

```
m1[m1%5==0]
```

```
array([ 0,  5, 10])
```

```
m1[m1%2==0]
```

```
!gdown 1c0ClC8SrPwJq5rrkyMKyPn80nyHcFikK
#download text file
```

```
Downloading...
From: https://drive.google.com/uc?id=1c0ClC8SrPwJq5rrkyMKyPn80nyHcFikK
To: /content/survey.txt
100% 2.55k/2.55k [00:00<00:00, 4.20MB/s]
```

```
#load text file
score = np.loadtxt('survey.txt',dtype='int')
score
```

```
array([ 7, 10,  5, ...,  5,  9, 10])
```

```
#nps = %promoters - %detractors
```

```
len(score)
```

```
1167
```

```
promoters = score[score>=9]
no_promoters =len(promoters)
no_promoters
```

```
609
```

```
detractors = score[score<=6]
no_detracters = len(detractors)
no_detracters
```

```
332
```

```
per_promoters = (no_promoters/len(score))*100
per_detracters = (no_detracters/len(score))*100
nps = per_promoters - per_detracters
nps
```

```
23.73607540702657
```

```
import numpy as np
x = np.array([-5, 9 , 20 , 25, -3, 5, 16, 10,-8])
x[(x >= -5) & (x <= 15)] *= -1
print(x)
```

```
[  5  -9  20  25   3  -5  16 -10  -8]
```

```
import numpy as np

def seq(start, length, step):

    sequence = start + length * step
    return np.arange(start, sequence, step, dtype=int)

start = 5
length = 10
step = 3

answer = seq(start,length, step)
print(answer)
```

```
[ 5  8 11 14 17 20 23 26 29 32]
```

```
import numpy as np

def seq(start, length, step):

    sequence = start + length * step
    return np.arange(start, sequence, step, dtype=int)

start = 5
length = 10
step = 3

ans = seq(start, length, step)
print(ans)
```

```
[ 5  8 11 14 17 20 23 26 29 32]
```

```
import numpy as np
arr = np.array([1, 2, 3, 4])
print(arr[2] + arr[-2])
```

```
6
```