

# Numpy in python

*on December 04, 2019*



## What is Numpy ?

- Numpy is the fundamental package for scientific computing in python.
- It is an open source numerical python library.
- It contains a multi-dimensional array and matrix data structures.
- The library contains a large number of mathematical, algebraic and transformation functions.
- It is a wrapper around a library implemented in C.
- It is widely used in machine learning and artificial intelligence projects.
- website : <https://numpy.org/>
- Documentation : <https://numpy.org/doc/1.17/user/index.html>

## How to install Numpy ?

```
pip install numpy
```

## numpy vs list

For storing numerical values we can use list also and do mathematical computation on it. But numpy performance is much better than python list. Let's see why.

The following python code will plot a graph between space and number of element stored. Run this code and see the result.

```
import numpy as np
import sys
import matplotlib.pyplot as plt

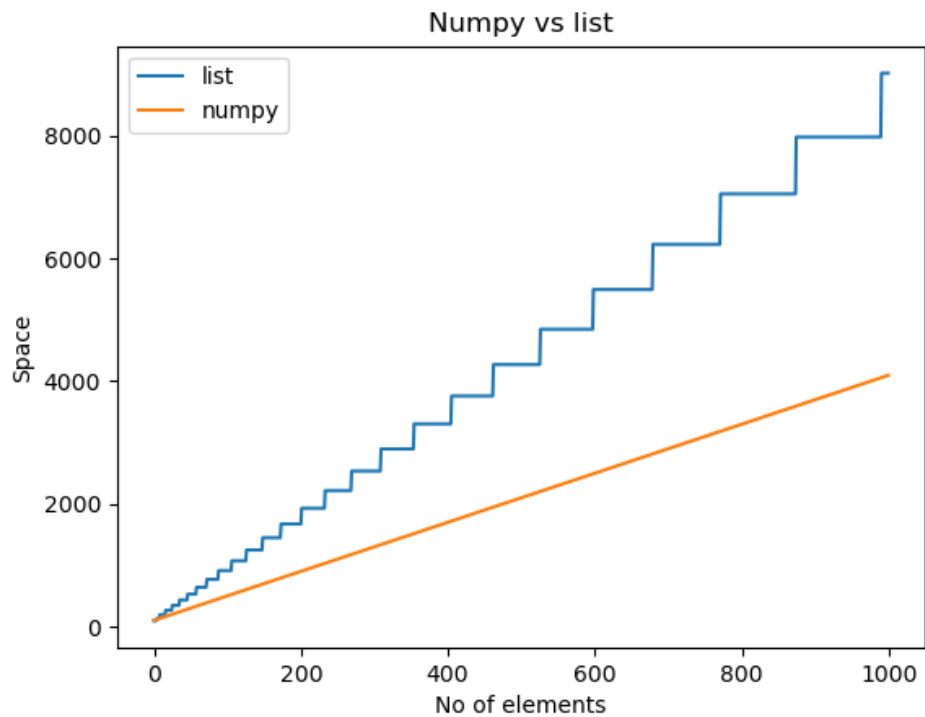
#save the results for plotting
list_size = list()
numpy_size = list()

list1 = list()
numpy1 = None

#create a list and np array of size 1 to 1000
for i in range(1,1001):
    list1.append(i)
    numpy1 = np.array(list1)
    list_size.append(sys.getsizeof(list1))
    numpy_size.append(sys.getsizeof(numpy1))

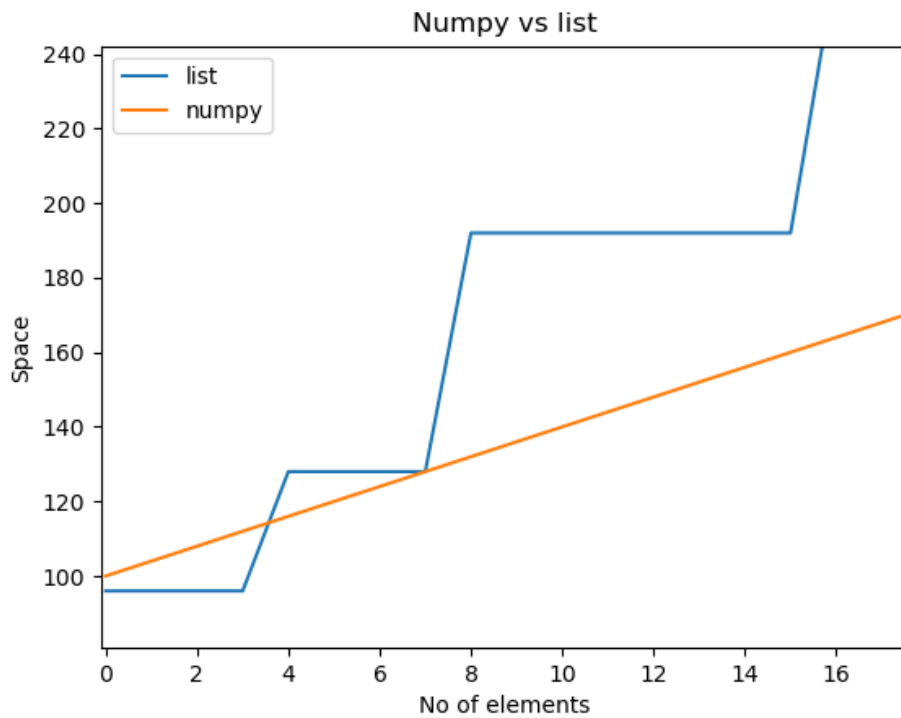
#plotting
plt.plot(list_size, label="list")
plt.plot(numpy_size, label="numpy")
plt.xlabel('No of elements')
plt.ylabel('Space')
plt.title('Numpy vs list')
```

```
plt.legend()  
plt.show()
```



As it is clear now, **numpy takes lesser space than list**.

But if you store less elements (<10) then numpy array will take much space.



Let's check its performance in terms of execution time.

```
import numpy as np
import sys
import matplotlib.pyplot as plt
import time

#create a huge length list and numpy array
l = [1,2,3,4,5,6,7,8,9,10]
l = 10000000*l
n = np.array(l)

#computing time to find sum of all elements in list
start_time = time.time()
m1 = sum(l)
end_time = time.time()
print((end_time-start_time)*1000, "ms")

#computing time to find sum of all elements in numpy
start_time = time.time()
```

```
mn = np.sum(n)
end_time = time.time()
print((end_time-start_time)*1000, "ms")
```

Run the above code and you can easily see the difference. I got the following results.

1468.7044620513916 ms

62.50905990600586 ms

As you can see numpy is almost 25 times faster than list. So, **numpy is faster than list.**

**So, these are the reasons why numpy is better than list.**

**Update :**

**Run this code for plotting comparison graph between list and numpy based upon execution time.**

```
import numpy as np
import sys
import matplotlib.pyplot as plt
import time

#save the results for plotting
list_exec_time = list()
np_exec_time = list()

# 1 Million length list
```

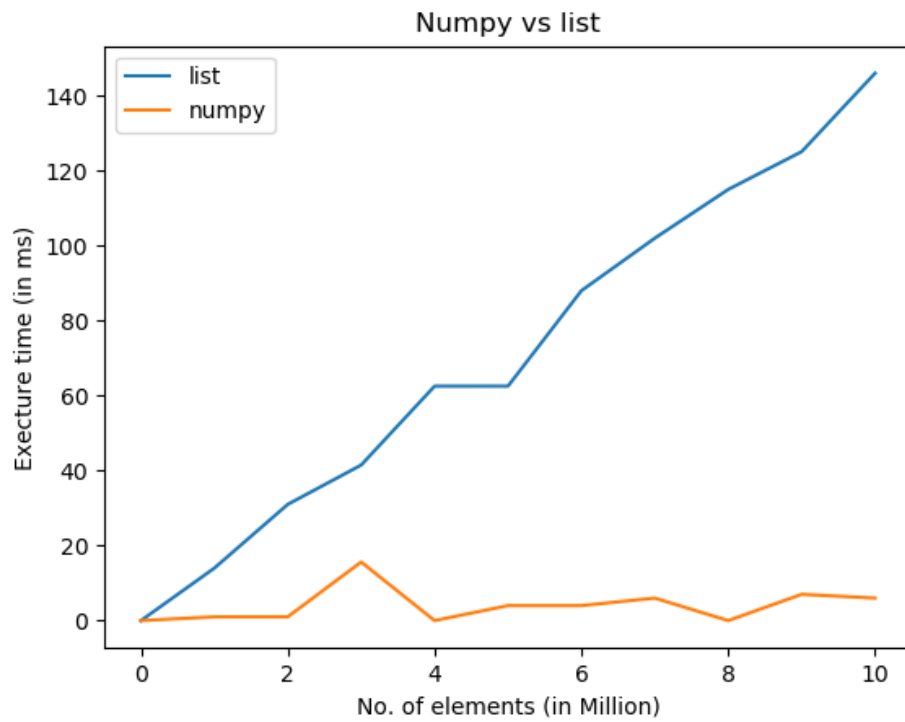
```
l = 100000*[1,2,3,4,5,6,7,8,9,10]

#create a list and np array of size 1 to 1000
for i in range(11):
    list1 = i*l

    start_time = time.time()
    list_sum = sum(list1)
    end_time = time.time()
    list_exec_time.append((end_time-start_time)*1000)

    np1 = np.array(list1)
    start_time = time.time()
    np_sum = np.sum(np1)
    end_time = time.time()
    np_exec_time.append((end_time-start_time)*1000)

#plotting
plt.plot(list_exec_time, label="list")
plt.plot(np_exec_time, label="numpy")
plt.xlabel('No. of elements (in Million)')
plt.ylabel('Execution time (in ms)')
plt.title('Numpy vs list')
plt.legend()
plt.show()
```



MACHINELEARNING

PYTHON



Enter your comment...