

# Numpy vs Lists

```
In [2]: #Let's define a list in python.  
heights = [74, 75, 72, 72, 71]
```

```
In [3]: # Print the heights.  
heights
```

```
Out[3]: [74, 75, 72, 72, 71]
```

```
In [6]: # Try to multiply heights with a scalar.  
heights * 2.54
```

```
-----  
TypeError                                Traceback (most recent call last)  
<ipython-input-6-e7573032a4ae> in <module>  
      1 # Try to multiply height with a scalar.  
----> 2 heights * 2.54  
  
TypeError: can't multiply sequence by non-int of type 'float'
```

```
In [7]: import numpy as np
```

```
In [9]: # Deine a NumPy array  
np_heights = np.array([74, 75, 72, 72, 71])
```

```
In [10]: np_heights
```

```
Out[10]: array([74, 75, 72, 72, 71])
```

```
In [11]: # Print the type of a NumPy array.  
type(np_heights)
```

```
Out[11]: numpy.ndarray
```

```
In [12]: # Multiple height (NumPy array) with a scalar.
```

```
np_heights * 2.54
```

```
Out[12]: array([187.96, 190.5 , 182.88, 182.88, 180.34])
```

```
In [ ]:
```

NumPy comes with its own set of methods and operations

```
In [40]: # Let's define two lists and perform '+' operation on that.  
list_1 = [1,2,3]  
list_2 = [4,5,6]  
list_1 + list_2
```

```
Out[40]: [1, 2, 3, 4, 5, 6]
```

```
In [41]: # Let's define two NumPy array and perform '+' operation on that.  
np1 = np.array([1,2,3])  
np2 = np.array([4,5,6])  
np1 + np2
```

```
Out[41]: array([5, 7, 9])
```

Working with N-D Arrays

```
In [45]: np_heights
```

```
Out[45]: array([74, 75, 72, 72, 71])
```

```
In [46]: type(np_heights)
```

```
Out[46]: numpy.ndarray
```

```
In [ ]:
```