

- -> notebooks from this lecture: <https://github.com/ine-rmotr-curriculum/freecodecamp-intro-to-numpy>
- -> broadcasting and vectorised operations
- -> multi-dimensional arrays in numpy
- -> these are optimised to be fast
- **-> vectorising an operation**
 - -> passing an entire vector into a function
 - -> doing operations on the entire vector
 - -> broadcasting it to each one of the elements in the operation
 - -> we are applying the operations to each of the elements in the array
 - -> numpy is immutable
 - -> do an operation on an array and it returns a new array
 - -> += -=, times equals
 - -> for example, you can add 100 to each element in an array by taking the variable which stored the array and adding 100 to it
 - -> list comprehensions
- **-> you can times two vectors together, and two arrays together as well**
 - -> you can do a scalar operation on an array
 - -> or you can operate two arrays together
 - -> you need to the arrays to have the same shape for this
- **-> why vectorised operations are important**
 - -> they are commonly used
 - -> for data visualisation
- -> the next lecture -> boolean arrays
- **-> question**

What is the value of a after you run the following code?

```
a = np.arange(5)
a + 20
```

[20, 21, 22, 24, 24]

[0, 1, 2, 3, 4] <- This one, because the value of the array doesn't change after it's been operated on

[25, 26, 27, 28, 29]