- -> notebooks from this lecture: https://github.com/ine-rmotr-curriculum/freecodecamp-intro-to-numpy
- ***-> the type of data you are working with, what it represents***
  - -> e.g the age of a person
  - -> different requirements in terms of storage size, depending on the context
  - ***-> you can calculate the number of its required to store the numbers you want to store***
    - -> 2^n numbers with n bits
    - -> the number of bits required to store the amount of numbers we want to store
    - -> we can store billions of numbers
    - -> being efficient and optimising data from financial transactions, for example
    - -> numpy has advanced numeric processing -> this allows you to select the number of bits you want to take for an integer
    - -> bytes / bits
    - -> you can calculate the number of bits required, and then the number of bits it is actually reserving for the calculation -> and the latter is a lot larger in Python, because it's a higher level language
      - -> this is one of the reasons we use numpy
      - -> you can control the amount of bits you want the memory to take
      - -> but in Python because it's a higher level language, it wouldn't be as specific with this number which would end up making the calculations more inefficient than they needed to be
- ***-> numpy for arrays***
  - -> lists, dictionaries <- these aren't optimised for higher level computing
  - -> if we have a list of numbers
  - -> it won't be guaranteed that the list will contain all of those numbers
  - -> you can't rely directly on advanced CPUs for transforming matrices, because Python is a high level language -> but with numpy, you can do this
  - -> numpy lets you select the number of bits you want to allow it to store an integer
  - -> in numpy, you can create numbers / control their size in terms of bits
  - -> np int <- and then you can specify the number of bits you want the computer to use
  - -> you need fast array processing
  - -> in bottlenecks, working with larger amounts of data -> this is when you use numpy
  - -> this is efficient
  - -> binary arithmetic and how numbers / computer architecture works
- ***-> question***
  - ***About how much memory does the integer 5 consume in plain Python?***
    - 32 bits
    - 20 bytes <- This one
    - 16 bytes
    - 8 bits