

- Create a Numpy Array containing numbers from 1 to 64
  - Plot the array as a line plot using matplotlib
  - Change the created array dimensions to 8 x 8
  - What happens if you:
    - Change the 8x8 matrix shape to (2, 4)?
    - Change the 8x8 matrix shape to (4, 2)?
    - Change the 8x8 matrix shape to (3, 3)?
    - Change the 8x8 matrix shape to (2, 2, 2)?
    - Change the 8x8 matrix shape to (2, -1, 2)?
- Create a new array containing numbers from 0-7
  - What is the shape of the array?
  - What happens if you add the array to the 8x8 matrix?
  - What happens if you resize the array to (1, 8) and add to the 8x8 matrix?
  - What happens if you resize the array to (8, 1) and add to the 8x8 matrix?
  - (Hint:- You may want to look at Broadcasting)
  - Write your findings and observations down in the cell below
- Create a matrix containing random numbers of size 4x4
  - Pad the matrix with zeros to create a matrix of size 6x6
  - Write to crop the 6x6 matrix back into the original 4x4 matrix (Numpy array indexing may be useful?)
- Learn about vectorization of matrix operations in numpy
- Create two matrices of shape (2, 4) and perform the following, if possible:
  - Element wise multiplication
  - How to transpose the matrix?
  - Can they be multiplied together (matrix multiplication)? If yes, perform the multiplication, else show a way to multiply the matrices together.
- Create the matrix shown below:

12	20	30	0
8	12	2	0
34	70	37	4
112	100	25	12

- 
- Slice the array to obtain subarrays shown in each color
- Find the:
  - unique elements in the array

- Number of unique elements
  - The maximum element
  - The minimum element
  - The index of the maximum element
  - The index of the minimum element
  - The mean of the array
  - The standard deviation
- Perform the following operations:
  - Sum the array column wise (Read about axis in numpy)
  - Sum the array row wise
  - Find maximum element for each column
  - Find maximum element for each row
  - Find the mean across each column
  - Find the mean across each row
  - Change the datatype of the array to : int, float and object
- Considering that the rows represent each data point and columns represent the features which we will feed to a ML model, perform the following feature scaling operations (without SKLearn):
  - Normalization
  - Standardization
- Load an image using opencv, which gives you a numpy array and inspect its shape. Interpret the shape of the array.
  - Plot each color channel individually using matplotlib
- Create a 3d matrix with 1s in the diagonal, 0 in all other positions
- Implement 1d convolution in numpy. Given any (1, n) shape array, it should perform convolution with a given (1, m) array
- Implement 2d convolution in numpy. And show the output before and after convolution on any image.