NUMPY

numpy as np

np.__version__

type(array_variable)

np.array([_,_,_,_])

array_variable.ndim

array_variable.shape

array_variable(operator)numeric

Vectorized math functions
np.sqrt(array_variable)
np.round(array_variable)
np.floor(array_variable)
np.ceil(array_variable)
np.pi

any condition written using array_ variable inside print or into a variable gives/ results the elements which satisfy the condition
ex:- scores[scores < 60] = 0
print(scores[scores >= 90])

variable = np.random.default_rng()
variable.integers(low = {numeric},high = {numeric},size=(number_of_matrix,rows,coloumns))
new_variable = np.random.default_rng(seed = 1)

for floating pointers:-
np.random.uniform(numeric},high = {numeric},size=(number_of_matrix,rows,coloumns)

for shuffling:-
varibale.shuffle(array_variable)

Slicing
variable_name[start:end:step]
for slicing coloumn section :-
variable_name[row_section,coloumn_section]
if want to display all elements in row and edit in coloumn make sure to place a clon in row side
ex:- s[:,1:6:2]

Filtering
array_variable = array_variable[condition with array_variable]
use & , | , ^ (similar to C language)
np.where(condition , array_variable , replicateed_value_if_condition is false)

Broadcasting
used to virtually expand dimensions
ex array_variable * array_variable

Aggregate Functions
np.sum(array_variable)
np.mean(array_variable)
np.std(array_variable)
np.var(array_variable)
np.min(array_variable)
np.max(array_variable)
np.argmin(array_variable)
np.argmax(array_variable)
np.sum(array_variable,axis=1) #r
np.sum(array_variable,axis=0) #c

array_variable (arthimetic_operator) array_variable