**NumPy Cheat Sheet**

|  |  |  |
| --- | --- | --- |
| **Name (functions,objects)** | **Syntax with Examples** | **uses** |
| Install NumPy | >>pip install NumPy | working with arrays,faster than lists. |
| Import conventions | Import numpy as np  arr=np.array([1,2,3,4]) |  |
| creating arrays | arr=np.array([1,2,3,4]) –1-D  arr=np.array([[1,2,3],[4,5,6]]) -2-D  arr=np.array([[1,2,3],[4,5,6],  [7,8,9]] | Creating arrays using arr=np.array |
| Data types | * i - integer * b - boolean * u - unsigned integer * f - float * c - complex float * m - timedelta * M - datetime * O - object * S - string * U - unicode string * V - fixed chunk of memory for other type ( void ) | NumPy has some extra data types, and refer to data types with one character, like i for integers, u for unsigned integers etc. |
| a.shape | a = np.array([[1,2],[1,1],[0,0]])  print(np.shape(a)) | The shape attribute of NumPy array a keeps a tuple of integers. Each integer describes the number of elements of the axis. |
| a.ndim | print(np.ndim(a)) | The ndim attribute is equal to the length of the shape tuple. |
|  | a = np.array([[2, 0], [0, 2]])  b = np.array([[1, 1], [1, 1]])  print(a\*b) | The asterisk (star) operator performs the Hadamard product,i.e., multiplies two matrices with equal shape element-wise. |
| Slicing arrays | a=np.array(1,2,3,4,5,6,7)  print(a[1:5]) | Slicing array from 1 to 5 |
| copy | a=np.array(1,2,3,4,5)  x=a.copy() |  |
| shaping | a=np.array([1,2,3,4],[5,6,7,8])  Print(a.shape) | Print the shape of 2-D array |
|  |  |  |
| Reshaping | arr = np.array([1,2,3,4,5,6,7,8,9])  newarr=array.reshape(4,3)  print(newarr) | Shape from 1-D to 2-D |
| Joining arrays | arr1 = np.array([1,2,3])  arr2 = np.array([4, 5, 6])  arr = np.concatenate((arr1, arr2))  print(arr) |  |
| Splitting arrays | arr = np.array([1, 2, 3, 4, 5, 6])  newarr = np.array\_split(arr, 3)  print(newarr) | Splitting array in 3 parts usinf=g np.array.split function |
| Searching arrays | arr = np.array([1, 2, 3, 4, 5, 4, 4])  x = np.where(arr == 4)  print(x) | Search any array using np.where function |
| Sorting arrays | arr = np.array(['kusum', 'shruti', 'saurav'])  print(np.sort(arr)) | Sorting arrays alphabetically |
| Matrix addition | A = np.array([[2, 4], [5, -6]])  B = np.array([[9, -3], [3, 6]])  C = A + B  print(C) | Addition of two matrix,we can subtract,divide,multiply two matrices |
| Rank of matrix | a = np.arange(1, 10)  print(a)  rank = np.linalg.matrix\_rank(a)  print(rank) | Rank of any matrix by using matrix.rank(a) function |
| Determinant of matrix | a = np.array([[2, 2, 1],                 [1, 3, 1],                 [1, 2, 2]])  print(a)  det = np.linalg.det(a)  print( np.round(det)) | Find out determinant of any matrix |
| Shape of matrix | a=np.matrix(12)  print('The matrix with 12 random values:', a) | Shape of any matrix |
| True inverse | a = np.array([[2, 2, 1],                 [1, 3, 1],                 [1, 2, 2]])  print(a)  det = np.linalg.det(a)  print(np.round(det))  inv = np.linalg.inv(a)  print("Inverse of a = ")  print(inv) |  |
| Pseudo inverse | a = np.array([[2, 8],                 [1, 4]])  print(a)  det = np.linalg.det(a)  print(np.round(det))  pinv = np.linalg.pinv(a)  print("Pseudo Inverse of a = ")  print(pinv) |  |