List

1. The insert() method inserts an item at the specified index.
   Mylist.insert(Index, Value)
2. To add an item to the end of the list, use the append() method
   Mylist.append(Value)
3. To append elements from another iterable(list,tuple,dict) to the current list, use the extend() method.
   Mylist.extend(IterableName)
4. The remove() method removes the specified item.
   Mylist.remove(value)
5. The pop() method removes the specified index.
   Mylist.pop(index)
6. If you do not specify the index, the pop() method removes the last item.
   Mylist.pop()
7. The del keyword also removes the specified index:
   del Mylist[index]
8. The del keyword can also delete the list completely.
   del Mylist
9. The clear() method empties the list.
   Mylist.clear()
10. List objects have a sort() method that will sort the list alphanumerically, ascending, by default.
    Mylist.sort()
11. To sort descending, use the keyword argument reverse = True:
    Mylist.sort(reverse = True)
12. to make a copy, use the built-in List method copy().
    Mylist2 = Mylist1.copy()
13. Another way to make a copy is to use the built-in method list().
    mylist = list(thislist)
14. One of the easiest ways to join lists are by using the + operator.
    Mylist3 = Mylist1 + Mylist2
15. Another way To join two lists
    ```
    for x in list2:
      list1.append(x)
    ```
16. To check length of list
    Print(len(Mylist))

Numpy

1. np.array() :- to create array

   np.array([[1,2,3],[4,5,6]])
2. ndim - to check dimension of array

   returns an integer

   print(arrayName.ndim)
3. ndmin :- to create n dimensional arrays
   ```
   arr = np.array([1, 2, 3, 4], ndmin=5)
   ```
4. dtype :- to check data type of array

   returns data type

   print(arr.dtype)

   Or

   arr = np.array([1, 2, 3, 4], dtype='S')

   similarly

   arr = np.array([1, 2, 3, 4], dtype='i')
5. astype() :- The astype() function creates a copy of the array, and allows you to specify the data type as a parameter.

   newarr = arr.astype('i')
6. copy() :- to make copy of an array

   returns an array

   newarr = arr.copy()
7. base :- to check if array owns its data or not

   returns none if does not own and returns original array if it owns

   print(arrayname.base)
8. shape :- to check the number of elements in array for each dimension in row, column tuple form

   print(arr.shape)
9. np.ndenumerate(arrayname) :- to print with indices

   for idx,x in np.ndenumerate(arrayName)

   print(idx,x)
10. np.concatenate(arr1,arr2) :- Join two arrays

    arr3 = np.concatenate(arr1,arr2)
11. axis = 1 ; - used to join 2d arrays

    arr = np.concatenate((arr1, arr2), axis=1)
12. np.stack(arr1,arr2) :- to stack two arrays

    arr3 = np.stack((arr1, arr2), axis=1)
13. np.hstack(arr1,arr2) :- to stack along row

    arr3 = np.hstack((arr1,arr2)) # output will be in single row
14. np.vstack(arr1,arr2) :- to stack along columns

    arr3 = np.vstack((arr1,arr2)) # output will be in two rows for two arrays of 2*3

15. np.dstack(arr1,arr2) :- to stack along height
    arr3 = np.dstack((arr1,arr2)) # # output will be in three rows in 2*3 matrix

Matplotlib

- plt.plot(xCordinateArray,yCordinateArray) :- to plot lines
  plt.plot(xCordinatesArray,yCordinatesArray)
- plt.show() :- to show plot
  plt.show()
- plt.plot(xCordinateArray,yCordinateArray,'o') :- to plot without line(plotting only points)
  plt.plot(xCordinateArray,yCordinateArray,'o')
- plt.plot(xCordinateArray,yCordinateArray,marker = 'o') :- to plot with markers on line
  plt.plot(xCordinateArray,yCordinateArray,marker = 'o')
- marker|line|color :- standard form to format marker
- ms = 20 :- to format marker size
  plt.plot(ypoints, marker = 'o', ms = 20)
- mec = 'r' :- to color edge of the marker
  plt.plot(ypoints, marker = 'o', ms = 20, mec = 'r')
- mfc = 'r' :- to color fill of the marker
  plt.plot(ypoints, marker = 'o', ms = 20, mfc = 'r')
- linestyle = ':' :- to change linestyle
  plt.plot(xpoints, ypoints, linestyle = ':')
- color = 'r' :- to change line color
  plt.plot(ypoints, color = 'r')
- plt.xlabel("string- ") :- to set labels (same for y axis)
  plt.xlabel("Average Pulse")
- plt.title('string'):- to give it a title
  plt.title('distance-time graph')
- plt.grid(true) :- to enable grid
  plt.grid(true)
- plt.legend(["This week", "Last week"]) :- to add legends