WHAT IS AN ARRAY?

* An array is a data structure, which can hold more than one value at a time. It is collected or ordered series of elements of same type

IS PYTHON LIST IS SAME AS AN ARRAY?

* Python Arrays and lists have the same way of storing data.
* Arrays take only single data type elements but lists can have any type of data.
* Therefore, other than a few operations, the kind of operations performed on them are different.

HOW TO CREATE AN ARRAYS IN PYTHON?

* Arrays in Python can be created after importing the array module.
  1. Without Alias: Import array
  2. Using Alias: Import array as arr
  3. USING \*: from array import \*

ACCESSING ARRAY ELEMENTS

* Access elements using index values
* Indexing starts at 0 and not from 1. Hence, the index number is always 1 less than length of the array.
* Negative index values can be used as well. The point to remember is that negative indexing starts from the reverse order of traversal i.e from right to left.

BASIC ARRAY OPERATIONS

* Operations
  + Finding the length of an array
  + Adding/Changing element of an array
  + Removing/deleting elements of an array.
  + Array concatenation
  + Slicing
  + Looping through an Array

1. Finding the length of an array
   1. Length of an array is the number of elements that are actually present in an array.
   2. You can make use of len() function to achieve this.
   3. The len() function returns an integer value that is equal to the number of elements present in that array.
   4. Syntax: len(array\_name)
2. Adding Elements to an array
   1. Function used to add elements to an array:
      1. Append()
         * Used when you want to add a single element at the end of an array.
         * a.append(7),add value at end of array list
         * Output : 1,2,3,4,5,6,7
      2. Extend()
         * Used when you want to add more than one element at the end of an array.
         * a.extend([8,9,10,11,12])
         * Output : 1,2,3,4,5,6,7,8,9,10,11,12
      3. Insert()
         * Used when you want to add an element at a specific position in an array.
         * a.insert(2,8), first element shows index value and second element shows element you want to add in list.
         * Output: 1,2,8,3,4,5,6,7,8,9,10,11,12
3. Removing elements of an array
   1. Functions used to remove elements of an Array:
      1. Pop()
         * Used when you want to remove an element and return it.
         * a.pop(-1): remove last element of array list
         * a.pop(2): remove element from index 2
         * a.pop(-2): remove second last element
         * a.pop(): remove last element
         * output: 1,2,3,4,5,6,7,8,9
      2. Remove()
         * Used when you want to remove an element with a specific value without returning it.
         * a.remove(9): remove element 9 in first occurrence.
4. Array Concatenation
   1. Array concatenation can be done as follows using the + symbol
   2. b = arr.array('i',[1,2,3,4])

c = arr.array('i',[5,6,7,8])

d = arr.array('i')

d = b + c

d

Output: array('i', [1, 2, 3, 4, 5, 6, 7, 8])

* 1. concatenate same type data elements of an array.

1. Slicing an Array
   1. An array can be sliced using : symbol. The returns of range of elements that we have specified by index numbers.
   2. d[0:5]: shows elements from 0 index to 5 index.

d[0:-2]: shows elements from 0 index to -2 index but not show -1, -2 values.

d[::-1]: reverse array list of element

d: shows original array list

1. Looping through an Array
   1. We can loop through an array easily using for and while loops.
      1. For: iterates over the items of an array specified number of times.
         * for x in d:

print (x)

Output: 1,2,3,4,5,6,7,8

* + - * for x in d[0:-3]:

print(x)

Output: 1,2,3,4,5

* + - * for x in d[0:3]:

print (x)

Output: 1,2,3

* + 1. While: iterates over the elements until certain condition is met.
       - temp=0

while temp < d[5]:

print(d[temp])

temp=temp+1

Output: 1,2,3,4,5,6

* + - * temp=0

while temp < len(d):

print(d[temp])

temp=temp+1

Output: 1,2,3,4,5,6,7,8