

PYTHON ASSIGNMENT

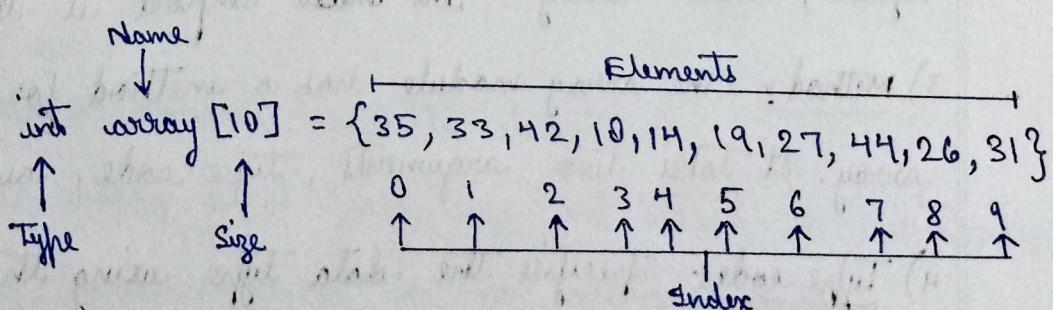
UNIT - III

1) (a) What is an Array? Can we use List as a substitute of arrays? Justify

Ans:- An array is a collection of items stored at contiguous memory locations. The idea is to store multiple items of the same type together. Arrays are used to store multiple values in one single variable. Arrays can be multi-dimensional, and all elements in an array need to be of the same type, all integers, or all floats. A user can treat lists as arrays. Most of the data structures make use of arrays to implement their algorithms. Following are the important terms to understand the concept of Array:

- Elements:- Each item stored in an array is called an element.
 - Index's Each location of an element in an array has a numerical index, which is used to identify the element.

Arrays can be declared in various ways in different languages.
Below is an illustration.



Python does not have native array data structure, but it has the list which is mutable, which means we can modify the content present within the list. We can

store data of heterogeneous data types. List is much more general and can be used as a multidimensional array quite easily.

(Q) How to use array module in the program? What are the advantages of using array module. Explain in detail.

Ans:- Array is created in Python by importing array module to the python program. Then the array is declared as shown below.

```
from array import *
arrayName = array (Typecode, [Initializers])
                (OR)
import array
arrayName = array.array (typecode, [array, items])
```

→ Array Syntax:-

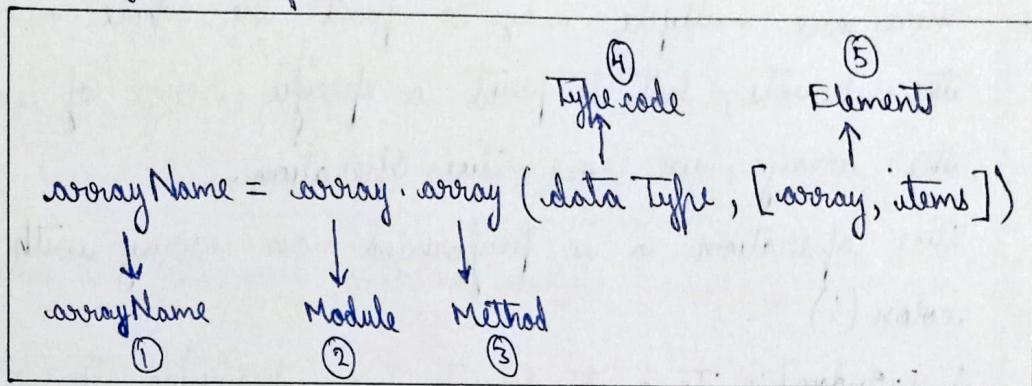
- 1) Identifier: specifies a name like usually you do for variables
- 2) Module: Python has a special module for creating array in Python, called "array". You must import it before using it.
- 3) Method: The array module has a method for initializing the array. It takes two arguments, type code, and elements.
- 4) Type code: specifies the data type using the type codes available.

Ex:- int → 'i'

float → 'f'

double → 'd'

②
5) Elements :- Specifies the array elements within the square brackets, for example [130, 450, 103]



→ Example Program:-

Coder

```
import array as arr
numbers = arr.array('i', [1, 2, 3])
numbers.append(4)
print(numbers)
```

Output

```
array('i', [1, 2, 3, 4])
```

→ Advantages of array data structure:

- 1) Arrays store multiple data of similar types with the same name.
- 2) It allows random access to elements.
- 3) As the array is of fixed size and stored in contiguous memory locations, there is no memory shortage or overflow.
- 4) It is helpful to store any type of data with a fixed size.

Q) Write short notes on slicing an array in different ways.

Ans: There are multiple ways to print the whole array with all the elements, but to print a specific range of elements from the array, we use slice operation.

Slice operation → is performed on array with the use of colon (:) .

- 1) [: Index] :- To print elements from beginning to a range
- 2) [: - Index] :- To print elements from end.
- 3) [Index :] :- To print elements from specific index till the end.
- 4) [Start Index : End Index] :- To print elements within a range.
- 5) [:] :- To print whole list with the use of slicing operation.
- 6) [::-1] :- To print whole array in reverse order.

Example program

Codes

```
import array as arr.  
numbers_list = [2, 5, 62, 5, 42, 52, 48, 5]  
numbers_array = arr.array('i', numbers_list)  
print (numbers_array[2:5])  
print (numbers_array[:-5])  
print (numbers_array[5:])  
print (numbers_array[:])
```

3

Output:-

```
array ('i', [62, 5, 42])
array ('i', [2, 5, 62])
array ('i', [52, 48, 5])
array ('i', [2, 5, 62, 5, 42, 52, 48, 5])
```

3) Explain the following with a sample program.

(i) Creating an array.

Ans An array is created in Python by importing array module to the Python program.

Code:-

```
from array import *
array1 = array('i', [10, 20, 30, 40, 50])
for x in array1:
    print(x)
```

Output:-

```
10
20
30
40
50
```

(ii) Accessing array elements

Ans. We can access each element of an array using the index of the element.

Code:-

```
from 'array' import *
array1 = array('i', [10, 20, 30, 40, 50])
print(array1[0])
print(array1[1])
```

Output:

10

20

(iii)

Different ways to add an element into an array.

Ans. Insert Operation is used to insert one or more data elements into an array. Based on the requirement, a new element can be added at the beginning, end, or any given index of array. We add a data element at the middle of the array using the python in-built "insert() method".

Code:

```
from array import *
array1 = array('i', [10, 20, 30, 40, 50])
array1.insert(1, 60)
for xc in array1:
    print(xc)
```

Output:

10

60

20

30

40

50

We can add one item to a list using "append() method"

or add several items using "extend() method".

We can also concatenate two arrays using '+' operator.

Code

```
import array as arr  
numbers = arr.array ('i', [1,2,3])  
numbers.append(4)  
print (numbers)
```

Output

```
array ('i', [1,2,3,4])
```

Code

```
import array as arr  
numbers = arr.array ('i', [1,2,3])  
numbers.extend ([5,6,7])  
print (numbers)
```

Output

```
array ('i', [1,2,3,5,6,7])
```

Code

```
import array as arr  
odd = arr.array ('i', [1,3,5])  
even = arr.array ('i', [2,4,6])  
numbers = arr.array ('i')  
numbers = odd + even  
print (numbers)
```

Output

```
array ('i', [1,3,5,2,4,6])
```

- (iv) Different ways to delete an element from array.
- Ans Deletion refers to remove an existing element from the array and re-organizing all the elements of an array. We remove a data element at the middle of the array using the python in-built "remove() method".

Code

```
from array import *
array1 = array('i', [10, 20, 30, 40, 50])
array1.remove(40)
for x in array1:
    print(x)
```

Output

```
10
20
30
50
```

An error arises if element doesn't exist in the set.
"pop()" function can also be used to remove and return an element from the array, but by default it removes only the last element of the array; to remove an element from a specific position of the array, index of the element is passed as an argument to the "pop() method".

We can delete one or more items from an array using Python's "del" statement.

Ques

Code

```
import array as arr
number = arr.array ('i',[1,2,8,3,4])
del number[2]
print (number)
```

Output

```
array ('i',[1,2,3,4])
```

(v)

Searching an element.

Ans

You can perform a search for an array element based on its value or its index. We search a data element using the python in-built "index () method".

Code

```
from array import *
array1 = array ('i',[10,20,30,40,50])
print (array1.index(40))
```

Output

3

4)

Mention the advantages of using Numpy module instead of array module?

Ans

Numpy is a Python library used for working with arrays. It also has functions for working in domain of linear algebra, fourier transformation and matrices. Numpy was created in 2005 by Travis Oliphant. It is an open source project and you can use it freely. Numpy stands for Numerical Python.

Why use Numpy?

- 1) In python, we have lists that serve the purpose of arrays, but they are slow to process.
- 2) Numpy aims to provide an array object that is up to 50 times faster than traditional python lists.
- 3) The array object in Numpy is called ndarray, it provides a lot of supporting functions that make working with ndarray very easy.
- 4) Arrays are very frequently used in data science, where speed and resources are very important.
- 5) It consumes less memory.
- 6) It is fast as compared to the python lists.

Q

5) Write short notes on the following.

(i) arange.

Ans. The numpy arange function is a tool for creating numeric sequences in python. It returns an evenly spaced numeric values within an interval, stored as Numpy array.

Code:

```
import numpy as np
A = np.arange(4)
print('A = ', A)
```

Output:

A = [0 1 2 3]

(ii) reshape

Ans. Reshaping means changing the shape of an array. The shape of an array is the number of elements in each dimension. By reshaping, we can add or remove or change the number of elements in each dimension.

Code:

```
import numpy as np.
arr = np.array([1, 2, 3, 4, 5, 6, 7, 8, 9])
newarr = arr.reshape(3,3)
print(newarr)
```

Output:

[[1 2 3]
 [4 5 6]
 [7 8 9]]

(iii) Array of zeros and ones.

Ans. zeros() and ones() are the Numpy library functions to create two different arrays. zeros() function is used, to create an array based on the particular shape & type. All array elements are initialized to 0, which is created by the zeros() function. ones() function works like the zeros() function. But the elements of the array created, by the ones() function are initialized to 1.

Code

```
import numpy as np  
zeros_array = np.zeros((2,3))  
print(zeros_array)  
  
ones_array = np.ones((1,5), dtype = np.int32)  
print(ones_array)
```

Output

```
[[0. 0. 0.]  
 [0. 0. 0.]]  
  
[[1 1 1 1 1]]
```

(iv) Dimensions

Ans The number of dimensions and items in an array is defined by its shape, which is a tuple of n non-negative integers that specify the size of each dimension. The type of items in the array is specified by a separate data-type object (dtype), one of which is associated with each ndarray.

(2)

- Q) Write a program to perform the following operations on 2 dimensional matrix of 3×3 size.

- (i) Addition

Ans

Code

```
x = [[1, 2, 3],
      [4, 5, 6],
      [7, 8, 9]]
```

```
y = [[5, 8, 1],
      [6, 7, 3],
      [4, 5, 9]]
```

```
result = [[0, 0, 0],
          [0, 0, 0],
          [0, 0, 0]]
```

for i in range(len(x)):

for j in range(len(x[0])):

result[i][j] = x[i][j] + y[i][j]

for r in result:

print(r)

Output

[17, 15, 4]

[10, 12, 9]

[11, 13, 18]

(ii) Subtraction

Ans. Code:

```
x = [[12, 7, 3],  
     [4, 5, 6],  
     [7, 8, 9]]
```

```
y = [[5, 8, 1],  
     [6, 7, 3],  
     [4, 5, 9]]
```

```
result = [[0, 0, 0],  
          [0, 0, 0],  
          [0, 0, 0]]
```

for i in range(len(x)):

for j in range(len(x[0])):

$$\text{result}[i][j] = x[i][j] - y[i][j]$$

for sr in result:

print(sr)

Output

[7, -1, 2]

[-2, -2, 3]

[3, 3, 0]

③

(iii) Multiplication

Ans

Code

$$X = [[12, 7, 3], [4, 5, 6], [7, 8, 9]]$$

$$Y = [[5, 8, 1, 2], [6, 7, 3, 0], [4, 5, 9, 1]]$$

$$\text{result} = [[0, 0, 0, 0], [0, 0, 0, 0], [0, 0, 0, 0]]$$

for i in range ($\text{len}(X)$):

 for j in range ($\text{len}(Y[0])$):

 for k in range ($\text{len}(Y)$):

$$\text{result}[i][j] += X[i][k] * Y[k][j]$$

for o in range :

 print(o)

Output

$$[114, 160, 60, 27]$$

$$[74, 97, 73, 14]$$

$$[119, 157, 112, 23]$$

(iv)

Ans.

Transpose.

Code:

```
X = [[12, 7],  
     [4, 5],  
     [3, 8]]
```

```
result = [[0, 0, 0],  
          [0, 0, 0]]
```

```
for i in range(len(X)):
```

```
    for j in range(len(X[0])):
```

```
        result[j][i] = X[i][j]
```

```
for r in result:
```

```
    print(r)
```

Output:

```
[12, 4, 3]
```

```
[7, 5, 8]
```

(4)

7) Explain about types of arrays with example programs?

Ans. There are two types of arrays:

(i) One-Dimensional Arrays: - A one-dimensional array is a kind of linear array. It involves single sub-scripting. The '[]' brackets is used for the subscript of the array and to declare and access the elements from the array.

Code

```
from array import *
a = [1, 2, 3]
print (a)
Output:
[1, 2, 3]
```

(ii) Multi-Dimensional Arrays:

(a) Two-Dimensional Arrays: - An array involving two subscripts [], [] is known as a two dimensional array. They are also known as the array of the array. Two-dimensional arrays are divided into rows and columns and are able to handle the data of the table.

Code

```
from array import *
a = [[1, 2, 3], [4, 5, 6]]
print (a)
Output:
[[1, 2, 3], [4, 5, 6]]
```

(b) Three-dimensional Arrays : We use three-dimensional arrays to create two or more tables of the elements to declare the array elements.

Code :-

```
from array import *
a = [[[1,2,3],[4,5,6]],[[7,8,9],[10,11,12]]]
print(a)
```

Output :-

```
[[[1,2,3],[4,5,6]],[[7,8,9],[10,11,12]]]
```

8) Creation of single and multi dimensional arrays using numpy with example.

Ans. (i) One-dimensional arrays : An array that has 0-D arrays as its elements is called one-dimensional or 1-D array.

Code :-

```
import numpy as np
arr = np.array([1,2,3,4,5])
print(arr)
```

Output :-

```
[1 2 3 4 5]
```

(ii) Two-Dimensional Arrays: An array that has 1-D arrays as its elements is called a 2-D array. These are often used to represent matrices or 2nd order Tensors.

Code:

```
import numpy as np  
arr = np.array ([[1,2,3], [4,5,6]])  
print (arr)
```

Output:

```
[[1 2 3]  
 [4 5 6]]
```

(iii) Three-Dimensional Arrays: An array that has 2-D arrays (matrices) as its elements is called 3-D array. These are often used to represent a 3rd order Tensor.

Code:

```
import numpy as np  
arr = np.array ([[ [[1,2,3], [4,5,6]], [[1,2,3], [4,5,6]] ]])  
print (arr)
```

Output:

```
[[[1 2 3]  
 [4 5 6]]]
```

```
[[[1 2 3]  
 [4 5 6]]]
```