Lecture 12 – Arrays

- An **array** is a collection of homogeneous items where each item has its own index value, and the index starts from 0.

- Python doesn't have built-in support for Arrays, but we can import array and use them.

- There is another datatype similar to arrays in Python, i.e., Lists which are useful as arrays in Python but are different in a way that lists can hold any type of values, but Arrays store only similar type of values.

- lists are built-in datatype in Python whereas Arrays you have to import from array module.

- **Index**: is the number representing a value in the array and always start with 0.
- **element**: is the value in an array.
- **len():** is the total count of elements in an array.
- **append():** This is the method to add an element to the array.
- **remove():** is the method to remove an element from the array.

The array is stored in **contiguous** memory locations.

Example

Assume that you have the following :

array ('i', [1, 2 , 3, 4 , 5, 6, 7, 8, 9,10]), then to access these values we will use the following format.

a[0] => 1

a[1] => 2

a[2] => 3

a[3] => 4

a[4] => 5

a[5] => 6

a[6] => 7

a[7] => 8

a[8] => 9

a[9] => 10

You can also use a **for-in** loop also to loop through the given array:

```
for i in a
    print a[i]
```

Which will display the values from 1 to 10.

How to Create Arrays in Python?

Import the array method.

Example

```
import array as arr
```

Where:

arr => is an alias

The syntax to create an array is:

```
array(typecode  ,[initializer])
```

Where:

typecode is int or float or double or the type of value the array holds and

initializer is the optional value and can be of any type like list, string, or any iterable elements of a particular type.

The typecode table :

TypeCode	C Type	Python Type	Value (in Bytes)
i	signed int	int	4
I	Unsigned int	long	4
b	signed char	int	1
B	Unsigned char	int	1
h	signed short	int	2
H	Unsigned short	int	2
l	signed long	int	4
L	Unsigned long	int	4
f	float	float	4
d	double	float	8

Typecode: i (signed int)

```
import array as arr
a=arr.array('i', [10 , 20 ,30] )
print("Element at 0th index: " , a[0])
print("Element at 1st  index: " , a[1])
print("Element at 2nd index: " , a[2])
```

Sample Run:

```
Element at 0th index:  10
Element at 1st  index:  20
Element at 2nd index:  30
```

Typecode: I (unsigned int)

```
import array as arr
a=arr.array('I', [10 , 20 ,30] )
print("Element at 0th index: " , a[0])
print("Element at 1st  index: " , a[1])
print("Element at 2nd index: " , a[2])
```

Sample Run:

```
Element at 0th index:  10
Element at 1st  index:  20
Element at 2nd index:  30
```

Typecode: b (signed char)

```
import array as arr
a=arr.array('b', [10 , 20 ,30] )
print("Element at 0th index: " , a[0])
print("Element at 1st  index: " , a[1])
print("Element at 2nd index: " , a[2])
```

Sample Run:

```
Element at 0th index:  10
Element at 1st  index:  20
Element at 2nd index:  30
```

Typecode: B (unsigned char)

```
import array as arr
a=arr.array('B', [10 , 20 ,30] )
print("Element at 0th index: " , a[0])
print("Element at 1st  index: " , a[1])
print("Element at 2nd index: " , a[2])
```

Sample Run:

```
Element at 0th index:  10
Element at 1st  index:  20
Element at 2nd index:  30
```

Typecode: h (signed short)

```
import array as arr
a=arr.array('h', [10,20,30] )
print("Element at 0th index: " , a[0])
print("Element at 1st  index: " , a[1])
print("Element at 2nd index: " , a[2])
```

Sample Run:

```
Element at 0th index:  10
Element at 1st  index:  20
Element at 2nd index:  30
```

Typecode: H (unsigned short)

```
import array as arr
a=arr.array('H', [10,20,30] )
print("Element at 0th index: " , a[0])
print("Element at 1st  index: " , a[1])
print("Element at 2nd index: " , a[2])
```

Sample Run:

```
Element at 0th index:  10
Element at 1st  index:  20
Element at 2nd index:  30
```

Typecode: l (signed Long)

```
import array as arr
a=arr.array('l', [10,20,30] )
print("Element at 0th index: " , a[0])
print("Element at 1st  index: " , a[1])
print("Element at 2nd index: " , a[2])
```

Sample Run:

```
Element at 0th index:  10
Element at 1st  index:  20
Element at 2nd index:  30
```

Typecode: L (unsigned Long)

```
import array as arr
a=arr.array('L', [10,20,30] )
print("Element at 0th index: " , a[0])
print("Element at 1st  index: " , a[1])
print("Element at 2nd index: " , a[2])
```

Sample Run:

```
Element at 0th index:  10
Element at 1st  index:  20
Element at 2nd index:  30
```

Typecode: f (float)

```
import array as arr
a=arr.array('f', [10,20,30] )
print("Element at 0th index: " , a[0])
print("Element at 1st  index: " , a[1])
print("Element at 2nd index: " , a[2])
```

Sample Run:

```
Element at 0th index:  10.0
Element at 1st  index:  20.0
Element at 2nd index:  30.0
```

Typecode: d (double)

```
import array as arr
a=arr.array('d', [10,20,30] )
print("Element at 0th index: " , a[0])
print("Element at 1st  index: " , a[1])
print("Element at 2nd index: " , a[2])
```

Sample Run:

```
Element at 0th index:  10.0
Element at 1st  index:  20.0
Element at 2nd index:  30.0
```

Array Methods in Python

1. typecode() : This function **returns the value of typecode** used in the given array.

Syntax: **array.typecode()**

Example

```
import array as arr
a1 = arr.array('i', [100,200,300] )
print(a1.typecode)
```

Sample Run:

```
i
```

2. insert() : It **adds an element to the array before** the index

Syntax: array.insert(index, element)

Example

```
import array as arr
a = arr.array('i', [100,200,300])
a.insert(3, 400);
#looping through array a
for i in a:
   print(i)
```

Sample Run:

```
100
200
300
400
```

3. update() : It **updates a particular value** at an index to the new value.

Syntax: arrayname[index] = value

Example

```
import array as arr
a = arr.array('i', [100,200,300] )
a[1] = 400
for i in a:
    print(i)
```

Sample Run:

```
100
400
300
```

Example

```
import array as arr
num = arr.array('i', [1, 2, 3, 4, 5, 6])
num[0] = 10
print(num)
num[1:4] = arr.array('i', [100, 200, 300])
print(num)
```

Sample Run

```
array('i', [10, 2, 3, 4, 5, 6])

array('i', [10, 100, 200, 300, 5, 6])
```

4. delete() : This function **removes t**he element from the array.

Syntax: array.remove(element)

Example

```
import array as arr
a = arr.array('i', [100,200,300] )
a.remove(100)
for i in a:
    print(i)
```

Sample Run:

```
200
300
```

Example

```
import array as arr
num = arr.array('i', [2, 3, 4, 5, 6])
del num[3]
print(num)
```

Sample Run:

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

Example

```
import array as arr
num = arr.array('i', [2, 3, 4, 5, 6])
del num[3] # removing the fourth element
print(num)
del num
print(num)
```

Sample Run

```
array('i', [2, 3, 4, 6])
Traceback (most recent call last):
  File
"C:\Users\HG\AppData\Local\Programs\Python\Python3
10\test5.py", line 6, in <module>
    print(num)
NameError: name 'num' is not defined. Did you mean:
'sum'?
```

5. append() : This function **appends** the element to the **end** of the array.

Syntax: array.append(element)

Example

```
import array as arr
a = arr.array('i', [100,200,300] )
a.append(400)
for i in a:
        print(i)
```

Sample Run:

```
100
200
300
400
```

6. reverse() : This function **reverses the order of elements** in the given array.

Syntax : array.reverse()

Example

```
import array as arr
a = arr.array('i', [100,200,300] )
a.reverse()
for i in a:
    print(i)
```

Sample Run:

```
300
200
100
```

7. count() : This function **returns how many times the element occurred** in the given array.

Syntax: array.count(element)

Example

```
import array as arr
a3 = arr.array('i', [100,200,300,100,400,100,500] )
print(a3.count(100))
```

Sample Run:

```
3
```

8. index() This method returns “I”, which is the index and the smallest value of the first occurrence of x in the array.

Syntax: array.index(x)

Example

```
import array as arr
a3 = arr.array('i', [700,200,300,100,400,100,500] )
print(a3.index(100))
```

Sample Run:

```
3
```

9. pop() : This function **removes and returns the element that has an index I** of the given array. By default, it removes and returns the last element.

Syntax:

array.pop([i])

Example

```
import array as arr
a3 = arr.array('i', [100,200,300] )
print(a3.pop(0))
print(a3)
```

Sample Run:

```
100
array('i', [200, 300])
```

10. itemsize() : This function return **the length of one array element in bytes.**

Syntax:

array.itemsize()

Example

```
import array as arr
a3 = arr.array('i', [100,200,300] )
print(a3.itemsize)
```

Sample Run:

```
4
```

11. len() method : This method gives the array length.

Syntax: len(arrayname)

Example

```
import array as arr
a3 = arr.array('i', [100,200,300,400,500] )
print(len(a3))
```

Sample Run:

```
5
```

Slicing an Array

Python has a slicing feature which allows to access pieces of an array.

Basically, slicing an array is done by using a given range (eg. 2nd to 5th position). This is done by using indexes separated by a colon `[x : y]`

Example

```
numbers  = arr.array('i', 100,200,300,500,600,700,800,900,1000] )


a= numbers [3:5]
b= numbers [ :2]
c= numbers [-4:]
d= numbers [ : ]

print ( a )
print ( b )
print ( c )
print ( d )
```

Sample Run

array('i', [500, 600])
array('i', [100, 200])
array('i', [700, 800, 900, 1000])
array('i', [100, 200, 300, 500, 600, 700, 800, 900, 1000])