Arrays in Python

What we will learn?

- help() function
- Creating array
- Ways of writing import statement
- Indexing
- Slicing
- Methods in array class
    - adding/udating elements
    - Deleteing elements
    - Reversing array
    - other methods
- Variabes in array class
- Reading array from user
- Example

Array

> It is an object that stores a group of elements (or values) of same datatype. Two important points in case of arrays in python

> - Array can store only one tye of data.
> - Array can increase or decrease their size dynamically.

help()

> The **help()** method calls the built-in Python help system.

> type help() at python prompt to get the **help prompt**.

```
help()
```

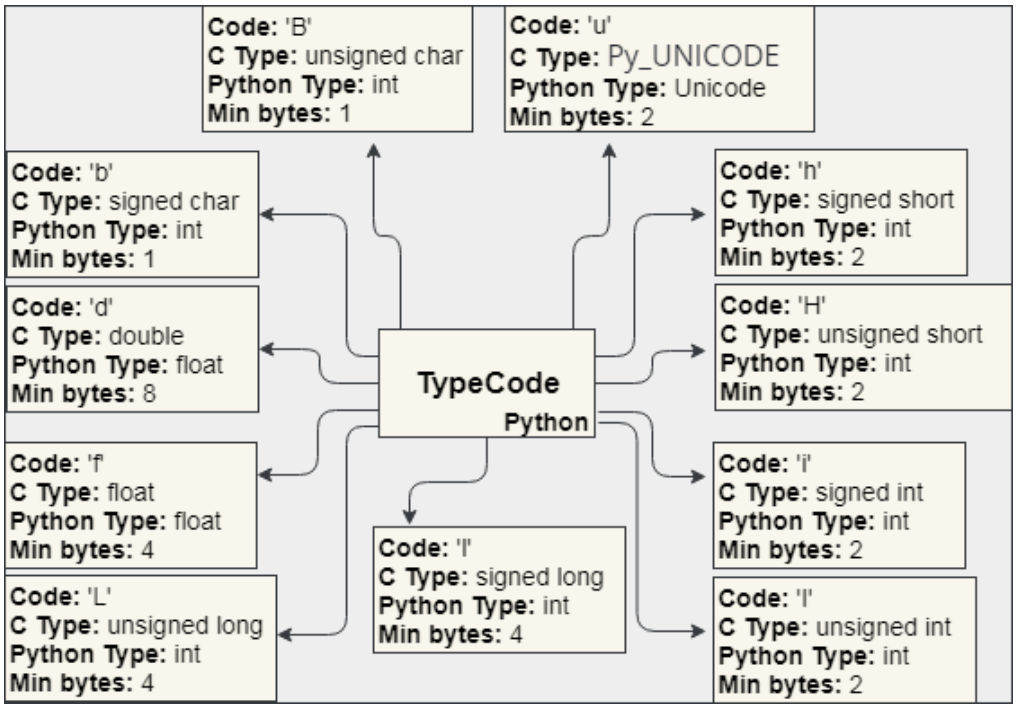
Creating an array

Syntax:

arrayname = array(type code, [elements])

- **elements** - must be written in square bracket, separated by space.

- **typecode**



Importing the Array Module

3 ways to import a module

- import array
- import array as arr
- from array import *

```
#Program to create an integer type array
import array
a = array.array('i',[1,12,14,-15,17])
print('The array elements are: ')
for i in a:
  print (i,'\t',end = '') #displaying in single line
```

```
The array elements are:
1       12      14      -15     17
```

```
#Program to create a float type array
import array as arr
a = arr.array('d',[1.5,122.3,40,-75.555,17.098])
print('The array elements are: ')
for i in a:
  print (i,'\t',end = '') #displaying in single line
```

```
The array elements are:
1.5     122.3   40.0    -75.555         17.098
```

```
#Program to create a character type array
#from array import *
from array import array
a = array('u',['a','b','c','d'])
print('The array elements are: ')
for i in a:
  print (i,'\t',end = '') #displaying in single line
```

```
The array elements are:
a       b       c       d
```

```
#program to create array from another array
from array import *
a1 = array('i',[1,2,3,4,5,6,7,8,9,10])

# using same typecode and multiplying each element by 5
a2 = array(a1.typecode, [i*5 for i in a1] )
print('The array elements in a2 are: ')
for i in a2:
  print (i,'\t',end = '') #displaying in single line
```

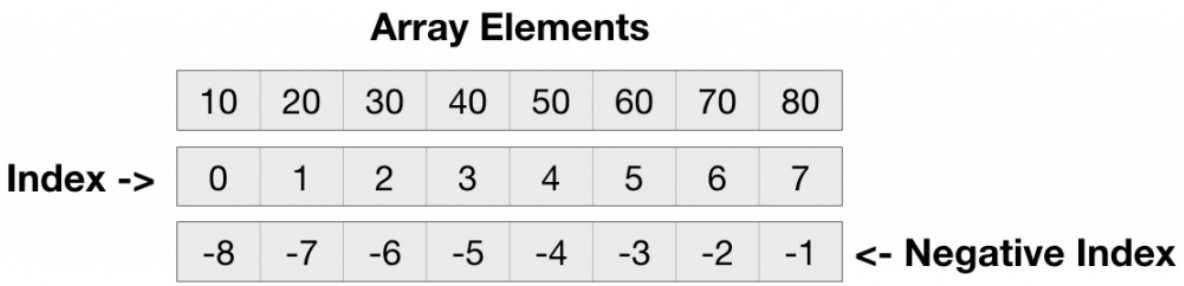
```
The array elements in a2 are:
5       10      15      20      25      30      35      40      45      50
```

Indexing

> An index represents position of the element in an array.

> The index starts from 0.

There is a support for both forward and backward indexing.



len():

To find out the number of elements in a sequence. This function returns integer value, giving the total number of element in a sequence.

```
#program to print elements of an array using array index.
from array import *

a = array('i',[100,200,300,400])

n = len(a)
print("Total number of elements in array: ", n)

#Printing array elements using indexes
print('Array elements are:',end='\t')
for i in range(n):
  print(a[i],end = '\t')
```

```
Total number of elements in array:  4
Array elements are:     100     200     300     400
```

Slicing

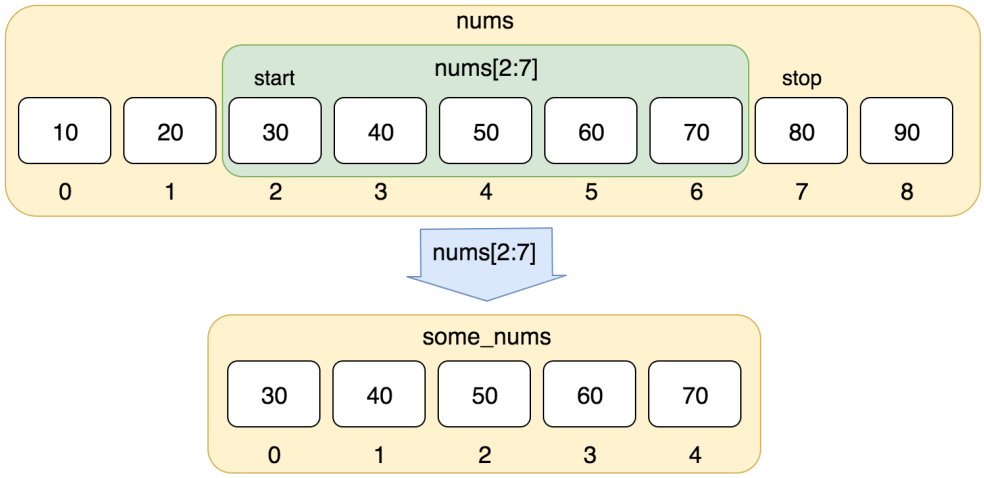
A slice represents piece of the array.

It is the creation of a new sub-array from the given array on the basis of the user-defined starting and ending indices.

Syntax: **arr[start : stop : step]**

- **start**: starting index from which we need to slice the array arr. By default set to 0.

- **stop** : ending index, before which the slicing operation would end. By default equal to the length of the array
- **step** : steps the slicing process would take from start to stop. By default set to 1.




```
#Program to understand the effect of slicing operation
a = array('i',[10,20,30,40,50,60,70,80,90,100])
print("a[0:6:1] -->", a[0:6:1])
print("a[1:9:2] -->",a[1:9:2])
print("a[:] -->",a[:])
print("a[::] -->",a[::])
print("a[0:] -->",a[0:])
print("a[5:] -->",a[5:])
print("a[:5] -->",a[:5])
print("a[::2] -->",a[::2])
print("a[-4:] -->",a[-4:])
print("a[-4:-1] -->",a[-4:-1])
print("a[-4::-1] -->",a[-4::-1])
```

```
a[0:6:1] --> array('i', [10, 20, 30, 40, 50, 60])
a[1:9:2] --> array('i', [20, 40, 60, 80])
a[:] --> array('i', [10, 20, 30, 40, 50, 60, 70, 80, 90, 100])
a[::] --> array('i', [10, 20, 30, 40, 50, 60, 70, 80, 90, 100])
a[0:] --> array('i', [10, 20, 30, 40, 50, 60, 70, 80, 90, 100])
a[5:] --> array('i', [60, 70, 80, 90, 100])
a[:5] --> array('i', [10, 20, 30, 40, 50])
a[::2] --> array('i', [10, 30, 50, 70, 90])
a[-4:] --> array('i', [70, 80, 90, 100])
a[-4:-1] --> array('i', [70, 80, 90])
a[-4::-1] --> array('i', [70, 60, 50, 40, 30, 20, 10])
```

Methods in array class

1. **append(x)**: Append a new item with value x to the end of the array.
2. **count(x)**: Return the number of occurrences of x in the array.

3. **extend(iterable)**:

> - Append items from iterable to the end of the array.
> - If iterable is another array, it must have exactly the same type code; if not, TypeError will be raised.
> - If iterable is not an array, it must be iterable and its elements must be the right type to be appended to the array.

4. **fromlist(list)**:

> - Append items from the list.
> - This is equivalent to for x in list: a.append(x) except that if there is a type error, the array is unchanged.

5. **index(x)**: Return the smallest i such that i is the index of the first occurrence of x in the array.
6. **insert(i, x)**:

> - Insert a new item with value x in the array at position i.
> - Negative values are treated as being relative to the end of the array.

7. **pop([i]):**

> - Removes the item with the index i from the array and returns it.
> - The optional argument defaults to -1, so that by default the last item is removed and returned.

8. **remove(x)**:Remove the first occurrence of x from the array.
9. **reverse()**:Reverse the order of the items in the array.
10. **tolist()**:Convert the array to an ordinary list with the same items.

```
#program to demonstrate methods of array class
from array import *
a1 = array('i',[11,22,33,55])
#Methods adding element(s) to the array
print("original array ->", a1)
a1.append(66)
print("After appending 66, array -->",a1)
a1.insert(3,44)
print("After inserting 44 at index 3, array -->",a1)
t1 = (77,88)
a1.extend(t1)
print("After extnding a tuple, array -->",a1)
l1 = [99,111]
```

```
a1.fromlist(l1)
print("After appending using fromlist, array -->",a1)
```

```
original array -> array('i', [11, 22, 33, 55])
After appending 66, array --> array('i', [11, 22, 33, 55, 66])
After inserting 44 at index 3, array --> array('i', [11, 22, 33, 44, 55, 66])
After extnding a tuple, array --> array('i', [11, 22, 33, 44, 55, 66, 77, 88])
After appending using fromlist, array --> array('i', [11, 22, 33, 44, 55, 66, 77, 88, 99
```

```
#modifying elements in array
a1[1] = 21
print("after updating element at index 1, array -->", a1)
```

```
after updating element at index 1, array --> array('i', [11, 21, 33, 44, 55, 66, 77, 88,
```

```
#Methods or statements for deleteing elements from the array
a1.pop()
print('After pop(), array ->',a1)
a1.pop(4) #removes element at index 4
print('After pop(4), array ->',a1)
a1.remove(21) #Removes first occurence of 21
print('After remove(21), array ->',a1)

'''The del keyword is used to delete objects.
   In Python everything is an object,
   so the del keyword can also be used to delete variables,
   lists, or parts of a list etc.'''
del a1[5]
print ('After del a1[5], array ->',a1)
```

```
After pop(), array -> array('i', [11, 21, 33, 44, 55, 66, 77, 88, 99])
After pop(4), array -> array('i', [11, 21, 33, 44, 66, 77, 88, 99])
After remove(21), array -> array('i', [11, 33, 44, 66, 77, 88, 99])
After del a1[5], array -> array('i', [11, 33, 44, 66, 77, 99])
```

```
#miscellaneous methods
print(a1)
print('Total no.of elemets in a1 ->',len(a1))
a1.append(77)
print(a1)
print("count of element 21 is {} and 77 is {}".format(a1.count(21),a1.count(77)))
print("position of 33 is",a1.index(33) )
```

```
array('i', [11, 33, 44, 66, 77, 99])
Total no.of elemets in a1 -> 6
array('i', [11, 33, 44, 66, 77, 99, 77])
```

```
      count of element 21 is 0 and 77 is 2
```

```
#method for reversing the array

#using slice operator
a2 = a1[::-1] # a1[-1::-1]
print("Original array:                ", a1)
print("Reverse using slice:           ", a2)
print()
#using reversed function
#The reversed() function returns the reversed iterator of the given sequence.
a3 = reversed(a1)
print("Original array:                ", a1)
print("Reverse using reversed function:", a2)
print()
#using reverse() method
a1.reverse()
print("Reverse using reverse method:   ", a1)
```

```
     Original array:                 array('i', [11, 33, 44, 66, 77, 99, 77])
     Reverse using slice:            array('i', [77, 99, 77, 66, 44, 33, 11])

     Original array:                 array('i', [11, 33, 44, 66, 77, 99, 77])
     Reverse using reversed function: array('i', [77, 99, 77, 66, 44, 33, 11])

     Reverse using reverse method:   array('i', [77, 99, 77, 66, 44, 33, 11])
```

Variables (data members) in array class

1. **typecode** : represents type code character used to create the array.
2. **itemsize** : size of item stored in array (in bytes)

```
print("Typecode of a1 is", a1.typecode)
print("Size of items in a1 is", a1.itemsize)
print("Total size of a1 in bytes is", len(a1)*a1.itemsize)
```

```
     Typecode of a1 is i
     Size of items in a1 is 4
     Total size of a1 in bytes is 28
```

Reading array from user

```
from array import *
#using list comprehension
inp = [int(i) for i in input('Enter array element seperated by space: ').split()]
a4 = array('i',[])
```

```
a4 = array('i',[])
a4.fromlist(inp)
print(a4)
```

```
     Enter array element seperated by space: 1 2 3 4 5
     array('i', [1, 2, 3, 4, 5])
```

```
#reading one element at a time
a5 = array('i',[])

n = int(input('Enter no. of elements: '))

for i in range(n):
  x = int(input('Enter element a5[{}] : '.format(i)))
  a5.append(x)

print(a5)
```

```
     Enter no. of elements: 4
     Enter element a5[0] : 11
     Enter element a5[1] : 22
     Enter element a5[2] : 33
     Enter element a5[3] : 44
     array('i', [11, 22, 33, 44])
```

```
#python prgram to find first repeated element in an array
'''
        Array elements are: 10 20 30 20 40
        20 repeated @ 3 index
'''
from array import *
a6 = array('i',[])

n = int(input('Enter no. of elements: '))
for i in range(n):
  x = int(input('Enter element a5[{}] : '.format(i)))
  a6.append(x)

print("Array is:",a6)
pos = -1
for i in range(n):
  for j in range(i+1,n):
    if a6[i] == a6[j]:
      ele = a6[j]
      pos = j
      break
  if pos != -1:
    break

if pos == -1:
  print("There is no repeated element")
else:
```

```
else:
  print("%d repeated @ %d index\n"%(ele,pos))
```

```
    Enter no. of elements: 5
    Enter element a5[0] : 77
    Enter element a5[1] : 22
    Enter element a5[2] : 11
    Enter element a5[3] : 77
    Enter element a5[4] : 33
    Array is: array('i', [77, 22, 11, 77, 33])
    77 repeated @ 3 index
```

```
2#python prgram to find first repeated element in an array
'''
        Array elements are: 10 20 30 20 40
        20 repeated @ 3 index
'''
from array import *
a6 = array('i',[])

n = int(input('Enter no. of elements: '))
for i in range(n):
  x = int(input('Enter element a5[{}] : '.format(i)))
  a6.append(x)

print("Array is:",a6)
pos = -1
for i in range(n):
  if a6.count(a6[i]) > 1:
      ele = a6[i]
      pos = a6[i+1::].index(a6[i])+(i+1)
      break

  if pos != -1:
    break

if pos == -1:
  print("There is no repeated element")
else:
  print("%d repeated @ %d index\n"%(ele,pos))
```

```
    Enter no. of elements: 5
    Enter element a5[0] : 77
    Enter element a5[1] : 22
    Enter element a5[2] : 11
    Enter element a5[3] : 77
    Enter element a5[4] : 33
    Array is: array('i', [77, 22, 11, 77, 33])
```

```
77 repeated @ 3 index
```

Note:

1. With array class, there is no support for array of strings. This can be achieved using list in python
2. Array class has no support for multidimensional array. we can use multidimensional array using third party library like numpy. Other option can be using nested list.