

ARRAYS IN PYTHON

- Notes By Jagdish Chavan
- For Code and Projects follow on [Github \(https://github.com/JagdishChavan081\)](https://github.com/JagdishChavan081) | [Linkedin \(https://www.linkedin.com/in/jagdishchavan\)](https://www.linkedin.com/in/jagdishchavan)

What Is an Array?

- An array is an object that stores a group of elements(or values) of same datatype.
- it Stores and process a group of elements easily.
- Note Arrays can store only one type of data. Arrays can increase or decrease their size dynamically
- In Python there is a standard module by the name 'array' that helps us to create and use array.

Advantages of Arrays

- Array are similar to list main difference is array is homogenous in nature which plays as upper hand role when dealing with huge number of elements as it uses less memory than list and they offer faster execution than list .
- Size of an array is not fixed in Python, depends on our specifications how many elements we want to store in the beginning.
- Array can grow and shrink memory dynamically.
- many methods are available in array module itself for processing of data.

Creating an Array

- **syntax:-** `arrayname = array(type code, [elements])`
- type code is nothing but type of elements we are going to store in array, where elements is just data that we want to store.

Table 1: The type code to create arrays

Type code	Ctype	Minimum size in bytes
'b'	signed integer	1
'B'	unsigned integer	1
'i'	signed integer	2
'I'	unsigned integer	2
'l'	signed integer	4
'L'	unsigned integer	4
'f'	floating point	4
'd'	double precision floating point	8
'u'	unicode character	2

In [45]: *# Write a Python program to create an integer type array and check its type*

```
from array import *  
a = array('i',[1,2,3,4])  
print(type(a))  
  
for element in a:  
    print(element,end='')
```

```
<class 'array.array'>  
1234
```

In [49]: *# Write a Python program to create an integer type array and check its type*

```
import array  
  
a = array.array('i',[1,2,3,4])  
print(type(a))  
  
for element in a:  
    print(element,end='')
```

```
<class 'array.array'>  
1234
```

In [53]: *# write a python program to create an array with group of characters.*

```
from array import *  
  
arr = array('u',['a','b','c','d','e'])  
  
for element in arr:  
    print('array element {} is of type {}'.format(element, type(element)))
```

```
array element a is of type <class 'str'>  
array element b is of type <class 'str'>  
array element c is of type <class 'str'>  
array element d is of type <class 'str'>  
array element e is of type <class 'str'>
```

Creating one Array from another Array

syntax:- `arr2 = array(arr1.typecode,(a for a in arr1))`

- `arr1.typecode` gives typecode character of arr1

In [57]: *# write a Python program to create one array from another array.*

```
from array import *

arr1 = array('d', [3.14, 0.96, 3.24, 4])

arr2 = array(arr1.typecode, (a*3 for a in arr1))
for i in arr2:
    print(i, end=',')

9.42,2.88,9.72,12.0,
```

Indexing and Slicing on Arrays

In [61]: *# Writew a Python program to retrive the elements of an array using array inde x.*

```
from array import *

x = array('i', [10, 20, 30, 40, 50])

#find number of elements in the array
n = len(x)

#display array elements using ineking
for i in range(n):
    print(x[i], end=',')

10,20,30,40,50,
```

In [65]: *#Write a Python program to retrive elements of an array using while loop*

```
from array import *

x = array('i', [10, 20, 30, 40, 50])
n = len(x)

#display array element using indexing
i=0
while i<n:
    print(x[i], end=',')
    i+=1

10,20,30,40,50,
```

In [69]: *#write a python program that helps to know the effects of slicing operations on an array*

```
from array import *

x = array('i', (10,20,30,40,50))

y=x[1:4]
print('y=',y)

#create array y with elements from 0th till the last element in x
z = x[0:]
print('z=',z)

#create an array with last 4 elements
a=x[-4:]
print('a=',a)

y= array('i', [20, 30, 40])
z= array('i', [10, 20, 30, 40, 50])
a= array('i', [20, 30, 40, 50])
```

Processing the array

- arrays class of arrays module offers methods to process the array easily.
- **syntax:- objectname.method()**

a.append(x): Adds an element x at the end of the existing array a.

```
In [ ]: from array import *

a = array('i', [10,20,30,40,50])
print('original array a = ',a)

#append element in a
a.append(30)
a.append(4)
print('\nafter appending a = :',a)
```

a.insert(i,x): Insert x in the position i in the array.

In [38]: `from array import *`

```
a = array('i',[10,20,30,40,50])
print('original array a = ',a)
```

```
#insert element in a
a.insert(1,99)
```

```
print('\nafter inserting a = :',a)
```

original array a = `array('i', [10, 20, 30, 40, 50])`

after inserting a = : `array('i', [10, 99, 20, 30, 40, 50])`

a.count(): Returns the number of occurrence of x in a

In [41]: `from array import *`

```
a = array('i',[10,20,30,40,50,10,20,30,55])
print('original array a = ',a)
```

```
#count element
n = a.count(10)
```

```
print('\nafter counting 10 in a = :',n)
```

original array a = `array('i', [10, 20, 30, 40, 50, 10, 20, 30, 55])`

after counting 10 in a = : 2

a.extend(x): Appends x at the end of the array a.'x' can be another array or an iterable object.

In [44]: `from array import *`

```
a = array('i',[10,20,30,40,50])
x= [1,2,3]
```

```
a.extend(x)
print(a)
```

`array('i', [10, 20, 30, 40, 50, 1, 2, 3])`

a.remove(x): Removes the first occurrence of x in the array a. Raises value error if not found.

In [48]: `from array import *`

```
a = array('i',[10,20,30,40,50])
print('original array a = ',a)

#remove element in a
a.remove(20)

print('\nafter removing a = :',a)
```

original array a = `array('i', [10, 20, 30, 40, 50])`

after removing a = : `array('i', [10, 30, 40, 50])`

a.pop(): remove last element using pop() method.

In [52]: `from array import *`

```
a = array('i',[10,20,30,40,50])
print('original array a = ',a)

#remove the last element in a
a.pop()

print('\nafter popping a = :',a)
```

original array a = `array('i', [10, 20, 30, 40, 50])`

after popping a = : `array('i', [10, 20, 30, 40])`

a.pop(x): based on index position element get popped.

In [56]: `from array import *`

```
a = array('i',[10,20,30,40,50])
print('original array a = ',a)

#remove the last element in a
a.pop(2)

print('\nafter popping a = :',a)
```

original array a = `array('i', [10, 20, 30, 40, 50])`

after popping a = : `array('i', [10, 20, 40, 50])`

a.index(): finding position of element using index() method.

```
In [60]: from array import *

a = array('i',[10,20,30,40,50,30])
print('original array a = ',a)

#finding position of element
n = a.index(30)
print('first occurance of 30 is at index position ',n)

original array a = array('i', [10, 20, 30, 40, 50, 30])
first occurance of 30 is at index position 2
```

a.tolist(): Converts the array 'a' into a list.

```
In [64]: from array import *

a = array('i',[10,20,30,40,50,30])
print('original array a = ',a)

#converting array to list
lst = a.tolist()
print(lst)
print(type(lst))

original array a = array('i', [10, 20, 30, 40, 50, 30])
[10, 20, 30, 40, 50, 30]
<class 'list'>
```

a.reverse: Reverse the order of elements in the array a.

```
In [68]: from array import *

#The original array
arr = [11, 22, 33, 44, 55]
a = array('i',arr)

a.reverse() #reversing using reverse()
print("After reversing Array:",a)
print(type(a))

After reversing Array: array('i', [55, 44, 33, 22, 11])
<class 'array.array'>
```

a.fromlist(lst): Append item form the lst to the end of the array.'lst' can be any list or iterable object

```
In [ ]: from array import *

lst = [1,2,3,5,8]

a = array('i',[10,20,30,40,50,30])

a.fromlist(lst)
print(a)
```

```
In [67]: import array
f = open("f.bin", "wb")
array.array("i", [1, 2, 3]).tofile(f)
f.close()

ints = array.array("i")
f = open("f.bin", "rb")
ints.fromfile(f, 3)
print (ints)

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

```
In [ ]: # Write a Python Program to storing students marks into an array and finding t
otal marks and percentage of marks

from array import *

str = input('Enter marks:').split(' ')

#store marks into mark array
marks = [int(num) for num in str]

#display the marks and find total
sum = 0
for x in marks:
    print(x)
    sum+=x
print('total marks:',sum)

#display th epercentage
n = len(marks)
percentage = sum/n
print('percentage:',percentage)
```



```

In [46]: # write a python program to sort the array elements using bubble sort technique.

from array import *

x = array('i',[])

#storing the element
print('how many elements?', end='')
n= int(input())

for i in range(n):
    print('\nenter the element',end='')
    x.append(int(input()))

print('\norigianl array:',x)

#Bubble sort
flag = False
for i in range(n-1):
    for j in range(n-1-i):
        if x[j]>x[j+1]:
            t=x[j]
            x[j]=x[j+1]
            x[j+1]=t
            flag = True
    if flag ==False:
        break
    else:
        flag=False
print('sorted array = ',x)

how many elements?
enter the element
origianl array: array('i', [2])
sorted array = array('i', [2])

```

In [50]: *# Write a Python program to search for the position of an element in an array using sequential search.*

```
from array import *

#create a empty array
x = array('i',[])

#store the lement into array
print('How many element?', end = '')
n = int(input())

for i in range(n):
    print('\nEnter element',end = '')
    x.append(int(input()))

print('\nOrigianl array',x)

print('\nEnter the element to search',end='')
s = int(input())
flah = False

for i in range(len(x)):
    if s==x[i]:
        print('\nfound at position ',i+1)
        flag=True
if flag == False:
    print('Not found in array')
```

How many element?

Enter element

Enter element

Enter element

Origianl array array('i', [4, 5, 2])

Enter the element to searchNot found in array

In [54]: *# Write a Python Program to search for the position of an element in an array using index() method*

```
from array import *

x = array('i',[])

print("How many element?", end="")
n = int(input())

for i in range(n):
    print("\nEnter Element:", end='')
    x.append(int(input()))

print("\noriginal array:",x)
print('\nEnter element to search', end="")
s = int(input())

# index() method
try:
    pos = x.index(s)
    print('\nfound at position=',pos+1)

except ValueError:
    print('\nnot found in array')
```

```
How many element?
Enter Element:
Enter Element:
original array: array('i', [3, 4])

Enter element to search
not found in array
```

Hope you liked the content till now please dont forget to hit a star on [Github](https://github.com/JagdishChavan081) (<https://github.com/JagdishChavan081>).

Types of Arrays

- **Single dimension array:** represents only one row or one column of element.
- **Multi-dimensional arrays:** represents more than one row and column of element
- In Python we can create and work with single dimensional arrays only, Python does not support multi-dimensional arrays.
- we can construct multidimensional arrays using packages like numpy, following session are devoted for numpy only.

```
In [58]: # Write a python program to create a simple array using numpy.
```

```
import numpy

a = numpy.array([10,20,30,40])
print(type(a))
print(a)
```

```
<class 'numpy.ndarray'>
[10 20 30 40]
```

```
In [62]: #aliasing numpy as np
```

```
import numpy as np

a = numpy.array([10,20,30,40])
print(type(a))
print(a)
```

```
<class 'numpy.ndarray'>
[10 20 30 40]
```

```
In [66]: # another way
```

```
from numpy import *

a = array([10,20,30,40,50])
print(type(a))
a
```

```
<class 'numpy.ndarray'>
```

```
Out[66]: array([10, 20, 30, 40, 50])
```

** Creating array in numpy can be done in several ways. some important are as follow.

- Using array() function.
- Using linspace() function.
- Using Logspace() function.
- using arange() function.
- using zeros() and ones() functions

Creating Arrays using array()

- syntax: a = array([10,20,30,40,50])

```
In [70]: # Write a python program to create a character type array with a group of characters.
```

```
from numpy import *  
a = array(['a','b','c','d'], dtype = str)  
print(a)
```

```
-----  
TypeError                                Traceback (most recent call last)  
<ipython-input-70-950f6b7dbf79> in <module>  
      2  
      3 from numpy import *  
----> 4 a = array(['a','b','c','d'], dtype = str)  
      5 print(a)
```

TypeError: Field elements must be 2- or 3-tuples, got ''5''

```
In [15]: # Write a python program to create an array from another array.
```

```
from numpy import *  
a = array([1,2,3,4,5])  
b = array(a)  
c = a
```

```
#display array  
print(a)  
print(b)  
print(c)
```

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

Creating array using linspace

- **syntax** `a = linspace(start, stop,n)`
- Used to create an array with evenly spaced points between a starting point and ending point.
- **start**: starting point of array.
- **stop**: represents ending point.
- **n**: number of parts element should be divided, if n is omitted, then it is taken as 50.

```
In [18]: # write a Python Program to creating an array with 5 equal points using linspace()
```

```
from numpy import *  
  
# divide 0 to 10 into 5 parts  
a = linspace(0,10,5)  
a
```

```
Out[18]: array([ 0. ,  2.5,  5. ,  7.5, 10. ])
```

Creating array using logspace()

- logspace() produces evenly spaced points on a logarithmically spaced scale.
- **syntax:** `a = logspace(start, stop, n)`
- start: starting point of array.
- stop: represents ending point.
- n: number of parts element should be divided, if n is omitted, then it is taken as 50
- The function represents value starting from 10^1 to 10^4 .

In [20]: *# Write a python program to create an array using logspace()*

```
from numpy import *  
  
a= logspace(1,4,5)  
  
#find no of elements in a  
n =len(a)  
  
#repeat from 0 to n-1 times  
for i in range(n):  
    print('%.1f' %a[i],end=', ')  
  
10.0,56.2,316.2,1778.3,10000.0,
```

Creating array using arange().

- same as range function in python.
- **syntax:** `a = arange(start, stop, stepsize)`

In [21]: *# Write a Python program to create an array with even number up to 10.*

```
from numpy import *  
  
#create an array using arange() function  
a = arange(2,11,2)  
print(a)  
  
[ 2  4  6  8 10]
```

Creating arrays using zeros() and ones() functions.

- **zeros()** function is use to create an array with all zeros.
- **syntax:** `zeros(n, datatype)`
- **ones()** function is use to create an array with all 1s.
- **syntax:** `ones(n, datatype)`

In [22]: *# write a python program to create an numpy array with zeros and ones.*

```
from numpy import *  
a = zeros(5, int)  
print(a)  
  
b = ones(5, int)  
print(b)
```

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

Mathematical Operations on Arrays

In [7]: *# Write a Python Program to perform some mathematical operations on a numpy.*

```
from numpy import *  
  
# create a numpy array using array() function  
a = array([10,20,30,40,50])  
print('\noriginal array',a)  
  
# arithmetic operations on array  
print('\n After adding 5:',a+5)  
print('\n after subtracting 5',a-5)  
print('\n after multiplying:',a*5)  
print('\n after division:',a/5)  
print('\n after modulus with 5',a%5)
```

original array [10 20 30 40 50]

After adding 5: [15 25 35 45 55]

after subtracting 5 [5 15 25 35 45]

after multiplying: [50 100 150 200 250]

after division: [2. 4. 6. 8. 10.]

after modulus with 5 [0 0 0 0 0]

In [6]: *# Write a Python program to create a expression using numpy array*

```
from numpy import *  
  
a = array([10,20,30,40,50])  
  
print("Expression value:",(a+5)**2-10)
```

Expression value: [215 615 1215 2015 3015]

Mathematical Function in Numpy

- **sin(arr)**:calculates the sin value of each element in the array

In [8]: `from numpy import *`

```
a = array([10,20,30,40,50])  
print('\n sin value of array',sin(a))
```

sin value of array [-0.54402111 0.91294525 -0.98803162 0.74511316 -0.26237485]

- **cos(a)**:calculates cosine value of each element of the array

In [9]: `from numpy import *`

```
a = array([10,20,30,40,50])  
print('\n cosine value of array',cos(a))
```

cosine value of array [-0.83907153 0.40808206 0.15425145 -0.66693806 0.96496603]

- **tan(a)**:Calculates tangent value of each element in the array.

In [10]: `from numpy import *`

```
a = array([10,20,30,40,50])  
print('\n tan value of array',tan(a))
```

tan value of array [0.64836083 2.23716094 -6.4053312 -1.11721493 -0.27190061]

- Hope you like the work till here for more info on numpy array please do visit [Github](https://github.com/JagdishChavan081) (<https://github.com/JagdishChavan081>).

In []: