

```
# In Python, array is a collection of items stored at contiguous memory locations.
# The idea is to store multiple items of the same type together.
# Unlike Python lists (can store elements of mixed types), arrays must have all elements of same type.
# Having only homogeneous elements makes it memory-efficient.
```

```
# Syntax
# The syntax for creating an array in
# Python is -
# importing
# import array as array_name

# creating array
# obj = array_name.array(typecode[, initializer])
# Where,
# typecode - The typecode character used to specify the type of elements in the array.
# initializer - It is an optional value from which array is initialized. It must be a list, a bytes-like object, or iterable el
```

```
# Array in Python can be created by importing an array module.
# array( data_type , value_list ) is used to create array in Python with data type and value list specified in its arguments.
```

```
import array as arr
```

```
# typecode Python data type Byte size
# 'b' signed integer 1
# 'B' unsigned integer 1
# 'u' Unicode character 2
# 'h' signed integer 2
# 'H' unsigned integer 2
# 'i' signed integer 2
# 'I' unsigned integer 2
# 'l' signed integer 4
# 'L' unsigned integer 4
# 'q' signed integer 8
# 'Q' unsigned integer 8
# 'f' floating point 4
# 'd' floating point 8
```

```
# creating array of integers
a1 = arr.array('i',[1,2,3,4,5])
```

```
for i in a1:
    print(i)
```

```
1
2
3
4
5
```

```
# creating array of float
a1 = arr.array('d',[1,2,3.7,4,5])
```

```
for i in a1:
    print(i)
```

```
1.0
2.0
3.7
4.0
5.0
```

```
# creating array of char
a1 = arr.array('u','Govind')
```

```
for i in a1:
    print(i)
```

```
G
o
v
i
n
```

d

```
# Basic Operations on Python Arrays
# Following are the basic operations supported by an array -

# Traverse - Print all the array elements one by one.

# Insertion - Adds an element at the given index.

# Deletion - Deletes an element at the given index.

# Search - Searches an element using the given index or by the value.

# Update - Updates an element at the given index.
```

```
# accessing array elements
```

```
a1 = arr.array('i',[1,2,3,4,5])

print(a1[0]) # first element
print(a1[3]) # fourth element
```

```
1
4
```

```
# inserting element in array
```

```
a1 = arr.array('i',[1,2,3,4,5])

a1.insert(2,70)

for i in a1:
    print(i)
```

```
1
2
70
3
4
5
```

```
# deleting an array element
```

```
# remove()
a1 = arr.array('i',[1,2,3,1,4,5])

a1.remove(1)#removes first occurrence of 1

for i in a1:
    print(i)

# pop()
a1 = arr.array('i',[1,2,3,1,4,5])

a1.pop(1)#removes element at index 1

for i in a1:
    print(i)
```

```
2
3
1
4
5
1
3
1
4
5
```

```
# updating array element
```

```
a1 = arr.array('i',[1,2,3,1,4,5])
a1[3] = 3

for i in a1:
    print(i)
```

```
1
2
3
3
4
5
```

```
# getting length of array

a1 = arr.array('i',[1,2,3,1,4,5])

print(len(a1))
```

```
6
```

```
# The extend() function is simply used to attach an item from iterable to the end of the array.

a = arr.array('i', [1, 2, 3,4,5])

# using extend() method
a.extend([6,7,8,9,10])
print(a)
print(type(a))
```

```
array('i', [1, 2, 3, 4, 5, 6, 7, 8, 9, 10])
<class 'array.array'>
```

```
# reversing an array

a= arr.array('i',[1,2,3,4,5])

a.reverse()

for i in a:
    print(i)
```

```
5
4
3
2
1
```

```
# counting number of occurrences of an element

a = arr.array('i', [1, 2, 3, 4, 2, 5, 2])
count = a.count(2)

print("Number of occurrences of 2:", count)
```

```
Number of occurrences of 2: 3
```

```
# In order to search an element in the array we use a python in-built index() method.
# This function returns the index of the first occurrence of value mentioned in arguments.

a = arr.array('i', [1, 2, 3, 1, 2, 5])

# index of 1st occurrence of 2
print(a.index(2))

# index of 1st occurrence of 1
print(a.index(1))
```

```
1
0
```

```
# slicing in array

l = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]

a = arr.array('i', l)

Sliced_array = a[3:8]
print(Sliced_array)
```

```
Sliced array = [5, 6, 7, 8, 9, 10]
```

```
Sliced_array = a[5:]  
print(Sliced_array)
```

```
Sliced_array = a[::-1]  
print(Sliced_array)
```

```
array('i', [4, 5, 6, 7, 8])  
array('i', [6, 7, 8, 9, 10])  
array('i', [10, 9, 8, 7, 6, 5, 4, 3, 2, 1])
```