Introduction to Data Structures

Anoop S Babu
Faculty Associate
Dept. of Computer Science & Engineering
bsanoop@am.amrita.edu

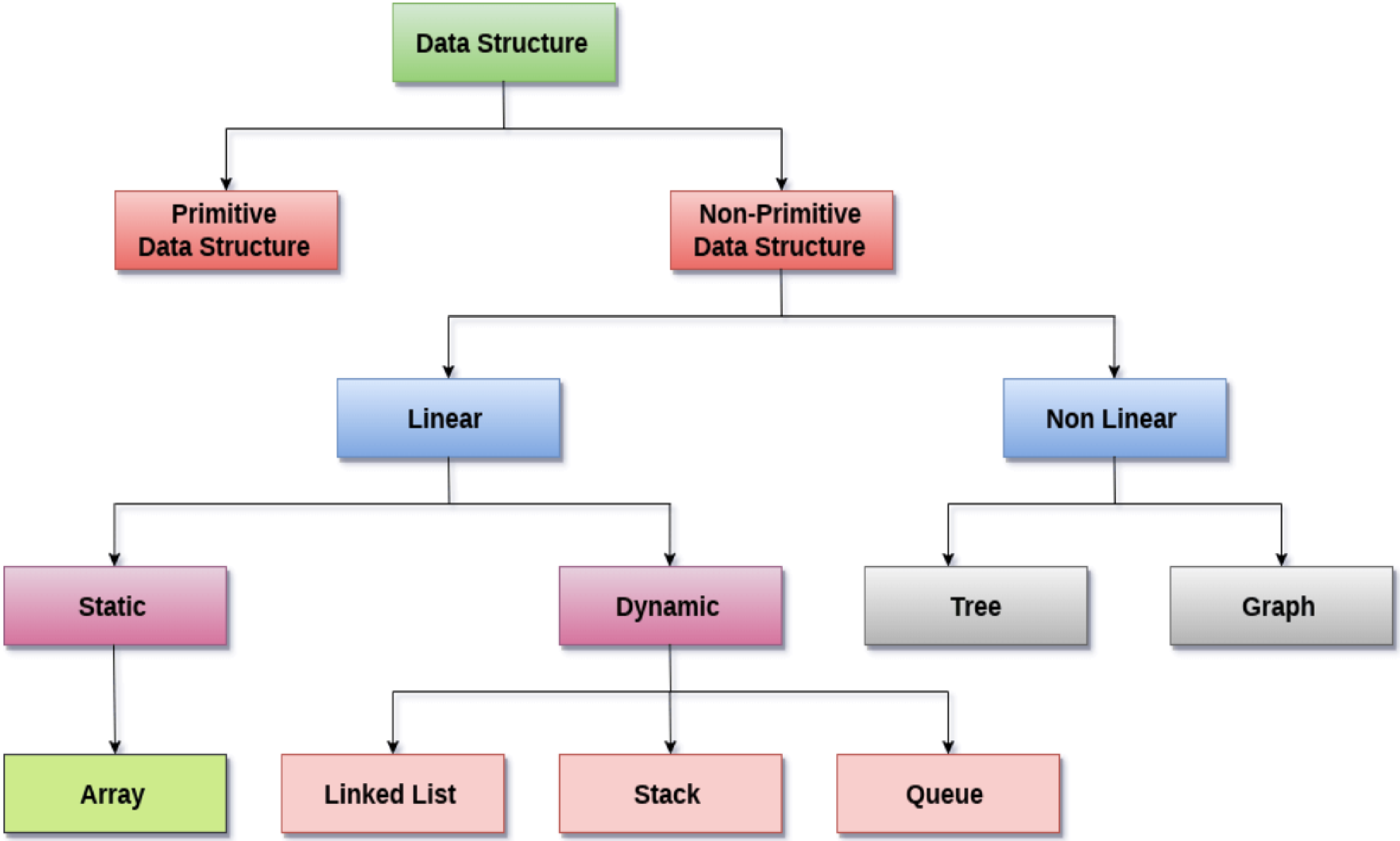
Introduction

- Data Structure is the **way to organize** and store **data** in the memory.
- Primitive v/s Non-primitive Data structures
  - Primitive data Structures are the **basic data structures** that directly operate upon the machine instructions.
    - Integers, Float
  - Non-primitive data structures are more **complicated data structures and are derived from primitive data structures.**
    - Arrays, Linked Lists, Stack, Tree

Introduction

- Liner v/s Non-Linear Data Structures
  - A linear data structure has all the **data elements arranged in sequential manner** and each member element is connected to its previous and next element.
    - Array, Linked list, Stack, Queue
  - Non liner data structures where **data elements are not arranged specifically in sequence** are non-linear data structures.
    - `Trees, Graphs

Classification of Data Structure

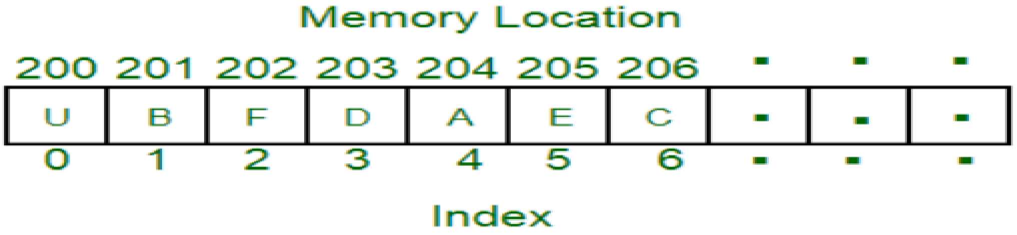


Operations on Data Structure

- **Traversing:** **visiting each element of the data structure** in order to perform some specific operation like searching or sorting.
- **Insertion:** the **process of adding the elements** to the data structure at any location.
- **Deletion:** process of **removing an element** from the data structure at any location.
- **Searching:** process of **finding the location of an element** within the data structure.
- **Sorting:** process of **arranging the data structure in a specific order.**
- **Merging:** clubbing/joining two similar data structures into one.

Arrays

- An array is a collection of **items stored at contiguous memory locations**.

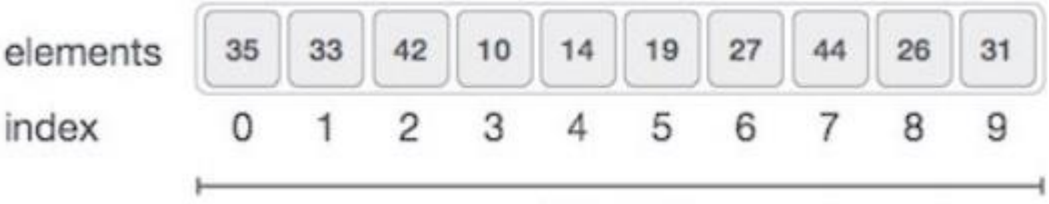


Memory Location

200	201	202	203	204	205	206	.	.	.
U	B	F	D	A	E	C	.	.	.
0	1	2	3	4	5	6	.	.	.

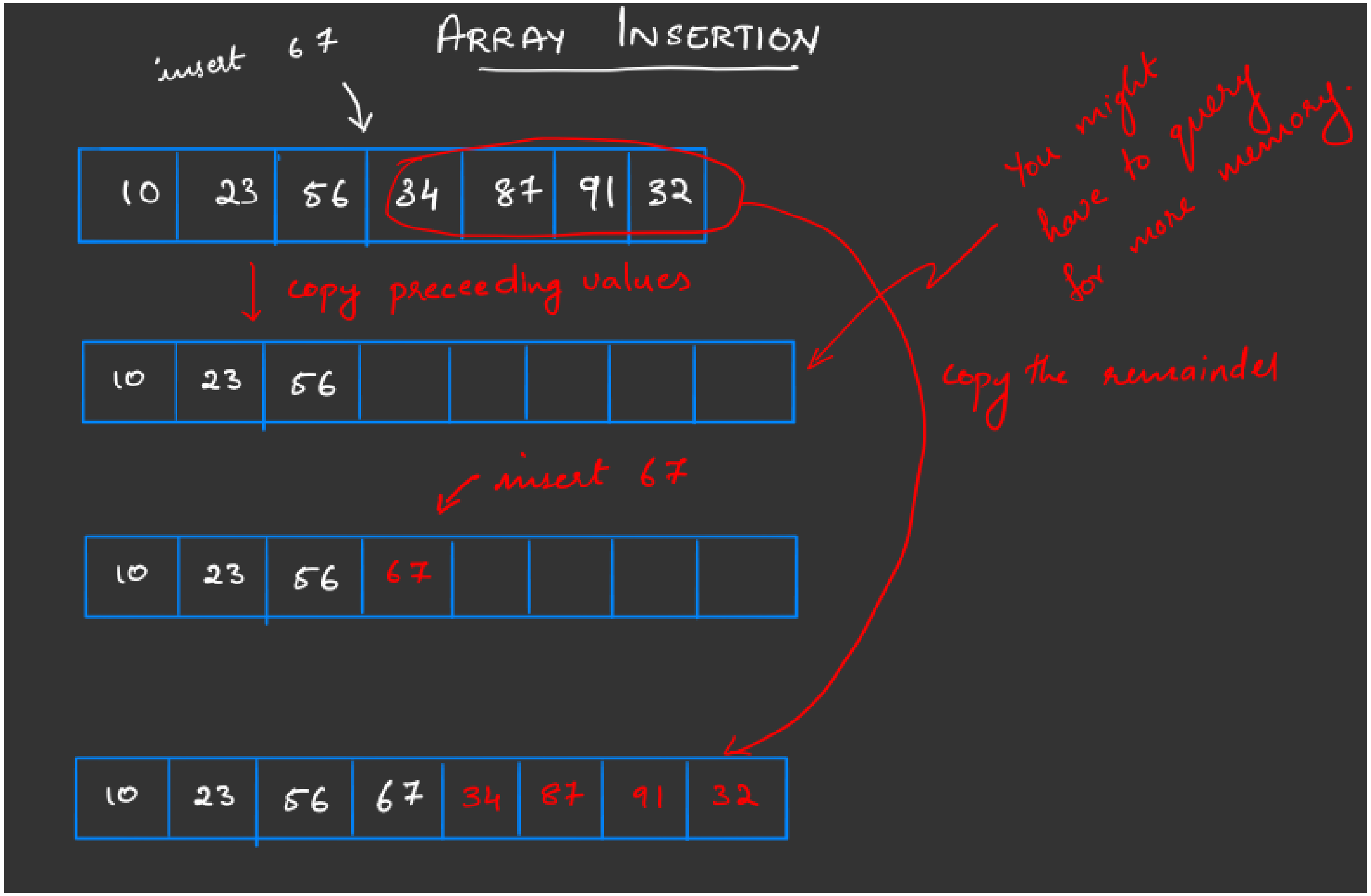
Index

- **Elements:** Each item stored in an array.
- **Index:** Each location of an element in an array has a numerical index.

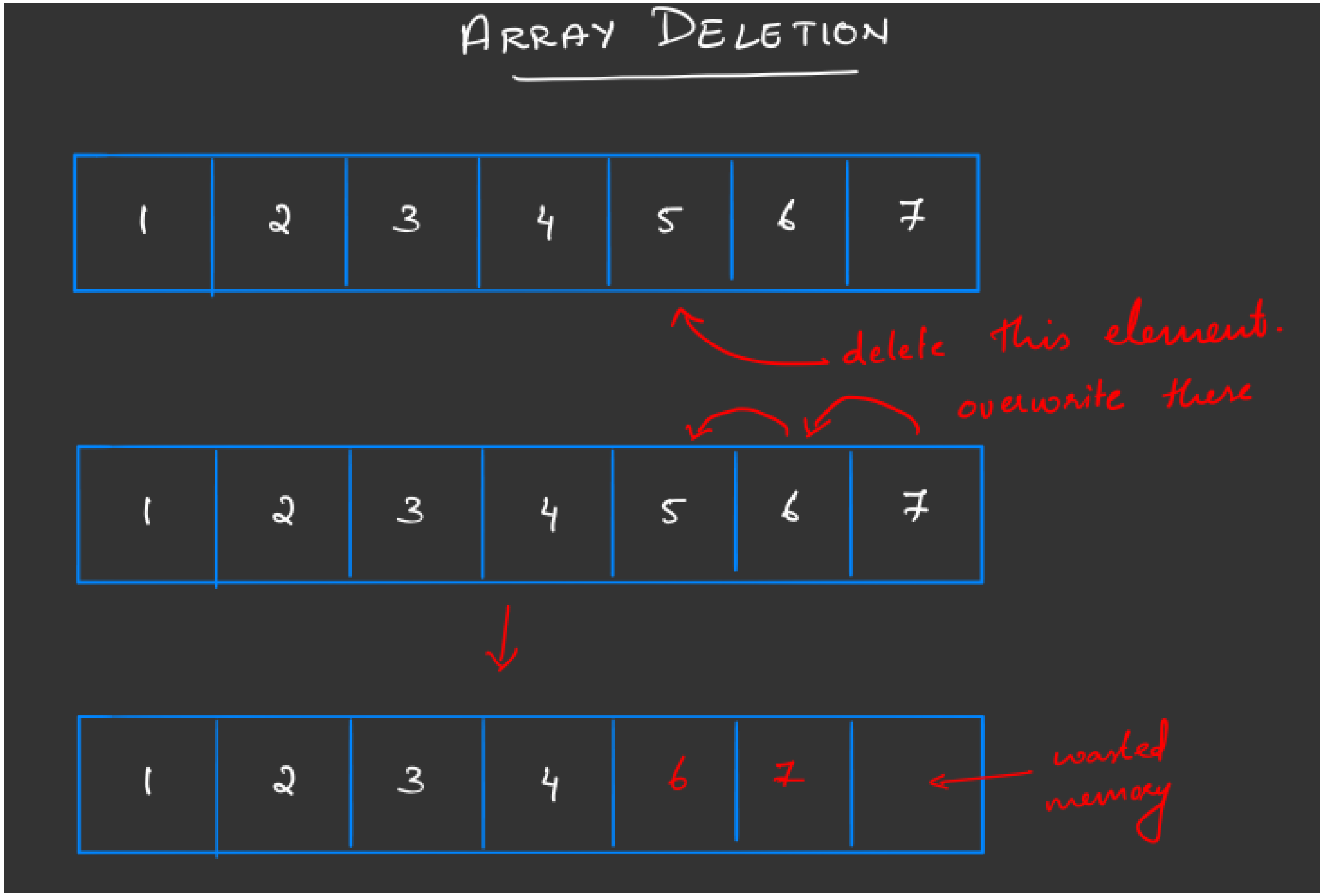


elements	35	33	42	10	14	19	27	44	26	31
index	0	1	2	3	4	5	6	7	8	9

Array: Insertion Operation

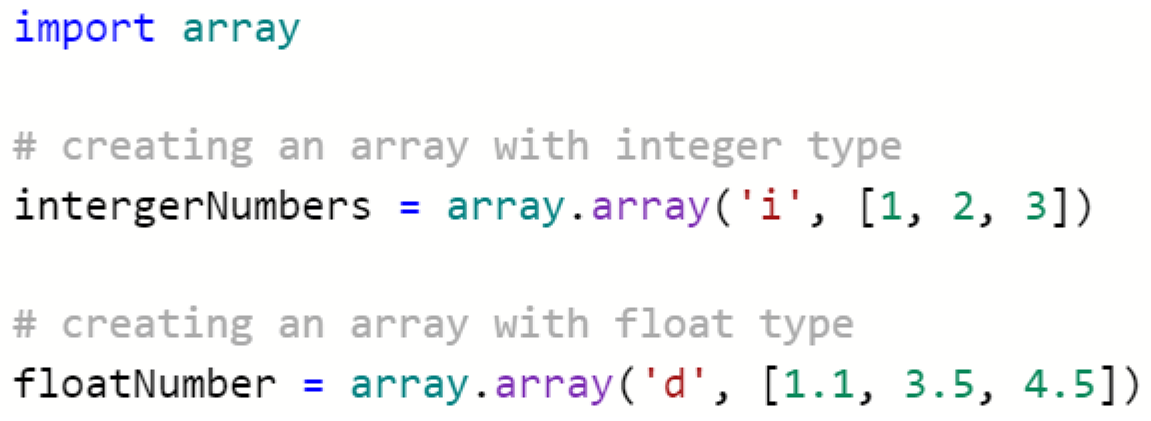


Array: Deletion Operation



Creating Python Arrays

- To create an array of numeric values, need to import the array.

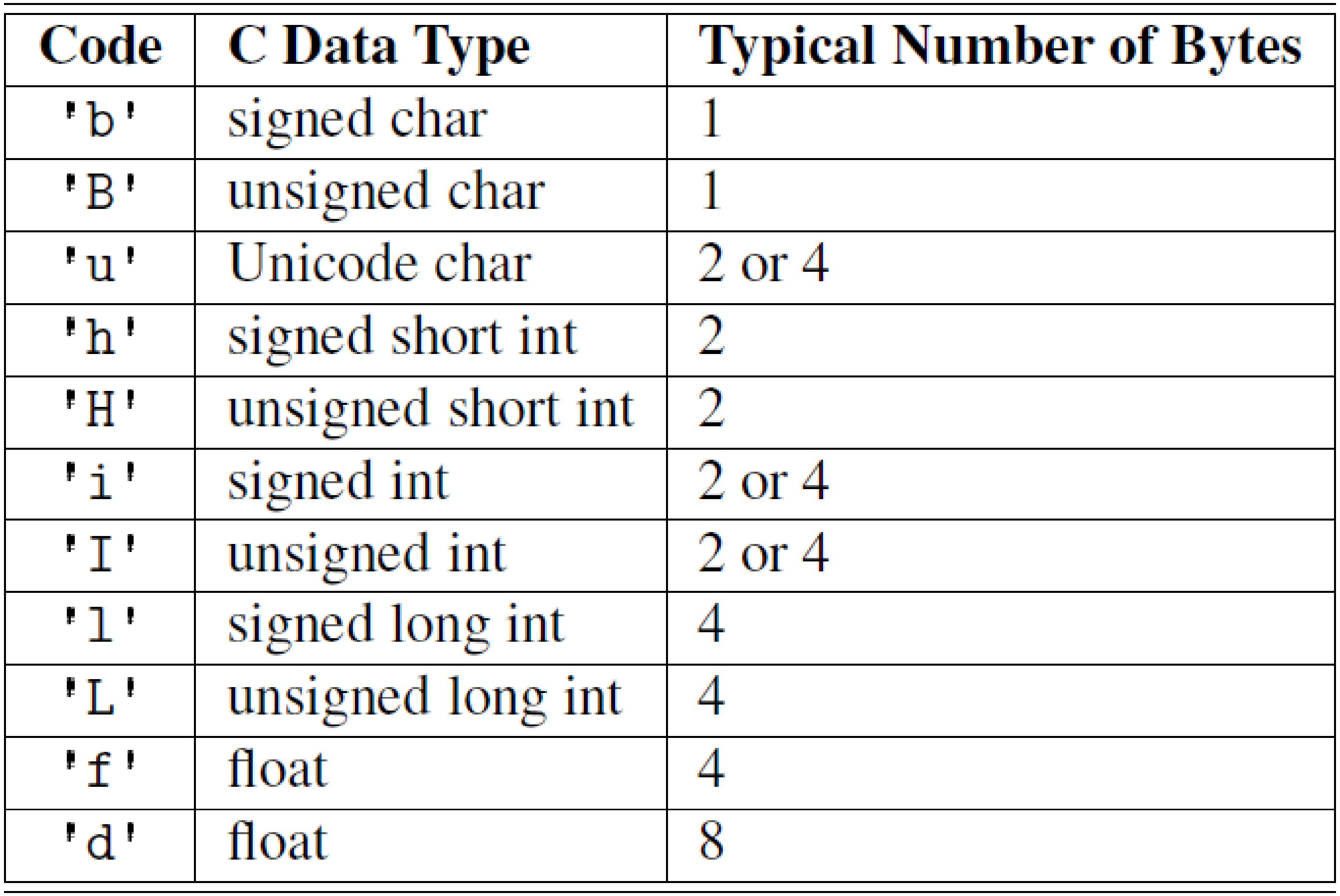


```
import array

# creating an array with integer type
intergerNumbers = array.array('i', [1, 2, 3])

# creating an array with float type
floatNumber = array.array('d', [1.1, 3.5, 4.5])
```

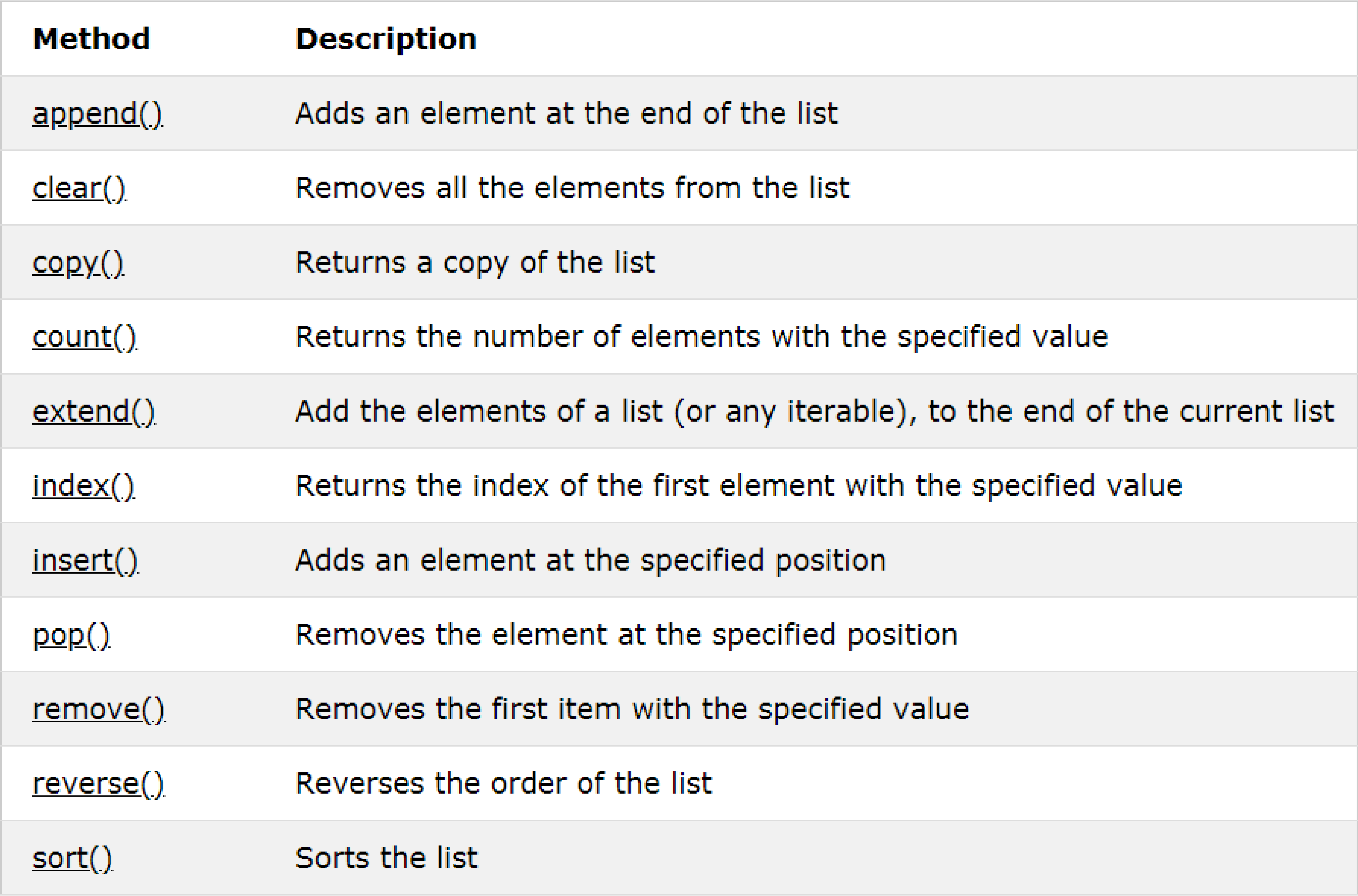
Creating Python Arrays



Code	C Data Type	Typical Number of Bytes
`'b'`	signed char	1
`'B'`	unsigned char	1
`'u'`	Unicode char	2 or 4
`'h'`	signed short int	2
`'H'`	unsigned short int	2
`'i'`	signed int	2 or 4
`'I'`	unsigned int	2 or 4
`'l'`	signed long int	4
`'L'`	unsigned long int	4
`'f'`	float	4
`'d'`	float	8

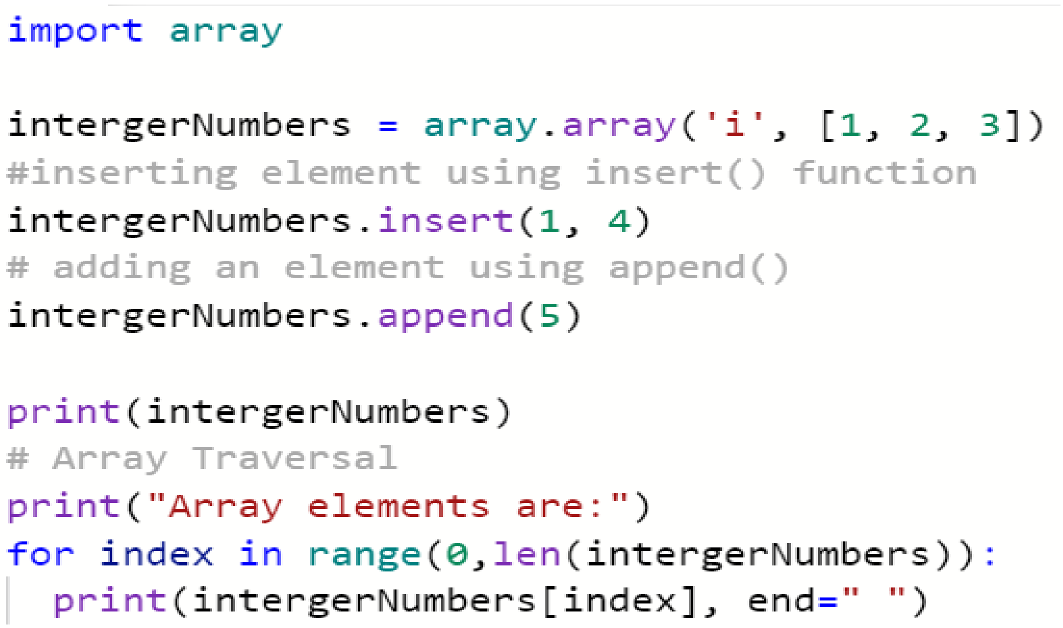
Type codes supported by the array module.

Python Array Methods



Method	Description
append()	Adds an element at the end of the list
clear()	Removes all the elements from the list
copy()	Returns a copy of the list
count()	Returns the number of elements with the specified value
extend()	Add the elements of a list (or any iterable), to the end of the current list
index()	Returns the index of the first element with the specified value
insert()	Adds an element at the specified position
pop()	Removes the element at the specified position
remove()	Removes the first item with the specified value
reverse()	Reverses the order of the list
sort()	Sorts the list

Example 1: Insertion and Traversal

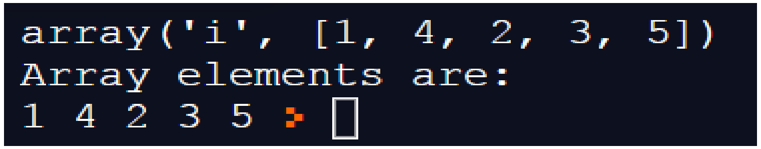


```python
import array

intergerNumbers = array.array('i', [1, 2, 3])
#inserting element using insert() function
intergerNumbers.insert(1, 4)
# adding an element using append()
intergerNumbers.append(5)

print(intergerNumbers)
# Array Traversal
print("Array elements are:")
for index in range(0,len(intergerNumbers)):
    print(intergerNumbers[index], end=" ")
```

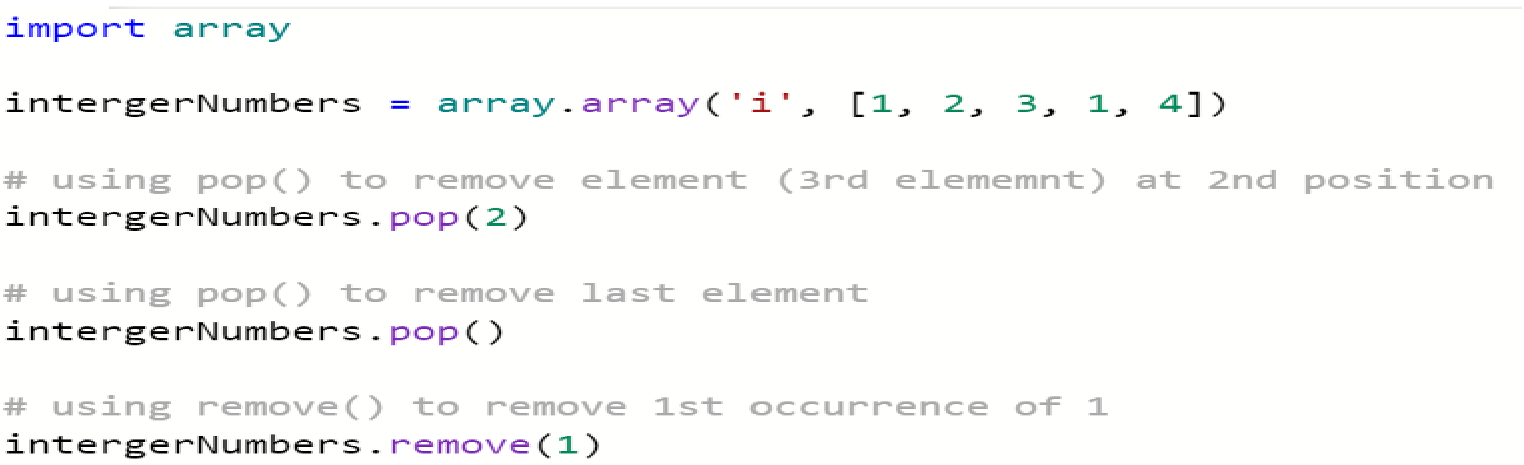
Output



```
array('i', [1, 4, 2, 3, 5])
Array elements are:
1 4 2 3 5
```

Example 2: Removing Elements from the Array

- **remove()**
  - remove() method only removes one element at a time.
  - it remove the first occurrence of the searched element.
  - in remove() method, an Error arises if element doesn’t exist.
- **pop()**
  - pop() function can also be used to remove and return an element from the array.
  - pop() method is to remove element from a specific position of the array.
  - by default it removes only the last element of the array.



```
import array

intergerNumbers = array.array('i', [1, 2, 3, 1, 4])

# using pop() to remove element (3rd elememnt) at 2nd position
intergerNumbers.pop(2)

# using pop() to remove last element
intergerNumbers.pop()

# using remove() to remove 1st occurrence of 1
intergerNumbers.remove(1)
```