Table of Contents

An array is a collection of similar data items stored at contiguous memory locations.

For example,

Example 01:

Array initialization:

```
int a[6] = {2, 4, 6, 8, 10, 12};
```

In above example,

- int is the type of the array
- ‘a’ is the name of the array
- a[6] shows number of elements in array

- Bracket {}, contains the array elements.
- Each element in array has its unique index number.

Index numbers of array elements in above example,

- 2 has index number 0
- 4 has index number 1
- 6 has index number 2
- 8 has index number 3
- 10 has index number 4
- 12 has index number 5

Example 02:

```
int a[ ] = {2, 4, 6, 8, 10, 12};
```

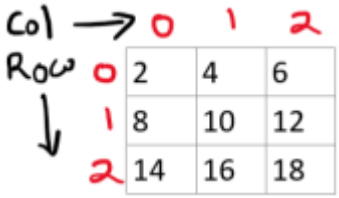
This example is same as example 01 above. Only size of a[ ] is not defined.

Types of arrays

One-dimensional array:

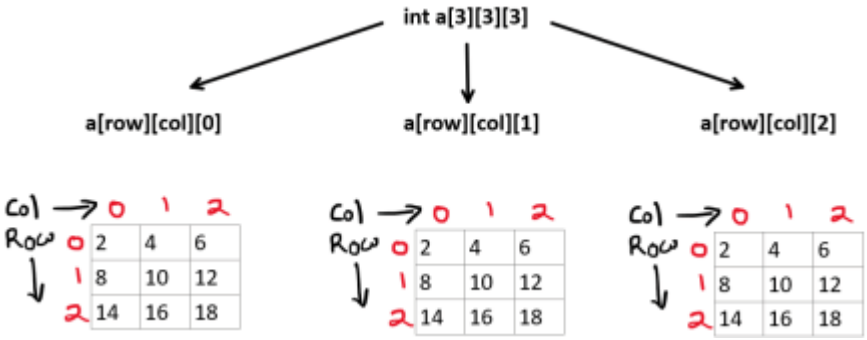
Multi-dimensional array:

Two-dimensional array



Three-dimensional array

Three-domensional array is like a cuboid.



Syntax for arrays

```
One-dimensional array
int arr[i];
```

```
Two-dimensional array
```

```
int arr[i][j];
```

```
Three-dimensional array
int arr[i][j][k];
```

Operations on array

1. Traversal : Visiting each element once.
2. Insertion : Process of inserting one or more elements in an array.
3. Deletion : Process of deleting one or more elements in an array.
4. Searching : Process of searching specific value in an array.
5. Sorting : Process of arranging elements in an array.

How to access array elements ?

Here the array index number is used.

```
#include <stdio.h>
int main() {
int a[5] = {2,4,6,8,10};
printf("%d\n",a[0]); // Accessing using index number
```

```
  printf("%d\n",a[1]);
  printf("%d\n",a[2]);
  printf("%d\n",a[3]);
  printf("%d",a[4]);
  return 0;
  }
```

Output

```
Index numbers for 2,4,6,8,10 is 0,1,2,3,4 respectively
2
4
6
8
10
```

Related Posts:

1. Traversal operation on array
2. Insertion Operation on Array
3. Net 42
4. Net 14
5. Net 13
6. Net 12
7. Net 35
8. Net 32
9. Net 29
10. Net 27
11. Net 52
12. Net 51

13. Net 45
14. Net 41
15. Net 38
16. Net 37
17. Net 36
18. GATE CS | Binary tree questions | Prof. Jayesh Umre
19. GATE | Binary Search Tree | Related Questions | Prof. Jayesh Umre
20. GATE 2004, Calculate height of Binary Tree | Prof. Jayesh Umre
21. GATE 2010 Binary tree descendent | Prof. Jayesh Umre
22. Linked List in Data Structure
23. Data Structures: Definition, Importance, Applications and types
24. Concepts of Data and Information