

# C Array

An array is defined as the collection of similar type of data items stored at contiguous memory locations. It is a fixed size . Arrays are the derived data type in C programming language which can store the primitive type of data such as int, char, double, float, etc.

## Properties of Array

The array contains the following properties.

- Each element of an array is of same data type and carries the same size, i.e., int = 4 bytes.
- Elements of the array are stored at contiguous memory locations where the first element is stored at the smallest memory location.
- Elements of the array can be randomly accessed since we can calculate the address of each element of the array with the given base address and the size of the data element.

## Declaration of C Array

We can declare an array in the c language in the following way.

1. `data_type array_name [array_size];`

```
int marks[5];
```

Here, int is the *data\_type*, marks are the *array\_name*, and 5 is the *array\_size*.

## Initialization of C Array

The simplest way to initialize an array is by using the index of each element. We can initialize each element of the array by using the index. Consider the following example.

1. `marks[0]=80; //initialization of array`
2. `marks[1]=60;`
3. `marks[2]=70;`

4. marks[3]=85;
5. marks[4]=75;

80	60	70	85	75
marks[0]	marks[1]	marks[2]	marks[3]	marks[4]

### Initialization of Array

#### C array example

1. `#include<stdio.h>`
2. `int main(){`
3. `int i=0;`
4. `int marks[5];//declaration of array`
5. `marks[0]=80;//initialization of array`
6. `marks[1]=60;`
7. `marks[2]=70;`
8. `marks[3]=85;`
9. `marks[4]=75;`
10. `//traversal of array`
11. `for(i=0;i<5;i++){`
12. `printf("%d \n",marks[i]);`
13. `}//end of for loop`
14. `return 0;`
15. `}`

#### Output

```
80
60
70
85
75
```

#### C Array: Declaration with Initialization

We can initialize the c array at the time of declaration. Let's see the code.

1. `int marks[5]={20,30,40,50,60};`

In such case, there is **no requirement to define the size**. So it may also be written as the following code.

1. `int marks[]={20,30,40,50,60};`

## Two Dimensional Array in C

The two-dimensional array can be defined as an array of arrays. The 2D array is organized as matrices which can be represented as the collection of rows and columns.

### Declaration of two dimensional Array in C

The syntax to declare the 2D array is given below.

1. `data_type array_name[rows][columns];`

Consider the following example.

1. `int twodimen[4][3];`

Here, 4 is the number of rows, and 3 is the number of columns.

### Initialization of 2D Array in C

In the 1D array, we don't need to specify the size of the array if the declaration and initialization are being done simultaneously. However, this will not work with 2D arrays. We will have to define at least the second dimension of the array. The two-dimensional array can be declared and defined in the following way.

1. `int arr[4][3]={{1,2,3},{2,3,4},{3,4,5},{4,5,6}};`

### Two-dimensional array example in C

1. `#include<stdio.h>`

```
2. int main(){
3. int i=0,j=0;
4. int arr[4][3]={1,2,3},{2,3,4},{3,4,5},{4,5,6}};
5. //traversing 2D array
6. for(i=0;i<4;i++){
7.     for(j=0;j<3;j++){
8.         printf("arr[%d] [%d] = %d \n",i,j,arr[i][j]);
9.     }//end of j
10. }//end of i
11. return 0;
12. }
```

### Output

```
arr[0][0] = 1
arr[0][1] = 2
arr[0][2] = 3
arr[1][0] = 2
arr[1][1] = 3
arr[1][2] = 4
arr[2][0] = 3
arr[2][1] = 4
arr[2][2] = 5
arr[3][0] = 4
arr[3][1] = 5
arr[3][2] = 6
```