

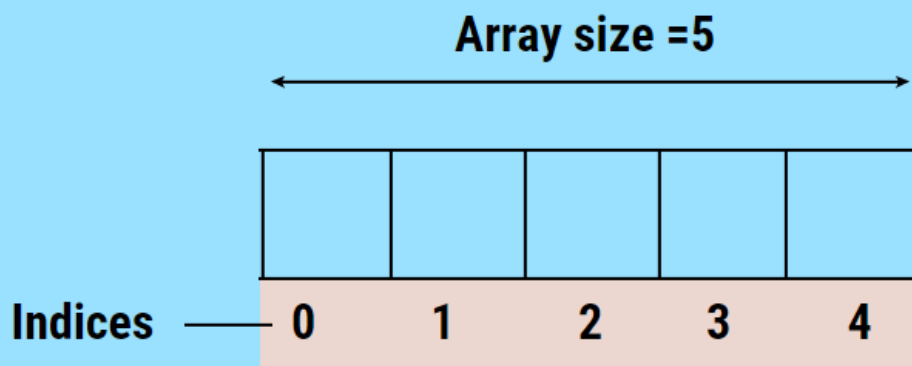


(<https://www.educba.com/software-development/>)

← (<https://www.educba.com/c-ftell/>)

→ (<https://www.educba.com/2-d-arrays-in-c/>)

Arrays in C Programming



educba.com

Introduction to Arrays in C Programming

The array is a type of data structure that is used to store homogeneous data in contiguous memory locations. Following are arrays in C programming.





(<https://www.educba.com/software-development/>)

Web development, programming languages, Software testing & others

Here index refers to the location of an element in the array. Let us imagine if $A[L]$ is the name of the array, where “A” is the variable name, and “L” is the length of the array, i.e. the number of elements present in the array.

Then $A[i]$ represents the element at that “i+1”th position in the array, .for example:

$A[6] = 72$ means element at 6+1 th location of the array.

Need for Array

It helps to represent a large number of elements using a single variable. It also makes accessing of element faster easy to store in memory location using the index of the array that represents the location of an element in the array...

Accessing Elements in Array

Accessing any element in the array is much easier and can be done in $O(1)$ complex

🔗 Popular Course in this category



C Programming Training (3 Courses, 5 Project)

3 Online Courses | 5 Hands-on Projects | 34+ Hours | Verifiable Certificate of Completion | Lifetime Access

★★★★★ 4.5 (8,612 ratings)





(<https://www.educba.com/software-development/>)

Related Courses

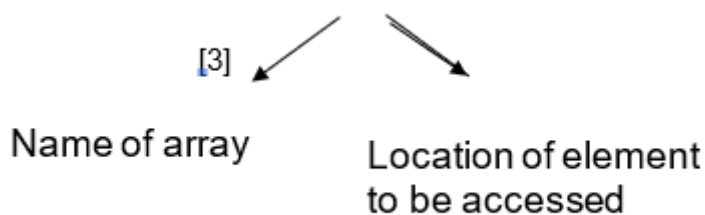
C++ Training (4 Courses, 5 Projects, 4 Quizzes) (<https://www.educba.com/software-development/courses/c-course/?btnz=edu-blg-inline-banner1>)

Java Training (40 Courses, 29 Projects, 4 Quizzes) (<https://www.educba.com/software-development/courses/java-course/?btnz=edu-blg-inline-banner1>)

Indexes of an array start from 0 till -1. 0 indicates the first element of the array, and -1 indicates the last element of the array. Similarly, -2 indicates the last but one element of the array.

For Example:

Let A be an array with length 7, and one needs to access the element with value 94, then he must use A[3].



3	5	85	94	62	5	2
---	---	----	----	----	---	---

Syntax

`printf("%d", A[3])` – This will print 94, where 3 is the index which we need to access and a is the variable of the array.





[\(https://www.educba.com/software-development/\)](https://www.educba.com/software-development/)

Syntax 1

int A[7] = {21,56,32,52,63,12,48} – Declaring the length and elements of array

C Program

```
#include<stdio.h>

int main{
    int a[7] = {21,56,32,52,63,12,48};
    int i;
    for(i=0;i<7;i++){
        printf("%d\n",a[i]);
    }
    return 0;
}
```

Output:

```
21
56
32
52
63
12
48
```

Syntax 2

int A[]={21,56,32,52,63,12,48} – Declaring the length of elements of array





[\(https://www.educba.com/software-development/\)](https://www.educba.com/software-development/)

```
int a[] = {21,56,32,52,63,12,48};  
int i;  
for(i=0;i<7;i++){  
    printf("%d\n",a[i]);  
}  
return 0;  
}
```

Output:

```
21  
56  
32  
52  
63  
12  
48
```

Syntax 3

int A[7]; – Declaring the length of the array only.

C Program

```
#include<stdio.h>  
  
int main{  
    int a[7] ;  
  
    int i;  
    printf("Please enter the array elements");  
    for(i=0;i<7;i++){
```





(<https://www.educba.com/software-development/>)

```
printf("%d\n", a[i]),
}
return 0;
}
```

Output:

```
Please enter the array elements2
3
4
5
6
7
8
Elements of array are2 3 4 5 6 7 8
```

Syntax 4

int A[7]={0};- Declaring length of the array and the element when an element is the same at all positions.

C Program

```
#include<stdio.h>
int main{
int a[7]={0} ;

int i;
printf("Elements of array are");
for(i=0;i<7;i++){
```





(<https://www.educba.com/software-development/>)

Output:

```
Elements of array are 0 0 0 0 0 0 0
```

Syntax 5

Declaring length of the array and also the value of elements where all values are the same

Case1 – `int a[3] = {[0..1]=3}` –

Case 2 – `int a[3] = {0};`–

Syntax 6

`int *a;`– Declaring array as a pointer to the location of elements.

No Index Out of Bound Checking

In case one attempts to access the element out of bounds of the array, no error is shown by the compiler; instead, it generates a warning. And also gives an unexpected output.

Example

`a[4] = {2,3,4,5};`

If we write `printf(a[4]);`

The output will be 225263545 – Unexpected



Also, In C, it is compiler does not error to initialize an array with more number elements than the specified length in the declaration. For example, the below program does not show an error



(<https://www.educba.com/software-development/>)

```
#include<stdio.h>

int main{

int arr[2]={10,22,56,32,45,89} ;

int i;

printf("Elements of array are");

for(i=0;i<2;i++){

printf("%d\n",arr[i]);

}

return 0;

}
```

Output:

```
main.c:4:25: warning: excess elements in array initializer
main.c:4:25: note: (near initialization for 'arr')
main.c:4:29: warning: excess elements in array initializer
main.c:4:29: note: (near initialization for 'arr')
main.c:4:33: warning: excess elements in array initializer
main.c:4:33: note: (near initialization for 'arr')
Elements of array are10 20
```

Retrieval of Elements in Array

Retrieval of elements of an array and printing them is a very easy task. It just requires one loop to print n elements of an array. thus the complexity of such a program is $O(n)$.

For eg- let int a[7] = {23,56,8,944,58,24,5};

Program for printing the elements of an array is

C Program





(<https://www.educba.com/software-development/>)

```
printf("Elements of array are");  
for(i=0;i<7;i++){  
printf("%d\n",arr[i]);  
}  
return 0;  
}
```

Output:

```
23  
56  
8  
944  
58  
24  
5
```

Multidimensional Array

C language also allows multidimensional arrays (<https://www.educba.com/multidimensional-array-in-c/>), i.e. arrays that can hold elements in rows as well as columns.

Declaration

While declaring the multidimensional array, one must specify the length of all dimensions except the left one because that is optional.

Example

Declaring array in the below manner will result in an error as dimensions other than left not specified.



Let's see an example



[\(https://www.educba.com/software-development/\)](https://www.educba.com/software-development/)

```
}
```

Example

Below is one of the correct syntax for declaration of multidimensional array in C.

```
Int a[][3]={  
  
{52,56,86},{44,6,21}  
  
}
```

Passing Array as Parameter in Function

Sometimes while making a function, we require the function to use a number of variables that it needs to take from different functions. At that time, those variables must be passed as a parameter to for that function call. But eventually, as the number of variables increases, we must use an array to pass the variable, or if some operations need to be performed on the arrays, then the need arises to pass a complete array as a parameter in a function. For passing an array as a variable to the function :

1. Call by Value

In this type of method calling, the actual values of the array are copied to the formal parameter where both are stored in a different location; thus, any change made in the values does not get reflected in the function.

C Program

```
#include <stdio.h>
```





(<https://www.educba.com/software-development/>)

```
int main()
{
    char arr[] = {'a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i', 'j'};
    for (int x=0; x<10; x++)
    {
        show(arr[x]); //value of array //elements are passed as an argument
    }
    return 0;
}
```

Output:

```
a b c d e f g h i j
```

2. Call by Reference

While calling a function when instead of passing the actual values of the array, the reference to the variable is passed as a parameter, then it is known as call by reference.

C Program

```
#include <stdio.h>
Void show( char ch)
{
    printf("%c ", ch);
}
int main()
{
```





(<https://www.educba.com/software-development/>)

```
argument
}
return 0;
}
```

Output:

```
1 2 3 4 5 6 7 8 9 0
```

3. Passing the Whole Array as an Argument

E.g., Let arr be an array of 7 elements. disp is a function to display the elements of an array which take 2 arguments, first that points to the first location of the array and other the length of the array (var2). While calling the function arr variable that points to the location of the first element of array and length, i.e. 7 is passed.

C Program

```
#include <stdio.h>
void disp( int *var1, int var2)
{
    for(int x=0; x<var2; x++)
    {
        printf("Value of var_arr[%d] is: %d \n", x, *var1);
        /*increment pointer for next element fetch*/
        var1++;
    }
}
```





(<https://www.educba.com/software-development/>)

```
return 0;  
}
```

Output:

```
Value of arr[0] is: 12  
Value of arr[1] is: 22  
Value of arr[2] is: 38  
Value of arr[3] is: 85  
Value of arr[4] is: 65  
Value of arr[5] is: 66  
Value of arr[6] is: 77
```

Memory Allocation of Array

Memory representation in C language is considered to be divided into 5 sections as given below:-

1. Text segment
2. Initialized data segment
3. Uninitialized data segment
4. Stack
5. Heap

Data, heap, and stack are the three segments where arrays can be allocated memory to store their elements, the same as other variables.

1. **Dynamic Arrays:** Dynamic arrays are arrays (<https://www.educba.com/dynamic-array-in-java/>), which needs memory location to be allocated at runtime. For these type of arrays, memory is allocated at the heap memory (<https://www.educba.com/what-is-heap-memory/>)





(<https://www.educba.com/software-development/>)

arrays. These types of arrays get memory allocated on the stack segment.

Character Array

In C, strings are considered as a single-dimensional array of characters with null character '\0' in its last position that the compiler automatically adds to it.

For example, "I love coding" is considered as a single dimension array in c of length 14, including the '\0' character in the end.

Declaration: There are 2 ways to declare and initialize the character array-

1. `char str[12] = "i love code";`
2. `char str[12] = {'I', 'l', 'o', 'v', 'e', ' ', 'c', 'o', 'd', 'e', '\0'};` – Here we must end it with '\0' character at the end.
3. `Char ch[3] = 'modi'` – Illegal declaration

Taking Input and Output

While taking input and displaying output in C for char array '%c' can be used `scanf()` and `printf()` function respectively.

While implementing the same for strings, "%s" can be used but stops scanning on the occurrence of the first whitespace character.

C Program:

```
#include <stdio.h>
#include<string.h>
int main()
```





(<https://www.educba.com/software-development/>)

```
printf("%s",str);  
return 0;  
}
```

Output:

```
main.c:8:15: warning: format '%[^  
Enter a stringLets start coding  
Lets start coding
```

Other than printf and scanf functions, C also provides string functions such as gets() and puts() to ignore (<https://www.educba.com/c-puts-function/>) white spaces of string while scanning and printing.

Conclusion

An array is a type of data structure used to store homogeneous data in a contiguous memory location. Arrays in Programming are used as a representation for different complex data structures such as a tree, heap, etc. C language allows multidimensional arrays (<https://www.educba.com/multidimensional-array-in-python/>) for all primitive data types. Strings are also represented as a character array with the null character '\0' as its last character. Arrays in Programming allow the fast retrieval and direct accessing of elements of an array using the index where the element is stored.

Recommended Articles

This is a guide to Arrays in C Programming. Here we discuss the Introduction, Needs of and Passing Array Functions, including Call by Value, Call by Reference, and Passing whole array as an argument. You may also look at the following articles to learn more –





[\(https://www.educba.com/software-development/\)](https://www.educba.com/software-development/)

C PROGRAMMING TRAINING (3 COURSES, 5 PROJECT)

- ☒ 3 Online Courses
- ☒ 5 Hands-on Projects
- ☒ 34+ Hours
- ☒ Verifiable Certificate of Completion
- ☒ Lifetime Access

Learn More

[\(https://www.educba.com/software-development/courses/c-programming-course/?btnz=educblg-inline-banner3\)](https://www.educba.com/software-development/courses/c-programming-course/?btnz=educblg-inline-banner3)

About Us

Blog (<https://www.educba.com/blog/?source=footer>)

Who is EDUCBA? (<https://www.educba.com/about-us/?source=footer>)





[\(https://www.educba.com/software-development/\)](https://www.educba.com/software-development/)

Contact Us (<https://www.educba.com/contact-us/?source=footer>)

Verifiable Certificate (<https://www.educba.com/software-development/verifiable-certificate/?source=footer>)

Reviews (<https://www.educba.com/software-development/reviews/?source=footer>)

Terms and Conditions (<https://www.educba.com/terms-and-conditions/?source=footer>)

Privacy Policy (<https://www.educba.com/privacy-policy/?source=footer>)

Apps

iPhone & iPad (<https://itunes.apple.com/in/app/educba-learning-app/id1341654580?mt=8>)

Android (<https://play.google.com/store/apps/details?id=com.educba.www>)

Resources

Free Courses (<https://www.educba.com/software-development/free-courses/?source=footer>)

Java Tutorials (<https://www.educba.com/software-development/software-development-tutorials/java-tutorial/?source=footer>)

Python Tutorials (<https://www.educba.com/software-development/software-development-tutorials/python-tutorial/?source=footer>)

All Tutorials (<https://www.educba.com/software-development/software-development-tutorials/?source=footer>)

Certification Courses

All Courses (<https://www.educba.com/software-development/courses/?source=footer>)





[\(https://www.educba.com/software-development/\)](https://www.educba.com/software-development/)

course/?source=footer)

Become a Selenium Automation Tester (<https://www.educba.com/software-development/courses/selenium-training-certification/?source=footer>)

Become an IoT Developer (<https://www.educba.com/software-development/courses/iot-course/?source=footer>)

ASP.NET Course (<https://www.educba.com/software-development/courses/asp-net-course/?source=footer>)

VB.NET Course (<https://www.educba.com/software-development/courses/vb-net-course/?source=footer>)

PHP Course (<https://www.educba.com/software-development/courses/php-course/?source=footer>)

© 2022 - EDUCBA. ALL RIGHTS RESERVED. THE CERTIFICATION NAMES ARE THE TRADEMARKS OF THEIR RESPECTIVE OWNERS.

