

Introduction to Pointers in Language

Ravindra Singh
Asst.Prof.
Computer Department
Durga Mahavidyalaya, Raipur (C.G)

A Pointer in C language is a variable that holds a memory address. This memory address is the address of another variable (mostly) of same data type.

In simple words, if one variable stores the address of second variable then the first variable can be said to **point towards** the second variable.

What is a Memory Address in C?

Whenever a variable is defined in C language, a memory location is assigned for it, in which it's value gets stored. We can check this memory address, using the & symbol.

If `var` is the name of the variable, then `&var` will give it's address.

Let's write a small program to see memory address of any variable that we define in our program.

```
include<stdio.h>

void main()
{
    int var = 7;

    printf("Value of the variable var is: %d\n", var);

    printf("Memory address of the variable var is: %x\n", &var);
}
```

value of the variable var is: 7

Memory address of the variable var is: bcc7a00

Also while using the `scanf()` function, we mention `&var` to take user input for any variable `var`.

```
scanf("%d", &var);
```

Copy

This is used to store the user input value to the memory address of the variable `var`.

What is a Pointer in C?

Like we mentioned earlier, a Pointer in C language is a variable that holds a memory address.

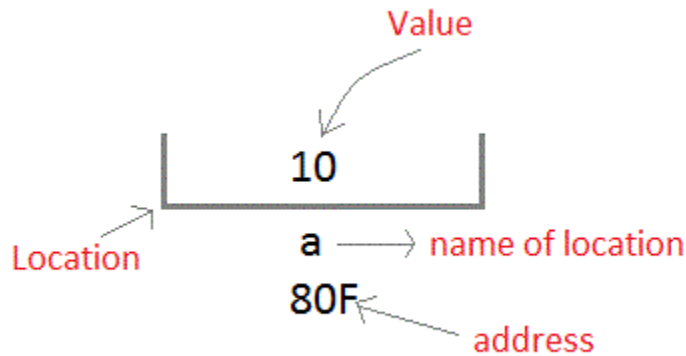
Pointers are used to access memory of a variable and manipulate the value stored in it.

Pointers are one of the most distinct and exciting features of C language. It provides power and flexibility to the language. Although pointers may appear a little confusing and complicated in the beginning, but trust me, once you understand the concept, you will be able to do so much more with C language.

Whenever a **variable** is declared in a program, system allocates a location i.e an address to that variable in the memory, to hold the assigned value. This location has its own address number, which we saw in the program above.

Let us assume that system has allocated memory location 80F for a variable a.

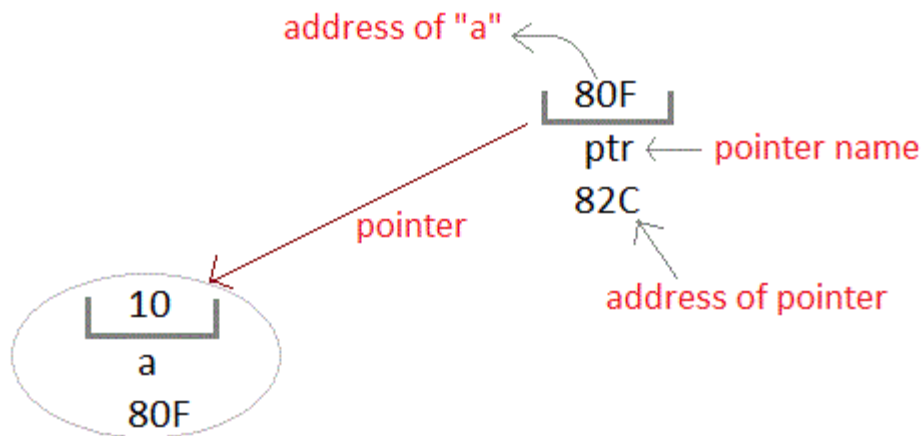
```
int a = 10;
```



We can access the value 10 either by using the variable name a or by using its address 80F.

The question is how we can access a variable using its address? Since the memory addresses are also just numbers, they can also be assigned to some other variable. The variables which are used to hold memory addresses are called **Pointer variables**.

A **pointer** variable is therefore nothing but a variable which holds an address of some other variable. And the value of a **pointer variable** gets stored in another memory location.



Pointer Variable in C

Like we mentioned above that a pointer is also a variable, but with a little twist, that is, it only stores address of other variables.

So if you have to define a pointer variable, the syntax is a little different.

Following is the **syntax** for declaring a variable as a pointer:

```
type *name;
```

Here, **type** is the data type of the pointer, and the **name** is the name of the pointer variable.

And the * operator with the name, informs the compiler that the variable is a pointer.

The data type of the pointer variable should be the same as of the variable to which the pointer is pointing.

Pointer Operators in C

There are two pointer operators in C, they are:

1. * operator
2. & operator

We have covered [operators in C](#) in detail separately.

The & operator returns the memory address of its operand. For example,

```
a = &b;
```

In the variable a the memory address of the variable b will get stored.

The * operators is the complement of &. This operator returns the value located at the given address.

For example, if a contains the memory address of the variable b, then the code,

```
c = *a;
```

Will store the value of the variable b into c.

Pointers in C Example

Let's see a basic code example where we will create a pointer and assign it a value.

```
int main(void)
{
    int x = 99;

    // declare a pointer
    int *ptr;

    // assign value to pointer
    ptr = &x;
```

```
printf("Value at ptr is: %d \n", *ptr);

printf("Address pointed by ptr is: %p \n", ptr);

return 0;
}
```

```
Value at ptr is: 99
Address pointed by ptr is: 0x7fff99c0e6c4
```

[Run Code →](#)

The %p format specifier is used for pointer variable.

Benefits of using pointers in C

Below we have listed a few benefits of using pointers:

1. Pointers are more efficient in handling Arrays in C and Structures in C.
2. Pointers allow references to function and thereby helps in passing of function as arguments to other functions.
3. Pointers also provide means by which a function in C can change its calling arguments.
4. It reduces length of the program and its execution time as well.
5. It allows C language to support Dynamic Memory management.

In the next tutorial we will learn syntax of pointers, how to declare and define a pointer, and using a pointer. See you in the next tutorial.