

What is the difference between actual and formal arguments in C?

Before going to the difference between *actual* and *formal* arguments in C, first understand the two terms - argument and parameter. Dennis M Ritchie in his classic book on C language mentioned, "We will generally use *parameter* for a variable named in the parenthesized list in a function definition, and *argument* for the value used in a call of the function. The terms *formal argument* and *actual argument* are sometimes used for the same distinction." So don't get confused if you find the question like difference between *actual* and *formal* arguments or difference between *actual* and *formal* parameters. They both are the same.

The major difference between *actual* and *formal* arguments is that *actual* arguments are the source of information; calling programs pass *actual arguments* to called functions. The called functions access the information using corresponding *formal arguments*.

The following piece of code demonstrates *actual* and *formal* arguments.

```
/* Demonstrating difference between actual and formal arguments */
#include <stdio.h>

int addTwoInts(int, int); /* Prototype */

int main() /* Main function */
{
    int n1 = 10, n2 = 20, sum;

    /* n1 and n2 are actual arguments. They are the source
       of data. Caller program supplies the data to called
       function in form of actual arguments. */
    sum = addTwoInts(n1, n2); /* function call */

    printf("Sum of %d and %d is: %d \n", n1, n2, sum);
}

/* a and b are formal parameters. They receive the values from
   actual arguments when this function is called. */
int addTwoInts(int a, int b)
{
    return (a + b);
}

OUTPUT
=====
Sum of 10 and 20 is: 30
```

In above piece of code, the variables `n1` and `n2` are called *actual arguments*, whereas the variables `a` and `b` are called *formal arguments*. When values are passed to a called function the values present in *actual arguments* are copied to the *formal arguments*. In case of above program

the values of `n1` and `n2` are `10` and `20` respectively. The `main()` would call `addTwoInts` with `n1` and `n2` as its *actual* arguments, and `addTwoInts` will send back the computed sum to the `main()`.

Conclusively, when a function is called all *actual arguments* (those supplied by the caller) are evaluated and each *formal argument* is initialized with its corresponding *actual argument*.

Hope you have enjoyed reading differences between *actual* and *formal* arguments in C. Please do [write us](#) if you have any suggestion/comment or come across any error on this page. Thanks for reading!