

Difference Between Actual and Formal Parameters

The major difference between actual and formal parameters is that formal parameters are defined when the function is called, whereas actual parameters are defined when the function is invoked.

Key Differences Between Actual and Formal Parameters

Actual Parameters	Formal Parameters
Defined when it is invoked.	Defined when the function is called.
Passed on calling a function	They will be in the called function.
Data types will not be mentioned.	The data type must be included.
They are written in the function call.	They are written in the function definition.
They may be constant values.	They can be local variables.

Actual and Formal Parameters

The parameters are used to either pass or send out the information in a program. It is an essential part of any programming language to use the function. They are used to perform a specific task. There are two types of parameters that are widely used:

- Actual Parameters
- Formal Parameters

What are the Actual Parameters?

An actual parameter is the value of a function that is passed when the function is being invoked. The caller will assign the actual value. It is a parameter that is determined while invoking a subroutine.

Example of Actual Parameters

```
#include <stdio.h>

void multiplication (int x, int y) {

int multiplication;
```

```
addition = x*y;  
printf(“%d”,multiplication);  
}  
void main () {  
multiplication (2,3);  
multiplication (4,5);  
}
```

What are the Formal Parameters?

Let us learn about the formal parameters. A formal parameter is defined while determining the subroutine. Formal parameters are also used in the header of the function.

The syntax of the formal parameters is:

```
<return type> <method name> (formal parameters) {  
//set of statements to be executed  
}
```