# Difference Between Actual and Formal Parameters

## Key Difference - Actual vs Formal Parameters

Using Functions is an important concept in programming. A function is a number of statements that can perform some kind of a specific task. If the programmer writes all statements as a single program, it will become complex. Functions can be used to avoid that. They are also known as methods. Each function will have their own functionality. Functions improve code optimization and code reusability. There can be functions provided by the programming language or the functions written by the programmer. Each function has a name to identify it. After performing a certain task using a function, it can return a value. Some functions do not return any value. The data necessary for the function to perform the task is sent as parameters. Parameters can be actual parameters or Formal Parameters. The **key difference** between Actual Parameters and Formal Parameters is that **Actual Parameters are the values that are passed to the function when it is invoked while Formal Parameters are the variables defined by the function that receives values when the function is called.**

## What are Actual Parameters?

Actual parameters are values that are passed to a function when it is invoked. Refer the below program.

#include <stdio.h>

void addition (int x, int y) {

int addition;

addition = x+y;

printf("%d",addition);

}

void main () {

addition (2,3);

addition (4,5);

}

According to the above C program, there is a function named addition. In the main function, the value 2 and 3 are passed to the function addition. This value 2 and 3 are the actual parameters. Those values are passed to the method addition, and the sum of two numbers will display on the screen. Again, in the main program, new two integer values are passed to the addition method. Now the actual parameters are 4 and 5. The summation of 4 and 5 will display on the screen.

## What are Formal Parameters?

A function or a method follows a syntax similar to those given below:

<return type> <method name> (formal parameters) {

//set of statements to be executed

}

The method name is to identify the method. The return type specifies the type of the value the method will return. If the method does not return a value, the return type is void. If the function is returning an integer value, then the return type is an integer. The formal parameter list is enclosed in parenthesis. The list contains variable names and data types of all the necessary values for the method. Each formal parameter is separated by a comma. When the method is not accepting any input values, then the method should have an empty set of parentheses after the method name. e.g. addition () { }; The statements that should be executed are enclosed in curly braces.
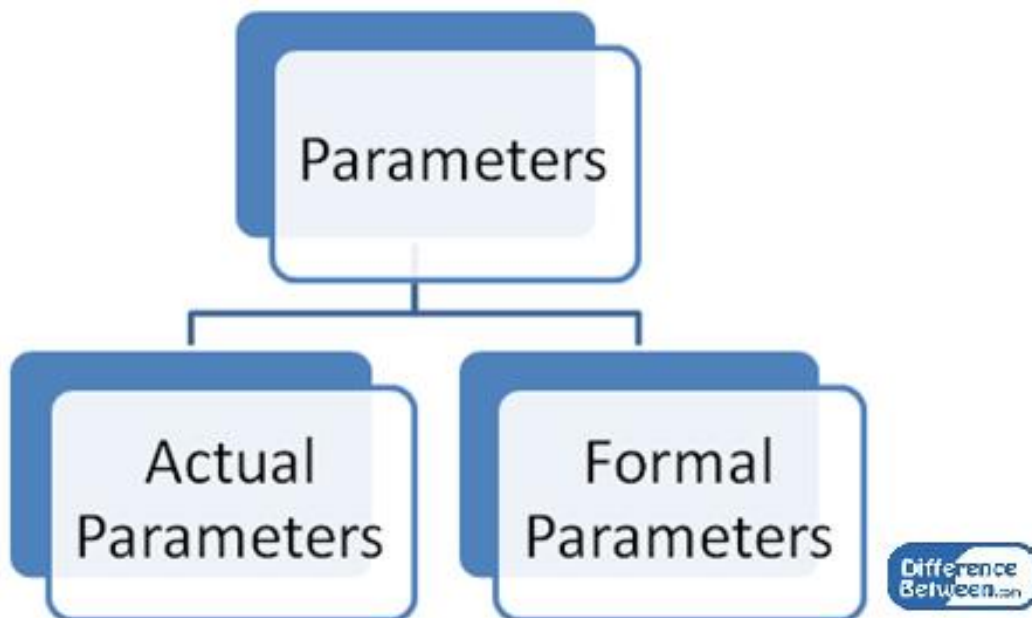


**Figure 01: Parameters**

Formal parameters are the variables defined by the function that receives values when the function is called. According to the above program, the values 2 and 3 are passed to the function addition. In the addition function, there are two variables called x and y. The value 2 is copied into variable x, and value 3 is copied into variable y. The variable x and y are not the actual parameters. They are copies of the actual parameters. They are known as formal parameters. These variables are only accessible within the method. After printing the addition of two numbers, the control is returned back to the main program.

## What are the Similarities Between Actual and Formal Parameters?

- Both are related to functions.
- Parameters are included inside the parenthesis.
- Each parameter is separated by a comma.

## What is the Difference Between Actual and Formal Parameters?

| Actual vs Formal Parameters | |
|---|---|
| The Actual parameters are the values that are passed to the function when it is invoked. | The Formal Parameters are the variables defined by the function that receives values when the function is called. |
| **Related Function** | |
| The actual parameters are passed by the calling function. | The formal parameters are in the called function. |
| **Data Types** | |
| In actual parameters, there is no mentioning of data types. Only the value is mentioned. | In formal parameters, the data types of the receiving values should be included. |

## Summary - Actual vs Formal Parameters

Using Functions is a useful concept in programming. Functions help to reduce code length and decrease complexity. It is also easy to do testing, debugging and improves code maintainability. Some functions might not need inputs, but some functions require inputs. It is possible to pass data to the functions as inputs. They are known as parameters. The two common terms that are related to functions are Actual Parameters and Formal Parameters. The difference between Actual Parameters and Formal Parameters is that Actual Parameters are the values that are passed to the function when it is invoked while Formal Parameters are the variables defined by the function that receives values when the function is called.

**Reference:**

1.tutorialspoint.com. "C Functions."  The Point.  Available here

## How to Cite this Article?

APA: Difference Between Actual and Formal Parameters.(2018 February 06). Retrieved (date), from http://differencebetween.com/difference-between-actual-and-vs-formal-parameters/

MLA: "Difference Between Actual and Formal Parameters" Difference Between.Com. 06 February 2018. Web.

Chicago: "Difference Between Actual and Formal Parameters." Difference Between.Com. http://differencebetween.com/difference-between-actual-and-vs-formal-parameters/ accessed (accessed [date]).