Passing by Value

In this lesson we will discuss the pass by value way of passing a parameter to a method in C#

WE'LL COVER THE FOLLOWING

Introduction

In **C#**, *arguments* can be passed to *parameters* either by *value* or by *reference*.

If *parameters* are passed as value type then a **new** copy of it will be created before *passing* to the *function*.

Meaning, for a value type *parameter*, if the function changes the *parameter* value then the **original** *variable* does **not** get changed.

Example

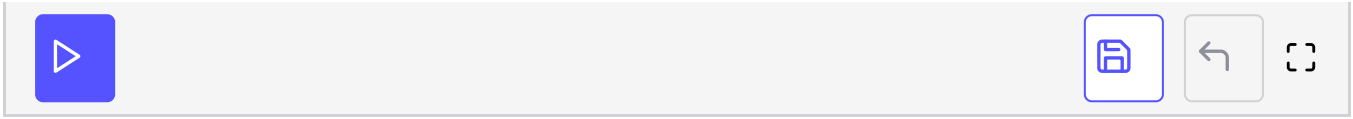
Let's take a look at an example in which value *arguments* are passed by value.

```
using System;

namespace FunctionApp
{
class Program
{
    static int  num = 3;
    static void Main()
    {
        System.Console.WriteLine("cube is {0}", cube(num));
        System.Console.WriteLine("if num is {0}", num);
    }
    //function with value type parameter
    private static int cube( int num)
    {
        int answer = num*num*num;
        //Assigning the calculated value to the parameter
        num = answer;
```

```
        return num;

    }
}
}
```



In the code above

- We make the function `cube(int num)`, and pass the parameter `num` to it by value.
- It computes the cube of `num`.
- Puts `num` equal to the *cube* computed.
- Returns updated `num` at end of function `cube`.
- The function `cube(num)` is then called in `Main()`.
- The value of `num` passed is **3**.
- The output **27** is then displayed on the console.

Online **number 18** we are updating the *parameter* value with the *calculated* value. As you can see it is not affecting the variable.

In the next lesson we will discuss the *pass by reference* way of passing a parameter to a method.