

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
```

```
namespace EjemplosFunciones
{
```

```
    class Program
    {
```

```
        //Esto es un ejemplo de un programa con funciones de varios tipos.
```

```
        //No vamos a hablar de modificadores, usaremos siempre static.
```

```
        //Las funciones son como programas independientes, que solo se relacionan entre ellas
```

```
        //a traves de lo que devuelven (tipo de funcion) o lo que reciben (parametros)
```

```
        //Las variables dentro de una funcion, son independientes de las otras funciones.
```

```
static void Main(string[] args)
{
    int num1, num2, resultado=0;
    //Primera funcion: Muy simple, es una funcion que no recibe ningun parametro ni devuelve nada
    Console.WriteLine("*****");
    Console.WriteLine("****Primera Funcion*****");
    Console.WriteLine("*****");
    mostrarMensaje();
}
```

```
*****
```

```
static void mostrarMensaje()
{
    //como la funcion no devuelve nada, es void.
    //Como no recibe ningun parametro, entre parentesis no pongo nada
    //En cada funcion podemos programar un programa entero
    //En esta mostraremos 5 veces un mensaje.
    int n ;
    for (n = 1; n < 6; n++)
    {
        Console.WriteLine("{0}.-Esto es un mensaje", n);
    }
}
```

```
//Vamos a pedir dos numeros al usuario, y vamos a realizar la misma funcion
```

```
//con distintas posibilidades.
```

```
Console.WriteLine("Entra dos numeros:");
num1 = int.Parse(Console.ReadLine());
num2 = int.Parse(Console.ReadLine());
```

```
//Segunda Funcion: Vamos a sumar estos numeros, y mostrar el resultado dese la funcion.
```

```
//para hacerlo, necesitamos enviar los numeros leidos aquí a la función
```

```
Console.WriteLine("*****");
Console.WriteLine("****Segunda Funcion*****");
Console.WriteLine("*****");
Console.WriteLine("Antes de la funcion: num1 vale {0}, num2 vale {1} y resultado vale {2}", num1, num2, resultado);
```

```
sumarNumeros1(num1, num2);
```

```
Console.WriteLine("Despues de la funcion, no se modifiko nada: num1 vale {0}, num2 vale {1} y resultado vale {2}", num1, num2, resultado);
```

```
*****
```

```
static void sumarNumeros1(int n1, int n2)
{
    int r = n1 + n2;
    Console.WriteLine("Dentro de sumarNumeros1. n1 vale {0}, n2 vale {1} y R vale {2}",n1,n2, r);
    //aunque modifiquemos ahora n1 o n2, veremos que no se modifican los del programa principal
    n1 = 3;
    n2 = 7;
    r = n1 + n2;
    Console.WriteLine("Dentro de sumarNumeros1, despues de modificar los numeros. n1 vale {0}, n2 vale {1} y R vale {2}", n1, n2, r);
}
```

```

}

//fijaros que las variables no tienen porque llamarse igual que en el programa principal.

//cuando llego aqui, el primer parametro que pongo donde llamo la funcion se guarda en n1

//y el segundo en n2

//aquí dentro las variables num1 y num2 no existen

```

```

//Tercera Funcion: Vamos a sumar estos numeros, y devolver el resultado al programa principal (main).

//para hacerlo, necesitamos enviar los numeros leídos aquí a la función, y guardar el resultado en una variable

//fijaros que como esta función devuelve un entero, no será static void sino que será static int

```

```

Console.WriteLine("*****");
Console.WriteLine("****Tercera Funcion****");
Console.WriteLine("*****");

Console.WriteLine("Antes de la funcion: num1 vale {0}, num2 vale {1} y resultado vale {2}", num1, num2, resultado);

resultado=sumarNumeros2(num1, num2); // En resultado se guardará r , que retorna el valor de la función.

Console.WriteLine("Después de la funcion, solo se modifico resultado: num1 vale {0}, num2 vale {1} y resultado vale {2}", num1,
num2, resultado);
.....
static int sumarNumeros2(int n1, int n2)
{
    int r = n1 + n2;
    Console.WriteLine("Dentro de sumarNumeros2. n1 vale {0}, n2 vale {1} y R vale {2}", n1, n2, r);

    //con esta instruccion finaliza la funcion y se guarda el valor de r en la variable
    //que ponemos a la izquierda del igual donde llamamos la funcion
    //es decir, en el main decimos: resultado=sumarNumeros2(num1,num2);, pues el valor
    //de r se guardará en la variable resultado
    return r;
}

```

```

//Cuarta Funcion: Hasta ahora pasabamos los parametros por valor. Vamos a pasar uno por referencia..

//Vamos a enviar tres parametros, y guardaremos el resultado en el tercero

//fijaros que reinicio resultado para que se vean los cambios

```

```

resultado = 0;
Console.WriteLine("*****");
Console.WriteLine("****Cuarta Funcion****");
Console.WriteLine("*****");

Console.WriteLine("Antes de la funcion: num1 vale {0}, num2 vale {1} y resultado vale {2}", num1, num2, resultado);

sumarNumeros3(num1, num2, ref resultado);

Console.WriteLine("Después de la funcion, solo se modifico resultado: num1 vale {0}, num2 vale {1} y resultado vale {2}", num1, num2, resultado);
*****

```

```

static void sumarNumeros3(int n1, int n2, ref int n3)
{
    //solo con esto, modificamos n3 y la variable resultado del programa principal.
    n3 = n1 + n2;
    //aunque modifiquemos n1 y n2, estos no quedan modificados en el programa principal.
    n1 = 45;
}

```

```

//Quinta Funcion: Vamos a hacer una funcion que guardara la suma y la resta.

//Como no podemos "devolver" (con return) dos valores, uno lo devolveremos con return y el otro

//con una variable por referencia

```

//Vamos a enviar tres parametros, y guardaremos la resta en el tercero y la suma con el return

```

    resultado = 0;
    Console.WriteLine("*****");
    Console.WriteLine("****Quinta Funcion****");
    Console.WriteLine("*****");
    int resultadoresta = 0;
    Console.WriteLine("Antes de la funcion: num1 vale {0}, num2 vale {1} y resultado vale {2}, y resultadoresta vale {3}", num1, num2, resultado, resultadoresta);

    resultado=sumarNumeros4(num1, num2, ref resultadoresta);

    Console.WriteLine("Despues de la funcion, solo se modifico resultado y resultadoresta: num1 vale {0}, num2 vale {1} y resultado vale {2}, y resultadoresta vale {3}", num1, num2, resultado, resultadoresta);

```

```

static int sumarNumeros4(int n1, int n2, ref int n3)
{
    //vamos a guardar la suma en r, y devolverla con return, y la resta en n3

    int r;

    r = n1 + n2;

    n3 = n1 - n2;

    //si ponemos el return antes de la linea n3=n1-n2, esa operación nunca se ejecutará.

    //Cuando una funcion llega al return, se para la función.

    return r;
}

```

//Nota: En éste ultimo ejemplo podíamos usar out en vez de ref, la única diferencia es

//que no hubiese hecho falta poner un 0 en resultadoresta, siempre y cuando no imprimamos su valor antes de llamar a la función.

```

Console.WriteLine("Pulsa intro para finalizar");

```

```

Console.Read();

```

```

}

```

```

}

```

```

}

```