

C sharp NET/Capítulo 5

Funciones

Una función te permite de encapsular una pieza de código y llamarlo desde otras partes de su código.

Estructura básica de una función

En C#, las funciones son básicamente como sigue:

```
<visibility> <return type> <name>(<parameters>)  
{  
    <function code>  
}
```

Ejemplo 5.1:

```
public void DoStuff()  
{  
    Console.WriteLine("I'm doing something...");  
}
```

Para llamar una función, simplemente hay que escribir su nombre, abrir un paréntesis, a continuación escribir los parámetros (si los tiene) separados por comas y por último cerrar el paréntesis. Siguiendo con el ejemplo 5.1:

```
DoStuff();
```

Ahora analicemos el código:

- La primera parte, PUBLIC, es la visibilidad, y es opcional. Si no se define ninguna, entonces la función será PRIVATE.
- Lo siguiente es el tipo de retorno. Debe ser cualquier tipo válido en C#. En el ejemplo 5.1, VOID significa que esta función no retorna absolutamente nada.
- Por último fíjate que esta función no tiene parámetros pero por ello no pierde los paréntesis sino que hay que ponérselos.

Ahora observa el siguiente ejemplo (Ejemplo 5.2):

C sharp NET/Capítulo 5

```
public int AddNumbers(int number1, int number2)
{
    int result = number1 + number2;
    return result;
}
```

Ahora la función devuelve un entero y necesita de 2 parámetros (ambos enteros). Sabemos que devuelve un valor porque justo antes de la llave de cierre de la función vemos RETURN y el nombre de una variable RESULT. Además cuando escribimos la visibilidad de esta función PUBLIC a continuación sale el tipo INT que es el tipo de la variable devuelta RESULT.

¿Cómo llamamos a esta función? Veamos cómo lo haríamos para el ejemplo 5.2:

```
int result = AddNumbers(10, 5);
```

“not all code paths return a value”

Si a la hora de crear la función pusiéramos algo como este ejemplo 5.3

```
public int AddNumbers(int number1, int number2)
{
    int result = number1 + number2;
    if(result > 10)
    {
        return result;
    }
}
```

el compilador sacaría el siguiente error: *AddNumbers(int, int)’: not all code paths return a value*. El compilador nos recuerda que tenemos una función que a veces no devuelve nada, aunque sí lo habíamos prometido

Podemos resolver el problema poniendo un valor que se devuelva por defecto:

```
public int AddNumbers(int number1, int number2)
{
    int result = number1 + number2;
    if(result > 10)
    {
        return result;
    }
}
```

C sharp NET/Capítulo 5

```
}  
    return 0;  
}
```