

PROGRAMACIÓN

Guía de estudio 7: Sólo funciona para el metro

Funciones con parámetros en C#

Nivel: Secundario - Modalidad Educación Técnico-Profesional.
Ciclo: Segundo ciclo.
Especialidades: Informática, Programación.

Introducción

Como pudimos ver en la anterior guía, las funciones son muy útiles. Pero más útiles serían si pudiéramos pasarle información. En esta guía vamos a ver como ejecutar funciones con parámetros, es decir, que su comportamiento varía según valores que le podemos pasar.

¿Qué estamos aprendiendo?: Cómo crear y utilizar funciones que reciben valores y funciones que reciben y devuelven valores.

Recursos

Video nro 17 del curso:

<https://youtu.be/Ci6hPyUygv4>



Video nro 18 del curso:

https://youtu.be/k_jVeHEG-pc



Secuencia didáctica

A esta altura del curso ya debés estar super afilado. Esta guía es una continuación de la guía anterior, así que deberías tener bien claro lo que ya se vio de funciones para entender lo que vamos a ver (repasalo si es necesario). Mirá los videos y respondé las preguntas. Spoiler alert: acá los ejercicios ya se empiezan a poner cada vez más complicados.

<p>https://youtu.be/Ci6hPyUygv4</p>	
<p>https://youtu.be/k_jVeHEG-pc</p>	

Actividades

Después de mirar el video del link o del QR, te invitamos a que reflexiones con las siguientes preguntas.

No olvides escribir tus respuestas, te facilitarán realizar trabajos prácticos y repasar los temas que hayas estudiado con esta guía.

- 1) ¿Cuántos valores podemos pasarle a una función? ¿Tienen que ser todos del mismo tipo? Sí creo una función que devuelve un número, pero no guardo el resultado en ninguna variable ¿se ejecuta igual?

Ejemplifico esto último:

```
static void Main(string[] args)
{
```

```

String n;
n = Console.ReadLine();
print(n);
Console.ReadKey();
}
static int print(String Nombre)
{
    Console.WriteLine(Nombre);
    return 0;
}

```

- 2) Hacer una función que convierta grados a radianes sabiendo que 360° son 2π radianes. Hacer que funcione tanto para ángulos positivos como negativos.
- 3) Vamos a hacer un programa de facturación para un local.

"calculoyprint": función que recibirá dos valores: Precio unitario y cantidad de productos. La función deberá imprimir por pantalla "se han sumado ___ \$" reemplazando los guiones por el precio total de los productos (precio unitario por cantidad). Deberá devolver el mismo precio que se muestra en pantalla.

Main: deberá pedir el precio del producto y cuántas unidades de ese producto se quieren llevar. Se deberá mostrar el importe asociado a cada producto. El programa deberá terminar cuando se no se ingrese ningún precio (se toca "enter" sin nada escrito) y deberá mostrar el importe final.

- 4) Teniendo el siguiente programa escrito en el Main, escribí el código necesario para que funcione.

```

int a, b;
String op;
a = int.Parse(Console.ReadLine());
op = Console.ReadLine();
b = int.Parse(Console.ReadLine());
Console.WriteLine(calculadora(a, b, op));
Console.ReadKey();

```

- 5) Hacé un programa que pida un número y se lo pase a la función "caracteristicas". Esta función deberá devolver un número entero acorde:

- a) si es 0 -> 0
- b) si es par ->1
- c) si es impar ->2
- d) si es primo ->3
- e) par y negativo ->4
- f) impar y negativo ->5

Luego, en el Main se deberá imprimir en pantalla las características del número.
Pista: ¿no habíamos visto algo que iba número por número analizando el caso?

- 6) En una fábrica, a los trabajadores le descuentan el 1% de su salario cada vez que falta si su sueldo es mayor a 10000\$, el 3% si su sueldo es mayor a 25000\$ y el 5% si su sueldo es mayor a 50000\$. Hacer una función "descuentoSueldo" que reciba como parámetro el sueldo y devuelva el monto con el porcentaje descontado.

Para seguir aprendiendo...

No se si ya les pasó, pero suele ser común que nos olvidemos algún parámetro. También puede pasar que el valor que le pasamos sea casi siempre el mismo. Por suerte, C# nos da una pequeña ayudita. Existe algo que se llama parámetros por defecto. Estos nos permiten definir un valor que va a tomar el parámetro en caso de que no se especifique. Un ejemplo de esto es:

```
static void Main(string[] args)
{
    int a = sumar10();
    Console.WriteLine(a);
    Console.ReadKey();
}

static int sumar10(int i = 0)
{
    i += 10;
    return i;
}
```

- a) Implementa esto en el haciendo que la función printRect funcione así:

printRect(5, 4):	printRect(3):
salida:	salida:
*****	***
*****	***
*****	***

- b) ¿Cómo usarías esto para el ejercicio 3? Hacé el programa.