

# Què és C#?

- C#(pronunciat si sharp en anglès) és un llenguatge de programació orientat a objectes desenvolupat i estandarditzat per Microsoft com a part de la seva plataforma .NET.
- La sintaxi bàsica deriva de C/C++ i utilitza el model d'objectes de la plataforma .NET, similar al de Java, encara que inclou millores derivades d'altres llenguatges.
- El nom C Sharp va ser inspirat pel signe '#' que es compon de quatre signes '+' enganxats.
- Documentació: <https://learn.microsoft.com/es-es/dotnet/csharp/>

# Característiques

- Tipat estatic i fort.
- Orientat a objectes, tot en C# son objectes.
- Basat en C++ i Java, fortment estructurat amb us de Claus {} per delimitar blocs.

# Variables i tipus de dades

Tipo de dato de C#	Tamaño	Rangos	¿Con signo?	Sufijo
sbyte	8 bits (1 byte)	-128 a 127	Si	
byte	8 bits (1 byte)	0 a 255	No	
short	16 bits (2 bytes)	-32.768 a 32.767	Si	
ushort	16 bits (2 bytes)	0 a 65.535	No	
int	32 bits (4 bytes)	-2.147.483.648 a 2.147.483.647	Si	
uint	32 bits (4 bytes)	0 a 4.294.967.295	No	u
long	64 bits (8 bytes)	-9.223.372.036.854.775.808 a 9.223.372.036.854.775.807	Si	l
ulong	64 bits (8 bytes)	0 a 18.446.744.073.709.551.615	No	ul
float	32 bits (4 bytes)	6 a 9 dígitos		f
double	64 bits (8 bytes)	15 a 17 dígitos		d
decimal	128 bits (16 bytes)	28 a 29 dígitos		m
bool		true, false		

*decimal precio = 19.99m; // Sin el sufijo `m`, esto sería un `double` por defecto*

# Operators

- =

- <

- ||

- %

- +

- >=

- -

- <=

- /

- ==

- \*

- !=

- >

- &&

- +=

- -=

- ++

- --

# Comentaris

- //

- /\*                      \* /

Totes les sentències finalitzen amb ;

# Estructures Condicionals

- if
- if...else
- switch (case, break, default)

```
void DisplayMeasurement(double measurement)
{
    switch (measurement)
    {
        case < 0.0:
            Console.WriteLine($"Measured value is {measurement}; too low.");
            break;

        case > 15.0:
            Console.WriteLine($"Measured value is {measurement}; too high.");
            break;

        case double.NaN:
            Console.WriteLine("Failed measurement.");
            break;

        default:
            Console.WriteLine($"Measured value is {measurement}.");
            break;
    }
}
```

# Bucles

- while
- do...while
- for

C#

```
for (int i = 0; i < 3; i++)  
{  
    Console.Write(i);  
}  
// Output:  
// 012
```

# Arrays

## 1. Declaració d'arrays

- Declaració sense inicialització  
`int[] numeros;`  
`numeros = new int[5];` // Crea un array amb 5 posicions (totes amb valor 0)
- Declaració amb mida i inicialització explícita  
`int[] edats = new int[3] { 20, 25, 30 };`
- Declaració amb inicialització implícita (sense mida)  
`string[] noms = { "Anna", "Iván", "Marcos" };`

## 2. Accés als elements

```
Console.WriteLine(noms[0]); // "Anna"  
noms[1] = "Jorge";          // Modifica el valor en la posició 1
```



# Arrays

## 3. Recorregut d'un array

```
foreach (string nom in noms)
{
    Console.WriteLine(nom);
}
```

## O bé amb un bucle tradicional:

```
for (int i = 0; i < noms.Length; i++)
{
    Console.WriteLine(noms[i]);
}
```

## 4. Arrays multidimensionals

### **Bidimensional (matriu)**

```
int[,] matriu = new int[2, 3]
{
    { 1, 2, 3 },
    { 4, 5, 6 }
};
Console.WriteLine(matriu[1, 2]); // 6
```

# Arrays

## ◆ 6. Propietats i mètodes útils

Propietat / Mètode	Descripció	Exemple
<code>.Length</code>	Retorna el nombre total d'elements	<code>numeros.Length</code>
<code>.Rank</code>	Retorna el nombre de dimensions	<code>matriu.Rank</code>
<code>Array.Sort()</code>	Ordena els elements	<code>Array.Sort(numeros)</code>
<code>Array.Reverse()</code>	Inverteix l'ordre	<code>Array.Reverse(numeros)</code>

# Exemple d'una funció que captura el clic d'un ratoli

```
private void picCola_Click(object sender, EventArgs e)
{
}
```

- picCola\_Click es el nom de la funció.
- Object sender arreplega el botó que genera l'esdeveniment.
- EventArgs arreplega información adicional sobre l'esdeveniment, si en hi ha, i la dextra en la variable e.
- Després podem assignar aquesta funció al botó que desitgem desde Visual Studio.

# Funcions amb String

- `cadena.ToLower();` //minúscules
- `cadena.ToUpper();` //majúscules
- `cadena.Trim();` // treu espais del principi i el final
- `cadena.ToCharArray();` //torna un `char[]`
- `cadena.Split(' ', ' ');` //crea un vector de strings sent aquest el separador
- `cadena.Replace ("a",("b"));` //canvia a per b
- ...

# Creació de classes

```
using System; // Importa el espacio de nombres System, que incluye clases para manejar la consola
class Persona
{
    // Propiedades de la clase Persona
    public string Nombre { get; set; } // Propiedad pública de tipo string para almacenar el nombre de la persona. Get para acceder a su contenido,
                                     set para establecerle contenido
    public int Edad { get; set; }    // Propiedad pública de tipo int para almacenar la edad de la persona

    // Constructor de la clase Persona
    public Persona(string nombre, int edad)
    {
        // Asigna los valores de los parámetros a las propiedades de la clase
        Nombre = nombre;
        Edad = edad;
    }

    // Método Saludar de la clase Persona
    public void Saludar()
    {
        Console.WriteLine($"Hola, me llamo {Nombre} y tengo {Edad} años.");
    }
}

// Clase principal Program que contiene el método Main
class Program
{
    // Método Main: punto de entrada del programa
    static void Main()
    {
        // Crea una instancia de la clase Persona, inicializando sus propiedades mediante el constructor
        Persona persona1 = new Persona("Ana", 30);

        // Llama al método Saludar de la instancia persona1
        persona1.Saludar(); // Imprime el saludo en la consola
    }
}
```

# Herencia en C#

```
using System;
```

```
// Clase base
```

```
class Animal
{
    public string Nombre { get; set; }
    public int Edad { get; set; }

    public void Describir()
    {
        Console.WriteLine($"Soy un animal llamado {Nombre} y tengo {Edad} años.");
    }
}
```

```
// Clase derivada
```

```
class Perro : Animal
{
    public void Ladrar()
    {
        Console.WriteLine("Guau guau!");
    }
}
```

```
class Program
```

```
{
    static void Main()
    {
        // Crear una instancia de la clase Perro
        Perro miPerro = new Perro();

        // Usar las propiedades heredadas de Animal
        miPerro.Nombre = "Fido";
        miPerro.Edad = 3;

        // Llamar al método heredado de Animal
        miPerro.Describir();

        // Llamar al método específico de Perro
        miPerro.Ladrar();
    }
}
```

# Public void, Static void

Característica	public void	static void
Acceso	Público, accesible desde otras clases	Público, accesible desde otras clases
Instancia requerida	Sí, necesita una instancia de la clase	No, se puede llamar directamente desde la clase
Alcance	Asociado a una instancia específica	Asociado a la clase en sí misma

## Exemple public void

```
class MiClase
{
    // Método público, no estático
    public void MetodoPublico()
    {
        Console.WriteLine("Este es un método público.");
    }
}

class Program
{
    static void Main()
    {
        // Crear una instancia de MiClase
        MiClase instancia = new MiClase();

        // Llamar al método público a través de la instancia
        instancia.MetodoPublico();
    }
}
```

## Exemple static void

```
class MiClase
{
    // Método público y estático
    public static void MetodoEstatico()
    {
        Console.WriteLine("Este es un método estático.");
    }
}

class Program
{
    static void Main()
    {
        // Llamar al método estático directamente desde la clase
        MiClase.MetodoEstatico();
    }
}
```

# Conversió entre '9' (char) i 9 (int)

## ✅ Opción 1: restando '0'

Es la forma más simple y eficiente:

csharp

```
char c = '9';  
int numero = c - '0'; // 57 - 48 = 9  
Console.WriteLine(numero); // 9
```

## ✅ Opción 2: usando char.GetNumericValue()

csharp

```
char c = '9';  
int numero = (int)char.GetNumericValue(c);  
Console.WriteLine(numero); // 9
```

## ✅ Opción 3: usando int.Parse()

Si tienes un **carácter** o **cadena de un solo carácter**, también puedes convertirlo así:

csharp

```
char c = '9';  
int numero = int.Parse(c.ToString());  
Console.WriteLine(numero); // 9
```



# De int a char

## ✓ Opción 1: sumando '0'

csharp

```
int numero = 9;
char c = (char)(numero + '0');
Console.WriteLine(c); // '9'
```

## ✓ Opción 2: usando ToString()

Si quieres pasarlo a texto:

csharp

```
int numero = 9;
string s = numero.ToString(); // "9"
char c = s[0];                // '9'
```

Gràcies per la vostra atenció