

# Introduzione a C#

**C#**, si legge «c sharp», può essere considerato il linguaggio di programmazione per eccellenza del **Framework .NET**: diversamente dagli altri linguaggi, come Visual Basic o C++, esso è nato espressamente per la nuova piattaforma. In questo senso, è significativo il fatto che Microsoft stessa si sia servita di C# per scrivere gran parte delle librerie di .NET .

Uno degli slogan che hanno accompagnato C# fin dalla sua nascita lo presenta come «**un linguaggio facile come Java, potente come il C++**». In effetti, come vedremo, le **somiglianze** con i due linguaggi sono molte, e non solo dal punto di vista della sintassi.

Il corso **si rivolge** a chi ha già dimestichezza con la programmazione ad oggetti e con linguaggi simili a Java o C++.

Sono volutamente tralasciate le spiegazioni sulla **sintassi di base**: per esempio i costrutti if, switch, for, while, etc. Chi già conosce linguaggi come c++, Java, php o Javascript non vedrà grosse novità.

Il **.NET framework SDK**, ovvero l'insieme di tutti gli strumenti necessari alla creazione di software per la piattaforma .NET, è liberamente **scaricabile** dal sito Microsoft.

Dunque è sufficiente **usare un editor di testi** per sviluppare applicazioni .NET, anche se quando ci si orientasse verso programmi più complessi e dotati di interfaccia grafica, l'adozione di un ambiente di sviluppo visuale diventa quasi indispensabile (SharpDevelop o Visual Studio).

Realizziamo subito il nostro **primo programma**. Costruiremo un'applicazione "console", ovvero eseguibile dal prompt dei comandi. Per scrivere un programma di questo tipo non è strettamente necessario un **ambiente di sviluppo**, data l'assenza di elementi grafici o di connessioni complesse tra classi. Pensiamo di voler realizzare la classica applicazione che stampa a video la stringa "Ciao Mondo!", aspetta che l'utente preme il tasto INVIO e, quindi, termina.

Il codice è il seguente:

```
namespace HelloWorld
{
    public class Program
    {
        public static void Main(string[] args)
        {
            System.Console.WriteLine("Hello World!");

            System.Console.Write("Premi un tasto per uscire.. ");
            System.Console.ReadKey(true);
        }
    }
}
```

Chi ha già una conoscenza di Java potrà notare una certa somiglianza, dal punto di vista della sintassi, tra i due linguaggi. Per eseguire il programma, salviamolo in un file di testo con nome, ad esempio, **Program.cs**. L'estensione **.cs** contraddistingue i file sorgenti di C#.

L'espressione **class Program** serve per definire una classe di nome **Program** all'interno del **Namespace HelloWorld**.

Quello di **classe** è un concetto fondamentale, che approfondiremo nel corso delle prossime lezioni, al momento basti sapere che tutto il codice di un programma C# definisce classi e scambi di informazioni tra classi.

**Main** è un metodo particolare. Il cosiddetto punto di ingresso dell'applicazione, ovvero il primo metodo che viene richiamato quando si avvia il programma.

Tutti i programmi eseguibili realizzati con C#, compresi quelli per Windows, devono avere una (e solo una) classe con all'interno un metodo main, altrimenti in fase di esecuzione si otterrà un messaggio di errore.

L'**istruzione System.Console.WriteLine()** stampa a video il messaggio, mentre **System.Console.ReadKey(true)** fa sì che l'applicazione aspetti che l'utente preme un tasto prima di terminare.

Abbiamo così realizzato un semplice programma in C#, per prendere subito confidenza con i primi strumenti e per stimolare la nostra curiosità su alcuni concetti che saranno trattati con maggior dettaglio nelle prossime lezioni.

I **programmi** C# possono essere costituiti da uno o più file. Ciascun file può non contenere o contenere più **Namespace**.

Un *Namespace* può contenere tipi, quali *classi*, *struct*, *interfacce*, *enumerazioni* e *delegati*.

Di seguito viene illustrata la struttura di base di un programma C# che contiene tutti questi elementi.

```
// Struttura generale di un programma C#

using System;

namespace YourNamespace
{
    class YourClass
    {
    }

    struct YourStruct
    {
    }

    interface IYourInterface
    {
    }

    delegate int YourDelegate();

    enum YourEnum
    {
    }

    namespace YourNestedNamespace
    {
        struct YourStruct
        {
        }
    }

    class YourMainClass
    {
        static void Main(string[] args)
        {

            //Your program starts here...

        }
    }
}
```