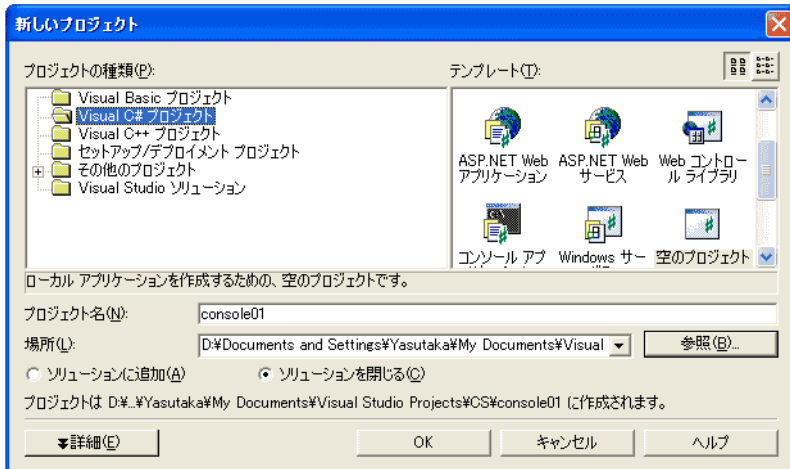


第1章 最初の1歩



この章ではC#で簡単なコンソール・アプリケーションを作ります。難しい理屈は抜きにして、とにかく「動く」プログラムを作ります。

まず、VS.netの設定です。

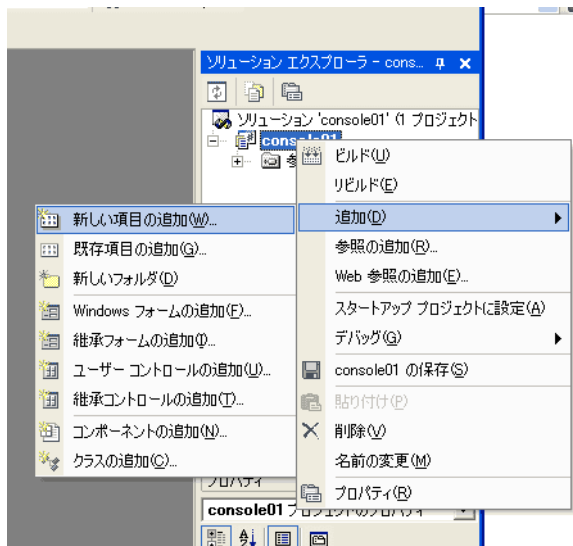


メニューの「ファイル」「新規作成」「プロジェクト」を選択すると左のようなダイアログボックスが出現します。

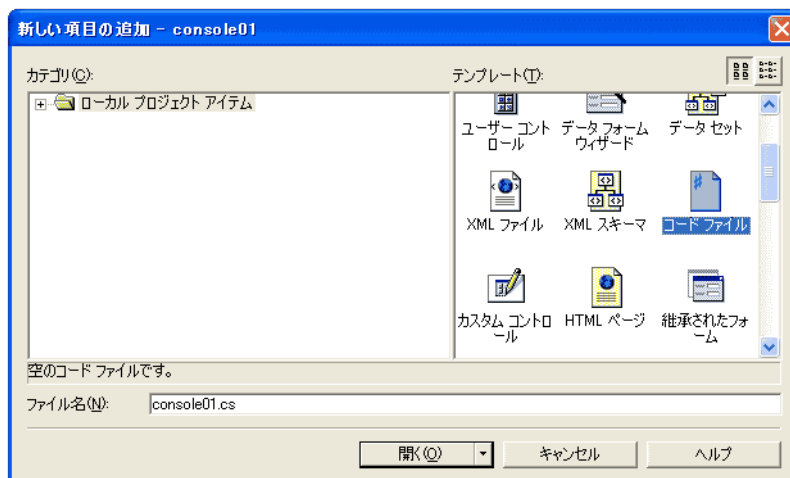
プロジェクトの種類は「Visual C#プロジェクト」を選択します。テンプレートは「空のプロジェクト」を選択します。ここで、「コンソール・アプリケーション」を選択してもよいのですが、そうするとソースが、ちょっと変わってきます。ソースをこのHPと全く同じにしたいのであれば、「空のプロジェクト」にしてください。

プロジェクト名はここでは「console01」としておきました。場所はそれぞれの環境でわかりやすいところにおいてください。

必要事項を記入したら「OK」ボタンを押します。



ソリューションエクスプローラの「console01」を右クリックして「追加」「新しい項目の追加」を選択します。



「新しい項目」として「コードファイル」を選択します。

ファイル名は「console01.cs」とします。C#では、ソースファイルの拡張子は「cs」となります。

さて、「開く」ボタンを押すとコードがすぐに入力できる状態となっていますので、次のようにタイプしてください。

```
// console01.cs

using System;

class console01
```

```

{
    public static int Main()
    {
        Console.WriteLine("猫でもわかるプログラミング!");

        return 0;
    }
}

```

あとは、メニューの「ビルド」「console01のビルド」を選択すれば、できあがりです。デバッガを使わないのであれば、「ビルド」「構成マネージャ」を選択して「アクティブソリューション構成」を「Release」にしてください。

「ビルド」や「アクティブソリューション構成」の変更はツールバーからの方が簡単です。

さて、ソースを見てわかることは、C言語のmain関数に相当するものがMainと大文字になっていることです。また、Main関数(Mainメソッドと言った方が正しい)自体がクラス(後の章で解説)の中にあることです(Mainメソッドですからクラスの中にあるのが当たり前!)。C#ではすべてのコードがどこかのクラスに属していないといけません。

また、C#では#includeは使いません。最初にusing Systemとあるのは、Systemという名前空間(後の章で解説)を参照しなさいという意味です。

このSystemには、Consoleクラスが含まれています。ConsoleクラスにはWriteLineというメソッドが含まれています。

ConsoleクラスのWriteLineメソッドを使う時はConsole.WriteLine(...);のように「.」で結びます(このピリオドのことを「分割子」といいます)。本来、System名前空間のConsoleクラスのWriteLineメソッドなので

System.Console.WriteLine(...);

と書くべきなのですが、using Systemとあるので、Systemを省略できるのです。

また、お気づきのようにコメントは「//」が使えます。これよりこの行の最後までがコメントとして扱われます。

また、「/* */」も使えます。これは、「/*」と「*/」の間がコメントとなります。複数行にまたがることも可能です。

では、次の章から少しずつC#の秘密(?)を覗いていくことにします。

[\[C# Index\]](#) [\[総合Index\]](#) [\[Next Chapter\]](#)

Update 12/Aug/2002 By Y.Kumei

当ホームページの一部または全部を無断で複写、複製、転載あるいはコンピュータ等のファイルに保存することを禁じます。