

# テキストファイルを読み書きしてみよう

## ● テキストファイルからデータを読む方法

入門編第4号まででプログラムにデータを入力する方法として、Console.ReadLineを使った入力や、テキストボックスを使った入力がありましたが、入門編第5号では**テキストファイル**からデータを読み込んでみます。

**テキストファイル**を読み込むには**StreamReader**を使用します。ですが、この**StreamReader**を使うには準備が必要です。

```

1 using System;
2 using System.Collections.Generic;
3 using System.Linq;
4 using System.Text;
5 using System.Threading.Tasks;
6 using System.IO;
7
8 namespace テキストファイルの読み込み方法
9 {
10     class Program
11     {
12         static void Main(string[] args)
13         {
14         }
15     }
16 }
17
    
```

using System.IO;を追加します

準備ができれば、**StreamReader**を使ってみましょう。

```

static void Main(string[] args)
{
    // StreamReaderオブジェクトの宣言とファイルオープン
    StreamReader sr = new StreamReader(@"c:\greeting.txt", Encoding.Default);

    // string型変数の宣言と読み込んだ1行分のデータの代入
    string greeting = sr.ReadLine();

    // ファイルクローズ
    sr.Close();

    // greetingの内容を画面に表示
    Console.WriteLine(greeting);

    // Enterキーが押されるまで待機
    Console.ReadLine();
}
    
```

テキストファイル名を指定

「@」を付けると「¥」は特殊記号扱いではなくなります

文字コードを指定 (コラム参照)

ReadLine は 1 行分のデータを読み込む命令

Close は**テキストファイル**を閉じる命令 (必須)

プログラムが入力できたら、指定した箇所にテキストファイルをおいて実行してみます。

読み込む**テキストファイル**



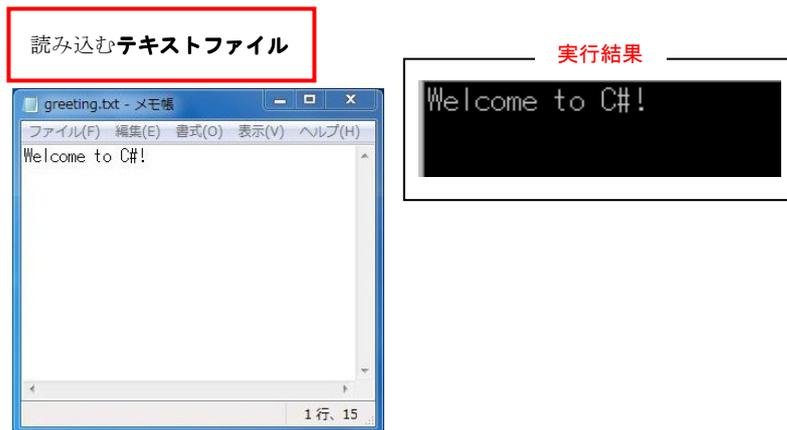
**テキストファイル**の場所



実行結果



今度は**テキストファイル**の内容を変更して再度実行してみます。



このように**テキストファイル**の記述を変更するだけでプログラムの結果が変更されました。

### ✍ 文字コード

コンピュータ上で文字を利用するために各文字に対して数値が割り当てられています。これを文字コードと言います。文字コードにはいろいろなコードがあり、コードが異なると文字が正しく表示されない現象が発生します。これが文字化けの正体です。

**StreamReader**で指定している「Encoding.Default」はOSで指定されている文字コードを使うという意味になります。

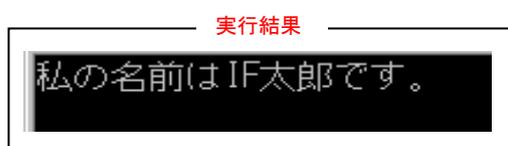
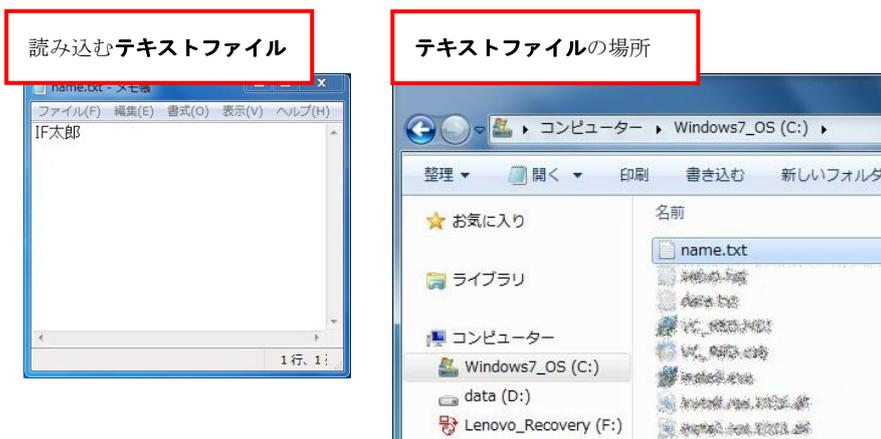
### 🖥 練習1

以下のようなプロジェクトを作成してください。

ソリューション名：テキストファイルの読み込み  
プロジェクト名：テキストファイルの読み込み  
プロジェクトテンプレート：コンソール アプリケーション

名前が記述されているテキストファイルを読み込んで表示するプログラムを作成してください。

読み込むテキストファイルの内容



● 複数行のデータを読み込む方法

テキストファイルに複数行のデータが入っていた場合は繰り返し処理と組み合わせてデータを読み込みます。では、値段の一覧が記述されているテキストファイルを読み込んで合計を計算するプログラムを作成してみましょう。

```

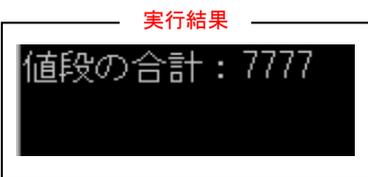
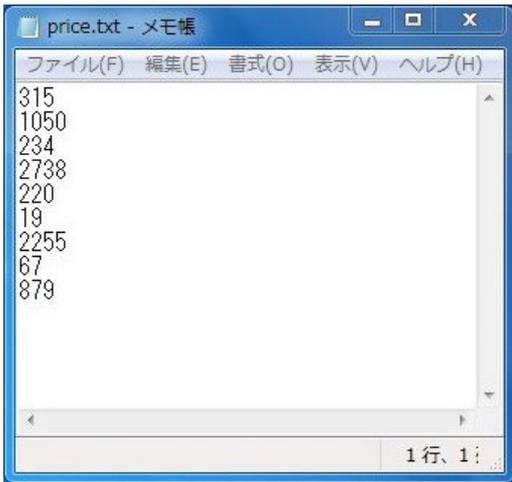
6  using System.IO;
7
8  namespace 複数行のデータを読み込む
9  {
10     class Program
11     {
12         static void Main(string[] args)
13         {
14             // StreamReaderオブジェクトの宣言とファイルオープン
15             StreamReader sr = new StreamReader(@"c:\price.txt", Encoding.Default);
16
17             // int型変数の宣言と初期化
18             int total = 0;
19
20             // データを読み終わるまで処理を繰り返す
21             while (-1 < sr.Peek())
22             {
23                 // string型変数の宣言とデータの代入
24                 string price = sr.ReadLine();
25
26                 // string型変数をint型に変換して合計を計算
27                 total += int.Parse(price);
28             }
29
30             // ファイルクローズ
31             sr.Close();
32
33             // totalの内容を画面に表示
34             Console.WriteLine("値段の合計:" + total.ToString());
35
36             // Enterキーが押されるまで待機
37             Console.ReadLine();
38         }
39     }
40 }

```

Peek は次の読み込むデータを取得してくれます  
取得したデータが-1 の場合は読み込めないこと  
を意味します

Close はテキストファイルを閉じる命令 (必須)

読み込むテキストファイル



このように繰り返し処理とテキストファイルの読み込みを組み合わせることで、大量のデータを扱うことができます。