

Les entrées sorties en C#

StreamWriter et StreamReader

Pour lire ou écrire dans un fichier vous disposez des deux fonctions suivantes : StreamWriter et StreamReader

Avant tous, il faut insérer la bibliothèque `System.IO` au début du programme : `using System.IO;`

1. StreamWriter

StreamWriter : C'est une fonction interne du langage C# qui vous permet d'écrire dans un fichier :

Syntaxe et déclaration:

```
StreamWriter fOut = new StreamWriter("E:\\test\\test.txt");
```

Dans cet exemple fOut est le nom de l'objet qui représente le fichier test.txt sur lequel vous allez écrire. L'emplacement du fichier est dans le répertoire: E:\test\

Pour cette exemple si le fichier test.txt existe son contenu sera écrasé et remplie de nouveau sinon il sera créé.

Pour écrire dans le fichier vous devez utiliser la fonction WriteLine() que vous connaissez déjà, par exemple, pour écrire "allo" dans votre fichier vous devez écrire ce qui suit :

```
fOut.WriteLine("allo").
```

Si initialement on ne veut pas écraser le contenu du fichier et on veut en ajouter, on doit déclarer l'objet fOut comme suit :

```
StreamWriter fOut = new StreamWriter("E:\\test\\test.txt", true);
```

Dans ce cas le contenu du fichier sera préservé et tous les ajouts vont se faire au-dessous du contenu initial.

À la fin de votre programme vous devez tout le temps appeler la fonction `close ()` pour fermer votre fichier

Pour notre exemple vous devez ajouter ceci à la fin de votre programme : `fOut.Close();`

420-121-LG : Compléments de programmation

Exemple copié de la page web de mon collègue Patrice Roy ([lien¹](#))

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.IO;
namespace ÉcrireCossins
{
    class Program
    {
        static void Main(string[] args)
        {
            StreamWriter fOut = new StreamWriter("../..../nombres.txt");
            int [] vals = { 2,3,5,7,11 };
            for (int i = 0; i < vals.Length; ++i)
            {
                fOut.WriteLine(vals[i]);
            }
            fOut.Close();
        }
    }
}
```

2. StreamReader

StreamReader: C'est une fonction interne du langage C# qui vous permet de lire le contenu d'un fichier :

Syntaxe et déclaration:

```
StreamReader fIn = new StreamReader("E:\\test\\test.txt");
```

Dans cet exemple fIn est le nom de l'objet qui représente le fichier test.txt que vous allez lire et qui se trouve à dans le répertoire: E:\test\

Pour cet exemple si le fichier test.txt n'existe pas une exception sera levée.

Pour Lire du fichier vous devez utiliser la fonction ReadLine() que vous connaissez déjà, par exemple, pour lire la première ligne du fichier vous devez écrire ceci : fIn.ReadLine()

À chaque appel de cette fonction un saut de ligne se fait automatiquement dans votre fichier de lecture.

Pour vérifier si vous avez atteint la fin du fichier, vous pouvez utiliser l'attribut booléen EndOfStream.

¹ <http://h-deb.clg.qc.ca/Sources/Cdiese--EntreesSortiesTexte.html>

420-121-LG : Compléments de programmation

À la fin de votre programme vous devez tout le temps appeler la fonction `close ()` pour fermer votre fichier comme dans le cas de l'écriture.

Exemple copié de la page web de mon collègue Patrice Roy ([lien](#)²)

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.IO;
namespace LireCossins
{
    class Program
    {
        static void Main(string[] args)
        {
            StreamReader fIn = new StreamReader("../.../.../.../nombres.txt");
            int nbLu;
            while (!fIn.EndOfStream)
            {
                nbLu = int.Parse(fIn.ReadLine());
                Console.WriteLine("J'ai lu le nombre {0}", nbLu);
            }
            fIn.Close();
        }
    }
}
```

² <http://h-deb.clg.qc.ca/Sources/Cdiese--EntreesSortiesTexte.html>