

# File management

## Writing to a text file

Create a program to ask the user for several sentences (until they just press Enter) and store them in a text file named "sentences.txt"

```
class WritingFile
{
    static void Main()
    {
        try
        {
            StreamWriter file = File.CreateText("test.dat");
            string line;

            do
            {
                Console.Write("Enter a sentence: ");
                line = Console.ReadLine();

                if (line.Length != 0)
                {
                    file.WriteLine(line);
                }
            }
            while(line.Length != 0);

            file.Close();
        }
        catch (Exception)
        {
            Console.WriteLine("Error!!!");
        }
    }
}
```

## Appending to a text file

Create a program to ask the user for several sentences (until they just press Enter) and store them in a text file named "sentences.txt". If the file exists, the new content must be appended to its end.

```
class AppendingTextFile
{
    static void Main()
    {
        try
        {
            StreamWriter file = File.AppendText("data.dat");
            string line;

            do
            {
                Console.Write("Enter a sentence: ");
                line = Console.ReadLine();

                if (line != "")
                    file.WriteLine(line);
            }
            while(line != "");

            file.Close();
        }
        catch (Exception)
        {
            Console.WriteLine("Error!!!");
        }
    }
}
```

## Display file contents

Create a program to display all the contents of a text file on screen (note: you must use a StreamReader). The name of the file will be entered in the command line or (if there is no command line present) asked to the user by the program.

```
class DisplayFileContent
{
    static void Main()
    {
        Console.Write("Enter name of file: ");
        string name = Console.ReadLine();

        try
        {
            StreamReader file; = File.OpenText(name);
            string line=" ";

            do
            {
                line = file.ReadLine();
                if (line != null)
                {
                    Console.WriteLine(line);
                }
            }
            while (line != null);
        }
        catch (Exception e)
        {
            Console.WriteLine("Error!!!");
        }
    }
}
```

## Extended TextToHTML (files)

Expand the TextToHtml class, so that it can dump its result to a text file. Create a method `ToFile`, which will receive the name of the file as a parameter.

Hint: You must use a "StreamWriter"

```
class TextToHTML
{
    private string[] html;
    private int lines;
    private int count;

    public TextToHTML()
    {
        count = 0;
        lines = 1000;

        html = new string[lines];
    }

    public void ToFile(string nameFile)
    {
        try
        {
            StreamWriter file = File.CreateText(nameFile);
            file.WriteLine( ToString() );
            file.Close();
        }
        catch (Exception e)
        {
            Console.WriteLine("Error!!!");
        }
    }

    public void Add(string line)
    {
        if (count < lines)
        {
            html[count] = line;
            count++;
        }
    }

    public string ToString()
    {

```

```
string textHtml;

textHtml = "\n";
textHtml += "\n";

for (int i = 0; i < count; i++)
{
    textHtml += "

";

    textHtml += html[i];

    textHtml += "

\n";
}

textHtml += "\n";
textHtml += "\n";

return textHtml;
}

public void Display()
{
    Console.Write( ToString() );
}

}

class Test
{
    static void Main()
    {
        TextToHTML textToHTML = new TextToHTML();
        textToHTML.Add("Hello");
        textToHTML.Add("How are you?");

        textToHTML.Display();

        textToHTML.ToFile("prueba.html");
    }
}
```

## TXT to HTML translator

Create a "Text to HTML converter", which will read a source text file and create a HTML file from its contents. For example, if the file contains:

Hola  
Soy yo  
Ya he terminado

The resulting HTML file should contain

Hola  
  
Soy yo  
  
Ya he terminado

The name of the destination file must be the same as the source file, but with ".html" extension (which will replace the original ".txt" extension, if it exists). The "title" in the "head" must be taken from the file name.

```
class TXTtoHTML
{
    static void Main()
    {
        Console.Write("Enter name of file: ");
        string nameTXT = Console.ReadLine();
        string nameHTML = nameTXT.Substring(0, nameTXT.Length - 4);

        if (File.Exists(nameTXT))
        {
            StreamReader fileTXT = File.OpenText(nameTXT);
            StreamWriter fileHTML = File.CreateText(nameHTML + ".html");

            fileHTML.WriteLine("");
            fileHTML.WriteLine("");
            fileHTML.WriteLine("");
            fileHTML.WriteLine("");
            fileHTML.WriteLine("");
        }
    }
}
```

```
string line;
do
{
line = fileTXT.ReadLine();
if (line != null)
fileHTML.WriteLine("

" + line + "

");
}
while (line != null);

fileHTML.WriteLine("");
fileHTML.WriteLine("");

fileTXT.Close();
fileHTML.Close();
}
}
```