

Läsa och skriva filer i C#



File

- File är en klass med statiska metoder för att...
 - ... kontrollera ifall en fil existerar eller inte.
 - ... skapa filer.
 - ... läsa filer.
 - ... ändra filer.

```
File.Exists(path)
```

Path

- Path är en klass som innehåller statiska metoder för att...
 - ... bygga ihop sökvägar.
 - ... hämta olika delar av en sökväg, tex filtyper, filnamn, mapp osv.

```
Path.Combine("C:/", "test.txt");
```

Environment

- Environment är en klass som innehåller konstanter med...
 - ... information om systemet (OS, aktiv användare på datorn osv)
 - ... sökvägar till standardiserade foldrar (MyDocuments, AppData, Desktop osv)

```
Environment.UserName
```

StreamWriter

- StreamWriter är en klass som kan användas för att skriva till filer.
- StreamWriter skriver löpande med hjälp av:
 - Write()
 - WriteLine()
- När StreamWriter instansieras skickas det önskade målet med. Text En sökväg till en fil.
- En StreamWriter måste stängas med Close() när man är färdig med skrivandet.

StreamReader

- StreamReader är en klass som kan användas för att läsa från filer.
- StreamReader skriver löpande med hjälp av:
 - Read()
 - ReadLine()
- När StreamReader instansieras skickas det önskade målet med. Tex En sökväg till en fil.
- En StreamReader måste stängas med Close() när man är färdig med skrivandet.

using statement

- Ett using statement i löpande kod kan användas för att begränsa livslängden på ett objekt.

- Ex:

```
using (StreamWriter sw = new StreamWriter(path))  
{  
};
```

- Här görs sw.Close() vid slutet av using.
- Om man inte kapslar in koden kommer objektet leva till nästa avslutande måsvinge ('}').

```
using StreamWriter sw = new StreamWriter(path);  
}  
using var sw
```

Demo

- Repo för Demo: <https://github.com/niklas-hjelm/ReadWriteFileDemo>

