

- Namespace: System.IO
- Ähnlich wie Console
 - Zusätzliche Angabe eines Dateipfades
- Einfacher Umgang mit Text- oder Binärdateien
 - File.ReadAllLines
 - File.ReadAllText
 - File.WriteAllLines
 - File.WriteAllText
 - File.Exists

ReadAllText: Text aus einer Datei lesen

Klasse



Die Datei, die
gelesen werden soll



```
string content = File.ReadAllText("MyText.txt", Encoding.Default);
```

Methoden
Name



Richtige Kodierung
von Umlauten und
Sonderzeichen



Rückgabewert!



Öffnet eine Textdatei,
liest den gesamten Text
in einen String und
schließt dann die Datei.

Beispiel: ReadAllText

Maja.txt

```
1 In einem unbekannten Land
2 Vor gar nicht allzu langer Zeit
3 War eine Biene sehr bekannt
4 Von der sprach alles weit und breit
5
6 Und diese Biene die ich meine nennt sich Maja
7 Kleine, freche, schlaue Biene Maja
```

```
string content = File.ReadAllText("Maja.txt", Encoding.Default);
```

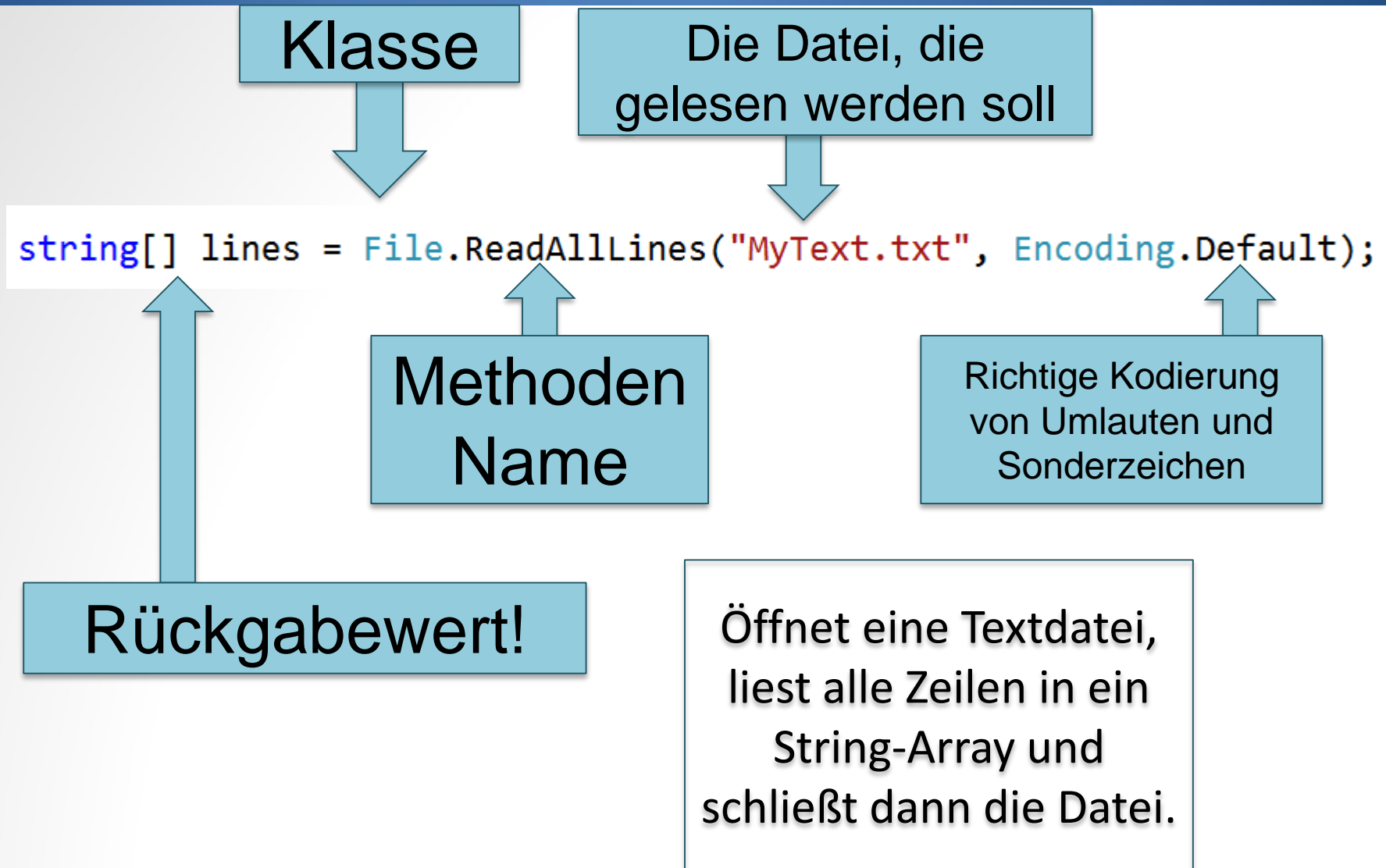
content "In einem unbekannten Land\r\nVor gar nicht allzu langer Zeit\r\nWar eine Biene sehr bekannt\r\nVon der sprach alles weit und breit\r\n\r\nUnd die

Vergleiche – Anzeige aller Zeichen:

Maja.txt

```
1 In einem unbekannten LandCRLF
2 Vor gar nicht allzu langer ZeitCRLF
3 War eine Biene sehr bekanntCRLF
4 Von der sprach alles weit und breitCRLF
5 CRLF
6 Und diese Biene die ich meine nennt sich MajaCRLF
```

ReadAllLines: Alle Zeilen einer Datei lesen



Beispiel: ReadAllLines

```
string[] lines
```

[0]	"In einem unbekannten Land"
[1]	"Vor gar nicht allzu langer Zeit"
[2]	"War eine Biene sehr bekannt"
[3]	"Von der sprach alles weit und breit"
[4]	""
[5]	"Und diese Biene die ich meine nennt sich Maja"
[6]	"Kleine, freche, schlaue Biene Maja"
[7]	"Maja fliegt durch ihre Welt"
[8]	"Zeigt uns das was ihr gefällt"
[9]	"Wir treffen heute uns're Freundin Biene Maja"
[10]	"Diese kleine, freche Biene Maja"
[11]	"Maja, alle lieben Maja"
[12]	"Maja Maja"
[13]	"Maja Maja"
[14]	"Maja, erzähle uns von Dir"

```
ding.Default);
```

WriteAllText, WriteAllLines

- Überschreiben der Datei mit einem string:

```
File.WriteAllText("MyText.txt", content, Encoding.Default);
```

- Überschreiben der Datei mit einem string-Array:
 - Beispiel: Reversieren der Zeilen

```
for (int i = 0; i < lines.Length / 2; i++)  
{  
    string swap = lines[i];  
    lines[i] = lines[lines.Length - 1 - i];  
    lines[lines.Length - 1 - i] = swap;  
}  
File.WriteAllLines("MyText.txt", lines, Encoding.Default);
```

```
string[] lines = File.ReadAllLines(FILENAME, Encoding.Default);
```

- Default Encoding =
`System.Text.Encoding.WindowsCodePage`
- Allgemeinere Encodings:
 - `Encoding.ASCII`
 - `Encoding.Unicode`
 - `Encoding.UTF8`

- Wie kann ich Dateipfade im C#-Quelltext angeben? Der Backslash \ wird ja für Escape-Sequenzen verwendet ...
 - **Maskierung:** Hierzu muss dem Sonderzeichen ein Backslash vorangestellt werden

```
static string FILENAME =  
    "C:\\Users\\Birgit\\Desktop\\CSharpFileExample.txt";
```

- **Unterdrückung der Escapesequenzen** durch Voranstellen von „@“

```
static string FILENAME =  
    @"C:\Users\Birgit\Desktop\CSharpFileExample.txt";
```


Working Directory (Arbeitsverzeichnis)

- Working Directory = Arbeitsverzeichnis
 - Das Ausführungsverzeichnis der Anwendung (= dort wo die .exe-Datei liegt)
 - Wird nur ein Dateiname (ohne Pfad) angegeben, sucht Windows die Datei im Working Directory
 - Das working directory bei einer C#-Applikation, die aus Visual Studio gestartet wurde, ist by default „<solution>\bin\Debug“

```
// Aktuelles Arbeitsverzeichnis mit \bin\debug  
string dir = Directory.GetCurrentDirectory();
```

▲ SimpleFileExample

▲ bin

Debug

▷ obj

Properties

Directory

- Namespace: System.IO
- Einfacher Umgang mit Verzeichnissen

Methode	Beschreibung
CreateDirectory	Erzeugt ein Verzeichnis oder Unterverzeichnis mit dem übergebenen Namen.
Delete	Löscht ein Verzeichnis.
Exists	Überprüft, ob das angegebene Verzeichnis existiert.
GetDirectories	Liefert die Namen aller Unterverzeichnisse eines spezifizierten Ordners als string[] zurück.
GetFiles	Liefert alle Dateinamen eines spezifizierten Ordners als string[] zurück.
GetParent	Liefert den Namen des übergeordneten Verzeichnisses.
Move	Verschiebt ein Verzeichnis samt Dateien an eine neue Speicherlokalität.

Directory: Beispiele

```
//Erstellen eines Verzeichnisses:  
Directory.CreateDirectory("c:\\tmp\\test\\dotnet");  
  
if (Directory.Exists("c:\\tmp\\test\\dotnet"))  
{  
    Console.WriteLine("Es existiert!!");  
}  
  
Directory.Delete(@"c:\tmp\test\dotnet");  
  
string[] files = Directory.GetFiles("c:\\");
```

Fehler bei Filehandling → Exceptions

- Verschiedene Fehler können auftreten
 - Datei nicht vorhanden
 - Zugriff auf Datei nicht erlaubt
 - Datei bereits geöffnet

```
string fileName = dir + "\\nicht_vorhanden.csv";  
string[] zeilen = File.ReadAllLines(fileName);
```

