

Bits, Bytes, Daten, Dateien

Bei der Informationsverarbeitung mit Computern wird digital gearbeitet. Die kleinste Informationseinheit bzw. Zuordnungseinheit ist dabei das **Bit** (Kunstwort aus „Binary digit“). Es kann nur zwei Zustände annehmen. Diese beiden Zustände haben die Werte 1 (Strom fließt) und 0 (Strom fließt nicht). In anderen Kontexten werden sie häufig auch anders benannt.

Im Computer werden selten einzelne Bits verarbeitet. Die kleinste Informationseinheit, auf die ein Rechner leicht zugreifen kann, ist ein **Byte**: Ein Byte ist eine geordnete Folge von 8 Bit. Ein Byte lässt sich als eine geordnete Folge von 8 „0/1-Zuständen“ darstellen, z. B.:

"1001 1101". Das Leerzeichen in der Mitte soll nur die Lesbarkeit erhöhen. Da jedes Bit 2 verschiedene Zustände annehmen kann, kann ein Byte 2^8 , also 256 verschiedene Zustände annehmen.

Interpretiert man ein Byte als die Darstellung einer natürlichen Zahl im **Dual-System** (2er-System, Stellenwertsystem zur Basis 2, Binärsystem), dann lassen sich in einem Byte die 256 Zahlen von 0 bis $2^8 - 1$, also von 0 bis 255 darstellen. Das Byte "1001 1101" könnte also die natürliche Zahl

$$1001\ 1101_2 = 1 \cdot 2^7 + 0 \cdot 2^6 + 0 \cdot 2^5 + 1 \cdot 2^4 + 1 \cdot 2^3 + 0 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0 = 128 + 16 + 8 + 4 + 1 = 157$$

darstellen.

Ein Byte kann aber auch als Zeichen interpretiert werden. Dabei verwendet man so genannte Kodierungstabellen, z. B. ASCII, ANSI. Bei den beiden gerade genannten entspricht dem Zeichen A der Wert „0100 0001“. Außerdem verwendet man den so genannten Unicode, der aus mehreren Bytes besteht.

Die Darstellung im Zweiersystem ist aber recht lang und unübersichtlich. Deshalb verwendet man in der Informatik das 16er-System, auch **Hexadezimal-System** oder kurz **Hex-System** genannt:

$$22_{16} = 2 \cdot 16^1 + 2 \cdot 16^0 = 32 + 2 = 34$$
$$0100\ 0001_2 = 41_{16} = 65$$

Zur Darstellung der Stellenwerte größer als 9 brauchen wir im Hex-System neue Ziffern. Dazu verwendet man die ersten Buchstaben des Alphabets. Im Hex-System kann also jede Stelle mit einer der Ziffern {0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F} belegt werden:

$$1001\ 1101_2 = 9D_{16} = 9 \cdot 16^1 + 13 \cdot 16^0 = 144 + 13 = 157$$

Computerprogramme speichern die **Daten**, die der Benutzer erzeugt, üblicherweise in **Dateien**. Dabei werden die verschiedenen Arten von Dateien durch Erweiterungen der Dateinamen voneinander unterschieden: Microsoft Word erzeugt normalerweise Dateien mit der Endung „.doc“. Ein Malprogramm schreibt hingegen Dateien mit der Endung „.bmp“ oder „.jpg“.

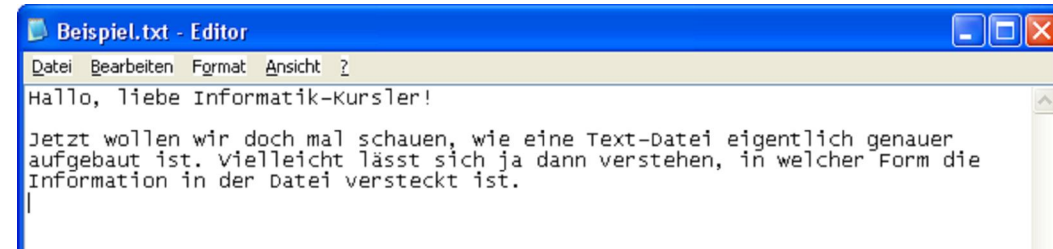
Rein technisch betrachtet sind **alle** Dateien nichts weiter als **Folgen von Byte-Werten**. Mit einem speziellen Programm, nämlich einem Hex-Viewer, kann man solche Byte-

Folgen auf dem Bildschirm darstellen. Dabei werden die gelesenen Bytes einfach nebeneinander geschrieben.

Der Unterschied liegt nur an der Interpretation der Daten. Ein bestimmtes Programm kann mit bestimmten Folgen von Byte-Werten etwas anfangen, andere dagegen nicht. Dabei gibt es zwei Grundtypen von Dateien: **Programm-Dateien** mit Prozessor-Befehlen und **Daten-Dateien** enthalten Benutzer-Informationen.

Aufgaben:

1. Wandle die Zahlen $1001\ 1001_2$, 55_{16} und $A9_{16}$ in die entsprechende Dezimalzahl um.
2. Rechne die Dezimalzahlen 170_{10} in die beiden anderen Zahlensysteme um.
3. Erstelle mit dem Editor (im Startmenü unter Alle Programme – Zubehör) den folgenden Text:



Speichere ihn als Beispiel.txt ab. Kopiere den Text in ein Word-Dokument und speichere das Word-Dokument als Beispiel.doc ab.

Die folgenden Aufgaben werden mit Hilfe des Programms Hex-Viewer bearbeitet, das sich im Tauschverzeichnis befindet. Bitte vor der Benutzung die Datei HexViewer.exe in Eigene Dateien kopieren. *Die Ergebnisse bitte aufschreiben!*

4. Untersuche die Dateien Beispiel.txt und Beispiel.doc. Vergleiche ihren Inhalt. Was fällt auf?
5. Untersuche einige Bilddateien, die Du auf dem PC findest. Was kannst du bei Ihnen beobachten?
6. Untersuche jetzt einige C# Programme, die Du bisher im Unterricht geschrieben hast. Was fällt am Anfang der Dateien auf? Wie steht es mit dem Rest der Datei?
7. Kopiere die Dateien StringTest.exe und IntText.exe aus dem Tauschverzeichnis in Eigene Dateien. Untersuche sie mit dem Hex-Viewer. Betrachte insbesondere den Bereich ab Adresse 1720. Was fällt auf?

Quelltext der Programme:

<pre>StringTest.exe: public static void Main(string[] args) { // String s definieren String s="Hallo Welt!"; // Ausgabe auf dem Bildschirm Console.WriteLine(s); // Programm anhalten, damit Ausgabe // lesbar bleibt Console.ReadLine(); }</pre>	<pre>IntTest.exe: public static void Main(string[] args) { // Ganze Zahl z definieren int z=5; // Ausgabe auf dem Bildschirm Console.WriteLine(z.ToString()); // Programm anhalten, damit Ausgabe lesbar // bleibt Console.ReadLine(); }</pre>
---	--