

Zeichendarstellung mit Unicode und UTF-8

ASCII - American Standard Code for Information Interchange



In der [Mittelstufe](#) wurde die Codierung von Text mithilfe des ASCII-Standards besprochen. Hierbei wird jedem Zeichen ein Wert zwischen 0 und 255 (8 Bit) zugewiesen. Oben siehst du die ASCII-Codetabelle, leere Zellen enthalten Steuerzeichen, welche für die Darstellung am PC nötig waren. Die wichtigsten Steuerzeichen sind in der Tabelle beschrieben.

In einem früheren, hauptsächlich in Amerika benutzten Standard waren lediglich die Zeichen von 0 bis 127 definiert, das letzte, achte Bit wurde zur Fehlerüberprüfung verwendet. Erst später wurde das 8. Bit dazu genommen, um weitere Zeichen, wie z.B. die deutschen Umlaute codieren zu können.



(A1)

Wandle die nachfolgenden Wörter, die in Hexadezimal-Darstellung vorliegen, in lesbaren Text um:

1. 49 6E 66 6F 72 6D 61 74 69 6B
2. 42 69 6E E4 72
3. 43 6F 6D 70 75 74 65 72

Mit einer 8-Bit-Codierung lassen sich nicht mehr Zeichen darstellen, was insbesondere bei anderen Sprachen – wie z.B. griechisch – andere Codierungen nötig machte. Da in diesen Sprachen jedoch die bei uns gebräuchlichen Umlaute nicht benötigt werden, wurde der durch das 8. Bit hinzugekommene Block vom Zeichen 128 bis 255 für die dortigen Zeichen verwendet. Diese und andere länderspezifischen Codierungen lassen sich z.B. unter https://de.wikipedia.org/wiki/ISO_8859 nachschauen.



(A2)

Welche der obigen Wörter würden mit den griechischen Zeichensatz falsch dargestellt werden und warum?

Unicode - UTF-8

Um Probleme, die sich zum einen mit unterschiedlichen Zeichensätzen, zum anderen auch durch andere Sprachen, die mehr als 128 Zeichen umfassen, ergeben haben, wurde der Unicode-Standard entwickelt. Hier kann ein einzelnes Zeichen in der UTF-8-Codierung bis zu 4 Bytes umfassen, nach folgenden Regeln:

- Ist das erste Bit eine 0, so besteht das Zeichen lediglich aus einem Byte, die 128 verbleibenden Möglichkeiten entsprechen dabei genau dem ASCII-Code.
- Ansonsten beginnt das erste Byte mit so vielen 1en, wie viele Bytes das Zeichen umfasst. Alle weiteren Bytes beginnen mit 10. Die Bits der Dualdarstellung des Unicodes müssen also in „6er-Paketen“ auf mehrerer Bytes verteilt werden, die jeweils mit 10 beginnen.

Beispiele:

$$y = 79_{16} = 0111\ 100_2$$

Beginnt mit einer Null und ist nicht länger als ein Byte → die letzten 7Bit werden verwendet, um zu codieren, also ein „ASCII k“ in UTF-8

UTF-8: 0110 1011

From:
<https://wiki.qg-moessingen.de/> - QG Wiki

Permanent link:
<https://wiki.qg-moessingen.de/faecher:informatik:oberstufe:codierung:utf8:start?rev=1634136068>

Last update: **13.10.2021 16:41**

