

# Lernziele zum Thema “Bits and Bytes”

## Überblick

- Stellenwertsysteme
- Speicherung/Kodierung von Zeichen (Morse-Code, ASCII, Unicode)
- Speicherplatzbedarf
- Logische Ausdrücke/Schaltungen
- disjunktive Normalform
- Bau eines 4-Bit-Addierers (Addition zweier 4-stelliger Binärzahlen) mit Logism

## Konkreter

- Zahlen zwischen verschiedenen Stellenwertsystemen umrechnen, insbesondere Binär-, Dezimalsystem, aber auch Fünfer- und Hexadezimalsystem
- Python-Notation für Binär- und Hexadezimalzahlen: `0b` bzw. `0x`
- „Primarschulstoff“ in diesen Systemen (also Zählen, Kleines Einsplus eins, Kleines Einmaleins, schriftliche Addition und Multiplikation)
- Kenntnis der logischen Verknüpfungen (AND, OR, NOT)
- Wahrheitstafeln zu logischen Ausdrücken angeben können
- logischen Ausdruck zu Wahrheitstafel angeben (disjunktive Normalform)
- logischen Ausdruck als “Logism”-Schaltung zeichnen und umgekehrt logischen Ausdruck (bzw. logische Ausdrücke bei mehreren Outputs) zu gegebener “Logism”-Schaltung angeben.
- Was macht ein Halbaddierer? Welche Wahrheitstabelle hat er?
- Was macht ein Volladdierer? Welche Wahrheitstabelle hat er?
- Was macht ein 4-Bit-Addierer? Wie lässt er sich aus Halb- und Volladdierern bauen?
- Morse-Code (falls dazu eine Aufgabe kommt, werde ich eine Tabelle bereitstellen)
- ASCII-Codierung, d. h. Umwandlung von ASCII-Code in Zeichen und umgekehrt (falls dazu eine Aufgabe kommt, werde ich eine Tabelle bereitstellen)
- NP-Klassen:
  - Caesar-Verschlüsselung kennen
  - Die folgenden Python-Programm so weit verstehen, dass ähnliche Programme verstanden werden: Konvertierungsfunktion in ein gewünschtes Stellenwertsystem, Caesar-Verschlüsselung. Voraussichtlich wird in der Prüfung ein ähnliches Python-Programm angegeben, dessen Ausgabe Sie angeben müssen.