

Informatik Begriffe

1. Informationen und Daten

Daten sind Informationen, die in eine Form umgewandelt wurden, die für einen Computer übersetzt wurden, welche effizienter von einem Computer kopiert/verarbeitet werden können.

Eine **Information** jedoch, ist eine für den Menschen verständliche Aussage

2. Basis und Alphabet eines Zahlensystems

Die **Basis** eines Zahlensystems ist bei z. B. beim Dezimalsystem 10, beim Binärsystem 2 (beim Hexadezimalsystem 16 usw.), wobei die Basis der Anzahl an verfügbaren Ziffern entspricht.

3. Umwandlung

1. Dezimal zu Binär:

Nimm bei deiner Beliebigen Zahl₁₀ die grösstmögliche Potenz von 2, also 1, 2, 4, 8, 16 usw. Sagen wir diese Zahl ist 57. Die grösste Potenz wäre demnach 32 (, da 64 grösser ist als 57). 32 ist = 2^5 , deshalb notieren wir 10000_2 . Als nächstes Subtrahieren wir 32 von 57, was 25 ergibt. Hier wäre die grösste Potenz 16, also 1000_2 . Also notieren wir 11000_2 . $25 - 16 = 9$ die kleinste Potenz wäre 8. Wiederhole diesen Schritt, bis du also keinen Resten mehr hast. Und falls du richtig gerechnet hast, solltest du jetzt die Binärzahl 111001_2 erhalten.

2. Binär zu Hexadezimal

Nehmen wir an, du hast eine Binärzahl 1001010001001110_2 . Diese teilst du jetzt in viergliedrige Pärchen auf, also 1001 0100 0100 1110. Das Hexadezimalsystem funktioniert wie folgt: wir zählen wie im Dezimalsystem von 0 bis 9, hängen aber anstatt 10 bis 15 ein A bis F an. Also 0, 1, 2, 3, 4, 5, ..., 9, A, B, C, D, E, F. Jetzt wandelst du jedes dieser viergliedrigen Teiler in eine Dezimalzahl um. 1001 wäre demnach 9, diese wandelst du in eine Hex-Zahl um, wobei du 9 bekommst, da 9 im Hex-System vertreten ist. Das letzte Glied, 1110 wäre aber 14, in Hex also E. Wenn du diesen Schritt für jedes Glied wiederholst und sie in richtiger Reihenfolge aufschreibst, solltest du $944E_{16}$ erhalten.

3. Dezimal zu Hexadezimal

Für diesen Prozess wandelst du diese Dezimalzahl zuerst in eine Binärzahl um (Schritt 1.) und danach diese Binärzahl in eine Hexadezimalzahl. Du kombinierst für diesen Schritt direkt beide der Vorherigen Operationen!

4. Bitbreite

Die **Bitbreite** beschreibt die Anzahl Bits, die beschrieben werden können. Beispiele sind 8-Bit, 32-Bit, 64-Bit usw. So kann man mit einem 32-Bit-Betriebssystem z. B. 2^{32} Bits, also 4'294'967'296 Bits bzw. 536'870'912 Bytes bzw. 4 Gigabytes (, da 1 Megabyte nicht genau 1000 Bytes sind, sondern 1024, wegen der gleichnamigen Potenz von 2 und 1GB 1024 Megabytes, d.h. 1'048'576 Bytes) darstellen.

ASCII z. B. braucht 8 Bits, da man mit einer 8-Bit-Bandbreite 256 (2^8) Zeichen belegen kann. Hätte man ein z. B. 5-Bit-System wäre es nur möglich 32 (2^5) Zeichen zu belegen.

5. Codierung, Decodierung, Alphabet

Mit der **Codierung** beschreibt man die Verschlüsselung bzw. Umwandlung von menschlich-lesbaren Informationen in Maschinen-«Code», also z. B. ins Binärsystem. Die **Decodierung** ist damit genau das Gegenteil. Das Alphabet ist der Schlüssel für die Codierung / Decodierung, ohne welchen diese nicht funktionieren würde.

6. ASCII codieren

Kein Begriff, lerne das am besten selbst.

7. ASCII decodieren

„

8. Bits / Bytes

Bytes bestehen aus 8 Bits, welche zwei Stadien einnehmen können, an (1) und aus (0). Ein Byte ist die kleinste Datenmenge.

9. Datentypen in Python

1. Integer

Ein Integer (bspw.: $a = 1$) ist eine Ganzzahl.

2. Float

Ein Float ist eine Kommazahl (bspw.: $b = 1.0$)

3. String

Ein String ist eine Zeichenkette (bspw.: $c = \text{"Hello World"}$).

10. Arithmetische Operationen in Python

1. Addition:

```
a = 4  
b = 7
```

```
console.log(a + b) => 11
```

2. Subtraktion:

```
console.log(a - b) => -3
```

3. Multiplikation:

```
console.log(a * b) => 28
```

4. Division

```
console.log(a / b) => 0.57
```

5. Rest / Modulo

```
console.log(a % b) => 3
```

6. Floor Division

```
console.log(a // b) => 0 (hier wird nicht gerundet, sondern die Nachkommastellen entfernt)
```

11. Daten einlesen zur Lösung eines Problems

Das kannst du am besten selbst machen, da das kein Begriff ist, der wirklich erklärbar ist.

12. Das EVA-Prinzip

EVA – kurz für **E**ingabe **V**erarbeitung **A**usgabe. In einem Eingabeteil werden die Benutzereingaben von entsprechenden Variablen erfasst. In dem darauffolgenden Verarbeitungsteil werden die von den Variablen verwalteten Daten weiterverarbeitet. In einem Ausgabeteil werden abschließend berechnete Ergebnisse auf dem Bildschirm ausgegeben.