

LF09:08:Exkurs: Bitweise Verknüpfungen bzw. bitweise und logische Operatoren

0) Exkurs: Bitweise Verknüpfung: Kontext [→ ZP:Sheet:2]

Anknüpfung: Was bedeutete & im Kontext einer **MAC-Adresse** und ihrem **OUI**?

1) Bitweise Operatoren [→ ZP:Sheet:3]

Syntax: **NOT** :- '!', **AND** :- '&', **OR** :- '|'

Abbildung: $\{0, 1\} \rightarrow \{0, 1\}$

	NOT [!]	OR []	AND [&]
R1	!1 → 0	1 1 → 1	1 & 1 → 1
R2	!0 → 1	1 0 → 1	1 & 0 → 0
R3		0 1 → 1	0 & 1 → 0
R4		0 0 → 0	0 & 0 → 0

Unterschied von Zahl und Darstellung [→ ZP:Sheet:4]

- Integer bleibt Integer - unabhängig von der Darstellung!
- Deshalb: Vor bitweiser Verknüpfung keine gesonderte Umwandlung 'dezimal → binär' nötig.

Vorführbeispiele:

- bitweises NICHT: ! (0111) → 1000
- bitweises ODER: | (0101, 0010) → 0111 (= lax gesagt: *bitweise verodern*)
- bitweises UND: & (0110, 1011) → 0010 (= *ausmaskieren* bzw. lax gesagt: *bitweise verundern*)

vgl. [https://de.wikipedia.org/wiki/Bitweiser_Operator][https://de.wikipedia.org/wiki/Bitweiser_Operator\(\)](https://de.wikipedia.org/wiki/Bitweiser_Operator)

3) Logische Operatoren in Computersprachen [→ ZP:Sheet:5]

Die Definitionen ähneln sich:

Syntax: **NOT** :- '!', **AND** :- '&&', **OR** :- '||' Abbildung: $\{T, F\} \rightarrow \{T, F\}$

	NOT [!]	OR []	AND [&&]
R1	$\neg T \rightarrow F$	$T \vee T \rightarrow T$	$T \wedge T \rightarrow T$
R2	$\neg F \rightarrow T$	$T \vee F \rightarrow T$	$T \wedge F \rightarrow F$
R3		$F \vee T \rightarrow T$	$F \wedge T \rightarrow F$
R4		$F \vee F \rightarrow F$	$F \wedge F \rightarrow F$

5) Logische Operatoren in der Logik [→ ZP:Sheet:6]

- werden mit Wahrheitswertetabellen definiert.
- mit 'ausmultiplizierten' Kombinationen

1	p	q	! p	! q	p q	p && q
2	---	---	---	---	---	---
3	w	w	f	f	w	w
4	w	f	f	w	w	f
5	f	w	w	f	w	f
6	f	f	w	w	f	f

Anmerkung:

- \vee steht für oder
- $\text{und} = (1w+3f)$ ist die strukturelle Opposition von $\text{oder} = (3w+1f)$
- darum ist das logische Symbol für und das umgekehrte \vee

Lesehinweis: die logischen Zeichen sind im Font dieses Textes nicht darstellbar, nur in der Zen-Präsentation, Sheet 6. Sorry.

6) Leseaufgaben für Programmierer:

ÜBUNG LF09:08:Bit-Operatoren:01

- ☐ Beschreiben Sie, was [→ ZP:Sheet:7] besagt.

Lösung:

(A) Was ist hier logische los? [→ ZP:Sheet:7]

```
1 # PYTHON-Example:
2 x=2
3 if x % 2 == 0 AND x % 3 == 0 :
4     print("durch 2 und 3 teilbar")
5 else:
6     print("nicht durch 2 und 3 teilbar")
```

```
1 /* C++ Example */
2 int x=2;
3 if ((x % 2) && (x % 3)) {
4     cout << "durch 2 und 3 teilbar";
5 }
6 else {
7     cout << "nicht durch 2 und 3 teilbar";
8 }
```

Lösung: Logische Verknüpfung zweier Aussagen

ÜBUNG LF09:08:Bit-Operatoren:02

- ☐ Beschreiben Sie, was [→ ZP:Sheet:8] besagt.

Lösung:

(B) Was ist hier logische los? [→ ZP:Sheet:8]

```
1 # PYTHON-Riddle:
2 x=2
3 if (((x % 2) AND ((4 & x) == 1)) :
4     print("logical revolution!")
5 else:
6     print("häh?")
```

Lösung: Unerfüllbare Bedingung

- $2 \% 2$ ist niemals positiv
 - $4 \& 2 == 0b0100 \& 0b0010 == 0b0000$ ist niemals 1
 - if-statement braucht logischer Verknüpfung wegen nur ersten Test auszurechnen
-

ÜBUNG LF09:08:Bit-Operatoren:03

- ☐ Bitte berechnen Sie $168 \& 128$.

☐ Bitte berechnen Sie $(2==2 \ \&\& \text{"Birne"}==\text{"Apfel"})$
