

LF09:08:Bitoperatoren für MAC- und IPv4-Adresssysteme in der Ausbildung zur Fachinformatikerin¹

Karsten Reincke

GS-LDK

7. Januar 2026

¹ Diese Präsentation stammt aus dem Open-Source-Projekt [proTironeComputatri](#), ist [CC-BY-4.0](#) lizenziert und wurde auf Basis von [proScientia.itx](#) entwickelt.

LF09:08:Exkurs:Bitweise Verknüpfung

```
1 # (C) 2025 K.Reincke: proTironeComputatri snippet [CC-BY-4.0]
2
3     0x48 0xe7 0xda 0x06 0xef 0x85    # MAC-Adresse
4 & 0xff 0xff 0xff 0x00 0x00 0x00    # OUI-Bitmask
5 -----
6 = 0x48 0xe7 0xda 0x00 0x00 0x00    # Organizationally Unique Identifier
```

LF09:08:Exkurs:Bitweise Operatoren

Bitweise Operatoren in Computersprachen

- sind ! , | und &
- sind ein- oder zweistellige Funktionen in der Menge {1, 0}
- bilden '{1,0}' bzw. '{1,0}x{1,0}' auf '{1,0}' ab

	<i>not = !</i>	<i>or = </i>	<i>and = &</i>
R1	$!1 \rightarrow 0$	$1 \mid 1 \rightarrow 1$	$1 \& 1 \rightarrow 1$
R2	$!0 \rightarrow 1$	$1 \mid 0 \rightarrow 1$	$1 \& 0 \rightarrow 0$
R3		$0 \mid 1 \rightarrow 1$	$0 \& 1 \rightarrow 0$
R4		$0 \mid 0 \rightarrow 0$	$0 \& 0 \rightarrow 0$

Bitweise Operatoren nehmen n Zahlen als Argumente und verknüpfen die darin gesetzten an derselben Position gesetzten Bits nach diesen Regeln

LF09:08:Exkurs:Bitweise Operatoren

```
1 # (C) 2025 K.Reincke: proTironeComputatri snippet [CC-BY-4.0]
2
3 theNumber = 42
4
5 print(theNumber, " : ",(bin)(theNumber), " : ",(hex)(theNumber), " : ")
6
7 # prints 42 in 3 modes of representation: "42 : 0b101010 : 0x2a"
```

Integer bleibt Integer - unabhängig von der Darstellung!

⇒ Vor bitweiser Verknüpfung
keine gesonderte Umwandlung
'dezimal → binär' nötig.

LF09:08:Exkurs:Logische Operatoren

Logische Operatoren in Computersprachen

- sind `!`, `||` und `&&`
- sind ein- oder zweistellige Funktionen in der Menge $\{T, F\}$
- bilden ' $\{T, F\}$ ' bzw. ' $\{T, F\} \times \{T, F\}$ ' auf ' $\{T, F\}$ ' ab

	$NOT = !$	$OR = $	$AND = &&$
R1	$!T \rightarrow F$	$T T \rightarrow T$	$T \&\& T \rightarrow T$
R2	$!F \rightarrow T$	$T F \rightarrow T$	$T \&\& F \rightarrow F$
R3		$F T \rightarrow T$	$F \&\& T \rightarrow F$
R4		$F F \rightarrow F$	$F \&\& F \rightarrow F$

Logische Operatoren nehmen n Wahrheitswerte als Argumente und verknüpfen diese nach obigen Regeln

LF09:08:Exkurs:Wahrheitswertetabellen

Logische Operatoren in der Logik sind

- sind \neg , \vee und \wedge
- bilden ' $\{w,f\}$ ' bzw. ' $\{w,f\} \times \{w,f\}$ ' auf ' $\{w,f\}$ ' ab

Die Definition erfolgt durch Wahrheitswertetabellen:

p	q	$\neg p$	$\neg q$	$p \vee q$	$p \wedge q$
w	w	f	f	w	w
w	f	f	w	w	f
f	w	w	f	w	f
f	f	w	w	f	f

Mit Wahrheitswertetabellen kann man weitere 'logische Operatoren' definieren, z.B. \rightarrow (wenn, dann)

LF09:08:Logische Operatoren:Was geht?

```
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 }
```

LF09:08:Bitweise + logische Operatoren:Was geht?

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

Herzlichen Dank für Ihre Aufmerksamkeit!



Diese Präsentation gehört zum Open-Source-Projekt *proTironeComputatri*², initiiert v. Karsten Reincke, Hohenahr³. Die Unterrichtseinheiten stehen unter den Bedingungen der CC-BY-4.0-Lizenz zur freien Verfügung. Die Quellen dazu finden Sie unter *protico.ltx*⁴.

² → <https://github.com/pro-tirone-computatri/>

³ → <https://github.com/pro-tirone-computatri/protico.ltx/CONTRIBUTORS.md>

⁴ → <https://github.com/pro-tirone-computatri/protico.ltx>

Bildnachweise

-  von Karsten Reincke. Lizenziert unter proTirone-Logo-License. Bereitgestellt auf github. (may only be used as logo for proTirone)