

LF09:09:IPv4-Adressen und -Netze für die Ausbildung zur Fachinformatikerin ¹

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.

	B4	B3	B2	B1
IPv4-String	"192.168.0.0"			
dezimal aufgelöst	192	168	0	0
hexadezimal aufgelöst	0xC2	0xA8	0x00	0x00
binär aufgelöst	11000000	10101000	00000000	00000000

B4	B3	B2	B1
2^8	2^8	2^8	2^8
$2^{(8+8+8+8)}$			
$= 2^{(32)}$			
$= 4.294.967.296$			

ergibt bei 8.000.000.000 Erdbewohnerinnen

0.53 IPv4-Adressen pro Person ;-)



- ① 19216800
- ② C2.A8.0.0
- ③ 192.168.320.0

	B4	B3	B2	B1
IPv4-Adresse	192	168	0	42
Subnetzmaske	255	255	255	0

(bin)(IPv4-Adresse)	11000000	10101000	00000000	00101010
& (bin)(Subnetzmaske)	11111111	11111111	11111111	00000000
= (bin)(Netzadresse)	11000000	10101000	00000000	00000000
Netzadresse	192	168	0	0

Broadcastadresse	192	168	0	255
Gatewayadresse	192	168	0	1

LF09:09:Exkurs Notationsumwandlung

dec \Rightarrow bin

0. 43 : 2 = 21 % 1 \uparrow
1. 21 : 2 = 10 % 1 \uparrow
2. 10 : 2 = 5 % 0 \uparrow
3. 5 : 2 = 2 % 1 \uparrow
4. 2 : 2 = 1 % 0 \uparrow
5. 1 : 2 = 0 % 1 \uparrow

7.	6.	5.	4.	3.	2.	1.	0.
0	0	1	0	1	0	1	1

bin \Rightarrow dec

7.	6.	5.	4.	3.	2.	1.	0.
0	0	1	0	1	0	1	1
$0 \cdot 2^7$	$0 \cdot 2^6$	$1 \cdot 2^5$	$0 \cdot 2^4$	$1 \cdot 2^3$	$0 \cdot 2^2$	$1 \cdot 2^1$	$1 \cdot 2^0$
		32	+	8	+	2	+ 1 = 43

Anregung: Warum wäre [45] ein erklärtechnisch schlechtes Beispiel, nicht aber die *Weltsuperzahl* 42?

	B4	B3	B2	B1
IPv4-Adresse	192	168	0	42
Subnetzmaske	255	255	255	0

- Subnetzmaske 'spaltet' IPv4-Adresse in
 - **Netzanteil** (hier 'B4.B3.B2') und
 - **Hostanteil** (hier 'B1')
- \Rightarrow
 - **Netzadresse** :- Netzanteil \circ kleinste Zahl im Hostanteil
 - **Broadcastadresse** :- Netzanteil \circ größte Zahl im Hostanteil
 - **Gatewayadresse** :- 'Netzadresse + 1' | 'Broadcastadresse -1' | ...



192.168.0.0/24

LF09:09:IPv4-Netze:Berechnungen A

```
1  # (C) 2025 K.Reincke: proTironeComputatri snippet [CC-BY-4.0]
2
3  import ipaddress
4
5  # insert a(n) (in)valid net description
6  IP_ADDR='192.168.0.2'
7  SN_MASK='255.255.255.0'
8  NT_ADDR='192.168.2.0'
9
10 # Rule: IP_ADDR & SN_MASK -> NT_ADDR:
11
12 ip_addr_int=(int)(ipaddress.ip_address(IP_ADDR))
13 nt_addr_int=(int)(ipaddress.ip_address(NT_ADDR))
14 sn_mask_int=(int)(ipaddress.ip_address(SN_MASK))
15
16 if ((ip_addr_int&sn_mask_int)==nt_addr_int):
17     print("net data fulfills IP_ADDR & SN_MASK -> NT_ADDR!")
18 else:
19     print("net data does not fulfill IP_ADDR & SN_MASK -> NT_ADDR!")
20
```



→ `cx.py2rf` (= *python-to-refresh*-Lektion) in *proTironeComputatri*:

<https://github.com/pro-tirone-computatri/protico.lesson/lf.cx/cx.py2rf>

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)