

George Boole (1815 – 1864) Vater der Computerlogik[1]

1847 verfasste Boole ein Buch mit dem Titel „Die mathematische Analyse der Logik“.

Es wird von der Menge {0,1} ausgegangen. Es stehen also lediglich die Elemente 0 und 1 zur Verfügung. Boole wies wahren Aussagen der Wert 1 und falschen den Wert 0 zu. Die **Grundoperationen** der Booleschen Algebra sind Konjunktion, Disjunktion und Negation. Meist wird mit zwei Ausgangsgrößen gerechnet, es sind aber auch mit mehr als zwei möglich.

Konjunktion = „**Und-Verknüpfung**“ Die Konjunktion ist genau dann 1 (TRUE), wenn das erste **und** das zweite Argument 1 ist. In jedem anderen Fall ist sie 0 (FALSE)

AND	0	1
0	0	0
1	0	1

```
a=20
b=7
if (a<50 and b<10):
   print('True')
else:
   print('False')            #Ausgabe True

if (a<50 and b>10):
   print('True')
else:
   print('False')            #Ausgabe False
```

Disjunktion = „**Oder-Verknüpfung**“. Das „Oder“ ist als „einschließendes Oder“ und nicht im Sinne von „entweder oder“ zu verstehen. Die Disjunktion ist genau dann 1 (True), wenn das erste **oder** das zweite Argument wahr sind. Sie ist nur dann 0 (False), wenn beide verknüpften Argumente ebenfalls 0 sind.

OR	0	1
0	0	1
1	1	1

```
a=20
b=7

if (a<50 or b>10):
   print('True')
else:
   print('False')     #Ausgabe True

if (a<10 or b>10):
   print('True')
else:
   print('False')     #Ausgabe False
```

[1] Vgl. Heinz-Nixdorf-Forum unter https://blog.hnf. de/der-vater-der-computerlogik/

Kontravalenz = „**Entweder-Oder-Verknüpfung**“. Die Kontravalenz ist genau dann 1 (True), wenn **nur** das erste bzw**. nur** das zweite Argument wahr sind. Sie ist dann 0 (False), wenn beide verknüpften Argumente 0 der 1 sind.

XOR	0	1
0	0	1
1	1	0

Die Negation (**Nicht-Operator**) ist keine binäre Verknüpfung, verlangt also nur ein Argument.

NOT	0	1
	1	0

Vergleichsoperatoren

Operator	Bedeutung	Beispiel
==	gleich	5 == 5 → True
!=	ungleich	3 != 4 → True
<	kleiner als	2 < 7 → True
<=	kleiner oder gleich	5 <= 7 → True
>	größer als	7 > 4 → True
>=	größer oder gleich	4 >= 4 → True
is	ist dasselbe Objekt	a is b
is not	ist nicht dasselbe Objekt	a is not b
in	ist enthalten in	'a' in 'Auto'
not in	ist nicht enthalten in	'x' not in 'Auto'

```
a=42
b=17
print(a is b)      # False
print(a is not b)  # True
print('x' in 'Auto') #False
print('x' not in 'Auto') #True
```

Bitweise Operatoren – sie vergleichen einzelnen Bits von Ganzzahlen

Operator	Bedeutung	Beispiel
&	bitweises UND	19 & 7 → 3
|	bitweises ODER	19 | 7 → 23
^	bitweises XOR	19 ^ 7 → 20
~	bitweise Negation	~19 → -20

Beispiel C = 00010011 **XOR** D = 00000111

```
00010011    = 19
00000111    = 7
------------
00010100    = 20
```