Vergleichsausdrücke - Logische Ausdrücke

In [1]:
```
2>7  #2 größer 7
```

Out[1]: False

In [2]:
```
2<7  #2 kleiner 7
```

Out[2]: True

In [ ]:
```
2==7  #Ist 2 gleich 7?
```

In [3]:
```
2!=7  #Ist 2 ungleich 7?
```

Out[3]: True

In [4]:
```
5==5  #Ist 5 gleich 5?
```

Out[4]: True

In [5]:
```
3<3  #ist 3 kleiner 3?
```

Out[5]: False

In [6]:
```
3<=3 #Ist 3 kleiner gleich 3?
```

Out[6]: True

In [7]:
```
not(5==5) #Ist 5 nicht gleich 5?
```

Out[7]: False

In [8]:
```
(2<7) and (5>4)  #Ist 2 kleiner 7 UND 5 größer 4?
```

Out[8]: True

In [9]:
```
(2>7) or (4>5)  #Ist 2 größer 7 ODER 4 größer 5?
```

Out[9]: False

In [10]:
```
not(3!=3) #Ist 3 nicht ungleich 3?
```

Out[10]: True

In [11]:
```
((8<5) or (4!=7)) and (not(4==5))  #Ist 8 kleiner 5 oder 4 ungleich 7 und nicht
```

Out[11]: True

Logische Operatoren

In [12]:
```
# Bitweise Verschiebung von n Stellen nach links "<<"

zahl = 5  # Beispielzahl
```

```
# Verschiebung um 2 Stellen nach links
verschoben = zahl << 2

# Ausgabe der Ergebnisse
print("Ursprüngliche Zahl:", zahl ,"und binär" , bin(zahl))
print("Nach links verschoben um 2 Stellen:", bin(verschoben), ' und ergibt dezim
```

```
Ursprüngliche Zahl: 5 und binär 0b101
Nach links verschoben um 2 Stellen: 0b10100  und ergibt dezimal 20
```

In [13]:
```
# Bitweise Verschiebung von n Stellen nach rechts ">>"

zahl = 20  # Beispielzahl

# Verschiebung um 3 Stellen nach rechts
verschoben = zahl >> 3

# Ausgabe der Ergebnisse
print("Ursprüngliche Zahl:", zahl, 'und binär', bin(zahl))
print("Nach rechts verschoben um 3 Stellen:", bin(verschoben), ' und ergibt dezi
```

```
Ursprüngliche Zahl: 20 und binär 0b10100
Nach rechts verschoben um 3 Stellen: 0b10  und ergibt dezimal 2
```

In [14]:
```
# Bitweises UND "&"

zahl1 = 12  # Beispielzahl 1: 1100 in binär
zahl2 = 7   # Beispielzahl 2: 0111 in binär

ergebnis = zahl1 & zahl2

# Ausgabe der Ergebnisse
print("Zahl 1 binär:", bin(zahl1),'und dezimal:' ,zahl1)
print("Zahl 2 binär:", bin(zahl2),'und dezimal:' ,zahl2)
print("Ergebnis der bitweisen UND-Verknüpfung:", bin(ergebnis), 'und ergibt dezi
```

```
Zahl 1 binär: 0b1100 und dezimal: 12
Zahl 2 binär: 0b111 und dezimal: 7
Ergebnis der bitweisen UND-Verknüpfung: 0b100 und ergibt dezimal 4
```

In [15]:
```
# Bitweises ODER "|"

zahl1 = 12  # Beispielzahl 1: 1100 in binär
zahl2 = 7   # Beispielzahl 2: 0111 in binär

ergebnis = zahl1 | zahl2

# Ausgabe der Ergebnisse
print("Zahl 1 binär:", bin(zahl1),'und dezimal:' ,zahl1)
print("Zahl 2 binär:", bin(zahl2),'und dezimal:' ,zahl2)
print("Ergebnis der bitweisen ODER-Verknüpfung:", bin(ergebnis), 'und ergibt dez
```

```
Zahl 1 binär: 0b1100 und dezimal: 12
Zahl 2 binär: 0b111 und dezimal: 7
Ergebnis der bitweisen ODER-Verknüpfung: 0b1111 und ergibt dezimal 15
```

In [16]:
```
# Bitweises Exclusiv-ODER "^"

zahl1 = 12  # Beispielzahl 1: 1100 in binär
zahl2 = 7   # Beispielzahl 2: 0111 in binär
```

```
ergebnis = zahl1 ^ zahl2

# Ausgabe der Ergebnisse
print("Zahl 1 binär:", bin(zahl1),'und dezimal:' ,zahl1)
print("Zahl 2 binär:", bin(zahl2),'und dezimal:' ,zahl2)
print("Ergebnis der bitweisen XOR-Verknüpfung:", bin(ergebnis), 'und ergibt dezi
```

```
Zahl 1 binär: 0b1100 und dezimal: 12
Zahl 2 binär: 0b111 und dezimal: 7
Ergebnis der bitweisen XOR-Verknüpfung: 0b1011 und ergibt dezimal 11
```

Übungsaufgabe

1. Gegeben sind zwei positive ganze Zahlen zahl1 und zahl2. Führen Sie die folgenden Schritte durch und geben Sie die Zwischenergebnisse sowie das Endergebnis in binärer Form aus:

   a. Verschieben Sie zahl1 um 2 Stellen nach links.

   b. Führen Sie eine bitweise UND-Verknüpfung zwischen dem verschobenen Ergebnis und zahl2 durch.

   c. Führen Sie eine bitweise ODER-Verknüpfung zwischen zahl1 und zahl2 durch.

2. Bitte schreiben Sie ein Programm, welches die duale Zahl (dez. 218) um 2 Bits nach links verschiebt und sowohl die Dualzahl als auch die entsprechende Dezimalzahl ausgibt.

3. Bitte schreiben Sie ein Programm, welches die Netz-IP der IP-Adresse 192.128.64.7/255.255.255.0 bestimmt. Zerlegen Sie sich dafür die IP-Adresse in vier Teile und wandeln Sie diese im ersten Schritt in Binärzahlen um. Anschließend vergleichen Sie die einzelnen Teile miteinandern, indem Sie sie bitweise UND verknüpfen (Vorlesung Netzwerktechnik Folie 52). Im letzten Schritt sollen die einzelnen Teile wieder zusammengefügt werden.