

# Zahlenumrechner

In diesem Skript werden Zahlen (in console) umgewandelt zwischen die folgende Zahlensysteme : Dezimal, Hexadezimal, Oktal und Binär  
Der Umwandlung zwischen die zahlen systeme erfolgt mit zwölf Funktionen, die unter zwei Grundlogiks bearbeiten werden.

- Grundlogik 1 : (hier Dezimal zum Hex)

```
# [converts from Decimal to Hexdecimal]

def decToHex(zahl_eingabe):
    hex_ergebnis = []

    if zahl_eingabe < 0:
        hex_ergebnis.append(0)
    elif zahl_eingabe < 10:
        hex_ergebnis.append(zahl_eingabe)
    else:
        decToHex(zahl_eingabe // 16)
        rest = zahl_eingabe % 16
        if rest < 10:
            hex_ergebnis.append(rest)
        elif rest == 10:
            hex_ergebnis.append("A")
        elif rest == 11:
            hex_ergebnis.append("B")
        elif rest == 12:
            hex_ergebnis.append("C")
        elif rest == 13:
            hex_ergebnis.append("D")
        elif rest == 14:
            hex_ergebnis.append("E")
        elif rest == 15:
            hex_ergebnis.append("F")

    # * is to unpack the list(no commas and bracets) end = '' to let element print on the same line
    return print(*hex_ergebnis, end=' ')
```

Die Funktion `decToHex` erwartet ein Benutzereingabe von Dezimal zahlen als parameter.

Erst ist die `hex_ergebnis` leere `liste` zu initialisieren.

Dann erfolgt die Überprüfung von die eingegebene Dezimal zahl.

1. Wenn die zahl unter null ist dann wird immer die null in die `liste` hinzufügt und die zahl null wir rückgegeben.(die umwandlung von negativen zahlen wird indiesem programm nicht gemacht werden)

2. Sonst wenn die eingegebene zahl unter 10 ist, dann wird die zahl in die `liste` hinzufügt und dann als rückgabe ausgegeben würden.  
( zahlen von 0 bis 9 sind ähnlich in Dezimal und Hex)

3. Sonst wird hier die handlung von zahlen ab 10 erfolgt

Die Funktion `decToHex` wird rekursiv ergänzt, deren parameter wid durch 16 genau geteilt.(mit genau durchteilen ist gemeint dass als ergebnis wird kein komma zahlen ausgegeben) durch die funktion aufruf erfolgt die direkte ausgabe von die durchgeteilte eingegebene zahl(wird nicht in die liste hinzugefügt)

```
if select == "h":  
    sysconv.decToHex(zahl)
```

4. Initialisere rest mit Benutzereingabe modulo 16. Dann wird ein überprüfung für die rest wert durchgeführt.

Wenn rest kleiner als 10 ist, dann wird die rest ergebnis in die `liste` eingepackt.

5. Sonst erfolgt ein fallunterscheidung zwischen 6 fälle und zwar,

- rest ergebnis ist gleich 10 dann, die buchstabe A wird in die liste hinzugefügt.
- rest ergebnis ist gleich 11 dann, die buchstabe B wird in die liste hinzugefügt.
- rest ergebnis ist gleich 12 dann, die buchstabe C wird in die liste hinzugefügt.
- rest ergebnis ist gleich 13 dann, die buchstabe D wird in die liste hinzugefügt.
- rest ergebnis ist gleich 14 dann, die buchstabe E wird in die liste hinzugefügt.
- rest ergebnis ist gleich 15 dann, die buchstabe F wird in die liste hinzugefügt.

Als rückgabe ist die liste ausgabe.

- Grundlogik 2: (hier Binär zum Dezimal)

```
def binToDec(zahl_eingabe):  
    # convert int to string, to break it into list of digits  
    dec_ergebnis = list(str(zahl_eingabe))  
    dec = 0  
    counter = 0  
  
    # reversed to calculate Bin string from right to left  
    # go through the list in reverse and multiply every element(i) by 2^counter, the counter starts from 0 and will be addup 1 for every loop iteration  
    for i in reversed(dec_ergebnis):  
        dec += (2**counter) * int(i)  
        counter += 1  
  
    del dec_ergebnis[:] # delete old values(zahl_eingabe)  
    dec_ergebnis.append(dec) # add new values(dec sum)  
    return print(*dec_ergebnis, end=' ')
```

Die Funktion **binToDec** erwartet ein Benutzereingabe von Binär(wird interpretiert als Dezimal zahl) als parameter.

Initialisiere **dec\_ergebnis** liste die Benutzereingabe beinhaltet und dieser inhalt zum Text umwandelt, dadurch erfolgt die speration von die Elemente, jeder einzelne element wird behandelt.

Initialisiere dec mit Wert 0

Initialisiere counter mit Wert 0

Für jedes Element aus umgekehrt gelesene **dec\_ergebnis** liste in i

(Die liste wird umgekehrt gelsen, da die umrechnung von Binär erfolgt von rechts nach links)

In dec wird die summe von 2 hoch counter wert mal **liste** element der als ganz zahl wieder zu interpretieren.

In counter wird 1 addiert für jeder iteration durchlauf.

dieser vorgang wiederholt sich bis alle **liste** elemente zusammen addiert sind.

Die **liste** inhalt von dem zum Text umgewandelte Benutzereingabe wird gelöscht.

Die dec Wert wird in die **liste** hinzugefügt

Als rückgabe ist die **liste** ausgabe.