# Zahlenumwandlung – Basissysteme

**Ac 2013-2016**

Es soll gezeigt werden, wie Zahlen (auch mit Nachkommaanteil) von einem Basissystem in ein anderes umgewandelt werden können. Dabei werden Basen von 2 bis 36 verwendet. Für jede Basis g sind Ziffern von 0 bis g-1 erlaubt ! Bei Basen größer als 10 verwendet man für Ziffern oberhalb von 9 die Buchstaben A(=10), B(=11), …, Z(=35). Das Dualsystem (g=2) hat die 2 Ziffern 0 und 1, das Dezimalsystem (g=10) hat die Ziffern 0 bis 9 und das 36-System hat die Ziffern 0 bis Z .

> Die Basis g ist eine natürliche Zahl ≥ 2 . Jede Zahl z in dieser Basis g mit (n+1) Vorkommastellen und m Nachkommastellen lässt sich eindeutig folgendermaßen darstellen (<u>g-adische Entwicklung von z</u>):
>
> $$z = a_n g^n + a_{n-1} g^{n-1} + \ldots + a_1 g^1 + a_0 + a_{-1} g^{-1} + a_{-2} g^{-2} + \ldots + a_{-m} g^{-m} \; ; \; a_n \neq 0$$
>
> Dafür schreibt man auch: $z = a_n \, a_{n-1} \ldots a_1 \, a_0 \, , \, a_{-1} \, a_{-2} \ldots a_{-m}$
>
> **Beispiele:** $4723{,}58_{10}$ $\quad D7C3{,}A6_{16}$ $\quad 110100{,}001_2$ $\quad 0{,}05_8$ $\quad K4ZY{,}0T_{36}$
>
> Die tiefgestellte Zahl gibt die Basis der Zahl an .

## <u>I. Umwandlung von einem beliebigen g-System in das 10-System:</u>

Gegeben ist eine Zahl z im g-System . Gesucht ist das Äquivalent im 10-System.
Die Darstellung im 10-System entspricht der obigen, aber die Anzahl der Vorkommastellen und die Anzahl der Nachkommastellen verändern sich gegebenenfalls .

Beispiel für eine Zahl im Zweiersystem (g = 2), die in das 10-System umgewandelt wird:

$$z = 1001{,}01_2 = 1\cdot 2^3 + 0\cdot 2^2 + 0\cdot 2^1 + 1 + 0\cdot 2^{-1} + 1\cdot 2^{-2} = 8 + 0 + 0 + 1 + 0 + 0{,}25 = 9{,}25_{10}$$

Die gesuchte Darstellung der Zahl z im 10-System kann mithilfe des sog. *HORNER-Schemas* ( geschicktes Ausklammern ! ) berechnet werden. Die Verwendung des Schemas erkennt man in den Schleifen des unten notierten Algorithmus.

(1) Der Vorkommateil von z sei **zVk** : ( Hinweis: zVk ist eine <u>natürliche Zahl</u> )
$$zVk = ((a_n\cdot g + a_{n-1})\cdot g + a_{n-2})\cdot g + \ldots + a_1)\cdot g + a_0$$

(2) Der Nachkommateil von z sei **zNk** : ( Hinweis: zNk ist eine <u>reelle Zahl</u> der Form 0, … )
$$zNk = (a_{-1}\cdot g^{m-1} + a_{-2}\cdot g^{m-2} + \ldots + a_{-m})/g^m$$ und somit
$$\mathbf{zNk = ( \ldots (a_{-1}\cdot g + a_{-2})\cdot g + \ldots + a_{-m+1})\cdot g + a_{-m})/g^m}$$

*<u>Daraus ergibt sich der Algorithmus für die Umwandlung g → 10 :</u>*

```
Eingaben
Lies g  // Zahl im Bereich 2,3, … ,9,11, … ,36
Für i von n ab bis 0 wiederhole
  Lies a[i] // Ziffern/Buchst. im Bereich 0,1,2, … , g-1
Für i von 1 bis m wiederhole
  Lies b[i] // Ziffern/Buchst. im Bereich 0,1,2, … , g-1

Berechnung von zVk ( Vorkommateil )
zVk = a[n]
Für i von n-1 ab bis 0 wiederhole
   zVk = g·zVk + a[i]

Berechnung von zNk ( Nachkommateil = 0,***** )
zNk = b[1] // nat.Zahl; entspricht a-1
Für i von 2 bis m wiederhole
   zNk = zNk·g + b[i]  // g = Dezimalzahl; zNk = nat.Zahl
zNk = zNk / g^m // evtl. Periode, dann zNk ungenau

Ausgabe
Schreib zVk+zNk  // gewünschte Dezimalzahl
```

*<u>Beispielrechnung:</u>*

$8A{,}F5_{16}$
also a1=8  a0=10
b1=15  b2=5

zVk = 8
zVk = 16·8+10 = 138

zNk = 15
zNk = (15·16+5) = 245
zNk = $245/16^2$ = 0,95703125

Ergebnis also:

$8A{,}F5_{16} = 138{,}95703125_{10}$

*<u>Weitere Beispiele:</u>* $HalloKarlo_{36} = 1756423207595580_{10}$ $\quad 1B{,}CD6F_{16} = 27{,}8024749755859375_{10}$ $\quad AFFE_{16} = 45054_{10}$

## II. Umwandlung vom 10-System in ein beliebiges g-System:

> z sei im 10-System gegeben (vgl. Formel oben) und soll in das g-System umgewandelt werden:
> $$z = a_k 10^k + a_{k-1}10^{k-1} + \dots + a_1 10^1 + a_0 + a_{-1}10^{-1} + a_{-2}10^{-2} + \dots + a_{-j}10^{-j} \;;\; a_k \neq 0$$
>
> Gesucht ist somit die g-System-Darstellung
> $$z = c_n g^n + c_{n-1}g^{n-1} + \dots + c_1 g^1 + c_0 + c_{-1}g^{-1} + c_{-2}g^{-2} + \dots + c_{-m}g^{-m} \;;\; c_n \neq 0$$

**Der Vorkommateil zVk** kann dann eindeutig nach Potenzen von g entwickelt werden, wobei die entstehenden Koeffizienten im Bereich { 0, 1, 2, … g-1 } liegen . Dies geschieht durch fortwährende Abspaltung von Divisionsresten. Man dividiert also durch g und bildet den Rest der Division usw. .

Es gilt: $\quad zVk = c_n g^n + c_{n-1}g^{n-1} + \dots + c_1 g^1 + c_0$

Für die erste Division erhält man $\mathrm{int}(zVk/g) = \mathrm{int}(c_n g^{n-1} + c_{n-1}g^{n-2} + \dots + c_1 + c_0 / g)$
Da aber wegen $c_0 < g$ gilt: $\mathrm{int}(c_0/g) = 0$ , bleibt: $\mathrm{int}(zVk/g) = \mathrm{int}(c_n g^{n-1} + c_{n-1}g^{n-2} + \dots + c_1)$ .
Daraus folgt: $g\cdot \mathrm{int}(zVk/g) = c_n g^{n-1} + c_{n-1}g^{n-2} + \dots + c_1$
Der Rest ist $r = zVk - g\cdot \mathrm{int}(zVk/g) = c_0$ . Somit hat man bereits $c_0$ abgespalten !
Dieses Abspaltungsverfahren führt man dann für int(zVk/g) statt zVk fort, usw. .

*Beispielrechnung für eine Zahl im Zweiersystem (g = 2):*

Sei zVk = 37. Es wird nun fortwährend durch 2 dividiert und der Rest notiert:

```
37 / 2 = 18 Rest 1.
18 / 2 = 9 Rest 0.
9 / 2 = 4 Rest 1.
4 / 2 = 2 Rest 0.
2 / 2 = 1 Rest 0.
1 / 2 = 0 Rest 1.
```

Beim Ergebnis 0 endet der Algorithmus. Die Reste (oberster Wert ganz rechts notiert) sind **100101**.
Dies ist die Dualzahldarstellung von **37**, wie man leicht nachprüfen kann.

**Der Nachkommateil zNk** lässt sich auf ähnliche Weise bestimmen wie zVk, nur wird hier mit g fortwährend multipliziert .

Es gilt: $\quad zNk = c_{-1}g^{-1} + c_{-2}g^{-2} + \dots + c_{-m}g^{-m}$

Dann erhält man: $\mathrm{int}(g\cdot zNk) = \mathrm{int}(g\cdot(c_{-1}g^{-1} + c_{-2}g^{-2} + \dots + c_{-m}g^{-m})) =$
$\mathrm{int}(c_{-1} + c_{-2}g^{-1} + \dots + c_{-m}g^{-m+1}) = c_{-1}$
Dies gilt, weil alle anderen Teilterme der Klammer im Intervall [0;1[ liegen und ihr int-Anteil damit 0 ist !
Wir haben also bereits $c_{-1}$ , den ersten Koeffizienten nach dem Komma, abgespalten !
Der Rest ist $r = g\cdot zNk - \mathrm{int}(g\cdot zNk) = c_{-2}g^{-1} + c_{-3}g^{-2} + \dots + c_{-m}g^{-m+1}$ .
Dieser Rest wird im nächsten Schritt wieder für zNk eingesetzt, und das Verfahren wiederholt.
Der Algorithmus endet, falls zNk = 0 wird. Dies ist aber in den meisten Fällen nicht so, weil häufig eine periodische Darstellung für den Nachkommaanteil im g-System entsteht.
Man muss dann eine Schranke für die Anzahl der Stellen angeben.

*Beispielrechnung 1 für eine Zahl im Zweiersystem (g = 2):*

Sei zNk = 0,25 im 10-System gegeben.
Dann ist int(2·0,25) = int(0,5) = **0** . Dies ist bereits die erste Stelle nach dem Komma.
Der Rest ist r = 0,5 – 0 = 0,5. Dies ist das neue zNk.
Wir bilden wieder int(2·0,5) = int(1) = **1**. Die zweite Stelle nach dem Komma.
Der Rest ist r = 1 – 1 = 0. Dies ist das neue zNk.
Wegen zNk = 0 bricht hier der Algorithmus ab.

Ergebnis: $0{,}25_{10} = 0{,}01_2$ .

*<u>Beispielrechnung 2 für eine Zahl im 3-System (g = 3):</u>*

Sei zNk = 0,75 im 10-System gegeben.
Dann ist int(3·0,75) = int(2,25) = **<u>2</u>** . (erste Stelle nach dem Komma)
Der Rest ist r = 2,25 – 2 = 0,25. Dies ist das neue zNk.
Wir bilden wieder int(3·0,25) = int(0,75) = **<u>0</u>**. Die zweite Stelle nach dem Komma.
Der Rest ist r = 0,75 – 0 = 0,75. Dies ist das neue zNk.
Hier können wir den Algorithmus beenden, denn zNk = 0,75 war bereits zu Beginn gegeben.
Daher werden sich die beiden ersten Schritte stets wiederholen und damit die Folge 20 immer wieder entstehen.
Wir haben eine Periode !

Ergebnis: $0{,}75_{10} = 0{,}\overline{20}_3$ .

*<u>Weitere Beispiele (ohne Nachweis):</u>*

$0{,}64_{10} = 0{,}\overline{\mathbf{10100011110101110000}}_2$
$0{,}2_{10} = 0{,}\overline{\mathbf{0011}}_2$
$0{,}6_{10} = 0{,}\overline{\mathbf{1210}}_3$

$6{,}13_{10} = 6{,}4OHA2VOHA2VOHA2VOHA2VOHA2VOHA2_{36}$
$5919{,}9_{10} = EFJ{,}I_{20}$
$1/7_{10} = 0{,}\overline{\mathbf{1}}_8$

<u>Überlegungen zur Genauigkeit:</u>
Wegen der ungenauen Rechnung mit reellen Zahlen rechnen wir auch für zNk mit natürlichen Zahlen:
Dazu müssen wir nur zNk mit **mult := $10^m$** multiplizieren, dann erhalten wir die Ziffernfolge hinter dem Komma.
Da der Exponent von 10 jetzt stets ≥ 0 ist und $c_i \in [0 ; 9]$ treten nun keine gebrochenen Anteile mehr auf .
*Selbstverständlich müssen wir bei der int()- sowie der Rest-Bildung auch wieder durch **mult** dividieren !*

*<u>Algorithmus für die Umwandlung 10 → g :</u>*

```
Eingaben
Lies g      // Zahl im Bereich 2,3, … ,9,11, … ,36
Lies zVk    // Ziffern im Bereich 0, … , 9
Lies zNk    // Ziffern im Bereich 0, … , 9
Lies maxSt  // maximale Stellenzahl,z.B. 50

Berechnung der Koeff. c[i] von zVk
j = 0
wiederhole
   c[j] = zVk mod g
   zVk = zVk div g
   j = j+1
bis zVk = 0
für i von j-1 ab bis 0 wiederhole Schreib c[i]

Berechnung der Koeff. von zNk
mult = 10^m     // m ist die Länge von zNk
zaehler = 0
wiederhole
   zaehler = zaehler+1
   zNk = zNk·g
   Schreib zNk div mult // Ergebnisse > 9 in A,B, … umwandeln
   zNk = zNk mod mult
bis zNk = 0 oder (zaehler = maxSt) oder „Periode erkannt“
```

*<u>Beispielrechnung 1 für ein ganzzahliges zNk , umzuwandeln in das 2-System (g = 2):</u>*

Sei z = 0,75 im 10-System gegeben, also zNk=75 g=2 mult=$10^2$ = 100
zNk·g div mult = 75·2 div 100 = 150 div 100 = **<u>1</u>**
zNk = 150 mod 100 = 50
zNk·g div mult = 50·2 div 100 = 100 div 100 = **<u>1</u>**
zNk = 100 mod 100 = 0 // Ende

Ergebnis: $0{,}75_{10} = 0{,}11_2$

*Beispielrechnung 2 für ein ganzzahliges zNk , umzuwandeln in das 16-System (g = 16):*

Sei z = 0,82 im 10-System gegeben, also zNk=82  g=16  mult = $10^2$ = 100
zNk·g div mult = 82·16 div 100 = 1312 div 100 = **13 = D**
zNk = 1312 mod 100 = 12
zNk·g div mult = 12·16 div 100 = 192 div 100 = **1**
zNk = 192 mod 100 = 92
zNk·g div mult = 92·16 div 100 = 1472 div 100 = **14 = E**
zNk = 1472 mod 100 = 72
zNk·g div mult = 72·16 div 100 = 1152 div 100 = **11 = B**
zNk = 1152 mod 100 = 52
zNk·g div mult = 52·16 div 100 = 832 div 100 = **8**
zNk = 832 mod 100 = 32
zNk·g div mult = 32·16 div 100 = 512 div 100 = **5**
zNk = 512 mod 100 = 12 (12 war bereits oben als Rest, also Periode !)
Daher sind die nächsten Ziffern:  1 E B 8 5 1 E B 8 5 1 ...

Ergebnis:   $0,82_{10} = 0,D\overline{1EB85}_{16}$