

Uebungen ■ Exercices

8. Datentypen ■ Types de données

Die Gliederung dieses Kurses folgt in groben Zügen dem Buch von Nancy Blachman: A Practical Approach.... Hinweis: Kapitel 8 lesen!

■ L'articulation de ce cours correspond à peu près à celle du livre de Nancy Blachman: A Practical Approach....
Indication: Lire le chapitre 8.

Run mit WIN+*Mathematica* Version 5.2

■ Testé avec *Mathematica* version 5.2+WIN

WIR94/98/99/2000/2007 // Copyright Rolf Wirz

Aufgabe 1 ■ Problème 1

Untersuche die Interne Darstellung von:
■ Examine la représentation intérieure de

a) 35 a

```
In[1]:= FullForm[35a]
```

```
Out[1]//FullForm=  
Times[35, a]
```

b) a 35

```
In[2]:= FullForm[a35]
```

```
Out[2]//FullForm=  
a35
```

c) - x

In[3]:= FullForm[-x]

Out[3]//FullForm=
Times[-1, x]

d) - 3

In[4]:= FullForm[-3]

Out[4]//FullForm=
-3

e) x - 3

In[5]:= FullForm[x - 3]

Out[5]//FullForm=
Plus[-3, x]

f) 3 - x

In[6]:= FullForm[3 - x]

Out[6]//FullForm=
Plus[3, Times[-1, x]]

g) Sqrt[5]

In[7]:= FullForm[Sqrt[5]]

Out[7]//FullForm=
Power[5, Rational[1, 2]]

h) Series[Tan[x],{x,0,9}]

In[8]:= Series[Tan[x],{x,0,9}]

Out[8]= $x + \frac{x^3}{3} + \frac{2x^5}{15} + \frac{17x^7}{315} + \frac{62x^9}{2835} + O[x]^{10}$

In[9]:= FullForm[Series[Tan[x],{x,0,9}]]

Out[9]//FullForm=
SeriesData[x, 0, List[1, 0, Rational[1, 3], 0,
Rational[2, 15], 0, Rational[17, 315], 0, Rational[62, 2835]], 1, 10, 1]

In[10]:=

g) Normal[Series[Tan[x],{x,0,9}]]

In[10]:= Normal[FullForm[Series[Tan[x],{x,0,9}]]]

Out[10]//FullForm=
Plus[x, Times[Rational[1, 3], Power[x, 3]], Times[Rational[2, 15], Power[x, 5]],
Times[Rational[17, 315], Power[x, 7]], Times[Rational[62, 2835], Power[x, 9]]]

Aufgabe 2 ■ Problème 2

- a) Vergleich von $(x^y)^z$ mit $x^{(y^z)}$
 ■ Comparaison de $(x^y)^z$ avec $x^{(y^z)}$

$x^y{}^z$

`In[11]:= 2^3^4`

`Out[11]= 2417851639229258349412352`

$x^{(y^z)}$

`In[12]:= 2^(3^4)`

`Out[12]= 2417851639229258349412352`

$(x^y)^z$

`In[13]:= (2^3)^4`

`Out[13]= 4096`

- b) Vergleich der FullForm
 ■ Comparaison de FullForm

$x^y{}^z$

`In[14]:= 2^3^4`

`Out[14]= 2417851639229258349412352`

`In[15]:= FullForm[2^3^4]`

`Out[15]//FullForm=`
`2417851639229258349412352`

$x^{(y^z)}$

`In[16]:= 2^(3^4)`

`Out[16]= 2417851639229258349412352`

`In[17]:= FullForm[2^(3^4)]`

`Out[17]//FullForm=`
`2417851639229258349412352`

$(x^y)^z$

```

In[18]:= (2^3)^4
Out[18]= 4096

In[19]:= FullForm[(2^3)^4]
Out[19]//FullForm=
4096

```

Aufgabe 3 ■ Problème 3

Untersuchung der Memory-Belegung bei Operationen ■ Examiner l'occupation de la mémoire lors d'opérations

Operation:

Liste von 1000 Integer-Zufallszahlen zwischen 100 und 200 rechnen:

■ Opération:

Calculer une liste de 1000 nombres aléatoires "Integer" entre 100 et 200:

```

In[20]:= aufruf1 = MemoryInUse[];
          kleineInt = Table[Random[Integer, {100, 200}],
                           {1000}];

In[22]:= aufruf2 = MemoryInUse[];
          aufruf2 - aufruf1

Out[23]= 6776

```

Operation:

Liste von 1'000 Integer-Zufallszahlen zwischen 300 und 400 rechnen:

■ Opération:

Calculer une liste de 1'000 nombres aléatoires integer entre 300 et 400:

```

In[24]:= kleineInt = Table[Random[Integer, {300, 400}],
                           {1000}];

In[25]:= aufruf3 = MemoryInUse[];
          aufruf3 - aufruf2

Out[26]= 5368

```

Wieviel Memory ist mehr benutzt worden?

(Mathematica ist so gebaut worden, weil eine Klasse mehr benutzt wird...)

■ Combien de mémoire en plus a-t-on occupé?

(Mathematica a été construit ainsi, parce qu'on utilise une classe de plus...)

Operation:

Liste von 1'000 Real-Zufallszahlen (default-Präzision) rechnen:

■ Opération:

Calculer une liste de 1'000 nombres aléatoires réels (précision default):

```

In[27]:= realInt = Table[Random[], {1000}];

```

```
In[28]:= aufruf4 = MemoryInUse[];
          aufruf4 - aufruf3
```

```
Out[29]= 9568
```

Aufgabe 4 ■ Problème 4

Was ist die grösste speicherbare quadratische Matrix bei 4 Megabytes?

■ Quelle est la plus grande matrice carrée qu'on peut mémoriser?

Annahme: Zur Speicherung einer Floating-point-Zahl braucht es $32 = 2^5$ Bytes.

■ Supposition: Pour mémoriser un nombre à virgule flottante on utilise

$32 = 2^5$ bytes.

```
In[30]:= NSolve[2^x==10^6,x]
```

Solve::ifun : Inverse functions are being used by Solve, so some solutions may not be found; use Reduce for complete solution information. Mehr...

```
Out[30]= {{x -> 19.9316}}
```

```
In[31]:= ?NSolve
```

NSolve[lhs==rhs, var] gives a list of numerical approximations to the roots of a polynomial equation. NSolve[{eqn1, eqn2, ... }, {var1, var2, ... }] solves a system of polynomial equations. Mehr...

1 Mb entspricht etwa 2^{20} Bytes.

■ 1 Mb correspond à environ 2^{20} bytes.

Anzahl Floating-point Zahlen in 4 Mb:

■ Nombre (quantité) de nombres à virgule flottante dans 4 Mb:

```
In[32]:= a = 4 2^19.9316/2^5
```

```
Out[32]= 125003.
```

Seitenlänge der quadratischen Matrix:

■ Longueur d'un côté de la matrice carrée:

```
In[33]:= s = Floor[Sqrt[a]]
```

```
Out[33]= 353
```

"Putzmaschine" einsetzen

■ Employer la "machine de nettoyage"

```
In[34]:= (* Old Form: Remove["Global`*"] *)
```

```
In[35]:= Remove["Global`*"]
```