

# Uebungen ■ Exercices

## 10. Lokale Variablen und prozedurales Programmieren ■ Variables locales et programmation procédurale

Die Gliederung dieses Kurses folgt in groben Zügen dem Buch von Nancy Blachman: A Practical Approach.... Hinweis: Kapitel 10 lesen!

- L'articulation de ce cours correspond à peu près à celle du livre de Nancy Blachman: A Practical Approach....  
Indication: Lire le chapitre 10.

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

**Bestimme die Anzahl Iterationsschritte, die notwendig sind, um "Wurzel aus 2" auf 30 Dezimalen mittels der Newton-Methode zu approximieren.**

- **Détermine le nombre de pas d'itération qui sont nécessaires pour approximer "racine de 2" jusqu'à 30 décimales par la méthode de Newton.**

Starte mit dem Wert 2. Berechne die Wurzel mit der Formel  $x_{(i+1)} = (x_{(i)} + 2/x_{(i)})/2$ . Erkläre wie diese Formel entsteht. (Löse  $x^2 - 2 = 0$  ...)

- Démarre avec la valeur 2. Calcule la racine par la formule  $x_{(i+1)} = (x_{(i)} + 2/x_{(i)})/2$ . Explique comment cette formule s'est formée. (Résous  $x^2 - 2 = 0$  ...)

```
In[1]:= Clear[newtonMethode, counter, zahl, approx];
newtonMethode[x_]:=
  Module[{
    counter = 1,
    zahl = N[x,30],
    approx = N[x,40]
  },
  While[approx^2 != zahl,
    Print[counter, ": ", N[approx,30]];
    approx = (approx + 2/approx)/2;
    ++counter;
  ];
  Print[counter, ": ", N[approx,30]];
]
```

```
In[3]:= newtonMethode[2]

1:  2.000000000000000000000000000000
2:  1.500000000000000000000000000000
3:  1.416666666666666666666666666667
4:  1.41421568627450980392156862745
5:  1.41421356237468991062629557889
6:  1.41421356237309504880168962350
7:  1.41421356237309504880168872421
```

Vergleich: ■ Comparaison:

```
In[4]:= N[Sqrt[2],30]

Out[4]= 1.41421356237309504880168872421
```

## Aufgabe 2 ■ Problème 2

**Definiere eine Funktion, die als Argumente drei Zahlen aufnimmt und die Summe der Quadrate der beiden grösseren Zahlen ausgibt.**

**■ Définis une fonction qui prend 3 nombres comme arguments et sort la somme des carrés des deux nombres plus grands.**

Definition: ■ Définition:

```
In[5]:= Clear[f];
f[x_,y_,z_]:=
  Module[{i,j,liste,min},
    min=Min{x,y,z};
    liste=Sort[{x,y,z}];
    restQuadr=Rest[{x,y,z}]^2;
    Print[Apply[Plus,restQuadr]]
  ]

General::spell1 :
Possible spelling error: new symbol name "min" is similar to existing symbol "Min". Mehr...
```

Ausprobieren: ■ **Essayer:**

```
In[7]:= {f[0,0,0],f[0,0,1],f[0,1,1],f[1,1,1],f[1,1,2],f[1,2,3]}
```

```
0
```

```
1
```

```
2
```

```
2
```

```
5
```

```
13
```

```
Out[7]= {Null, Null, Null, Null, Null, Null}
```

```
In[8]:= f[3]
```

```
Out[8]= f[3]
```

Was ist passiert?

■ Que s'est-il passé?

### Aufgabe 3 ■ Problème 3

**Annahme: Familien haben Kinder, bis erstmals ein Knabe kommt. Mache eine Simulation mit 1000 Familien um herauszufinden, wieviele Kinder etwa durchschnittlich eine Familie haben wird (Abschätzung). Wieviele Töchter und wieviele Söhne werden in einer Familie zu erwarten sein?**

**■ Supposition: Des familles ont des enfants jusqu'à la naissance du 1er garçon. Fais une simulation avec 1000 familles pour obtenir le nombre moyen d'enfants qu'une famille aura (estimation). A combien de filles et à combien de fils pourra s'attendre une famille?**

Lösung von Nancy Blachman:

■ Solution de Nancy Blachman:

```
In[9]:= Clear[makeFamily];
makeFamily[ ] :=
Module[{
  children = {}
},
While[Random[Integer] == 0,
  AppendTo[children, "girl"]
];
Append[children, "boy"]
]

In[11]:= makeFamily::usage="makeFamily[ ] returns a list
of children."

Out[11]= makeFamily[ ] returns a list of children.
```

```
In[12]:= Clear[numChildren];
numChildren[n_Integer]:=
Module[{
  allChildren
},
allChildren = Flatten[Table[
  makeFamily[ ],{n}]];
{
  avgChildren -> Length[
    allChildren]/n,
  avgBoys -> Count[allChildren,
    "boy"]/n,
  avgGirls -> Count[allChildren,
    "girl"]/n
}
]
```

```
In[14]:= numChildren::usage="numChildren[n] returns
statistics on the number of children
from n families."
```

```
Out[14]= numChildren[n] returns statistics on the number of children from n families.
```

```
In[15]:= numChildren[1000]
```

```
Out[15]= {avgChildren ->  $\frac{493}{250}$ , avgBoys -> 1, avgGirls ->  $\frac{243}{250}$ }
```

## Aufgabe 4 ■ Problème 4

### Mischen von Kartenspielen:

#### ■ Mélanger des jeux de cartes:

Beim perfekten Mischen teilt der Spieler das Kartenspiel genau in der Mitte und mischt das Spiel im "Reissverschlussverfahren": Abwechslungsweise kommt von jeder Seite je eine Karte, wobei sicher sein muss, dass die erste Karte von der ersten Hälfte zuerst "hingelegt" wird (das Spiel wird umgekehrt gehalten). Z.B. wenn man ein Spiel mit 10 Karten hat, ursprünglich geordnet in der Reihenfolge {1, 2, 3, 4, 5, 6, 7, 8, 9, 10}, so ist es nach einmal perfektem Mischen in der Reihenfolge {1, 6, 2, 7, 3, 8, 4, 9, 5, 10}.

- Schreibe eine Funktion "perfektMischen", die das perfekte Mischen einer geraden Anzahl von Karten simuliert.
- Schreibe die Funktion "wiederOrdnung", die die Anzahl perfekter Mischungen berechnet um das Spiel wieder in die ursprüngliche Ordnung zu bringen. Wieviel mal ist es bei 52 Karten?

■

Un joueur qui mélange les cartes à la perfection, divise le jeu de cartes exactement par la moitié et mélange les cartes par la méthode de la "fermeture éclair": De chaque côté il arrive une carte alternativement, mais on doit être sûr que la 1ère carte vient de la 1ère moitié (on tient le jeu à l'envers). P.ex. si on a un jeu de 10 cartes dans l'ordre d'origine {1, 2, 3, 4, 5, 6, 7, 8, 9, 10}, on obtient l'ordre suivant en mélangeant une fois {1, 6, 2, 7, 3, 8, 4, 9, 5, 10}.

- Ecris une fonction "perfectMischen" qui simule un mélange parfait d'un nombre pair de cartes.
- Ecris la fonction "wiederOrdnung" qui calcule le nombre de mélange parfaits pour remettre le jeu dans l'ordre d'origine. Combien de mélanges faut-il pour un jeu de 52 cartes?

```
In[16]:= Clear[mischen, perfektMischen, mischen1,
perfektMischen1];

mischen::usage="mischen[ ] erzeugt eine Liste
```

```

        von perfekt gemischten Zahlen -- donne
        une liste de nombres mélangés à la
        perfection.";
mischen[n_]:=
Module[{matrix1,matrix2},
  listeIn=Range[n];
  matrix1=Partition[listIn,n/2];
  matrix2=Transpose[matrix1];
  listeOut=Flatten[matrix2];
  Print["Liste_In: ", listeIn, " ",
    "Liste_Out: ", listeOut]
];

perfektMischen::usage="perfektMischen[ ] greift
auf den Modul mischen[ ] im Falle einer
eingegebenen geraden natuerlichen Zahl --
utilise le Modul mischen[ ] dans le cas
d-un nombre donné pair, naturel.";

perfektMischen[n_Integer]:=
Module[{listeIn},
  If[(OddQ[n] || n <= 0),
    Print["Bitte gerade, positive Zahl
eingeben -- S.v.p. entrer nombre positif
pair."], mischen[n],
    Print["Bitte gerade, positive Zahl
eingeben -- S.v.p. entrer nombre positif
pair."]]
];

mischen1::usage="mischen1[ ] erzeugt eine Liste
von perfekt gemischten Zahlen ohne
Output -- donne une liste de nombres
mélangés à la perfection sans output.";
mischen1[n_]:=
Module[{matrix1,matrix2},
  listeIn=Range[n];
  matrix1=Partition[listIn,n/2];
  matrix2=Transpose[matrix1];
  listeOut=Flatten[matrix2];
];

perfektMischen1::usage="perfektMischen1[ ]
greift auf den Modul mischen[ ]
im Falle einer eingegebenen geraden
natuerlichen Zahl. Im Unterschied zu
perfektMischen[ ]
wird hier die Ausgabe unterdruckt
-- utilise le Modul mischen[ ] dans le cas
d-un nombre donné pair, naturel. Par contre
à perfektMischen[ ] l-output est retenu.";

perfektMischen1[n_Integer]:=
Module[{}],
  If[(OddQ[n] || n <= 0),
    Print["Bitte gerade, positive Zahl
eingeben -- S.v.p. entrer nombre positif
pair."], mischen1[n],
    Print["Bitte gerade, positive Zahl
eingeben -- S.v.p. entrer nombre positif
pair."]]];

```

```
In[27]:= perfektMischen[100]
```

```
Liste_In: {1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20,  
21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41,  
42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62,  
63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80, 81, 82, 83,  
84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96, 97, 98, 99, 100} Liste_Out:  
{1, 51, 2, 52, 3, 53, 4, 54, 5, 55, 6, 56, 7, 57, 8, 58, 9, 59, 10, 60, 11, 61, 12, 62, 13, 63,  
14, 64, 15, 65, 16, 66, 17, 67, 18, 68, 19, 69, 20, 70, 21, 71, 22, 72, 23, 73, 24, 74, 25, 75, 26,  
76, 27, 77, 28, 78, 29, 79, 30, 80, 31, 81, 32, 82, 33, 83, 34, 84, 35, 85, 36, 86, 37, 87, 38, 88,  
39, 89, 40, 90, 41, 91, 42, 92, 43, 93, 44, 94, 45, 95, 46, 96, 47, 97, 48, 98, 49, 99, 50, 100}
```

```
In[28]:= listeOut
```

```
Out[28]= {1, 51, 2, 52, 3, 53, 4, 54, 5, 55, 6, 56, 7, 57, 8, 58, 9, 59, 10, 60, 11, 61,  
12, 62, 13, 63, 14, 64, 15, 65, 16, 66, 17, 67, 18, 68, 19, 69, 20, 70, 21, 71,  
22, 72, 23, 73, 24, 74, 25, 75, 26, 76, 27, 77, 28, 78, 29, 79, 30, 80, 31, 81,  
32, 82, 33, 83, 34, 84, 35, 85, 36, 86, 37, 87, 38, 88, 39, 89, 40, 90, 41,  
91, 42, 92, 43, 93, 44, 94, 45, 95, 46, 96, 47, 97, 48, 98, 49, 99, 50, 100}
```

```

In[29]:= Clear[mischenAnzahl];

mischenAnzahl::usage="mischenAnzahl[ ] greift auf
den Modul perfektMischen1[ ] und wendet
diesen an, bis die ursprüngliche
Reihenfolge wieder hergestellt ist. Ein
Zaehler zaehlt die Anzahl Mischablaeufe
-- utilise le Modul perfektMischen1[ ]
et l-utilise jusqu-à l-ordre original est
remis. Un compteur compte le nombre de
mélanges.";

mischenAnzahl[n_]:=
Module[{liste, counter=1, matrix1, matrix2},
mischen1[n];
liste =listeIn;
While[(!listeOut != liste) &&
counter < 10000), Print[counter," ",
listeIn," ", listeOut];
matrix1=Partition[listOut,n/2];
matrix2=Transpose[matrix1];
listeOut=Flatten[matrix2];
++counter;
]
Print[counter," ",listeIn," ", listeOut];
Print["Anzahl Mischungen --
Nombre de mélanges: ", counter];

]

perfektMischenAnzahl::usage="perfektMischenAnzahl[ ]
greift auf den Modul mischenAnzahl[ ] im
Falle einer eingegebenen geraden
natuerlichen Zahl --
utilise le Modul mischenAnzahl[ ] dans le
cas d-un nombre donné pair, naturel..";

perfektMischenAnzahl[n_Integer]:=
Module[{}],
If[OddQ[n] || n <= 0),
Print["Bitte gerade, positive Zahl
eingeben-- S.v.p. entrer nombre positif
pair."], mischenAnzahl[n],
Print["Bitte gerade, positive Zahl
eingeben-- S.v.p. entrer nombre positif
pair."]]

Syntax::newl:
The newline character after "
]"
is understood as a multiplication operator.
Mehr...

In[34]:= perfektMischenAnzahl[1]

Bitte gerade, positive Zahl eingeben-- S.v.p. entrer nombre positif pair.

In[35]:= perfektMischenAnzahl[0]

Bitte gerade, positive Zahl eingeben-- S.v.p. entrer nombre positif pair.

```

*In[36]* := **perfektMischenAnzahl[2]**

1 {1, 2} {1, 2}

Anzahl Mischungen --      Nombre de mélanges:    1

*In[37]* := **perfektMischenAnzahl[4]**

1 {1, 2, 3, 4} {1, 3, 2, 4}

2 {1, 2, 3, 4} {1, 2, 3, 4}

Anzahl Mischungen --      Nombre de mélanges:    2

*In[38]* := **perfektMischenAnzahl[6]**

1 {1, 2, 3, 4, 5, 6} {1, 4, 2, 5, 3, 6}

2 {1, 2, 3, 4, 5, 6} {1, 5, 4, 3, 2, 6}

3 {1, 2, 3, 4, 5, 6} {1, 3, 5, 2, 4, 6}

4 {1, 2, 3, 4, 5, 6} {1, 2, 3, 4, 5, 6}

Anzahl Mischungen --      Nombre de mélanges:    4

*In[39]* := **perfektMischenAnzahl[8]**

1 {1, 2, 3, 4, 5, 6, 7, 8} {1, 5, 2, 6, 3, 7, 4, 8}

2 {1, 2, 3, 4, 5, 6, 7, 8} {1, 3, 5, 7, 2, 4, 6, 8}

3 {1, 2, 3, 4, 5, 6, 7, 8} {1, 2, 3, 4, 5, 6, 7, 8}

Anzahl Mischungen --      Nombre de mélanges:    3

*In[40]* := **perfektMischenAnzahl[10]**

1 {1, 2, 3, 4, 5, 6, 7, 8, 9, 10} {1, 6, 2, 7, 3, 8, 4, 9, 5, 10}

2 {1, 2, 3, 4, 5, 6, 7, 8, 9, 10} {1, 8, 6, 4, 2, 9, 7, 5, 3, 10}

3 {1, 2, 3, 4, 5, 6, 7, 8, 9, 10} {1, 9, 8, 7, 6, 5, 4, 3, 2, 10}

4 {1, 2, 3, 4, 5, 6, 7, 8, 9, 10} {1, 5, 9, 4, 8, 3, 7, 2, 6, 10}

5 {1, 2, 3, 4, 5, 6, 7, 8, 9, 10} {1, 3, 5, 7, 9, 2, 4, 6, 8, 10}

6 {1, 2, 3, 4, 5, 6, 7, 8, 9, 10} {1, 2, 3, 4, 5, 6, 7, 8, 9, 10}

Anzahl Mischungen --      Nombre de mélanges:    6



**In[41]:= perfektMischenAnzahl[12]**

```

1 {1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12} {1, 7, 2, 8, 3, 9, 4, 10, 5, 11, 6, 12}
2 {1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12} {1, 4, 7, 10, 2, 5, 8, 11, 3, 6, 9, 12}
3 {1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12} {1, 8, 4, 11, 7, 3, 10, 6, 2, 9, 5, 12}
4 {1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12} {1, 10, 8, 6, 4, 2, 11, 9, 7, 5, 3, 12}
5 {1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12} {1, 11, 10, 9, 8, 7, 6, 5, 4, 3, 2, 12}
6 {1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12} {1, 6, 11, 5, 10, 4, 9, 3, 8, 2, 7, 12}
7 {1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12} {1, 9, 6, 3, 11, 8, 5, 2, 10, 7, 4, 12}
8 {1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12} {1, 5, 9, 2, 6, 10, 3, 7, 11, 4, 8, 12}
9 {1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12} {1, 3, 5, 7, 9, 11, 2, 4, 6, 8, 10, 12}
10 {1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12} {1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12}

Anzahl Mischungen --      Nombre de mélanges:      10

```

**In[42]:= perfektMischenAnzahl[14]**

```

1 {1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14} {1, 8, 2, 9, 3, 10, 4, 11, 5, 12, 6, 13, 7, 14}
2 {1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14} {1, 11, 8, 5, 2, 12, 9, 6, 3, 13, 10, 7, 4, 14}
3 {1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14} {1, 6, 11, 3, 8, 13, 5, 10, 2, 7, 12, 4, 9, 14}
4 {1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14} {1, 10, 6, 2, 11, 7, 3, 12, 8, 4, 13, 9, 5, 14}
5 {1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14} {1, 12, 10, 8, 6, 4, 2, 13, 11, 9, 7, 5, 3, 14}
6 {1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14} {1, 13, 12, 11, 10, 9, 8, 7, 6, 5, 4, 3, 2, 14}
7 {1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14} {1, 7, 13, 6, 12, 5, 11, 4, 10, 3, 9, 2, 8, 14}
8 {1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14} {1, 4, 7, 10, 13, 3, 6, 9, 12, 2, 5, 8, 11, 14}
9 {1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14} {1, 9, 4, 12, 7, 2, 10, 5, 13, 8, 3, 11, 6, 14}
10 {1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14} {1, 5, 9, 13, 4, 8, 12, 3, 7, 11, 2, 6, 10, 14}
11 {1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14} {1, 3, 5, 7, 9, 11, 13, 2, 4, 6, 8, 10, 12, 14}
12 {1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14} {1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14}

Anzahl Mischungen --      Nombre de mélanges:      12

```

**In[43]:= perfektMischenAnzahl[16]**

```

1 {1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16}
   {1, 9, 2, 10, 3, 11, 4, 12, 5, 13, 6, 14, 7, 15, 8, 16}
2 {1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16}
   {1, 5, 9, 13, 2, 6, 10, 14, 3, 7, 11, 15, 4, 8, 12, 16}
3 {1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16}
   {1, 3, 5, 7, 9, 11, 13, 15, 2, 4, 6, 8, 10, 12, 14, 16}
4 {1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16}
   {1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16}

Anzahl Mischungen --      Nombre de mélanges:      4

```

```
In[44]:= perfektMischenAnzahl[18]

1 {1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18}
   {1, 10, 2, 11, 3, 12, 4, 13, 5, 14, 6, 15, 7, 16, 8, 17, 9, 18}

2 {1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18}
   {1, 14, 10, 6, 2, 15, 11, 7, 3, 16, 12, 8, 4, 17, 13, 9, 5, 18}

3 {1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18}
   {1, 16, 14, 12, 10, 8, 6, 4, 2, 17, 15, 13, 11, 9, 7, 5, 3, 18}

4 {1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18}
   {1, 17, 16, 15, 14, 13, 12, 11, 10, 9, 8, 7, 6, 5, 4, 3, 2, 18}

5 {1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18}
   {1, 9, 17, 8, 16, 7, 15, 6, 14, 5, 13, 4, 12, 3, 11, 2, 10, 18}

6 {1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18}
   {1, 5, 9, 13, 17, 4, 8, 12, 16, 3, 7, 11, 15, 2, 6, 10, 14, 18}

7 {1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18}
   {1, 3, 5, 7, 9, 11, 13, 15, 17, 2, 4, 6, 8, 10, 12, 14, 16, 18}

8 {1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18}
   {1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18}

Anzahl Mischungen --      Nombre de mélanges:      8
```

```
In[45]:= (* perfektMischenAnzahl[20] *)
```

```
In[46]:= (* perfektMischenAnzahl[52] *)
```

## Aufgabe 5 ■ Problème 5

### Zum Unterschied zwischen "Block" und "Module" ■ Quant à la différence entre "Block" et "Module"

Betrachte die folgenden Beispiele. Sie sind bis auf die Anweisungen "Block" resp. "Module" identisch. Wieso gibt *Mathematica* verschiedene Resultate aus?

■ Considère les exemples suivants. Ils sont identiques, sauf les directives "Block" resp. "Module". Pourquoi est-ce que *Mathematica* sort des résultats différents?

```
In[47]:= Clear[a, i];
a = i;
i = 3;
Block[{i},
  Table[a, {i, 2}]
]
```

```
Out[50]= {1, 2}
```

```
In[51]:= Clear[a, i];
a = i;
i = 3;
Module[{i},
  Table[a, {i, 2}]
]
```

```
Out[54]= {3, 3}
```

---

## "Putzmaschine" einsetzen

### ■ Employer la "machine de nettoyage"

```
In[55]:= (* Old Form: Remove["Global`*"] *)
```

```
In[56]:= Remove["Global`*"]
```