Kurs ▮ Cours

14. Input und Output
▮ Input et output

Die Gliederung dieses Kurses folgt in groben Zügen dem Buch von Nancy Blachman: A Practical Approach.... Hinweis: Kapitel 14 lesen!
▮ L'articulation de ce cours correspond à peu près à celle du livre de Nancy Blachman: A Practical Approach....
Indication: Lire le chapitre 14.

Run mit WIN+*Mathematica* Version 5.2
▮ Testé avec *Mathematica* version 5.2+WIN

Problem: Datenimport und Datenexport in und von *Mathematica*.
▮ Problème: Importation et exportation de données dans et de *Mathematica*.

14.1. Input
▮ Input

14.1.1. Dateneingabe via Tastatur während des Programmlauf
▮ Entrée des données par les touches pendant que le programme marche

Studiere: ▮ Etudie:

```
In[1]:= ?Input

        Input[ ] interactively reads in one Mathematica expression.
           Input["prompt"] requests input, using the specified string as a prompt. Mehr...

In[2]:= ?InputString

        InputString[ ] interactively reads in a character string.
           InputString["prompt"] requests input, using the specified string as a prompt. Mehr...
```

```
In[3]:= ?BaseForm

        BaseForm[expr, n] prints with the numbers in expr given in base n. Mehr...

In[4]:= BaseForm[15,2]+BaseForm[17,2]

Out[4]= 1111_2 + 10001_2

In[5]:= benutzerInput = Input[
        "Schreibe eine Zahl zwischen 1 und 10: \
        Ecris un nombre entre 1 et 10:"]

Out[5]= 8
```

Achtung: Das nachfolgende Programm bricht erst ab, wenn keine Integer-Zahl eingegeben wird! Probiere:

Attention: Le programme suivant s'interrompt seulement si on n'entre pas le nombre Integer! Essaie:

```
In[6]:= dualUmwandlung:=
            Module[{benutzerInput},
                    While[
                        IntegerQ[
                            userInput = Input[
            "Schreibe eine Zahl zwischen 1 und 10: \
            Ecris un nombre entre 1 et 10:"]
                                ],
                        Print[userInput, " = ",
                                BaseForm[userInput, 2]]
                    ]
            ];

        dualUmwandlung

        1 = 1_2

        2 = 10_2

        3 = 11_2

        4 = 100_2

        5 = 101_2

        6 = 110_2

        7 = 111_2

        8 = 1000_2

        9 = 1001_2
```

(* Input ==> 0.1 !!! *)

```
In[8]:=
```

14.1.2. Datenimport von einem File
Importation de données d'un fichier

Achtung: Achte auf das zu deinem Betriebssystem lauffähige Beispiel!
Der Umgang mit externen Dateien von *Mathematica* aus hat ab Version 3.0 eine

grosse Veränderung erfahren!
■ Attention: Prends seulement les exemples qui tournent sur ton propre système!
Le traitement des fichiers externes depuis *Mathematica* a beaucoup changé depuis la version 3.0!

Beispiel:
(Das File AAAdaten) befindet sich wahrscheinlich in Deinem Verzeichnis C:\Programme\WolframResearch\-*Mathematica*\3.0 (win98). - Sonst musst Du Deinen mir nicht bekannten Pfad suchen!)
■ Exemple:
(Le file AAAdanten.nb (aadaten.ma) se trouve probablement dans le directoire C:\Programme\WolframResearch\-*Mathematica*\3.0 (win98). - Si non tu dois chercher ton "path" que je ne connais pas!)

Mathematica 3.02, Windows98
Beispiel: neues Verzeichnis erstellen:
■ Faire un nouveau directoire:

```
In[239]:=
        CreateDirectory["C:\work\MathematicaData\NewDir"]

Out[239]=
        C:\work\MathematicaData\NewDir
```

```
In[9]:= x = Sin[3] // N >> AAADaten

In[10]:= << AAADaten

Out[10]= 0.14112
```

Beispiel: Erstelle ein Verzeichnis C:\work (falls nicht vorhanden).
(Das File AABDaten befindet sich in Deinem Verzeichnis "work" (win98), (aadaten.ma: "Mate" (win3.1, 95....), "WirzMathematicaFiles/Daten" (NeXT) usw.) . - Sonst musst Du Deinen mir nicht bekannten Pfad angeben!)
■ Exemple: Ouvre un directoire C:\work (si cela n'existe pas encore).
(Le file AABDaten se trouve dans le directoire "work" (win98), (aadaten.ma: "Mate" (win3.1, 95....), "Wirz-MathematicaFiles/Daten" (NeXT) etc.). - Si non tu dois m'indiquer ton "path" que je ne connais pas!)

Schreibe Daten ins File C:\work\AABDaten:
■ Ecrire des données dans le fichier C:\work\AABDaten:

```
In[11]:= "3 4 5 1 2 3 6 6.3 7.8 4 r s t hallo
            7777 6666 4.321 1.11111, {1,2,3,4}" >>
          C:\\work\\MathematicaData\\AABDaten;

In[12]:= {"Cos[3]//N", Cos[3] // N} >> C:\\work\\MathematicaData\\AACDaten;

In[13]:= {3 4 5 1 2 3 6 6, "3 4 5 1 2 3 6 6",
            {3 4 5 1 2 3 6 6}} >> C:\\work\\MathematicaData\\AACDaten;
```

Lese das File C:\work\AABDaten:
■ Lire le fichier C:\work\AABDaten:

```
In[14]:= << C:\\work\\MathematicaData\\AABDaten

Out[14]= 3 4 5 1 2 3 6 6.3 7.8 4 r s t hallo  7777 6666 4.321 1.11111, {1,2,3,4}
```

In[15]:= **!! C:\\work\\MathematicaData\\AABDaten**

```
"3 4 5 1 2 3 6 6.3 7.8 4 r s t hallo  7777 6666 4.321 1.11111, {1,2,3,4}"
```

win98:
Erstelle nötigenfalles eine Textdatei "c:\work\AAEDaten.txt" mit Inhalt:
■ Si nécessaire composer un fichier "c:\work\AAEDaten.txt" avec le contenu:
1 2 3 4 5 6 7
1.1 1.2 1.3 1.4
11 12 13 14 115
a b c d e f g
Sin[5]\\N hallo
"hallo"

Erstelle ebenso eine Textdatei "c:\work\AAFDaten.txt" mit Inhalt:
■ Composer aussi un fichier "c:\work\AAEDaten.txt" avec le contenu:
1 2 3 4 5 6 7
1.1 1.2 1.3 1.4
11 12 13 14 115

Inhalt anschauen:
■ Regarder le contenu:

In[16]:= **!! "C:\\work\\MathematicaData\\AAFdaten.txt"**

```
1234  k34ljök4j8sn3i 43 45 dkr kjkj
fjkldjklf
12345
677890
```

In[17]:= **(* !!f:\Mathe/Daten/aadaten0 (* win3.1 etc. *) *)**

File einlesen und ausführen:
■ Lire le fichier et exécuter:

In[18]:= **<< C:\\work\\MathematicaData\\AAFdaten.txt**

Out[18]= 677890

In[19]:= **(* <<f:\Mathe/Daten/aadaten1 (* win3.1 etc. *) *)**

Was ist los gewesen? Schauen wir weiter:
■ Que s'est-il passé? Continuons d'observer:

In[20]:= **?<<**

```
<<name reads in a file, evaluating each expression in it, and returning the last one.
```
Mehr…

In[21]:= **?Get**

```
<<name reads in a file, evaluating each expression in it, and returning the last one.
```
Mehr…

In[22]:= **!! C:\\work\\MathematicaData\\AAEDaten.txt**

```
1234  99 2 3 4  5 11 1.2  1.12334455
1.123344335
12345
677890
```

In[23]:= **(* !!f:\Mathe/Daten/aadaten1 (* win3.1 etc. *) *)**

In[24]:= **<< C:\\work\\MathematicaData\\AAEDaten.txt**

Out[24]= 677890

In[25]:= **(* <<f:\Mathe/Daten/aadaten1 (* win3.1 etc. *) *)**

In[26]:= **Get["C:\\work\\MathematicaData\\AAEDaten.txt"]**

Out[26]= 677890

In[27]:= **(* Get["f:\Mathe/Daten/aadaten1"]**
(* win3.1 etc. *) *)

In[28]:= **1 7 3 4 2 9 (*Produkt! *)**

Out[28]= 1512

Spezifisch einlesen:

Lire spécifiquement:

In[29]:= **?ReadList**

ReadList["file"] reads all the remaining expressions in a file, and returns a list of them. ReadList["file", type] reads objects of the specified type from a file, until the end of the file is reached. The list of objects read is returned. ReadList["file", {type1, type2, ... }] reads objects with a sequence of types, until the end of the file is reached. ReadList["file", types, n] reads only the first n objects of the specified types. Mehr...

In[31]:= **ReadList["C:\\work\\MathematicaData\\AAEDaten.txt", Real]**

Out[31]= {1234., 99., 2., 3., 4., 5., 11., 1.2, 1.12334, 1.12334, 12345., 677890.}

In[32]:= **(* ReadList["f:\Mathe/Daten/aadaten0", Real] *)**
(* win3.1 etc. *)

In[34]:= **?Real**

Real is the head used for real (floating-point) numbers. Mehr...

In[35]:= **ReadList["C:\\work\\MathematicaData\\AAEDaten.txt", Number]**

Out[35]= {1234, 99, 2, 3, 4, 5, 11, 1.2, 1.12334, 1.12334, 12345, 677890}

In[36]:= **(* ReadList["f:\Mathe/Daten/aadaten0", Number] *)**
(* win3.1 etc. *)

In[38]:= **?Number**

Number represents an exact integer or an approximate real number in Read. Mehr...

In[39]:=
```
ReadList["C:\\work\\MathematicaData\\AAFDaten.txt", Byte]
```

Out[39]=
```
{49, 50, 51, 52, 32, 32, 107, 51, 52, 108, 106, 246, 107, 52, 106,
 56, 115, 110, 51, 105, 32, 52, 51, 32, 52, 53, 32, 100, 107, 114, 32,
 107, 106, 107, 106, 10, 102, 106, 107, 108, 100, 106, 107, 108, 102,
 32, 10, 49, 50, 51, 52, 53, 10, 54, 55, 55, 56, 57, 48, 10, 32, 32, 32}
```

In[40]:=
```
(* ReadList["f:\Mathe/Daten/aadaten0", Byte] *)
                                  (* win3.1 etc. *)
```

In[42]:=
```
?Byte
```

Byte represents a single byte of data in Read. Mehr...

In[43]:=
```
ReadList["C:\\work\\MathematicaData\\AAFDaten.txt", Character]
```

Out[43]=
```
{1, 2, 3, 4,  ,  , k, 3, 4, l, j, ö, k, 4, j, 8,
 s, n, 3, i,  , 4, 3,  , 4, 5,  , d, k, r,  , k, j, k, j,
, f, j, k, l, d, j, k, l, f,  ,
, 1, 2, 3, 4, 5,
, 6, 7, 7, 8, 9, 0,
,  ,  ,  }
```

In[44]:=
```
(* ReadList["f:\Mathe/Daten/aadaten0",
                       Character]  (* win3.1 etc. *) *)
```

In[45]:=
```
?Character
```

Character represents a single character in Read. Mehr...

In[46]:=
```
ReadList["C:\\work\\MathematicaData\\AAFDaten.txt", String]
```

Out[46]=
```
{1234  k34ljök4j8sn3i 43 45 dkr kjkj, fjkldjklf , 12345, 677890,    }
```

In[47]:=
```
(* ReadList["f:\Mathe/Daten/aadaten0", String]
                                  (* win3.1 etc. *) *)
```

In[48]:=
```
?String
```

String is the head of a character string "text". Mehr...

In[49]:=
```
ReadList["C:\\work\\MathematicaData\\AAFDaten.txt", Record]
```

Out[49]=
```
{1234  k34ljök4j8sn3i 43 45 dkr kjkj, fjkldjklf , 12345, 677890,    }
```

In[50]:=
```
(* ReadList["f:\Mathe/Daten/aadaten0", Record] *)
                                  (* win3.1 etc. *)
```

In[52]:=
```
?Record
```

Record represents a record in Read, Find and related functions. Mehr...

In[53]:=
```
ReadList["C:\\work\\MathematicaData\\AAFDaten.txt", Word]
```

Out[53]=
```
{1234, k34ljök4j8sn3i, 43, 45, dkr, kjkj, fjkldjklf, 12345, 677890}
```

In[54]:=
```
(* ReadList["f:\Mathe/Daten/aadaten0", Word] *)
                                  (* win3.1 etc. *)
```

In[56]:=
```
?Word
```

Word represents a word in Read, Find and related functions. Mehr...

```
In[57]:= (* ReadList["f:\Mathe/Daten/aadaten1", Real] *)
                                  (* win3.1 etc. *)

In[59]:=  (* ReadList["f:\Mathe/Daten/aadaten1", Number] *)
                                  (* win3.1 etc. *)

In[61]:= (* ReadList["f:\Mathe/Daten/aadaten2", Number] *)
                                  (* win3.1 etc. *)

In[63]:= (* !!f:\Mathe/Daten/aadaten2    (* win3.1 etc. *) *)

In[64]:= ReadList["C:\\work\\MathematicaData\\AAEDaten.txt", {Number, Number}]

Out[64]= {{1234, 99}, {2, 3}, {4, 5}, {11, 1.2}, {1.12334, 1.12334}, {12345, 677890}}

In[65]:= (* ReadList["f:\Mathe/Daten/aadaten1",
                {Number, Number}] *)    (* win3.1 etc. *)

In[66]:= ReadList["C:\\work\\MathematicaData\\AAEDaten.txt",
         {Number, Number, Number, Number}]

Out[66]= {{1234, 99, 2, 3}, {4, 5, 11, 1.2}, {1.12334, 1.12334, 12345, 677890}}

In[67]:= (* ReadList["f:\Mathe/Daten/aadaten1",
                {Number, Number, Number, Number}] *)
                                  (* win3.1 etc. *)

In[69]:= ReadList["C:\\work\\MathematicaData\\AAEDaten.txt", Cos[Number]]

Out[69]= {Cos[1234], Cos[99], Cos[2], Cos[3], Cos[4], Cos[5],
          Cos[11], 0.362358, 0.43267, 0.43267, Cos[12345], Cos[677890]}

In[70]:= ReadList["C:\\work\\MathematicaData\\AAEDaten.txt",Cos[Number]] // N

Out[70]= {-0.798551, 0.0398209, -0.416147, -0.989992, -0.653644,
          0.283662, 0.0044257, 0.362358, 0.43267, 0.43267, 0.111436, -0.961386}

In[71]:= (* ReadList["f:\Mathe/Daten/aadaten1",
                Cos[Number]] *)     (* win3.1 etc. *)

In[72]:= ReadList["C:\\work\\MathematicaData\\AAFDaten.txt", Cos[Number]] // N

        Read::readn : Syntax error reading a real number from C:\work\MathematicaData\AAFDaten.txt. Mehr…

Out[72]= {-0.798551}

In[73]:= (* ReadList["f:\Mathe/Daten/aadaten1",
                Cos[Number]] // N *)    (* win3.1 etc. *)

In[74]:= (* !!f:\Mathe/Daten/aadaten3 *) (* win3.1 etc. *)

In[75]:= ReadList["C:\\work\\MathematicaData\\AAFDaten.txt",
         {String, Number, Number}]

        Read::readn : Syntax error reading a real number from C:\work\MathematicaData\AAFDaten.txt. Mehr…

Out[75]= {{1234  k34ljök4j8sn3i 43 45 dkr kjkj, $Failed, EndOfFile}}

In[76]:= (* ReadList["f:\Mathe/Daten/aadaten3",
         {String, Number, Number}] *)    (* win3.1 etc. *)

In[77]:= (* !!f:\Mathe/Daten/aadaten4 *) (* win3.1 etc. *)
```

```
In[78]:= ReadList["C:\\work\\MathematicaData\\AAFDaten.txt",
          {Number, Number, String}]
```

```
Read::readn : Syntax error reading a real number from C:\work\MathematicaData\AAFDaten.txt. Mehr…
```

```
Out[78]= {{1234, $Failed, EndOfFile}}
```

```
In[79]:= (*ReadList["f:\Mathe/Daten/aadaten4",
          {Number, Number, String}] *)  (* win3.1 etc. *)
```

```
In[80]:= (*!!f:\Mathe/Daten/aadaten5 *)  (* win3.1 etc. *)
```

```
In[81]:= (* ReadList["f:\Mathe/Daten/aadaten5",
          {Number, Number, String}] *)    (* win3.1 etc. *)
```

Probiere eigene files!

- Essaie tes propres fichiers!

14.1.3. Einlesen von Records und Words
Lire Records et Words

Studiere die "Befehle":

- Etudie les "ordres":

```
In[82]:= ?*Record*
```

System`

```
NullRecords  RecordLists
Record       RecordSeparators
```

```
In[83]:= ?RecordLists
```

```
RecordLists is an option for ReadList which specifies whether objects
   from separate records should be returned in separate sublists. Mehr…
```

```
In[84]:= ?RecordSeparators
```

```
RecordSeparators is an option for Read, Find and related functions which
   specifies the list of strings to be taken as delimiters for records. Mehr…
```

```
In[85]:= ?WordSeparators
```

```
WordSeparators is an option for Read, Find and related functions which
   specifies the list of strings to be taken as delimiters for words. Mehr…
```

```
In[86]:= ?NullRecords
```

```
NullRecords is an option for Read and related functions which specifies whether null
   records should be taken to exist between repeated record separators. Mehr…
```

```
In[87]:= ?NullWords
```

```
NullWords is an option for Read and related functions which specifies whether
   null words should be taken to exist between repeated word separators. Mehr…
```

In[88]:= **?TokenWords**

```
TokenWords is an option for Read and related functions
   which gives a list of token words to be used to delimit words. Mehr...
```

Studiere die Anwendungen:

■ Etudie les applications:

In[89]:= **!!C:\\work\\MathematicaData\\AABDaten**

```
"3 4 5 1 2 3 6 6.3 7.8 4 r s t hallo  7777 6666 4.321 1.11111, {1,2,3,4}"
```

In[90]:= **!! C:\\work\\MathematicaData\\AAGDaten.txt**

```
asdf 5  34 dkfjke jkdf
```

In[91]:=
```
(* !!f:\Mathe/Daten/spsheet0 *) (* win3.1 etc. *)
```

In[92]:=
```
ReadList["C:\\work\\MathematicaData\\AAEDaten.txt",
 {Number, Number, Number}]
```

Out[92]= {{1234, 99, 2}, {3, 4, 5}, {11, 1.2, 1.12334}, {1.12334, 12345, 677890}}

In[93]:=
```
(* ReadList["f:\Mathe/Daten/spsheet0",
 {Number, Number, Number}] *) (* win3.1 etc. *)
```

In[94]:=
```
ReadList["C:\\work\\MathematicaData\\AAFDaten.txt",
 Word, RecordLists -> True]
```

Out[94]= {{1234, k34ljök4j8sn3i, 43, 45, dkr, kjkj}, {fjkldjklf}, {12345}, {677890}}

In[95]:=
```
(*ReadList["f:\Mathe/Daten/spsheet0",
 Word, RecordLists ->True] *) (* win3.1 etc. *)
```

In[96]:=
```
ReadList["C:\\work\\MathematicaData\\AAFDaten.txt",  Word]
```

Out[96]= {1234, k34ljök4j8sn3i, 43, 45, dkr, kjkj, fjkldjklf, 12345, 677890}

In[97]:=
```
(* ReadList["f:\Mathe/Daten/spsheet0",
 Word] *)          (* win3.1 etc. *)
```

In[98]:=
```
ReadList["C:\\work\\MathematicaData\\AAFDaten.txt",
          RecordLists -> True]
```

Out[98]= {{2387790 dkr k34ljök4j8sn3i kjkj}, {fjkldjklf}, {12345}, {677890}, {Null}}

In[99]:=
```
(* ReadList["f:\Mathe/Daten/spsheet0",
       RecordLists ->True] *) (* win3.1 etc. *)
```

In[100]:=
```
Clear[x];
(* x = ReadList["f:\Mathe/Daten/spsheet0",
        Word,
        WordSeparators -> {"\t"},
        RecordLists ->True] *) (* win3.1 etc. *)
```

In[102]:=
```
Clear[x];
x = ReadList["C:\\work\\MathematicaData\\AAGDaten.txt",
        Word,
        WordSeparators -> {"\t"},
        RecordLists -> True]
```

Out[103]=
```
{{asdf 5  34 dkfjke jkdf}, {   }}
```

In[104]:=
```
?Word
```

```
Word represents a word in Read, Find and related functions. Mehr...
```

In[105]:=
```
ToExpression[x]
```

Out[105]=
```
{{170 asdf dkfjke jkdf}, {Null}}
```

In[106]:=
```
?ToExpression
```

```
ToExpression[input] gives the expression obtained by interpreting strings
   or boxes as Mathematica input. ToExpression[input, form] uses interpretation
   rules corresponding to the specified form. ToExpression[input, form, h]
   wraps the head h around the expression produced before evaluating it. Mehr...
```

In[107]:=
```
FullForm[x]
```

Out[107]//FullForm=
```
List[List["asdf 5  34 dkfjke jkdf"], List["   "]]
```

In[108]:=
```
(* ReadList["f:\Mathe/Daten/spsheet1",
        {Word, Number, Number, Number},
        WordSeparators -> {" "},
        RecordLists ->True] *) (* win3.1 etc. *)
```

In[109]:=
```
ReadList["C:\\work\\MathematicaData\\AAGDaten.txt",
        {Word, Number, Number, Number},
        WordSeparators -> {" "},
        RecordLists -> True]
```

```
Read::readn : Syntax error reading a real number from C:\work\MathematicaData\AAGDaten.txt. Mehr…
```

Out[109]=
```
{{{asdf, 5, 34, $Failed}}}
```

In[110]:=
```
FullForm[%]
```

Out[110]//FullForm=
```
List[List[List["asdf", 5, 34, $Failed]]]
```

14.1.3. Ein längeres Beispiel
Un exemple plus long

Zuerst definieren wir Funktionen. Damit kŏnnen wir obigen Befehl "nachbilden":

D'abord nous définissons des fonctions. Avec cela nous pouvons "imiter" l'ordre ci-dessus.

In[111]:=

```
Clear[tabSeparate, tabSep];
tabSeparate[str_String]:= Apply[tabSep,
        Characters[str]];
tabSep[x__String,"\t",y___String]:=
        Flatten[{tabSep[x],tabSep[y]},1];
tabSep[x___String]:=
        ToExpression[StringJoin[x]]/;
        FreeQ[{x},"\t"];
ReadList["C:\\work\\MathematicaData\\AAGDaten.txt",
        tabSeparate[String]]
```

Out[115]=

```
{170 asdf dkfjke jkdf, Null}
```

In[116]:=

```
(*Clear[tabSeparate, tabSep];
tabSeparate[str_String]:=Apply[tabSep,
        Characters[str]];
tabSep[x__String,"\t",y___String]:=
        Flatten[{tabSep[x],tabSep[y]},1];
tabSep[x___String]:=
        ToExpression[StringJoin[x]]/;
        FreeQ[{x},"\t"];
ReadList["f:\Mathe/Daten/spsheet1",
        tabSeparate[String]](* win3.1 etc. *)*)
```

In[117]:=

```
?Characters
```

```
Characters["string"] gives a list of the characters in a string. Mehr…
```

In[118]:=

```
?FreeQ
```

```
FreeQ[expr, form] yields True if no subexpression in expr
   matches form, and yields False otherwise. FreeQ[expr, form, levelspec]
   tests only those parts of expr on levels specified by levelspec. Mehr…
```

Achtung! *Mathematica* ist nicht für Datamanagement und Datamanipulation gebaut. Man kann Daten in speziellen Datamanagement-Sprachen oder Programmen (low-level- Programmierung) viel schneller parsen (einlesen und interpretieren) sowie manipulieren. Dies ist bei grösseren Datenmengen wesentlich. Man denke z. B. an (1000 x 1000)-Matrizen in der Statistik, z.B. bei 1000 Beobachtungen von 1000 Variablen.

Attention: *Mathematica* n'est pas conçu pour le "management" ni pour la manipulation des données. On peut lire et intérpréter ("parser") de même que manipuler beaucoup plus rapidement les données à l'aide de langages "datamanagement" ou programmes (programmation low-level). Cela est essentiel pour des quantités plus importantes de données. On pense p.ex. aux matrices (1000 x 1000) dans la statistique, p.ex. 1000 observations de 1000 variables.

14.1.4. Externe Kommandos bis exklusive Version 3.0 (für WIN98 nicht zu empfehlen)
▮ Commandements extérieurs jusqu'à version 3.0, exclusive (pour WIN98 pas recommandé)

Mit "!Kommando" kann ein externes Kommando ausgeführt werden. Leider haben wir auf NeXT keinen ModulaII-Compiler oder Java-Compiler, so dass wir selbst solche Kommandos bauen könnten. Eine andere Programmiersprache, die den Zweck erfüllen würde, wird an der Schule momentan nicht unterrichtet. Hier ist für NeXT das Programm "aa_ascii", das Umlaute in TeX-Format übersetzt. Studiere das Beispiel:

▮ Par "commandement !" on peut exécuter un commandement extérieur. Malheureusement nous n'avons pas de compilateur pour ModulaII ou Jave sur NeXT pour pouvoir construire de tels commandements. Une autre langue de programmation qui serve pour le même but n'est actuellement pas enseigné à cette école. Ici on utilise pour NeXT le programme "aa_ascii" qui transforme les "Umlaute" (métaphonie, voyelle infléchie) de l'allemand dans le format TeX. Etudie l'exemple:

```
In[119]:=
        !!C:\\work\\MathematicaData\\AABDaten

        "3 4 5 1 2 3 6 6.3 7.8 4 r s t hallo  7777 6666 4.321 1.11111, {1,2,3,4}"

In[120]:=
        (* !!f:\Mathe/Daten/spsheet2 *)

In[121]:=
        !C:\\work\\MathematicaData\\AABDaten

In[122]:=
        (* !f:\Mathe/Daten/aa_ascii
                f:\Mathe/Daten/spsheet2
                f:\Mathe/Daten/tex -tex;
        ReadList["f:\Mathe/Daten/tex",
                {Word, Number, Number, Number},
                 WordSeparators -> {" "},
                 RecordLists ->True] *)
```

Das "!Kommando" kann auch in ReadList stehen, falls es sich auf eine Operation auf einem File bezieht (und nicht zwei Files aufruft)! Folgendes geht nicht:

▮ Le "commandement !" peut se trouver aussi dans ReadList, s'il se rapporte à une opération sur un fichier (et n'appelle pas deux fichiers)! La chose suivante ne va pas:

```
In[123]:=
        (* ReadList["!f:\Mathe/Daten/aa_ascii
                f:\Mathe/Daten/spsheet2
                f:\Mathe/Daten/tex -tex ",
                {Word, Number, Number, Number},
                 WordSeparators -> {" "},
                 RecordLists ->True] *)
```

Externe Kommandos ab Version 3.0
▮ Commandements extérieurs à partir de version 3.0

Verwendung externer Programme für Datenmanipulationen von *Mathematica* aus mit Hilfe der Befehle "Install", "Links", "MLPut...",
"MLGet...", u.s.w. ..... .
Momentan sind keine Programme vorhanden, die problemlos auf irgendwelchen hier unbekannten Systemen laufen. Daher wird auf ein konkretes Beispiel verzichtet. Probiere daher mit eigenen Programmen.

▮ Utiliser des Programmes externes pour manipuler des données depuis *Mathematica* à l'aide des ordres "Install", "Links", "MLPut...",
"MLGet...", etc. .... .
Dans ce moment nous n'avons pas de programmes qui tournent sans problèmes sur n'importe quel système inconnu chez nous. C'est pourquoi nous renonçons à présenter un exemple concret. Essaie avec tes propres programmes.

14.2. Datenexport
▮ Exportation de données

14.2.1. Zeichenweise in ein File schreiben
▮ Ecrire les signes l'un après l'autre dans un fichier

Beispiel
▮ Exemple

Zuerst Daten erzeugen, die exportiert werden sollen:
▮ D'abord générer les données qui doivent être exportées:

```
In[124]:=
Clear[daten];
daten = Table[Random[Integer,{1,9}],{2},{6}]
```

```
Out[125]=
{{3, 1, 3, 2, 2, 3}, {9, 2, 2, 3, 3, 4}}
```

Studiere die Befehle:
▮ Etudie les ordres:

```
In[126]:=
?>>
```

```
expr >> filename writes expr to a file. Put[expr1,
   expr2, ... , "filename"] writes a sequence of expressions expri to a file. Mehr…
```

```
In[127]:=
?Put
```

```
expr >> filename writes expr to a file. Put[expr1,
   expr2, ... , "filename"] writes a sequence of expressions expri to a file. Mehr…
```

In[128]:=
```
??WriteString
```

```
WriteString[channel, expr1, expr2, ... ] converts the expri to strings,
   and then writes them in sequence to the specified output channel. Mehr…

Attributes[WriteString] = {Protected}
```

In[129]:=
```
?OpenWrite
```

```
OpenWrite["file"] opens a file to write output to it, and returns an OutputStream object.
 Mehr…
```

In[130]:=
```
?Close
```

```
Close[stream] closes a stream. Mehr…
```

In[131]:=
```
?Scan
```

```
Scan[f, expr] evaluates f applied to each element of expr in turn. Scan[f,
   expr, levelspec] applies f to parts of expr specified by levelspec. Mehr…
```

In[132]:=
```
?Rest
```

```
Rest[expr] gives expr with the first element removed. Mehr…
```

Zum Exportieren wollen wir eine Funktion definieren, die die Generierte in der Art Matrix "Zeilenvektor auf Zeile ... (Record) im File" schreibt:

■ Nous allons définir une fonction pour exporter, qui écrit la générée dans la matrice dans le sens "vecteur de ligne sur ligne .... (Record) dans le fichier":

In[133]:=
```
Remove[schreibMatrix];
Off[neufile];
schreibMatrix[fileName_String, data_List] :=
    Block[{ neufile = OpenWrite[fileName]},
        Scan[(WriteString[neufile,First[#]];
                  Scan[WriteString[neufile,
                  "\t",#]&,Rest[#]];
             WriteString[neufile,"\n"])&,
             data];
        Close[neufile]];
```

In[136]:=
```
daten = {{1, 2}, {3, 4, 5, 6}, {7, 8, 9, 10, a}, {b}}
```

Out[136]=
```
{{1, 2}, {3, 4, 5, 6}, {7, 8, 9, 10, a}, {b}}
```

In[137]:=
```
schreibMatrix["C:\\work\\MathematicaData\\AAHDaten", daten]
```

Out[137]=
```
C:\work\MathematicaData\AAHDaten
```

In[138]:=
```
(* schreibMatrix["f:\Mathe/Daten/neuDaten", daten] (* win3.1 etc. *) *)
```

In[139]:=

```
!! "C:\\work\\MathematicaData\\AAHDaten"
```

```
1   2
3   4   5   6
7   8   9   10  a
b
```

In[140]:=

```
(* !!f:\Mathe/Daten/neuDaten *) (* win3.1 etc. *)
```

In[141]:=

```
?schreibMatrix
```

```
Global`schreibMatrix

schreibMatrix[fileName_String, data_List] :=
 Block[{neufile = OpenWrite[fileName]}, Scan[(WriteString[neufile, First[#1]];
     Scan[WriteString[neufile,     , #1] &, Rest[#1]]; WriteString[neufile,
])&, data]; Close[neufile]]
```

14.2.2. Abspeichern von Definitionen
▮ Mémoriser des définitions

Beispiel
▮ Exemple

Studiere: ▮ Etudie:

In[142]:=

```
?Save
```

```
Save["filename", symbol] appends definitions associated with the specified symbol to a
   file. Save["filename", "form"] appends definitions associated with all symbols
   whose names match the string pattern form. Save["filename", "context`"] appends
   definitions associated with all symbols in the specified context. Save["filename",
   {object1, object2, ... }] appends definitions associated with several objects. Mehr...
```

In[143]:=

```
f[x_] := Sin[x^2] / (E^(-x) Cos[x]);
Save["C:\\work\\MathematicaData\\AA_f", f]
```

In[145]:=

```
(* f[x_] := Sin[x^2]/(E^(-x) Cos[x]);
Save["f:\Mathe/Daten/AAA_f", f]
                          (* win3.1 etc. *) *)
```

In[146]:=

```
!! C:\\work\\MathematicaData\\AA_f
```

```
f[x_] := Sin[x^2]/(Cos[x]/E^x)

x = {{"dddt  ejkrjke   dfjj  asdf  \\t dkfjke \\t jkdf"}, {"   "}}
f[x_] := Sin[x^2]/(Cos[x]/E^x)

x = {{"dddt  ejkrjke   dfjj  asdf  \\t dkfjke \\t jkdf"}, {"   "}}
f[x_] := Sin[x^2]/(Cos[x]/E^x)

x = {{"asdf 5  34 dkfjke jkdf"}, {"   "}}
```

In[147]:=
```
(* !!f:\Mathe/Daten/AAA_f  (* win3.1 etc. *) *)
```

14.2.3.Abspeichern von Output
▮ Mémoriser l'output

Studiere: ▮ Etudie:

In[148]:=
```
?>
```

```
x > y yields True if x is determined to be greater than y. x1 >
   x2 > x3 yields True if the xi form a strictly decreasing sequence. Mehr…
```

In[149]:=
```
?>>
```

```
expr >> filename writes expr to a file. Put[expr1,
   expr2, ... , "filename"] writes a sequence of expressions expri to a file. Mehr…
```

In[150]:=
```
?>>>
```

```
expr >>> filename appends expr to a file. PutAppend[expr1,
   expr2, ... , "filename"] appends a sequence of expressions expri to a file. Mehr…
```

In[151]:=
```
?Put
```

```
expr >> filename writes expr to a file. Put[expr1,
   expr2, ... , "filename"] writes a sequence of expressions expri to a file. Mehr…
```

Probiere aus: ▮ Essaie:

In[152]:=
```
Table[f[n], {n, 20}] >>
    C:\\work\\MathematicaData\\AAneuDaten
```

In[153]:=
```
(* Table[f[n], {n, 20}] >>
    f:\Mathe/Daten/neuDaten  (* win3.1 etc. *) *)
```

In[154]:=
```
!! C:\\work\\MathematicaData\\AAneuDaten
```

```
{E*Tan[1], E^2*Sec[2]*Sin[4], E^3*Sec[3]*Sin[9], E^4*Sec[4]*Sin[16],
 E^5*Sec[5]*Sin[25], E^6*Sec[6]*Sin[36], E^7*Sec[7]*Sin[49],
 E^8*Sec[8]*Sin[64], E^9*Sec[9]*Sin[81], E^10*Sec[10]*Sin[100],
 E^11*Sec[11]*Sin[121], E^12*Sec[12]*Sin[144], E^13*Sec[13]*Sin[169],
 E^14*Sec[14]*Sin[196], E^15*Sec[15]*Sin[225], E^16*Sec[16]*Sin[256],
 E^17*Sec[17]*Sin[289], E^18*Sec[18]*Sin[324], E^19*Sec[19]*Sin[361],
 E^20*Sec[20]*Sin[400]}
```

In[155]:=
```
(* !!f:\Mathe/Daten/neuDaten   (* win3.1 etc. *) *)
```

In[156]:=
```
Put[Table[f[n], {n, 20}], Sin[x], "Hallo",
 "C:\\work\\MathematicaData\\AAneuDaten"]
```

In[157]:=
```
(* Put[Table[f[n], {n, 20}], Sin[x], "Hallo",
 "f:\Mathe/Daten/neuDaten"] *)  (* win3.1 etc. *)
```

In[158]:=
```
!! C:\\work\\MathematicaData\\AAneuDaten
```

```
{E*Tan[1], E^2*Sec[2]*Sin[4], E^3*Sec[3]*Sin[9], E^4*Sec[4]*Sin[16],
 E^5*Sec[5]*Sin[25], E^6*Sec[6]*Sin[36], E^7*Sec[7]*Sin[49],
 E^8*Sec[8]*Sin[64], E^9*Sec[9]*Sin[81], E^10*Sec[10]*Sin[100],
 E^11*Sec[11]*Sin[121], E^12*Sec[12]*Sin[144], E^13*Sec[13]*Sin[169],
 E^14*Sec[14]*Sin[196], E^15*Sec[15]*Sin[225], E^16*Sec[16]*Sin[256],
 E^17*Sec[17]*Sin[289], E^18*Sec[18]*Sin[324], E^19*Sec[19]*Sin[361],
 E^20*Sec[20]*Sin[400]}
{{Sin["asdf 5  34 dkfjke jkdf"]}, {Sin["   "]}}
"Hallo"
```

In[159]:=
```
(* !!f:\Mathe/Daten/neuDaten *)  (* win3.1 etc. *)
```

Was ist mit den Daten von 14.2.3 passiert?
Aue s'est-il passé avec les données de 14.2.3?

In[160]:=
```
2.^0.5
```

Out[160]=
```
1.41421
```

In[161]:=
```
2.^0.5 >> C:\\work\\MathematicaData\\AAneuDaten1
```

In[162]:=
```
(* 2.^0.5 >> f:\Mathe/Daten/neuDaten  *)
                                (* win3.1 etc. *)
```

In[164]:=
```
!! C:\\work\\MathematicaData\\AAneuDaten1
```

```
1.4142135623730951
```

In[165]:=
```
(* !!f:\Mathe/Daten/neuDaten *)   (* win3.1 etc. *)
```

Wieso ist das abgespeicherte Resultat genauer als das auf dem Schirm ausgegebene Resultat?
- Pourquoi le résultat mémorisé est-il plus exact que le résultat sorti sur l'écran?

14.3. Manipulation von Strings
Manipulation de Strings

Studiere die folgenden Befehle:
- Etudie les ordres suivants:

In[166]:=
```
?Read
```

```
Read[stream] reads one expression from an input stream, and returns the expression.
   Read[stream, type] reads one object of the specified type. Read[stream, {type1,
   type2, ... }] reads a sequence of objects of the specified types. Mehr...
```

In[167]:=

?ReadList

ReadList["file"] reads all the remaining expressions in a file, and returns a list of them. ReadList["file", type] reads objects of the specified type from a file, until the end of the file is reached. The list of objects read is returned. ReadList["file", {type1, type2, ... }] reads objects with a sequence of types, until the end of the file is reached. ReadList["file", types, n] reads only the first n objects of the specified types. Mehr...

In[168]:=

?OpenAppend

OpenAppend["file"] opens a file to append output to it, and returns an OutputStream object. Mehr...

In[169]:=

?OpenRead

OpenRead["file"] opens a file to read data from, and returns an InputStream object. Mehr...

In[170]:=

?StringDrop

StringDrop["string", n] gives "string" with its first n characters dropped. StringDrop["string", -n] gives "string" with its last n characters dropped. StringDrop["string", {n}] gives "string" with its nth character dropped. StringDrop["string", {m, n}] gives "string" with characters m through n dropped. StringDrop[{s1, s2, ... }, spec] gives the list of results for each of the si. Mehr...

In[171]:=

?StringInsert

StringInsert["string", "snew", n] yields a string with "snew" inserted starting at position n in "string". StringInsert["string", "snew", -n] inserts at position n from the end of "string". StringInsert["string", "snew", {n1, n2, ... }] inserts a copy of "snew" at each of the positions ni. StringInsert[{s1, s2, ... }, "snew", n] gives the list of results for each of the si. Mehr...

In[172]:=

?StringJoin

"s1" <> "s2" <> ... , StringJoin["s1", "s2", ...] or StringJoin[{"s1", "s2", ... }] yields a string consisting of a concatenation of the si. Mehr...

In[173]:=

?StringReplace

StringReplace["string", s -> sp] or StringReplace["string", {s1 -> sp1, s2 -> sp2, ... }] replaces the string expressions si by spi whenever they appear as substrings of "string". StringReplace["string", srules, n] does only the first n replacements. StringReplace[{s1, s2, ... }, srules] gives the list of results for each of the si. Mehr...

In[174]:=

?StringReverse

StringReverse["string"] reverses the order of the characters in "string". Mehr...

In[175]:=

?StringTake

StringTake["string", n] gives a string containing the first n characters in "string". StringTake["string", -n] gives the last n characters in "string". StringTake["string", {n}] gives the nth character in "string". StringTake["string", {m, n}] gives characters m through n in "string". StringTake[{s1, s2, ... }, spec] gives the list of results for each of the si. Mehr...

In[176]:=

?StringLength

StringLength["string"] gives the number of characters in a string. Mehr...

In[177]:=

?StringPosition

StringPosition["string", "sub"] gives a list of the starting and ending character positions at which "sub" appears as a substring of "string". StringPosition["string", patt] gives all positions at which substrings matching the general string expression patt appear in "string". StringPosition["string", patt, n] includes only the first n occurrences of patt. StringPosition["string", {patt1, patt2, ... }] gives positions of all the patti. StringPosition[{s1, s2, ... }, p] gives the list of results for each of the si. Mehr...

In[178]:=

?StringMatchQ

StringMatchQ["string", patt] tests whether string matches the string pattern patt. StringMatchQ["string", RegularExpression["regex"]] tests whether string matches the specified regular expression. StringMatchQ[{s1, s2, ... }, p] gives the list of results for each of the si. Mehr...

In[179]:=

?StringQ

StringQ[expr] gives True if expr is a string, and False otherwise.

In[180]:=

?StringToStream

StringToStream["string"] opens an input stream for reading from a string. Mehr...

In[181]:=

?WriteString

WriteString[channel, expr1, expr2, ...] converts the expri to strings, and then writes them in sequence to the specified output channel. Mehr...

In[182]:=

?ToString

ToString[expr] gives a string corresponding to the printed form of expr in OutputForm. ToString[expr, form] gives the string corresponding to output in the specified form. Mehr...

Probiere aus: ■ Essaie:

In[183]:=

```
Table[StringJoin["daten.", ToString[i]],{i,6}]
```

Out[183]=

```
{daten.1, daten.2, daten.3, daten.4, daten.5, daten.6}
```

14.4. Eingabe- und Ausgabeform
▮ Formes d'entrée et de sortie

Studiere die folgenden Befehle:
▮ Etudie les ordres suivants:

In[184]:=
```
?InputForm
```
```
InputForm[expr] prints as a version of expr suitable for input to Mathematica. Mehr...
```

In[185]:=
```
?OutputForm
```
```
OutputForm[expr] prints as a two-dimensional
   representation of expr using only keyboard characters. Mehr...
```

Beispiele: ▮ Exemples:

In[186]:=
```
Clear[a,x,y];
a = x^5/x^2 y +x^y^5
```

Out[187]=
$$x^{y^5} + x^3\, y$$

In[188]:=
```
InputForm[a]
```

Out[188]//InputForm=
```
x^y^5 + x^3*y
```

In[189]:=
```
OutputForm[a]
```

Out[189]//OutputForm=
```
 5
y      3
x  + x  y
```

In[190]:=
```
a >> C:\\work\\MathematicaData\\AAneuDaten2
```

In[191]:=
```
(* a >> f:\Mathe/Daten/neuDaten *)  (* win3.1 etc. *)
```

In[192]:=
```
!! C:\\work\\MathematicaData\\AAneuDaten2
```
```
x^y^5 + x^3*y
```

In[193]:=
```
(* !!f:\Mathe/Daten/neuDaten *)   (* win3.1 etc. *)
```

In[194]:=
```
Read["C:\\work\\MathematicaData\\AAneuDaten2"]
```

Out[194]=
$$x^{y^5} + x^3\, y$$

```
In[195]:=
(* Read["f:\Mathe/Daten/neuDaten"](* win3.1 etc. *) *)

In[196]:=
b = Read["C:\\work\\MathematicaData\\AAneuDaten2"]

Out[196]=
EndOfFile

In[197]:=
(* b = Read["f:\Mathe/Daten/neuDaten"] *)
                                (* win3.1 etc. *)

In[199]:=
b

Out[199]=
EndOfFile

In[200]:=
!! C:\\work\\MathematicaData\\AAneuDaten2

x^y^5 + x^3*y

In[201]:=
(* !!f:\Mathe/Daten/neuDaten *)   (* win3.1 etc. *)

In[202]:=
Close["C:\\work\\MathematicaData\\AAneuDaten2"]

Out[202]=
C:\work\MathematicaData\AAneuDaten2

In[203]:=
(* Close["f:\Mathe/Daten/neuDaten"] *)
                                (* win3.1 etc. *)

In[205]:=
b = Read["C:\\work\\MathematicaData\\AAneuDaten2"]

Out[205]=
x^(y^5) + x^3 y

In[206]:=
(* b = Read["f:\Mathe/Daten/neuDaten"] *)
                                (* win3.1 etc. *)

In[208]:=
Close["C:\\work\\MathematicaData\\AAneuDaten2"]

Out[208]=
C:\work\MathematicaData\AAneuDaten2

In[209]:=
(* Close["f:\Mathe/Daten/neuDaten"] *)
                                (* win3.1 etc. *)
```

14.5. Uebersetzen von Ausdrücken in andere Programmiersprachen und Manipulation von Sourcecode

∎ Traduire des expressions en d'autres languages de programmation et manipulation de sourcecode

Uebersetzungen
∎ Tradcuction

Studiere die folgenden Befehle:
∎ Etudie les ordres suivants:

```
In[211]:=
```
```
?CForm
```
```
CForm[expr] prints as a C language version of expr. Mehr...
```

```
In[212]:=
```
```
?FortranForm
```
```
FortranForm[expr] prints as a Fortran language version of expr. Mehr...
```

```
In[213]:=
```
```
?TeXForm
```
```
TeXForm[expr] prints as a TeX version of expr. Mehr...
```

```
In[214]:=
```
```
?Splice
```
```
Splice["file"] splices Mathematica output into an external file. It takes text
   enclosed between <* and *> in the file, evaluates the text as Mathematica
   input, and replaces the text with the resulting Mathematica output. Mehr...
```

Beispiele: ∎ Exemple:

```
In[215]:=
```
```
Clear[a];
a = x^5/z^2 y +x^y^5 +
                ( x^2 Sin[x^5/z^2 y +x^y^5]);
```

```
In[217]:=
```
```
CForm[a]
```

```
Out[217]//CForm=
```
```
Power(x,Power(y,5)) + (Power(x,5)*y)/Power(z,2) +
   Power(x,2)*Sin(Power(x,Power(y,5)) +
      (Power(x,5)*y)/Power(z,2))
```

```
In[218]:=
```
```
FortranForm[a]
```

```
Out[218]//FortranForm=
```
```
            x**y**5 + (x**5*y)/z**2 +
          -  x**2*Sin(x**y**5 + (x**5*y)/z**2)
```

```
In[219]:=
```
```
TeXForm[a]
```

```
Out[219]//TeXForm=
```
```
x^{y^5}+\frac{y x^5}{z^2}+\sin
   \left(x^{y^5}+\frac{y x^5}{z^2}\right) x^2
```

Manipulation von Sourcecode
▮ Manipulation de sourcecode

Im folgenden Beispiel wird gezeigt, wie in den Source-Code eines externen Programms (z.B. in C) direkt *Mathematica*-Ausdrücke eingesetzt und dann von *Mathematica* übersetzt werden kŏnnen. Der externe Programmteil befindet sich im File AAA.mc. Das übersetzte Programm wird in AAA.c gespeichert!!!!!!!

▮ Dans l'exemple suivant on montre comme on peut mettre des expressions de *Mathematica* directement dans le Source-Code d'un autre programme extérieur (p.ex. dand C), ensuite ses expressions peuvent être traduites par *Mathematica*. La partie extérieur du programme se trouve dans le fichier AAA.mx resp. AAA.mc. Le programme traduit est mémorisé dans AAA.x resp. AAA.c !!!!!!!

In[220]:=
```
Sin[3] + Cos[3] >> C:\\work\\MathematicaData\\AAA.mx
```

In[221]:=
```
!! C:\\work\\MathematicaData\\AAA.mx

Cos[3] + Sin[3]
```

In[222]:=
```
(* !!f:\Mathe/Daten/AAA.mc  *)  (* win3.1 etc. *)
```

In[223]:=
```
?Splice

Splice["file"] splices Mathematica output into an external file. It takes text
   enclosed between <* and *> in the file, evaluates the text as Mathematica
   input, and replaces the text with the resulting Mathematica output. Mehr...
```

In[224]:=
```
Splice["C:\\work\\MathematicaData\\AAA.mx"]
```

Out[224]=
```
C:\work\MathematicaData\AAA.mx
```

In[225]:=
```
(* Splice["f:\Mathe/Daten/AAA.mc"]  *)
                                (* win3.1 etc. *)
```

In[227]:=
```
!! C:\\work\\MathematicaData\\AAA.mx

Cos[3] + Sin[3]
```

In[228]:=
```
(* !!f:\Mathe/Daten/AAA.c *) (* win3.1 etc. *)
```

14.6. Formate
Formats

Allgemeines
Généralités

Studiere die folgenden Befehle:
Etudie les ordres suivants:

```
In[229]:=
?Subscripted
```

```
System`Subscripted
```

```
In[230]:=
?Superscript
```

```
Superscript[x, y] is an object that formats as x with a superscript y. Mehr...
```

```
In[231]:=
?SequenceForm
```

```
SequenceForm[expr1, expr2, ... ] prints as the
   textual concatenation of the printed forms of the expri. Mehr...
```

Probiere aus: Etudie:

```
In[232]:=
Subscripted[abc[145]]
```

```
Out[232]=
abc145
```

```
In[233]:=
SequenceForm[uvw, Superscript[251]]
```

```
Out[233]=
uvw251
```

Ein Anwendungsbeispiel
Un exemple d'application

```
In[234]:=
Remove[a, n, m];
Format[a[n_, m_]] := Subscripted[a[n, m]];
Array[a, {2, 2}] // MatrixForm
```

```
Out[236]//MatrixForm=
(a1,1  a1,2)
(a2,1  a2,2)
```

"Putzmaschine" einsetzen
▮ Employer la "machine de nettoyage"

In[237]:=
```
(* Old Form: Remove["Global`@*"] *)
```

In[238]:=
```
Remove["Global`*"]
```