

Kurs ■ Cours

15. Packages

Die Gliederung dieses Kurses folgt in groben Zügen dem Buch von Nancy Blachman: A Practical Approach.... Hinweis: Kapitel 15 lesen!

- L'articulation de ce cours correspond à peu près à celle du livre de Nancy Blachman: A Practical Approach....
Indication: Lire le chapitre 15.

Run mit WIN+*Mathematica* Version 5.2

- Testé avec *Mathematica* version 5.2+WIN

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

Problem: Externe Programme und Programmpackete schreiben.

Mathematica selbst ist in C geschrieben. Externe Programme, sogenannte Packages, sind aber in *Mathematica* selbst geschrieben. Bsp.: "Calculus`DSolveIntegrals`"

- Problème: Ecris les programmes extérieurs et des paquets de programmes. *Mathematica* est écrit en C. Les programmes extérieurs, les dits Packages, sont écrits dans *Mathematica* même. Ex.:
"Calculus`DSolveIntegrals`"

15.1. Wieso Packages? Wie zugreifen?

■ Pourquoi des Packages? Comment les saisir?

15.1.1. Wieso Packages?

■ Pourquoi des Packages?

Es ist klar, dass man häufig gebrauchte Funktionen oder Prozeduren etc. nicht immer wieder neu schreibt. Andere möchten auch gerne profitieren. So schreibt man eben externe Programme, die nach Bedarf zuladbar sind.

- Il est clair qu'on n'écrit pas chaque fois à nouveau des fonctions ou des procédures etc. fréquentes. Les autres aimeraient aussi en profiter. Ainsi on écrit des programmes extérieurs qu'on peut ajouter quand on en a besoin.

15.1.2. Wie greife ich auf ein mitgeliefertes Package?

■ Comment saisir un Package livré

Die mitgelieferten Packages sieht man unter (Info, Open Function Browser, Packages) neu Help, Add-ons, Standard Packages, Nach dem Anklicken (Evaluate Template und dann Loaded Packages, Update Packages kann man im Browser das geladene Package anklicken und sieht die darin enthaltenen Funktionen) neu COPY-PASTE.

■ Les Packages livrés se trouvent sous (Info, Open Function Browser, Packages) nouveau Help, Add-ons, Standard Packages, Après avoir marqué à la souris (Evaluate Template et ensuite Loaded Packages, Update Packages on peut marquer à la souris le Package chargé dans le Browser, on y voit les fonctions qui y sont contenues.) nouveau COPY-PASTE.

Lade das nachfolgende Package und schaue im Browser nach, welche Funktionen darin enthalten sind (Browser: Loaded Packages, Update Packages, anklicken etc.) neu Help, Add-ons, Standard Packages,

■ Charge le Package suivant et regarde dans le Browser quelles fonctions y sont contenues. (Browser: Loaded Packages, Update Packages, marquer à la souris etc.) nouveau Help, Add-ons, Standard Packages,

```
In[1]:= Needs["Statistics`DescriptiveStatistics`"]
```

```
In[2]:= ? Mean
```

Mean[list] gives the statistical mean of the elements in list. Mehr...

```
In[3]:= <<Algebra`FiniteFields`
```

```
In[4]:= ?GF
```

GF[p, ilist][elist] represents an element of a finite field (Galois field) with prime characteristic p and where the coefficients of the irreducible polynomial are given by ilist (constant term listed first). The particular element is specified by elist, the list of coefficients of the polynomial representation of the element. GF[p, d], p prime and d a positive integer, is a special form giving a Galois field that is a degree d extension of the prime field of p elements. The resulting field has p^d elements. Mehr...

15.2. Contexts

■ Contextes

15.2.1. Das Wesen des Contexts

■ L'essence du contexte

Ein Context ist vergleichbar mit einem Directory (Verzeichnis) oder einem Subdirectory (Folder). An die Namen der Variablen oder Symbolen wird dort der Contextname angehängt, so dass es sich dann um lokale Objekte handelt. Das verhindert Namenkollision. Contexte können geschachtelt sein. Bisher haben wir immer im globalen Context gearbeitet. Probiere aus:

■ Un contexte est comparable à un répertoire (index) ou à un sous-répertoire (Subdirectory, Folder). On y ajoute le nom du contexte aux noms des variables ou des symboles, de façon qu'il s'agisse ensuite d'objets locaux. Cela évite des collisions de noms. Les contextes peuvent être emboîtés. Jusqu'à présent nous avons toujours travaillé dans le contexte global. Essaie:

In[5]:= ?\$Path

\$Path gives the default list of directories to search in attempting to find an external file.
Mehr...

In[6]:= \$Path

Out[6]= {C:\Programme\Wolfram Research\Mathematica\5.2\AddOns\JLink,
C:\Programme\Wolfram Research\Mathematica\5.2\AddOns\NETLink,
C:\Dokumente und Einstellungen\Rolf\Anwendungsdaten\Mathematica\Kernel,
C:\Dokumente und Einstellungen\Rolf\Anwendungsdaten\Mathematica\Autoload,
C:\Dokumente und Einstellungen\Rolf\Anwendungsdaten\Mathematica\Applications,
C:\Dokumente und Einstellungen\All Users\Anwendungsdaten\Mathematica\Kernel,
C:\Dokumente und Einstellungen\All Users\Anwendungsdaten\Mathematica\Autoload,
C:\Dokumente und Einstellungen\All Users\Anwendungsdaten\Mathematica\Applications,
. , C:\Dokumente und Einstellungen\Rolf,
C:\Programme\Wolfram Research\Mathematica\5.2\AddOns\StandardPackages,
C:\Programme\Wolfram Research\Mathematica\5.2\AddOns\StandardPackages\Startup,
C:\Programme\Wolfram Research\Mathematica\5.2\AddOns\Autoload,
C:\Programme\Wolfram Research\Mathematica\5.2\AddOns\Applications,
C:\Programme\Wolfram Research\Mathematica\5.2\AddOns\ExtraPackages,
C:\Programme\Wolfram Research\Mathematica\5.2\SystemFiles\Graphics\Packages,
C:\Programme\Wolfram Research\Mathematica\5.2\Configuration\Kernel}

15.2.2. Der Context-Pfad

■ Le chemin ("path") du contexte

Studiere die Befehle:

■ Etudie les ordres:

In[7]:= ?Context

Context[] gives the current context.
Context[symbol] gives the context in which a symbol appears. Mehr...

In[8]:= ?\$Context

\$Context is a global variable that gives the current context. Mehr...

In[9]:= ?\$ContextPath

\$ContextPath is a global variable that gives a list of contexts, after
\$Context, to search in trying to find a symbol that has been entered. Mehr...

In[10]:= ?*Context*

System`

Context	ContextToFileName	\$Context
Contexts	GetContext	\$ContextPath
ContextToFilename	SetFileLoadingContext	

```
In[11]:= ?Dir*
```

System`

```
DiracDelta      Direction DirectoryName
DirectedInfinity Directory DirectoryStack
```

```
In[12]:= ?Directory
```

Directory[] gives the current working directory. Mehr...

Probiere aus: ■ Essaie:

```
In[13]:= $ContextPath
```

```
Out[13]= {Algebra`FiniteFields`, Algebra`PolynomialPowerMod`,
          NumberTheory`NumberTheoryFunctions`, NumberTheory`FactorIntegerECM`,
          Statistics`DescriptiveStatistics`, Statistics`DataManipulation`, Global`, System`}
```

```
In[14]:= Context[ParametricPlot]
```

```
Out[14]= System`
```

```
In[15]:= Context[Integrate]
```

```
Out[15]= System`
```

15.2.2. Context und Symbole

■ Contexte et symboles

Beispiele

■ Essaie

Studiere: ■ Etudie:

```
In[16]:= x = 2;
          Remove[a];
          {Context[x], Context[a], Context[Tuten]}
```

```
Out[18]= {Global`, Global`, Global`}
```

In welchem Context sind die geschaffenen Symbole?

■ Dans quel contexte se trouvent les symboles créés?

```
In[19]:= Benzin = 50 Liter;
          Auto`Benzin =
          "Interessiert den Umweltschutz. --
          Interessant pour la protection naturelle.";
```

```
In[21]:= ?Benzin
```

```
Global`Benzin
```

```
Benzin = 50 Liter
```

```
In[22]:= ?*`Benzin
```

Global`

Benzin

Auto`

Benzin

```
In[23]:= ?Auto`Benzin
```

Auto`Benzin

Auto`Benzin = Interessiert den Umweltschutz. -- Interessant pour la protection naturelle.

Wird der Context "Global" angegeben bei einer Abfrage?

Teste das aus:

■ Est-ce que le contexte "Global" est iniqué lors d'une interrogation? Fais un test:

```
In[24]:= `StudentError`Hoppla =
        "Das war wohl nicht so gemeint!
        Ce n'est certainement pas l'intention!";
```

```
In[25]:= `StudentError`Hoppla
```

```
Out[25]= Das war wohl nicht so gemeint! Ce n'est certainement pas l'intention!
```

```
In[26]:= Global`StudentError`Hoppla
```

```
Out[26]= Das war wohl nicht so gemeint! Ce n'est certainement pas l'intention!
```

15.3. Context-Wechsel

■ Changement de contexte

15.3.1. Allgemeines

■ Généralités

Dies ist möglich mit der Variable \$Context. Studiere:

■ Cela est possible avec la variable \$Context. Etudie:

```
In[27]:= Benzin
```

```
Out[27]= 50 Liter
```

```
In[28]:= $Context = "Auto`"
```

```
Out[28]= Auto`
```

```
In[29]:= Benzin
```

```
Out[29]= Interessiert den Umweltschutz. -- Interessant pour la protection naturelle.
```

```
In[30]:= $Context
```

```
Out[30]= Auto`
```

```
In[31]:= $Context = "Global`"
```

```
Out[31]= Global`
```

```
In[32]:= Benzin
```

```
Out[32]= 50 Liter
```

15.3.2. Der Befehl "Begin"

■ L'ordre "Begin"

Mit den Befehlen Begin, BeginPackage, End, EndPackage wird der Context gewechselt. Studiere:

■ Par les ordres Begin, BeginPackage, End, EndPackage on change de contexte. Etudie:

```
In[33]:= ?Begin
```

```
Begin["context`"] resets the current context. Mehr...
```

```
In[34]:= ?BeginPackage
```

```
BeginPackage["context`"] makes context` and System` the only active contexts.
```

```
BeginPackage["context`", {"need1`", "need2`", ...}] calls Needs on the needi. Mehr...
```

```
In[35]:= ?End
```

```
End[ ] returns the present context, and reverts to the previous one. Mehr...
```

```
In[36]:= ?EndPackage
```

```
EndPackage[ ] restores $Context and $ContextPath to their values before the preceding
```

```
BeginPackage, and prepends the current context to the list $ContextPath. Mehr...
```

Probiere aus: ■ Essaie:

```
In[37]:= $Context
```

```
Out[37]= Global`
```

```
In[38]:= $ContextPath
```

```
Out[38]= {Algebra`FiniteFields`, Algebra`PolynomialPowerMod`,
          NumberTheory`NumberTheoryFunctions`, NumberTheory`FactorIntegerECM`,
          Statistics`DescriptiveStatistics`, Statistics`DataManipulation`, Global`, System`}
```

```
In[39]:= Begin["Versuch`"]
```

```
Out[39]= Versuch`
```

```
In[40]:= $Context
```

```
Out[40]= Versuch`
```

```
In[41]:= $ContextPath
```

```
Out[41]= {Algebra`FiniteFields`, Algebra`PolynomialPowerMod`,
          NumberTheory`NumberTheoryFunctions`, NumberTheory`FactorIntegerECM`,
          Statistics`DescriptiveStatistics`, Statistics`DataManipulation`, Global`, System`}
```

Nun führen wir eine Variable ein:

■ Nous introduisons une variable:

```
In[42]:= dumm = "frech und beschränkt --
               mal poli et borné"
```

```
Out[42]= frech und beschränkt -- mal poli et borné
```

```
In[43]:= Context[dumm]
```

```
Out[43]= Versuch`
```

```
In[44]:= End[ ]
```

```
Out[44]= Versuch`
```

```
In[45]:= $Context
```

```
Out[45]= Global`
```

```
In[46]:= $ContextPath
```

```
Out[46]= {Algebra`FiniteFields`, Algebra`PolynomialPowerMod`,
          NumberTheory`NumberTheoryFunctions`, NumberTheory`FactorIntegerECM`,
          Statistics`DescriptiveStatistics`, Statistics`DataManipulation`, Global`, System`}
```

```
In[47]:= ?dumm
```

```
Information::notfound : Symbol dumm not found. Mehr...
```

```
In[48]:= ?Versuch`dumm
```

```
Versuch`dumm
```

```
Versuch`dumm = frech und beschränkt -- mal poli et borné
```

In welchem Context sind wir jetzt und wie ist der Pfad gesetzt?

■ Dans quel contexte sommes-nous maintenant et comment le chemin est-il posé?

15.3.3. Der Befehl "Begin"

■ L'ordre "Begin"

Probiere aus: ■ Essaie:

```
In[49]:= BeginPackage["Hallowach`"]
```

```
Out[49]= Hallowach`
```

```
In[50]:= $Context
```

```
Out[50]= Hallowach`
```

```
In[51]:= $ContextPath
```

```
Out[51]= {Hallowach`, System`}
```

Was ist der Unterschied zwischen Begin und BeginPackage?

Probiere jetzt früher definierte Variablen aus:

■ Quelle est la différence entre Begin et BeginPackage?

Essaie maintenant des variables définies il y a un certain temps:

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

```
Information::notfound : Symbol Benzin not found. Mehr...
```

```
In[53]:= ?Global`Benzin
```

```
Global`Benzin
```

```
Global`Benzin = 50 Global`Liter
```

```
In[54]:= dumm
```

```
Out[54]= dumm
```

```
In[55]:= dumm =
"blöd und lieb zur Hälfte und beides
zusammen -- idiot et aimable à moitié
ou toutes les deux ensembles"
```

```
Out[55]= blöd und lieb zur Hälfte und beides  zusammen --
        idiot et aimable à moitié ou toutes les deux ensembles
```

```
In[56]:= EndPackage[]
```

```
In[57]:= $Context
```

```
Out[57]= Global`
```

```
In[58]:= $ContextPath
```

```
Out[58]= {Hallowach`, Algebra`FiniteFields`, Algebra`PolynomialPowerMod`,
        NumberTheory`NumberTheoryFunctions`, NumberTheory`FactorIntegerECM`,
        Statistics`DescriptiveStatistics`, Statistics`DataManipulation`, Global`, System`}
```

```
In[59]:= dumm
```

```
Out[59]= blöd und lieb zur Hälfte und beides  zusammen --
        idiot et aimable à moitié ou toutes les deux ensembles
```

```
In[60]:= Context[dumm]
```

```
Out[60]= Hallowach`
```

```
In[61]:= {Versuch`dumm, dumm}
```

```
Out[61]= {frech und beschränkt --    mal poli et borné,
        blöd und lieb zur Hälfte und beides  zusammen --
        idiot et aimable à moitié ou toutes les deux ensembles}
```

15.4. Zum Laden von Packages

■ Pour charger des Packages

15.4.1. Befehle

■ Ordres

Zur Verfügung stehen die Befehle <<, Get und Needs.

Studiere die Befehle:

■ Les ordres suivants sont à disposition: <<, Get et Needs.

Etudie les ordres:

In[62]:= ?<<

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

In[63]:= ?Get

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

In[64]:= ?Needs

Needs["context`"] loads an appropriate file if the specified context is not already in \$Packages. Needs["context`", "file"] loads file if the specified context is not already in \$Packages. Mehr...

In[65]:= ?\$Path

\$Path gives the default list of directories to search in attempting to find an external file. Mehr...

In[66]:= \$Path

```
Out[66]= {C:\Programme\Wolfram Research\Mathematica\5.2\AddOns\JLink,
C:\Programme\Wolfram Research\Mathematica\5.2\AddOns\NETLink,
C:\Dokumente und Einstellungen\Rolf\Anwendungsdaten\Mathematica\Kernel,
C:\Dokumente und Einstellungen\Rolf\Anwendungsdaten\Mathematica\Autoload,
C:\Dokumente und Einstellungen\Rolf\Anwendungsdaten\Mathematica\Applications,
C:\Dokumente und Einstellungen>All Users\Anwendungsdaten\Mathematica\Kernel,
C:\Dokumente und Einstellungen>All Users\Anwendungsdaten\Mathematica\Autoload,
C:\Dokumente und Einstellungen>All Users\Anwendungsdaten\
Mathematica\Applications, ., C:\Dokumente und Einstellungen\Rolf,
C:\Programme\Wolfram Research\Mathematica\5.2\AddOns\StandardPackages,
C:\Programme\Wolfram Research\Mathematica\5.2\AddOns\StandardPackages\StartUp,
C:\Programme\Wolfram Research\Mathematica\5.2\AddOns\Autoload,
C:\Programme\Wolfram Research\Mathematica\5.2\AddOns\Applications,
C:\Programme\Wolfram Research\Mathematica\5.2\AddOns\ExtraPackages,
C:\Programme\Wolfram Research\Mathematica\5.2\SystemFiles\Graphics\Packages,
C:\Programme\Wolfram Research\Mathematica\5.2\Configuration\Kernel}
```

Hinweis: Die Variable \$Path wird im File init.m gesetzt! (C:\Programme\Wolfram Research\Mathematica\3.0\Configuration\Kernel.) Dieses File kann von einem selbst modifiziert werden. (NeXT: Es soll sich bei privaten Installationen im eigenen Root befinden.) Beim Start von *Mathematica* wird es gelesen.

■ Retenir: La variable \$Path est mise dans le fichier init.m! (C:\Programme\Wolfram Research\Mathematica\3.0-

Configuration\Kernel.) On peut modifier soi-même ce fichier. (NeXT: Il doit se trouver dans le propre Root quand il s'agit d'installations privées.) Il est lu quand on démarre *Mathematica*.

15.4.2. Laden mit Get resp. << ■ Charger avec Get resp. <<

Bemerkung: "Packages" befindet sich in \$Path! Führe die folgenden Befehle nacheinander aus:

■ Remarque: "Packages" se trouve dans \$Path! Exécute les ordres suivants l'un après l'autre:

```
In[67]:= $Path
```

```
Out[67]= {C:\Programme\Wolfram Research\Mathematica\5.2\AddOns\JLink,
C:\Programme\Wolfram Research\Mathematica\5.2\AddOns\NETLink,
C:\Dokumente und Einstellungen\Rolf\Anwendungsdaten\Mathematica\Kernel,
C:\Dokumente und Einstellungen\Rolf\Anwendungsdaten\Mathematica\Autoload,
C:\Dokumente und Einstellungen\Rolf\Anwendungsdaten\Mathematica\Applications,
C:\Dokumente und Einstellungen\All Users\Anwendungsdaten\Mathematica\Kernel,
C:\Dokumente und Einstellungen\All Users\Anwendungsdaten\Mathematica\Autoload,
C:\Dokumente und Einstellungen\All Users\Anwendungsdaten\
Mathematica\Applications, ., C:\Dokumente und Einstellungen\Rolf,
C:\Programme\Wolfram Research\Mathematica\5.2\AddOns\StandardPackages,
C:\Programme\Wolfram Research\Mathematica\5.2\AddOns\StandardPackages\StartUp,
C:\Programme\Wolfram Research\Mathematica\5.2\AddOns\Autoload,
C:\Programme\Wolfram Research\Mathematica\5.2\AddOns\Applications,
C:\Programme\Wolfram Research\Mathematica\5.2\AddOns\ExtraPackages,
C:\Programme\Wolfram Research\Mathematica\5.2\SystemFiles\Graphics\Packages,
C:\Programme\Wolfram Research\Mathematica\5.2\Configuration\Kernel}
```

```
In[68]:= ?Helix
```

```
Information::notfound : Symbol Helix not found. Mehr...
```

```
In[69]:= Remove[Helix]
```

```
In[70]:= << Graphics/Shapes.m;
?Helix
```

```
Helix[{r:1, h:0.5, {m:2, n:20r}}] is a list of n*m
polygons approximating a helix with half height h and m turns. Mehr...
```

```
In[72]:= $ContextPath
```

```
Out[72]= {Graphics`Shapes`, Geometry`Rotations`, HalloWach`, Algebra`FiniteFields`,
Algebra`PolynomialPowerMod`, NumberTheory`NumberTheoryFunctions`,
NumberTheory`FactorIntegerECM`, Statistics`DescriptiveStatistics`,
Statistics`DataManipulation`, Global`, System`}
```

```
In[73]:= << Graphics`Shapes`;
?Helix
```

```
Helix[{r:1, h:0.5, {m:2, n:20r}}] is a list of n*m
polygons approximating a helix with half height h and m turns. Mehr...
```

```
In[75]:= << Graphics/Shapes.m;
?Helix
```

```
Helix[{r:1, h:0.5, {m:2, n:20r}}] is a list of n*m
polygons approximating a helix with half height h and m turns. Mehr...
```

```
In[77]:= Needs["Graphics`Shapes`"]

In[78]:= $ContextPath

Out[78]= {Graphics`Shapes`, Geometry`Rotations`, HalloWach`, Algebra`FiniteFields`,
          Algebra`PolynomialPowerMod`, NumberTheory`NumberTheoryFunctions`,
          NumberTheory`FactorIntegerECM`, Statistics`DescriptiveStatistics`,
          Statistics`DataManipulation`, Global`, System`}
```

15.4.3. Laden mit Needs ■ Charger avec Needs

Was ist der Unterschied gewesen zwischen Get und Needs?

■ Quelle était la différence entre Get et Needs?

```
In[79]:= Get["Graphics`Shapes`"]

In[80]:= Needs["Graphics`Shapes`"]
```

Versuche: ■ Essaie:

```
In[81]:= BeginPackage["Graphics`Shapes`",
                      "Geometry`Rotations`"]

Out[81]= Graphics`Shapes`

In[82]:= $ContextPath

Out[82]= {Graphics`Shapes`, Geometry`Rotations`, System`}
```

15.4.3. Wie sieht man den Inhalt eines Packages (Funktionen)? ■ Comment voit-on le contenu d'un Package (fonctions?)

Versuche: ■ Essaie:

```
In[83]:= ?Rotate2D

Rotate2D[vec, theta, {pt:{0,0}}] rotates the vector vec by angle theta about point pt. Mehr...

In[84]:= ?Geometry`Rotations`Rotate2D

Rotate2D[vec, theta, {pt:{0,0}}] rotates the vector vec by angle theta about point pt. Mehr...

In[85]:= ?Graphics`Shapes`Helix (*Old Vers.*)

Helix[{r:1, h:0.5, {m:2, n:20r}}] is a list of n*m
polygons approximating a helix with half height h and m turns. Mehr...

In[86]:= ?Geometry`Rotations`*
```

Geometry`Rotations`

Rotate2D Rotate3D RotationMatrix2D RotationMatrix3D

```
In[87]:= ?Graphics`Shapes`*      (*Old Vers.*)
```

Graphics`Shapes`

AffineShape	Helix	RotateShape	TranslateShape
Cone	MoebiusStrip	ShrinkPolygons	WireFrame
Cylinder	OutlinePolygons	Sphere	
DoubleHelix	PerforatePolygons	Torus	

```
In[88]:= ?Geometry`*`*`
```

Geometry`Rotations`

Rotate2D Rotate3D RotationMatrix2D RotationMatrix3D

Geometry`Rotations`Private`

phi psi pt theta vec

Achtung: private bedeutet "zugriffgeschützt"! Es wird keine spezifische Information abgegeben. Beispiel:

■ Attention: Privé signifie "ne peut pas être saisi"! On ne donne pas d'information spécifique. Exemple:

```
In[89]:= ?Geometry`Rotations`Private`phi
```

```
Geometry`Rotations`Private`phi
```

```
In[90]:= ?Geometry`Rotations`Private`*`
```

Geometry`Rotations`Private`

phi psi pt theta vec

```
In[91]:= ?Geometry`Rotations`Private`vec
```

```
Geometry`Rotations`Private`vec
```

Welche Packages sind da? Wir suchen z.B. ein Objekt, in dessen Namen hinter dem "" ein "a" vorkommt:

■ Quels Packages y a-t-il? Nous cherchons p.ex. un objet, dans le nom duquel il y a un "" suivi d'un "a" :

```
In[92]:= ?*`a`*
```

Global`

a

Algebra`FiniteFields`Private`

a a\$

BoxForm`

a apat\$ args arguments arg\$
 apat arg args\$ arguments\$ a\$

FE`

a allrules alphabetrules any

Graphics`Shapes`Private`

a averagepoints

Integrate`

aaaaa

NumberTheory`FactorIntegerECM`Private`

a

NumberTheory`NumberTheoryFunctions`Private`

a ac alen alen\$ any args as as\$ a\$

Solve`

a

Statistics`DescriptiveStatistics`Private`

a

System`Convert`MathMLDump`

annotations

System`Dump`

a	alltags\$	arg2	asp\$
addmessagebutton	any	arg2\$	axes
addmessagebuttons	applyUsageStyles	argname	axes\$
allowprotect	arg	args	a\$
allowprotect\$	arg1	args\$	
alltags	arg1\$	asp	

System`FEDump`

allNotebooks allNotebooks\$ arg args

System`HypergeometricDump`

a

System`LanguageEnhancements`

a

System`Private`

auto

15.5. Der Package-Stil

■ Le style Package

Studiere die folgenden Beispiele konsequent durch!

■ Fais une étude approfondie des exemples suivants!

Hier zwei Beispiele aus der *Mathematica*-Bibliothek

■ Voici deux exemples de la bibliothèque de *Mathematica*:

Beispiel 1 (Kopie aus der old Library, etwas kommentiert)

■ Exemple 1 (Copie de la old Library, un peu commentée)

(*****)

Adapted from

Roman E. Maeder: Programming in Mathematica,

Second Edition, Addison-Wesley, 1991.

Wichtig: Gute Dokumentation! ■ Important: Bonne documentation!

*****)

BeginPackage["Graphics`ArgColors`"]

(* Wichtig: \$Context und \$ContextPath wechseln! *)

(* ■ Important: Changer \$Context et \$ContextPath ! *)

(* Wichtig: Die hier definierten Funktionen, z.B. ArgColor, sind *)

(* nach aussen sichtbar. Die Symbole dafür werden in diesem *)

(* Context kreiert. *)

(* ■ Important: Les fonctions définies ici, p.ex. ArgColor, sont *)

(* visibles à l'extérieur. Les symboles en sont créés dans ce *)

(* contexte. *)

ArgColor::usage = "ArgColor[z] gives a color value whose hue is proportional
to the argument of the complex number z."

ArgShade::usage = "ArgShade[z] gives a gray level proportional
to the argument of the complex number z."

ColorCircle::usage = "ColorCircle[r, (light:1)] gives a color value whose hue
is proportional to r (mod 2Pi) with lightness light."

(* Wichtig: usage-Statement für alle aussen sichtbaren Funktionen *)

(* ■ Important: usage-Statement visibles pour toutes les fonctions *)

(* visibles à l'extérieur. *)

Begin["`Private`"]

(* Wichtig: \$Context wechselt auf `Private`, *)

(* ■ Important: \$Context devient `Private`, *)

(* \$ContextPath bleibt ■ reste. *)

(* Wichtig: Hier neu definierte Funktionen sind aussen unsichtbar! *)

(* ■ Important: Les fonctions définies ici nouvellement ne sont pas *)

(* visibles à l'extérieur! *)

HelperFunction[z_] := Print["Mich siehst Du von aussen nicht", z]

```
ArgColor[z_] := RGBColor[1, 1, 1] /; z == 0.0
```

```
ArgColor[z_] := ColorCircle[ Arg[z] ] /; NumberQ[N[arg]]
```

```
ArgShade[z_] := GrayLevel[1] /; z == 0.0
```

```
ArgShade[z_] := GrayLevel[N[(Pi + Arg[z])/(2Pi)] ] /; NumberQ[N[Arg[z]]]
```

```
ColorCircle[arg_, light_:1.0] := Hue[ N[arg/2/Pi], 1, light] /; NumberQ[N[arg]]
```

```
End[ ]
```

```
(* Wichtig: $Context wechselt zurück *)
```

```
(* auf den Wert vor dem Aufruf von `Private` *)
```

```
(* ■ Important: $Context reprend la forme d'origine *)
```

```
(* sur la valeur d'avant l'appel de `Private` *)
```

```
Protect[ArgColor, ArgShade, ColorCircle]
```

```
(* Wichtig: Die aussen sichtbaren Funktionen werden geschützt. *)
```

```
(* ■ Important: Les fonctions visibles à l'extérieur sont protégées. *)
```

```
EndPackage[ ]
```

```
(* Wichtig: $Context wird zurückgewechselt auf den alten Wert. *)
```

```
(* Wichtig: Zu $ContextPath wird der Name *)
```

```
(* "Graphics`ArgColors`" hinzugefügt. *)
```

```
(* ■ Important: $Context reprend la valeur d'origine. *)
```

```
(* ■ Important: On ajoute à $ContextPath le nom *)
```

```
(* "Graphics`ArgColors`" . *)
```

```

In[93]:= (*****

Adapted from
Roman E. Maeder: Programming in Mathematica,
Second Edition, Addison-Wesley, 1991.
(Obige Kopie aus der Library etwas abgeändert,
so dass sie auf NeXT... lauffähig ist.
■ La copie de Library ci dessus un peu changée,
de façon qu'elle puisse fonctionner sur NeXT
.....)

*****

BeginPackage["Test`"];

argColor1::usage =
"ArgColor[z] gives a color value whose hue is
proportional;
to the argument of the complex number z."
argShade1::usage =
"ArgShade[z] gives a gray level proportional
to the argument of the complex number z.";

colorCircle1::usage =
"ColorCircle[r, (light:1)] gives a color value
whose hue is proportional to r (mod 2Pi)
with lightness light.";

Begin["`Private`"];

argColor1[z_] := RGBColor[1, 1, 1] /;
z == 0.0 ;

argColor1[z_] := ColorCircle[ Arg[z] ] /;
NumberQ[N[Arg]] ;

argShade1[z_] := GrayLevel[1] /; z == 0.0
argShade1[z_] := GrayLevel[N[(Pi +
Arg[z])/(2Pi)] ] /;
NumberQ[N[Arg[z]]] ;

colorCircle1[arg_, light_:1.0] :=
Hue[ N[arg/2/Pi], 1, light] /;
NumberQ[N[Arg]] ;

End[];

Protect[argColor1, argShade1, colorCircle1]

EndPackage[];

Out[95]= ArgColor[z] gives a color value whose hue
is proportional; to the argument of the complex number z.

Out[105]=
{argColor1, argShade1, colorCircle1}

In[107]:=
(*****

Adapted from

```

Roman E. Maeder: Programming in Mathematica,
Second Edition, Addison-Wesley, 1991.
(Obige Kopie aus der Library etwas abgeändert,
so dass sie auf NeXT.. lauffähig ist, jetzt mit
Abfrage der Context-Variablen und des Context-
Pfades.

■ La copie de Library ci dessus un peu changée,
de façon qu'elle puisse fonctionner sur NeXT
.., maintenant avec interrogation (appel) des
variables du contexte et du chemin du
contexte.)

*****)

```
Print["1. Abfrage: $ContextPath = ",
      $ContextPath, " $Context = ", $Context];
```

```
BeginPackage["Test`"];
Print["2. Abfrage: $ContextPath = ",
      $ContextPath, " $Context = ", $Context];
```

```
argColor1::usage =
"ArgColor[z] gives a color value whose hue is
proportional;
to the argument of the complex number z."
argShade1::usage =
"ArgShade[z] gives a gray level proportional
to the argument of the complex number z.";
```

```
colorCircle1::usage = "ColorCircle[r,
(light:1)] gives a color value
whose hue is proportional to r (mod 2Pi)
with lightness light.";
```

```
Begin["`Private`"];
Print["3. Abfrage: $ContextPath = ",
      $ContextPath, " $Context = ", $Context];
```

```
argColor1[z_] := RGBColor[1, 1, 1] /;
z == 0.0 ;
```

```
argColor1[z_] := ColorCircle[ Arg[z] ] /;
NumberQ[N[Arg]] ;
```

```
argShade1[z_] := GrayLevel[1] /;
z == 0.0
argShade1[z_] := GrayLevel[N[(Pi +
Arg[z])/(2Pi)] ] /;
NumberQ[N[Arg[z]]] ;
```

```
colorCircle1[arg_, light_:1.0] :=
Hue[ N[arg/2/Pi], 1, light] /;
NumberQ[N[Arg]] ;
```

```
End[];
Print["4. Abfrage: $ContextPath = ",
      $ContextPath, " $Context = ", $Context];
```

```
Protect[argColor1, argShade1, colorCircle1]
```

```
EndPackage[];
Print["5. Abfrage: $ContextPath = ",
```

```
$ContextPath, " $Context = ", $Context];
```

```
1. Abfrage: $ContextPath =  
{Test`, Graphics`Shapes`, Geometry`Rotations`, System`} $Context = Graphics`Shapes`
```

```
2. Abfrage: $ContextPath = {Test`, System`} $Context = Test`
```

```
Out[111]=
```

```
ArgColor[z] gives a color value whose hue  
is proportional; to the argument of the complex number z.
```

```
3. Abfrage: $ContextPath = {Test`, System`} $Context = Test`Private`
```

```
SetDelayed::write : Tag argColor1 in argColor1[z_] is Protected. Mehr...
```

```
SetDelayed::write : Tag argColor1 in argColor1[z_] is Protected. Mehr...
```

```
SetDelayed::write : Tag argShade1 in argShade1[z_] is Protected. Mehr...
```

```
Out[118]=
```

```
$Failed
```

```
SetDelayed::write : Tag argShade1 in argShade1[z_] is Protected. Mehr...
```

```
SetDelayed::write : Tag colorCircle1 in colorCircle1[arg_, light_:1.] is Protected. Mehr...
```

```
4. Abfrage: $ContextPath = {Test`, System`} $Context = Test`
```

```
Out[123]=
```

```
{}
```

```
5. Abfrage: $ContextPath =  
{Test`, Graphics`Shapes`, Geometry`Rotations`, System`} $Context = Graphics`Shapes`
```

Beispiel 2 (Kopie aus der Library, ohne Kommentar)**■ Exemple 2 (Copie de Library, sans commentaire)**

(*****

Copyright 1991 by Roman E. Maeder

Adapted from

Roman E. Maeder: Programming in Mathematica,
Second Edition, Addison-Wesley, 1991.

Permission is hereby granted to make copies of this file for
any purpose other than direct profit, or as part of a
commercial product, provided this copyright notice is left
intact. Sale, other than for the cost of media, is prohibited.

Permission is hereby granted to reproduce part or all of
this file, provided that the source is acknowledged.

*****)

(* Enclosed with Mathematica by permission *)

```
BeginPackage["Examples`Collatz`"]
```

```
Collatz::usage =
```

```
"Collatz[n] gives a list of the iterates in the 3n+1 problem,  
starting from n. The conjecture is that this sequence always  
terminates."
```

```
StoppingTime::usage = "StoppingTime[n] finds the total stopping time  
of the integer n. This is the length of the Collatz sequence  
before hitting 1."
```

```
FindMaxima::usage = "FindMaxima[from] reports successive maxima of the  
total stopping time starting the search with the integer from."
```

```
Begin["`Private`"]
```

```
Collatz[n_Integer] := AppendCollatz[{}, n]
```

```
AppendCollatz[sofar_, 1] := Flatten[{sofar, 1}]
```

```
AppendCollatz[sofar_, n_Integer] :=  
AppendCollatz[{sofar, n}, 3 n + 1] /; OddQ[n]
```

```
AppendCollatz[sofar_, n_Integer] :=
```

```

AppendCollatz[{sofar, n}, n / 2]    /; EvenQ[n] && n != 0

StoppingTime[n_Integer?Positive] :=
  Module[{i=1, m=n}, While[m != 1, m = If[OddQ[m], 3m+1, m/2]; i++;i ]

FindMaxima[low_] :=
  Module[{m=0, n=0, i=low, j},
    While[ True,
      j = StoppingTime[i];
      If[j > m, m = j; n = i;
        Print["StoppingTime[", n, "] = ", m] ];
      i++;
      If[ Mod[i, 100] == 0, Print["i = ", i] ]
    ]
  ]

End[]

Protect[ Collatz, StoppingTime, FindMaxima ]

EndPackage[ ]

```

15.6. Namenskonflikte

■ Conflits de noms

15.6.1. Beispiel eines Konflikts

■ Exemple d'un conflit

Mehrfach verwendete Namen in verschiedenen Umgebungen

■ Noms utilisés de plusieurs façons dans des environs différents.

Studiere die folgende Abfolge:

■ Etudier la suite suivante:

```

In[126]:=
  LaplaceTransform[(1+4t+5t^2) Exp[-3t], t, s]

```

```

Out[126]=
  
$$\frac{10}{(3+s)^3} + \frac{4}{(3+s)^2} + \frac{1}{3+s}$$


```

Die gerufene Funktion war noch unbekannt. Das Symbol LaplaceTransform ist aber jetzt kreiert worden, allerdings ohne konkreten Inhalt.

■ La fonction appelée était encore inconnue. Le symbole LaplaceTransform a été créé maintenant, cependant sans contenu concret.

```
In[127]:=
?LaplaceTransform

LaplaceTransform[expr, t, s] gives the Laplace
transform of expr. LaplaceTransform[expr, {t1, t2, ... }, {s1,
s2, ... }] gives the multidimensional Laplace transform of expr. Mehr...
```

Laden des Packages:

■ Charger le Package:

```
In[128]:=
Needs["Calculus`LaplaceTransform`"]

Get::noopen : Cannot open Calculus`LaplaceTransform`. Mehr...

Needs::nocont : Context Calculus`LaplaceTransform` was not created when Needs was evaluated. Mehr...

Out[128]=
$Failed
```

```
In[129]:=
?LaplaceTransform

LaplaceTransform[expr, t, s] gives the Laplace
transform of expr. LaplaceTransform[expr, {t1, t2, ... }, {s1,
s2, ... }] gives the multidimensional Laplace transform of expr. Mehr...
```

```
In[130]:=
<<Calculus`FourierTransform`
```

```
In[131]:=
NFourierTransform[Exp[-t^2]/(1 + Abs[t]), t, 1]
```

```
Out[131]=
0.404024 + 0. i
```

Die in Global` vorher kreierte Funktion überschattet immer noch die in einem "Unter-Context" zugeladene Funktion mit demselben Namen.

■ La fonction créée auparavant dans Global` recouvre encore la fonction du même nom qui se trouve dans un sous-contexte.

```
In[132]:=
Remove[FourierTransform]

Remove::rmptc : Symbol FourierTransform is Protected and cannot be removed. Mehr...
```

Die in Global` existierende Funktion wird eliminiert. Die darunter liegende zeigt sich jetzt.

■ La fonction existant dans un Global` est élininée. Celle qui se trouve dessous devient visible.

```
In[133]:=
?FourierTransform

FourierTransform[expr, t,  $\omega$ ] gives the symbolic
Fourier transform of expr. FourierTransform[expr, {t1, t2, ... }, { $\omega$ 1,
 $\omega$ 2, ... }] gives the multidimensional Fourier transform of expr. Mehr...
```

Nun kann man sie auch verwenden:

■ Maintenant on peut aussi l'utiliser:

```
In[134]:=
    FourierTransform[(1+4t+5t^2), t, ω]

Out[134]=
     $\sqrt{2\pi} \text{DiracDelta}[\omega] - 4i\sqrt{2\pi} \text{DiracDelta}'[\omega] - 5\sqrt{2\pi} \text{DiracDelta}''[\omega]$ 
```

Was gibt es sonst noch im geladenen Package?

■ Qu'y a-t-il encore dans le Package chargé?

```
In[135]:=
    ?Calculus`FourierTransform`
```

Calculus`FourierTransform`

DTFourierTransform	NFourierCosSeriesCoefficient
FourierCoefficient	NFourierCosTransform
FourierCosCoefficient	NFourierExpSeries
FourierCosSeriesCoefficient	NFourierExpSeriesCoefficient
FourierExpSeries	NFourierSeries
FourierExpSeriesCoefficient	NFourierSinCoefficient
FourierFrequencyConstant	NFourierSinSeriesCoefficient
FourierOverallConstant	NFourierSinTransform
FourierSample	NFourierTransform
FourierSeries	NFourierTrigSeries
FourierSinCoefficient	NInverseDTFourierTransform
FourierSinSeriesCoefficient	NInverseFourierCoefficient
FourierTrigSeries	NInverseFourierCosTransform
InverseDTFourierTransform	NInverseFourierSinTransform
InverseFourierCoefficient	NInverseFourierTransform
NDTFourierTransform	\$FourierFrequencyConstant
NFourierCoefficient	\$FourierOverallConstant
NFourierCosCoefficient	

15.6.2. Der empfohlene Stil

■ Le style conseillé

Im Folgenden wird ein Prototyp eines Packages demonstriert, das ursprünglich von Henry Cejtin und Theo Gray stammen soll:

■ Dans ce qui suit on démontre un prototype d'un Package qui a été créé, soit-disant, par Henry Cejtin et Theo Gray.

```
In[136]:=
    (* OpenWrite["C:\\work\\MathematicaData\\AAKurs15.m"];
    WriteString["C:\\work\\MathematicaData\\AAKurs15.m", "\n",
    "Du sollst das Package von Hand in das File",
    "\n", "AAKurs15.m kopieren."
    ]; (* End WriteString etc. - continued .... *) *)

    OpenWrite["AAKurs15.m"];
    WriteString["AAKurs15.m", "\n",
    "Du sollst das Package von Hand in das File",
    "\n", "AAKurs15.m kopieren."
    ]; (* End WriteString etc. - continued .... *)
```

```

In[138]:=
" " >> AAKurs15.m;

In[139]:=
OpenWrite[
  "AAKurs15.m"];
WriteString[
  "AAKurs15.m",
  "\n",
  "!!!!!!
Du sollst das Package von Hand in das File",
  "\n", "AAKurs15.m kopieren."
  "Il faut copier le Package à la main dans le
  File", "\n", "AAKurs15.m kopieren." ];
(* End WriteString *)

Close["AAKurs15.m"];

(*
 * AAKurs15.m
 *
 * This package demonstrates the recommended
 * package style.
 *)

(*
 * Prepend the context ProtoNotebook` to
 * $ContextPath.
 * Then Needs will be able to determine when
 * the package
 * has been loaded.
 *)
BeginPackage["AAKurs15`"];
EndPackage[];

(*
 * Now the usage statements.
 *)
ExamplePlot::usage = "ExamplePlot[n,m] draws \
  n starbursts and m spirals. \
  ExamplePlot[n] draws n starbursts and
  no spirals.";

SamplePlot::usage =
  "SamplePlot is Vaporware. It \
  does nothing.";

(*
 * Now the global objects.
 *)
ExamplePlot;
SamplePlot;

(*
 * Now the private context. The objects here
 * are not \
 * intended to be accessed by the user.
 *)
Begin["AAKurs15`Private`"];

```

```

(*
 * Local objects.
 *)
`starBurst;
`spiral;

(*
 * Define starBurst and spiral.
 *)
starBurst[center_, radius_, n_] :=
  Block[
    {`r },
    Table[
      Line[{center,
        center + radius {Cos[r],
          Sin[r]}
        }],
      {r, 0, 2Pi - 2Pi/n, 2Pi/n}
    ]
  ];

spiral[center_, radius_, loops_] :=
  Block[
    {`t },
    Line[
      Table[center +
        radius t {Cos[t], Sin[t]}/(2Pi
          loops),
        {t, 0, 2Pi loops, 2Pi/30}
      ]
    ]
  ];

ExamplePlot[stars_] :=
  Show[
    Graphics[
      Table[
        starBurst[
          4 {Random[], Random[]},
          0.3 + Random[],
          20
        ],
        {stars}
      ]
    ],
    AspectRatio -> Automatic
  ];

ExamplePlot[stars_, spins_] :=
  Show[
    Graphics[{
      Table[
        starBurst[
          4 {Random[], Random[]},
          0.3 + Random[],
          20
        ],
        {stars}
      ],
      Table[
        spiral[
          4 {Random[], Random[]},

```

```

        0.3 + Random[],
        Random[Integer, {2, 5}]
    ],
    {spins}
]
}},
AspectRatio -> Automatic
];

(*
 * Protect now the function and lock the private
 * functions.
 *)
Protect[ExamplePlot, SamplePlot];
Append[Attributes[ExamplePlot], Locked];
Append[Attributes[SamplePlot], Locked];

End[];

General::aofil : AAKurs15.m already open as AAKurs15.m. Mehr...

OpenWrite::noopen : Cannot open AAKurs15.m. Mehr...

Syntax::newl:
The newline character after
"
           radius t {Cos[t], Sin[t]}/(2Pi"
is understood as a multiplication operator. Mehr...
```

Schaue ins File ■ Regarde le fichier

C:\Programme\Wolfram Research\Mathematica\3.0\AAKurs15.nb !!!!!!!!!!!!!

```

In[175]:=
Needs["AAKurs15`"]

In[176]:=
Needs["C:\\work\\MathematicaData\\AAKurs15.m"]

Needs::cxt : Invalid context specified at position 1 in Needs[C:\\work\\MathematicaData\\AAKurs15.m].
A context must consist of valid symbol names separated by and ending with `. Mehr...

Out[176]=
Needs[C:\\work\\MathematicaData\\AAKurs15.m]

In[177]:=
<< AAKurs15.m

General::noopen : Cannot open Du sollst das Package von Hand in das File. Mehr...

Syntax::sntx:
Syntax error in or before
"AAKurs15.m kopieren. Il faut copier le Package a` la main".
dans le File
^
(line 6 of "AAKurs15.m") Mehr...

In[178]:=
? AA*

Graphics`Shapes`AAKurs15
```

In[179]:=

\$Path

Out[179]=

```
{C:\Programme\Wolfram Research\Mathematica\5.2\AddOns\JLink,
C:\Programme\Wolfram Research\Mathematica\5.2\AddOns\NETLink,
C:\Dokumente und Einstellungen\Rolf\Anwendungsdaten\Mathematica\Kernel,
C:\Dokumente und Einstellungen\Rolf\Anwendungsdaten\Mathematica\Autoload,
C:\Dokumente und Einstellungen\Rolf\Anwendungsdaten\Mathematica\Applications,
C:\Dokumente und Einstellungen\All Users\Anwendungsdaten\Mathematica\Kernel,
C:\Dokumente und Einstellungen\All Users\Anwendungsdaten\Mathematica\Autoload,
C:\Dokumente und Einstellungen\All Users\Anwendungsdaten\Mathematica\Applications,
., C:\Dokumente und Einstellungen\Rolf,
C:\Programme\Wolfram Research\Mathematica\5.2\AddOns\StandardPackages,
C:\Programme\Wolfram Research\Mathematica\5.2\AddOns\StandardPackages\StartUp,
C:\Programme\Wolfram Research\Mathematica\5.2\AddOns\Autoload,
C:\Programme\Wolfram Research\Mathematica\5.2\AddOns\Applications,
C:\Programme\Wolfram Research\Mathematica\5.2\AddOns\ExtraPackages,
C:\Programme\Wolfram Research\Mathematica\5.2\SystemFiles\Graphics\Packages,
C:\Programme\Wolfram Research\Mathematica\5.2\Configuration\Kernel}
```

In[180]:=

Needs["AAKurs15`"]

In[181]:=

?SamplePlot

SamplePlot is Vaporware. It does nothing. Mehr...

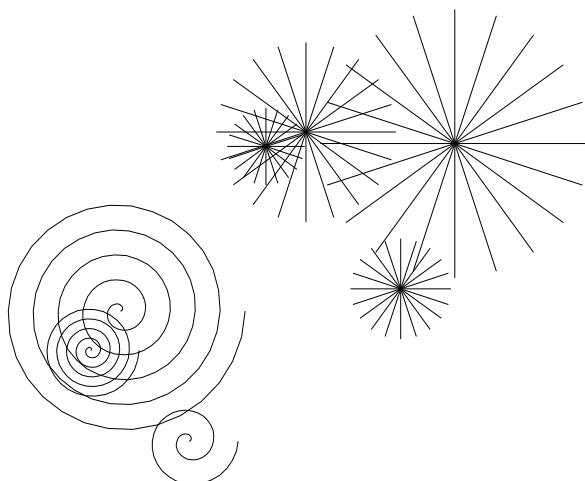
In[182]:=

?ExamplePlot

ExamplePlot[n,m] draws n starbursts and m spirals. ExamplePlot[n] draws n starbursts and no spirals. Mehr...

In[183]:=

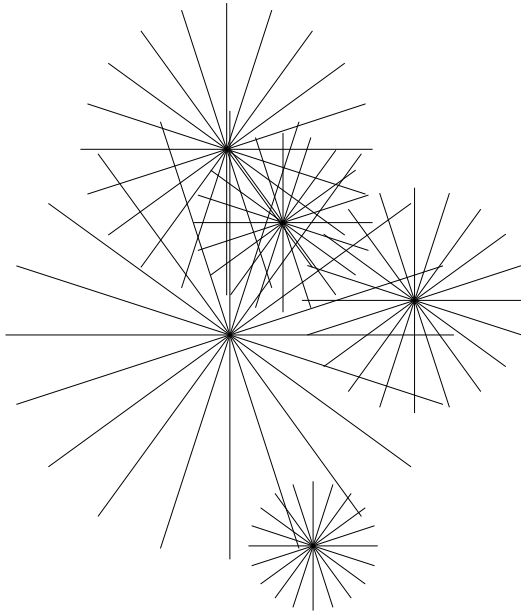
ExamplePlot[4, 3]



Out[183]=

- Graphics -

```
In[184]:=
  ExamplePlot[5]
```



```
Out[184]=
  - Graphics -
```

```
In[185]:=
  SamplePlot[5]
```

```
Out[185]=
  SamplePlot[5]
```

15.6.3. Einlesen eines externen Files in einen andern Context

■ Lire un fichier extérieur et le mettre dans un autre contexte

Im folgenden Beispiel wird gezeigt, wie ProtoNotbook` in den Context PlottingExamples` eingelesen wird. Probiere die verschiedenen Situationen durch:

■ Dans l'exemple suivant on montre comme ProtoNotbook` est mis dans le contexte PlottingExamples`. Essaie les différentes situations:

```
In[186]:=
  ?SymbolicSum

Information::notfound : Symbol SymbolicSum not found. Mehr...
```

```
In[187]:=
  Remove["Global`*"]
```

```
In[188]:=
  End[]
```

```
Out[188]=
  Graphics`Shapes`
```

```

In[189]:=
  ?*SamplePlot

SamplePlot is Vaporware. It      does nothing. Mehr...

In[190]:=
  $Context

Out[190]=
  Global`

In[191]:=
  Begin["PlottingExamples`"];
  $Context

Out[192]=
  PlottingExamples`

In[193]:=
  $ContextPath

Out[193]=
  {AAKurs15`, Calculus`FourierTransform`, Utilities`FilterOptions`,
   Test`, Graphics`Shapes`, Geometry`Rotations`, System`}

In[194]:=
  ?SymbolicSum

Information::notfound : Symbol SymbolicSum not found. Mehr...

In[195]:=
  <<Calculus`VectorAnalysis`

In[196]:=
  ?JacobianMatrix

JacobianMatrix[pt] gives the Jacobian matrix of the transformation from the
default coordinate system to the Cartesian coordinate system at the point pt.
JacobianMatrix[pt, coordsys] gives the Jacobian matrix of the transformation
from the coordinate system coordsys to the Cartesian coordinate system at the
point pt. If pt is not given, the default coordinate variables are used. Mehr...

In[197]:=
  ?*SamplePlot

SamplePlot is Vaporware. It      does nothing. Mehr...

In[198]:=
  Needs["Calculus`VectorAnalysis`"]

```

```
In[199]:=
  ?Calculus`VectorAnalysis`*
```

Calculus`VectorAnalysis`

ArcLengthFactor	Div	Pphi
Biharmonic	DotProduct	ProlateSpheroidal
Bipolar	Eeta	Rr
Bispherical	EllipticCylindrical	ScalarTripleProduct
Cartesian	Grad	ScaleFactors
ConfocalEllipsoidal	JacobianDeterminant	SetCoordinates
ConfocalParaboloidal	JacobianMatrix	Toroidal
Conical	Laplacian	Ttheta
CoordinateRanges	Llambda	Uu
Coordinates	Mmu	Vv
CoordinatesFromCartesian	Nnu	Xx
CoordinatesToCartesian	OblateSpheroidal	Xxi
CoordinateSystem	ParabolicCylindrical	Yy
CrossProduct	Paraboloidal	Zz
Curl	ParameterRanges	
Cylindrical	Parameters	

```
In[200]:=
  $Context
```

```
Out[200]=
  PlottingExamples`
```

```
In[201]:=
  $ContextPath
```

```
Out[201]=
  {Calculus`VectorAnalysis`, AAKurs15`, Calculus`FourierTransform`,
   Utilities`FilterOptions`, Test`, Graphics`Shapes`, Geometry`Rotations`, System`}
```

```
In[202]:=
  a = 1
```

```
Out[202]=
  1
```

```
In[203]:=
  a
```

```
Out[203]=
  1
```

```
In[204]:=
  End[]
```

```
Out[204]=
  PlottingExamples`
```

```
In[205]:=
  a
```

```
Out[205]=
  a
```

```

In[206]:=
    $Context

Out[206]=
    Global`

In[207]:=
    $ContextPath

Out[207]=
    {Calculus`VectorAnalysis`, AAKurs15`, Calculus`FourierTransform`,
     Utilities`FilterOptions`, Test`, Graphics`Shapes`, Geometry`Rotations`, System`}

In[208]:=
    ?Calculus`VectorAnalysis`*

```

Calculus`VectorAnalysis`

ArcLengthFactor	Div	Pphi
Biharmonic	DotProduct	ProlateSpheroidal
Bipolar	Eeta	Rr
Bispherical	EllipticCylindrical	ScalarTripleProduct
Cartesian	Grad	ScaleFactors
ConfocalEllipsoidal	JacobianDeterminant	SetCoordinates
ConfocalParaboloidal	JacobianMatrix	Toroidal
Conical	Laplacian	Ttheta
CoordinateRanges	Llambda	Uu
Coordinates	Mmu	Vv
CoordinatesFromCartesian	Nnu	Xx
CoordinatesToCartesian	OblateSpheroidal	Xxi
CoordinateSystem	ParabolicCylindrical	Yy
CrossProduct	Paraboloidal	Zz
Curl	ParameterRanges	
Cylindrical	Parameters	

15.6.4. Notebooks

■ Notebooks

Dieses File hier ist ein Notebook. In solchen Notebooks oder allgemein bei der interaktiven Benützung von Mathematica mit einem Front-End, das mit Zellen arbeitet, können Probleme mit dem Context auftreten, wenn man nicht aufpasst. Studiere das folgende Beispiel:

■ Le fichier est un Notebook. Dans des Notebooks de ce genre ou plus généralement quand on utilise interactivement *Mathematica* avec un Front-End qui travaille avec des cellules, il peut y avoir des problèmes avec le contexte quand on ne fait pas attention. Etudie l'exemple suivant:

```

In[209]:=
    Remove["Global`@*"]

```

```

In[210]:=
    esel = "Dummes menschliches Individuum,
    Anwesende immer ausgeschlossen. --- Etre humain
    bête, personnes présents exclus."

Out[210]=
    Dummes menschliches Individuum, Anwesende immer
    ausgeschlossen. --- Etre humain bête, personnes présents exclus.

In[211]:=
    Context[esel]

Out[211]=
    Global`

In[212]:=
    BeginPackage["Zoologie"];
    esel = "Säugetier im Dienste des Menschen.
    --- Mammifère au service des hommes";

    BeginPackage::cxt : Invalid context specified at position 1 in BeginPackage[Zoologie].
    A context must consist of valid symbol names separated by and ending with `. Mehr...

In[214]:=
    esel

Out[214]=
    Säugetier im Dienste des Menschen.          --- Mammifère au service des hommes

In[215]:=
    Context[esel]

Out[215]=
    Global`

In[216]:=
    Global`esel

Out[216]=
    Säugetier im Dienste des Menschen.          --- Mammifère au service des hommes

In[217]:=
    Zoologie`esel

Out[217]=
    Zoologie`esel

In[218]:=
    EndPackage[]

    EndPackage::noctx : No previous context defined. Mehr...

Out[218]=
    EndPackage[]

In[219]:=
    esel

Out[219]=
    Säugetier im Dienste des Menschen.          --- Mammifère au service des hommes

```

Wo ist der eingangs eingegebene "Wert" der Variablen "esel" geblieben?

■ Où a disparu la "valeur" de la variable "esel" entrée au début?

"Putzmaschine" einsetzen

■ Employer la "machine de nettoyage"

```
In[220]:=
(* Old Form: Remove["Global`@*"] *)
```

```
In[221]:=
Remove["Global`*"]
```