

## Darstellung von Listen, Vektoren, Matrizen

In Listen stehen Ausdrücke (engl.: *Expression*); Listen werden von `{ }` begrenzt. Es gibt ein Tutorial Making Lists of Objects.

Dies ist eine Liste von Zahlen:

```
{ 2, 3, 5, 7, 11 }
```

```
{2, 3, 5, 7, 11}
```

So entsteht eine Liste von Ausdrücken:

```
Clear[x]; x^% + 1
```

```
{1 + x2, 1 + x3, 1 + x5, 1 + x7, 1 + x11}
```

Listen können "differenziert" werden:

```
D[%, x]
```

```
{2 x, 3 x2, 5 x4, 7 x6, 11 x10}
```

Ersetzen von `x` durch `3` in der letzten Liste:

```
% /. x -> 3
```

```
{6, 27, 405, 5103, 649539}
```

Das kann auch passieren:

```
Clear[x];
```

```
D[{x, Sin[x], x == 0}, x]
```

```
{1, Cos[x], False}
```

Mit `Table` werden Listen erzeugt:

```
Table[xk + 2 k, {k, 3}]
```

```
{2 + x, 4 + x2, 6 + x3}
```

Damit können Vektoren und Matrizen erklärt werden.

So wird z.B. eine 3×3-Matrix  $m$  mit den Elementen  $m_{ij} = i + 2j$  erzeugt:

```
m = Table[i + 2 j, {i, 1, 3}, {j, 1, 4}]
```

```
{{3, 5, 7, 9}, {4, 6, 8, 10}, {5, 7, 9, 11}}
```

```
MatrixForm[m]
```

$$\begin{pmatrix} 3 & 5 & 7 & 9 \\ 4 & 6 & 8 & 10 \\ 5 & 7 & 9 & 11 \end{pmatrix}$$

So kann z.B. ein Vektor  $a$  erzeugt werden:

```
Clear[a]; Array[a, 4]
```

```
{a[1], a[2], a[3], a[4]}
```

Eine Diagonalmatrix:

```
Clear[a, b, c];
DiagonalMatrix[{a, b, c}] // MatrixForm
```

$$\begin{pmatrix} a & 0 & 0 \\ 0 & b & 0 \\ 0 & 0 & c \end{pmatrix}$$

Zu den Operationen aus der linearen Algebra finden Sie einen Einstieg über das Tutorial [Matrices and Linear Algebra](#). Dort gibt es weitere Hinweise zu [Vectors and Matrices](#), [Vector Operations](#), [Constructing Matrices](#), [Solving Linear Systems](#), [Eigenvalues and Eigenvectors](#) etc.

*Mathematica* stellt viele Möglichkeiten zur Verfügung, um mit Listen zu arbeiten. Das dazugehörige Tutorial ist [List Manipulation](#).

$x$  wird vorn in die Liste eingefügt:

```
Prepend[{a, b, c}, x]
{x, a, b, c}
```

$x$  wird hinten in die Liste eingefügt:

```
Append[{a, b, c}, x]
{a, b, c, x}
```

$x$  wird an der zweiten Stelle eingefügt:

```
Insert[{a, b, c}, x, 2]
{a, x, b, c}
```

Join[ <i>liste1</i> , <i>liste2</i> , ...]	hängt Listen aneinander.
Union[ <i>liste1</i> , <i>liste2</i> , ...]	bildet die Vereinigung.
Intersection[ <i>liste1</i> , <i>liste2</i> , ...]	bildet den Durchschnitt.
Complement[ <i>eall</i> , <i>liste1</i> , <i>liste2</i> , ...]	erzeugt die Liste der Elemente, die in <i>eall</i> sind, aber in keiner der <i>listen</i> .

```
Join[{a, b, c}, {x, y}, {a}, {{a}}]
{a, b, c, x, y, a, {a}}
```

```
Union[{c, a, d}, {c, a, s}]
{a, c, d, s}
```

## Namen

Reservierte Wörter und damit die Namen der von *Mathematica* vordefinierten Symbole, Objekte, Funktionen (kurz: Ausdrücke) beginnen immer mit einem großen Buchstaben. Die Namen sind vollausgeschriebene englische Worte (*in special american spelling*), außer es gibt eine gebräuchliche mathematische Abkürzung; also:

```
Abs, Arg, Cos, Cosh, Det,
ArithmeticGeometricMean, DoubleLongLeftRightArrow
```

um nur ein paar Beispiele zu nennen. Es heißt aber auch in *Mathematica* `Eigenvector`, `Eigen-`

value und Eigensystem. Im DocumentationCenter finden Sie jeden vordefinierten Ausdruck meist erklärt und teilweise auch mit vielen Beispielen versehen. Sehen Sie dazu auch den Index of Functions als Alphabetical Listing .

Einige Funktionen, deren Namen mit  $\mathbb{Q}$  enden (IntegerQ) "stellen Fragen" und antworten mit True bzw. False bzw. können die Frage nicht beantworten und geben sie deshalb zurück. Mathematische Funktionen oder Konstanten, die nach Personen benannt sind, haben normalerweise Namen der Art:

EulerGamma, JacobiSymbol, WeierstrassPPrime.

Ausdrücke, wie long\_int, (also mit Unterstrich) sind ungebräuchlich. Der Unterstrich hat eine spezielle Bedeutung, die später noch behandelt wird.

```
long_int = 10 000 000 000 000 000 000 000 000
```

```
10 000 000 000 000 000 000 000 000
```

Namen von selbstdefinierten Ausdrücken sollten immer mit einem kleinen Buchstaben anfangen, nicht mit einer Ziffer, dürfen aber große Buchstaben enthalten. Große und kleine Buchstaben werden also in *Mathematica* unterschieden; hopla und HoPla sind auch zweierlei, kommen im Duden nicht vor, wohl aber KG und kg. Namen von selbstdefinierten Ausdrücken sollten sich auch deutlich unterscheiden von den Namen der von *Mathematica* definierten Ausdrücke.

So kann die Liste der vordefinierten Ausdrücke beginnend mit den Buchstaben Nu abgefragt werden:

? Nu\*

▼ System`

Null	NumberForm
NullRecords	NumberFormat
NullSpace	NumberMarks
NullWords	NumberMultiplier
Number	NumberPadding
NumberFieldClassNumber	NumberPoint
NumberFieldDiscriminant	NumberQ
NumberFieldFundamentalUnits	NumberSeparator
NumberFieldIntegralBasis	NumberSigns
NumberFieldNormRepresentatives	NumberString
NumberFieldRegulator	Numerator
NumberFieldRootsOfUnity	NumericFunction
NumberFieldSignature	NumericQ

Was ist Null?

?? Null

Null is a symbol used to indicate the absence of an expression or a result. It is not displayed in ordinary output. When Null appears as a complete output expression, no output is printed. >>

```
Attributes[Null] = {Protected}
```

Durch Markieren von Null mit der Maus und Drücken der F1-Taste kommen Sie direkt zu der ausführlichen Beschreibung.

Auch das ist möglich:

? \*All

▼ System`

All	ClearAll	FitAll	HoldAll	NHoldAll	ReplaceAll
CheckAll	ExpandAll	ForAll	MapAll	RenderAll	WaitAll

Damit werden wir noch viel zu tun haben:

? :=

*lhs := rhs* assigns *rhs* to be the delayed value of *lhs*. *rhs* is maintained in an unevaluated form. When *lhs* appears, it is replaced by *rhs*, evaluated afresh each time. >>

Weiterhin kann auch bei selbstdefinierten Funktionen mit ? die Definition abgefragt werden. Dazu später noch mehr.

## Mathematische Konstanten

Die wichtigsten Beispiele sind:

Pi  $\pi$                     Kreiszahl  $\pi$  : 3.14159... ,  
 E e                        Eulersche Zahl e : 2.71828... ,  
 I i                         imaginäre Einheit i ,  
 Infinity                  $\infty$  ,  
 Degree                    $\pi/180$  ; Umwandlung von Gradmaß in Bogenmaß,  
 GoldenRatio  $\frac{1+\sqrt{5}}{2}$  ,  
 EulerGamma            Eulersche Konstante  $\gamma$  : 0.577216... .

**N[ Degree ]**

0.0174533

Die relative Maschinengenauigkeit, \$MachineEpsilon , meistens mit *eps* bezeichnet, ist die kleinste Maschinenzahl, für die in Standardgleitpunktarithmetik  $1.0 + eps > 1.0$  ist.

**\$MachineEpsilon**

$2.22045 \times 10^{-16}$

Sie können aber mit kleineren Gleitpunktzahlen rechnen:

$a = 10^{-256}$ ;  $b = a$ ;  $\frac{a+b}{20}$ .

$1. \times 10^{-257}$

## Zahlen

*Mathematica* unterscheidet vier Arten von Zahlen:

123                    Integer ganze Zahl.  
 17/4                  Rational rationale Zahl, dargestellt als Bruch ganzer Zahlen.  
 123.000              Real angenäherte reelle Zahl mit beliebig vorgegebener Genauigkeit.  
 1.1 + 3/4 I Complex   komplexe Zahl  $x + y I$ , wobei  $x$  und  $y$  von der Art Number sind; sie können also Integer, Rational, Real oder auch gemischt sein. Weiterhin gibt es die "Mengen" Integers, Rationals, Reals, Complexes.

Im Zusammenhang mit komplexen Zahlen treten in *Mathematica* u.a. folgende Ausdrücke auf:

<code>I</code>	$i$	$\sqrt{-1}$ .
<code>Re[z]</code>		Realteil von $z$ .
<code>Im[z]</code>		Imaginärteil von $z$ .
<code>Conjugate[z]</code>		das komplex Konjugierte von $z$ .
<code>Abs[z]</code>		Betrag von $z$ .
<code>Arg[z]</code>		Argument von $z$ .

Im Wolfram Demonstrations Project gibt es z.B. dieses Applet zu Complex Number .

`IntegerQ[x]` testet, ob  $x$  "integer" ist; ( Q wie *question*):

**IntegerQ[ 123. ]**

False

**NumberQ[ 123. ]**

True

**LetterQ["z"]**

True

Analog werden die sich selbst erklärenden Ausdrücke `EvenQ`, `OddQ`, `MatrixQ`, `MachineNumberQ`, `MemberQ`, `PrimeQ`, `VectorQ` gebraucht. Eine Liste der mit Q-endenden *Mathematica*-Ausdrücke liefert:

? \*Q

## ▼ System`

AlgebraicIntegerQ	InverseEllipticNomeQ	PossibleZeroQ
AlgebraicUnitQ	IrreduciblePolynomialQ	PrimePowerQ
ArgumentCountQ	LegendreQ	PrimeQ
ArrayQ	LetterQ	QHypergeometricPFQ
AtomQ	LinkConnectedQ	QuadraticIrrationalQ
BinaryImageQ	LinkReadyQ	RootOfUnityQ
CoprimeQ	ListQ	SameQ
DigitQ	LowerCaseQ	SatisfiableQ
DirectoryQ	MachineNumberQ	SquareFreeQ
DistributionDomainQ	MatchLocalNameQ	StringFreeQ
DistributionParameterQ	MatchQ	StringMatchQ
EllipticNomeQ	MatrixQ	StringQ
EvenQ	MemberQ	SymmetricMatrixQ
ExactNumberQ	NameQ	SyntaxQ
FileExistsQ	NumberQ	TautologyQ
FreeQ	NumericQ	TensorQ
HermitianMatrixQ	OddQ	TrueQ
HypergeometricPFQ	OptionQ	UnsameQ
ImageQ	OrderedQ	UpperCaseQ
InexactNumberQ	PartitionsQ	ValueQ
IntegerQ	PolynomialQ	VectorQ
IntervalMemberQ	PositiveDefiniteMatrixQ	

## ▼ PacletManager`

PacletNewerQ

Die Konvertierung zwischen einigen Arten von Zahlen geschieht so:

`N[x, n]` stellt  $x$  als eine  $n$ -stellige dezimale Gleitpunktzahl dar.  
`Rationalize[x]` gibt eine rationale Approximation von  $x$ .  
`Rationalize[x, Δx]` gibt eine rationale Approximation von  $x$  mit Fehlerschranke  $Δx$ .

**N[ 1234 / 9999, 30 ]**

0.1234123412341234123412341234123412

**N[ %, 20 ]**

0.12341234123412341234

**Rationalize[ % ]**

$$\frac{1234}{9999}$$

```
Rationalize[ 1.2 + 3.4 i ]
```

$$\frac{6}{5} + \frac{17 i}{5}$$

```
Rationalize[ N[ $\pi$ ], 10-6 ]
```

$$\frac{355}{113}$$