15. Output and Input

2016-06-13

```
$Version
10.0 for Mac OS X x86 (64-bit) (December 4, 2014)

Clear[a, b, c, x]
```

15.1 Printing Data and Expressions

Print[*expr1,expr2,...*]	print the ***expri*** with no spaces inbetween, but with a newline at the end. The output of **Print[]** is not a string

```
Print[a, b]; Print[c]
ab
c

Do[Print[i, "  ", i^2], {i, 5}]
1  1
2  4
3  9
4  16
5  25

Print[TableForm[{{11, 2}, {3.141, 444}}]]
11       2
3.141    444

Print[MatrixForm[{{11, 2}, {3.141, -444}}]]
( 11      2  )
( 3.141  -444 )

Print[Column[{11, 3.141, 444}]]
11
3.141
444

Print[Column[{11, 3.141, 444}, Left]]
11
3.141
444

Print[Column[{11, 3.141, 444}, Center]]
  11
3.141
 444

Print[Column[{11, 3.141, 444}, Right]]
   11
3.141
  444
```

15.1.0 Tables and Matrices

Column [*list*]	Typeset as a column of elements

Grid [*list*]	Typeset as a grid of elements
TableForm [*list*]	Print in tabular form

```
?? Grid
```

Grid[{{$expr_{11}$, $expr_{12}$, …}, {$expr_{21}$, $expr_{22}$, …}, …}]
is an object that formats with the $expr_{ij}$ arranged in a two-dimensional grid >>

```
Attributes[Grid] = {Protected, ReadProtected}

Options[Grid] = {Alignment → {Center, Baseline}, AllowedDimensions → Automatic,
  AllowScriptLevelChange → True, AutoDelete → False, Background → None,
  BaselinePosition → Automatic, BaseStyle → {}, DefaultBaseStyle → Grid,
  DefaultElement → □, DeleteWithContents → True, Dividers → {},
  Editable → Automatic, Frame → None, FrameStyle → Automatic, ItemSize → Automatic,
  ItemStyle → None, Selectable → Automatic, Spacings → Automatic}
```

Much more information can be found through the link given above.

```
Grid[{{a, b, c}, {x, y, z}}]
```

```
a b c
x y z
```

```
Grid[{{a, b, c}, {x, y^2, z^3}}, Frame → All]
```

a	b	c
x	y^2	z^3

```
Table[(i + 44)^j, {i, 3}, {j, 3}]
```

```
{{45, 2025, 91125}, {46, 2116, 97336}, {47, 2209, 103823}}
```

```
TableForm[%]
```

45	2025	91125
46	2116	97336
47	2209	103823

```
Grid[%]
```

45	2025	91125
46	2116	97336
47	2209	103823

```
Column[Range[1, 15, 3]]
```

```
1
4
7
10
13
```

15.1.1 Formats for outputs

ColumnForm[*list*]	Prints the elements of ***list*** one below the other with left justification
ColumnForm[*list,horiz*]	***horiz*** specifies the justification.
horiz: Left, Center, Right	are the specifications for the justification

```
Print[StringForm["x = ``, y = ``", a^2, b^2]]
```

x = a^2, y = b^2

StringForm["*controlstring*", *expr*$_i$, …]
prints as the text ***controlstring***$_i$, with the printed forms of the ***expr***$_i$ embedded.

```
Print[Row[{"x = ", a^2, ", y = ", b^2}]]
x = a^2, y = b^2

p = Print["Evaluating this won't make p into a string."]
Evaluating this won't make p into a string.

Information["p"]
Global`p

p = Null

ou = {6.7^-4, 6.7^6, 6.7^8}
{0.00049625, 90 458.4, 4.06068 × 10^6}

NumberForm[ou]
{0.00049625, 90458.4, 4.06068 × 10^6}

ScientificForm[ou]
{4.9625 × 10^-4, 9.04584 × 10^4, 4.06068 × 10^6}

ScientificForm[ou, 3]
{4.96 × 10^-4, 9.05 × 10^4, 4.06 × 10^6}

EngineeringForm[ou]
{496.25 × 10^-6, 90.4584 × 10^3, 4.06068 × 10^6}
```

NumberForm[*expr,tot*]	print at most ***tot*** digits of all aproximate real numbers in ***expr***
NumberForm[*expr,{n,d}*]	print at most ***n*** digits with ***d*** decimal digits
ScientificForm[*expr,tot*]	use scientific nottation with at most ***tot*** digits
EngineeringForm[*expr,tot*]	use engineering nottation with at most ***tot*** digits

```
pp = N[π^10, 25]
93 648.04747608302097371669

NumberForm[pp, 12]
93648.0474761

NumberForm[pp, {9, 2}]
93648.05

EngineeringForm[%%, 12]
93.6480474761 × 10^3

ScientificForm[%%%, 5]
9.3648 × 10^4
```

```
NumberForm[N[π^4, 20], DigitBlock -> 5]
NumberForm[N[π^4, 20], DigitBlock -> 5, NumberSeparator -> " "]
```

97.40909 10340 02437 236

97.40909 10340 02437 236

```
NumberForm[17 345 652.29876, DigitBlock -> 5]
```

1.73457×10^{7}

```
l1 = {7.99 × 10^12, 5.3, 9.2 * 10^-6, 2.44 × 10^-15};
```

```
NumberForm[l1, ExponentFunction -> (If[-10 < # < 10, Null, #] &)]
```

$\{7.99 \times 10^{12}, 5.3, 0.0000092, 2.44 \times 10^{-15}\}$

```
NumberForm[l1, NumberFormat → (Row[{#1, "E", #3}] &)]
```

{7.99E12, 5.3E, 9.2E-6, 2.44E-15}

There are additionl options :

```
?? NumberPadding
```

NumberPadding is an option for NumberForm and related functions which gives strings to use as padding on the left- and right-hand sides of numbers >>

Attributes[NumberPadding] = {Protected}

Default format without padding :

```
NumberForm[1 234 567.89, 10]
```

1.23456789×10^{6}

Pad with a tab on the left:

```
NumberForm[1 234 567.89, 10, NumberPadding → {"\t", ""}]
```

1.23456789×10^{6}

```
PaddedForm[12 345 678.9, 10, NumberPadding → {"\t", ""}]
```

1.23456789×10^{7}

```
EngineeringForm[12 345 678.9, 10, NumberPadding → {"\t", ""}]
```

12.3456789×10^{6}

```
ScientificForm[12 345 678.9, 10, NumberPadding → {"\t", ""}]
```

1.23456789×10^{7}

```
? SignPadding
```

SignPadding is an option for NumberForm and related functions that specifies whether padding should be inserted after signs >>

The default pads before signs :

```
NumberForm[{-1.23, 2.46}, 5, NumberPadding → {" ", " "}]
```

{   -1.23,   2.46}

Pad between signs and numbers instead:

```
NumberForm[{-1.23, 2.46}, 5, SignPadding → True, NumberPadding → {" ", " "}]
```

{-   1.23,   2.46}

15.1.1.2 Printing Tables with numbers having decimal points

Grid[] has the Option Alignment -> "." effecting alignment w.r.t. the decimal point.
Below an example provided by Robert Nowak.

```
Table[
   {x, 100 Sin@x} // N // NumberForm[#, {6, 3}, NumberPadding → {"  ", "0"}] & /@ # &,
   {x, -π, π - π / 8, π / 8}] //
  Prepend[#, {Style["x", Bold], Style["100 Sin x", Bold]}] & //
 Grid[#, Frame → All, Alignment → Center] &
```

x	100 Sin x
-3.142	0.000
-2.749	-38.268
-2.356	-70.711
-1.963	-92.388
-1.571	-100.000
-1.178	-92.388
-0.785	-70.711
-0.393	-38.268
0.000	0.000
0.393	38.268
0.785	70.711
1.178	92.388
1.571	100.000
1.963	92.388
2.356	70.711
2.749	38.268

15.1.2 Output for other programming languages

FortranForm[*expr*]	prints ***expr*** in FORTRAN language
CForm[*expr*]	prints ***expr*** in C language
TeXForm[*expr*]	prints ***expr*** in TEX language

```
Clear[a, b, c]

FortranForm[f = a b / c^2]
(a*b)/c**2

l1
{7.99 × 10^12, 5.3, 9.2 × 10^-6, 2.44 × 10^-15}

Column[FortranForm /@ l1]
7.99e12
5.3
9.199999999999998e-6
2.4400000000000002e-15

CForm[f = a b / c^2]
(a*b)/Power(c,2)

TeXForm[f = a b / c^2]
\frac{a b}{c^2}
```

```
ma =
```

$$\left\{\left\{\frac{1}{4}\,(-2\,\alpha_0+\alpha_2),\,-\frac{3\,\alpha_2}{4\,\sqrt{10}}\right\},\,\left\{-\frac{3\,\alpha_2}{4\,\sqrt{10}},\,\frac{1}{10}\,(-5\,\alpha_0-2\,\alpha_2)\right\}\right\};$$

```
MatrixForm[ma]
```

$$\left(\begin{array}{cc} \frac{1}{4}\,(-2\,\alpha_0+\alpha_2) & -\frac{3\,\alpha_2}{4\,\sqrt{10}} \\ -\frac{3\,\alpha_2}{4\,\sqrt{10}} & \frac{1}{10}\,(-5\,\alpha_0-2\,\alpha_2) \end{array}\right)$$

This matrix can be tranferred to a Latex F ile in the following way. Use **TeXForm[Matrix-Form[]]** and copy the output into the LatexFile.

```
TeXForm[MatrixForm[ma]]
```

```
\left(
\begin{array}{cc}
 \frac{1}{4} \left(\alpha _2-2 \alpha _0\right) & -\frac{3
   \alpha _2}{4 \sqrt{10}} \\
 -\frac{3 \alpha _2}{4 \sqrt{10}} & \frac{1}{10} \left(-5
   \alpha _0-2 \alpha _2\right) \\
\end{array}
\right)
```

```
\begin{eqnarray*}
m_a= \left(
\matrix{ \frac{-2\,{{\alpha }_0} + {{\alpha }_2}}{4} &   \frac{-3\,{{\alpha }_2}}{4\,{\sqrt{10}}} \cr
          \frac{-3\,{{\alpha }_2}}{4\,{\sqrt{10}}} & \frac{-5\,{{\alpha }_0} - 2\,{{\alpha }_2}}{10} \cr  }
 \right)
 \end{eqnarray*}
```

15.1.3 Traditional Form

Mathematica outputs polynomials with increasing order of the powers. The ordering of the printout my be changed by the command: **TraditionalForm**

```
Clear[t, x]
f = x^3 + 5 x^2 -  2 x + 4
```

$4 - 2\,x + 5\,x^2 + x^3$

```
f // TraditionalForm
```

$x^3 + 5\,x^2 - 2\,x + 4$

Derivatives may be written in traditional form:

```
Clear[f]
f[x_, t_] = (x^3 + 5 x^2 - 2 x + 4) Sin[ω t]
```

$(4 - 2\,x + 5\,x^2 + x^3)\,\mathrm{Sin}[t\,\omega]$

```
D[f[x, t], x] // TraditionalForm
```

$(3\,x^2 + 10\,x - 2)\,\sin(t\,\omega)$

```
D[f[x, t], t] // TraditionalForm
```

$(x^3 + 5\,x^2 - 2\,x + 4)\,\omega\,\cos(t\,\omega)$

```
Dt[f[x, t], t] // TraditionalForm
```

$$\left(3\,x^2\,\frac{dx}{dt} + 10\,x\,\frac{dx}{dt} - 2\,\frac{dx}{dt}\right)\sin(t\,\omega) + (x^3 + 5\,x^2 - 2\,x + 4)\left(t\,\frac{d\omega}{dt} + \omega\right)\cos(t\,\omega)$$

```
HoldForm[D[f[x, t], t]] // TraditionalForm
```

$$\frac{\partial f(x,t)}{\partial t}$$

15.2 Writing into Files

Export["*file.ext*", *expr*] exports data to a file, converting it to a format corresponding to the file extension *ext*.

Export["*file*", *expr*, "*format*"] exports data to a file, converting it to the specified format.

As formats one may use:

"Lines"	list of strings to be placed on separate lines
"List"	list of numbers or strings, to be placed on separate lines
"Table"	list of lists of numbers or strings, to be placed in *a* two- dimensional array
"Text"	single string of ordinary characters (**.txt**)
"Words "	list of strings to be separated by spaces

```
A = {{a1, b1, c1, d1}, {a2, b2, c2, d2}, {a3, b3, c3, d3}};
B = Transpose[A];
```

```
Export["test1.txt",
        Prepend[B, {"Spalte1", "Spalte2", "Spalte3"}], "Table"];
```

In your main directory you will find the file **test1.txt** with the following content:

Spalte1	Spalte2	Spalte3
a1	a2	a3
b1	b2	b3
c1	c2	c3
d1	d2	d3

```
Export["test2.txt", B, "List"];
```

In your main directory you will find the file **test2.txt** with the following content:
{a1, a2, a3}
{b1, b2, b3}
{c1, c2, c3}
{d1, d2, d3}

```
i1 = y[x]'; i2 = Sin[x];
```

i3 = Hold$\left[\int_1^2 x^2\, dx\right]$;

```
Export["test3.txt", {i1, i2, i3}, "Lines"];
```

In your main directory you will find the file **test3.txt** with the following content:
Derivative[1][y[x]]
Sin[x]
Hold[Integrate[x^2, {x, 1, 2}]]

```
zz = RandomReal[{-99, 99}, {3, 5}]
```

```
{{84.6237, -98.3331, -38.1242, -83.3217, -98.955},
 {10.3328, -22.0336, 86.4178, 19.9974, -34.4644},
 {-55.5559, 29.796, 1.19946, -70.2895, -89.0049}}
```

```
Export["zz.txt", zz, "Text"]
```

```
zz.txt
```

In your main directory you will find the file **zz.txt** with the content **zz** as given above but with more decimal places.

```
Export["zzt.txt", zz, "Table"]
```

```
zzt.txt
```

In your main directory you will find the file **zzt.txt** with the following content **zz** as given above but with more decimal places.

```
zzs = ( -40.26066849685475`   -97.16586911187056`   -95.38992712488499`   78.01086305551189`   26.20956935
        98.24543561601837`   -38.442614834196036`  -69.85730459204922`   26.28483009122988`   0.672697374
       -56.24646417038906`   -67.59242501805886`   -63.07203919468884`   89.04416998904975`   30.4905285
```

```
{{-40.2607, -97.1659, -95.3899, 78.0109, 26.2096},
 {98.2454, -38.4426, -69.8573, 26.2848, 0.672697},
 {-56.2465, -67.5924, -63.072, 89.0442, 30.4905}}
```

```
FullForm[zzs]
```

```
List[List[-40.26066849685475`, -97.16586911187056`,
  -95.38992712488499`, 78.01086305551189`, 26.209569354568146`],
 List[98.24543561601837`, -38.442614834196036`, -69.85730459204922`,
  26.28483009122988`, 0.6726973742467521`],
 List[-56.24646417038906`, -67.59242501805886`,
  -63.07203919468884`, 89.04416998904975`, 30.49052856339341`]]
```

```
Export["zzs.txt", zzs]
```

```
zzs.txt
```

The file zss.txt in the main director contains three lines as:

{-40.26066849685475, -97.16586911187056, -95.38992712488499, 78.01086305551189, 26.209569354568146}
{98.24543561601837, -38.442614834196036, -69.85730459204922, 26.28483009122988, 0.6726973742467521}
{-56.24646417038906, -67.59242501805886, -63.07203919468884, 89.04416998904975, 30.49052856339341}

15.2.1 Exporting pictures

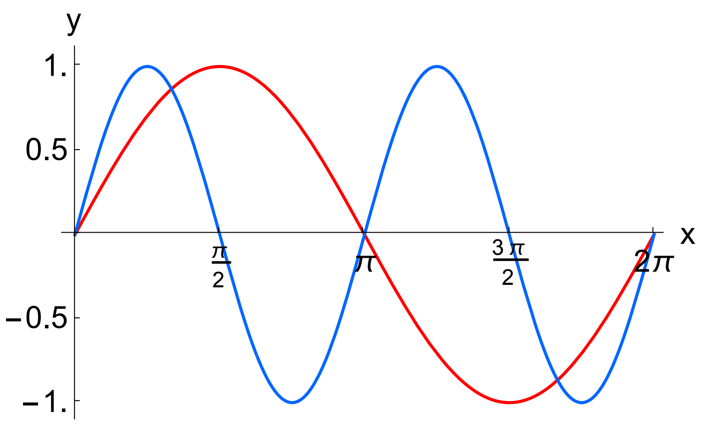
The most convenient way to include pictures or drawings into a LateX document is to use pdf LateX and to store the
pictures as pdf-files. One way is to export the pictures as pdf-files.

```
?? BaseStyle
```

BaseStyle is an option for formatting and related constructs that specifies the base style to use for them >>

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

```
plo = Plot[{Sin[x], Sin[2 x]}, {x, 0, 2 π},
  Ticks → {{{π / 2, "π/2"}, {π, "π"}, {3 π / 2, "3π/2"}, {2 π, "2π"}}, 0.5` Range[-2, 2]},
  AxesLabel → {"x", "y"}, PlotStyle → {Hue[0], Hue[0.6`]},
  BaseStyle → {FontSize → 16}]
```



```
Export["pic.pdf", plo]
```

```
pic.pdf
```

The file pic.pdf is stored in the main directory of the user.

In *Mathematica* on a MacIntosh one may also mark the figure in the notebook, use the command "print selection" and then the selection PDF;
this leads to a menue, where one chooses "Save as PDF"; then one may select the directory, where the file should be stored.

A selection of graphics formats is:
Raster Image Formats: "GIF", "JPEG", "TIFF", "JPG", ...
Vector Graphics Formats: "EPS", "PDF", "WMF", ...
Tabular and Spreadsheet Formats: "Table", "XLS", ...
Document Formats: "PDF", "EPS", "TEX", "Text", ...
Print Formats: "PDF", "EPS", "TEX", ...
Web Formats: "HTML", "XHTML", "GIF", "JPEG", ...
Compression and Archive Formats: "ZIP", "GZIP", "TAR", ...
Binary Formats: "Bit", "Byte", ...

15.3 Reading Files

Import["*name.ext*"] imports data from a file assuming that it is in the format indicated by the file extension `ext`, and converts it to a Mathematica expression.

Import["*file*", "*format*"] imports data in the specified format from a file.

Formats are the same as for **Export[]** .

```
i1 = Import["test1.txt"]
```

```
Spalte1    Spalte2    Spalte3
a1    a2    a3
b1    b2    b3
c1    c2    c3
d1    d2    d3
```

```
i1 = Import["test1.txt", "Table"]
```

```
{{Spalte1, Spalte2, Spalte3}, {a1, a2, a3}, {b1, b2, b3}, {c1, c2, c3}, {d1, d2, d3}}
```

```
FullForm[i1]
```

```
List[List["Spalte1", "Spalte2", "Spalte3"], List["a1", "a2", "a3"],
 List["b1", "b2", "b3"], List["c1", "c2", "c3"], List["d1", "d2", "d3"]]
```

```
Drop[Drop[i1, 1], -1] // MatrixForm
```

$$\begin{pmatrix} a1 & a2 & a3 \\ b1 & b2 & b3 \\ c1 & c2 & c3 \end{pmatrix}$$

```
Det[%]
```

```
-a3 b2 c1 + a2 b3 c1 + a3 b1 c2 - a1 b3 c2 - a2 b1 c3 + a1 b2 c3
```

```
FullForm[%]
```

```
Plus[Times[-1, "a3", "b2", "c1"], Times["a2", "b3", "c1"], Times["a3", "b1", "c2"],
 Times[-1, "a1", "b3", "c2"], Times[-1, "a2", "b1", "c3"], Times["a1", "b2", "c3"]]
```

This shows that the elements of the above list are not expressions; they are strings,
which may be transformed to expressions by the command **ToExpression[ ]** .The commands involving **Map** are explained in Chap.21.

```
FullForm[i2]
```

```
Sin[x]
```

```
i3 = MapAll[ToExpression, Drop[i1, 1]]
```

```
{{a1, a2, a3}, {b1, b2, b3}, {c1, c2, c3}, {d1, d2, d3}}
```

```
FullForm[%]
```

```
List[List[a1, a2, a3], List[b1, b2, b3], List[c1, c2, c3], List[d1, d2, d3]]
```

```
i2 = Import["test1.txt", "Words"]
```

```
{Spalte1, Spalte2, Spalte3, a1, a2, a3, b1, b2, b3, c1, c2, c3, d1, d2, d3}
```

```
i2 = Import["test2.txt", "Lines"]
```

```
{{a1, a2, a3}, {b1, b2, b3}, {c1, c2, c3}, {d1, d2, d3}}
```

```
MatrixForm[i2]
```

```
( {a1, a2, a3} )
( {b1, b2, b3} )
( {c1, c2, c3} )
( {d1, d2, d3} )
```

```
FullForm[i2]
```

```
List["{a1, a2, a3}", "{b1, b2, b3}", "{c1, c2, c3}", "{d1, d2, d3}"]
```

```
i2e = MapAll[ToExpression, i2]
```

```
{{a1, a2, a3}, {b1, b2, b3}, {c1, c2, c3}, {d1, d2, d3}}
```

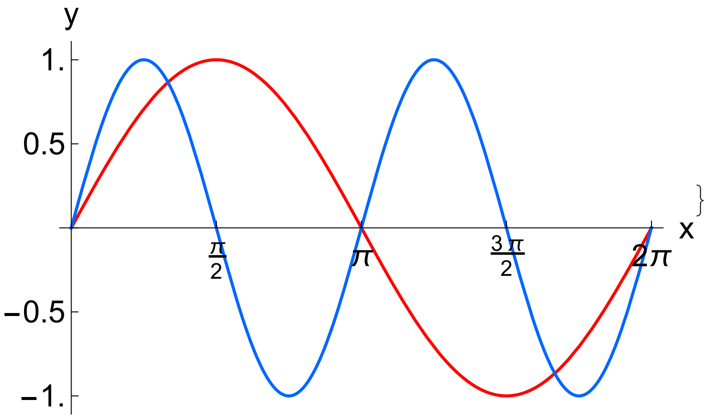
```
FullForm[i2e]
```

```
List[List[a1, a2, a3], List[b1, b2, b3], List[c1, c2, c3], List[d1, d2, d3]]
```

```
MatrixForm[i2e]
```

$$\begin{pmatrix} a1 & a2 & a3 \\ b1 & b2 & b3 \\ c1 & c2 & c3 \\ d1 & d2 & d3 \end{pmatrix}$$

```
Import["pic.pdf"]
```

{  }

15.4 Locating Files

Up to now we did not care in which directory files sare stored or to which directory they are saved.
Below this point is treated in different subscections for
PC's running under UNIX (including Macintosh computers) or
Windows computers.

15.5.1 UNIX Computers

On my home computer where I installed *Mathematica* 9 my user ID is Bernhard.

```
$Path
```

```
{/Applications/Mathematica.app/Contents/SystemFiles/Links,
 /Users/schnizer/Library/Mathematica/Kernel,
 /Users/schnizer/Library/Mathematica/Autoload,
 /Users/schnizer/Library/Mathematica/Applications, /Library/Mathematica/Kernel,
 /Library/Mathematica/Autoload, /Library/Mathematica/Applications, .,
 /Users/schnizer, /Applications/Mathematica.app/Contents/AddOns/Packages,
 /Applications/Mathematica.app/Contents/AddOns/LegacyPackages,
 /Applications/Mathematica.app/Contents/SystemFiles/Autoload,
 /Applications/Mathematica.app/Contents/AddOns/Autoload,
 /Applications/Mathematica.app/Contents/AddOns/Applications,
 /Applications/Mathematica.app/Contents/AddOns/ExtraPackages,
 /Applications/Mathematica.app/Contents/SystemFiles/Kernel/Packages,
 /Applications/Mathematica.app/Contents/Documentation/English/System,
 /Applications/Mathematica.app/Contents/SystemFiles/Data/ICC}
```

This shows the directories and subdirectories where the files are saved on a PC under Unix. This output indicates
how to the define the path for your file in your computer.
One way is to type the complete path as e.g.:

```
Export["/Users/schnizer/test15.txt", {{2.5, 5, 8.}, {3., 4, 7.}}, "Table"]
```

```
/Users/schnizer/test15.txt
```

The subdirectory selected above now contains the file test15.txt.

```
ip = Import["/Users/schnizer/test15.txt"]
```

```
2.5    5    8.
3.    4    7.
```

```
ipp = ReadList["/Users/schnizer/test15.txt", {Number, Number, Number}]
```

```
{{2.5, 5, 8.}, {3., 4, 7.}}
```

```
SetDirectory["/Users/schnizer"]
```

```
/Users/schnizer
```

```
Export["testsch.txt", {{x1, x2, x3}, {y1, y2, y3}}, "Table"]
```

```
testsch.txt
```

The filepath may also be selected by choosing "FilePath" in the menue "Insert".

15.5.2 Windows

The examples below are rather old. So present-day Windows may give a somewhat different picture.
Unfortunately, I have no facility to make tests. When you are using *Mathematica* under windows you should run the commands given below to see how they are working.
I should be grateful on reports on successes and failures.ES AND FAILURES

```
$Path
```

```
{/Applications/Mathematica.app/Contents/SystemFiles/Links,
 /Users/schnizer/Library/Mathematica/Kernel,
 /Users/schnizer/Library/Mathematica/Autoload,
 /Users/schnizer/Library/Mathematica/Applications, /Library/Mathematica/Kernel,
 /Library/Mathematica/Autoload, /Library/Mathematica/Applications, .,
 /Users/schnizer, /Applications/Mathematica.app/Contents/AddOns/Packages,
 /Applications/Mathematica.app/Contents/AddOns/LegacyPackages,
 /Applications/Mathematica.app/Contents/SystemFiles/Autoload,
 /Applications/Mathematica.app/Contents/AddOns/Autoload,
 /Applications/Mathematica.app/Contents/AddOns/Applications,
 /Applications/Mathematica.app/Contents/AddOns/ExtraPackages,
 /Applications/Mathematica.app/Contents/SystemFiles/Kernel/Packages,
 /Applications/Mathematica.app/Contents/Documentation/English/System,
 /Applications/Mathematica.app/Contents/SystemFiles/Data/ICC}
```

The directories listed above are used to store files under Windows. Under Windows the file path must be specified according to rules
differing from those used by the Unix (including Macintosh computers):

"L:\\symb_rech\\schnizer\\iowin.nb"

Note that the backslash must be typed **twice**. Following this rule, the export command looks as follows:

```
Export["schnizer\\test15.txt", {{2.5, 5, 8}, {3, 4, 7}}, "Table"]
```

```
schnizer\test15.txt
```

```
"schnizer\\test15.txt"
```

```
schnizer\test15.txt
```

To avoid the typing of the file path, you can use the point "Get File Path..." in the menu "Input". A directory where files should be saved by default is selected by following command:

```
Export["test2.txt", {{x1, x2, x3}, {y1, y2, y3}}, "Table"]
```

```
test2.txt
```

15.5 Accuracy and Precision

```
?? Accuracy
```

Accuracy[*x*] gives the effective number of digits to the right of the decimal point in the number *x*. >>

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

```
?? Precision
```

Precision[*x*] gives the effective number of digits of precision in the number *x*. >>

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

Accuracy[x] gives the effective number of digits to the right of the decimal point in the number x.
Precision[x] gives the effective number of digits of precision in the number **x**.

Accuracy gives Infinity when applied to exact numbers,such as integers.
Accuracy assumes a precision of $MachinePrecision when applied to machine-precision numbers.

Map$\left[\textbf{Accuracy}, \left\{123, \frac{123}{2}, \sqrt{123}, \pi, e, i\right\}\right]$

$\{\infty, \infty, \infty, \infty, \infty, \infty\}$

The `Accuracy` of a machine real number depends on the value of `$MachinePrecision` on your computer.

```
$MachinePrecision
```

```
15.9546
```

```
Map[Accuracy, {1.618034, 21.618034, 321.618034}]
```

```
{15.7456, 14.6198, 13.4472}
```

```
Map[Precision, {1.618034, 21.618034, 321.618034}]
```

```
{MachinePrecision, MachinePrecision, MachinePrecision}
```

However, the `Accuracy` of arbitrary-precision real numbers is machine-independent.

```
Map[Accuracy, {1.12345678901234567890123456789,
  21.12345678901234567890123456789, 321.12345678901234567890123456789}]
```

```
{29., 29., 29.}
```

```
Map[Precision, {1.12345678901234567890123456789,
  21.12345678901234567890123456789, 321.12345678901234567890123456789}]
```

```
{29.0506, 30.3248, 31.5067}
```

Accuracy can yield zero or a negative result:

```
Map[Accuracy, {23 451 234 567 834 987 678.}]
```

```
{0.}
```

Here is an approximate real number.

```
xp = 234.389879479373987397;
```

This gives the total number of digits entered to specify the real number.

```
Precision[xp]
```

```
20.3699
```

This evaluates π^{25} using numbers with 30 digits of precision.

$N[Pi^{25}, 30]$

$2.68377941431776454900992812440 \times 10^{12}$

The result has 30 digits of precision.

```
Precision[%]
```

```
30.
```

Mathematica treats 3.0 as a machine-precision number.

```
Precision[3.]
```

```
MachinePrecision
```

```
$MachinePrecision
```

```
15.9546
```

Giving anything less than **`$MachinePrecision`** digits yields a machine-precision number.

```
Precision[3.]
```

```
MachinePrecision
```

This evaluates $\Gamma\left[\frac{1}{7}\right]$ using 30-digit precision numbers.

```
N[Gamma[1/7], 30]
```

6.54806294024782443771409334943

In this case, the result has a precision of exactly 30 digits.

```
Precision[%]
```

30.

If you give input only to a few digits of precision, *Mathematica* cannot give you high-precision output.

```
N[Gamma[0.142], 30]
```

6.58965

```
Precision[%]
```

MachinePrecision

If you want *Mathematica* to assume that 0.142 is exact, then you have to show this explicitly.

```
N[Gamma[142/1000], 30]
```

6.58964729492039788328481917496

```
Precision[%]
```

30.

SetPrecision[*expr*, *n*]	create *a* number with ***n*** decimal digits of precision, padding with binary ze

SetAccuracy[*expr*, *n*]	yields a version of ***expr*** in which all numbers have been set to have accuracy **n**

```
$MinPrecision
```

0

```
$MaxPrecision
```

∞

```
lst = RandomReal[{0, 1}, 10]
```

{0.527126, 0.590817, 0.326976, 0.788102,
 0.722395, 0.265939, 0.813988, 0.1802, 0.277587, 0.607367}

```
lplst = SetPrecision[#, 20] & /@ lst
```

{0.52712570924263468797, 0.59081689422856009664,
 0.32697565393086236440, 0.78810243314690753280,
 0.72239528751426651532, 0.26593945514238881422, 0.81398816647705496408,
 0.18020032796169571654, 0.27758672378518767232, 0.60736690714814201009}

This creates a number with a precision of 50 decimal digits. The extra digits come from conversion to base 10.

```
SetPrecision[0.300000000000000, 40]
```

0.2999999999999999888977697537484345957637

```
x = SetPrecision[0.3000000000000000000000000000000000000000, 40]
```

0.3000000000000000000000000000000000000000

```
x^2
```

0.09000000000000000000000000000000000000000

```
Accuracy[%]
```

```
40.7447
```

```
Precision[%%]
```

```
39.699
```

By making the global assignment $MinPrecision = n, you can effectively apply SetPrecision[expr, n] at every step in a computation.This means that even when the number of correct digits in an arbitrary-precision number drops below **n**, the number will always be padded to have n digits. If you set $MaxPrecision = **n** as well as $MinPrecision = **n**, then you can force all arbitrary-precision numbers to have a fixed precision of **n** digits.In effect,what this does is to makeMathematica treat arbitrary-precision numbers in much the same way as it treats machine numbers—but with more digits of precision. Fixed-precision computation can make some calculations more efficient, but without careful analysis you can never be sure how many digits are correct in the results you get.

Here is a small number with 20-digit precision.

```
k = N[Exp[-60], 20]
```

$8.7565107626965203385 \times 10^{-27}$

With *Mathematica*'s usual arithmetic, this works fine.

```
k1 = Evaluate[1 + k] - 1
```

$8.7565107626965203385 \times 10^{-27}$

This tells *Mathematica* to use fixed-precision arithmetic.

```
$MinPrecision = $MaxPrecision = 20
```

```
20
```

```
k2 = Evaluate[1 + k] - 1
```

$8.7565107626965203385 \times 10^{-27}$

```
k1 - k2
```

$1.4620478072211105608 \times 10^{-58}$

```
$MinPrecision = $MaxPrecision = 40
```

```
40
```

```
k = N[Exp[-60], 40]
```

$8.756510762696520338488732800739166036557 \times 10^{-27}$

```
k3 = Evaluate[1 + k] - 1
```

$8.756510762696520338488732800739166036557 \times 10^{-27}$

```
k1 - k3
```

```
0
```

```
$MinPrecision = $MaxPrecision = 50
```

```
50
```

```
xn = SetPrecision[0.30000000000000000000000000000000000000000000000000, 50]
```

```
0.30000000000000000000000000000000000000000000000000
```

```
Precision[%]
```

```
50.
```

```
xn^2
```

0.0900000000000000000000000000000000000000000000000000

```
Accuracy[%]
```

51.0458

```
Precision[%%]
```

50.