Index of FAQs And Pitfalls

This index is organized by FAQ and Beware subject heading. Each entry in this index points to either an FAQ (?) or a Beware (!) entry. The format of each entry is: question and answer page (example page(s)). The plain text entries refer to FAQs while the **emboldened** entries point to a Beware entry. For example, a potential pitfall related to integer data types is highlighted on page 138 and put in context on page 117 - **integer**, 138 (117).

Arguments

:, see **type tag**
::, see **type tag**
data types
    integer, 138 (117)
    invalid, 138 (116)
    %i, 138 (117)
FFT, 194 (173)
procedure definition, argument types in, 112 (89)
type of, see **type tag**
type tag, 112, (89)

Book

! (Beware), 9 (2)
? (FAQ), 9 (2)

Boolean Operations

boolean, cannot evaluate, 110 (81)
for, end condition, 111 (85)
function definition using if, 110 (72)
if, in function definitions, 110 (72)
Customization
Return
    instead of Enter, 69 (39)
    on a Macintosh, 69 (39)
saving interface settings, 71 (63)

Data Input

%i, 138 (117)
arrays, defining n-dimensional, 70 (55)
file closing, 138 (124)
errors
    editing following an Enter, 69 (40)
    echoed input, 138 (113)
    missed end of file, 138 (116)
    invalid data type during read, 138 (116)
format types, 138 (117)
sscanf, 138 (117)

Data Structures

arrays
    converting to, 194 (172)
    defining n-dimensional, 70 (55)
    operating on, 34 (24)
evalhf, using, 69 (47, 50)
evalm, 71 (53, 56, 57), 194 (160)
hash tables, see tables
index, tables, see tables
last name evaluation
    arrays, 70 (55), 71 (57)
    matrix, 35 (24), 71 (53, 56, 57), 194 (160)

    table, 70 (55), 71 (57)
lists
    concatenation, 70 (53)
    converting to array, 194 (172)
    joining, 70 (53)
    operating on, 34 (24), 35 (18), **71 (53, 56)**, **194 (160)**
    portions of, 35 (18), 71 (53)
    removing elements, 35 (18), 71 (53)
matrices, operating on, 34 (24), 71 (53, 56, 57)
multiplication
    list, see lists
    matrix, 35 (24), 71 (53, 56, 57), 194 (160)
    non-commutative, 35 (24), 71 (53, 56, 57), 194 (160)
    scalar, 71 (57)
symbols, operating on, 110 (74)
remember tables, see table
sets, see lists
tables
    displaying, 70 (55)
    index, 111 (87)
    operating on, 34 (24)
    speed of, 111 (87)
vectors
    displaying, 70 (55)
    operating on, 34 (24), **71 (53, 56)**
    row column duality, 71 (57)
    using, 71 (57)

Data Transfer

data retrieval, 194 (172)
EOF (end of file)
    missed, 138 (116)
    readline, 138 (116)
format types, 138 (117)
read, see File I/O
reading ASCII files, see File I/O
save, 138 (121)
worksheet
    transferring from the Macintosh, 69 (41)
write file, see File I/O
writing ASCII files, see File I/O

Efficiency

arrays, converting to, 194 (172)
data retrieval, 194 (172)
evalhf, using, 69 (47, 50)
floating-point hardware, see evalhf
hash tables, 111 (87)
index, tables, 111 (87)
inert functions, 69 (47, 50)
kernelopts, 71 (61)
remember tables, 111 (87)

Evaluation

arrays
    displaying, 70 (55)
    evalm, 71 (53, 56, 57), 194 (160)
    operating on, 34 (24)
back quote (`), see delay quotes
boolean
    cannot evaluate, 110 (81)
code, package, 111 (88)
data retrieval, 194 (172)
delay quotes, 111 (98), 138 (126)
evaluation
    automatic, see substitution
    delayed, 138 (126)
floating-point hardware
    using, 69 (47, 50)
hash tables, 69 (50)
holding evaluation
    of a function 111 (98), 138 (126)
    of rand, 194 (163)
index, tables, see tables
inert functions,
    &*, 35 (24), 71 (53, 56, 57), 194 (160)
    and evalhf, 69 (50)
    why use, 69 (47)
last name evaluation, 70 (55), 71 (57)

- lists
  - operating on, 34 (24), 35 (18), 71 (53, 56), 194 (160)
- matricies, operating on, 34 (24), 71 (53, 56, 57)
- multiplication, see Data Structures
- rand, delayed, 194 (163)
- random delayed, 194 (163)
- **sets, operating on**, see lists
- substitution, automatic, 9 (7)
- symbols, operating on, 110 (74)
- tables, see Data Structures
- vectors
  - evaluation, 70 (55), 71 (57)
  - displaying, 70 (55)
  - operating on, 34 (24), **71 (53, 56)**

File I/O

- **%i**, 138 (117)
- **close file**, 138 (124)
- EOF (end of file), 138 (116)
- **file extensions**
  - **new**, 71 (69)
- file read
  - echoed input, 138 (113)
  - invalid data, 138 (116)
  - missed EOF, 138 (116)
- format type, see **%i**
- read, see file read
- reading ASCII files, see file read
- readline and end of file, 138 (116)
- save, 138 (121)
- write file
  - **closing**, 138 (124)
  - save, 138 (121)
  - **write**, 138, (122)
  - **writebytes**, 138, (122)
  - **writeline**, 138, (122)
  - **writeln**, 138, (122)
- writing ASCII files, see write file

Functions

- **angle bracket notation**, 112 (73)
- function definition
  - alternate syntax, see angle bracket notation
  - if, 110 (72)
  - multiple statements, 110 (72)
- functional programming, see function definition
- holding evaluation of a function 138 (126)
- if in function definitions, 110 (72)
- inert functions, see Evaluation

Hardware

- evalhf
  - using, 69 (47, 50)

Help

- code, printing, 111 (88)
- package listing, 34 (13)
- procedure definition, 111 (88)
- share, 69 (37)

Inert Functions

- &*, 35 (24), 71 (55, 56, 57), 194 (160)

Interface

- command line interface, see interface
- customizing, see Customization
- GUI, see interface
- interface
  - command line, 34 (4)
  - internal format, 34 (12)
  - to kernel, 34 (13)
  - similarities across platforms, 34 (12)
- kernelopts, 71 (61)
- linking to kernel, 34 (13)
- **ms**, 71 (69)

mws, 71 (69)
resource usage, 71 (61)
Return, see Customization
save interface settings, 71 (63)
status, 71 (61)
transferring worksheets from the Macintosh, 69 (41)
worksheet, see interface

Mathematics

arithmetic, exact, 70 (51)
automatic substitution, 9 (7)
bases, number, 70 (52)
E, 9 (9)
e, natural log base, see **E**
errors, computational, 70 (51)
evalhf, using, 69 (47, 50)
function definition, see Functions
lists, see Data Structures
multiplication, see Data Structures
operating on
- array elements 34 (24)
- matrix elements 34 (24), 71 (53, 56, 57)
- list elements, 34 (24), 35 (18), 71 (53, 56), 194 (160)
- symbols, 110 (74)
- table elements, 34 (24)
- vector elements, 34 (24), **71 (53, 56)**

sets, see Data Structures
substitution, automatic, 9 (7)
variables without assumptions, 110 (74)
vectors, 71 (57)

Matrix Algebra

&*, 35 (24), 71 (55, 56, 57), 194 (160)
arrays
- converting to, 194 (172)
- defining n-dimensional, 70 (55)
- displaying, 70 (55)

evalm, 71 (53, 56, 57), 194 (160)

Evaluation

- arrays, see last name evaluation
- matrix, see **&***

last name evaluation, 70 (55), 71 (57)
multiplication, see Data Structures
vector row column duality, 71 (57)

Names

approx, see numapprox
Domains, 35 (13)
E, 9 (9)
e, natural log base, see **E**
file extension, worksheet, 71 (69)
finance, 194 (140)
gauss, see Domains
geom3d, see geometry
geometry, 35 (13)
keywords, 35 (30)
libraries, see packages
long names, 34 (24)
ms, 71 (69)
mws, 71 (69)
np, see NPspinor
NPspinor, 35 (14)
numapprox, 35 (14)
optional arguments, 35 (30)
packages
- approx, see numapprox
- Domains, see Domains
- finance, see **finance**
- gauss, see Domains
- geom3d, see geometry
- geometry, see geometry
- **new names**, 35 (13-14)
- np, see NPspinor
- NPspinor, see NPspinor
- numapprox, see numapprox
- projgeom, see geometry
- **sub-packages**, 194 (145)

projgeom, see geometry
Operating System

- kernel, linking to, 34, (13)
- multi-tasking, 34 (13)
- **resource usage**, 71 (61)
- Return, see Customization
- starting Maple, see worksheet
- status, 71 (61)
- worksheet
  - launching a Maple worksheet, problems, 69 (39)
  - similarities across platforms, 34 (12)
  - transferring from the Macintosh, 69 (41)

Operators

- **&***, 35 (24), 71 (55, 56, 57), 194 (160)
- inert functions, 35 (24), 71 (55, 56, 57), 194 (160)

Packages

- approx, see numapprox
- code, Maple, see packages, code
- Domains, 35 (13)
- **finance**, 194 (140)
- gauss, see Domains
- geom3d, see geometry
- geometry, 35 (13)
- libraries, see packages
- long names, 34 (12)
- np, see NPspinor
- NPspinor, 35 (14)
- numapprox, 35 (14)
- packages, see Names
- projgeom, see geometry
- random sub-package, 194 (164)
- share library, 69 (37)
- **sub-packages**, 194 (145)
- third party libraries, 69 (37)

Printing

- displaying
  - arrays, 70 (55)
  - tables, 70 (55)
  - vectors, 70 (55)
  - matrix, see **evalm**
- **evalm**, 71 (53, 56, 57), 194 (160)
- last name evaluation, 70 (55), **71 (57)**

Procedures

- :, see **type tag**
- ::, see **type tag**
- arguments, type of, see **type tag**
- back quote (`), see delay quotes
- boolean, cannot evaluate, 110 (81)
- code, viewing, 111 (88)
- data types, see Arguments
- delay quotes, 138 (126)
- long names, 34 (12)
- rand, delayed, 194 (163)
- random, delayed, 194 (163)
- variables, local, 111 (83)

Programming

- **angle bracket notation**, 112 (73)
- code, viewing, 111 (88)
- for loop end condition, 111 (85)
- function definition, see Functions
- holding evaluation, see Evaluation
- if in function definitions, 110 (72)
- names
  - keywords, 35 (30)
  - long, 34 (24)
  - **optional arguments**, 35 (30)
- package
  - code, 111 (88)
  - functionality, 34 (13)
- procedure definition
  - argument type, 112 (89)
  - viewing body, 111 (88)
  - boolean, cannot evaluate, 110 (81)
- programming styles, 34 (11)
- rule based programming, see rule definition
- rule definition

  rule order, 111 (98)
  rule re-definition errors 112 (98)
  rules calling rules, 111, (98)
share library, 69 (37)
third party libraries, 69 (37)
variables, local, 111 (83)

Rules

delay quotes, 111 (98)
delayed evaluation, 111 (98)
rule based programming, see rule definition
rule definition
  rule order, 111 (98)
  rule re-definition errors 111 (98)
  rules calling rules, 111, (98)

Types

%i, 138 (117)
:, see **type tag**
::, see **type tag**
arguments, type of, see **type tag**
data types, see Arguments
procedure definition
  argument types in, 112 (89)
type tag, 112, (89)
variables, type of, 112 (89)