Mathematica Dos and Don'ts: Syntax

Michael A. Morrison

Version 1.7: February 2, 2000

DO remember the fundamental syntactical rules for built-in functions:

- The names of all *Mathematica* built-in commands begin with a captial letter (example: `Plot`, `Integrate`).
- If the name of a built-in command consists of more than one word, then all words begin with a captial letter and there are no spaces between words (example: `ListPlot`).
- All arguments to commands must be placed within *square brackets* (example: `Integrate[ Sin[x] , x]`) .

DON'T use parentheses to enclose the argument(s) of a function: `f[x]`, *not* `f(x)`.

DON'T use square brackets (`[]`) or curly braces (`{}`) to group terms in an algebraic expression; use parentheses (`()`) instead. Here are the rules:

chore	syntax
to group terms in an algebraic expression	parentheses `()`
to enclose the arguments to a function	square brackets `[]`
to enclose the elements of a list	curly braces `{}`
to refer to a particular element of a list	double square brackets `[[]]`

DO use an asterisk (*) (not a space) to denote multiplication.

DO remember the syntax for *Mathematica iterators*: `{i, imin, imax, istep}`.

DON'T use an equal sign (`=`) to set an option of a *Mathematica* command; use a right arrow `->` (a minus sign followed by a greater-than sign) instead (example:`PlotRange->All`).

DO enclose all *strings* (e.g., axis labels, plot labels) in double quotes.

DON'T assign expressions or values to symbol names that begin with a capital letter.

DON'T use as symbol names any of the independent variables you're likely to use in function definitions, such as x, y, z, t, θ, ρ, φ, etc. Don't use the symbol name `E` for the energy.

DON'T forget to append an underscore `_` *immediately* after each argument in the argument list in square brackets on the left-hand-side of a function definition.

DON'T use an underscore anywhere on the *right-hand-side* of a function definition.

DO use **postfix notation** (piping) to clarify the structure of *Mathematica* expressions that involve multiple commands; for example, `Expand[(x+3)^2]//Simplify`, not `Simplify[Expand[(x+3)^2]]`.