

Basic stuff -- assignments, arithmetic and functions

Most of the time, you will be using Maple as a kind of super-calculator. It is possible to write programs in Maple -- we will do this very occasionally, but usually you will be typing in Maple statements expecting a (more-or-less) immediate response. The most basic thing to remember about Maple statements is that they all must end either with a semi-colon (; - the usual way to end them) or a colon (: - you will use this occasionally when you don't want to see Maple's output).

The most simple Maple statements involve arithmetic on constants:

```
> 3+4;
```

7

You type in your statement at the > prompt, press "enter" and then Maple responds with the answer.

Other kinds of statements are **assignment** statements, Maple commands (for plotting, solving, etc.), **library management** statements (the "with" command for loading special programs into the computer memory), and a few other assorted special statements.

ASSIGNMENT STATEMENTS

These are the most basic and widely used Maple statements. They allow you to give names to expressions you want the computer to remember for later use. Assignment statements **always** have the "colon-equals" format:

```
name := expression
```

The thing on the left side of the `:=` in an assignment must be a **name** -- i.e., a sequence of letters and digits that begins with a letter. Certain names must be avoided (because Maple is already using them) -- a partial list of these is at the end of this section.

Unlike algebra, where variables usually stand for numbers and their names are only one letter long, in Maple names can stand for many different kinds of things (for example, it is possible to give a name to an entire equation -- see the section on the `solve` command for more information) and can be quite descriptive. For example, in a basic derivative problem concerning position, velocity and acceleration, it is permissible to use the names "position", "velocity", "acceleration" and "time". A typical statement in such a context might be

```
> velocity:=diff(position,time);
```

which means that velocity is the derivative of position with respect to time (`diff` is the name of one of Maple's commands for taking derivatives).

ASSIGNMENTS and BASIC ARITHMETIC and FUNCTIONS:

Let's start with a few simple assignments:

```
> a:=3; b:=5; c:=32/21;
```

$a := 3$

$b := 5$

$c := \frac{32}{21}$

Notice that you can put more than one Maple statement on a line of input. This kind of simple assignment statement is a case where you might want to use a colon instead of a semicolon -- using a semicolon tells Maple to "parrot" your assignment back to you, but a colon would suppress the output:

```
> a:=3: b:=5: c:=32/21:
```

Now, you can use a,b, and c as though they were the numbers they stand for:

```
> a+b;
```

8

```
> c*a;
```

$\frac{32}{7}$

Very important: You must always use the asterisk (*) for multiplication -- it is easy to forget it in expressions like $\sin(2*x)$ or $4*x^3$ -- but forgetting the asterisk usually results in a syntax error. It can also result in unexpected (wrong) output -- as when you type $\sin(ax)$ instead of $\sin(a*x)$.

Subtraction uses the minus sign (-) and division the slash (/). To raise a number to a power use the caret (^ - over the 6). Maple follows the usual "precedence" rules of arithmetic -- unless parentheses indicate otherwise, Maple exponentiates first, then multiplies and divides from left to right, then adds and subtracts from left to right. Can you predict the output of the following?

```
> 3+4^2-5*2;
```

9

Seeing what variables stand for . A useful kind of statement is one where you simply type a variable name and a semicolon -- Maple will tell you what the variable stands for. If Maple just returns the variable name, then (usually) the variable has nothing assigned to it.

```
> a;
```

3

```
> x;
```

Subsequent assignments affect previous ones : If y is specified as depending on x , and then x is assigned a value, then this value of x gets substituted into y . For example:

```
> y:=sqrt(x+3);
```

$$y := \sqrt{x+3}$$

```
> x:=sin(w);
```

$$x := \sin(w)$$

```
> y;
```

$$\sqrt{\sin(w)+3}$$

And if x gets changed, y will also change (because Maple remembers the original definition):

```
> x:=cos(t);
```

$$x := \cos(t)$$

```
> y;
```

$$\sqrt{\cos(t)+3}$$

If you get "tangled up" in such assignments, it is possible to "un-assign" a variable. To do this for x , type:

```
> x:='x';
```

$$x := x$$

Then x is back to not having any value, and so y is given once again in terms of x :

```
> y;
```

$$\sqrt{x+3}$$

There are some instances when you want to continue working in the same file, but get rid of all definitions of variables, loaded programs, etc.. To do this, you can type:

```
> restart;  
> y;
```

y

Notice that the value of y has now been erased.

Parentheses. Maple uses grouping symbols for specific purposes. There are parentheses (), brackets [] and braces { }. In general:

Parentheses () are used for grouping algebraic expressions together (as in $2*(x+4)^3$ for example), and for delimiting the arguments of functions (as in $\sin(x)$, etc..). Neither brackets nor braces may be used for this purpose (their use will result in a syntax error or some other unexpected result).

Brackets [] are used for delimiting "ordered lists" -- such as the ordered pair of numbers used to specify a point in the xy-plane. Brackets are also used for defining vectors and matrices, and for "selecting" things from lists (see the section on solve for an example of this).

Braces { } are used for delimiting sets, just as in ordinary math. Their most common use is for grouping expressions together when using Maple commands like plot or solve.

Basic functions that Maple knows : Maple has all the standard mathematical functions in its library -- it knows about trig functions, exponentials, logarithms, basic probability distributions, and many other familiar (and obscure) functions. The function names Maple knows are:

Trig and inverse trig functions: sin, cos, tan, cot, sec, csc,
arcsin , arccos , arctan , arccot , arcsec , arccsc

Hyperbolic functions: sinh, cosh, , arcsinh, etc...

Exponential and logarithm : Use $\exp(x)$ for "e to the x". Either $\log(x)$ or

$\ln(x)$ stands for the natural logarithm -- to get logs to other bases, use e.g., $\log[10](x)$ for log base 10 of x (this is another use of square brackets).

Square roots, etc: $\text{sqrt}(x)$ stands for the square root of x , $\text{abs}(x)$ for the absolute value of x .

You can see a longer list of functions Maple knows about using the Help browser (Contents under the Help menu) via the path Mathematics -> Basic Mathematics -> Initially known functions.

You can define your own functions : Say for example you found yourself using the expression $\text{sqrt}(1+x^2)$ all the time, plugging values into it, etc.. You might begin to wish that Maple had a built-in function that was $\text{sqrt}(1+x^2)$. It doesn't, but you can define one yourself, as follows:

```
> f:=x->sqrt(1+x^2);
```

$$f := x \rightarrow \sqrt{1+x^2}$$

The arrow in the input is made with the minus and the greater-than signs. This function definition is a special kind of assignment statement. After doing this, you can use the name f in the same way you use exp or sin or log :

```
> f(3*x+1);
```

$$\sqrt{2+9x^2+6x}$$

..and so forth. It is important to realize that you should NEVER, NEVER, NEVER try to define a function with an assignment statement like:

```
> f(x) := sqrt(x+2);
```

$$f(x) := \sqrt{x+2}$$

Maple will let you do it, but the resulting $f(x)$ will be USELESS -- you can't substitute into it or do anything with it that you normally do with functions. NEVER make assignments like this.

Important things to remember about functions and parentheses: It is essential to remember to use parentheses to delimit the argument of a function -- even though in handwritten math we sometimes cheat and write $\sin x$ instead of $\sin(x)$, this is not permitted in Maple. Nested parentheses are permitted -- as in

```
> sin(sqrt(x+2)+ln(abs(x)));
```

$$\sin(\sqrt{x+2} + \ln(|x|))$$

It is important that the parentheses be matched, i.e., that there are just as many left parentheses as right ones.

Special constants that Maple knows : Maple knows about several standard mathematical constants -- to see which ones, you can type

```
> constants;
```

false, γ , ∞ , true, Catalan, FAIL, π

The names of these constants in Maple input are

```
> false, gamma, infinity, true, Catalan, FAIL, Pi;
```

false, γ , ∞ , true, Catalan, FAIL, π

Maple is case-sensitive -- that is, it is important to put capital and lower-case letters where they belong. For example, Maple knows that

```
> sin(Pi);
```

0

but

```
> sin(pi);
```

$\sin(\pi)$

Names to avoid: There are many commands and special objects in Maple that have

pre-assigned names. There are some that are used so frequently (and, without your knowledge because they occur as part of the internal workings of Maple) that you must never assign any other value to their names. The following are names you should never use:

```
and by D do done elif else end
equation fi for from if in intersect
list local matrix minus mod not od
option options or proc quit read save
stop then to union vector while
```

Nor can the following names be used:

```
cos E exp I ln log
Pi sin string tan
```

and there are others -- if you think you may be treading on dangerous ground with your names (or if Maple begins to behave mysteriously and you think it may be due to some name conflict), you can type

```
> ?name
```

where name is the name you would like to check -- if you get a Maple help screen for the name, it is probably a good idea to avoid using it.

```
> ?beeblebrox
```

This results in the message:

which shows that the name "beeblebrox" is probably safe.