

# Basic Maple Commands

| Command                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 1. General Commands and Conventions                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
| $Pi$ , $\exp(1)$ , $I$<br>$f(a)$<br>$;$<br>$:$<br>$\%$<br>cursor on <i>name</i> , click on <i>help</i><br>$settime := time(); expression;$<br>$time() - settime;$<br>$a := expression;$<br>$a + b$ , $a - b$ , $a * b$ , $a/b$<br><br>$a^n$ ;<br>$\sqrt{a}$ ;<br>$evalf(expression, n);$<br>$evalb(a = b);$<br>$a[n];$<br>$plot(expression, x = a..b);$<br>$plot3d(expr, x = a..b, y = c..d);$<br><br>$f := x \rightarrow expr$<br>$f := (x, y, \dots) \rightarrow expr$<br>$a := proc(x, y) local z, w; ...; end;$ | the constants $\pi$ , $e$ and $i$<br>evaluating a function $f$ at $a$ ; e.g. $\sin(Pi)$<br>command end/result displayed<br>" " /result not displayed<br>output of previous line<br>help for <i>name</i><br>to get elapsed time for computing an expression<br><br>assignment<br>Addition, subtraction, multiplication, division of $a$ by $b$<br><br>$n$ -th power of $a$<br>the (exact) square root of $a$<br>numerical value of <i>expression</i> to $n$ -digit accuracy<br>logical comparison (gives <i>true</i> or <i>false</i> )<br>$n$ -th element of list $a$<br>2-dim plot of <i>expression</i> for $x$ between $a$ and $b$<br>3-dim plot of <i>expr</i> for $x$ between $a$ and $b$ and $y$ between $c$ and $d$<br><br>definition of a one-variable function $f(x)$<br>definition of multi-variable function $f(x, y, \dots)$<br>definition of subroutine $a$ |
| 2. Elementary Number Theory                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
| $iquo(a, b)$ ; or $floor(a/b)$ ;<br>$irem(a, b)$ ; or $modp(a, b)$ ;<br>$frac(x)$ ;<br>$igcd(a, b)$ ;<br>$igcdex(a, b, 'x', 'y');$<br>$x, y$ ;<br>$ithprime(n)$ ;<br>$isprime(n)$ ;<br>$ifactor(n)$ ;<br>$a \&^n \bmod m$ ; or $Power(a, n) \bmod m$ ;                                                                                                                                                                                                                                                              | integral part of the quotient $a/b$<br>remainder of division of $a$ by $b$<br>the fractional part of $x$<br>the gcd of $a$ and $b$<br>the extended gcd<br>to extract the values of the above extended gcd<br>the $n$ -th prime number<br>test whether or not $n$ is prime (gives <i>true</i> or <i>false</i> )<br>factor $n$ into its prime factors<br>compute $a^n \bmod m$ efficiently                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |

| Command                                                                                                                                                                                                                                                                                                                                                                                                  | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 3. Sets and Lists: Basic Structure                                                                                                                                                                                                                                                                                                                                                                       |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
| $s := \{1, 2, 3, 4, 5\};$<br>$a := [1, 2, 3, 4, 5];$<br>$s := \{seq(f, i = 1..5)\};$<br>$a := [seq(f, i = 1..5)];$<br>$nops(a);$<br>$a[i]$<br>$a[i..j]$ or $[op(i..j, a)]$<br>$a[select(k \rightarrow k < m \text{ or } k > n, 1..nops(a))];$<br>$member(e, a);$<br>$member(e, a, 'p');$<br>$type(s, set);$<br>$type(a, list);$                                                                          | <p>defines a set <math>s</math>: an unordered sequence of elements</p> <p>defines a list <math>a</math>: an ordered sequence of elements</p> <p>create the set <math>s</math> consisting of the elements <math>f(1), \dots, f(5)</math>; here <math>f</math> is an expression (depending on <math>i</math>)</p> <p>create the list <math>a</math> consisting of the elements <math>f(1), \dots, f(5)</math>; here <math>f</math> is an expression (depending on <math>i</math>)</p> <p>the number of elements in list <math>a</math></p> <p>the <math>i</math>th element of the list <math>a</math></p> <p>the list consisting of elements <math>i</math> through <math>j</math> (inclusive)</p> <p>list <math>a</math> with elements <math>m</math> through <math>n</math> dropped</p> <p>test whether <math>e</math> occurs in list <math>a</math> (<i>true</i> or <i>false</i>)</p> <p>the position(s) at which <math>e</math> occurs in <math>a</math></p> <p>check whether <math>s</math> is a set (has type “set”); gives <i>true</i> or <i>false</i></p> <p>check whether <math>s</math> is a list (has type “list”); gives <i>true</i> or <i>false</i></p> |
| 4. Operations on Sets and Lists                                                                                                                                                                                                                                                                                                                                                                          |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
| $s := convert(a, set);$<br>$a := convert(s, list);$<br>$s \text{ union } t;$ or $\text{'union'}(s, t, \dots)$<br>$s \text{ intersect } t;$<br>$s \text{ minus } t$<br>$[op(a), op(b), \dots]$<br>$a := [e, op(a)];$<br>$a := [op(a), e];$<br>$a := subsop(i = e, a);$<br>$a := subsop(i = NULL, a);$<br>$[op(a[1..n - 1]), e, op(a[n..nops(a)])];$<br>$sort(a);$<br>$[select(bool, a)];$<br>$map(f, a);$ | <p>convert a list to a set</p> <p>convert a set to a list</p> <p>combine sets <math>s, t, \dots</math>, removing repeated elements</p> <p>intersection of sets <math>s</math> and <math>t</math></p> <p>the set of elements which are in <math>s</math> but not in <math>t</math></p> <p>concatenate (join) the lists <math>a, b, \dots</math></p> <p>add element <math>e</math> at the beginning of list <math>a</math></p> <p>add element <math>e</math> at the end of list <math>a</math></p> <p>replace the <math>i</math>th element of the list <math>a</math> by <math>e</math></p> <p>delete <math>i</math>th element from list <math>a</math></p> <p>insert <math>e</math> at position <math>n</math> in list <math>a</math></p> <p>sort the elements of list <math>a</math> (into a standard order)</p> <p>list consisting of the elements of <math>a</math> for which the boolean-valued function <math>bool</math> is true</p> <p>apply the function <math>f</math> to each element of the list <math>a</math></p>                                                                                                                                      |

| Command                                                                                                                                                                                                                                   | Description                                                                                                                                                                                                                                                                                                                                                                                                                         |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 5. Character Strings                                                                                                                                                                                                                      |                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| $str := \text{"This is a string"};$<br>$length(str);$<br>$substring(str, m..n);$<br>$[seq(substring(str, k..k), k = 1..length(str))]$<br>$searchtext(st, str)$<br>$cat(s1, s2, \dots)$<br>$convert(expr, string);$<br>$type(str, string)$ | defining a character string<br>the number of characters in a string<br>extract a substring from string $str$ starting with the $m$ th and ending with the $n$ th character<br>give the list of characters in a string<br>find the place where $st$ occurs in string $str$<br>join the strings $s1, s2, \dots$ together<br>convert an expression to a string (textual form)<br>check whether $str$ is a string ( $true$ or $false$ ) |
| 6. Boolean expressions                                                                                                                                                                                                                    |                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| $b := true; b := false;$<br>$=, <>, <, <=, >, >=$<br>$and, or, not$<br>$evalb(bool)$<br>$type(b, boolean)$                                                                                                                                | assigning true/false to the variable $b$<br>relation operators (equal, not equal, less than, etc.); can be used to form boolean expressions<br>logical operators ( $\rightarrow$ boolean expressions)<br>evaluate the boolean expression $bool$ (gives $true$ or $false$ )<br>check whether $b$ is a boolean expression ( $true$ or $false$ )                                                                                       |
| 7. Looping control                                                                                                                                                                                                                        |                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| $for\ i\ to\ m\ do;\ expr;\ od;$<br>$for\ i\ from\ n\ to\ m\ by\ s\ do;\ expr;\ od;$<br>$while\ test\ do;\ expr;\ od;$<br>$for\ i\ from\ n\ to\ m\ by\ s\ while\ test\ do;\ expr;\ od;$<br>$RETURN(expr)$ or $return(expr)$               | evaluate $expr$ repeatedly with $i$ varying from 1 to $m$ in steps of 1<br>evaluate $expr$ repeatedly with $i$ varying from $n$ to $m$ in steps of $s$<br>evaluate $expr$ until $test$ becomes false<br>evaluate $expr$ repeatedly with $i$ varying from $n$ to $m$ in steps of $s$ as long as $test$ is true<br>(explicit) return from a subroutine, assigning the value $expr$ to the subroutine                                  |
| 8. Conditionals                                                                                                                                                                                                                           |                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| $if\ test\ then\ statmt\ fi;$<br>$if\ test\ then\ statmt_1\ else\ statmt_2\ fi;$                                                                                                                                                          | execute the statement (sequence) $statmt$ only if $test$ is true<br>execute the statement (sequence) $statmt_1$ if $test$ is true, otherwise execute $statmt_2$                                                                                                                                                                                                                                                                     |

| Command                               | Description                                                                                                  |
|---------------------------------------|--------------------------------------------------------------------------------------------------------------|
| 9. Complex Numbers                    |                                                                                                              |
| $z := x + y * I;$                     | defining a complex number                                                                                    |
| $abs(expr);$                          | the absolute value of $expr$                                                                                 |
| $argument(expr)$                      | the argument of $expr$                                                                                       |
| $Re(expr); Im(expr);$                 | the real and imaginary part of $expr$                                                                        |
| $conjugate(expr);$                    | the complex conjugate of $expr$                                                                              |
| $evalc(expr)$                         | evaluating an expression (as a complex number)                                                               |
| $convert(expr, polar)$                | convert $expr$ to its polar form                                                                             |
| $type(expr, complex)$                 | check that $expr$ has type “complex”                                                                         |
| 10. Polynomials                       |                                                                                                              |
| $f := x^n + a_1 * x^{(n-1)} + \dots;$ | defining a polynomial $f = f(x)$ (assuming that $x$ has no value)                                            |
| $type(f, polynom(integer, x))$        | check that $f$ is an integer polynomial in $x$                                                               |
| $degree(f, x)$                        | degree of $f$ in $x$                                                                                         |
| $coeff(f, x, n)$                      | extract the coefficient of $x^n$ in $f$                                                                      |
| $coeffs(f, x)$                        | list of coefficients of $f(x)$                                                                               |
| $lcoeff(f, x)$                        | the leading (highest) coefficient of $f(x)$                                                                  |
| $tcoeff(f, x)$                        | the constant (trailing) coefficient of $f(x)$                                                                |
| $collect(f, x)$                       | collect all coefficients of $f$ which have the same powers in $x$                                            |
| $expand(expr)$                        | distribute products over sums                                                                                |
| $sort(f)$                             | sort into decreasing order                                                                                   |
| $subs(x = a, f)$                      | evaluate $f(x)$ at $x = a$                                                                                   |
| $Eval(f, x = a) \bmod p;$             | evaluate $f(x) \pmod{p}$ at $x = a$                                                                          |
| $f \bmod n;$                          | reduce the coefficients of $f$ modulo $n$                                                                    |
| $quo(f, g, x); rem(f, g, x);$         | the quotient and remainder of division of $f$ by $g$ (viewed as polynomials in $x$ )                         |
| $gcd(f, g, x)$                        | the greatest common divisor of $f(x)$ and $g(x)$                                                             |
| $gcd(f, g, x, 's', 't')$              | the extended Euclidean algorithm of $f(x)$ and $g(x)$ ; i.e. $s, t$ satisfy $f * s + g * t = g := gcd(f, g)$ |
| $factor(f)$                           | factor $f$ into its irreducible factors                                                                      |
| $Factor(f) \bmod p$                   | factor $f$ modulo $p$                                                                                        |
| $roots(f)$                            | find the rational roots of $f$                                                                               |
| $interp(x, y, t)$                     | The Lagrange Interpolation polynomial                                                                        |