# Maple gyorsreferencia

Wettl Ferenc Ver. 0.1 2006-09-25

| Általános utasítások | |
|---|---|
| `;` | parancs vége |
| `:` | parancs vége, de elnyomja a kiírást |
| `quit; stop; done;` | kilépés karakteres módban |
| `restart;` | Újraindítás |
| `with(`*csomag*`):` | a *csomag* betöltése |
| `?`*téma* | a *téma* keresése |
| *a*,*b* `[`*a*,*b*`]` `{`*a*,*b*`}` | sorozat, lista, halmaz |
| `%, %%, %%%` | az előző három utasítás eredménye (ditto) |
| `..` | *a*`..`*b* *a*-tól *b*-ig terjedő 1-esével lépő tart. |
| `" "` | karakterlánc megadása |
| `\|\|` | karakterláncok összefűzése |
| ‘ ‘ | név különleges karakterekből |

| Függvények | |
|---|---|
| trigonometrikus | `sin cos tan cot` |
| inverz trig. | `arc...` |
| (négyzet)gyök | `sqrt(`*x*`) surd(`*x*,*n*`)` |
| exponenciális | `exp` |
| logaritmus | `ln log10 log[`*b*`](`*x*`)` |
| abszolút ért. | `abs` |
| faktoriális | `!` |

| Szimbolikus kalkulátor | | |
|---|---|---|
| műveleti jelek | `+ - * / ^` | `(23+32)^2*3;` |
| számláló, nevező | `numer denom` | `denom( 3/5 );` |
| relációjelek | `<, >, =, <=, >=, <>` | `1<=2; 2<>3;` |
| logikai műveletek | `and or xor implies not` | `1<2 or 2<1` |
| konstansok | `Pi I exp(1) infinity NULL` | `sin(Pi); sum(1/n, n=1..infinity);` |
| értékadás | `:=` | `f := x+y;` |
| függvény | `->` | `g := (x,y) -> x+y; h := x->x^2;` |
| kifejezés kiértékelése | `eval, subs` | `eval(f, x=1,y=2); subs(x=1,y=2,f);` |
| lebegőpontos kiértékelés | `evalf(`*kif [,pontosság]*`)` | `evalf( Pi, 100 );` |
| komplex és logikai kiért. | `evalc evalb` | `evalc(1/(a+I)); evalb(evalf(3^Pi>Pi^3));` |
| függvény kiértékelése | | `f(1,2); h(y);` |
| kifejezésből függvény | `unapply(`*függvény*,*változók*`)` | `F := unapply(f,x,y); F(1,2);` |
| függvénykompozíció | `@ @@` | `f:=sin@(x->x^2); g:=(x->x^2)@@3;` |
| késleltetett kiértékelés | `’ ’ evaln` | `x:=1; x:=’x’; x:=evaln(x);` |
| határérték | `limit` | `limit( sin(x)/x, x=0 );` |
| differenciálhányados | `diff, D` | `diff( exp(x)*sin(y), x ); D(sin);` |
| integrál | `int` | `int( x^2, x );` |
| kiértékeletlen | `Limit Diff Int value` | `a := Int( x^2, x=1..3 ); value(a);` |
| egyenletek | `solve fsolve` | `solve( a*x=b, x ); fsolve( sin(x)=1/3, x );` |
| megoldás+értékadás | `assign` | `S := solve( a*x=b, x ); assig(S);` |
| egyenlet jobb/bal oldala | `rhs lhs` | |
| egyszerűsítés | `simplify` | `simplify(exp(a+ln(b*exp(c))));` |
| felbontás | `expand` | `expand(sin(x+y));` |
| normál alak | `normal` | `normal( 1/x+x/(x+1) );` |
| együtthatók összevonása | `collect` | `f := a*ln(x)-ln(x)*x-x; collect(f,ln(x));` |
| átalakítás más fomára | `convert` | |
| operandusonkénti | `map map2` | `map( sin, x+y );` |
| redő | `foldl foldr` | `foldl( F, id, a, b, c, d );` |
| operandusok kezelése | `op nops` | `> u:=x+y; nops(u); op(2,u);` |

| Programozás | |
|---|---|
| feltételes utasítás | `if` *felt* `then` *utasítások* `[elif` *felt* `then` *utasítások]* `[else` *utasítások]* `end if` (régi: `fi`) |
| for ciklus | `[for` *név]* `[from` *kif]* `[by` *kif]* `[to` *kif]* `[while` *felt]* `do` *utasítások* `end do` (régi: `od`) |
| for-in ciklus | `[for` *név]* `[in` *kif]* `[while` *felt]* `do` *utasítások* `end do` (régen: `od`) |
| kiugrás a ciklusból | `break` |
| ugrás a következőre | `next` |
| kifejezéssorozat | *kifejezés*`$`*i*=*m*..*n*`;` *kifejezés*`$`*n*`;` `$`*m*..*n*`;` |
| kifejezéssorozat | `seq(`*kifejezés*, *i*=*m*..*n*`); seq(`*kifejezés*, *i*=*x*`);` |
| eljárás (procedúra) | `proc(`*arg*`)` `[local` *nevek*`;]` `[global` *nevek*`;]` *állítások* `end proc` |
| visszatérés | `return` *kif1*, *kif2*,... *kifn* |