

HP Prime PPL Reference

Programmer's guide · 1102 commands, functions & variables by topic · from the HP Prime User Guide (2018)

Page 1 explains the HP Prime Programming Language (PPL): program structure, objects, variables and scope, operators, tests, branching, loops, functions, errors, input/output, graphics, strings, lists & matrices, apps & views, user keys, CAS access, base integers, idioms; a drawing of the keyboard sits under the navigation bar and the Toolbox menu tree follows the guide. The following pages index every command, function and system variable in the User Guide's Functions & Commands, Programming, Variables, Lists, Matrices, Units and Geometry chapters by category and subgroup. Hover any entry for its syntax, description and example. The last pages cover Python — MicroPython, the hpprime bridge, every module — and the calculator itself: its files and its electronics.

Type: ● Home / PPL (UPPERCASE, Home + programs + CAS) ● CAS (lowercase; from PPL use CAS.name() or CAS("...")) ● App function (qualify: App.name)

● Statement / keyword ● Variable ● Python *italic* also listed elsewhere



The keyboard

Each key carries up to three functions: its own legend, the **blue** one printed on its lower-left after **Shift**, and the **orange** letter or symbol at its lower-right after **ALPHA**. The black block at the top holds the app keys — Symb, Plot, Num are the three views of the current app, **Shift** gives each its Setup — with the rocker wheel for the cursor and Help, Esc, CAS, Home, View, Menu around it. The bottom line of the screen is a row of six **touch buttons** that change with the view.

	then key	blue function (ASIN, List, Matrix, Program, Units, Chars...); twice locks it
ALPHA key	· ALPHA ALPHA	one orange letter · alpha lock (ALPHA again to leave)
ALPHA	key	lowercase letter
Toolbox	· Templ · Vars	the tool-box icon opens the Toolbox menus · the template icon the math templates · Vars the variables menu
Esc	· On	cancel / clear the line (Esc clears) · On is Off; On also interrupts a running program
Enter	· Enter ≈	evaluate · force a numeric result from the CAS
Help		long context help for whatever is selected; Help toggles User-key mode

Key codes for GETKEY run 0 (Apps) to 50 (+), left to right and top to bottom — see User Keys & Key Codes.

Program Structure

```
EXPORT NAME(a, b)           // header: name + parameters
BEGIN
    LOCAL x, y:=0;           locals (optionally initialized)
    x := a*b;                 every statement ends with ;
    RETURN x+y;               value of the program call
END;                          // END; closes the block
```

ELEMENT	NOTES
EXPORT f(...)	Global: callable from Home, other programs, the User menu (Toolbox ▶ User). Without EXPORT the function is private to the file.
f(...); above header	Forward declaration — needed when a function is called before it is defined in the file.
EXPORT V; / EXPORT V:=1;	Global user variable (Vars ▶ User). Put it above the function that assigns it.
// text	Comment to end of line.
#pragma mode(...)	Pins separator / integer settings for the compile (Shift Menu ▶ Insert pragma).
Name rules	Letters, digits, <code>_</code> ; must start with a letter; case-sensitive. Reserved names (A–Z, L1, M1, Xmin...) cannot be program or variable names.

RUNNING

- Home: type NAME(3,4) Enter — a program with parameters prompts for them when run from the catalog.
- Program Catalog ▶ Run looks for a function called START(); with several EXPORT ed functions it lists them.
- Value returned = the RETURN value, else the last statement's value.
- Several EXPORT ed functions in one file appear as a folder in the User menu (delete the auto template first).

EDITOR KEYS

Cmnds menu (Strings, Drawing, Matrix, App, Integer, I/O, More) · TmplIt menu (Block, Branch, Loop, Variable, Function) inserts structures with blanks · Check = syntax check · Shift Menu = Create user key / Insert pragma · Shift 1 = Program catalog.

Objects & Literals

TYPE	LITERAL / EXAMPLE	NOTES
Real	3.14 1E-3 -5	12 significant digits, exponent ±499. TYPE =0. Home vars A–Z, θ.
Integer	#1101b #FFh #17o #42d #FF:24h	Base-marked binary integer; wordsize 1–64 (:bits). TYPE =1.
String	"text" "say" ""hi"" ""a\\nb"	Double the quote to embed it; \\ newline, \\ backslash. Join with +. TYPE =2.
Complex	(3,4) 3+4*i	Z0–Z9. TYPE =3.
Vector / matrix	[1,2,3] [[1,2], [3,4]]	M0–M9. Elements M1(i,j). TYPE =4.
List	{1,"a",{2,3}}	Any mix of types, nestable. L0–L9. TYPE =6.
Expression	'X^2+1' F1(X)	Single quotes prevent evaluation. Store in F0–F9, E0–E9 etc.
Function object	x→x^2 (CAS)	Mapping used by map/apply/select... TYPE =8.
Unit	5_m 3_kmh	Number _ unit. CONVERT(5_m,1_ft). TYPE =9.
Graphic (GROB)	G0 ... G9	Screen is G0. Not assignable with := (use DIMGROB/BLIT).
Color	#FF0000 RGB(255,0,0) #FF:24h	Colors are integers 0xRRGGBB; alpha in high byte for PIXON.
Sexagesimal	54°52'34.68"	Shift a b/c key; →HMS / HMS→.
Interval / range	1..5 x=0..2*π	CAS ranges for seq, solve guesses, delrows...

TYPE(obj) → 0 Real · 1 Integer · 2 String · 3 Complex · 4 Matrix · 5 Error · 6 List · 8 Function · 9 Unit · 14.x CAS object. CAS type(x) returns DOM_... names.

Variables & Scope

```
LOCAL i, s:="", L:={};           local to the function; untyped
A := 5; 5 ▶ A; sto(5, A);         three ways to store
L1(3) := 7; M1(2,1) := 0;        element assignment
Function.Xmin                    qualified app variable
MYPROG.count                     another program's variable
EXPORT SIDES;                    above the header: global user var
EXPORT ratio:=0.15;              global with initial value
```

KIND	WHERE / RULES
Home (system)	Reserved, typed, global: A–Z θ real · Z0–Z9 complex · L0–L9 lists · M0–M9 matrices · G0–G9 graphics · settings (HAngle, HFormat, Base...). Cannot hold another type or be deleted.
App	Reserved per app (Xmin, F1, C1, Root...). Same name in several apps ⇒ qualify: Polar.Xmin. Setting one changes the app.
CAS	Lowercase names, untyped, may stay symbolic. Home sees them too (5*q).
User	Created by assignment in Home (asks to confirm) or EXPORT ed from a program; listed under Vars ▶ User next to the program. Untyped.
Local	LOCAL inside a function; disappears on exit; can hold any type; convention: lowercase names.

- Uppercase X, Y, Z... are Home reals — the same X the Function app plots. Prefer LOCALs to avoid clobbering.
- Names are case-sensitive: MaxTemp ≠ maxTemp.
- Later EXPORT of the same name wins over an earlier program's.
- Reset a user variable: Vars ▶ User ▶ Delete, or DelHVars("name"); app vars via DelAVars.
- Introspection: HVars, AVars, Programs, Notes, AFiles read/write by name or index.

Operators & Precedence

#	HIGHEST → LOWEST (SAME LEVEL: LEFT TO RIGHT)
1	() — innermost first
2	! √ x ⁻¹ x ²
3	n√ (NTHROOT)
4	^ 10^n (ALOG)
5	negation, * / MOD (and implied multiplication 2X)
6	+ -
7	< > ≥ ≤ == ≠ <> =
8	AND NOT
9	OR XOR
10	left argument of (where)
11	:=

```
-5^2 → -25; (-5)^2 → 25           parenthesize negatives
2X ≡ 2*X; AB ≡ A*B                 implied multiplication
6 < 8 < 11 → 1                     chained comparisons allowed
{1,2}+{10,20} → {11,22}; 5*{1,2}  lists/matrices vectorize
M1.*M2 M1./M2 M1.^2                element-wise matrix ops
"ab"+"cd" → "abcd"; "n="+5        + concatenates strings
                                     (numbers coerced)
M1*M2, M1/M2 (=M1*M2-1), M1^~1    matrix products / inverse
7 MOD 3 → 1; %(20,50) → 10         modulo, percent
x^2+1 | x=3                         where: substitute
```

`=` builds an equation (for solve/E1); `==` tests equality; `:=` assigns. In RPN entry mode operators execute immediately (Home only).

Tests & Logic

<code>x == 3; x <> 3; x ≠ 3</code>	1 (true) or 0 (false)
<code>a < b AND NOT c; p OR q; p XOR q</code>	any non-zero is true
<code>0 ≤ A ≤ 1</code>	chained range test
<code>EQ({1,2},{1,2}) → 1; {1,2}=={1,3} → {1,0}</code>	list equality vs element-wise
<code>IFTE(x<0, -x, x)</code>	inline if
<code>TYPE(v)==2; type(v)==DOM_STRING</code>	type tests (Home / CAS)
<code>POS(L1, v) > 0; contains(L1,v)</code>	membership
<code>INSTRING(s,"x") > 0</code>	substring test
<code>ISKEYDOWN(4); GETKEY == 30</code>	key tests
<code>ISCHECK(1); isPrime(n); even(n)</code>	app / CAS predicates

- Tests yield 0/1 reals; `IF` / `WHILE` / `UNTIL` treat any non-zero as true.
- `IF` with a *list* test: both branches must return single objects or equal-length lists (element-wise selection).
- Comparisons of strings are lexical; of lists element-wise.
- Complex results from real inputs (e.g. `ASIN(2)`) need `HComplex=1`, else an error.

Conditionals

```
IF test THEN commands; END;

IF test THEN commands1; ELSE commands2; END;

CASE

  IF t1 THEN c1; END;  first true branch runs, then CASE ends

  IF t2 THEN c2; END;  up to 127 branches

  DEFAULT c3;          optional

END;

IFTE(test, a, b)

IFERR c1; THEN c2; [ELSE c3;] END;
```

```
CASE

  IF A<0 THEN RETURN "negative"; END;

  IF 0≤A≤1 THEN RETURN "small"; END;

  DEFAULT RETURN "large";

END;
```

- No `ELSE IF` keyword — nest `ELSE IF ... END; END;` or use `CASE`.
- Every `END` takes a semicolon.
- `PIECEWISE(c1,e1,...)` and `when(c,a,b)` are the CAS/Symbolic-view equivalents of `IFTE`.

Loops

```
FOR k FROM 1 TO 10 [STEP 2] DO ... END;

FOR k FROM 10 DOWNT0 1 [STEP 2] DO ... END;

WHILE test DO ... END;

REPEAT ... UNTIL test;

BREAK;    BREAK(2);  leave 1 / n loop levels

CONTINUE;      next iteration
```

```
// count-controlled with real steps

FOR X FROM Xmin TO Xmax STEP (Xmax-Xmin)/318 DO

  PIXON(X, SIN(X)); END;

// sentinel loop

REPEAT INPUT(n,"Sides","N=", "≥ 4",4); UNTIL n≥4;

// wait for a key

WHILE GETKEY==-1 DO END;    // or: WAIT(-1) / FREEZE

// iterate a list

FOR j FROM 1 TO SIZE(L1) DO s:=s+L1(j); END;
```

- The loop variable is any variable (LOCAL recommended); the FOR bound is evaluated once; the counter is left at its last value.
- Vectorized alternatives are faster: `MAKELIST`, `ΣLIST`, `EXECON`, `map`, `MAKEMAT`.
- `ITERATE(expr, var, start, n)` applies `expr` `n` times.

Functions & Subroutines

```
ROLLDIE();           // forward-declare private helper

EXPORT ROLLMANY(n, sides)

BEGIN

  LOCAL k, roll, res;

  res := MAKELIST(0,X,1,2*sides,1);

  FOR k FROM 1 TO n DO

    roll := ROLLDIE(sides)+ROLLDIE(sides);

    res(roll) := res(roll)+1;

  END;

  RETURN res;

END;

ROLLDIE(n) BEGIN RETURN 1+RANDINT(n-1); END;
```

- Parameters are positional, untyped, passed by value; no defaults or optional args (test `TYPE` yourself, or pass a list).
- `RETURN expr;` ends the function; `RETURN;` alone returns nothing. Recursion is allowed.
- Functions can return any object — return a list to give back several results (`RETURN {a,b};`).
- Names must be unique across all programs; a helper without `EXPORT` is visible only in its file.
- Calling another program's exported function: just call it (it must be compiled/present). Program stored source is not shared — only exported names.
- `DEFINE` (Shift x t θ n) creates simple one-line user functions without a program.
- Program is compiled on save; a syntax error prevents running.

Errors & Debugging

```
IFERR

  M2 := M1^-1;

THEN

  MSGBOX("Singular matrix, code "+Ans);

ELSE

  MSGBOX("ok");

END;
```

- Inside `THEN` , `Ans` holds the error number.
- Uncaught runtime errors stop the program with a message; syntax errors are caught by **Check** in the editor.
- **Debug** in the Program Catalog (or `DEBUG(prog)`) steps line by line: Step, Skip, Vars (watch), Stop, Continue. `KILL`; stops a debug run.
- CAS: `breakpoint` / `halt` pause CAS programs.
- Timing: `TEVAL(expr)` seconds; `TICKS` ms since boot (`t:=TICKS; ...` ; `(TICKS-t)/1000`).
- Print debugging: `PRINT(x)` writes to the terminal (Esc/any key hides it; `PRINT()` clears).
- Common runtime errors: *Bad argument type/count, Invalid dimension, Undefined name, Insufficient memory, Infinite result*. Assigning the wrong type to a typed Home var (list into A) errors.
- `MEMORY()` free RAM/flash; `VERSION` firmware.

Input

```
INPUT(r, "Circle", "r=", "Enter radius", 1, 5)

// var, title, label, help, reset, initial → 1 OK / 0 Cancel
INPUT({A,B}, "Two values", {"A:", "B:"}, {"help A", "help B"},
{0,0}, {1,2});

INPUT({ {n, [0], {20,40,1}}, {flag, 0, {65,15,1}},
        {c, {"Red", "Blue"}, {20,40,2}} }, "Options",
{"n", "on", "color"});

// field forms: {var, [types], {x%,w%,line}} edit ·
{var, 0/1, pos} checkbox

// {var, n>1, pos} radio group of n · {var,
{items}, pos} chooser

CHOOSE(k, "Pick", "Euler", "Gauss", "Newton"); // k := 1..n,
0 = cancel

MSGBOX("Really?", 1) // OK/Cancel → 1/0

EDITLIST(L1); EDITMAT(M1, {"Title", {"r1", "r2"}, {"c1", "c2"}},
0);

k := GETKEY; // key id 0..50, -1 = none (non-
blocking)

IF ISKEYDOWN(4) THEN ... END; // Esc held?

m := MOUSE; // {{x,y,x0,y0,type},...} type: 0 new 1 done 2 drag
3 stretch 4 rotate 5 long

WAIT(2); WAIT(0) = 1 min; WAIT(-1) → key/mouse event (newer
fw)
```

- INPUT type matrices: `[0]` real, `[2]` string (quotes hidden), `[-1]` any; several allowed `[0,2]` .
- Max 7 lines per page; more create pages (title may be a list of titles).
- Run from Home with parameters missing ⇒ the calculator prompts for them.

Output & Text

```
PRINT("Area = "+A); terminal (scrolling
text)
```

```
PRINT(); clear terminal
MSGBOX("Done"); modal box, waits for OK
TEXTOUT_P("Hi", 10, 20, 2, #FF0000, text,x,y,font,color,clip
100, #FFFFFF); width,bg
TEXTOUT("y=x²", 0, 0); Cartesian coords in
current plot window
RECT(); FREEZE; clear screen; keep
drawing until keypress
DRAWMENU("Run", "Quit", "", "", "", "Help 6 soft labels; read taps
"); via MOUSE / GETKEY
STRING(2/3, 2, 4) → "0.6667" mode 0 cur 1 Std 2 Fix 3
Sci 4 Eng 5 Float 6
Round (+7 fraction, +14 mixed)
STRING(F1) STRING(L1) STRING(M1) object → its text form
EXPR("2+3") → 5; EXPR("X+10") string → evaluated
format(9.3456, "s3") → "9.35" CAS f/s/e formats
x := TEXTOUT_P(s, x, y); returns x after the text
```

- Fonts: 0 = Home-settings font, 1 small, 2 large (newer firmware accepts sizes 10–22 / 1–7).
- Screen 320×240 px; the bottom 22 px is the soft-menu area; the status bar is not drawable.
- After the program ends the screen is redrawn — use `FREEZE` or `WAIT` to hold graphics.
- Number display in strings follows HFormat/HDigits unless STRING mode/precision given.

Graphics

G0 = screen, **G1–G9** off-screen buffers (lost at power-off). Two coordinate systems: plain names use the current app's Cartesian window (Xmin... Ymax); `_P` versions use pixels (0,0 top-left → 319,239). Colors: `RGB(r,g,b,[a])` or `#RRGGBB` ; alpha ≥128 in RGB = transparent.

```
RECT([G],x1,y1,x2,y2,edge,fill) RECT() clears screen;
RECT(G1) clears G1
DIMGROB_P(G1, 320, 240, #FFFFFF) allocate/size a GROB
(or from list data)
PIXON_P(x,y[,c]) PIXOFF_P GETPIX_P pixels
LINE_P([G],x1,y1,x2,y2[,c]) line (also multi-
line/3D form)
ARC_P([G],x,y,r[,a1,a2,c]) circle / arc, angles
in current mode
TRIANGLE_P(x1,y1,x2,y2,x3,y3,c[,c2,c3] filled, optional
[,a]) gradient & alpha
FILLPOLY_P(({x1,y1),(x2,y2),...}, c[,a]) filled polygon
TEXTOUT_P(txt,[G],x,y[,font,c,w,bg]) text; returns end x
BLIT_P([G0,dx1,dy1,dx2,dy2], copy region (sprites;
[G1,sx1,sy1,sx2,sy2,c,a]) c = transparent
color)
```

```
SUBGROB_P(G0,x1,y1,x2,y2,G2) copy part of a GROB
into another
INVERT_P([G,x1,y1,x2,y2]) reverse video
GROBW_P(G) GROBH_P(G) size
C-PX(x,y) PX-C(x,y) convert coordinates
FREEZE; hold the display
```

DOUBLE BUFFERING

```
DIMGROB_P(G1,320,240,#FFFFFF); // back buffer
REPEAT RECT_P(G1); ... draw on G1 ...; BLIT_P(G0,G1); UNTIL
GETKEY≥0;
```

Optional args fill left to right: `RECT_P(40,90,#000000)` = x1,y1,edge. Plot-view background images: ImageName/ImageDisplay vars. Geometry-app plots use `plotfunc`, `polygon` ... (CAS).

Strings

```
DIM("abc") → 3 length (SIZE works
too)
LEFT(s,n) RIGHT(s,n) MID(s,pos[,n]) slices, 1-based
INSTRING("banana","na") → 3 find (0 = not found)
REPLACE("12345",3,"99") → "12995" overwrite from
position
ROTATE("12345",2) → "34512" rotate left (negative
= right)
UPPER(s) LOWER(s) case
CHAR(65) → "A"; CHAR({72,105}) → "Hi" codes → string
ASC("AB") → {65,66} string → codes
"n="+n; s+CHAR(10) concatenate; newline
STRING(expr[,mode,prec]) number/object → string
EXPR("3*4") → 12 string → value /
expression
cat("x=",3) → "x=3" CAS concat
s(3) → code of 3rd char indexing a string
gives the character
code (see L1(2,4) in
the guide)
STRINGFROMID(56) built-in UI strings by
id
latex(expr) LaTeX text (CAS)
```

- Escapes: `""` quote, `\n` , `\\` . Strings compare with `==` , `<` lexically.
- Build long strings in a loop with `+` ; for many pieces collect in a list.
- Number formatting: `STRING(x,2,3)` = FIX 3; `STRING(x,3,4)` = SCI 4.
- Program source is itself a string: `Programs("NAME")` , notes via `Notes("name")` .

Lists & Matrices

L1:={}; L1(0):=7	index 0 assigns past the end = append
L1 := CONCAT(L1, {v})	append (also L1(SIZE(L1)+1):=v)
MAKELIST(X^2, X, 1, 10, 1)	build from formula
L1(3); L1({2,4}); L1(2,3)	element, sublist 2..4, nested index
SIZE(L1); POS(L1,x); SORT(L1); REVERSE(L1)	basics
ΣLIST nLIST ΔLIST cumSum	reductions
DIFFERENCE INTERSECT UNION	set ops (Home)
EXECON("&1*2", L1); EXECON("&1+&2",L1,L2)	elementwise expression
map(L1, x→x^2); select(x→x>2, L1)	CAS functional forms
L1 + 1; L1 * L2; SIN(L1)	vectorized arithmetic

M1 := MAKEMAT(I*J, 3, 3); IDENMAT(3)	create (I,J = row,col)
M1(2,3); M1(2,3):=5; M1(2)	element, assign, row
SIZE(M1) → {rows,cols}; TRN DET RANK RREF	info
ADDROW(M1,{1,2,3},2) DELCOL(M1,1) SWAPROW REDIM(M1,{4,4})	commands: modify the variable in place
SUB(M2, M1, {1,1}, {2,2})	submatrix
M1*M2 M1^~1 M1.*M2 M1	algebra
EIGENVAL EIGENVV LU QR SVD LSQ CROSS dot	numeric linear algebra

Functions return values and leave variables alone (**TRN(M1)**); **commands** (ADDROW, DELCOL, SCALE, SWAPCOL, REDIM, REPLACE, SUB) change the named variable. Vectors are 1-row: **[1 2 3]** . Lists hold anything; matrices hold numbers.

Apps & Views

STARTAPP("Function")	launch app (runs its START())
STARTVIEW(1, 1)	view n, redraw
Function.CHECK(1); UNCHECK(2); ISCHECK(3)	select Symbolic definitions
F1 := "SIN(X)"; 'SIN(X)' ► F1	define a plot function
Xmin:=-5; Xmax:=5; Ymin:=-2; Ymax:=2;	plot window
D1 := L1; H1 := {'D1','',1,0,#FF:24h}	Stats 1Var data & analysis
Do1VStats(H1); MeanX; sX	run stats, read results

S1 := {'C1','C2','',1,'',#FF:24h,1,#FF:24h}	Stats 2Var: linear fit
SOLVE(X^2-2,X,1); E1:='X^2=2'	Solve app
RECURSE(U,U(N-1)*N,1,2) ► U1	Sequence app
Spreadsheet.A1:=5; =SUM(A1:A9)	Spreadsheet (qualified cell name from Home/programs)

STARTVIEW N	VIEW	N	VIEW
0	Symbolic	7	View menu
1	Plot	8... 13	Special views (Split Plot Detail, Split Plot Table, Autoscale, Decimal, Integer, Trig in Function)
2	Numeric	-1	Home
3	Symbolic Setup	-2	Home Settings
4	Plot Setup	-3	Memory Manager
5	Numeric Setup	-4	App Library
6	App Info	-5... -8	Matrix, List, Program, Notes catalogs

CUSTOM APP PROGRAM

```
// App Library ► Save (base app) → app program is 1st in catalog
EXPORT DiceSim() BEGIN END;
VIEW "Roll", ROLL() BEGIN ... STARTVIEW(1,1); END; // View menu entry
START() BEGIN D1:={}; STARTVIEW(6,1); END; // hooks: START RESET Symb
Plot() BEGIN Xmin:=0; STARTVIEW(1,1); END; // SymbSetup Plot PlotSetup Num NumSetup Info
```

View functions take no parameters — pass data through EXPORTed variables. Export those from a separate program called in START() so values persist. Icons/notes: **ANote** , **AFiles("icon.png")** .

User Keys & Key Codes

```
KEY K_Sin // Shift Menu ► Create user key writes this
BEGIN
RETURN "ALOG("; // text inserted when the key is pressed in User mode
END;
```

Prefixes: **K_** plain · **KS_** Shift · **KA_** Alpha · **KSA_** Shift+Alpha. Names (case-sensitive): Apps Symb Up Help Esc Home Plot Left Right View Cas Num Down Menu Vars Math Templ Xtn Abc Bksp Power Sin Cos Tan Ln Log Sq Neg Paren Comma Enter Eex 0–9 Div Alpha Mul Minus On Dot Space Plus. Activate: Shift Help (User) = one-shot user mode (1U in the title bar); Shift Help Shift Help = persistent (↑U) until pressed again.

GETKEY / ISKEYDOWN CODES (0 = TOP-LEFT ... 50 = BOTTOM-RIGHT)

ROW	CODES
Apps Symb ▲ Help	0 1 2 3
Esc Home Plot ◀ ▶	4 5 6 7 8
View CAS Num ▼ Menu	9 10 11 12 13
Vars Toolbox Templ xtθn a b/c ☒	14 15 16 17 18 19
x^y SIN COS TAN LN LOG	20 21 22 23 24 25
x² +/- () , Enter	26 27 28 29 30
EEX 7 8 9 ÷	31 32 33 34 35
ALPHA 4 5 6 ×	36 37 38 39 40
Shift 1 2 3 –	41 42 43 44 45
On 0 . Space +	46 47 48 49 50

Soft-menu taps and screen touches come through **MOUSE** (or **WAIT(-1)** on newer firmware), not GETKEY.

CAS from PPL

CAS.factor(x^2-1)	call any CAS function
CAS("int(sin(x),x,0,pi)")	evaluate CAS text (safest quoting)
CAS.simplify('(X^2-1)/(X-1)')	quote to stop Home evaluating X
r := CAS.solve('x^2=2', 'x');	result comes back as CAS object / list
EXPR("solve(x^2=2,x)")	alternative
CAS.zeros(...); CAS.diff(F1(X),X)	works with app functions
approx(...) evalf(...) exact(...)	numeric ↔ exact
CAS.purge(x); restart	clear CAS variables
assume(n, integer); additionally(n>5)	domain hints

- CAS names are lowercase and case-sensitive (**solve** , not **SOLVE** — that is the Solve app's function). Home functions are uppercase and also usable inside CAS.
- Uppercase single letters (X, Y...) are Home reals — inside CAS calls they evaluate to their current values unless quoted; CAS variables are lowercase and can stay symbolic.
- CAS results are exact (fractions, √) by default; wrap with **approx()** or **evalf** , or store into a Home real to force a number.
- CAS view has its own settings (Shift CAS): Exact, Complex, Use i, Simplify level, Increasing, Number Format, epsilon, recursion limits.
- Program in pure CAS: newer firmware **#CAS ... #END** blocks; older: create the function in CAS view with **name(x):=...** .
- Menu Display setting (Home Settings p.2) toggles descriptive vs command names in Math/CAS menus.

Base Integers



#1101b #17o #FFh #42d #FF:16h	literal with base marker (:bits wordsize)
#4Dh * #11101b → #8B9h	mixed bases: result in base of first operand
#100b/#11b → #1b	integer division truncates
BITAND BITOR BITXOR BITNOT BITSLSL BITSR	bitwise ops (variadic AND/OR/XOR)
B-R(#1101b) → 13; R-B(13) → #Dh	convert to/from real
SETBASE(#34o,1) → #11100b	1 bin 2 oct 3 hex
SETBITS(#1111b,15) GETBITS GETBASE	wordsize / query
Base:=3; Bits:=32; Signed:=1	Home settings vars (Base 0 bin 1 oct 2 dec 3 hex)
hamdist(#12h,#38h) → 3; shift(#5h,2)	CAS extras

- Default base is set in Home Settings ▶ Integers (hex by default), wordsize 32, optional signed (± reduces range one bit).
- Numbers without **#** are reals; mixing a real in gives a real result.
- Shift – (Base) on a highlighted result opens the Edit Integer dialog (bit view, shift, one's/two's complement).
- Colors are convenient as hex integers: **#FF0000h** is not needed — drawing commands accept plain **#FF0000**.

Idioms

t:=TICKS; ...; MSGBOX((TICKS-t)/1000)	time a section
WHILE GETKEY==–1 DO END;	wait for any key
REPEAT k:=GETKEY; UNTIL k≠–1;	same, keeping the code
IF k==4 THEN BREAK; END;	Esc exits a loop
RECT(); TEXTOUT_P(s,10,10); FREEZE;	show text until a key
L1:=CONCAT(L1,{x}); L1(0):=x	append (L1(0):=x also appends)
L1({2,SIZE(L1)})	tail of a list
ΣLIST(L1)/SIZE(L1)	mean without the Stats app
MAKELIST(RANDINT(1,6),X,1,10,1)	10 dice
1+RANDINT(5) RANDINT(1,6)	random 1–6
STRING(x,2,2)+" s"	format with 2 decimals
IFERR EXPR(s) THEN 0 END;	safe parse of user text
Ans	last result / error number in IFERR
M1(i,j):=M1(i,j)+1	update an element
EXECON("&1^2",L1)	square every element

HAngle:=1	force degrees from a program
Programs("NAME"):=src	write a program from a string
x:=INPUT(v,"t","v="); IF x==0 THEN RETURN; END;	honor Cancel
Function.Xmin:=-1	set another app's window
0 ▶ Axes (yes: 0 turns axes ON)	watch inverted flags: Axes, GridDots, Labels(1 on 2 off)

Newer Firmware Notes

The bundled User Guide is the 2018 edition (firmware 13333). Physical Primes and the 2.1.14425 macOS emulator run later firmware. Additions below are widely documented by HP/community but are **not in the guide** — verify on your unit. **Python has its own section at the end of this sheet**, checked on firmware 2.2; the lines here are only the PPL-side syntax.

```
#PYTHON myscript      // MicroPython in a PPL file (fw
2.1.14588+)

from hpprime import * // eval("PPL"), keyboard(), mouse(),
pixon...

print(eval("L1"))      // read a Home variable

#END

EXPORT RUNPY() BEGIN PYTHON(myscript); END; //
PYTHON(name,args...)

#CAS

cube(x):=x^3;          // CAS-language definitions in a
PPL file

#END
```

- A program whose first line is **#PYTHON** is a pure Python program (Python App / hpprime module; **from hpprime import *** and **from graphic import ***).
- **WAIT(–1)** blocks until a key or touch and returns the key code, or a list {type,x,y} for mouse events.
- **TEXTSIZE(str,font)** → {w,h}; TEXTOUT fonts 1–7 / pixel sizes 10–22 in later builds.
- Later firmware also added: **ICON** resources in app files, larger **Programs / Notes** support, more STRING options, **MOUSE** improvements, Python **hpprime** graphics (**fillrect**, **blit**).
- Emulator/desktop: paste source into the Program editor (Shift 1 ▶ New), or use HP Connectivity Kit; sources are compiled to bytecode on the device.

Toolbox Menus the menu tree behind the Toolbox key, drawn — plus every other menu a key opens

How the Prime's menus work

The Prime has no soft-key menus in the 48/50 sense: the six touch buttons along the bottom of the screen belong to the current view, and the big function menus live behind the **Toolbox** key (the little tool-box icon, row of Vars). It opens a full-screen menu with five tabs — **Math**, **CAS**, **App**, **User**, **Catlg** — each a list of **categories** that cascade to the right into their entries and sub-menus. Navigate by touch, or with the cursor keys and **Enter**; every entry is **numbered**, so typing **2** then **3** inside Math picks Arithmetic → Modulus without looking. **Esc** leaves the menu; there is no “up” key — the tabs are always one tap away.

Picking an entry pastes the function into the entry line with its parentheses (`MOD( , simplify( )`). By default the menus show the **descriptive names** used in the trees below; turn off *Menu Display* in Home Settings to see the command names from the index instead. Math functions are UPPERCASE and work everywhere; CAS functions are lowercase and, from a program, are called as `CAS.name( )`.

WORKING WITH LISTS

Toolbox Math 6 (List)	Make List · Sort · Reverse · Concatenate · Position · Size · ΔLIST · ΣLIST · ΠLIST
 7 (List catalog)	L0–L9 with the list editor;  8 types the braces { }
Toolbox Catlg then type L	jumps the alphabetical catalogue to the L commands
in a program	MAKELIST(X², X, 1, 5), SORT, REVERSE, CONCAT, POS, SIZE, ΔLIST, ΣLIST, ΠLIST – see Lists in the index

Menu contents follow the HP Prime User Guide (2018): “Math menu”, “CAS menu”, “App menu” in Functions & commands, and the List and Matrix chapters. Newer firmware adds entries (Python, Graph 3D) without changing the tree.

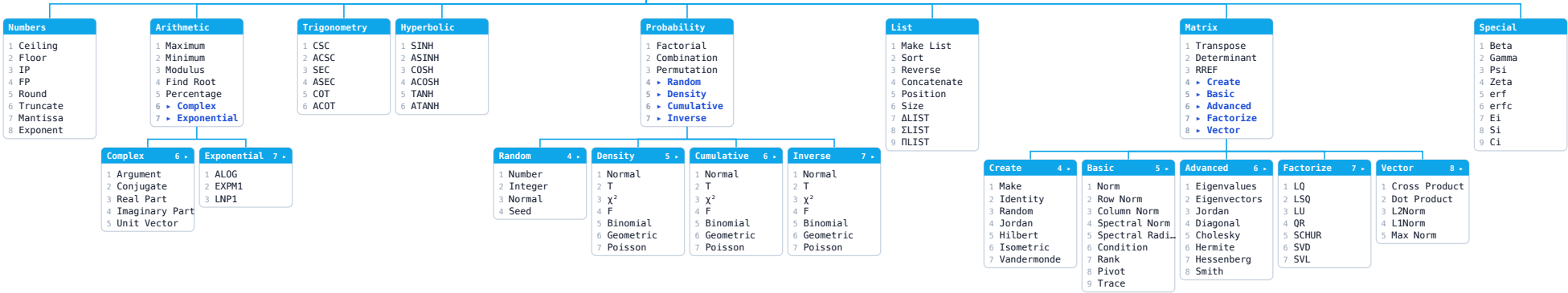
Every menu a key opens

Toolbox key	the five Toolbox menus: Math · CAS · App (functions of every app: Function, Solve, Spreadsheet, Statistics 1Var/2Var, Inference, Finance, Linear Solver, Triangle Solver, Common) · User (your EXPORTed functions) · Catlg (everything, A–Z)
Vars	variables in tabs: Home (Real, Complex, List, Matrix, Notes, Programs, Settings...) · CAS · App (the current app’s) · User
Templ	the math-template picker: fractions, roots, x , Σ, ∏, ∫, derivatives, limits, matrices, piecewise
Apps	the App Library;  Apps shows an app’s Info
Menu	the context menu of the current view;  Menu = Paste (the clipboard history)
Symb · Plot · Num	the three views of the current app;  gives each view’s Setup form
 7 List  4 Matrix	the List and Matrix catalogs (L0–L9, M0–M9) with their editors
 1 Program  0 Notes	the Program catalog (with the editor and Debug) and the Notes catalog
 Vars Chars	the character map, by page: Greek, math, arrows, box drawing...
 Templ Units	the Units menu: Length, Area, Volume, Time, Speed, Mass, Force, Energy, Power, Pressure, Temperature, Electricity, Light, Angle, Viscosity, Radiation – plus the Tools sub-menu (CONVERT, MKSA, UFACTOR, USIMPLIFY)
 Toolbox Mem	the Memory Manager;  xt0n Define opens the function definer
 Home  CAS	Home Settings (4 pages) · CAS Settings
 9  6  –  3	small pop-ups: ! ∞ → · ≤ ≥ ≠ · Base (#, b/o/d/h) · π
Help	context help for the current key, command or app;  Help arms User-key mode (1U / iU)

 is the blue Shift key; the orange ALPHA key gives the orange letters. Shifted functions are printed in blue on the key itself, alpha letters in orange at its corner — see the keyboard at the top of the page.

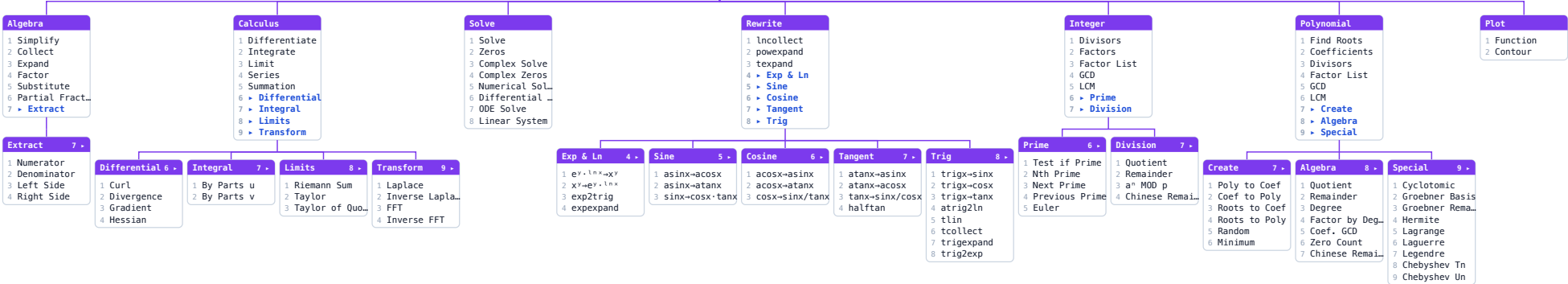
Toolbox press the Toolbox key, then the Math tab · tap a tab, then a category · type an entry's number to pick it · Esc leaves

Math CAS App User Catlg



Toolbox press the Toolbox key, then the CAS tab · tap a tab, then a category · type an entry's number to pick it · Esc leaves

Math CAS App User Catlg



STRUCTURE

- **EXPORT** Make a function global (Toolbox ▶ User menu)
- **BEGIN END** Delimit a function body / block
- **RETURN** Return a value and leave the function
- **;** Statement separator
- **//** Comment: from // to the end of the line is ignored by the...
- **#pragma** Compiler directive placed at the top of a program: forces...
- **KILL** Stop step-by-step (debug) execution
- **DEBUG** Runs a program under the debugger from Home or...

BRANCH

- **IF THEN** Run commands when test is non-zero
- **IF THEN ELSE** Two-way branch (list tests select element-wise)
- **CASE** Multi-way branch: first true IF runs, optional...
- **IFTE** Inline conditional (function form of IF)
- **IFERR** Run commands1; on error run commands2 (error...)
- **IFERR ELSE** Error trap with an ELSE branch for the no-error...

LOOPS

- **FOR** Counted loop from start to finish, +1 each pass
- **FOR STEP** Counted loop with explicit increment
- **FOR DOWN** Counted loop counting down, -1 each pass
- **FOR STEP DOWN** Counted loop counting down by increment
- **WHILE** Loop while test is true (test before each pass)
- **REPEAT** Loop until test is true (test after each pass)
- **BREAK** Exit the loop (BREAK(n): n levels)
- **CONTINUE** Jump to the next iteration

VARIABLES & SCOPE

- **LOCAL** Declare variables local to the function
- **EXPORT** Make variables global (Vars ▶ User); place above the header
- **:=** Assign: variable := expression
- **▶** Store: value ▶ variable (Shift EEX)
- **sto** CAS store: sto(value, var)
- **CopyVar** Copies the first variable into the second variable without...
- **purge** Unassign a CAS variable
- **restart** Purge all CAS variables

APP PROGRAMS

- **START** App-program hook: runs when the app is started (tapping...)
- **RESET** App-program hook: runs when the app is reset (Reset in th...)
- **Symb** App-program hook: function named Symb() in the app...
- **SymbSetup** App-program hook: runs when Shift Symb (Symbolic Setup...)
- **Plot** App-program hook: runs when the Plot key is pressed
- **PlotSetup** App-program hook: runs when Shift Plot (Plot Setup) is...
- **Num** App-program hook: runs when the Num key is pressed...
- **NumSetup** App-program hook: runs when Shift Num (Numeric Setup) i...
- **Info** App-program hook: runs when Shift Apps (Info) is pressed
- **VIEW** Add a View-menu entry that runs a function of the app...
- **KEY** Prefix for user-key definitions: KEY K_Name BEGIN...

CAS FROM PPL

- **CAS** Call a CAS function or read a CAS variable from Home/PPL
- **QUOTE** Return an expression unevaluated
- **EVAL** Evaluate an expression
- **EXPR** Parses a string into a number or expression and returns the result...
- **→** CAS mapping arrow x→expr (anonymous function)
- **#CAS** Newer firmware: everything between #CAS and #END in a P...

PYTHON

- **#PYTHON** Newer firmware (2021+, v2.1.14588 and later): embeds...
- **PYTHON** Newer firmware: runs an embedded #PYTHON block by...

DEBUGGING

- **breakpoint** Used in programming to insert an intentional stoppi...
- **halt** Used in programming to go into step-by-step debuggin...
- **TEVAL** Seconds taken to evaluate an expression
- **TICKS** Milliseconds since boot
- **TYPE** Type number of an object (0 real 1 int 2 str 3 cplx 4 mat...)
- **VERSION** Firmware/hardware/CAS version strings
- **MEMORY** Free memory / storage as a list, or one of them

Operators & Tests

ARITHMETIC

- **+** Addition symbol
- **-** Subtraction symbol
- ***** Multiplication symbol
- **/** Division symbol
- **^** Power symbol
- **x²** Square (x² key)
- **x⁻¹** Reciprocal / matrix inverse (Shift ÷)
- **√** Inserts a square root sign
- **NTHROOT** The nth root of x (Shift x^ay key inserts the ^a√ template;...
- **!** Factorial (Γ(x+1) for non-integers)
- **%** x percent of y
- **%TOTAL** y as a percentage of x
- **%CHANGE** Percent change from x to y
- **MOD** Remainder of value1/value2
- **NEG** Unary minus
- **()** Inserts opening parenthesis
- **E** Power-of-10 exponent entry (EEX key)

ELEMENT-WISE (MATRICES)

- **.*** Term by term multiplication for matrices
- **./** Term by term division for matrices
- **.^** Term by term exponentiation for matrices
- **hadamard** Hadamard bound, or element-wise product

STORE & ASSIGN

- **:=** Assign: variable := expression
- **▶** Store: value ▶ variable (Shift EEX)
- **sto** CAS store: sto(value, var)

COMPARISON

- **==** Equality test
- **<>** Inequality test
- **≠** Inequality test
- **<** Strict less-than-inequality test
- **<=** Less than or equal inequality test
- **≤** Less than or equal inequality test
- **>** Strict greater than inequality test
- **>=** Greater than or equal inequality test
- **≥** Greater than or equal inequality test
- **=** Equality symbol
- **EQ** Tests two lists for equality element by element; returns 1 ...
- **compare** Compares two objects and returns 1 if type(Obj1)...

LOGIC

- **AND** Logical And
- **OR** Logical Or
- **NOT** Returns the logical inverse of a Boolean expression
- **XOR** Exclusive or

WHERE / SUBSTITUTION

- **|** Where: substitute values / state a...
- **when** CAS inline conditional when(test,a,b)
- **PIECEWISE** Piecewise function from (condition,...)
- **IFTE** Inline conditional (function form of IF)
- **interval2center** Returns the center of an interval a..b

RPN STACK (HOME, RPN ENTRY)

- **DUP** RPN stack: duplicate level 1 (press Enter with an empty entry line)
- **DROP** RPN stack: remove level 1 (Backspace with empty entry line)
- **SWAP** You can swap the position of the objects on stack level 1 with thos...
- **ROLL** There are two roll commands: • Tap to move the selected item to stac...
- **PICK** Copies the selected item to stack level 1
- **DUPN** Duplicates all items between (and including) the highlighted item an...
- **DROPN** Deletes all items in the stack from the highlighted item down to and...
- **→LIST** RPN stack: builds a list from the highlighted item down to level 1
- **ECHO** Places a copy of the selected result on the entry line and leaves the...
- **CLEAR** RPN: Shift Esc clears the whole stack/history

ROUNDING & PARTS

- **CEILING** Round up to integer
- **FLOOR** Round down to integer
- **IP** Integer part
- **FP** Fractional part
- **ROUND** Round to n decimals (negative n = significant digits)
- **TRUNCATE** Truncate to n decimals
- **trunc** Truncate to n decimals (CAS)
- **iPart** Integer part (CAS)
- **MANT** Mantissa
- **XPON** Exponent (power of 10)
- **exact** Decimal → exact rational/real
- **float** Real (float) domain marker for assume/type
- **evalf** Numeric evaluation to n significant digits
- **format** Real → string with f/s/e format

COMPARE & SIGN

- **MAX** Maximum of values or a list
- **MIN** Minimum
- **ABS** Absolute value / modulus / Frobenius norm
- **SIGN** Sign (-1,0,1) or unit complex vector
- **MOD** Remainder of value1/value2
- **a b/c** Toggle last result decimal ↔ fraction ↔ mixed (Shift: ↔ D°M°S°)

EXPONENTIAL & LOG

- **LN** Natural logarithm
- **LOG** Common (base-10) logarithm
- **e^** Natural exponential e^x (Shift LN)
- **EXP** e^ (CAS/Home)
- **ALOG** 10^x
- **EXPM1** e^x - 1
- **LNP1** ln(1+x)
- **logb** Logarithm to base b
- **sqrt** Square root
- **surd** nth root: x^(1/n)
- **exp2pow** e^(n·ln x) → x^n
- **pow2exp** a^b → e^(b·ln a)

TRIGONOMETRIC SIN Sine COS Cosine TAN Tangent ASIN Arc sine: $\sin^{-1}x$ ACOS Arc cosine: $\cos^{-1}x$ ATAN Arc tangent: $\tan^{-1}x$ CSC Cosecant ACSC Arc cosecant SEC Secant ASEC Arc secant COT Cotangent ACOT Arc cotangent HYPERBOLIC SINH Hyperbolic sine ASINH Inverse hyperbolic sine COSH Hyperbolic cosine ACOSH Inverse hyperbolic cosine TANH Hyperbolic tangent ATANH Inverse hyperbolic tangent ANGLE & HMS -HMS Decimal $\rightarrow$ D°M'S" HMS$\rightarrow$ D°M'S" $\rightarrow$ decimal HAngle Sets the angle format for the Home view AAngle Sets the angle mode CONSTANTS π π PI π (CAS) e Euler constant e i Imaginary unit MAXREAL Largest representable real (Home 9.999999999999999E499) MINREAL Smallest positive real (Home 1E-499) SPECIAL FUNCTIONS Beta Beta function $B(a,b)$ Gamma Gamma function $\Gamma(a)$ Psi Polygamma: nth derivative of digamma at a Zeta Riemann zeta $\zeta(x)$ erf Error function erfc Complementary error function Ei Exponential integral Si Sine integral Ci Cosine integral Airy_Ai Airy function Ai Airy_Bi Airy function Bi Dirac Dirac delta Heaviside Heaviside step (1 for $x \geq 0$)	Complex Numbers15 PARTS RE Real part IM Imaginary part ARG Argument (angle) of a complex number CONJ Complex conjugate ABS Absolute value / modulus / Frobenius norm SIGN Sign $(-1,0,1)$ or unit complex vector evalc Complex expression $\rightarrow$ re + i·im affix Point $\rightarrow$ complex number CONVERT polar_point Polar $(r,\theta) \rightarrow$ point rectangular_coordinates $[r,\theta] \rightarrow [x,y]$ polar_coordinates Point $\rightarrow [r,\theta]$ normalize Vector / complex... mult_c_conjugate Multiply top and botto... HComplex Enables a complex... AComplex Sets the complex...	RANDOM NUMBERS RANDOM Random real (0–1, 0–a, a–b, or n of them) RANDINT Random integer (0/1, 0–a, a–b, or n of them) RANDNORM Random normal variate $N(\mu,\sigma)$ RANDSEED Seed the random generator srand Seed the CAS random generator randbinomial Random binomial variate randchisquare Random χ^2 variate randexp Random exponential variate randfisher Random F variate randgeometric Random geometric variate randpoisson Random Poisson variate randstudent Random t variate randvector Vector of random integers/reals ranm Random vector or matrix randMat Random integer matrix stored in a variable randpoly Random polynomial coefficients DENSITY (PDF) NORMALD Normal pdf $(\mu,\sigma$ default 0,1) STUDENT Student's t pdf CHISQUARE χ^2 pdf FISHER F pdf BINOMIAL Binomial probability of k successes GEOMETRIC Geometric pdf POISSON Poisson probability of k events betad Beta pdf cauchy Cauchy pdf exponential Exponential pdf gammad Gamma pdf negbinomial Negative binomial pdf uniform Uniform pdf weibull Weibull pdf	CUMULATIVE (CDF) NORMALD_CDF Normal lower-tail cdf STUDENT_CDF Student's t cdf CHISQUARE_CDF χ^2 cdf FISHER_CDF F cdf BINOMIAL_CDF Binomial cdf (k or k1..k2) GEOMETRIC_CDF Geometric cdf POISSON_CDF Poisson cdf betad_cdf Beta cdf cauchy_cdf Cauchy cdf exponential_cdf Exponential cdf gammad_cdf Gamma cdf negbinomial_cdf Negative binomial cdf uniform_cdf Uniform cdf weibull_cdf Weibull cdf INVERSE CDF NORMALD_ICDF Inverse normal cdf STUDENT_ICDF Inverse t cdf CHISQUARE_ICDF Inverse χ^2 cdf FISHER_ICDF Inverse F cdf BINOMIAL_ICDF Inverse binomial cdf GEOMETRIC_ICDF Inverse geometric cdf POISSON_ICDF Inverse Poisson cdf betad_icdf Inverse beta cdf cauchy_icdf Inverse Cauchy cdf exponential_icdf Inverse exponential cdf gammad_icdf Inverse gamma cdf negbinomial_icdf Inverse negative binomial cdf uniform_icdf Inverse uniform cdf weibull_icdf Inverse Weibull cdf	DESCRIPTIVE mean Mean of a list (weights... median Median stddev Sample standard... stddevp Population standard... variance Variance quartile1 First quartile quartile3 Third quartile quartiles [min, Q1, median, Q... quantile Quantile at a value 0–1 correlation Correlation coefficient covariance Covariance covariance_correlation [covariance, correlation] cumSum Cumulative sums REGRESSION & FIT linear_regression Fit $y=a \cdot x+b \rightarrow [a,b]$ exponential_regression Fit $y=b \cdot a^x \rightarrow [a,b]$ logarithmic_regression Fit $y=a \cdot \ln x+b$ logistic_regression Logistic fit $y'/y=a \cdot y+b$ polynomial_regression Best-fit polynomial... power_regression Fit $y=b \cdot x^m \rightarrow [m,b]$ linear_interpolate Regular samples of... spline Natural spline... lagrange Lagrange interpolatin...
--	--	---	---	--

Lists

44

REDUCE		
● ΣLIST	Sum of elements	
● πLIST	Product of elements	
● ΔLIST	First differences	
● deltalist	Differences of consecutive terms	
● cumSum	<i>Cumulative sums</i>	
● sum	Discrete sum Σ expr for var=a..b (2 args: discrete...	
● product	Product of expr over a range, or of a list/matrix	
● count	Sum of f over elements, or count passing a test	
APPLY A FUNCTION		
● map	Apply a function or test to each element	
● apply	Apply a function to each element	
● zip	Apply a binary function element-wise to two lists	
● EXECON	Evaluate an &-expression over list elements	
● EVALLIST	Evaluate each element of a list	
● ITERATE	Apply expr to var n times	
EDITOR		
● EDITLIST	Starts the List Editor loading listvar and displays the...	

Strings		23
BUILD & CONVERT		
● STRING	Object or number → string, with number-form...	
● EXPR	Parses a string into a number or expression a...	
● cat	Concatenate objects into a string	
● format	Real → string with f/s/e format	
● latex	Expression → LaTeX string	
● CHAR	Returns the string corresponding to the...	
● ASC	Returns a list containing the ASCII codes of string	
● STRINGFROMID	Returns, in the current language, the built-in...	
● +	Addition symbol	

INSPECT		
●	DIM	Returns the number of characters in string
●	SIZE	<i>Number of elements of a list or string; {rows,cols} of a matrix;...</i>
●	INSTRING	Returns the index of the first occurrence of str2 in str1
●	head	<i>First element</i>
●	tail	<i>All but the first element</i>
●	length	<i>Length of a list, string or sequence</i>
SLICE & EDIT		
●	LEFT	Return the first n characters of string str
●	RIGHT	Returns the last n characters of string str
●	MID	Extracts n characters from string str starting at index pos
●	ROTATE	Permutation of characters in string str
●	REPLACE	Replaces part of object1 with object2 beginning at start
●	SUB	<i>Extract a sub-object (list, matrix, string, graphic) into a variable</i>
●	LOWER	Converts uppercase characters in a string to lowercase
●	UPPER	Converts lowercase characters in a string to uppercase

Matrices & Vectors		94
CREATE		
● M0..M9	Reserved matrix/vector Home variables	
● MAKEMAT	Matrix from an expression in I (row) and J (col)	
● IDENMAT	Identity matrix of size n	
● identity	Identity matrix	
● matrix	pxq matrix of a value or a mapping of (j,k)	
● randMat	Random integer matrix stored in a variable	
● ranm	Random vector or matrix	
● randvector	Vector of random integers/real	
● JordanBlock	Jordan block matrix	
● hilbert	Hilbert matrix	
● vandermonde	Vandermonde matrix	
● mkisom	Matrix of an isometry	
● companion	Companion matrix of a polynomial	
● diag	List → diagonal matrix; matrix → diagonal vector	

SHAPE & ELEMENTS

SIZE	Number of elements of a list or string; {rows,cols} of a...
rowDim	Number of rows
colDim	Number of columns
row	Row n (or rows in an interval)
col	Column n as a vector
TRN	Transpose (conjugate transpose for complex)
transpose	Transpose (no conjugation)
REDIM	Resize a matrix/vector (pads with 0)
SUB	Extract a sub-object (list, matrix, string, graphic) into ...
subMat	Submatrix by corner indices
REPLACE	Overwrite part of a matrix/vector/list/string from ...
list2mat	Split a list into rows of n → matrix
mat2list	Matrix elements → vector

ROW / COLUMN COMMANDS

ADDCOL	Insert a column before column n
ADDROW	Insert a row before row n
DELCOL	Delete a column
DELROW	Delete a row
delcols	Delete column(s)
delrows	Delete row(s)
SWAPCOL	Swap two columns
SWAPROW	Swap two rows
rowSwap	Swap two rows
SCALE	Multiply a row by a value
mRow	Multiply a row by an expression
SCALEADD	row1 := value-row1 + row2
rowAdd	Add row i to row j
pivot	Gaussian elimination pivoting on (n,m)
EDITMAT	Open the Matrix Editor on a matrix (title, headers, read-...

FACTORIZE & SOLVE - LU {L,U,P} with P·A=L·U - lu PA=LU decomposition - LQ {L,Q,P} with P·A=L·Q - QR QR factorization: returns R, stores Q,R in vars - SCHUR Schur decomposition - SVD Singular value decomposition {U,S,V} - SVL Singular values - LSQ Least-squares solution of A·X=B - simult Solve linear system(s) in matrix form - linsolve Solve a linear system of equations - basis Basis of the span of the rows - ibasis Basis of the intersection of two spans - ker Kernel (null space) - image Image (column space) - gramschmidt Orthonormal basis for a scalar product	CAS: Algebra & Rewrite 78 SIMPLIFY & EXPAND - simplify Simplify an expression - normal Expanded irreducible (normal) form - ratnormal Rewrite as an irreducible rational... - expand Expand products and powers - collect Collect like terms; factor over the given... - factor Factor a polynomial - cFactor Factor over the complex numbers - sqrfree Square-free factorization (group... - texpend Expand transcendental functions (sin(2x),... - lin Linearize exponentials: exp(x)^n·exp(y) →... - tsimplify Rewrite transcendentals as... - canonical_form Quadratic trinomial → a(x-h)²+k - reorder Reorder variables of an expression as given	EQUATIONS & PARTS - left Left side of an equation / left end of an interval - right Right side of an equation / right end of an interval - part nth subexpression of an expression - has 1 if the variable occurs in the expression - lname List of variable names in an expression - lvar List of variable-dependent subfunctions in an expression - algvar Matrix of variables ordered by algebraic extension - subst Substitute a value for a variable - Where: substitute values / state a domain for variables in an... - id Identity function: returns its argument(s) as a vector	INVERSE-TRIG REWRITE - asin2acos asin x → π/2 - acos x - asin2atan asin x → atan(x/√(1-x²)) - acos2asin acos x → π/2 - asin x - cos2atan acos x → π/2 - atan(x/√(1-x²)) - atan2asin atan x → asin(x/√(1+x²)) - atan2acos atan x → π/2 - acos(x/√(1+x²)) - atrig2ln Inverse trig → natural logarithms VARIABLES & ASSUMPTIONS - assume State an assumption about a CAS variable... - additionally Add a further assumption (additionally(n>5)) - purge Unassign a CAS variable - restart Purge all CAS variables - sto CAS store: sto(value, var) - QUOTE Return an expression unevaluated - EVAL Evaluate an expression - evalf Numeric evaluation to n significant digits - unapply Expression + variable → function object - function_diff Derivative of a function as a mapping - type CAS type of an object (DOM_LIST,...) - float Real (float) domain marker for assume/type - when CAS inline conditional when(test,a,b) - PIECEWISE Piecewise function from (condition, expression)...	INTEGRALS - int Indefinite or definite integral ∫ Integral (template key or int) - ibpu Integration by parts given u - ibpdv Integration by parts given v - preval F(b) - F(a) - romberg Numeric definite integral (Romberg) - residue Residue of an expression at a point LIMITS & SERIES - limit Limit at a point or ±∞ (dir -1 left, 0 both, ...) - series Series expansion at a point (default ord...) - taylor Taylor expansion (default x=0, order 5) - divpc Taylor polynomial of a quotient of... - pade Padé rational approximation - order_size Remainder O-term of a series - bounded_function Marker returned by limit for a bounded... - sum_riemann Closed form of a sum seen as a...
VECTORS - CROSS Cross product - dot Dot product - l1norm l1 norm - l2norm l2 (Euclidean) norm - maxnorm l∞ norm - normalize Vector / complex divided by its norm - vector Vector between two points	FRACTIONS - numer Numerator of a simplified fraction - denom Denominator of a simplified fraction - partfrac Partial-fraction decomposition - cpartfrac Partial fractions over the complex field - propfrac A/B → Q + R/B (proper fraction) - f2nd [numerator, denominator] of a... - comDenom Sum of fractions → one fraction over a... - mult_conjugate Multiply top and bottom by the... - mult_c_conjugate Multiply top and bottom by the... - exact Decimal → exact rational/real	EXP & LOG REWRITE - lncollect ln a + n·ln b → ln(a·b^n) - lnexpand ln(3x) → ln 3 + ln x - powexpand a^(b+c) → a^b·a^c - exp2pow e^(n·ln x) → x^n - pow2exp a^b → e^(b·ln a) - expexpand exp(3x) → exp(x)^3 - exp2trig e^(ix) → cos x + i·sin x - sincos Complex exponentials → sin and cos - hyp2exp sinh/cosh/tanh → exponentials - halftan_hyp2exp trig → tan(x/2), hyperbolics → exp	TRIG REWRITE - trigexpand Expand sin(3x), cos(a+b)... - tlin Linearize trig products and powers - tcollect Linearize and collect sin/cos of the same angle - trigsin Simplify with sin²+cos²=1, preferring sin - trigcos Simplify with sin²+cos²=1, preferring cos - trigtan Simplify with sin²+cos²=1, preferring tan - trig2exp sin/cos/tan → complex exponentials - halftan sin, cos, tan → tan(x/2) - shift_phase Apply a π/2 phase shift to a trig expression - sin2costan sin x → cos x·tan x - cos2sintan cos x → sin x / tan x - tan2sincos tan x → sin x / cos x - tan2sincos2 tan x → sin 2x / (1+cos 2x) - tan2cossin2 tan x → (1-cos 2x) / sin 2x	SUMS & PRODUCTS - sum Discrete sum ∑ expr for var=a..b (2 args: discrete antiderivative) - Σ Summation template - product Product of expr over a range, or of a list/matrix - seq Vector of expr for var over an interval and step
			CAS: Calculus 40 DERIVATIVES - diff Derivative / partial derivative (x\$3 = third...) ∂ Partial-derivative template - nDeriv Numeric derivative (f(x+h)-f(x-h))/2h - function_diff Derivative of a function as a mapping - fMax x that maximizes an expression - fMin x that minimizes an expression	VECTOR CALCULUS - grad Gradient (vector of partial derivatives) - curl Curl of a vector field - divergence Divergence of a vector field - hessian Hessian matrix - laplacian Laplacian - potential Scalar potential of a gradient field - vpotential Vector potential U with curl(U)=V

TRANSFORMS laplace Laplace transform ilaplace Inverse Laplace transform fft Discrete Fourier transform (or in $\mathbb{Z}/p\mathbb{Z}$) ifft Inverse discrete Fourier transform ztrans z-transform of a sequence fourier_an nth Fourier cosine coefficient a_n fourier_bn nth Fourier sine coefficient b_n fourier_cn nth complex Fourier coefficient c_n	APP SOLVERS Solve2x2 Solve a 2x2 linear system Solve3x3 Solve a 3x3 linear system LinSolve Solve a 2x2/3x3 system from an... QuadSolve Real roots of ax^2+bx+c QuadDelta Discriminant b^2-4ac LinearSlope Slope through two points LinearYIntercept y-intercept from a point and slope AAS Triangle from angle, angle, side ASA Triangle from angle, side, angle SAS Triangle from side, angle, side SSA Triangle from side, side, angle SSS Triangle from three sides DoSolve Solve the current Triangle Solver...	EVALUATE POLYEVAL Evaluate a coefficient polynomial at a value peval Evaluate a coefficient polynomial at a value horner Evaluate a polynomial by Horner's method ptayl Taylor polynomial Q with $P(x)=Q(x-a)$ DIVIDE & GCD quo Polynomial quotient rem Polynomial remainder quorem [quotient, remainder] of polynomials gcd GCD of polynomials lcm LCM of polynomials egcd Extended polynomial GCD [U,V,D] abcuv U,V with $A\cdot U+B\cdot V=C$ modgcd Polynomial GCD (modular algorithm) ezgcd Multivariate polynomial GCD (EZ-GCD) lgcd GCD of a list/vector of integers or polynomials divis Divisors of a polynomial factors Factors with multiplicities factor_xn Factor out x^n content GCD of the coefficients icontent GCD of the integer coefficients primpart Polynomial divided by its content resultant Resultant of two polynomials syvester Sylvester matrix chinrem Polynomial Chinese remainder SPECIAL POLYNOMIALS cyclotomic Coefficients of the nth cyclotomic polynomial hermite Hermite polynomial H_n laguerre Laguerre polynomial legendre Legendre polynomial P_n tchebyshev1 Chebyshev T_n tchebyshev2 Chebyshev U_n lagrange Lagrange interpolating polynomial spline Natural spline through points randpoly Random polynomial coefficients	GRÖBNER & FORMS gbasis Gröbner basis greduce Remainder modulo a Gröbner basis GF Galois field $\text{GF}(p^n)$ a2q Symmetric matrix $\rightarrow$ quadratic form q2a Quadratic form $\rightarrow$ matrix gauss Quadratic form as sum of squares (Gauss) pmin Minimal polynomial of a matrix	MODULAR powmod $a^n \bmod p$ ichinrem Chinese remainder for $[a,p],[b,q]$ chrem Chinese remainders for two lists fracmod Fraction a/b with $n \equiv a/b \pmod{p}$ pa2b2 Prime $p \equiv 1 \pmod{4} \rightarrow [a,b]$ with $a^2+b^2=p$ jacobi_symbol Jacobi symbol Legendre_symbol Legendre symbol (or Legendre polynomial) hamdist Hamming distance between two integers
Solvers 29 SYMBOLIC solve Solve equation(s) exactly; guess or interval allowed zeros Real zeros of an expression / system cSolve Complex solutions of equation(s) cZeros Complex zeros of an expression / system linsolve Solve a linear system of equations deSolve Solve an ordinary differential equation rsolve Closed form of a recurrence relation seqsolve Closed form of a recurrence (x = previous term) NUMERIC fSolve Numeric solve (guess/interval; bisection/newton/newtonj) fsolve Numeric solve of an equation or system newton Newton iteration root estimate FNROOT Numeric root of an expression near a guess odesolve Numeric ODE solution at a final value SOLVE Solve-app: solve En for var from a guess proot Numeric roots of a polynomial (or coefficient vector) POLYROOT Zeros of a polynomial given by coefficient vector	Polynomials 55 ROOTS & COEFFICIENTS proot Numeric roots of a polynomial (or coefficient...) POLYROOT Zeros of a polynomial given by coefficient... POLYCOEF Coefficients of the polynomial with the giv... pcoef Roots $\rightarrow$ coefficients fcoeff Roots/poles with orders $\rightarrow$ rational function froot Roots and poles of a rational function with... coeff Coefficient vector of a polynomial (or one...) lcoeff Leading coefficient symb2poly Polynomial $\rightarrow$ coefficient vector poly2symb Coefficient vector $\rightarrow$ polynomial degree Degree of a polynomial valuation Lowest-degree exponent complexroot Complex roots with multiplicities / isolating... crationalroot Complex rational roots sturmab Number of sign changes / roots in $[a,b]$ sturmseq Sturm sequence		Integers & Number Theory 27 DIVISIBILITY idivis List of divisors of an integer ifactor Prime factorization of an integer ifactors Prime factors with multiplicities $[p_1,m_1,p_2,m_2,\dots]$ gcd Greatest common divisor of integers lcm Least common multiple of integers igcd GCD of integers, rationals or polynomials iegcd Extended GCD $[u,v,d]$ with $au+bv=d$ lgcd GCD of a list/vector of integers or polynomials iquo Integer quotient irem Integer remainder iquorem [quotient, remainder] iabcuv $[u,v]$ with $au+bv=c$ even 1 if even odd 1 if odd PRIMES isPrime Primality test ithprime nth prime ($n \leq 200000$) nextprime Next prime after n prevprime Previous prime before n euler Euler's totient $\phi(n)$	Binary Integers (#) 16 BASE & WORDSIZE Base Default integer base (0 bin 1 oct 2 dec 3 hex) Bits Integer wordsize Signed Signed integers on/off SETBASE Display an integer in another base GETBASE Base of an integer (0 default 1 bin 2 oct 3 hex) SETBITS Set the wordsize of an integer GETBITS Wordsize used by an integer B-R Base-marked integer $\rightarrow$ decimal real R-B Real $\rightarrow$ integer in the default base BITWISE BITAND Bitwise AND BITOR Bitwise OR BITXOR Bitwise XOR BITNOT Bitwise NOT BITSL Shift left n bits BITSR Shift right n bits shift Shift list elements (or bits of an integer)
	Input & Output 15 DIALOGS INPUT Dialog box: one or many fields $\rightarrow$ variables; returns 1 OK / 0... CHOOSE Choose box; var gets item number (0 = Cancel) MSGBOX Message box; optional OK/Cancel returns 1/0 EDITLIST Starts the List Editor loading listvar and displays the... EDITMAT Open the Matrix Editor on a matrix variable			

KEYS & TOUCH GETKEY Key id 0–50 of first buffered key, –1 if none (non-blocking) ISKEYDOWN Returns true (non-zero) if the key whose key_id is provide... MOUSE Touch/pointer state: {x,y,x0,y0,type} lists WAIT Pause n seconds (0 = 1 min; –1 = until key/touch on new... TEXT OUTPUT PRINT Write to the terminal (no argument clears it) TEXTOUT Draw text (Cartesian coords); returns end x TEXTOUT_P Draw text at pixel x,y with font/color/clip/background;... TEXTSIZE Newer firmware: returns {width, height} in pixels of a... FREEZE Hold the drawn screen until a key is pressed DRAWMENU Draws a six-button menu at the bottom of the display, wi...	COLORS & COORDINATES RGB Returns an integer number that can be used as the color parameter for... C→PX Converts from Cartesian coordinates to screen coordinates PX→C Converts from screen coordinates to Cartesian coordinates PIXELS & LINES PIXON Set a pixel (Cartesian) to a color PIXON_P Set a pixel (x,y) to a color PIXOFF Set a pixel to white (Cartesian) PIXOFF_P Set a pixel to white (pixels) GETPIX Color of a pixel (Cartesian) GETPIX_P Color of a pixel (pixels) LINE Draw a line (Cartesian); advanced form draws many... LINE_P Draw a line (pixels); advanced form draws many lines with 3... SHAPES ARC Circle or arc (Cartesian center, radius in pixels) ARC_P Circle or arc (pixel coordinates) TRIANGLE Filled triangle (Cartesian), gradient colors, alpha, z-... TRIANGLE_P Filled triangle (pixels), gradient colors, alpha, z-... FILLPOLY Filled polygon from a list of points (Cartesian) FILLPOLY_P Filled polygon from a list of pixel points TEXT TEXTOUT Draw text (Cartesian coords); returns end x TEXTOUT_P Draw text at pixel x,y with font/color/clip/background;... TEXTSIZE Newer firmware: returns {width, height} in pixels of a... DRAWMENU Draws a six-button menu at the bottom of the display, wi...	APP CONTENT VARIABLES AVars App variables: list names, read/write by index or name DeIAVars Delete an app variable by index or name AFiles Files attached to the app: names, contents, store AFilesB Binary access to app files: size, read bytes, write bytes DeIAFiles Delete a file attached to the app ANote ANote returns the note associated with an HP app AProgram AProgram returns the program associated with an... FUNCTION APP AREA Function-app function: signed area (numeric integral) betwe... EXTREMUM Function-app function: x-value of the extremum of Fn neares... ISECT Function-app function: x-value where Fn and Fm intersect,... ROOT Function-app function: root of Fn nearest to guess (Fcn ▶... SLOPE Function-app function: numeric derivative (slope) of Fn at x... F0...F9 Function-app definitions in X [Function] SEQUENCE APP RECURSE Sequence-app function used to define a sequence Un in a... U0...U9 Sequence definitions in N (set with RECURSE) [Sequence] SOLVE APP SOLVE Solve-app: solve En for var from a guess E0...E9 Solve-app equations [Solve] STATISTICS 1VAR Do1VStats Compute 1-var statistics for Hn into result vars SetFreq Set the frequency column/value of Hn SetSample Set the data column of Hn H1...H5 Stats 1Var analyses: {'data','freq',plottype,option,... D0...D9 Stats 1Var data columns (lists) [Statistics 1Var]	STATISTICS 2VAR Do2VStats Compute 2-var statistics for Sn into result vars SetDepend Set the dependent column of Sn SetIndep Set the independent column of Sn PredX Predict x from y using the first active fit PredY Predict y from x using the first active fit Resid Residuals of an analysis S1...S5 Stats 2Var analyses: {'x','y','freq','fit','expr',color,ma... C0...C9 Stats 2Var data columns (lists) [Statistics 2Var] INFERENCE DoInference Run the current Inference... HypZ1mean The one-sample Z-test for a mean HypZ2mean The two-sample Z-test for means HypZ1prop The one-proportion Z-test HypZ2prop The two-sample Z-test for comparin... HypT1mean The one-sample t-test for a mean HypT2mean The two-sample T-test for means ConfZ1mean The one-sample Normal confidenc... ConfZ2mean The two-sample Normal confidenc... ConfZ1prop The one-sample Normal confidenc... ConfZ2prop The two-sample Normal confidenc... ConfT1mean The one-sample Student's T... ConfT2mean The two-sample Student's T... Chi2GOF Chi-square goodness of fit test Chi2TwoWay Chi-square two-way test LinRegrTConfSlope The linear regression... LinRegrTConfInt The linear regression... LinRegrTMeanResp The linear regression... LinRegrTPredInt The linear regression... LinRegrTTest The linear regression t-test	FINANCE (TVM) CalcFV Solves for the future value of an investment or loan CalcIPYR Solves for the interest rate per year of an investment or loan CalcNbPmt Solves for the number of payments in an investment ... CalcPMT Solves for the value of a payment for an investment ... CalcPV Solves for the present value of an investment or loan DoFinance Solve TVM for a variable (like tapping Solve) TvmFV Solves for the future value of an investment or loan TvmIPYR Solves for the interest rate per year of an investment or loan TvmNbPmt Solves for the number of payments in an investment ... TvmPMT Solves for the value of a payment for an investment ... TvmPV Solves for the present value of an investment or loan
Graphics (G0–G9)37 SCREEN & GROBS G0...G9 Graphics variables (GROBs) RECT Draw / fill a rectangle in Cartesian coords; RECT()... RECT_P Draw / fill a rectangle in pixel coords; RECT_P(G1) clears... DIMGROB Create/size a GROB (Cartesian) filled with a colo... DIMGROB_P Create/size a GROB wxh pixels filled with a color or li... GROBW Width of a GROB GROBW_P Width of a GROB in pixels GROBH Height of a GROB GROBH_P Height of a GROB in pixels SUBGROB Copy an area of a GROB into another GROB (Cartesian) SUBGROB_P Copy an area of a GROB into another GROB (pixels) BLIT Copy a region of one GROB into another (Cartesian),... BLIT_P Copy a region of one GROB into another (pixels),... INVERT Reverse video of a region (Cartesian) INVERT_P Reverse video of a region (pixels) FREEZE Hold the drawn screen until a key is pressed	Apps & Views117 LAUNCH & VIEWS STARTAPP Launch an app by name (runs its START()) STARTVIEW Open view n of the current app (0 Symb 1 Plot 2 Num ... VIEW Add a custom option to the View menu (block form) CHECK Select (check) Symbolic definition n UNCHECK Deselect Symbolic definition n ISCHECK 1 if Symbolic definition n is checked			

FINANCE (OTHER) IntConvNom Returns the nominal interest rate IntConvEff Returns the effective interest rate IntConvCPYR Returns the number of compounding periods i... DateDays Days between dates (optional 360-day... CashFlowIRR Returns the Internal Rate of Return CashFlowMIRR Returns the Modified Internal Rate of Return CashFlowFMRR Returns the Financial Management Rate of... CashFlowTotal Calculates the total of all inputs CashFlowNPV Calculates the Net Present Value CashFlowNFV Calculates the Net Future Value CashFlowNUS Calculates the Net Uniform Series CashFlowPB Calculates the Discounted Payback... Depreciate Returns the depreciation schedule when given t... BrkEvFixed Returns the fixed cost to develop and market a... BrkEvQuant Returns the quantity of units sold BrkEvCost Returns the cost per unit BrkEvPrice Returns the unit price BrkEvProfit Returns the profit ChangePrice Calculates sales price given item cost and... ChangeCost Calculates item cost given sales price and... PercentMargin Returns the margin, a percentage of cost; tha... PercentMarkup Returns the markup, a percentage of price; th... ChangeOld Returns the old number in a percent change... ChangeNew Returns the new number in a percent change... PercentTotal Calculates the part-total percentage PercentChange Calculates the percent change BondYield Returns the yield percent to maturity (or call) dat... BondPrice Returns the bond price per 100.00 face value ... BlackScholes Returns both the call price and put price for...		SPREADSHEET Cell Spreadsheet cell access; also Spreadsheet.A1 from... SUM Calculates the sum of a range of numbers AVERAGE Calculates the arithmetic mean of a range of numbers AMORT Amortization STAT1 The STAT1 function provides a range of one-variable... STAT2 The STAT2 function provides a range of two-variable... REGRS Attempts to fit the input data to a specified function... PredY Returns the predicted Y for a given x PredX Returns the predicted x for a given y HypZ1mean The one-sample Z-test for a mean HypZ2mean The two-sample Z-test for the difference of two means HypZ1prop The one-sample Z-test for a proportion HypZ2prop <i>The two-sample Z-test for comparing two proportions</i> HypT1mean The one-sample t-test for a mean HypT2mean The two-sample T-test for the difference of two means ConfZ1mean The one-sample Normal confidence interval for a... ConfZ2mean The two-sample Normal confidence interval for the... ConfZ1prop The one-sample Normal confidence interval for a... ConfZ2prop The two-sample Normal confidence interval for the... ConfT1mean The one-sample Student's T confidence interval for a... ConfT2mean The two-sample Student's T confidence interval for the...		Geometry (CAS) 110 POINTS point Point from coordinates point2d Random-position points element Point on an object / slider a..b midpoint Midpoint center Center of a circle single_inter One intersection point near a point inter All intersections of two curves division_point Point dividing AB in ratio k barycenter Weighted barycenter isobarycenter Centroid of points orthocenter Orthocenter harmonic_conjugate Harmonic conjugate of a... harmonic_division Harmonic division polar_point <i>Polar $(r,\theta) \rightarrow point$</i> LINES segment Segment half_line Ray line Line (2 points, equation, or... parallel Parallel through a point perpendicular Perpendicular through a point perpen_bisector Perpendicular bisector tangent Tangent(s) to a curve median_line Median line altitude Altitude bisector Angle bisector exbisector Exterior angle bisector LineHorz Horizontal line y=a LineVert Vertical line x=a LineTan Tangent line to f at a value polar Polar line of a point radical_axis Radical axis of two circles vector <i>Vector between two points</i>		POLYGONS triangle Triangle isosceles_triangle Isosceles triangle right_triangle Right triangle equilateral_triangle Equilateral triangle quadrilateral Quadrilateral parallelogram Parallelogram rhombus Rhombus rectangle Rectangle square Square polygon Polygon from points isopolygon Regular polygon open_polygon Open polygonal line convexhull Convex hull vertices Vertices of a polygon vertices_abca Closed vertex list CURVES circle Circle circumcircle Circumcircle excircle Excircle incircle Incircle ellipse Ellipse hyperbola Hyperbola parabola Parabola conic Conic from an equation reduced_conic Reduced form of a conic locus Locus of a point pole Pole of a line powerpc Power of a point w.r.t. a circle		PLOTS plotfunc Plot $y=f(x)$ plotparam Parametric plot plotpolar Polar plot plotseq Sequence (cobweb) plot plotimplicit Implicit plot plotinequation Plot inequality region plotcontour Contour plot plotfield Slope field plotode ODE solution plot plotlist Polyline through points polygonplot Polylines from matrix columns polygonscatterplot Scatter+polyline from matrix... TRANSFORMS translation Translate by a vector reflection Reflect in a line or point rotate Rotate about a point homothety Dilation (scale about a point) similarity Similarity (scale + rotate) projection Orthogonal projection onto a curve inversion Inversion in a circle reciprocation Pole/polar reciprocation MEASURE abscissa x-coordinate ordinate y-coordinate coordinates Coordinates of points affix <i>Point $\rightarrow$ complex number</i> equation Cartesian equation of an object parameq Parametric equation polar_coordinates <i>Point $\rightarrow [r,\theta]$</i> distance Distance distance2 Squared distance radius Radius perimeter Perimeter slope Slope area Area of a polygon/circle or... angle Angle at a vertex arcLen Arc length extract_measure Value of a measurement...	
--	--	--	--	---	--	---	--	---	--

TESTS	
● is_collinear	Points colline...
● is_concyclic	Points on one...
● is_element	Point on object?
● is_parallel	Lines parallel?
● is_perpendicular	Lines perpen...
● is_orthogonal	Orthog... lines/ci...
● is_isosceles	Isoscel...
● is_equilateral	Equilat...
● is_parallelogram	Paralle... (1)...
● is_rectangle	Rectan...
● is_rhombus	Rhomb...
● is_square	Square?
● is_conjugate	Conjugal... points/...
● is_harmonic	Harmo... division?
● is_harmonic_circle_bundle	Harmo... circle...
● is_harmonic_line_bundle	Harmo... line...

Time, Date & Units	
CLOCK	
● TICKS	Milliseconds since boot
● Time	Clock time as H°MM°SS" (read or set)
● Date	System date YYYY.MMDD (read or set)
● Toff	Auto-off delay in ms
● WAIT	Pause n seconds (0 = 1 min; -1 = until key/touch on newer fw)
DATES (YYYY.MMDD)	
● DATEADD	Add days to a YYYY.MMDD date
● DAYOFWEEK	Day of week 1=Mon...7=Sun
● DELTADAYS	Days between two dates
● DateDays	Days between dates (optional 360-day calendar)
SEXAGESIMAL	
● →HMS	Decimal → D°M'S"
● HMS→	D°M'S" → decimal

UNITS	
● CONVERT	Convert a measurement to another unit (also bases,...
● MKSA	Express in base MKSA units
● UFACTOR	Factor a compound unit into constituent units
● USIMPLIFY	Simplify a unit expression

Variables & System	
HOME VARIABLES	
● A...Z	Reserved real-number Home variables A to Z (and 0)
● 0	Real Home variable 0 (theta), same family as A–Z. Independent...
● Z0...Z9	Reserved complex-number Home variables
● L0...L9	Reserved list Home variables
● M0...M9	Reserved matrix/vector Home variables
● G0...G9	Graphics variables (GROBs)
● Ans	Last result; Ans(n) = nth history item; error number inside IFERR

HOME SETTINGS	
● HAngle	Sets the angle format for the Home view
● HFormat	Sets the number display format used in the Home...
● HDigits	Sets the number of digits for a number format other tha...
● HSeparator	Home setting: digit-grouping / decimal separator style...
● HComplex	Enables a complex result from a real input
● Entry	Contains an integer that indicates the entry mode
● Base	Default integer base (0 bin 1 oct 2 dec 3 hex)
● Bits	Integer wordsize
● Signed	Signed integers on/off
● Language	Contains an integer indicating the system...
● Date	System date YYYY.MMDD (read or set)
● Time	Clock time as H°MM°SS" (read or set)
● Toff	Auto-off delay in ms

APP SYMBOLIC SETUP	
● AAngle	Sets the angle mode
● AComplex	Sets the complex number mode
● ADigits	Contains the number of decimal places to use for the Fixed,...
● AFormat	Defines the number display format used for number displ...

INTROSPECTION	
● HVars	Home user variables: names, read/write, function params
● DelHVars	Delete a home user variable by index or name
● Notes	Notes by index or name; assign to...
● Programs	Program sources by index or name; assign to...
● AVars	App variables: list names, read/write by index or name
● DeLAVars	Delete an app variable by index or name
● AFiles	Files attached to the app: names, contents, store
● AFilesB	Binary access to app files: size, read bytes, write bytes
● DeLAFiles	Delete a file attached to the app
● ANote	ANote returns the note associated with an HP app
● AProgram	AProgram returns the program associated with an...
● MEMORY	Free memory / storage as a list, or one of them
● VERSION	Firmware/hardware/CAS version strings
● TYPE	Type number of an object (0 real 1 int 2 str 3 cplx 4 mat ...
● type	CAS type of an object (DOM_LIST,...

App Variables: Plot & Numeric	
RANGES & AXES	
● Xmin, Xmax	Sets the minimum and maximum horizontal value...
● Ymin, Ymax	Sets the minimum and maximum vertical values o...
● Xtick	Sets the distance between tick marks for the horizont...
● Ytick	Sets the distance between tick marks on the vertical...
● Xzoom	Sets the horizontal zoom factor
● Yzoom	In Plot View, press then . Scroll to Set Factors, selec...
● Axes	Turns axes on or off
● Labels	Draws labels in Plot View showing X and Y ranges
● GridDots	Turns the background dot grid in Plot view on or off
● GridLines	Turns the background line grid in Plot View on or off
● Cursor	Sets the type of cursor
● Recenter	Recenters at the cursor when zooming
● Method	Graphing method: 0 adaptive 1 fixed segments 2 fixed d...

INDEPENDENT VARIABLE	
● Tmin, Tmax	Sets the minimum and maximum independent...
● Tstep	Sets the step size for the independent variable...
● 0min, 0max	Sets the minimum and maximum independent...
● 0step	Sets the step size for the independent variable...
● Nmin, Nmax	Defines the minimum and maximum values f...
● SeqPlot	Enables you to choose between a Stairstep or...
● Hmin, Hmax	Defines the minimum and maximum values f...
● Hwidth	Sets the width of histogram bars [Statisti...
● SImark...S5mark	Sets the mark to use for scatter plots [Statistics...
● PixSize	Sets the dimensions of each square pixel in th...
● ScrollText	Determines whether the current command in Pl...

BACKGROUND IMAGE	
● ImageName	Controls which image is set as the background i...
● ImageDisplay	Controls how a background image is...
● ImageOpacity	Controls the opacity of a background image in Pl...
● ImageXmin	Controls where the left edge of a background...
● ImageXmax	Controls where the right edge of a background...
● ImageYmin	Controls where the bottom edge of a background...
● ImageYmax	Controls where the top edge of a background...

GRAPH 3D	
● Zmin	Contains the minimum z-value for Plot view [Graph 3D]
● Zmax	Contains the maximum z-value for Plot view [Graph 3D]
● Ztick	Contains the tick mark spacing for the z-axis [Grap...
● Zzoom	Contains the zoom factor for the z-axis [Graph 3D]
● BoxAxes	Controls how the three axes are drawn [Graph 3D]
● BoxDots	Controls how grid dots are drawn on the box frame...
● BoxFrame	Controls how the box frame is drawn [Graph 3D]
● BoxLines	Controls how the grid lines are drawn [Graph 3D]
● BoxScale	Controls the scale factor used to draw the box frame [Grap...
● BoxSides	Controls which faces of the box frame are colored [Grap...
● KeyAxes	Controls whether the key axes are displayed in the top-left...
● PoseXaxis	Contains the x-coordinate of the endpoint of the rotation...
● PoseYaxis	Contains the y-coordinate of the endpoint of the rotation...
● PoseZaxis	Contains the z-coordinate of the endpoint of the rotation...
● PoseTurn	Contains the angle of rotation (in radians) of the pose axis...
● Surface	Contains a list that defines the color scheme [Graph 3D]
● FZ0..FZ9	Graph 3D definitions in X and Y [Graph 3D]

NUMERIC VIEW	
● NumStart	Sets the starting value for a table in Numeric view...
● NumStep	Sets the step size (increment value) for the independent...
● NumZoom	Sets the zoom factor in the Numeric view [Function...
● NumType	Sets the table format [Function Parametric Polar...
● NumIndep	Specifies the list of independent values (or two-...
● NumXStart	Sets the starting number for the X-values in a table in...
● NumYStart	Sets the starting value for the Y-values in a table in Numer...
● NumXStep	Sets the step size (increment value) for the independent ...
● NumYStep	Sets the step size (increment value) for the independent Y...
● NumXZoom	Advanced Graphing
● NumYZoom	Sets the zoom factor for the values in the Y column in th...

SYMBOLIC DEFINITIONS F0...F9 Function-app definitions in X [Function] E0...E9 Solve-app equations [Solve] R0...R9 Polar definitions in θ [Polar] X0,Y0...X9,Y9 Parametric definitions in T [Parametric] U0...U9 Sequence definitions in N (set with RECURSE)... H1...H5 Stats 1Var analyses: { 'data', 'freq', 'plotype', 'optio... S1...S5 Stats 2Var analyses: { 'x', 'y', 'freq', 'fit', 'expr', 'color', ... FZ0...FZ9 Graph 3D definitions in X and Y [Graph 3D] DATA COLUMNS C0...C9 Stats 2Var data columns (lists) [Statistics 2Var] D0...D9 Stats 1Var data columns (lists) [Statistics 1Var]	STATS 1VAR RESULTS NbItem Contains the number of data points in the current 1-variable... MinVal Contains the minimum value of the data set in the current 1-... Q1 Contains the value of the first quartile in the current 1-variable... MedVal Contains the median in the current 1-variable analysis (H1-H5) Q3 Contains the value of the third quartile in the current 1-variable... MaxVal Contains the maximum value in the current 1-variable analysis... ΣX Contains the sum of the data set in the current 1-variable analysi... ΣX2 Contains the sum of the squares of the data set in the current 1-... MeanX Contains the mean of the data set in the current 1-variable analysi... sX Contains the sample standard deviation of the data set in the... σX Contains the population standard deviation of the data set in the... serrX Contains the standard error of the data set in the current 1-variabl... ssX Contains the sum of the squared deviations of x for the current...
--	---

FINANCE OTHER	
● NomInt	Nominal interest rate
● EffInt	Effective interest rate
● IntCPYR	Number of times the interest compounds p...
● DateOne	The first date used in a date calculation
● DateTwo	The second date used in a date calculation
● DateDiff	The difference between the two dates
● Date360	Determines whether to use a standard...
● CFData	Cash-flow data: list of {flow,count};...
● InvestInt	The cash flow investment interest rate
● SafeInt	The cash flow safe interest rate
● CFPYR	The number of cash flows per year
● CostAsset	The depreciable cost of an asset at time of...
● SalvageAsset	The amount of money an asset can be sold ...
● FirstAsset	The month the asset is first placed into service
● LifeAsset	The expected useful life of a product
● FactorDepr	The declining balance factor as a percentag...
● FirstDateAsset	The date of first use for French-style...
● FixedCost	The fixed cost of developing and...
● Quantity	The number of units sold
● VariableCost	The manufacturing cost per unit
● SalePrice	The sales price per unit
● Profit	The expected profit
● Cost	The cost of an item in markup calculations
● Price	The sales price in markup calculations
● Margin	The margin in markup calculations based on...
● Markup	The markup percentage in markup calculations
● OldValue	The old value in percent-change...
● NewValue	The new value in percent-change...
● Total	The percentage of the total in part-total...
● Change	The percent change in percent-change...
● SetDate	The settlement date of a bond

● MatDate	The maturity date or call date of a bond
● CpnPer	The coupon percentage
● CallPrice	The call price or value
● YieldBond	The yield percent to maturity of a bond
● PriceBond	The price per 100.00 value of a bond
● Bond360	Determines whether to use a standard...
● SemiAnnual	Determines whether payments are made ...
● Accrued	The accrued interest of a bond
● Modified	The modified Macaulay duration of a bond
● Macaulay	The Macaulay duration of a bond
● StockPrice	The stock price
● StrikePrice	The strike price
● TimeMarket	The time to maturity of an option
● RiskFree	The risk free interest rate
● Volatility	The volatility of an asset
● Dividend	The dividend percentage
● BSCallPrice	The call price of an option
● BSPutPrice	The put price of an option
LINEAR & TRIANGLE SOLVER	
● LSystem	Contains a 2x3 or 3x4 matrix which represents a 2x2 or 3x3...
● SideA	The length of Side a
● SideB	The length of Side b
● SideC	The length of Side c
● AngleA	The measure of angle A. Sets the measure of angle A. The...
● AngleB	The measure of angle B. Sets the measure of angle B. The...
● AngleC	The measure of angle C. Sets the measure of angle C. The...
● TriType	Corresponds to the status of in the Numeric view of the Triang...

Python on the HP Prime

MicroPython 1.9.4 · the hpprime bridge · every importable module — module lists and hpprime signatures probed on this Prime (firmware 2.2) on 2026-08-30

Where Python lives — and three ways to run it

Firmware 2.1.14181 (2021) and later carry **MicroPython 1.9.4** — Python 3.4 syntax — as the **Python app** plus a bridge module, `hpprime`. Your Prime reports `sys.version 3.4.0`, `sys.implementation (micropython, 1.9.4)`, `sys.platform 'HP Prime'` and a **1,024,512-byte heap**.

```
#PYTHON blockname          // 1. a block inside any PPL
program
import sys
print("hi", sys.argv)

#END

EXPORT run(a)

BEGIN
    PYTHON(blockname, a);    // run it; the arguments arrive in
    sys.argv
END;

#PYTHON EXPORT f(a, b)      // 2. the same block callable like
a function
...

#END                        // then simply f(1, 2) from
PPL or Home

PYTHON({blockname, 4000000}, a) // heap in bytes for this
run (G1 ≤ ~16 MB, G2 ≤ ~250 MB)
```

3. The Python app (Apps → Python): **Symb** is the script editor (Tmplt and Cmds menus; select a command and press Help for its page), **Num** is the console — single-line entry, output history above. The console auto-imports every script of the app as `import name`, so call `name.func()` or type `from name import *` yourself.  Plot opens the app settings, where the heap size is set. **On** interrupts a running script.

Block names carry no `.py`; `PYTHON(name)` takes the bare name, unquoted. Keep the block name different from the `EXPORT` name. From the Mac, `tools/tohp --program file.py` wraps a script this way and sends it; the Connectivity Kit edits programs and drops `.py` files onto the Python app.

The bridge: hpprime.eval and passing values

```
hpprime.eval("expr")      evaluate any Home/PPL
                           expression or statement and
                           return its value as a
                           Python object — an int for
                           integers and colours,
                           float, str, list for {...},
                           nested lists for matrices

eval("L1")    eval("A")    read a Home variable

eval("A:=7")   eval("L1:=
{1,2,3}")       store into Home variables
                (also AVars("name"):=...)
```

```
eval('CAS.eval("int(ln(x),x)')')  run the CAS from Python —
or import cas;
cas.caseval("int(ln(x),x)")

eval("NORMALD_CDF(2)")           every command in the index
eval("RGB(255,0,0)")             is one string away

eval("TICKS")    eval("Time")    millisecond counter · clock
                                as decimal hours (19.9030 =
                                19:54:11) — there is no
                                time module

eval("WAIT(0.1)")               delay · last key code or -1
eval("GETKEY")                  · is Esc held
eval("ISKEYDOWN(4)")

eval('MSGBOX("done")')          dialogs and input forms
eval('INPUT(A,"t","a=")')       come from PPL

sys.argv                       the PYTHON(name, a, b, ...)
                                arguments, as
                                strings/numbers in a list —
                                index from 0

print(...)                     goes to the Python console
                                (Num view) or the terminal
                                a PPL program was printing
                                to

input("a: ")                   prompts on the terminal and
                                returns a str — cast with
                                int() / float()

r := PYTHON(block, ...)        always 0 — a block's last
                                expression is not returned
                                to PPL; hand results back
                                through variables, AVars or
                                Notes

eval("Programs")  eval("Notes") catalogs as Python lists of
eval("HVars")    eval("AFiles") names

eval("HAngle")   eval("Bits")   settings come back as
eval("Date")     floats: HAngle 0.0 =
                                radians · Bits 32.0 · Date
                                2026.083 = YYYY.MMDD
```

Getting a result *back* to PPL: store it with `eval("V:=...")` or `eval('AVars("X"):=...')`, or write a Note with `eval('Notes("R"):=...')` (how this sheet's probes reported). Inside that PPL text a backslash or a double quote ends the string early — strip them (`repr()` of a multi-line string puts `\n` in) — and numbers like `2e+1` can trip the parser: format them yourself or go through `cas.caseval`.

Graphics with hpprime — the G0–G9 buffers

The first argument of every drawing call is the graphic number: **0 is the screen**, 1–9 are the off-screen buffers `G1 – G9` shared with PPL. Colours are **24-bit ints** (`0xFF0000` red, or `eval("RGB(r,g,b)")`); the screen is `320 × 240`, (0,0) top-left. Each function has a `_c` twin using the Cartesian window from `set_cartesian`.

```
dimgrob(G, w, h, color)      create/resize G1–G9
```

```
pixon(G, x, y, color)        one pixel

line(G, x1, y1, x2, y2, color) line

rect(G, x, y, w, h, color)    rectangles — outline
fillrect(G, x, y, w, h, edge, fill) in one colour, or
                                filled with edge +
                                fill

circle(G, x, y, r, color)    arc(G, x, y, r, a1, a2, color) circles and arcs

textout(G, x, y, text, color) text in the current
                                font — five
                                arguments, nothing
                                more (checked on this
                                unit); there is no
                                font parameter

blit(G1, x, y, G2)           copy G2 onto G1 —
                                draw a frame on G1,
                                then blit(0, 0, 0, 1)

strblit(G1, x, y, w, h, G2)  scaled copy of all,
strblit2(G1, x, y, w, h, G2, x2, y2, w2, h2) or a part, of G2

grobw(G)  grobh(G)           size of a buffer

eval("GETPIX_P(G1,x,y)")     there is no pixel
                                read in hpprime —
                                borrow PPL's

graphic.draw_line(x1, y1, x2, y2, c) the KhiCAS-style
...                                module draws on the
                                screen with Casio-
                                like names — see the
                                index
```

Keys and touch

```
k = hpprime.keyboard()      non-blocking bitmask
                                of every key held
                                right now

if k & (1 << 4): ...         test a key by its
                                GETKEY code (Esc = 4,
                                Enter = 30, ▲ 2 ▼ 12
                                ◀ 7 ▶ 8)

k & (1<<41)  k & (1<<36)     Shift and ALPHA are
                                ordinary bits;
                                several keys at once
                                are fine

hpprime.mouse()              touch: a list for two
                                fingers (x, y, ... or
                                -1 when up) — poll it
                                in a loop

eval("GETKEY")               PPL-style: the last
                                key pressed, -1 if
                                none
```

```
while True: ... if eval("ISKEYDOWN(4)": the usual exit test
break
```

On always interrupts the script – it is not readable

Codes: Apps 0 · Symb 1 · ▲ 2 · Help 3 · Esc 4 · Home 5 · Plot 6 · ◀ 7 · ▶ 8 · View 9 · CAS 10 · Num 11 · ▼ 12 · Menu 13 · Vars 14 · Toolbox 15 · Templ 16 · xtθn 17 · a b/c 18 · ☒ 19 · x' 20 · SIN 21 · COS 22 · TAN 23 · LN 24 · LOG 25 · x² 26 · +/- 27 · () 28 · , 29 · Enter 30 · EEX 31 · 7 32 · 8 33 · 9 34 · ÷ 35 · ALPHA 36 · 4 37 · 5 38 · 6 39 · × 40 · Shift 41 · 1 42 · 2 43 · 3 44 · – 45 · On 46 · 0 47 · , 48 · _ 49 · + 50

The language — MicroPython 1.9.4 is Python 3.4

<code>x = 3; y = 2.5; z = 3+4j</code>	ints are arbitrary precision (<code>sys.maxsize 2³¹-1</code> for the fast path), floats are double, complex built in
<code>7 // 2 7 % 2 2 ** 10 divmod(7, 2)</code>	integer division, modulo, power
<code>"abc"[1:] "a,b".split(",") "{} {:.2f}".format(a, b) "%d" % n</code>	strings: slicing, methods, <code>.format</code> and <code>%</code> – no f-strings (they arrived in 1.17)
<code>[x*x for x in range(5) if x%2] {k: v for ...} {a, b}</code>	list / dict / set comprehensions; generators with <code>(...)</code> and <code>yield</code>
<code>def f(a, b=1, *args, **kw): return a</code>	defaults, <code>*args</code> , <code>**kwargs</code> , keyword arguments, <code>lambda</code> , closures, <code>nonlocal</code> , <code>global</code>
<code>class P(Base): def __init__(self, x): self.x = x</code>	classes, inheritance, <code>super()</code> , properties, <code>@staticmethod</code> / <code>@classmethod</code> , <code>__str__</code> , <code>__eq__</code> , <code>__lt__</code>
<code>try: ... except ValueError as e: ... finally: ...</code>	exceptions, <code>raise</code> , custom exception classes, <code>assert</code>
<code>with open(...) as f:</code>	context managers work (uio files); no real file system though
<code>for i, v in enumerate(lst): ... zip map filter sorted(key=) reversed any all min max sum</code>	the usual iteration toolkit
<code>int("ff", 16) bin(x) hex(x) x.to_bytes(4, "little") bytes / bytearray / memoryview</code>	bits and bytes; <code>ustruct</code> for records
<code>dir(obj) help(obj) type(x) isinstance eval("1+1") exec(src)</code>	introspection and dynamic code (<code>eval</code> here is Python's, not <code>hprime.eval</code>)

Not in 3.4 / 1.9.4: f-strings, the walrus `:=`, `async/await` keywords beyond the basics, type hints are parsed but ignored, `dataclasses`, `match`. Recursion is shallow (~77 levels on a G1, ~99 on a G2) unless `PYTHON()` is given a bigger stack.

Standard modules — as imported on this Prime

Present (probe, 2026-08-30): `hprime`, `cas`, `graphic`, `matplotlib`, `linalg`, `arit`, `math`, `cmath`, `urandom`, `sys`, `gc`, `micropython`, `array`, `ucollections`, `uerrno`, `uhashlib`, `uio`, `ure`, `ustruct`, `utimeq`, `builtins`. Everything else in the MicroPython manual is **absent** – notably `time`, `utime`, `os`, `json`, `random`, `collections`, `io`, `re`, `struct`, `itertools`, `functools`, `string`, `copy`: use the `u`-prefixed names (`urandom`, `ucollections`, `uio`, `ure`, `ustruct`) and reach PPL for time.

math	floats only – use <code>**</code> and <code>//</code> on ints – <code>acos</code> <code>acosh</code> <code>asin</code> <code>asinh</code> <code>atan</code> <code>atan2</code> <code>atanh</code> <code>ceil</code> <code>copysign</code> <code>cos</code> <code>cosh</code> <code>degrees</code> <code>e</code> <code>erf</code> <code>erfc</code> <code>exp</code> <code>expm1</code> <code>fabs</code> <code>floor</code> <code>fmod</code> <code>frexp</code> <code>gamma</code> <code>isfinite</code> <code>isinf</code> <code>isnan</code> <code>ldexp</code> <code>lgamma</code> <code>log</code> <code>log10</code> <code>log2</code> <code>modf</code> <code>pi</code> <code>pow</code> <code>radians</code> <code>sin</code> <code>sinh</code> <code>sqrt</code> <code>tan</code> <code>tanh</code> <code>trunc</code>
cmath	complex numbers: <code>3+4j</code> , <code>abs(z)</code> , <code>z.real</code> – <code>cos</code> <code>e</code> <code>exp</code> <code>log</code> <code>log10</code> <code>phase</code> <code>pi</code> <code>polar</code> <code>rect</code> <code>sin</code> <code>sqrt</code>
random / urandom	<code>import random</code> or <code>urandom</code> – same module – choice <code>getrandbits</code> <code>randint</code> <code>random</code> <code>randrange</code> <code>seed</code> <code>uniform</code>
sys	<code>platform == 'HP Prime'</code> ; <code>argv</code> carries <code>PYTHON()</code> arguments – <code>argv</code> <code>byteorder</code> <code>exc_info</code> <code>exit</code> <code>implementation</code> <code>maxsize</code> <code>modules</code> <code>path</code> <code>platform</code> <code>print_exception</code> <code>stderr</code> <code>stdin</code> <code>stdout</code> <code>version</code> <code>version_info</code>
gc	heap accounting – see Limits – <code>collect</code> <code>disable</code> <code>enable</code> <code>isenabled</code> <code>mem_alloc</code> <code>mem_free</code> <code>threshold</code>
micropython	<code>const(1)</code> makes a compile-time constant – <code>const</code> <code>heap_lock</code> <code>heap_unlock</code> <code>kbd_intr</code> <code>mem_info</code> <code>opt_level</code> <code>pystack_use</code> <code>qstr_info</code> <code>stack_use</code>
array	typed arrays: <code>array("i", [1,2,3])</code> – <code>array</code>
ucollections	<code>import ucollections</code> – <code>OrderedDict</code> <code>deque</code> <code>namedtuple</code>
uerrno	<code>errno</code> constants – <code>EACCES</code> <code>EADDRINUSE</code> <code>EAGAIN</code> <code>EALREADY</code> <code>EBADF</code> <code>ECONNABORTED</code> <code>ECONNREFUSED</code> <code>ECONNRESET</code> <code>EEXIST</code> <code>EHOSTUNREACH</code> <code>EINPROGRESS</code> <code>EINVAL</code> <code>EIO</code> <code>EISDIR</code> <code>ENOBUFS</code> <code>ENODEV</code> <code>ENOENT</code> <code>ENOMEM</code> <code>ENOTCONN</code> <code>EOPNOTSUPP</code> <code>EPERM</code> <code>ETIMEDOUT</code> <code>errorcode</code>
uhashlib	SHA-256 only – <code>sha256</code>
uio	in-memory files; <code>StringIO</code> for building text – <code>BytesIO</code> <code>FileIO</code> <code>StringIO</code> <code>TextIOWrapper</code> <code>open</code>
ure	small regex engine: <code>groups</code> , <code>classes</code> , no lookahead – <code>DEBUG</code> <code>compile</code> <code>match</code> <code>search</code>
ustruct	binary packing – <code>calcsz</code> <code>pack</code> <code>pack_into</code> <code>unpack</code> <code>unpack_from</code>
utimeq	timer queue used by schedulers – <code>utimeq</code>

```

builtins      plus the exception classes — ArithmeticError
               AssertionError AttributeError BaseException
               EOFError Ellipsis Exception GeneratorExit
               ImportError IndentationError IndexError
               KeyError KeyboardInterrupt LookupError
               MemoryError NameError NotImplemented
               NotImplementedError OSError OverflowError
               RuntimeError StopAsyncIteration
               StopIteration SyntaxError SystemExit
               TypeError UnicodeError ValueError
               ZeroDivisionError abs all any bin bool
               bytearray bytes callable chr classmethod
               compile complex delattr dict dir divmod
               enumerate eval exec filter float frozenset
               getattr globals hasattr hash help hex id
               input int isinstance issubclass iter len
               list locals map max memoryview min next
               object oct open ord pow print property range
               repr reversed round set setattr slice sorted
               staticmethod str sum super tuple type zip

```

HP's own modules — hpprime, cas, graphic, matplotlib, linalg, arit

hpprime (arc arc_c blit blit_c circle circle_c dimgrob dimgrob_c eval fillrect fillrect_c get_cartesian grobh grobh_c grobw grobw_c keyboard line line_c mouse pixon pixon_c rect rect_c set_cartesian strblit strblit2 strblit2_c strblit_c textout textout_c) is HP's bridge and fast-drawing module, detailed in the cards above. The other five come from Bernard Parisse's Xcas/KhiCAS work and mirror what NumWorks and Casio users know:

cas	caseval xcas eval_expr — the CAS from Python, results as strings; get_key is a stub (returns 0)
graphic	black blue clear clear_screen cyan draw_arc draw_circle draw_filled_arc draw_filled_circle draw_filled_polygon draw_filled_rectangle draw_line draw_pixel draw_polygon draw_rectangle draw_string fill_rect get_pixel green magenta red set_pixel show show_screen white yellow — pixel drawing on the screen with Casio/KhiCAS names, colour constants included; show_screen() waits for a key
matplotlib	arrow axis bar barplot boxplot boxwhisker clf grid hist histogram linear_regression_plot plot scatter scatterplot show text vector — a pyplot subset: plot, scatter, bar, hist, boxplot, text, arrow, axis, grid, show
linalg	abs add apply arange conj cross det dot egv eig eigenvecs eye fft horner identity idn ifft im imag inv linspace matrix mul ones pcoeff peval pi proot rand ranm ranv re real rref shape size solve sub transpose zeros — a numpy.linalg stand-in: matrix, zeros, eye, inv, det, solve, eig, rref, fft...
arit	asc char euler gcd iegcd ifactor isprime lcm nextprime nprimes prevprime — integer arithmetic: gcd, lcm, iegcd, isprime, ifactor, nextprime, euler

Function-by-function detail is in the Python index below; the Cmds menu of the Python app lists the same names and Help explains each one.

Limits and gotchas

- **Heap:** 1 MB by default in the app (about 1,016 KB free after `import gc`); raise it in  Plot settings or per call with `PYTHON({name, bytes})`. `gc.mem_free()` tells you where you stand; `gc.collect()` reclaims.
- **No clock, no sleep:** `time` is absent. Busy-wait on `eval("TICKS")`, or `eval("WAIT(0.1)")`.
- **Console is one line:** multi-line statements only in scripts; copy a history line back with the touch menu.
- **Names:** the console imports scripts as modules — qualify calls or `from x import *`. Blocks in PPL programs are invisible to the app.
- **Types across the bridge:** eval returns int for whole numbers (including based numbers and colours), float otherwise; lists ↔ lists; strings need quoting inside the PPL text: `eval('"abc"')`.
- **Keys from the Mac:** remote key injection drops uppercase A and E (firmware 2.2) — `tohp` works around it; irrelevant on the keyboard itself.
- **Blocking calls:** `graphic.show_screen()`, `matplotlib.show()` and `input()` wait for a key — a script driven over USB looks frozen until someone presses one.
- **Exam mode** can disable Python; scripts survive firmware updates.

Idioms

```

import hpprime as h

t0 = h.eval("TICKS")

while h.eval("TICKS") - t0 < 500: pass           # 0.5 s delay

h.dimgrob(1, 320, 240, 0xFFFFF)                # double
buffer

for i in range(100):

    h.fillrect(1, 0, 0, 320, 240, 0xFFFFF, 0xFFFFF)

    h.circle(1, 160 + i, 120, 20, 0, 0xFF0000)

    h.blit(0, 0, 0, 1)

    if h.keyboard() & (1 << 4): break           # Esc

xs = h.eval("L1")                               # list in
from Home

h.eval("L2:={\" + \",\".join(str(x*x) for x in xs) + \"}") #
list out

import cas

print(cas.caseval("solve(x^2-2=0,x)"))           # CAS as text

from linalg import *

A = matrix([[2,1],[1,3]]); print(solve(A, [1,2]), det(A))

import matplotlib as plt

plt.plot([0,1,2,3],[0,1,4,9]); plt.grid(True); plt.show() #
show() waits for a key

```

Python:_hpprime19

BRIDGE

eval

eval(ppl_text) — Evaluate a PPL/Home expression or staterne...

INPUT

keyboard

keyboard() — Non-blocking: an int bitmask of every key...

mouse

mouse() — Touch state for two fingers as a pair of tuples — (...

GRAPHICS

dimgrob

dimgrob(G, w, h, color) — Create or resize...

grobw

grobw(G) — Width in pixels of graphic G (0 ...

grobh

grobh(G) — Height in pixels of graphic G (0 ...

pixon

pixon(G, x, y, color) — Light one pixel of G at ...

line

line(G, x1, y1, x2, y2, color) — Line on G fro...

rect

rect(G, x, y, w, h, color) — Rectangle at (x, y),...

fillrect

fillrect(G, x, y, w, h, edge, fill) — Filled rectangle...

circle

circle(G, x, y, r, color) — Circle of radius r centr...

arc

arc(G, x, y, r, a1, a2, color) — Arc of radius ...

textout

textout(G, x, y, text, color) — Draw text on ...

blit

blit(G1, x, y, G2) — Copy graphic G2 onto G1 at...

strblit

strblit(G1, x, y, w, h, G2) — Stretch-blit: all of G...

strblit2

strblit2(G1, x, y, w, h, G2, x2, y2, w2, h2) —...

get_cartesian

get_cartesian() — The Cartesian window use...

set_cartesian

set_cartesian(xmin, ymin, xmax, ymax) —...

*_c

pixon_c line_c rect_c fillrect_c circle_c arc_c...

Python:_cas & graphic19

CAS

caseval

caseval(text) — Evaluate text in the CAS and return the...

xcas

xcas(text) — Same engine, Xcas syntax; the three entry...

eval_expr

eval_expr(text) — Evaluate a CAS expression

get_key

get_key() — Documented as a key read; returns 0...

GRAPHIC

clear_screen

clear_screen() — Clear the...

show_screen

show_screen() — Push the...

set_pixel

set_pixel(x, y, c) — Set pixel (x,...

get_pixel

get_pixel(x, y) — Read a pixel...

draw_pixel

draw_pixel(x, y, c) — Same as...

draw_line

draw_line(x1, y1, x2, y2, c) — Line

draw_rectangle

draw_rectangl... y, w, h, c) —...

draw_polygon

draw_polygon(... [x2,y2],...], c) ...

draw_filled_polygon

draw_filled_pol... c) — Filled...

draw_circle

draw_circle(x, y, r, c) — Circle

draw_filled_circle

draw_filled_cir... y, r, c) — Filled...

draw_arc

draw_arc(x, y, rx, ry, t1, t2, c) —...

draw_filled_arc

draw_filled_arc... y, rx, ry, t1, t2, ...

draw_string

draw_string(s, x, y, c) — Text at ...

colours

black white red green blue cya...

Python:_matplotlib, linalg, arit34

MATPLOTL

plot

plot(x, y [, style]) —...

scatter

scatter(x, y) — Points...

bar

bar(x, h) — Bar chart...

hist

hist(data, bins) —...

boxplot

boxplot(data) — Box-an...

linear_regression_plot

linear_regr... y) — Point...

arrow

arrow(x, y, dx, dy) —...

vector

vector(x, y, dx, dy) —...

text

text(x, y, s) — Label at...

axis

axis(xmin, xmax, ymi...

grid

grid(True) — Toggle the...

clf

clf() — Clear the figure

show

show() — Display the...

LINALG

matrix

matr... [...]...

zeros ones eye identity idn

zero... m)...

arange linspace

aran... b,...

rand ranm ranv

rand() ran...

add sub mul dot cross

add(A, B)...

transpose inv det rref

tran... inv(...

solve

solv... b) ...

eig egv eigenvects

eig(A) egv(...

fft ifft

fft(v) ifft(v...

apply

appl... A) ...

shape size

sha... size...

re im real imag conj abs

re(A) im(A...

pcoeff proot peval horner

pco... proo...

pi

pi — π

ARIT

gcd lcm

gcd(a, b)...

iegcd

iegc... b) ...

isprime

ispri... — ...

nextprime prevprime nprimes

next... prev...

ifactor

ifact... — ...

euler

eule... — ...

asc char

asc(s) char...

Python: standard modules15

IMPORTABLE ON THIS PRIME (VERIFIED)

math

import math — floats only — use ** and // on ints ...

cmath

import cmath — complex numbers: 3+4j, abs(z),...

urandom

import urandom — import random or urandom —...

sys

import sys — platform == 'HP Prime'; argv carries...

gc

import gc — heap accounting — see Limit...

micropython

import micropython — const(1) makes a compli...

array

import array — typed arrays: array("i", [1,2,3])...

ucollections

import ucollections — import ucollections ...

uerrno

import uerrno — errno constants · members:...

uhashlib

import uhashlib — SHA-256 only · members:...

uio

import uio — in-memory files; StringIO for buildin...

ure

import ure — small regex engine: groups, classes,...

ustruct

import ustruct — binary packing · members:...

utimeq

import utimeq — timer queue used by schedul...

builtins

import builtins — plus the exception classes ...

ABSENT

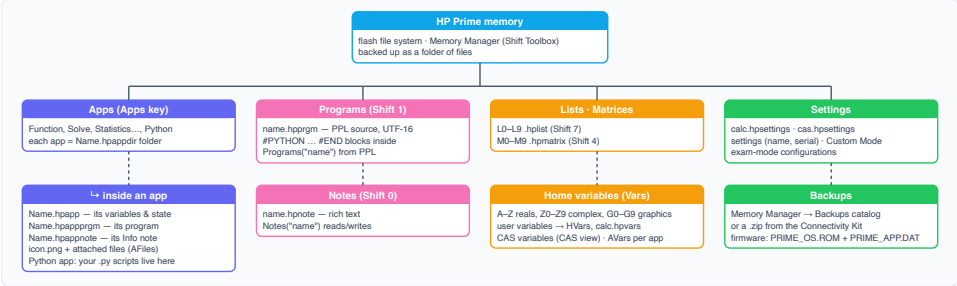
not available

— time / utime (no sleep, no clock — use...

Files & Hardware what the calculator stores, and what it is made of

Software files on the Prime

Everything you create is an object in the calculator's flash file system, and every object has a file form the Connectivity Kit shows as a folder tree: **Programs, Notes, Lists, Matrices, Apps** (each a folder), **Variables**, settings, exam configurations. This unit reports itself as **software 2.2** (revision 15157, built 2024-09-01), CAS 1.5.0, MicroPython 1.9.4, hardware version C.

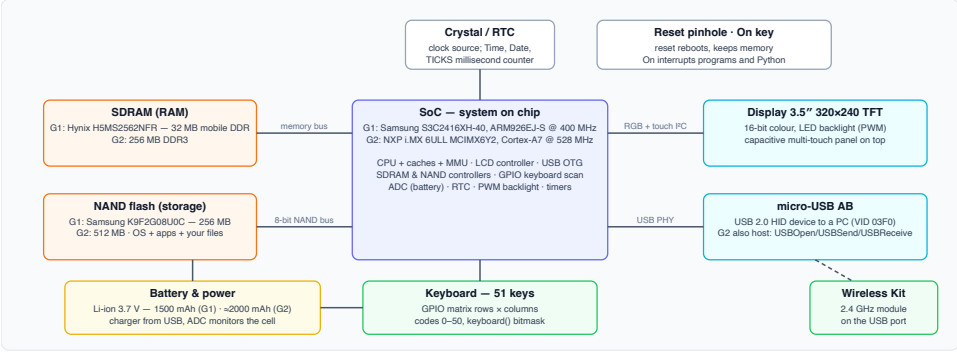


From a program, the same tree is a set of lists: `Programs`, `Notes`, `AFiles`, `AVars`, `HVars` each return the names (Python gets them as lists through `hprime.eval()`; `VERSION` returns the About text.

OBJECT	FILE · WHERE	FORMAT	REACH IT
Program	.hpprgm — Program catalog (1)	UTF-16LE text, BOM optional; PPL source, may hold #PYTHON...#END and #cas...end blocks	Programs("name") reads/writes the source from PPL; the Connectivity Kit editor; tohp --program from the Mac
App	Name.hpappdir — a folder (Apps key)	Name.hpapp (the app's variables and state, binary), Name.hpappprgm (its program), Name.hpappnote (its Info note), icon.png, plus any attached files	AFiles / AFilesB list, read and write the attached files; AVars the variables; the Connectivity Kit app editor has Modes, Program, Info, Files, Variables tabs
Python script	.py — attached to the Python app	plain text; the app's Symb view edits them, the Num console imports them (import name — call name.func())	drop .py files onto the Python app in the Connectivity Kit; or keep Python inside a .hpprgm as a #PYTHON block
Note	.hpnote — Notes catalog (0)	UTF-16LE rich text	Notes("name") from PPL — a handy place for a script to write results (this sheet's probe did exactly that)
List / Matrix	L0–L9 .hplist · M0–M9 .hpmatrix (7 / 4)	small binary records (an empty list is 8 bytes, an empty matrix 24)	the List and Matrix editors; L1 := {...} from Home or a program
Home variables	calc.hpvars	binary container (4-byte magic 7C 61 8A B2, shared by settings and app files)	A–Z, Z0–Z9 (complex), user variables; Vars key → Home
Settings	calc.hpsettings · cas.hpsettings · settings · Custom Mode	binary; “settings” carries the calculator name and serial number	Home (Home Settings), CAS; Custom Mode is the exam/custom mode definition
Exam & class	exam-mode configurations, .hppoll / .hpreult	Connectivity Kit objects	Exam Mode (Esc from the Apps screen), polls sent by a teacher
Backup	Backups catalog on the calculator · .zip from the Connectivity Kit	a snapshot of everything above	Memory Manager (Toolbox) → Backups; libhpcalcs “recv backup” drops every object as a file
Firmware	PRIME_OS.ROM + PRIME_APP.DAT (about 12 MB)	signed OS image	installed by the Connectivity Kit, or from a USB drive on a G2; the calculator keeps user data across updates

Electronics — CPU, memory and I/O

Two generations share one design: a single ARM system-on-chip with everything on it, one RAM chip, one NAND flash chip, and a keyboard/power board — HP used ordinary documented parts, no custom ASIC. The G1 (2013) is a 400 MHz ARM9; the G2 (2018) is a 528 MHz Cortex-A7 with eight times the RAM and roughly three times the speed.



HP PRIME G1

Model	NW280AA (2013) · mainboard EA656MB, keyboard/power board EA656KB
SoC	Samsung S3C2416XH-40 — ARM926EJ-S (ARMv5TEJ) at 400 MHz, 16 KB I / 16 KB D cache, MMU, 2-D accelerator, LCD controller, USB 2.0 device + host, NAND controller, ADC, RTC
RAM	Hynix H5MS2562NFR-E3M — 32 MB mobile DDR SDRAM (256 Mb, 4 M x 4 banks x 16)
Flash	Samsung K9F2G08U0C-SCB0 — 256 MB (2 Gb) SLC NAND; OS + user files
Battery	Li-ion 3.7 V 1500 mAh (5.55 Wh), WiseWod 484461AR, IEC 11CP6/51/63 — up to ~15 h
Firmware	HP OS on a bare-metal RTOS; newRPL has a G1 port that drives this hardware directly

HP PRIME G2

Model	2AP18AA (2018), later G8X92AA
SoC	NXP i.MX 6ULL MCIMX6Y2 — ARM Cortex-A7 (ARMv7-A) at 528 MHz, NEON, MMU; LCD (eLCDIF), USB OTG, NAND (GPMI), ADC, PWM, RTC on chip
RAM	256 MB DDR3 SDRAM
Flash	512 MB NAND
Battery	Li-ion 3.7 V ≈ 2000 mAh — ~24 h of continuous use
Firmware	FreeRTOS-based HP OS; U-Boot/Linux have been booted on it; about 3x the G1's speed

BOTH

Display	3.5" 320 x 240 TFT, 16-bit colour (114 ppi), LED backlight (PWM)
Touch	capacitive multi-touch panel (two fingers — mouse() reports both)
Keyboard	51 keys scanned as a GPIO matrix; codes 0–50 top-left to bottom-right; keyboard() returns the whole matrix as a bitmask
USB	micro-AB, USB 2.0. To a computer the Prime is a HID device (VID 03F0 HP) — not a disk; the Connectivity Kit, libhpcalcs and this sheet's tohp talk HID reports. The G2 is also a host: USBOpen / USBSend / USBReceive drive an attached device
Wireless	HP Prime Wireless Kit: a 2.4 GHz module on the calculator's USB port paired to a PC dongle (classroom use)
Other	real-time clock (Time, Date, TICKS ms counter), no beeper, no SD slot, reset pinhole above the battery door

Sources: TI-Planet's teardown of the DVT prototype (board and chip references), Wikipedia's HP Prime article (G2 SoC, memory), TI-Planet's G2 review, the newRPL Prime-G1 port in your tree (which drives the S3C2416 directly), the USB HID notes in retro/hp-prime. The G2's RAM/NAND part numbers and both touch controllers are not published in any teardown I could find.

Moving things: drag between the Connectivity Kit's calculator pane and content pane; calculator-to-calculator over the USB cable or the wireless kit via Memory Manager (Clone / Send); from the command line with libhpcalcs (send file = menu 5, full backup = menu 7; single-file receive fails on fw 2.2). The Prime is a USB HID device, never a disk. Memory Manager () Toolbox) shows what is using memory and holds on-calculator backups.