What's CuteHSP

CuteHSP is a very easy programming language.

The executable platform is Windows / macOS / Linux.

Instructions

Usage	Description
end	End the program.
run p1 -	Discard the currently executing program, read the file p1, and execute it.
goto p1	Jump to label p1.
gosub p1	Jump to label p1 of the subroutine.
return	Return from the subroutine.
repeat p1	Repeats the range enclosed by repeat-loop.
loop	End instruction of iterative processing.
continue	It returns to the repeat command while repeating processing.
break	Exit from iterative processing.
if p1 -	If condition p1 is satisfied, the subsequent instructions of that line are executed.
else -	When the if condition is not satisfied, else instruction and after are executed.
dim p1,p2 -	Create an array variable (integer) with variable name p1 and array number p2.
ddim p1,p2 -	Create an array variable (real number) with variable name p1 and array number p2.
sdim p1,p2,p3 -	Create a string type array variable with variable name p1, number of characters p2, number of array p3.
font p1,p2,p3 -	Specify the font. p1 is the TTF filename, p2 is the font size, and p3 is the smoothing specification (0 or 16).
mes p1	Display the character string p1 on the screen.
picload p1	Read the image file p1 and display it on the screen.
input p1,p2,p3 -	Get key input value. p1: Variable name to store the input value. p2: Maximum number of characters assigned to variable.

```
-                      p3: Line feed code recognition flag (0=none, 1=LF, 2=CR+LF).

beep p1,p2,p3,p4       p1 is the frequency, p2 is the playback length(ms), p3 is the
-                      waveform type, p4 is the volume (0 to 30000) and sounds.

                         [Type of waveform (Default:2)]

                              0:Sine wave     1:sawtooth wave     2:Square wave

                              3:Triangular wave     4:White noise

bload p1,p2            Read contents of filename p1 into character type variable p2.

bsave p1,p2            Save the contents of character type variable p2 as filename p1.

poke p1,p2,p3          Write byte value p3 to p2 byte of character type variable p1.

wait p1                Wait for p1 milliseconds.

stop                   Wait until the window is closed with [x] button.

title p1               Display character string p1 in title bar.

pset p1,p2             Draw a dot on the coordinates (p1,p2).

line p1,p2,p3,p4       Draw a line from coordinates (p1,p2) to coordinates (p3,p4).

boxf p1,p2,p3,p4       Fill rectangle from coordinates (p1,p2) to coordinates (p3,p4).

circle p1,p2,p3,p4     Draws a circle that fits within the rectangle from coordinates
-                      (p1,p2) to coordinates (p3,p4).

paint p1,p2            Fill the closed region containing the coordinates (p1,p2)
-                      with the current color.

redraw p1              Set the redraw switch to ON (1) or OFF (0).

pos p1,p2              Set the coordinates (p1,p2) to the current position.

color p1,p2,p3         Set the RGB color (p1,p2,p3) as the current color.

stick p1               Store key information in numeric variable p1.

                         [Key information]

                              1  Cursor left        2  Cursor up

                              4  Cursor right        8  Cursor down

                             16  Space              32  Enter

                             64  Ctrl              128  ESC

                            256  Left mouse button  512  Right mouse button

                           1024  TAB

------------------------------------------------------------------------------
```

Functions

Usage   Description

int(p1)   Return p1 as an integer value.

double(p1)   Return p1 as a real number (double precision floating point).

abs(p1)   Return p1 as absolute value.

str(p1)   Return p1 as a string.

rnd(p1)   Returns a random number from 0 to p1-1.

powf(p1,p2)   Returns the result of p1 raised to the power of p2.

peek(p1,p2)   Get the byte value of p2 byte of character type variable p1.

System variables

Name   Description

stat   Status (integer) after instruction or function execution stored.

refdval   The real type return value is stored in refdval.

refstr   String type return value is stored.

cnt   Counter value of repeat-loop.

strsize   The number of bytes of the file read by the bload is stored.

mousex   X coordinate of mouse cursor.

mousey   Y coordinate of mouse cursor.

mousel   1 if the left mouse button is pressed, 0 if not pressed.

mouser   1 if the right mouse button is pressed, 0 if not pressed.

Assignment to variable

```
a=12            ;Assign integer

b=12.3          ;Assign real number

c="message"     ;Assign a string
```

Arithmetic operators

```
a=10+4          Addition

b=12.8-6.3      Subtraction

c=9*4           Multiplication

d=8/2           Division

e=15\7          Remainder
```

Comparison operators

```
12>5        12 is greater than 5

12>=5       12 is greater than or equal to 5

12<5        12 is less than 5

12<=5       12 is less than or equal to 5

12=5        12 is equal to 5

12!5        12 is not equal to 5
```

Logical operators

```
12&5        AND

12|5        OR
```

Annotation description

If you write a ";" (semicolon), subsequent statements will not be executed.

pos 30,20   ;This part will not be executed.

Colon

Multiple instructions can be described on one line by using ":" (colon).

pos 50,100:mes "Coordinate specification and character display."

Label

You can use labels as jump destinations for goto or gosub instructions.

Label name is "*" (asterisk) followed by an arbitrary name with alphanumeric characters.

    gosub *la001

    stop

*la001

    mes "Sub Routine"

    return

Sample programs

A text file with the extension ".hs" is a sample program.

Program execution method

From the console (command line), make the following entries.

```
C:
cd \cutehsp_win
cutehsp kakiget.hs
```

Specify program file name as argument.

If you omit the program file name, "start.hs" is given as an argument.

Type of executable file

CuteHSP Minimum

CuteHSP standard version.

The window screen 640 x 480 dots, the displayable color is 16.77 million colors.

Executable file name "cutehsp" (* Windows version requires the DLL file "glfw3.dll")

Unusable instructions: input, picload, font, mes, beep

CuteHSP Extra

CuteHSP extended version.

The window screen 640 x 480 dots, the displayable color is 16.77 million colors.

Executable file name "cutehspx" (* Windows version requires the DLL file "glfw3.dll" and "OpenAL32.dll")

When displaying characters on the window screen, a font file such as "tiny.ttf" is also necessary.

Unusable instructions: input

CuteHSP Console

CuteHSP console version.

User I/O is standard input/output, without a window screen.

Executable file name "cutehspcl"

Unusable instructions: picload, font, beep, title, stop, stick, wait, color, pos, pset, line, boxf, circle, redraw

Recommended editor for programming

Like CuteHSP, we recommend the cross-platform editor "Geany" which runs on Windows / macOS / Linux.

Geany

https://www.geany.org/Download/Releases

For Linux, you can install from the following command.

        sudo apt-get install geany

Below is a method to execute while editing a program with Geany.

1. Launch Geany and select "Set Build Commands" from the menu "Build".

2. Set the external execution command to "Execute commands".

   For example, enter 'C:\cutehsp_win\cutehspx.exe "%f"'.

3. After that, you can execute the program being edited by "Execute" on the toolbar or F5 key.

CuteHSP Homepage

https://github.com/kikeroga3/tinyhsp

E-mail

cutehsp@outlook.jp