

■ソフトの概要

CuteHSPはWindows、macOS、Linux上で動作するクロスプラットフォームなトイプログラム言語です。あらゆるユーザがエントリーモデルとしてプログラミングを習得するために、あるいはホビーとして楽しめるようにと、最小限の命令とシンプルな言語仕様でつくられています。

■インストール・アンインストール方法

ファイルを解凍するとサブフォルダがWindows、macOS、Linux用に分かれていますので、ご自身の環境に合ったCuteHSPを任意のフォルダにコピーしてご利用ください。アンインストール方法は、コピーしたフォルダごと削除で問題ありません。

■実行ファイル

CuteHSPには利用目的に応じた3種類の実行ファイルがあります。

CuteHSPミニマム

実行ファイル名「cutehsp」（※Windows版はDLLファイル「glfw3.dll」も必要）

表示は640×480ドット固定のウィンドウ画面が1枚、表示可能色は1677万色で、CuteHSP標準版の位置付けです。

使用不可命令：input、picload、font、mes、beep

CuteHSPエクストラ

実行ファイル名「cutehspx」

（※Windows版はDLLファイル「glfw3.dll」「OpenAL32.dll」も必要）

ウィンドウ画面に文字表示する場合は「tiny.ttf」等のフォントファイルも必要です。使用可能なフォントファイルのサイズは2MBまでとなっています。ご注意ください。

表示は640×480ドット固定のウィンドウ画面が1枚、表示可能色は1677万色で、CuteHSP拡張版の位置付けです。ミニマムの上位互換となっています。

使用不可命令：input

CuteHSPコンソール

実行ファイル名「cutehspcl」

ウィンドウ画面をもたないコンソール版CuteHSPです。ユーザI/Oは標準入出力になります。

使用不可命令：picload、font、beep、title、stop、stick、wait、color、pos、pset、line、paint、boxf、circle、redraw

使用不可システム変数：mousex、mousey、mouse1、mouser

■プログラム実行方法

基本的にはコンソール(コマンドライン)から以下のように入力すると実行できます。

cutehsp start.hs ←第1引数にプログラムファイル名を指定する。

また「start.hs」というファイル名はデフォルト(既定)のプログラムファイル名となっており、実行ファイル「cutehsp」と同じフォルダにあれば、単に

cutehsp

と入力するだけでもプログラム「start.hs」が自動的に読み込まれ、実行されます。

※Windowsの場合、プログラムファイル(例：start.hs)を実行ファイル(例：cutehsp)にドラッグ&ドロップ、または拡張子「.hs」を「cutehsp」に関連付けしてプログラムファイルをダブルクリックしても実行できます。

■命令表

書式	説明
end	プログラムを終了する
run p1	実行中のプログラムを破棄してファイルp1を読み込み、実行する
goto p1	ラベルp1へジャンプする
gosub p1	サブルーチンのラベルp1へジャンプする
return	サブルーチンから戻る
repeat p1	repeat～loopで囲まれた範囲をp1回繰り返し処理する
loop	繰り返し処理の終端命令
continue	繰り返し処理中でrepeat命令まで戻る
break	繰り返し処理から抜ける
if p1	条件p1を満たしていればその行の以降の命令を実行する
else	if条件が満たされなかった場合にelse命令以降が実行される
dim p1, p2	変数名p1、配列数p2の配列変数(整数)を作成する
ddim p1, p2	変数名p1、配列数p2の配列変数(実数)を作成する
sdim p1, p2, p3	変数名p1、文字数p2、配列数p3の文字列型配列変数を作成する
font p1, p2, p3	フォントの設定を行います。p1は使用するTTFファイルを拡張子まで含めて指定。p2はフォントサイズ(上限100)、p3はスムージングの指定(0でスムージングなし、16でスムージングあり)です。
mes p1	文字列p1を画面に表示します
picload p1	p1で指定した画像ファイルを読み込み、画面に表示します
input p1, p2, p3	キー入力を取得する。p1:入力値を格納する変数名。 p2:変数に代入される最大文字数。p3:改行コード認識フラグ (0=なし/1の場合はLFを、2の場合はCR+LFを改行と認識)
beep p1, p2, p3, p4	p1は周波数、p2は再生する長さ(ミリ秒)、p3は波形の種類、p4は音量(0～30000)を指定して音を鳴らします(p3, p4は省略可) [波形の種類(デフォルト:2)]

- | | |
|-----------|---------|
| 0 正弦波 | 1 ノコギリ波 |
| 2 矩形波 | 3 三角波 |
| 4 ホワイトノイズ | |

bload p1, p2	ファイル名p1の内容を文字型変数p2に読み込む
bsave p1, p2	文字型変数p2の内容をファイル名p1で保存する
poke p1, p2, p3	文字型変数p1のp2バイトめにバイト値p3を書き込む
wait p1	p1ミリ秒待つ
stop	ウィンドウが[×]でクローズされるまで待つ
title p1	タイトルバーに文字列p1を表示する
pset p1, p2	座標p1, p2にドットを描画する。p1, p2が省略された場合はカレントポジションに描画する
line p1, p2, p3, p4	座標p1, p2から座標p3, p4まで線を描画する
paint p1, p2	座標p1, p2を含む閉領域を現在の色で塗りつぶす
boxf p1, p2, p3, p4	座標p1, p2から座標p3, p4まで矩形を塗りつぶす
circle p1, p2, p3, p4	座標p1, p2から座標p3, p4までの矩形内に収まる円形を描く
redraw p1	p1が0なら再描画スイッチをオフ、1ならオンにする。 p1が省略されたらオンにする
pos p1, p2	座標p1, p2をカレントポジションに設定する
color p1, p2, p3	RGBカラーp1, p2, p3をカレントカラーに設定する
stick p1	数値変数p1にキー情報を格納する

[キー情報]

- | | |
|--------------|--------------|
| 1 カーソルキー左(←) | 2 カーソルキー上(↑) |
| 4 カーソルキー右(→) | 8 カーソルキー下(↓) |
| 16 スペースキー | 32 Enterキー |
| 64 Ctrlキー | 128 ESCキー |
| 256 マウスの左ボタン | 512 マウスの右ボタン |
| 1024 TABキー | |

■関数表

書式	説明
int (p1)	p1を整数値として返す
double (p1)	p1を実数値(倍精度浮動小数点)として返す
abs (p1)	p1を絶対値として返す
str (p1)	p1を文字列として返す
rnd (p1)	0からp1-1までの乱数値を返す
powf (p1, p2)	p1のp2乗(べき乗)の結果を実数値で返す
peek (p1, p2)	文字型変数p1のp2バイトめのバイト値を取得して返す

■システム変数

変数名	説明
stat	命令や関数実行後のステータス(整数値)が格納される
refdval	実数型の戻り値はstatではなくrefdvalに格納される
refstr	文字列型の戻り値が格納される
cnt	repeat-loopのカウンタ値
strsize	bload命令で読み込んだファイルのバイト数が格納される
mousex	マウスのx座標
mousey	マウスのy座標
mousel	マウスの左ボタンが押されていれば1、押されていなければ0
mouser	マウスの右ボタンが押されていれば1、押されていなければ0

命令34個、関数7個、システム変数が9個と全部あわせてもわずか50個と、まさにタイニーなプログラム言語となっています。

■言語仕様

値

整数、倍精度浮動小数点、文字列が使用可能です。

a=12 ;整数を代入
b=12.3 ;倍精度浮動小数点数を代入
c="message" ;文字列を代入

変数

変数名には半角英数字を使用できますが、先頭は必ず英字にしてください。変数はプログラム実行開始時に16個の要素を持つ整数型の配列に初期化されています。例えば「dim a, 16」と宣言しなくても「a(13)=5」はエラーになりませんが、「a(19)=1」は「dim a, 20」と前もって宣言しておかないとエラーになります。

配列は一次元のみサポートしており、配列の要素へのアクセスは()を使います。()を使わずに変数名のみを書いた場合は要素0への参照となります。例えば「a=12」と「a(0)=12」は同じ意味の代入式になります。

明示的に変数の型を指定したい場合はdim、ddim、sdim命令で初期化します。

演算

下記の演算子が使用できます。

算術演算

a=10+4 加算

b=12.8-6.3 減算

c=9*4 乗算

d=8/2 除算

e=15¥7 剰余

比較演算

12>5 大なり

12>=5 大なりまたは等しい

12<5 小なり

12<=5 小なりまたは等しい

12=5 等号(等しい)

12!=5 不等号(等しくない)

論理演算

12&5 AND 論理積

12|5 OR 論理和

コメント

その行に書いた「;」(カンマ)以降はコメントとして認識されます。

pos 30, 20 ;ここはコメントなので実行されない

コロンの

「:」(コロンの)を利用することで複数の命令を1行で記述することができます。

pos 50, 100:mes “座標指定と文字表示”

ラベル

goto、gosub命令の飛び先としてラベルが使用可能です。ラベル名には「*」の後に半角英数字で任意の名前を付けることができます。

```
gosub *la001

stop

*la001

mes "Sub Routine"

return
```

■サンプルプログラム

拡張子「.hs」のテキストファイルがサンプルプログラムです。ファイル名に「cl_～.hs」と付けてあるのがCuteHSPコンソール用、「ex_～.hs」と付けてあるのがCuteHSPエクストラ用のプログラムです。それ以外はCuteHSPミニマムで実行することができます。一通りお試しくださいとCuteHSPでどんなことが出来るのか大体わかります。実行しながらプログラムソースのほうもご確認いただけると文法も理解しやすいかと思います。

■エラーメッセージ

プログラムに間違いがあると、実行時にタイトルバーにエラーメッセージが表示されます。

■プログラミングに便利なエディタ

テキストエディタなら何でも大体OKですが、ここではCuteHSP同様にWindows、macOS、Linux上で動作するクロスプラットフォームな軽量エディタ「Geany」を紹介しましょう。Windows版、macOS版インストーラは下記URLから入手できます

Geany

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

Linuxであれば下記コマンドからインストール可能です。

```
sudo apt-get install geany
```

以下、Geanyでプログラムを編集しながら実行する方法を紹介します。

1. Geanyを起動し、メニュー「ビルド」から「ビルドコマンド設定」を選択。
2. ダイアログ「ビルドコマンドを設定」の設定枠「コマンドを実行」の「Execute」ボタンのすぐ右の入力枠に外部実行コマンド(※)を設定。
※例えば「C:\¥cutehsp_win¥cutehsp.exe "%f"」と設定する。
3. 「Execute」ボタンをクリックし、「cutehsp」と入力し、ボタン名を変更。
4. 最後に「OK」で設定を保存します。

#	ラベル	コマンド	作業ディレクトリ	リセット
Haskell コマンド				
1.	Compile	ghc --make "%f"		🗑
2.				🗑
3.	Lint	hlint "%f"		🗑
正規表現エラー:				🗑
独立したコマンド				
1.	メイク(M)	make		🗑
2.	カスタム ターゲットをメイク(T)...	make		🗑
3.	オブジェクトをメイク(O)	make %e.o		🗑
4.				🗑
正規表現エラー:				🗑
注意: 項目 2 はダイアログを表示し、結果をコマンドに渡します				
コマンドを実行				
1.	cutehsp	utehsp_win¥cutehsp.exe "%f"		🗑
2.				🗑
%d, %e, %f, %o, %i はコマンドやディレクトリ項目に置き換えられます, 詳細はマニュアルを参照してください				
				キャンセル(C) OK(O)

5. 以後、メニューの「ビルド」配下に「cutehspx」が追加されているはずです。

このメニューを選択すれば編集中のプログラムが実行できます。

また、ツールバーの「実行」、F5キーでも同じように実行できます。



Geanyのバグ情報

Windows版のGeanyバージョン1.30(または1.30.1)では文字入力が半角カナになってしまう不具合があるようですので修正されるまではバージョン1.29の使用をおすすめします。

また、Geany以外にもプログラムを編集しながら実行することができるエディタは多々あります。WindowsならTeraPadやサクラエディタ、macOSならCotEditor、Linuxなら定番のgeditなどなど、色々として自分に使いやすいものを選ぶと良いでしょう。

■クロスプラットフォームなソースコード

CuteHSPのソースコード「cutehsp.c」は機種依存しないC言語のコードで書かれています。画面処理にはGLFW(OpenGL)、サウンド処理にはOpenALライブラリを使用しており、各環境用にライブラリを導入することで同じソースコードをWindows、macOS、Linux上でコンパイルすることが可能です。

■CuteHSP開発ホームページ

<https://github.com/kikeroga3/tinyhsp>

■作者への連絡先

cutehsp@outlook.jp

■取り扱い種別

フリーソフト

■再配布について

商用・非商用を問わず、CuteHSPの実行ファイルは皆さんが自分で作ったプログラムに同梱して再配布してもらってかまいません。

■The MIT License

以下に定める条件に従い、本ソフトウェアおよび関連文書のファイル（以下「ソフトウェア」）の複製を取得するすべての人に対し、ソフトウェアを無制限に扱うことを無償で許可します。これには、ソフトウェアの複製を使用、複写、変更、結合、掲載、頒布、サブライセンス、および/または販売する権利、およびソフトウェアを提供する相手に同じことを許可する権利も無制限に含まれます。

上記の著作権表示および本許諾表示を、ソフトウェアのすべての複製または重要な部分に記載するものとします。

ソフトウェアは「現状のまま」で、明示であるか暗黙であるかを問わず、何らの保証もなく提供されます。ここでいう保証とは、商品性、特定の目的への適合性、および権利非侵害についての保証も含みますが、それに限定されるものではありません。

作者または著作権者は、契約行為、不法行為、またはそれ以外であろうと、ソフトウェアに起因または関連し、あるいはソフトウェアの使用またはその他の扱いによって生じる一切の請求、損害、その他の義務について何らの責任も負わないものとします。