



簡易電子説明書

目次

ソフト本体に付属する電子説明書とほぼ同内容の記事が
ご覧いただけます。

はじめてのプチコン

- 8 プチコンについて
- 9 操作方法
- 10 BASICについて
- 11 TOP MENUについて
- 12 作品公開とダウンロード
- 13 プロジェクトとファイルの管理
- 14 オプション

プログラムを作る

- 15 BASICを起動する
- 16 キーボードの使い方
- 17 DIRECTモードで命令しよう
- 18 EDITモードでプログラム入力
- 19 EDITモードの編集機能
- 20 プログラムの保存と読み込み
- 21 サンプルプログラムについて
- 22 HELPの使い方
- 23 SMILEツールの使い方

BASIC入門

- 24 PRINTと変数
- 25 変数の使い方
- 26 入力と条件判断
- 27 コンピュータの色 (RGB)
- 28 グラフィック命令
- 29 サウンド命令

詳細情報

- 31 立体視
- 32 画面構成
- 33 BG
- 34 SPRITE

プチコン3号は、ニンテンドー3DS上で手軽にプログラムを作ることができるツールです。3Dモードにも対応していますから、立体視機能を活かしたプログラムを作れます。

利用規約

- 本製品により作成されたプログラムや画像等のリソースは、公開機能によって多くの人が見たり実行したりする可能性があります。他人が不快な気持ちになるようなもの、プライベートな情報、他人の権利（肖像権やプライバシー権、著作権など）を侵害するものなどは公開しないでください。
- 公衆に著しく迷惑をかける不良行為を行ったり、わけつけ、名誉棄損にあたる画像を公表したりした場合、法令や条例によって罰せられる恐れがあります。
- スマイルブームはお客様が公開した情報および公開したプログラム等によって生じたトラブルについては一切責任を持ちません。
- 公開作品を見て不快に感じたお客様からの通報により公開作品を作者への確認等を行わず無条件に削除する可能性があります。あらかじめご了承ください。

BASIC互換性上の注意点

「BASIC基本仕様」のページも併せてご参照ください。

- 本製品に搭載されているBASICは、これまでに発売されているBASICとは互換性がありません。また弊社プチコンおよびプチコンmkIIとの互換性もありませんので、プログラム移植の際には文法の違いにご注意ください。
- 内部で扱われている数値は、倍精度実数型、または整数型です。2進数計算による誤差が発生することがありますので、厳密な数値計算が必要な用途には適しません。
- 本製品では行番号を使った分岐は使用できません。分岐先の行に@で始まるラベル(名札)を設定しておき、行番号の代わりにラベルを指定して分岐します。
- 複雑な計算を行ったり、画面に多数の絵を表示するようなプログラムを書くと、実行速度が低下することがあります。
- SAVE・DELETE等のファイルへの書き込みが行われる命令を繰り返し実行すると、読み書きに時間がかかるようになります。
- 本製品のプログラム編集機能には、1行の長さの制限はありませんが、メモリーの上限に達した時点で入力できなくなります。
- 本製品では配列変数を使用する際、必ずDIM命令による宣言が必要です。省略するとエラーになります。
- 本製品での配列変数のかっこは、()ではなく[]を使用します。
()は演算の優先順位を指定する際と、関数の引数を指定する際に使用します。
- 代入命令のLETは、本製品では廃止いたしました。「変数名=式」の形式で、=記号のみをご使用ください。
- べき乗の演算記号^や↑は、本製品では使用できません。POW()関数をご使用ください。
- IF文によるの条件式は、他の言語にならい、==で「等しい」、!=で「異なる」となっています。従来のBASICには=で「等しい」、<>で「異なる」となっているものが多くありますので、特にご注意ください。なお代入は = です。
- 本製品のFOR～NEXT命令は、先に条件を判断します。FOR I=0 TO -1 のように、STEP1では成り立たない状態の場合、FOR内部を実行せずにスキップし、NEXT以降の命令に続きます。従来のBASICと違い、最低1回実行されることはありませんのでご注意ください。
- 本製品のNEXT命令には、「NEXT I」のような変数名を指定することはできません。指定してもエラーにはなりませんが、単なる「NEXT」と同じ働きになります。
- 本製品のON～GOTO、ON～GOSUBの制御変数は、1からではなく0からとなっています。
- 従来のBASICのINT()関数は、本製品ではFLOOR()関数となっています。
- 割り算を含む計算を行い、必要な数値が整数部分だけの場合は、FLOOR()関数を使って整数部分だけを利用してください。座標計算などで誤差が積み重なると、微妙なズレが生じてしまいます。
- 本製品のRND()関数は整数値を返します。実数値が必要なときは、RNDF()関数をご使用ください。
- 本製品のグラフィック命令では、座標指定に()や-を使用しません。
他製品の例：LINE(0.0)-(639,399)
本製品の例：GLINE 0,0,399,239

9 操作方法

本製品は開発ツールのため、ゲームとは操作方法が異なります。

十字ボタン	EDITモード時にカーソルを移動。 ・ Lボタンを押しながら上下でページ単位の移動、左右で行頭と行末への移動（スライドパッドの上下左右でも操作可能）
Aボタン Bボタン Xボタン Yボタン	ユーザーがプログラム実行時に利用できます。 EDITモード時、AボタンはENTERキー、YボタンはBSキーとして機能します。
Lボタン Rボタン	キーボードのSHIFTキーと同じ働きとなり、キートップに紫色で書かれている文字を入力するときに使用します。プログラム実行時はユーザーが利用できます。
STARTボタン	プログラムを実行します。 プログラム実行中はプログラムを中断する働きをします。

プチコン3号はコンピューターを動かすプログラムを作るツールです。

プログラムとは？

プログラムとは、コンピューターに対して動作の指示を出す命令を、順番に並べたものです。コンピューターはプログラムを忠実に実行します。もちろんゲームソフトもプログラムによって作られています。

BASICはプログラム言語

人間が使う言葉「言語」にもいろいろな種類があるように、プログラムにもいろいろな文法の言語があります。本製品で採用している「BASIC」は、人間の言葉（ただし英語）に近い単語を使うので、わかりやすいのが特長です。

たとえば…

画面に文字を書け！…PRINT命令

音を鳴らせ！…BEEP命令

数字を記憶しろ！…変数と代入命令

数字の大小を調べろ！…比較と分岐命令

動作を繰り返せ！…ループ命令

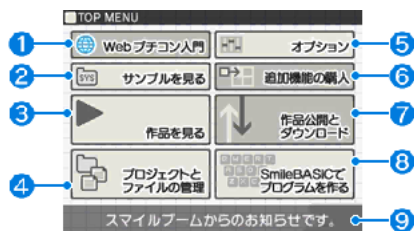
データを保存しろ！…SAVE命令

BASICにはたくさんの命令があります。使いこなすと、コントローラの操作に合わせてキャラクタを動かしたり、音を出したりというような動作がいろいろでき、楽しいゲームやツールを作ることができます。

もちろん、間違った命令を書けば、**バグ**と呼ばれる誤動作も生み出せます。

現在、ゲームやシステムを作っている会社にも、初めてのころにBASICを使ってプログラムを覚えた人がたくさんいます。プチコンを通してプログラムの楽しさを味わっていただければ、こんなにうれしいことはありません。

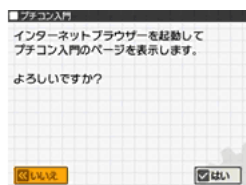
11 TOP MENUについて



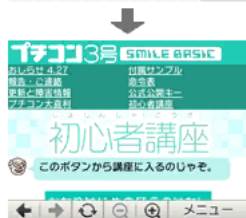
TOP MENU（トップメニュー）は、プチコン3号を起動すると最初に表示される画面です。

①webプチコン入門

プチコンやBASICを初めて触る人のために初心者用の説明が用意された「プチコン入門」の公式ページをブラウザで表示します。（インターネットに接続できる環境が必要です）



インターネットへ接続することを確認するダイアログが表示されます。



プチコンに戻る時はHOMEボタンを押してブラウザを終了させてください。

ページ内の情報は定期的に更新されるため、この表示とは異なる可能性があります

②サンプルを見る

サンプルとして用意されたプログラムやゲームを見るための機能です。見たいファイルを選んで「決定」ボタンを押すとSmileBASICが起動してプログラムが実行されます。

- EX1～8 少しずつ複雑になるプログラム
- GAME1～7 ジャンルの異なるゲーム
- SB???? 各種ツールのプログラム

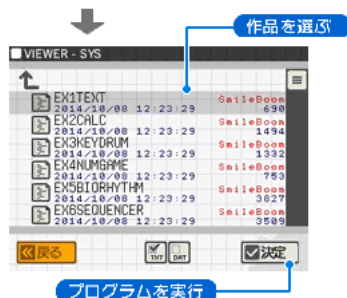
くわしい説明は「サンプルプログラムについて」のページをご覧ください。

③作品を見る

保存された作品を実行する機能です。プロジェクトを選んで「ファイル選択」ボタンを押し、実行したい作品を選んでから「決定」ボタンを押すとSmileBASICが起動してプログラムが実行されます。



SYSフォルダ内は、「サンプルを見る」で表示される内容と同じものです。



次のいずれかのとき、実行中のプログラムは停止して、トップメニューに戻ります。

- ・プログラムが終わりまで実行された
- ・プログラム内でエラーが発生した

- ・ STARTボタンが押された
- ・ END命令やSTOP命令が実行された

===Press ENTER to Exit=== と表示されて停止したときは、下画面のENTERキーを押してください。TOP MENUに戻ります。

④プロジェクトとファイルの管理

プログラムやデータなどのファイルは、標準状態では DEFAULT という名前のプロジェクトフォルダに保存されます。たくさん作品を作って、ファイルが増えすぎたときは、新しいプロジェクトフォルダを作って使い分けことができます。詳しくは「プロジェクトとファイルの管理」のページを参照してください。

⑤オプション

キーボードのSMILEボタンに割り当てる機能などを設定します。詳しくは「オプション」のページを参照してください。

⑥追加機能の購入

プチコン3号を強化するための機能やサービスをニンテンドー e ショップから購入することができます。ボタンを押すと商品一覧が表示されて、購入したい商品を選択後「購入」ボタンを押すと、ニンテンドー e ショップへ接続し購入の手続きが始まります。なお、購入の際には商品ごとに設定された価格分の残高が必要となります。（ニンテンドー e ショップへの接続にはインターネットに接続できる環境が必要となります。）

⑦作品公開とダウンロード

作ったプログラムをサーバーに保管したり、他の人に公開したりするときに使います。詳しくは「作品公開とダウンロード」のページを参照してください。

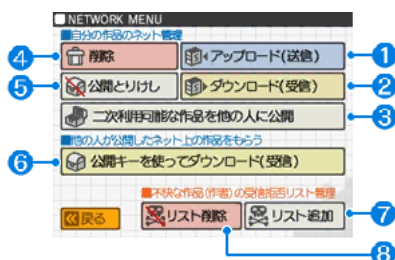
⑧SmileBASICでプログラムを作る

SmileBASICを起動してキミだけのプログラムを作ってみましょう。詳しくは「プログラムを作る」以降のページを参照してください。

⑨お知らせ表示エリア

プチコン3号に関する重要な情報やイベントの告知などスマイルブームからのお知らせが流れてきます。（お知らせを受け取りたくない場合は、オプションから「お知らせを受け取る」のチェックを外してください。

12 作品公開とダウンロード



ネットワークに接続してファイルをスマイルブームサーバーに保管したり完成した作品を公開する場合に利用するメニューです。

このメニューを実行するためにはインターネットに接続できる環境が必要です。また、アップロードやダウンロードを行うためには、ニンテンドーネットワークIDの登録が必要です。詳しくは本体の取扱説明書をご覧ください。

ニンテンドーネットワークIDの認証が必要な機能を実行する最初の1回だけ認証のためのダイアログが表示されます。

①アップロード(送信)

保存されているプロジェクト（またはファイル）を、スマイルブームサーバーにアップロードします。



- スマイルブームサーバーには、ニンテンドーネットワークIDに結び付いた作品の保管先が10個分用意されています。それぞれを「スロット」と呼びます。プロジェクトを選んでファイルをまとめて保存しても、ファイルを1つだけ選んで保存してもスロットは1つ使われます。
- アップロードするプロジェクトを十字ボタンで選択します。ファイル1つだけを選択する場合は、「ファイルを選択」ボタンを押してファイル一覧から選びます。

⚠ **SYSフォルダ、およびDEFAULTフォルダはアップロードできません。**

- 「決定」ボタンを押します。確認メッセージが表示されます。

- ・「はい」…アップロードが行われます。
- ・「いいえ」…アップロードは行いません。

大量データをアップロードすると、大変時間がかかります。完了のメッセージが表示されるまでお待ちください。

スマイルブームサーバーにアップロードしただけでは、まだファイルは公開されません。「二次利用可能な作品を他の人に公開」を行った時点で他の人に見える状態になります。

②ダウンロード(受信)

スマイルブームサーバー上に保存されているスロットごとのプロジェクト（またはファイル）を、ダウンロードします。



- 十字ボタンでダウンロードしたいスロットを選択し「決定」ボタンを押します。確認のメッセージが表示されます。

- ・「はい」…ダウンロードが行われます。
- ・「いいえ」…ダウンロードは行いません。

大量データをダウンロードすると、大変時間がかかります。完了のメッセージが表示されるまでお待ちください。

ダウンロード中は電源ボタンをOFFにしたり、SDカードやゲームカードを抜いたりせず、そのままお待ちください。

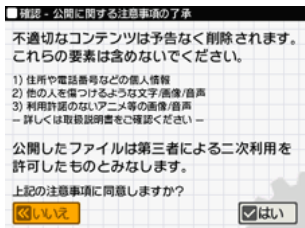
③二次利用可能な作品を他の人に公開

スマイルブームサーバーにアップロードされた作品を公開する機能です。公開することで「公開キー」が生成されて他の人に作品を渡すことができます。

公開までの流れは以下ようになります。

- 公開に関する注意事項了承（二次利用許可）

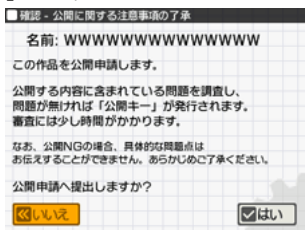
- アップロードされたファイルから選択
- 公開申請への提出意思の確認
- 審査対象を検査（サーバーにて行われる）
- 問題が無ければ公開キーを発行



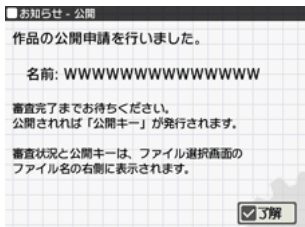
- 公開に関する警告画面が表示されます。この画面で「はい」ボタンを押すと、条件に同意したものとみなされます。
 作品を公開すると、公開キーを受け取った人が本製品を所有していれば、世界中の誰でも作品（プログラムやデータ等）を見ることができるようになります。軽はずみな気持ちで他の人の悪口や住所・電話番号などの個人情報を含んだ作品公開は絶対に行わないでください。悪意のある人物によってインターネット経由で不特定多数への拡散が行われると、場合によっては流出された人の人生を台無しにしてしまうことがあります。
 なお、公開した作品に含まれるプログラムとリソース（画像およびサウンド）は、第三者による2次利用を許可したものとみなされます。著作権侵害などの問題が発生しないように注意してください。公開後に削除通報が発生した作品はスマイルブームによって予告なく削除される可能性があります。



- 十字ボタンで公開したいファイルを選択し、Aボタンで決定します。あらためて、公開に関する注意事項が表示されます。公開申請を行う場合は「はい」をタッチします。



- ここから審査が始まります。スマイルブームサーバー側でファイル内容を分析し問題が無いことが確認でき次第、公開キーが発行されます。承認までには少し時間がかかります。



⚠ 審査の結果「公開NG」となる場合があります。公開申請が承認されない理由としては、公序良俗に反する要素が含まれている、削除要請のあったファイルが含まれている等の原因が考えられますが、詳細な原因についてはお答えすることができません。誠に申し訳ありませんが、あらかじめご了承ください。「公開NG」となった作品は他の人に渡すことができません。

- これで公開審査の手続きは終了です。あとはサーバーからの審査結果をお待ちください。

④削除



- 十字ボタンで削除するファイルのあるスロットを選択し、Aボタンで決定します。
- 削除するファイルを十字ボタンで選択し、Aボタンでチェックマークを付けます。
- 決定ボタンを押すと、削除してもよいか、確認のメッセージが表示されます。
 - 「はい」…削除が行われます。
 - 「いいえ」…削除は行いません。

⑤公開とりけし

スロットの公開を取り消し、他の人がアクセスできないようにします。



十字ボタンで公開を取り消すスロットを選択し、Aボタンで決定します。確認メッセージが表示されます。

- ・「はい」…公開が取り消されます。
- ・「いいえ」…公開状態は変更されません。

⑥公開キーでネット上の作品をダウンロード

スマイルブームサーバーに公開されているファイルを、ダウンロードして受け取ります。



1. サーバーにアクセスするために、公開キーを入力します。公開キーが正しくないとファイルをダウンロードできません。
2. ダウンロード先のプロジェクトフォルダを選択します。
3. 決定ボタンを押すと、ダウンロードしてもよいか、確認のメッセージが表示されます。
 - ・「はい」…ダウンロードが行われます。
 - ・「いいえ」…ダウンロードは行いません。

大量データをダウンロードすると、大変時間がかかります。完了のメッセージが表示されるまでお待ちください。

ダウンロード中は電源ボタンをOFFにしたり、SDカードやゲームカードを抜いたりせず、そのままお待ちください。

⑦受信拒否リストへの登録

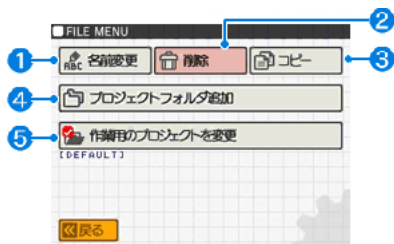
不適切な行為や問題のある行為をする人からの通信を拒否するため、受信拒否リストに登録することができます。

ファイルの一覧が表示されるので、受信拒否したい人が作成したファイルを選択し、決定してください。

⑧受信拒否リストへのクリア

受信拒否リストをクリアします。今まで登録されていた人からの通信を受けられるようになります。受信拒否リストは必ずすべてのリストをまとめて消去します。個別の削除には対応していません。

13 プロジェクトとファイルの管理



プログラムやデータを保存する場所である「プロジェクトフォルダ」を新たに作成したり、ファイルの名前変更・コピー・削除を行うためのメニューです。

① 名前の変更

プロジェクトフォルダやファイルの名前を変更します。

1. 名前を変更したいプロジェクトフォルダやファイルを選択します。



2. 簡易キーボードで新しい名前を入力します。
3. 名前が変更されます。

ファイルの名前はRENAME命令でも変更できます。

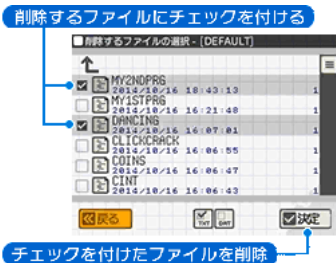
② 削除

不要になったファイルを削除します。

1. 削除するファイルが含まれているプロジェクトフォルダを選択します。



2. 削除するファイルを選択します。



3. 「ファイルを削除しますか？」という確認メッセージが表示されます。
 - ・「はい」…削除が行われます。
 - ・「いいえ」…削除は行われません。

⚠ 削除したファイルは復活できません。操作を間違えないようにご注意ください。

ファイルの削除はDELETE命令でも行えます。

③ コピー

プロジェクトフォルダやファイルをコピーし、プロジェクトフォルダには新しい名前をつけます。

◆プロジェクトフォルダのコピー

1. コピーしたいプロジェクトフォルダを1つ選択して決定します。
2. 簡易キーボードで新しい名前を入力します。
3. 新しい名前がついて、プロジェクトフォルダがコピーされます。

◆ファイルのコピー

1. コピーしたいファイルのあるプロジェクトフォルダを1つ選択します。
2. 「ファイルを選択」ボタンを押します。
3. コピーしたいファイルにチェックを付けて選択します。
4. コピー先にするプロジェクトフォルダを選択します。
5. ファイルがコピー先プロジェクトフォルダにコピーされます。

※同じ名前がある場合は確認画面が出ます

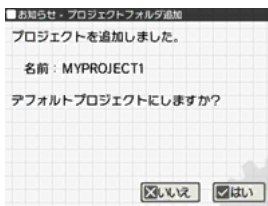
④プロジェクトフォルダを作る

新しいプロジェクトフォルダを作ります。

1. 簡易キーボードが表示されるので、新しいプロジェクトフォルダの名前を入力します。



2. プロジェクトフォルダが作成されます。
作ったプロジェクトフォルダをデフォルトプロジェクトにするかどうか（すぐに使うか）を選択します。



- ・「はい」…作ったプロジェクトフォルダをデフォルトにします。
- ・「いいえ」…今はデフォルトにしません。

デフォルトプロジェクトを変更すると、それまでのプロジェクトフォルダのプログラムやファイルは見えなくなります。デフォルトプロジェクトはTOP MENUの「オプション」→「デフォルトプロジェクトの割当て」や、PROJECT命令で変更できます。

⑤作業用プロジェクトを変更

BASICで扱うファイルを保存する、デフォルトのプロジェクトフォルダを変更します。

⚠ 初期状態では「DEFAULT」というプロジェクトが割り当てられています。

- ・今までのプロジェクトフォルダのファイルは、BASICから読み込むことができなくなりますが、消えてしまったわけではありません。
再びデフォルトプロジェクトの割当てで選択しなおせば、扱えるようになります。

デフォルトプロジェクトの変更は、BASICのPROJECT命令でも行えます。
「PROJECT '''」と入力すると、初期状態のプロジェクトに戻ります。

作業用プロジェクトの考え方

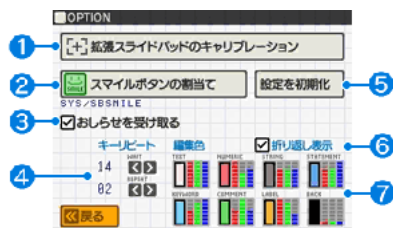
プチコン3号では、カレントプロジェクトとして以下の3つの状態が存在します。

1. 起動時のカレントプロジェクト(OPTIONで設定)
2. 非実行時のカレントプロジェクト(PROJECT命令)
3. 実行時のカレントプロジェクト(EXEC等の実行時)

それぞれ設定時には下位のプロジェクト設定も更新されます。

例えば、起動時のカレントプロジェクトを変更した場合、非実行時のカレントプロジェクトと実行時のカレントプロジェクトも同時に変更されます。

実行開始(RUN、ツール実行、VIEWERからのプログラム実行)時には、実行時カレントプロジェクトは非実行時のカレントプロジェクトが初期値として設定されます。



ブチコン3号の動作オプションを設定します。

① 拡張スライドパッドのキャリブレーション

拡張スライドパッドの調整を行います。拡張スライドパッドを取り付けてから、画面の指示に従って操作してください。

② SMILEボタンの割当て

キーボードのSMILEボタンに割り当てるプログラムを変更することができます。

◆プログラムの注意

SMILEボタンに登録するためには、プログラムの先頭に「OPTION TOOL」命令を書く必要があります。

③ おしらせを受け取る

スマイルブームからイベント等のおしらせを受け取りたいときにチェックを付けます。受け取ったメッセージはトップメニューの下部に表示されません。

④ キーリピート

BASICで文字を入力するときに、同じキーを押し続けると繰り返し入力されます。この繰り返しの早さを設定することができます。

- ・「WAIT」…1文字目が入力されてから、2文字目が入力されるまでの時間
- ・「REPEAT」…2文字目以降が連続入力される間隔

⑤ 設定を初期化

OPTION画面の設定を初期状態に戻します。

⑥ 折り返し表示

プログラムのEDITモードでは、1行が長いと次の行に折り返して表示されますが、折り返さずに1行に表示したいときにチェックを付けます。この場合、横スクロールが発生します。

⑦ 編集色

EDITモードではプログラムを見やすくするため、命令や数値のように種類の違う要素を色分けしています。それぞれの表示色を変更することができます。

- ・「TEXT」…文字
- ・「NUMERIC」…数字
- ・「STRING」…" "で囲まれた文字列
- ・「STATEMENT」…命令
- ・「COMMENT」…コメント
- ・「LABEL」…@で始まるラベル
- ・「BACK」…背景

15 BASICを起動する

トップメニューから「SmileBASICでプログラムを作る」を選択すると、SmileBASICが起動します。
SmileBASICには、次の2つの動作モードがあります。

2つの動作モード

●DIRECT（ダイレクト）モード



キーボードから命令を入力し、実行するモードです（EDITモードからDIRECTボタンで切り替え）

最初はこのモードです。



●EDIT（エディット）モード



プログラムを入力するためのテキスト編集モードです（DIRECTモードからEDITボタンで切り替え）



プログラムはEDITモードで入力し、DIRECTモードの命令で保存・実行します。

モードの切り替え

モードの切り替えは、キーボードの下のシステムボタンで行います。

◆システムボタン



1	EDITボタン DIRECTモードからEDITモードに切り替えます。
2	DIRECTボタン EDITモードからDIRECTモードに切り替えます。
3	TOP MENUボタン SmileBASICを終了し、TOP MENUに戻ります。プログラムは消えません。
4	スマイルボタン 便利なスマイルツールを呼び出します。

◆プログラムSLOTについて



5	プログラムSLOT切り替え 本製品にはプログラムを記憶する場所が4つあり、それをプログラムSLOTといいます。通常使われるのはSLOT0です。SLOT1～3は補助用のため、必ずしも使う必要はありません。
---	---

・その他のシステムボタンについては、「EDITモードの編集機能」のページを参照してください。

DIRECTモード

DIRECTモードは、命令をその場で実行するためのモードです。EDITモードで作ったプログラムも、DIRECTモードでRUN命令を実行することにより動作します。

◆上画面…コンソール画面

キーボードで入力した命令や、実行結果が表示されます。



◆下画面…キーボード

コンソール画面に命令を入力します。



- ・キーボード操作については、「キーボードの使い方」のページを参照してください。

DIRECTモードでは、命令を1つ1つ、入力するたびに実行します。複数の命令をまとめて実行したいときは、プログラムにする必要があります。プログラムはEDITモードで入力します。

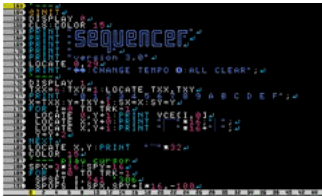
EDITモード

EDITモードは、プログラムを入力するための専用のモードです。

◆上画面…編集画面

入力されているプログラムが表示されます。

改行（**ENTER**キー）で自動的に行番号が増加します。1行が長い場合は自動的に折り返します。



◆下画面…キーボード

編集画面にプログラムを入力します。



- ・キーボード操作については、「キーボードの使い方」のページを参照してください。

16 キーボードの使い方

DIRECTモードでもEDITモードでも、文字は下画面に表示されているキーボードから入力します。

入力する文字の切り替え

英数字・記号・カナの切り替えや、大文字・小文字の切り替えが可能です。



1 文字切替キー

英数字 記号 カナ

2 SHIFTキー

次に入力する1文字の、大文字・小文字を切り替えます。
一部の記号も切り替わります。

3 CAPキー

SHIFTキーをロックして、切り替えた大文字・小文字の状態でも連続的に入力できるようにします。
もう一度押すと元に戻ります。

記号・カナの入力について

文字切替キーで記号やカナの入力に切り替えると、キーボードの一部のキーが変わり、さらに多くの種類の文字を入力できます。

記号のとき



カナのとき



1 記号のジャンルを切り替えます。全部で7ジャンルあります。

2 ひらがな・カタカナを切り替えます。

スペースと改行

文字の間を空けたいときはスペースキーを、改行したいときはENTERキーを押します。



1 スペースキー

1文字分のスペース（空白文字）を入力します。

BASICでは、命令の区切りにスペースを入力しなくてはならないことがあります。よく確かめて入力してください。

×BEEP5 ENTER
○BEEP 5 ENTER
↑

スペースキーを押して空ける

2 ENTERキー

ENTERキーは大切な働きをします。

- DIRECTモードのとき
画面に入力した命令を、コンピューター内部に送り込んで実行します。
ENTERキーを押さなければ、命令を入力しても実行されません。
- EDITモードのとき
改行し、カーソルを次の行の先頭に移動します。

この電子説明書では、命令の終わりにENTERキーを押すことを、記号  で表記しています。

例)

```
PRINT "HELLO" 
```

↑

↑

スペースキーを押す ENTERキーを押す

間違いやすい文字

ゼロとオー

BASICでは、数字のゼロと英字のオーは厳密に区別されます。間違えて入力すると正しく動作しません。
キーの場所も近いので注意してください。



1・I (アイ)・l(小文字のエル)

形が似ていても、違う文字は違う意味として区別されます。記号のマイナスとカナの長音（ー）なども、混用はできませんので注意してください。

引用符と区切り記号

	ダブルコーテーション 頻繁に使用されます。ダブルコーテーションで挟んだ部分は、数字ではなく文字列として扱われます。 135 →数値 135 (ひゃくさんじゅうご) "135"→文字列 (いち さん ご)
	セミコロン 命令とパラメーターの区切りなどに使います。 <pre>PRINT "きんがくは";A</pre> ↑
	コロン 複数の命令を並べるときなどに使います。 <pre>BEEP 5:GOSUB @DM:PRINT "BYE"</pre> ↑ ↑ セミコロンとコロンが並ぶこともあるので注意してください。 <pre>PRINT "○";:GOTO @TOP</pre> ↑↑
	カンマ 引数やデータの区切りなどに使います。 <pre>GLINE 0,0,399,239</pre> ↑ ↑ ↑
	ピリオド 小数点を示すときなどに使います。 カンマと混在して現れることがあるので注意してください。 <pre>DATA 3.14, 1.08, 36.5</pre> ↑ ↑ ↑

文字の挿入と上書き

文字を重ねて入力したときの動作には、**挿入モード**と**上書きモード**の2つがあります。
INSキーを押すと、それらを切り替えることができます。



	挿入モード 文字を入力すると、カーソルの位置に割り込みます。カーソルより右側にあった文字列はそろって右にずれます。
	上書きモード 文字を入力すると、カーソルの位置にあった文字は消され、新しい文字で置き換えられます。

文字の削除

入力した文字を削除するには、BSキー やDELキー を使います。



1	BSキー カーソルのひとつ左の文字を削除します。右側にあった文字列は、一緒に左にずれます。
2	DELキー カーソルのひとつ右の文字（上書きモードのときはカーソルと重なっている文字）を削除します。右側にあった文字列は、そろって左にずれます。

DIRECTモードでは、BSキーやDELキーで削除した文字や、上書きされた文字を、UNDOキーで復活させることはできません。
 EDITモードではUNDOキーが有効です。

ファンクションキー

キーボードの一番上には5つのファンクションキーがあり、よく使う命令をワンタッチで入力することができます。



ファンクションキーの内容は、KEY命令によって変更可能です。

例)ファンクションキー3番を「FILES」に変更

DIRECTモードにて、
 KEY 3,"FILES"

プログラム読み書き支援機能

キーボードが表示されているときにLボタンを押すと、ファンクションキー部分にプログラムファイルの読み書き支援用ボタンが表示されます。LOADボタンを押すとファイル一覧ダイアログが開き、選択したファイルをLOADします。SAVEボタンでは保存用のファイルを選択しSAVEします。現在読み込まれているファイル名の確認も可能です。

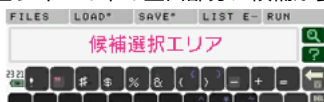


命令入力支援機能

命令を入力する際、最初の数文字を入力するだけで、該当する命令を一覧表示してくれます。

1. 候補選択エリア

ファンクションキーの下に空白部分に候補が表示されます。



2. 最初の1文字を入力

たとえばGのキーを押すと、候補選択エリアにGで始まる命令が並びます。



候補をタッチすると入力されます。

17 DIRECTモードで命令しよう

DIRECTモードは、コンピュータに命令を直接与えるモードです。

命令待ちのOK記号

コンソール画面には「OK」の文字が表示され、その下の行にカーソルが点滅しています。

```
OK
|
```

この状態を「命令待ち」といいます。命令を入力するときは、OKの表示を確認してください。

BASICの命令を入力しよう

簡単な命令をいくつか試してみましょう。

◆音を出す「BEEP（ビーブ）命令」

カーソルの位置から、次のように入力してください。面白い音が出ます。

```
BEEP 5 ENTER
```

- ・BEEPのあとには、SPACEキーで空白を入れてください。
- ・最後にENTERキーを押してください。
- ・音が出ないときは、サウンドボリュームを調整してみてください。

今度はBEEPの後ろの数字を変えて入力してみてください。

```
BEEP 7 ENTER
```

先ほどとは違う音が出ました。

このように、命令と一緒に指定する情報のことを、「**引数**」（ひきすう）といいます。引数の有無や意味は、命令によって違います。

◆画面をクリアする「CLS命令」

次のように入力してください。

```
CLS ENTER
```

画面の文字がすべて消え、命令待ちの「OK」だけになりました。CLS命令は画面をクリアする命令です。

◆文字の色を変える「COLOR命令」

コンソール画面の基本の文字色は白ですが、別の色で表示することもできます。

次の命令を入力してください。

```
COLOR 3 ENTER
```

- ・COLORの英字のオーと間違えて、数字のゼロを入力しないように注意してください。

実行結果は次のようになります。

```
COLOR 3
OK      ←実行結果
|
```

白だったOK表示が、赤になりました。引数3は「赤色」を意味する数値です。

「COLOR 15」と入力し、ENTERキーを押すと、元の白に戻ります。

```
COLOR 15
OK
|
```

色番号には0～15があります。ただし0は透明、1は黒です。指定すると文字が見えなくなりますから注意してください。

エラーについて

コンピュータは命令を厳密に解釈しますから、1文字でも間違いがあると実行しません。

試しに、でたらめな命令「ABC」を入力し、ENTERキーを押してください。

```
ABC
Undefined function  ←エラーが出た
OK
|
```

「ピッ」と警告音が鳴り、「Undefined function」と表示されました。

これは「そのような機能はありません」という意味のエラーメッセージです。

エラーメッセージの内容については、「エラーメッセージ表」のページを参照してください。

プログラムとは？

DIRECTモードでは、命令を1つ入力するたびに実行されますから、そのつど次の命令を入力しなくてはなりません。

しかし、ゲームのように複雑な動きをさせるには、最初に多くの命令を全部まとめて入力しておく必要があります。

それを実現するのが「プログラム」です。プログラムとは、複数の命令を手順通りに並べたものです。次のページでプログラムにチャレンジしてみましょう。

18 EDITモードでプログラム入力

簡単な命令を組み合わせてプログラムを作ってみましょう。

プログラムを入力するため、EDITモードに切り替えてください。



簡単なプログラムの入力

ここでは次のような手順のプログラムを入力してみます。全部で3行です。

1. 画面をクリアする …CLS命令
2. 「HELLO」と表示する …PRINT命令
3. 音を出す …BEEP命令

◆1行目を入力し、ENTERキー

1行目には画面をクリアする命令、CLS命令を入力します。



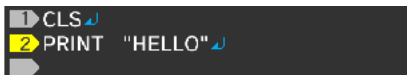
先頭の黄色いマークは、現在入力中の行という意味です。行番号が自動的に付きます。

行末の  マークは改行マークといい、行の終わりを示します。

1行を入力したら、ENTERキーを押して改行します。

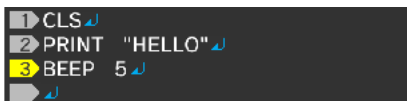
◆続きを入力

2行目には、画面に「HELLO」と表示する命令を入力します。文字を表示する命令は、PRINT命令です。



PRINT命令には、引数として「表示する文字列」を指定します。BASICでは、文字列はダブルコーテーション(")で囲むルールがあります。忘れずに入力してください。

同様に3行目も入力して完成です。



プログラムの実行…RUN命令

DIRECTモードに切り替え、入力したプログラムを実行しましょう。



プログラムを実行する命令は、「RUN（ラン）」命令です。



プログラムが正しく入力されていれば、画面がクリアされ、HELLO」と表示されて音が鳴ります。



思い通りの結果にならなかったときは、EDITモードに戻って間違いを調べます。

◆プログラムの強制停止はSTARTボタン

プログラムは先頭の行から順に実行され、最後の行で終了します。しかしプログラムによっては無限に実行されるものがあります。

このようなときは、3DS本体のSTARTボタンを押してください。プログラムが中断します。

プログラムの消去…NEW命令

入力したプログラムを一度に消去するには、DIRECTモードでNEW（ニュー）命令を実行します。

プログラムがすべて消去されます。



⚠ NEWを実行すると、すべてのプログラムSLOTが消去されます。十分注意してください。

◆特定のプログラムSLOTだけ消去する

NEW命令には引数として「消去するプログラムSLOT」があります。SLOT0～3に対応し、0～3の数字で指定します。

NEW 0 →SLOT0のみ消去

NEW 1 →SLOT1のみ消去

NEW 2 →SLOT2のみ消去

NEW 3 →SLOT3のみ消去

NEW →すべて消去

EDITモードには、プログラムを作りやすいよう、いろいろな編集機能が用意されています。

行を空ける・分ける・つなぐ

プログラムの途中で1行を空けたり、行を分けたりつないだりする方法です。

◆空の行を挿入する

1. 挿入したい行の次の行の先頭にカーソルを合わせます。

挿入したい位置 → 1 CLS ↵
2 BEEP 5 ↵
↑
カーソル

2. ENTERキーを押します。

行が挿入された → 1 CLS ↵
2 ↵
3 BEEP 5 ↵

◆長い行を2行に分ける

1. 分けたい位置にカーソルを合わせます。

分けたい位置
1 CLS PRINT ↵
2 BEEP 5 ↵

2. ENTERキーを押します。

1 CLS ↵
2 PRINT ↵
3 BEEP 5 ↵

◆2行を1行につなぐ

1. 上の行の改行マークの1つ前にカーソルを合わせます。

カーソル
2 PRINT ↵
3 "HELLO" ↵

2. DELキーを押します。

2 PRINT "HELLO" ↵

複数行の編集・アンドウ

EDITモードのシステムボタンには、範囲を指定したコピーや貼り付けなど、プログラムの編集に適した機能が用意されています。



1	SELボタン … 範囲選択 プログラム中でコピーや削除をしたい範囲を選択します。 1. 範囲の先頭の文字にカーソルを合わせます。 2. SELボタンを押します。 3. 範囲の末尾の文字までカーソルを移動します。選択されている範囲は白い背景色で表されます。
2	COPYボタン … 選択範囲のコピー 選択した範囲の行を内部に取り込みます。 ※画面には変化がありません。
3	PASTEボタン … コピー内容の貼り付け COPYボタンで内部に取り込んだ行を、カーソル位置から貼り付けます。
4	L.INSボタン … 1行挿入 カーソルの次に1行を挿入し、カーソルをその行に移動します。
5	UNDO(アンドウ)ボタン … 元に戻す 削除やPASTEなどで失敗したときに、1つ前の状態に戻します。

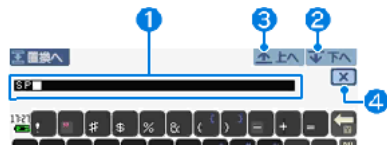
長いプログラム内の移動

- ・Lボタンを押しながら十字ボタンの上下を押すと、画面単位で上下にスクロールします。
- ・Lボタンを押しながらスライドパッドを上下に動かすと、プログラムの先頭行・最終行にジャンプします。

プログラム内の検索・置換

◆検索機能

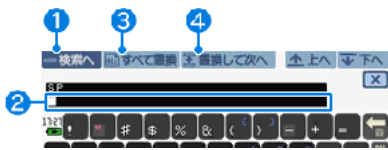
EDITモードでは、プログラムの中から指定した単語を検索することができます。
キーボード右上の検索ボタンを押すと、キーボードが検索モードに変わります。



1	検索キーワード入力欄 検索したい単語を入力します。
2	下へボタン（次を検索） カーソル位置から下方向に検索し、見つかった単語を白の背景で表します。
3	上へボタン（前を検索） カーソル位置から上方向に検索し、見つかった単語を白の背景で表します。
4	検索終了ボタン 検索モードを終了します。

◆置換機能

検索機能で左上の「置換へ」ボタンを押すと、置換モードに変わります。検索された単語を別な単語に置き換えることができます。



1	検索へボタン ボタンを押すと検索モードに戻ります。
2	置換キーワード入力欄 置換する単語を入力します。
3	すべて置換ボタン 発見した単語をすべて置換します。
4	置換して次へボタン カーソル位置から下方向に検索し、最初に見つかった単語を置換します。もう一度押すと、次の単語を置換します。

20 プログラムの保存と読み込み

作ったプログラムはSDカードに保存することができます。

SAVE命令…プログラムの保存

プログラムを保存するには、DIRECTモードで「SAVE（セーブ）」命令を使います。

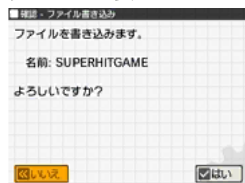
書式 SAVE "ファイル名"

- ・ファイル名に使える文字は、英数字と_（アンダースコア）

1. 例として、プログラムに「TEST1」とファイル名を付けて保存してみます。

```
SAVE "TEST1" ENTER
```

2. 下画面に確認メッセージが表示されます。



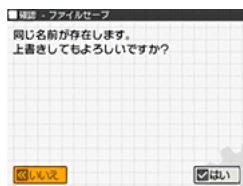
- ・「はい」…保存が始まります。
- ・「いいえ」…保存を行いません。

保存が終了し、完了メッセージが表示されるまで、電源をOFFにしたりSDカードを抜いたりせず、そのままお待ちください。

3. 保存が終了すると、完了メッセージが表示されます。「了解」を押してください。

◆同じ名前のファイルがすでにあったら

同じ名前のファイルがすでにSDカードに存在するときは、確認メッセージが表示されます。



- ・「はい」…ファイルを上書き保存します。
⚠ 上書きすると、古いファイルの内容は消えます。元に戻せませんのでご注意ください。
- ・「いいえ」…保存を行いません。

◆応用：他のプログラムSLOTを保存する

SAVE命令で保存されるのは、プログラムSLOT0の内容です。もしSLOT1のプログラムを保存したいときは、SLOT番号を指定するため、ファイル名の前に「PRG1:」と入力してください。

```
SAVE "PRG1:TEST1" ENTER
```

同様に、「PRG2:」や「PRG3:」で、SLOT2やSLOT3を保存可能です。

◆補助：プログラムのSAVE支援機能

現在編集中のプログラムをSAVE（保存）する場合は、下画面にキーボードが表示されている状態でLボタンを押すと表示されるSAVEボタンを押して保存するファイルを選択する方法もあります。



LOAD命令…プログラムの読み込み

保存したプログラムは、LOAD（ロード）命令により、読み込んで実行することができます。

⚠ プログラムを読み込むと、現在プログラムSLOT0に入力されているプログラムは上書きされて消えます。

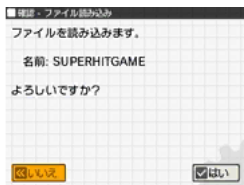
書式 LOAD "ファイル名"

- ・保存時に付けたファイル名

1. 例として、ファイル「TEST1」を読み込んでみます。

```
LOAD "TEST1" ENTER
```

2. 下画面に確認メッセージが表示されます。



- ・「はい」…読み込みが始まります。
- ・「いいえ」…読み込みを行いません。

読み込みが終了し、完了メッセージが表示されるまで、電源をOFFにしたりSDカードを抜いたりせず、そのままお待ちください。

3. 読み込みが終了すると、完了メッセージが表示されます。「了解」を押してください。

◆応用：他のプログラムSLOTに読み込む

LOAD命令で読み込むと、プログラムはSLOT0に読み込まれます。しかし、ファイル名の先頭に「PRG1:」と入力すると、SLOT1に読み込むことができます。

```
LOAD "PRG1:TEST1" ENTER
```

同様に、「PRG2:」でSLOT2に、「PRG3:」でSLOT3に読み込みます。

◆補助：プログラムのLOAD支援機能

現在編集中のプログラムSLOTに新しいプログラムをLOAD（読み込む）する場合は、下画面にキーボードが表示されている状態でLボタンを押すと表示されるLOADボタンを押して読み込むファイルを選択する方法もあります。

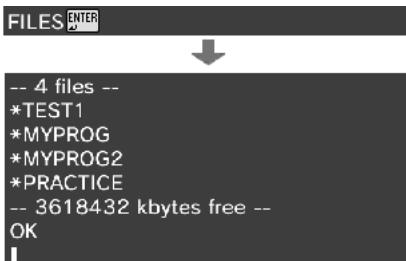


FILES命令…ファイルの一覧を表示

FILES（ファイルズ）命令を使うと、SDカードに保存したファイルを一覧表示することができます。

書式 FILES

- ・SDメモリーカードに保存されているファイルの一覧をコンソール画面に表示



DELETE命令…ファイルの削除

SDカードに保存したファイルは、DELETE（デリート）命令で削除することができます。

⚠ 削除したファイルを復活することはできません。ファイル名を間違えないよう、よく確かめておいてください。

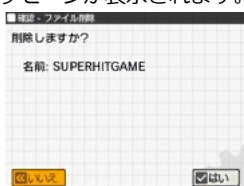
書式 DELETE "ファイル名"

- ・保存時に付けたファイル名

1. 例としてファイル「TEST1」を削除します。

```
DELETE "TEST1" ENTER
```

2. 下画面に確認メッセージが表示されます。



- ・「はい」…削除が実行されます。
- ・「いいえ」…削除を行いません。

削除が終了し、完了メッセージが表示されるまで、電源をOFFにしたりSDカードを抜いたりせず、そのままお待ちください。

3. 削除が終了すると、完了メッセージが表示されます。「了解」を押してください。

21 サンプルプログラムについて

本製品には、サンプルプログラムが含まれています。詳細については各サンプルのプログラム内容をご覧ください。

サンプルプログラムの遊び方

サンプルプログラムを実行する場合は、トップメニューの「サンプルを見る」ボタンを押して、ファイル一覧から選んで実行します。



サンプルプログラムの使い方

サンプルプログラムの使い方は、自分で作ったプログラムの場合と同じです。すべてのサンプルプログラムは、SYSプロジェクトの中に格納されています。

1. DIRECTモードに切り替える
2. LOAD "SYS/ファイル名"
3. RUN
4. 停止する場合はSTARTボタンを押す
5. EDITモードでプログラムを確認

すべてのサンプルプログラムは、SAVE命令を使って別の名前で保存することができます。スマイルブームに確認することなく、自由に改造していただけます。

基本サンプル

BASICの基本的な命令を利用した学習用のサンプルプログラムです。EDITモードでプログラムを表示させて何をしているか確認してみましょう。

ファイル名と簡易説明	
EX1TEXT	コンソールへの文字表示
EX2CALC	文字入力を使った簡易計算機
EX3KEYDRUM	キーボードを使った簡易鍵盤とドラム
EX4NUMGAME	数当てゲーム
EX5BIORHYTHM	バイオリズムのグラフィック表示
EX6SEQUENCER	タッチ操作を使った簡易シーケンサー
EX7ALIEN	複数の敵を動かして弾を発射するデモ
EX8TECDEMO	命令ごとの説明と技術評価用デモ

高度なサンプル

さまざまなBASIC命令を駆使したサンプルです。各種ツールサンプルは、ゲームを作るときに必要なデータを作るときに利用できます。これらのサンプルもEDITモードでプログラムを確認できるので、BASICプログラムが理解できるようになったら、自分だけの機能を追加することも可能です。

ファイル名と簡易説明	
GAME1DOTRC	敵の攻撃をよけながら画面上のドットをすべて消す絵文字で作ったレースゲーム
GAME2RPG	

敵を倒しながら3D風ダンジョンを進むロールプレイングゲーム
GAME3JUMP 敵や障害物をよけながらゴールを目指す横スクロールジャンプアクションゲーム
GAME4SHOOTER ステージが進むと強化される敵を倒してスコアを稼ぐ弾幕シューティングゲーム
GAME5VS 大きなキャラクターが剣で戦うシンプルな対戦型格闘ゲーム
GAME6TALK 適当な質問に答えると適当な答えが返ってくるナンセンスな占い？ゲーム
GAME7EXPAD 拡張スライドパッド専用。スティックを使ったお手玉ゲーム

本製品には、命令の書式や使い方をその場で確認できる、HELP機能が搭載されています。
DIRECTモードでもEDITモードでも使用できます。

HELP機能の使い方

例として、LOCATE命令の説明を表示します。

1. 調べたい命令を入力します。

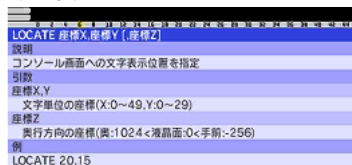
ENTERキーは押さないでください。

LOCATE[←カーソルの左側の命令が、調べる対象になります。

2. キーボードのHELPボタンを押します。

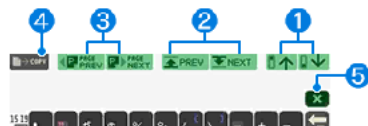


3. 上画面に説明が表示されます。



説明文が長いときは一部のみ表示されます。下記の ボタンを押し、上下スクロールして全文を確認してください。なお、引数の違いなど1つの命令に複数の説明が存在する場合があります。先頭の見出し部分に (1/2 →) のような表記がある場合は、スティックを左右に動かすことでページを変更することができます。

◆説明の操作方法



1	説明を上下に1行ずつスクロールします。 スライドパッドを上下に動かしてもスクロールできます。
2	説明を上下に1項目ずつスクロールします。 説明→引数→戻り→使用例
3	説明のページを切り替えます。 スライドパッドを左右に動かしても切り替えられます。
4	使用例をコピーします。 コピーした使用例は、PASTEボタンを押して、カーソル位置に貼り付けて利用できます。
5	HELP機能を終了します。

◆説明の読み方 ※例：LOCATE命令

1. タイトル部分：書式

LOCATE 座標X,座標Y [,座標Z]

- ・命令と引数の並べ方を示しています。
- ・[]で囲まれた引数は省略可能です。

2. 説明

説明
コンソール画面への文字表示位置を指定

- ・命令の働きの説明です。

3. 引数

引数
座標X,Y
文字単位の座標(X:0~49,Y:0~29)
座標Z
奥行方向の座標(奥:1024<液晶面:0<手前:-256)

- ・引数の働きの説明です。
- ・複数の引数がある場合があります。

4. 戻り値

戻り
なし

- ・ 戻り値を持つ命令（関数）の場合に、その説明が表示されます。

5. 使用例

例
LOCATE 20,15

- ・ 具体的な使用例です。
- ・ COPYボタンでコピーすることができます。

HELPの補助機能

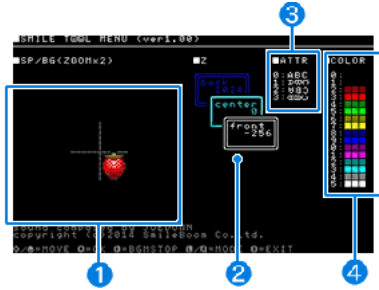
調べたい単語が無い状態でヘルプボタンを押すとヘルプの操作方法や、命令の一覧等を表示させることができます。

キーボードのSMILEボタンを押すと、SMILEツールが起動します。

SMILEツールでは、プログラムに必要な情報を確認できます。マップエディタなどの高度なツールも起動できます。

SMILEツールの画面

◆上画面



- | | |
|---|---|
| 1 | あらかじめ登録されている、SPRITEのキャラ定義番号と、その画像を確認することができます。スライドパッドを上下に動かして他のキャラを確認します。 |
| 2 | 表示要素のZ座標についての説明です。一番手前が-256、一番奥が1024です。 |
| 3 | ATTR命令で文字を回転させる際の、値と回転方向が記されています。 |
| 4 | COLOR命令で指定する文字色/背景色の番号が記されています。 |

◆下画面



- | | |
|---|---|
| 1 | 一覧表示エリア
②～⑥で選択した項目の一覧を表示。
十字ボタンで選択します。 |
| 2 | 効果音一覧に切り替え
BEEP用のプリセット音を確認。
Aボタンで再生。 |
| 3 | BGM一覧に切り替え
BGMPLAY用のプリセット音楽を確認。
Aボタンで再生。Yボタンで停止。 |
| 4 | MML用楽器音色一覧に切り替え
BGMPLAY命令のMMLで使用する楽器番号を確認。GM音源相当。
Aボタンで再生。Yボタンで停止。 |
| 5 | SPRITE定義画像一覧に切り替え
現在定義されているSPDEFの定義番号と内容を確認。 |
| 6 | BG画像一覧に切り替え
現在BGに定義されている画像を確認。 |
| 7 | 高度な編集ツール
・ PAINT (キャラ作成)
・ MAP (BGスクリーン作成)
・ ANIM. (アニメーション作成)
・ WAVE (サンプリングと波形編集)
詳しくは、次項を参照してください。 |
| 8 | SMILEツールの終了 |

	SMILEツールを終了します。 ※Xボタンを押しても終了できます。
9	電卓入力 数値を入力しENTERを押すと現在選択中のリストが指定番号に変化します。

高度な編集ツール

SMILEツールから起動できる高度な編集ツールです。いずれもYボタンでファイルモードに入り、Xボタンで終了します。

PAINT（キャラ作成）



BGやSPRITEのキャラデータを作成するツールです。色見本や編集ツールを選択して、EDITエリアに描きこみます。Yボタンを押してファイルモードに入り、Lを入力してENTERを押すと読み込みを行います。Sを入力してENTERを押すと保存します。保存したデータはプログラムから利用可能です。
(例)LOAD"GRP5:MYBG"

MAP（BGスクリーン作成）



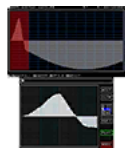
背景用BGキャラを並べて町並みなどのBGスクリーンデータを作成するツールです。キャラ見本からキャラを選んでEDITエリアに貼り付けます。保存したデータは配列に読み込むことでプログラムから利用可能です。

ANIM.（アニメーション作成）



SPDEFの定義内容調整、およびSPANIM用のアニメーションデータを登録するツールです。アニメのテスト再生ではプログラムSLOT1を上書きします。

WAVE（サンプリングと波形編集）



マイクからのサンプリングや、自分で書いた波形を楽器音として利用するためのツールです。SAVEしたファイルは配列としてLOADすることでWAVSET命令でBEEPやBGMPLOYの楽器音として利用が可能です。

プログラムを作るのに必要な、基本的な命令について紹介します。プログラムが初めての方はお読みください。

文字を表示する…PRINT命令

「PRINT（プリント）命令」は、画面に文字を表示する重要な命令です。次のプログラムを入力してください。

```
1 PRINT "HELLO"
2 PRINT "A"
```

入力したら、DIRECTモードで実行します。

RUN

```
HELLO
A
OK
|
```

PRINT命令は、「ダブルコーテーションで囲まれた文字列を、そのまま表示します。

◆セミicolon「;」とカンマ「,」の働き

1行目の "HELLO" の右に、; (セミicolon) を入力して実行してみてください。

```
1 PRINT "HELLO";
2 PRINT "A"
```

```
HELLOA
OK
|
```

「HELLO」と「A」がくっきました。

通常、PRINT命令は文字を表示すると、自動的に改行します。しかしセミicolon「;」を入れると改行が入らず、次の文字が密着します。

今度はセミicolon「;」を、カンマ「,」に変更してみてください。

```
1 PRINT "HELLO",
2 PRINT "A"
```

```
HELLO  A
OK
|
```

カンマ「,」の場合は、次の文字が一定の間隔だけ開いて表示されます。

"A"とAの違い…変数の知識

PRINT "A"と書くべきところ、ダブルコーテーション(")を忘れたらどうなるのでしょうか。

```
PRINT A
```

←"で囲まなかった

```
RUN
0
OK
|
```

←ゼロが表示された

0が表示されました。これは、ダブルコーテーションのないAは、「変数A」という意味になるためです。

◆変数とは

コンピューターの中には、数字や文字を記憶するメモリーがたくさん並んでいます。



BASICではメモリーに値を記憶する際、名前を付けて管理します。メモリーに名前を付けたものを「変数」といいます。



ダブルコーテーションのない「PRINT A」は、文字Aを表示するのではなく、「変数A」の中身を表示する」という意味になります。たまたま変数Aの中身が0だったため、0が表示されました。

```
PRINT "A"  ← Aという文字
PRINT A    ← 変数A
```

変数については、次のページでもっと詳しくご紹介します。

好きな位置に文字を出す…LOCATE命令

PRINT命令で文字を表示する位置（座標）は、LOCATE（ローケイト）命令で指定することができます。

書式 LOCATE X座標, Y座標

- ・X座標は右方向の文字数（0～49）
- ・Y座標は下方向の文字数（0～29）

次のプログラムは、X=10, Y=3の位置にHELLOと表示します。



◆文字の奥行きも指定できる

表示位置には奥行（Z座標）も指定できます。3Dモードのときは立体視が可能です。

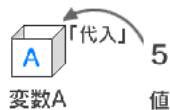
書式 LOCATE X座標, Y座標, Z座標

- ・Z座標は奥行きの指定（プラスで画面奥、ゼロで3Dスクリーン面と同じ、マイナスで画面手前）
- ・画面奥は0～1024、画面手前は0～-256で指定

変数を使用して、数値の記憶や計算を行う方法を紹介します。

変数に値を入れる =記号

変数に中身(値)を入れることを「代入」といいます。



この図では、変数Aに数値5を代入しています。
これをBASICで書くには、次のようにします。

A=5

これは、Aと5が等しいという意味ではなく、「Aに5を入れろ」という意味の命令になります。まだAという変数がなければ、空いているメモリーのどこかにAという名前が付けられて、新たに作られます。

本当に変数Aに5が入ったかどうかをPRINT命令で確認してみましょう。

```
1▶ A=5 ↵ ←変数Aに5を代入
2▶ PRINT A ↵ ←変数Aの内容を表示
↓ 実行結果
5
OK
|
```

◆数値変数に文字列は代入できない

ここでご紹介している変数は、正確には「数値変数」といい、文字列は代入することができません。A="HELLO"のような使い方をすると、エラーになります。

文字列を代入するには、あとで紹介する「文字列変数」を使います。

◆計算式も書ける

=の右には数値ひとつだけではなく、計算式も書くことができます。ただし、掛け算の記号はアスタリスク「*」、割り算の記号はスラッシュ「/」を使います。

2つの変数A、Bを用意し、それぞれに2+3、3÷2の計算結果を入れてみましょう。

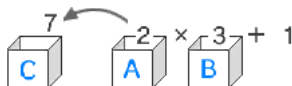
```
1▶ A=2+3 ↵
2▶ B=3/2 ↵
3▶ PRINT A, B ↵
↓ 実行結果
5      1.5
OK
|
```

◆変数同士の計算

数値同士の計算だけではなく、変数同士の計算も行えます。

```
1▶ A=2 ↵
2▶ B=3 ↵
3▶ C=A*B+1 ↵ ←A×B+1の値をCに代入
4▶ PRINT A, B, C ↵ ←A、B、Cの値を表示
```

3行目にて、AとBの内容を掛け、それに1を足してCに代入しています。



実行結果は次のようになります。

```
2      3      7
OK
|
```

◆円の面積を出してみよう

半径2の円の面積を計算するプログラムを書いてみましょう。

円の面積は、半径×半径×円周率です。円周率は、ここでは3.14としましょう。

```
1▶ R=2 ↵
2▶ PI=3.14 ↵
3▶ S=R*R*PI ↵
4▶ PRINT "メンセキハ";S ↵
```

2行目で、変数PIに小数3.14を代入しています。変数名は1文字でなくても構いません。英字で始まり、英数字とアンダースコア(_)からなる任意の長さの名前が付けられます。

3行目で面積を計算し、変数Sに代入しています。

4行目のPRINT命令に注意してください。まず「メンセキハ」と表示し、セミicolon「;」でぴったり付けて、計算結果のSを表示しています。

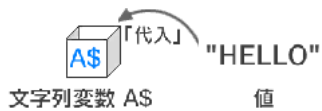
実行結果は次のようになります。

```
メンセキハ12.56
OK
|
```

文字列と文字列変数

文字列を代入するには「文字列変数」というものを使います。

文字列変数は、変数名の末尾に\$記号を付けて表します。



これをBASICで書くと、次のようになります。

```
A$="HELLO"
```

PRINT命令では、数値変数と同様、文字変数の値も表示することができます。次のプログラムは、文字変数NAME\$に「ALICE」を代入してから、PRINT命令で表示しています。

```
1 NAME$="ALICE"
2 PRINT "HELLO ";NAME$
```

↓ 実行結果

```
HELLO ALICE
OK
|
```

◆文字変数の足し算

文字変数同士は、足し算によって結合することができます。

```
1 A$="HELLO "
2 B$="WORLD"
3 C$=A$+B$ ←A$とB$の足し算
4 PRINT C$
```

↓ 実行結果

```
HELLO WORLD
OK
|
```

26 入力と条件判断

キーボードから文字を入力したり、条件に応じて自動的に処理が変わったりするプログラムを考えてみましょう。

文字を入力する…INPUT

前ページの円の面積を求めるプログラムは、半径の変数の値がプログラムの中で固定されていました。これではプログラムを書き換えない限り、他の半径の面積を計算できません。

```
1 R=2 ←半径が2に固定されている
2 PI=3.14
3 S=R*R*PI
4 PRINT "メンセキハ";S
```

そこで、1行目のRを内部に固定せず、外部(キーボード)から好きな数字を入力できるようにすれば便利です。そのための命令がINPUT (インプット)です。

```
1 INPUT R ←Rをキーボードから入力
2 PI=3.14
3 S=R*R*PI
4 PRINT "メンセキハ";S
```

これを実行すると、次のようになります。

```
RUN
? | ←INPUT命令が入力を待っている
```

INPUT命令は、画面に「?」マークを表示し、キーボードから数値が入力されるのを待ちます。入力されたら、その値を変数(ここではR)に代入します。

半径3の円の面積を出すために、3と入力してENTERキーを押してみます。

```
RUN
? 3 ←3を入力した
メンセキハ28.26
OK
|
```

◆ガイドメッセージを表示して入力

実行するといきなり「?」だけが表示されるのでは、知らない人には何をすればいいのかわかりません。INPUT命令にはガイドメッセージを表示する機能があります。

次のプログラムは、「ハンケイハ?」と画面に表示してから、変数Rの入力待ちになります。

```
1 INPUT "ハンケイハ";R
2 PI=3.14
3 S=R*R*PI
4 PRINT "メンセキハ";S
```

↓ 実行結果

```
RUN
ハンケイハ? 3
メンセキハ28.26
OK
|
```

◆INPUT命令 まとめ

書式	INPUT "ガイドメッセージ"; 変数 ・ ガイドメッセージは省略可 (省略のときはセミコロンも不要) ・ 変数には文字変数も使用可能
使用例	INPUT "ネンレイハ"; AGE キーボードから数値が入力されるのを待ち、変数AGEに代入

指定場所にジャンプする…GOTOとラベル

先ほどのプログラムは、一度円の面積を表示したら終了してしまいました。

繰り返し計算を行うには、最後の行に達したら、再びプログラムの先頭に戻ってくれたほうが便利です。そのための命令が「GOTO (ゴートゥー)」です。

書式	GOTO ジャンプ先ラベル名 ・ ラベル名は、@から始まる任意の英数字(例: @TOP)
使用例	GOTO @TOP 「@TOP」というラベルのついた行に強制ジャンプする

プログラムは先頭の行から順に命令を実行しますが、GOTOを使うと指定した行に強制ジャンプ可能です。
ジャンプ先の行には、あらかじめ「ラベル」という名前を付けておきます。先頭に@（アットマーク）を付けると、その行はGOTOのラベルとして扱われます。

```
1 @TOP ↵ ←ラベルを用意しておく
2 INPUT "ハンケイハ";R ↵
3 PI=3.14 ↵
4 S=R*R*PI ↵
5 PRINT "メンセキハ";S ↵
6 GOTO @TOP ↵ ←@TOPにジャンプ
```

1行目にジャンプ先ラベル「@TOP」を用意しています。この行は単なる目印であり、これ自体では何もしません。
2～5行目で半径を入力し、結果を表示したら、6行目のGOTO命令で、ラベル「@TOP」にジャンプします。これにより、プログラムは先頭から繰り返します。

↓ 実行結果（実行中）

```
RUN
ハンケイハ? 3 ←1回目の入力
メンセキハ28.26
ハンケイハ? 2 ←2回目の入力
メンセキハ12.56
ハンケイハ? | ←3回目の入力待ち
```

◆強制停止はSTARTボタン

このプログラムは、常にGOTO命令で先頭に戻ってしまうため、強制的に止めるにはSTARTボタンを押してください。

次に、入力に応じて自動的にプログラムが止まるようにしてみましょう。そのためには条件判断を使います。

条件判断…IF～THEN

BASICでは変数の値を調べて、条件が合うときに限り命令を実行させることができます。そのときに使うのがIF(イフ) ～THEN (ゼン) 命令です。

書式	IF 条件式 THEN 実行する命令 ・条件式は、A==0、A>4などの比較 == 等しい != 等しくない > より大きい < より小さい >= 以上 <= 以下 ・THEN以下は、条件を満たしたときに実行する命令
使用例	IF A\$=="YES" THEN PRINT "アタリ" もし変数A\$が「YES」なら、「アタリ」と表示してから次の行に進む。そうでなければ何もせず次の行に進む。

前のプログラムを改造し、「ハンケイハ?」に対してゼロが入力されたら終了するようにしてみます。
半径の入力後に、条件判断命令を挿入します。

```
1 @TOP ↵
2 INPUT "ハンケイハ";R ↵
3 IF R==0 THEN END ↵ ←条件判断
4 PI=3.14 ↵
5 S=R*R*PI ↵
6 PRINT "メンセキハ";S ↵
7 GOTO @TOP ↵
```

「IF R==0 THEN END」は、「もしRが0と等しければ、END命令（プログラム終了）を実行しなさい」という意味になります。
比較の場合、「R=0」ではなく「R==0」になる点に注意してください。従来のBASICとも異なります。

↓ 実行結果

```
RUN
ハンケイハ? 3 ←1回目は3を入力
メンセキハ28.26
ハンケイハ? 0 ←2回目は0を入力
OK ←終了した
|
```

プログラムの終了…END命令

上のプログラムで使いましたが、プログラムを途中で終了させる命令として、「END命令」があります。
プログラムは最後の行に達すると終了するので、条件分岐のないプログラムでは必要ないことがあります。

書式	END プログラムを終了する。
----	--------------------

補足：IF～THEN ～ELSE～ENDIF

●プログラムに詳しい方へ

本製品ではIF～THEN～ELSEも使用できます。
またENDIFを使用することにより、条件成立時・不成立時に、複数行にわたる命令も記述できます。

書式

IF 条件式 THEN
 成立時命令
 成立時命令…
[ELSE
 不成立時命令
 不成立時命令…]
ENDIF

27 コンピュータの色(RGB)

プログラムの中で使われる色（RGB）は、本などの印刷物に使われている色（CMY）とは違います。コンピュータの中で使われている色について簡単に説明します。

色の三原色（印刷用の色）



印刷に使われる色は、絵の具の色と同じで色を混ぜると黒く濁ってしまいます。上の図だと3つの色がすべて混ざった部分はB（BLACK=黒）になっています。完全に混ざると真っ黒になります。印刷に使うプリンターのインクカートリッジもこの色要素です。（プリンターの場合色を混ぜるだけでは完全な黒にはできないので黒は別に持っていることが多い）

光の三原色（液晶TVなどの色）



ゲームの画面は液晶に表示されていますが、液晶は光を発光することで色を表現しています。印刷用の色とは違って、光の三原色であるRED（レッド：赤成分）とGREEN（グリーン：緑成分）とBLUE（ブルー：青成分）を混ぜた場合、光が強くなるのでWHITE（ホワイト：白）に近づきます。赤緑青の英語の頭文字をとってRGB（アールジービー）と呼び、プログラムの中ではRGBを使って色を指定します。

よく使われる色のRGB値

色見本	R	G	B	色の名前
■ ■ ■	0	0	0	黒
■ ■ ■	255	255	255	白
■ ■ ■	224	224	224	薄いグレー
■ ■ ■	128	128	128	グレー
■ ■ ■	64	64	64	濃いグレー
■ ■ ■	255	0	0	赤
■ ■ ■	255	96	208	ピンク
■ ■ ■	160	32	255	紫
■ ■ ■	80	208	255	水色
■ ■ ■	0	32	255	青
■ ■ ■	96	255	128	黄緑
■ ■ ■	0	192	0	緑
■ ■ ■	255	224	32	黄色
■ ■ ■	255	160	16	オレンジ
■ ■ ■	160	128	96	茶色
■ ■ ■	255	208	160	薄紅色

たとえば、画面に赤い線を引くときは・・

```
GLINE 0,0,399,239,RGB( 255,0,0 )
```

と、書きます。最後の方にRGB()という命令がありますが、この命令の1つ目が赤成分Rの値(0～255)、次が緑成分Gの値、最後が青成分Bの値が入ります。緑の線だとRGB(0,255,0)、青い線だとRGB(0,0,255)と書きます。

画面にグラフィックを表示するための、基本的な命令を紹介します。

グラフィック画面と座標

本製品でのグラフィックは、グラフィック画面に描画します。グラフィック画面とはコンソール画面の裏側に重なっている存在です。

グラフィック画面は横400ドット×縦240ドットの解像度を持ち、横方向をX座標、縦方向をY座標として扱います。



グラフィック命令を試す

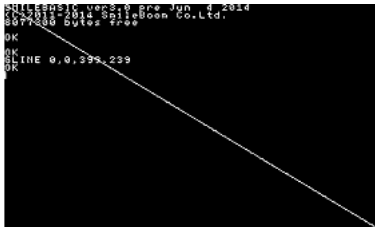
まずは簡単に、DIRECTモードでグラフィック命令を入力し、画面に直線を引いてみましょう。もちろん、EDITモードでプログラムにしても構いません。

◆直線を引く「GLINE命令」

次のように入力してください。

```
GLINE 0, 0, 399, 239 ENTER
```

GLINE命令は、グラフィック画面に直線を引く命令です。実行結果は次のようになります。



画面を斜めに横切る線が表示されました。

書式	GLINE 始点X, 始点Y, 終点X, 終点Y, 色 ・ 始点は直線の引き始めの座標、終点は直線の引き終わりの座標 ・ 色(色コード)は省略可能
使用例	GLINE 0, 0, 399, 239 始点(0, 0)から、終点(399,239)まで直線を引く

◆グラフィック画面を消去する「GCLS命令」

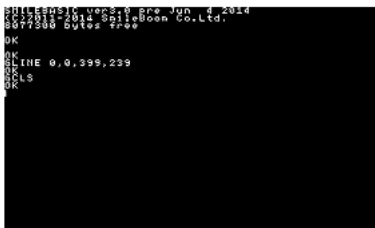
グラフィック画面に描いたものは、CLS命令では消去できません。グラフィック画面を消去するには「GCLS命令」を使います。

書式	GCLS グラフィック画面を消去する
----	-----------------------

※本製品では一般に、Gで始まる命令はグラフィック画面に対する命令です。

```
GCLS ENTER
```

↓ 実行結果



文字は消えずに線だけが消えました。

コンソール画面とグラフィック画面は別々に管理されており、CLS命令ではコンソール画面のみ、GCLS命令ではグラフィック画面のみが消去されます。

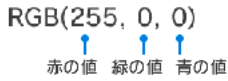
画面の構造についてもっと詳しく知りたい方は、「画面構成」のページを参照してください。

色について

GLINE命令をはじめ、グラフィック命令では描画する色を指定できるものがあります。色の指定方法には各種ありますが、グラフィックソフトを使ったことのある方なら、RGB指定がわかりやすいでしょう。

◆RGBによる色指定方法

光の3原色である赤(R)・緑(G)・青(B)を、それぞれ0～255の数値で表す方法です。
たとえば明るい赤は、次のように表記します。



次の色指定は、初期のコンピュータの基本8色に近いものです。これを基本に数字をいろいろ調整して、もっと自然な色を作れます。

黒色	■	RGB(0,0,0)
赤色	■	RGB(255,0,0)
緑色	■	RGB(0,255,0)
青色	■	RGB(0,0,255)
黄色	■	RGB(255,255,0)
水色	■	RGB(0,255,255)
紫色	■	RGB(255,0,255)
白色		RGB(255,255,255)

◆GCOLOR命令

GCOLOR命令は、これから描画するグラフィックの色を指定します。

書式	GCOLOR 色コード ・色コードはRGB形式
使用例	GCOLOR RGB(255,0,0) 今後描画するグラフィックの色を赤にする

そのほかのグラフィック命令

四角形や円など、基本的な図形を描く命令の一部を紹介します。ほかの命令については、サンプルプログラムや命令リストのページを参照してください。

◆GPSET命令…点を打つ

書式	GPSET 座標X, 座標Y ,色コード ・座標X,座標Y…点を打つ座標 ・色コードは省略可能
使用例	GPSET 199,119

◆GBOX命令…四角形を描く

書式	GBOX 始点X, 始点Y, 終点X, 終点Y, 色 ・始点X,Y…四角形の左上の座標 ・終点X,Y…四角形の右下の座標 ・色コードは省略可能
使用例	GBOX 0,0,100,80

◆GFILL命令…四角形を描いて塗りつぶす

書式	GFILL 始点X, 始点Y, 終点X, 終点Y, 色 ・始点X,Y…四角形の左上の座標 ・終点X,Y…四角形の右下の座標 ・色コードは省略可能
使用例	GFILL 110,40,50,20

◆GCIRCLE命令…円を描画する

書式	GCIRCLE 中心X, 中心Y, 半径, 色 ・中心X,Y…円の中心の座標 ・半径…円の半径 ・色コードは省略可能
使用例	GCIRCLE 110,40,50,20

◆GPAINT命令…図形内部を塗りつぶす

書式	GPAINT 開始X, 開始Y, 塗り色, 境界色 ・開始X,Y…塗りつぶし開始座標（塗りたい図形の内部を指定） ・塗り色コード…塗りつぶす色（省略可能）
----	---

	・ 境界色コード…塗りつぶしの境界線とする色（省略可能）
使用例	GPAINT 110, 40, RGB(0,255,255) 座標(110,40)から水色で、下地以外の色に当たるまで塗りつぶし

効果音やBGMを鳴らすための基本的な命令を紹介します。応用的な命令がこのほかにもたくさんありますので、それらの使い方についてはサンプルプログラムを参照してください。

効果音を鳴らす…BEEP命令

DIRECTモードのところで出てきたBEEP命令は、短い効果音を鳴らす命令です。プリセットで多数登録されている効果音を、番号を指定して鳴らします。

書式	BEEP 効果音番号 ・効果音番号は0～133 ・効果音番号は省略可(0と同じ)
使用例	BEEP 8

応用的な使い方として、周波数、音量、パンポットの調整が可能です。実際に命令を入力して、HELP機能で確認してください。

効果音番号とその内容は、SMILEボタンで表示される「BEEP」のリストで確認可能です。

BGMを鳴らす…BGMPLAY命令①

ゲーム中などに使うBGMは、BGMPLAY命令で簡単に鳴らすことができます。すぐに使えるプリセットBGMが43曲用意されています。

書式	BGMPLAY トラック番号, BGM番号 ・トラック番号：同時に複数曲を演奏する際のトラックを指定（0～7）省略可能 ・BGM番号：演奏するプリセット曲の番号を指定（0～42）※128～255はユーザー定義曲
使用例	BGMPLAY 12

プリセット曲には、1回の演奏が終わると停止するものと、繰り返し演奏されるものがあります。この場合、次のBGMSTOP命令を実行するまでは止まりません。

BGM番号とその内容は、SMILEボタンで表示される「BGM」のリストで確認可能です。

BGMを止める…BGMSTOP命令

再生中のBGMを停止する命令です。

書式	BGMSTOP ・全トラックの演奏をすぐに停止
----	----------------------------

トラックを指定すると、フェードアウトさせて止めることもできます。

書式	BGMSTOP トラック番号, フェードアウト時間 ・トラック番号：演奏を止めるトラック（0～7） ・フェードアウト時間：音量を徐々に下げ、停止するまでの秒数 0または省略で即時停止
使用例	BGMSTOP 0,2

次のプログラムを入力して、実行してみてください。

```
1 BGMPLAY 12 ↵
2 WAIT 60*5 ↵
3 BGMSTOP 0, 3 ↵
```

1行目でBGM12番の演奏が始まります（トラック番号を省略しているので、トラック0で演奏されています）。

2行目はWAIT命令といい、指定した時間（1/60秒単位）だけ待つ働きをします。WAIT 60で1秒待ちます。ここでは5秒待つことになります。

3行目で、トラック0で演奏中のBGMを、3秒かけてフェードアウトして止めます。

結果、BGMが演奏されてから5秒後にフェードアウトが始まり、その3秒後にBGMが完全に止まります。

音階を鳴らす…BGMPLAY命令

昔のBASICにあった、MML（Music Macro Language）を使用した音楽演奏も可能です。

次の命令を実行してください。

```
BGMPLAY "CDEFG2AB<C" ENTER
```

ダブルコーテーション"に挟まれた文字列を、楽譜とみなして演奏する方法です。個々の英字は音名を示します。

◆BGMPLAY命令（MML演奏）

書式	BGMPLAY "MML" ・ MML：楽譜を簡略化した演奏用の文字列（下記参照）
使用例	BGMPLAY ":0CCC :1REE :2RRG" チャンネルを3つ使い、和音を演奏

おもなMML要素

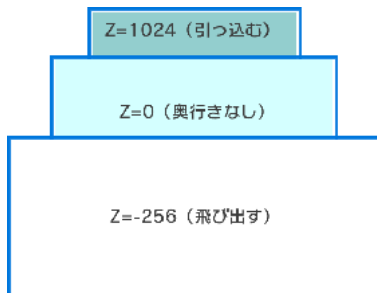
A～G…音名（音を鳴らす） C ド D レ E ミ F ファ G ソ A ラ B シ ※階名の一例 #または+…半音上げる C# D# E# F# G# A# G# C+ D+ E+ F+ G+ A+ G+ -(マイナス)…半音下げる C- D- E- F- G- A- B- R…休符
音名のあとの数字…音長 C1 C2 C4 C8 C16 C32 それぞれ全音符、2分音符、4分音符、8分音符、16分音符、32分音符 音長のあとの.(ピリオド)…付点 C1. C2. C4. C8. C16. C32. 音長がそれぞれの半分伸びる L数値 デフォルト音長の指定
O数値 オクターブ指定(O0～O8) < 1オクターブ上げる > 1オクターブ下げる
:数値 チャンネル指定(:0～:15) T数値 テンポ指定(T1～T240) V数値 音量変更(V0～V127) @数値 音色の変更(@0～@127)

MMLと文字を入力してからHELPボタンを押すことでMMLについての詳しい説明を見ることができます

プチコンではグラフィックや、文字、スプライトなどに奥行き値を設定することで、左右の目に見える視差を利用し映像を飛び出す（引っ込める）ことができます。立体視は表現力が上がりますが、あまり激しい変化を行うと目が疲れてしまうため、このページの注意点を参考に極端な視差が生じない作り方を心がけてください。また、他の人の作品等で視差設定が大きすぎると感じた場合や目が疲れるなどの症状が発生した場合は、休憩して目を休めてください。

立体視について

立体視は上画面のみ対応しています。様々な表示物に対して奥行き座標（Z座標）を設定することで遠近感を与えることができます。（コンソール画面はLOCATE、グラフィック画面はGPRI0、スプライトはSPOFS、BGはBGOFSを使うことでZ座標を指定します）



立体視に関する注意事項

立体表現を取り入れる際には、以下の点に注意して遊んでいるときに目が疲れないような調整を行ってください。

- 1) 背景になる部分に白を多くしない
- 2) 奥行きの段階を背景と手前の2段にしない

白は距離感をつかみにくくする色であり、奥行きと手前の2段だけでは比較する対象が少ないため目が距離感をつかみにくくなります。どちらも目が疲れる原因となります。

- 3) 画面より前に飛び出る映像は視差を控える

液晶面よりも手前側（Zが0以下の設定）に対する飛び出す映像については目が疲れやすくなるので画面内に収まるような見え方の調整を行ってください。画面の端の方で立体表現を設定すると、視差が画面上に出ないことがあり目が距離感をつかみにくい状態が発生します。

もし、他の人が作った作品で立体視の調整が激しいものがあれば3Dを弱めにしてご利用ください。また、作っている途中でも目が疲れたら一度休憩して目を休めてください。

HOMEメニューにある「安全に使用するために」も合わせてご確認ください。

プチコンで扱う画面は、文字を表示するコンソール画面や、グラフィックを表示するグラフィック画面など、役割の違ういくつかの画面を重ねて表示するようになっています。

上画面（3Dスクリーン）

画面最奥から、背景、グラフィック画面、BG画面、SPRITE画面、コンソール画面の5面で構成されています。



背景色	すべての画面の後ろに表示される単色面
グラフィック画面	グラフィック命令で描いた図形が表示される画面
BG画面	キャラを敷き詰めてゲームのマップなどを作るための画面 ・1コマ16×16ドットのキャラを、最大128×127個並べて表示できる ・4層のレイヤーがあり、多重スクロールなどの表現が可能
SPRITE	BG画面などを背景として表示する、ゲームの主人公などのキャラに使用する要素 ・上画面・下画面合わせて、512個までのSPRITEを使用可能 ・サイズは16×16ドットを基本とし、個別に任意のサイズで使用可能 ・複数のSPRITEを連続表示し、アニメーションを作る命令あり
コンソール画面	PRINT命令などで文字を書くことができるテキスト画面

3Dスクリーンには奥行きがあり、その量はZ座標で表します。基準面はZ=0で、手前側はマイナスになります。奥行の範囲は1024~-256ですが、実際の見え方はもっと段階が少なくなり、3Dボリュームの調整によっても結果が変わります。

下画面（タッチスクリーン）

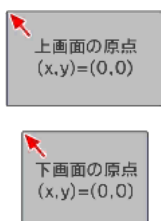
上画面と同様ですが、画面サイズが異なります。キーボードは下画面のみに表示されます。



表示要素の重ね順は、上画面と同様にZ座標で管理します。

上画面と下画面の位置関係

上画面と下画面は、横幅の中央同士が揃うように上下に並びます。座標原点(0,0)はそれぞれ左上となります。



上下連続表示

上画面と下画面は、`XSCREEN 4` 命令によって、上下をつないで1枚のように使うことができます。

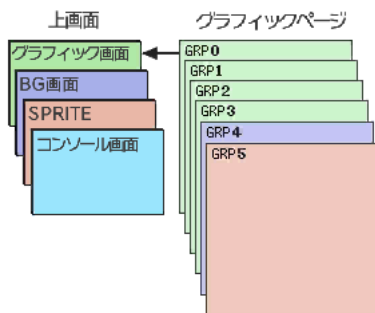


グラフィックページ

プチコン3号には、画面に表示する元画像を置く場所が、全部で6つあります。これらを**グラフィックページ**といいます。

それぞれGRP0～GRP5という名前がついており、決まった表示画面に対応しています。

たとえば「GRP0」というグラフィックページは、上画面のグラフィック画面に対応します。



GRP0にグラフィック命令で絵を描くか、画像データを配置すると、上画面のグラフィック画面にその絵が表示されます。

画面とグラフィックページの対応は、次のようになっています。

●上画面 (DISPLAY 0)

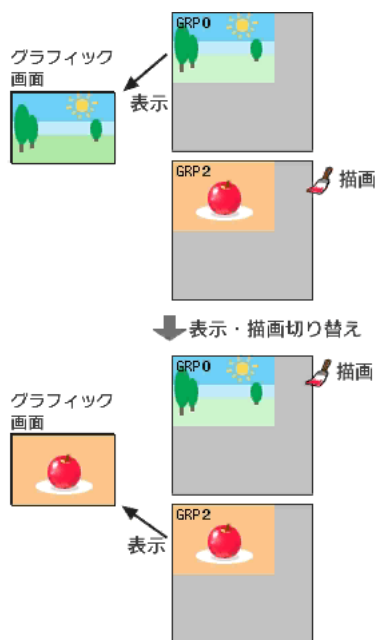
グラフィック画面	GRP0
BG画面	GRP5
SPRITE	GRP4

●下画面 (DISPLAY 1)

グラフィック描画	GRP1
BG画面	GRP5
SPRITE	GRP4

表示ページと描画ページ

複雑なグラフィックを描画すると、描画に時間がかかり、描画の途中過程が見えてしまうことがあります。これを防ぎたいときはグラフィックページをもう1枚使用し、一方のページに描画している間、もう一方のページを表示しておくことができます。



初期状態では、描画ページと表示ページは同じGRP0（下画面ではGRP1）ですが、**GPAGE命令**により、変更可能です。

カラー指定

プチコン3号では画面全体で65536色のカラーを使えます。グラフィックとコンソールでは指定方法が異なります。

表示要素	色数・色指定
グラフィックページ	ドット単位で32768色 指定方法はRGB関数を使用
コンソール画面 (テキスト)	透明色を含む16色から選択

●グラフィックページの色指定

内部的にはRGB各色5ビット+透明1ビット(RGBA=5551)となりますが、指定の際にはRGB関数を使用し、RGB各8ビットとして指定します。

◆描画色をRGB関数で指定する例

GCOLOR RGB(R, G, B)

- ・ R(赤)、G(緑)、B(青)の諧調を各0～255で指定

GCOLOR RGB(A, R, G, B)

- ・ A(透明)は255で不透明、それ以外で透明

●コンソール画面（テキスト）の色指定

1文字ごとに、文字色と文字の背景色を設定できます。それぞれ16色から1色を選びます。色と番号の対応は、SMILEツールの上画面に表示されています。

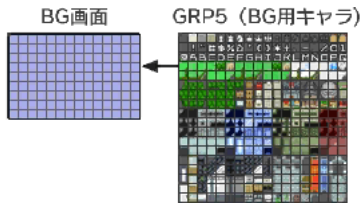
プチコン3号のSmileBASICには、動きのあるゲーム画面を作りやすいよう、BGとSPRITEという仕組みがあります。

BGについて

BGとはBack Ground（バックグラウンド）の略で、家庭用ゲームの画面背景によく使われてきた仕組みです。

16×16ドットのキャラを並べて大きな画面を作ります。同じキャラを並べれば、海や草原のような均一な絵を簡単に作れます。

本製品には、ゲームでよく使用するBG用キャラが多数登録されています。これをBGPOT命令でBG画面に配置するだけですぐ使えます。



◆BGキャラ

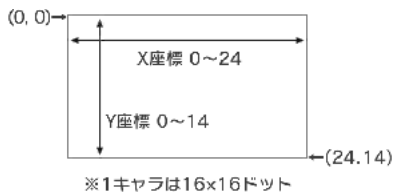
BGキャラは、グラフィックページのGRP5にあらかじめ読み込まれています。1つのキャラは16×16ドットで、0～1023までのキャラ番号が付いています。

何番のキャラがどのようなパターンになっているかは、SMILEツールの下画面で確認することができます。

◆BG画面とレイヤー

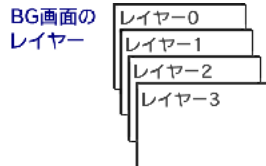
BGキャラは、BG画面に配置して初めて画面に表示されます。同じBGキャラをいくつも並べることができます。

BG画面はキャラ単位の座標を持ちます。一画面には横に25キャラ、縦に15キャラ並びます。なおBGSCREEN命令によって、画面を超える大きさのBG画面を使うことも可能です。



BG画面は、4枚のレイヤーが重なった構造になっており、それぞれにBGキャラを配置できます。遠景と近景を作り分けたり、多重スクロールさせたりという使い方ができます。

BGキャラを配置するには、必ずレイヤー番号(0～3)を指定する必要があります。



各レイヤーの表示位置は、BGOF命令により、ドット単位で変更することができます。これによりスクロールが可能です。

また重ね順は、BGOF命令でZ座標を指定することにより変更可能です。これは3D表示の奥行きにも関係します。

BGキャラを配置する…BGPOT命令

BGキャラを、BG画面の指定したレイヤーに配置する命令です。

書式	BGPOT レイヤー, X, Y, キャラ番号 ・レイヤー：キャラを配置するレイヤーの番号(0～3) ・X, Y：キャラを配置する座標 ・キャラ番号：配置するキャラの番号(0～1023)
使用例	BGPOT 0,12,7,1 レイヤー0の画面中央付近に、BGキャラ1番を配置

BGキャラを敷き詰める…BGFILL命令

指定した長方形の領域を、ひとつのBGキャラで埋める命令です。

書式	BGFILL レイヤー, 始点X, 始点Y, 終点X, 終点Y, キャラ番号 ・レイヤー：キャラを埋めるレイヤーの番号(0～3) ・始点X, Y：配置領域の左上の座標 ・終点X, Y：配置領域の右下の座標
----	---

	・キャラ番号：配置するキャラの番号(0～1023)
使用例	BGFILL 0, 1, 1, 23,13, 2 レイヤー0の画面外周から1キャラ内側の範囲を、BGキャラ2番で埋める

BG画面の表示位置を変更する…BGOFS命令

指定したレイヤーの表示位置と奥行きを変更する命令です。

書式	BGOFS レイヤー, X, Y, Z ・レイヤー：移動するレイヤーの番号(0～3) ・X, Y：表示位置の移動量（ドット単位） プラスでレイヤーが左に・上にずれる ・Z：奥行き方向の座標（奥：1024～液晶面：0～手前：-256） 省略可能
使用例	BGOFS 0, -3, -4 レイヤー0の表示位置を右に3ドット、下に4ドットずらす

BG画面をクリアする…BGCLR命令

BG画面の指定したレイヤーをクリアする命令です。

書式	BGCLR レイヤー ・レイヤー：クリアするレイヤーの番号(0～3)
使用例	BGCLR 0 レイヤー0の表示をクリアする

その他のBG命令

上記のほかにも多くのBG命令があります。使い方はHELPやサンプルプログラムで確認してください。以下は代表的なものです。

◆BGSCREEN命令

BG画面の最大サイズを変更する

◆BGSCALE命令

BG画面を拡大・縮小する

◆BGROT命令

BG画面を回転する

◆BGCOPY命令

BG画面の指定範囲を他の位置にコピーする

◆BGANIM命令

BGによるアニメーションを行う

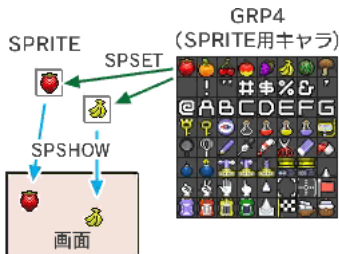
ゲームの背景として使うBGに対し、プレイヤー・敵のキャラクターや弾など、動きのある物体を作る仕組みがSPRITEです。

SPRITEについて

SPRITE（スプライト）はBGとともに、家庭用ゲームの表示によく使われてきた仕組みです。

基本16×16ドットのキャラを、1つの動く物体として画面に表示します。大きさは任意に設定できます。画面には最大512個まで表示可能です(あまりたくさん表示すると動きが遅くなります)。

本製品には、ゲームでよく使用するSPRITE用キャラが多数登録されています。これをSPSET命令で好きな管理番号のSPRITEに割り当て、SPSHOW命令を実行すると画面に表示されます。



◆SPRITE用キャラ

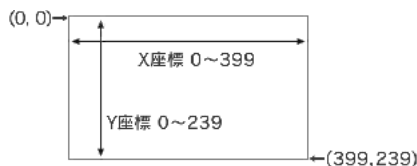
SPRITE用キャラは、グラフィックページのGRP4にあらかじめ読み込まれています。キャラの基本サイズは16×16ドットですが、それ以外のサイズもあります。

キャラ定義番号は、SMILEツールの「SPDEF」で確認してください。

◆SPRITEの画面表示

表示するSPRITEは、SPSET命令で512個まで準備できます。それぞれの番号のSPRITEに、どの番号のキャラ定義を割り当てるかを設定します。

SPRITEの表示座標はSPOFS命令を使い、ドット単位で指定します。BGと違い、キャラ単位の座標ではありません。



重ね順もZ座標を指定することにより変更可能です。これはBGのZ座標と同列に扱われるので、BGの手前に出したり、奥に隠したりすることもできます。

Z座標は3D表示の奥行きにも関係します。

SPRITEを準備する…SPSET命令

画面に表示するSPRITEを準備します。SPRITE同士を区別する管理番号を付け、SPRITEの絵柄とするキャラを選択します。

書式	SPSET 管理番号, キャラ定義番号 ・管理番号：新しいSPRITEにつける任意の番号(0~511) ・キャラ定義番号：このSPRITEの絵柄とするキャラ番号(0~1023)
使用例	SPSET 0,1 SPRITE 0番を準備し、絵柄としてキャラ1（みかん）を設定

SPSET命令は、表示するSPRITEをメモリー上に準備して画面に表示します。（表示させたくない場合は、SPHIDE命令を利用します）

SPRITEの表示位置を変更する…SPOFS命令

指定したSPRITEの表示位置と奥行きを変更する命令です。ゲームキャラクターを動かすときによく使います。

書式	SPOFS 管理番号, X, Y, Z ・管理番号：対象のSPRITEの管理番号(0~511) ・X, Y：画面上の表示座標 ・Z：奥行方向の座標（奥：1024~液晶面：0~手前：-256）省略可能
使用例	SPOFS 0, 192, 112 SPRITE 0番を、画面のほぼ中央に表示する

SPRITEを拡大縮小…SPSCALE命令

指定したSPRITEを拡大・縮小する命令です。

書式	SPSCALE 管理番号, 倍率X, 倍率Y - 管理番号：対象のSPRITEの管理番号(0～511) - 倍率X, 倍率Y：元の大きさに対する横・縦の倍率 0.5=50%, 1.0=100%, 2.0=200%
使用例	SPSCALE 0, 1.5, 1.5 SPRITE 0番のサイズを、縦・横とも1.5倍にする

SPRITEを回転…SPROT命令

指定したSPRITEを回転する命令です。

書式	SPROT 管理番号, 角度 - 管理番号：対象のSPRITEの管理番号(0～511) - 角度：時計回りの角度(0～360) これを超える角度やマイナスの角度（反時計回り）も設定可能。
使用例	SPROT 0, 30 SPRITE 0番を、時計回りに30度回転させる

SPRITEの基点を変更する…SPHOME命令

移動や拡大縮小、回転の際に、SPRITEのどの位置を基点にするかを変更する命令です。初期状態ではSPRITEの左上が基点になっています。

書式	SPHOME 管理番号, 基点X, 基点Y - 管理番号：対象のSPRITEの管理番号(0～511) - 基点X,Y：SPRITEの左上の角からの相対座標 SPRITEが16×16ドットの場合、X=8・Y=8でほぼSPRITE中央が基点となる
使用例	SPHOME 0, 8, 16 SPRITE 0番の基点を、左上の角から右に8ドット、下に16ドットの位置に変更する

その他のSPRITE命令

上記のほかにも多くのSPRITE命令があります。使い方はHELPやサンプルプログラムで確認してください。以下は代表的なものです。

◆SPHIDE命令

SPRITE表示を隠す

◆SPSHOW命令

隠れているSPRITE表示する

◆SPANIM命令

SPRITEによるアニメーションを行う

◆SPLINK命令

複数のSPRITEを連結し、多関節キャラを作る

◆SPUNLINK命令

SPLINK命令による連結を切り離す

◆SPCLR命令

指定したSPRITEの使用を終え、使っていたメモリーを空ける