

GRAPESのHTMLマニュアルのスクリプト部分

スクリプト

- スクリプトを用いてパラメータや残像・軌跡を自由にコントロールすることができます。

GRAPESスクリプトの特徴

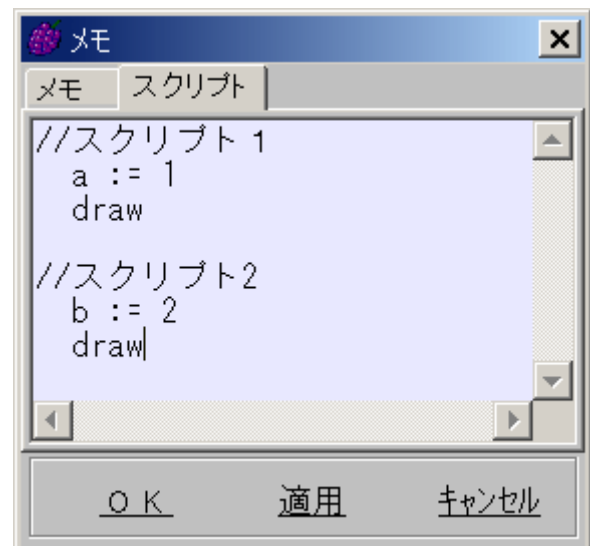
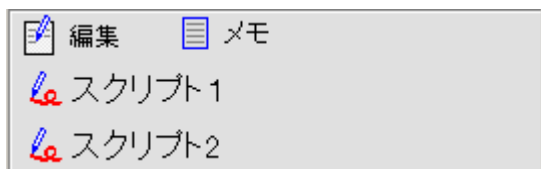
- スクリプトは、GRAPES 内で実行できる命令文で、簡単なプログラムを作ることができます。
- コントロールできるのは、
 - パラメータへの代入
 - 基本図形や曲線の要素への値の代入
 - 残像や表示のコントロール
 - (パラメータ値にもとづく) グラフの描画
 - 与えた波形を持つ音の出力

などで、関数への代入などはできません。

- ローカル変数はありませんが、13 個あるパラメータのすべてを自由に扱うことができます。
- 条件文“if then else”や繰り返し文“for next”, “while wend”が利用できます。
- 個々のスクリプトに対して、スクリプトボタンがメモエリアに表示されます。これをクリックしてスクリプトを実行します。
- データパネル上の指定したデータを変更した場合に、自動実行させることもできます。
- ひとつのスクリプトから別のスクリプトを呼び出して実行することができます。

スクリプトの記述

- スクリプトを書くには、メモウィンドウを開き [スクリプト] タグをクリックします。
- 連続したスクリプト行は、ひとつのスクリプトとみなされる。
- 複数行からなるスクリプトは、上の行から順に実行される。
- 2つ以上のスクリプトを記述するには、スクリプトの間に空白行を入れる。



スクリプト名の表示

- スクリプトは、最初の1行だけが、データパネルのメモエリア内のボタンとして表示される。したがって、スクリプトの1行目にはスクリプトの内容を表す文（スクリプト名など）を記述するのが望ましい。
- スクリプトの1行目に“HideScript”命令があるとき、このスクリプトは表示されない。

スクリプトの実行と中断

- メモエリア内にスクリプト実行用のボタンが表示されるので、これをクリックする。
- スクリプトは、先頭行から1行ずつ実行される。
- スクリプトの実行を中断するには、F1 キーを押す。
- スクリプトに OnChange 文があるとき、指定した要素の変更が加えられると実行される。

記述方法

- 1行には1命令しか書くことができない。
- 1命令を複数行に分けて書くことはできない。
- スクリプトにだけ使える語（For, Next, While, Wend, ClrAImg など）は、大文字でも小文字でもかまわない。
- パラメータ、図形要素、式などは、grapes の文法に従う。したがって、パラメータは必ず小文字で記述し、“Pi”などはそのとおりに記述する必要がある。
- 語と語の間には、半角空白を入れること。
- 行頭には任意個数の半角空白を入れることができる。
- 注釈は、ダブルスラッシュ“//”の後に記述する。
- スクリプトの命令をタイプする代わりに、マウスの右クリックで挿入することができます。



エラー処理

- 文法ミスは、メモウインドウの [OK] ボタンや [更新] ボタンを押したときにチェックされる。
 - 実行時エラーのうち、計算結果がエラーになる場合、実行中の行は無視され、次の行へ制御が移る。
-

スクリプトの文法

- **スクリプト名**

1. スクリプト 1 行目の記述内容がスクリプト名になります。
例：1 行目が「t := 0」のとき、スクリプト名は“t := 0”
2. 1 行目が注釈部分を含んでいるとき、ダブルスラッシュ“//”を除いた注釈部分がスクリプト名になる。
例：1 行目が「t := 0 //初期化」のとき、スクリプト名は“初期化”
3. スクリプトは「スクリプトボタン」としてメモエリアに表示されます。このとき、ボタン上に表示されているのがスクリプト名です。

- **代入文**

1. **パラメータ := 式**
式はGRAPE Sで計算できるものであること。
式が値を持たない場合、代入文は実行されない。
例：a := b²+3
2. **基本図形の要素 := 式**
式の値が代入される。
例：P.x := sin t
P := (2, 3)
3. **Inc(パラメータ, 式)**
4. **Dec(パラメータ, 式)**
式の値だけパラメータ値を増減する。
式を省略した場合 1 だけ増減する。
例：Inc(a, Pi/4), Inc(a)

- **条件文**

1. **If 条件文 Then 実行文**
2. **If 条件文 Then 実行文 Else 実行文**

3. **If 条件文 Then**
 実行文 (複数行でも可)
Endif
4. **If 条件文 Then**
 実行文 (複数行でも可)
Else
 実行文 (複数行でも可)
Endif

- **繰り返し**

1. **While 条件文**
 実行文 (複数行でも可)
Wend
 条件文が成り立つ間, While と Wend の間の実行文を繰り返し実行する.
2. **For パラメータ := 式1 To 式2 Step 式3**
 実行文 (複数行でも可)
Next
 For~Next は, 次の文と等価である.
 パラメータ := 式1
 While パラメータ \leq 式2
 実行文
 Inc(パラメータ, 式3)
 Wend
 式3を省略した場合1ずつ増加する.

- **描画命令と計算命令**

1. **Draw**
 現在のパラメーター値に従ってグラフを描く.
 軌跡を残さないときには, コロンをつけた"**Draw:**"を用いる.
 画面上のすべての表示が更新される.
2. **OverDraw**
 現在のパラメーター値に従ってグラフを描く.
 現在の画面の上に, 新しく計算したグラフを描き加える. 不要になったグラフの削除は行わないために描画は速い. ただし, 速さの点では, 次のCalcの方が勝っている.
 正しい画面表示にするには, Refresh や Draw を用いる.
3. **Calc**
 現在のパラメーター値に従ってグラフを描く.
 軌跡を残さないときには, コロンをつけた"**Calc:**"を用いる.
 画面表示の更新は一切されない. (したがって高速)
 画面表示が必要などときには, Refresh や Draw を用いる.
4. **Refresh**
 表示を更新する.

Calc 命令によって変更された残像や軌跡を表示するのに用いる。

また、メモも更新される。

Draw は計算してから描画するが、Refresh は描画の更新をするだけ。

5. RefreshMemo

メモだけを更新する

- **表示領域の変更**

1. **SetArea (x1, y1, x2, y2)**

2 点 (x1, y1) , (x2, y2) を対角線とする領域を、画面全体に表示する。

- **残像消去と残像化**

1. **ClrAImg**

すべての図形の残像を消去

2. **ClrAImg (図形名)**

図形の残像を消去

図形名はカンマで区切って複数記述することができます。

例 : ClrAImg (y1, C2, A, L8)

陽関数 y1, 陰関数 C2, 図形 A, 線分 L8 の残像を消す。

3. **SetAImg**

そのとき描かれているグラフの残像化

4. **AImgOff**

残像を残さない。残像 Off ボタンを押すのと同様

5. **AImgOn**

残像 Off 状態の解除

- **個々のグラフの表示・非表示**

1. **ShowObj (図形名)**

図形を表示する

図形名はカンマで区切って複数記述することができます。(HideObj, InvShowObj も同様)

例 : ShowObj (y1, C2, A, L8)

陽関数 y1, 陰関数 C2, 図形 A, 線分 L8 を表示する。

2. **HideObj (図形名)**

図形を非表示にする

3. **InvShowObj (図形名)**

図形の表示を切り替える。

これは、データパネルの表示スイッチ(図形ラベル)をクリックするのと同様です。

- **個々のグラフの色の変更**

1. **SetColor(図形名, 色)**

図形の表示色を変更します。基本図形や曲線では点の色、関数のグラフではグラフ色を、連結図形では線の色を変更する。

図形は、一度にはひとつしか指定できません。

色は、RGB を表す 24 ビット整数で与える。GRAPES は 16 進数を扱うことはできないので、rgb 関数や、hsl 関数を利用して色を指定する。

例：SetColor(y1 , rgb(0.8, 0, 1))

この例では、関数 y1 のグラフが紫色になる。

2. **SetLineColor(図形名, 色)**

図形の表示色を変更する。基本図形や曲線では線の色、関数のグラフではグラフ色を、連結図形では線の色を変更する。

3. **SetInnerColor(図形名, 色)**

基本図形、曲線、連結図形のペイント色を変更する。

- **与えた関数を波形とする音を出す。**

1. **Play (パラメータ, 開始秒, 終了秒, 波形関数)**

量子化 16 ビット, サンプリングレート 22050Hz (変更可能) , ステレオの音を出す。

- 例 : 振動数 440Hz の正弦波を 2 秒間鳴らすのなら,
Play(t , 0 , 2 , sin880Pit) (補足: 左記の“Pit”は π と t の積です)
- 例 : 左チャンネルから 440Hz, 右チャンネルから 441Hz の音を出すのなら,
Play(t , 0 , 2 , (sin880Pit , sin882Pit)) (補足: ステレオ出力時にはベクトルデータを与えます)
- パラメータは, “t” 以外のパラメータや変数 “x” でも構いません。
- 最大, 480,000 ÷ サンプリングレート (秒) まで可能です。
- 演奏が開始されるのは, すべてのデータの計算終了後です。演算速度の遅いマシンでは, 演奏開始までに時間がかかることがあります。

2. **Play**

- 下記 PlayAfter コマンドで作成した Wave データを演奏する。
- 最後に演奏した Wave データを再演奏する。

3. **PlayAfter (パラメータ, 開始秒, 終了秒, 波形関数)**

Play コマンドと異なり演奏しません。

- PlayAfter コマンドで作成された Wave データは, 作業用メモリー上に配置されたまま, 演奏されません。
- 作業用メモリー上に Wave データが残っている状態で次の PlayAfter コマンドが実行された場合, これら Wave データは結合されます。
- 作業用メモリー上に作成された Wave データを演奏するには, Play コマンドを実行します。
- 複数の Play () 命令を連続して実行した場合, 高性能なパソコンでは演奏順序が乱れることがあります。このような場合には, PlayAfter コマンドでひと繋りのデータを作成してから, 最後に Play コマンドで演奏するようにします。

4. **PlaySync (パラメータ, 開始秒, 終了秒, 波形関数)**

Play コマンドと異なるのは次の 1 点だけです.

- Play コマンドでの演奏はバックグラウンドで行われるのに対して, PlaySync コマンドでは演奏を終えるまで次の命令を実行できません.

5. **SetSampleRate(サンプリングレート)**

サンプリングレートを変更する.

- 例 : 48,000Hz に設定するには
SetSampleRate(48000)

6. **音の保存**

- 最後に出した音は, [ファイル] メニューの [音の保存] で, wave ファイルとして保存することができます.

• **時間待ち**

1. **Wait(式)**

式で指定した時間停止する. 単位はミリ秒.

2. **Settime(式)**

実行文 (複数行でも可)

Wait

式で指定した時間を, Settime を実行したときからカウントし始め, Wait 実行時に時間が残っていれば指定時間が経過するまで停止する.

3. **Pause**

スクリプトの実行を停止し, 入力を待つ.

Pause(メッセージ) とした場合, メッセージ表示する.

• **自動実行条件**

1. **On 要素名 Change**

この文はスクリプトの 1 行目か 2 行目に書く.

指定した要素が変更されるとスクリプトが自動実行される.

指定できる要素は, パラメータ, 図形(P など), ユーザー関数, 陽関数.

例 : On a, b, f Change (パラメーター a, b あるいは関数 f のいずれかが変更されると実行される)

• **サブスクリプトの呼び出し**

1. **Call(サブスクリプト名)**

同じプロジェクト内の別のスクリプトを実行することができます. 実行後は次の行に制御が移ります.

- スクリプトの実行停止

1. End

実行中のスクリプトを終了します。もし、他のスクリプトから Call 命令によって呼ばれているときには、元のスクリプトに制御が戻ります。

スクリプトの読み込み

- 記録してあるファイルからスクリプトを読み込むことができます。
 - 「メモの読込」を行うと、スクリプトを読み込むことができます。
このとき、関数やパラメータなどメモやスクリプト以外のデータは変更されません。
-