

AJAN

GUI コマンド

リファレンス

## 目 次

<b>第 1 章</b>	<b>はじめに</b>	<b>6</b>
<b>第 2 章</b>	<b>機能説明</b>	<b>7</b>
2.1	GUI の基本的な記述形 .....	7
2.1.1	GUI を使う宣言(OPTION GUI) .....	9
2.1.2	ウィンドウの作成と GUI 部品の貼り付け(GUCREATE WINDOW) .....	9
2.1.3	ウィンドウの表示(GUSHOW).....	10
2.1.4	イベントルーチンの登録(ON GUEVT CALL, GUEVT ON) .....	10
2.1.5	GUI メインイベントループを回す(GUMAIN LOOP).....	11
2.1.6	GUI メインイベントループを終了させる(GUMAIN QUIT) .....	11
2.2	高度な使い方.....	12
2.2.1	自分で GUI メインイベントループを作って回す(GUMAIN ONELOOP) .....	12
2.3	注意と参考情報.....	13
2.3.1	注意：GUI コマンドは、メインスレッドでのみ使う事 .....	13
2.3.2	参考：端末から GUI アプリを呼び出すと、警告メッセージが出力される .....	13
2.3.3	注意：AJAN 統合開発環境上でデバッグ時の、トレースバーの表示挙動.....	14
2.3.4	注意：スーパーユーザで実行する際の、日本語入力の制限 .....	14
2.3.5	注意：スーパーユーザで実行する際に「uid mismatch」警告文字 .....	14
2.3.6	注意：画面無しの状態では、GUI アプリは動かない .....	15
<b>第 3 章</b>	<b>リファレンス</b>	<b>16</b>
3.1	コマンド一覧.....	16
3.2	GUI に関する共通項目 .....	20
3.2.1	フォント設定 .....	20
3.2.2	サイズ設定 .....	22
3.2.3	ポジション設定 .....	23
3.2.4	ツールチップ設定 .....	24
3.2.5	共通項目の省略について .....	25
3.3	GUI に関する関数・命令 .....	25
3.3.1	コマンド個別説明 .....	25
3.3.1.1	OPTION GUI.....	25
3.3.1.2	GUCREATE WINDOW .....	26
3.3.1.3	GUCREATE BUTTON .....	27
3.3.1.4	GUUPD BUTTON.....	28
3.3.1.5	GUCREATE LABEL .....	29
3.3.1.6	GUUPD LABEL .....	30
3.3.1.7	GUCREATE TEXTBOX.....	31
3.3.1.8	GUUPD TEXTBOX .....	32
3.3.1.9	GUCREATE TEXTAREA .....	33
3.3.1.10	GUUPD TEXTAREA.....	34
3.3.1.11	GUCREATE COMBOBOX .....	34

3.3.1.12	GUADD COMBOBOX .....	35
3.3.1.13	GUUPD COMBOBOX.....	36
3.3.1.14	GUCREATE RADIOBTN.....	37
3.3.1.15	GUADD RADIOBTN .....	38
3.3.1.16	GUUPD RADIOBTN .....	39
3.3.1.17	GUCREATE CHECKBOX .....	39
3.3.1.18	GUUPD CHECKBOX.....	40
3.3.1.19	GUCREATE IMAGE .....	41
3.3.1.20	GUUPD IMAGE.....	43
3.3.1.21	GUUPD IMAGE FIXED .....	43
3.3.1.22	GUUPD IMAGE ROTATE.....	43
3.3.1.23	GUCREATE SCALE.....	44
3.3.1.24	GUUPD SCALE .....	45
3.3.1.25	GUCREATE SWITCH.....	46
3.3.1.26	GUUPD SWITCH.....	47
3.3.1.27	GUCREATE CALENDAR .....	48
3.3.1.28	GUUPD CALENDAR.....	49
3.3.1.29	GUCREATE VIDEO .....	50
3.3.1.30	GUUPD VIDEO .....	51
3.3.1.31	GUSHOW .....	53
3.3.1.32	GUMAIN LOOP.....	54
3.3.1.33	GUMAIN ONELOOP .....	54
3.3.1.34	GUMAIN QUIT.....	55
3.3.1.35	GUDSP.....	56
3.3.1.36	GUGET\$ .....	57
3.3.1.37	GUGET\$ VIDEO.....	58
3.3.1.38	GUSIZE.....	59
3.3.1.39	GUFONT.....	60
3.3.1.40	GUTIP.....	60
3.3.1.41	GUDELETE.....	61
3.3.1.42	GUMOVE WINDOW.....	61
3.3.1.43	GUMSGBOX.....	62
3.3.1.44	GUMOUSE.....	63
3.3.1.45	GUEVT ON .....	64
3.3.1.46	GUEVT OFF .....	65
3.3.1.47	ON GUEVT CALL.....	66
3.3.1.48	GUDIALOGBOX OPEN\$.....	68
3.3.1.49	GUDIALOGBOX SAVE\$.....	70
3.3.1.50	GUDIALOGBOX DIRECTORY\$ .....	72
3.3.1.51	GUNOTIFY SEND.....	74
3.4	グラフに関する関数・命令 .....	75
3.4.1	グラフに関する個別説明.....	75
3.4.1.1	棒グラフ(BARCHART).....	75
3.4.1.2	折れ線グラフ(LINEGRAPH).....	78
3.4.1.3	円グラフ(PIECHART) .....	81
3.4.2	コマンド個別説明 .....	84

3.4.2.1	GUCREATE BARCHART .....	84
3.4.2.2	GUADD BARCHART VALUE .....	85
3.4.2.3	GUUPD BARCHART VALUE .....	85
3.4.2.4	GUADD BARCHART TITLE .....	86
3.4.2.5	GUUPD BARCHART LABEL .....	87
3.4.2.6	GUADD BARCHART DATALABEL .....	88
3.4.2.7	GUUPD BARCHART DATALABEL .....	89
3.4.2.8	GUADD BARCHART RANGE .....	89
3.4.2.9	GUUPD BARCHART RANGE .....	90
3.4.2.10	GUUPD BARCHART GRID .....	91
3.4.2.11	GUUPD BARCHART DRAW .....	91
3.4.2.12	GUDSP BARCHART .....	92
3.4.2.13	GUCREATE LINEGRAPH .....	93
3.4.2.14	GUADD LINEGRAPH VALUE .....	94
3.4.2.15	GUUPD LINEGRAPH VALUE .....	95
3.4.2.16	GUADD LINEGRAPH TITLE .....	96
3.4.2.17	GUUPD LINEGRAPH LABEL .....	97
3.4.2.18	GUADD LINEGRAPH DATALABEL .....	98
3.4.2.19	GUUPD LINEGRAPH DATALABEL .....	99
3.4.2.20	GUADD LINEGRAPH RANGE .....	100
3.4.2.21	GUUPD LINEGRAPH RANGE .....	101
3.4.2.22	GUADD LINEGRAPH GUAGE .....	102
3.4.2.23	GUUPD LINEGRAPH GUAGE .....	103
3.4.2.24	GUUPD LINEGRAPH GRID .....	104
3.4.2.25	GUUPD LINEGRAPH DRAW .....	105
3.4.2.26	GUDSP LINEGRAPH .....	106
3.4.2.27	GUCREATE PIECHART .....	107
3.4.2.28	GUADD PIECHART VALUE .....	108
3.4.2.29	GUUPD PIECHART VALUE .....	109
3.4.2.30	GUADD PIECHART TITLE .....	110
3.4.2.31	GUADD PIECHART DATALABEL .....	111
3.4.2.32	GUUPD PIECHART DATALABEL .....	111
3.4.2.33	GUUPD PIECHART DRAW .....	112
3.4.2.34	GUDSP PIECHART .....	112
3.5	グラフィック描画に関する関数・命令 .....	113
3.5.1	グラフィック描画に関する個別説明 .....	113
3.5.2	コマンド個別説明 .....	114
3.5.2.1	GUCREATE PIC .....	114
3.5.2.2	GUGCLS .....	115
3.5.2.3	GUGCOLOR .....	116
3.5.2.4	GUGLINE .....	116
3.5.2.5	GUGRECTANGLE .....	117
3.5.2.6	GUGCIRCLE .....	118
3.5.2.7	GUGARC .....	119
3.5.2.8	GUGPSET .....	120
3.5.2.9	GUGDRAWTEXT .....	121

3.5.2.10	GUGDRAWFILE .....	122
3.5.2.11	GUGSAVEFILE .....	123
3.5.2.12	GUGDATAGET .....	124
3.5.2.13	GUGDATASET .....	125
3.5.2.14	GUGDRAWBLIT .....	126
3.6	画像処理に関する関数・命令 .....	127
3.6.1	コマンド個別説明 .....	127
3.6.1.1	用語説明 .....	127
3.6.1.2	IMG_READ .....	127
3.6.1.3	IMG_WRITE .....	128
3.6.1.4	IMG2GUG_DATA .....	129
3.6.1.5	GUG2IMG_DATA .....	129
3.6.1.6	IMG_CONVERT_COLOR .....	130
3.6.1.7	IMG_RESIZE .....	131
3.6.1.8	IMG_BINARIZE .....	132
3.6.1.9	IMG_BINARIZE_AUTO .....	133
3.6.1.10	IMG_FIND_CONTOURS .....	134
3.6.1.11	IMG_ERODE .....	136
3.6.1.12	IMG_DILATE .....	137
3.6.1.13	IMG_MEAN_FILTER .....	138
3.6.1.14	IMG_GAUSSIAN_FILTER .....	139
3.6.1.15	IMG_BILATERAL_FILTER .....	140
3.6.1.16	IMG_SOBEL_FILTER .....	141
3.6.1.17	IMG_LAPLACIAN_FILTER .....	142
3.6.1.18	IMG_SHARP_FILTER .....	143
3.6.1.19	IMG_CANNY_EDGE .....	144
3.6.1.20	IMG_GET_ROTATION_MATRIX .....	145
3.6.1.21	IMG_WARP_AFFINE .....	146
3.6.1.22	IMG_MATCH_TEMPLATE .....	147
3.6.1.23	IMG_PHASE_CORRELATION .....	149
3.6.1.24	IMG_WARP_POLAR .....	150
<b>第 4 章</b>	<b>サンプルプログラム</b>	<b>152</b>
4.1	サンプルプログラム .....	152
<b>第 5 章</b>	<b>索引</b>	<b>156</b>
<b>第 6 章</b>	<b>重要な情報</b>	<b>158</b>

## 第1章 はじめに

---

本ドキュメントは、AJANのGUI(グラフィカル・ユーザー・インタフェース)コマンドの説明を記載しています。

GUIコマンド以外のコマンド(標準コマンド、IO制御コマンドなど)は、別マニュアルを用意しているので、別途参照してください。

本ドキュメントでは、説明で表現している表記として下記のように定義します。

- ・ コマンドの書式の説明において、[ ]内の引数は省略できます。
- ・ 文字の大小について

コマンドは大文字 / 小文字のどちらでも動作します。

変数名は大文字 / 小文字も同じものとして扱われます。

ファイルパス / ファイル名は大文字/小文字で区別されます。



本ドキュメント記載の、AJANはIoT用プログラミング言語です。

Interface Linux System上でのみ動作可能です。

## 第2章 機能説明

### 2.1 GUIの基本的な記述形

GUI(グラフィカル・ユーザ・インタフェース)は、ウィンドウにボタンやテキストなどのGUI部品を配置し、利用者がボタンを押したり、テキストに文字列を入力するなどの操作に逐次対応するアプリケーションの操作形態を指します。

従来のアプリケーションは、制御や分岐以外では、ほぼ上から下へ順々に命令を実行しますが、GUIアプリケーションは、利用者からの操作に対して対応するので、一般にイベントドリブンと呼ばれる処理の流れになるように、プログラムを記述します。

以下に、GUIアプリケーションの基本的なサンプルコードの事例を示します。

```

1  OPTION GUI      ' GUI を使う事を宣言
2
3  ' ウィンドウの右上の「x」印をクリックした際のイベント処理
4  SUB EVT_WINDOW(NUM, NAME$, EVT)
5      ' GUMAIN LOOP の処理を終了させます
6      GUMAIN QUIT()
7  END SUB
8
9  ' ボタンをクリックした際のイベント処理
10 SUB EVT_BUTTON(NUM, NAME$, EVT)
11     ? "ボタンが押された:"; NUM; " ["; NAME$; "]" "; EVT
12     ? "テキストの内容:"; GUGET$ TEXTBOX(10)
13 END SUB
14
15 ' ===== 以下、メイン処理 =====
16
17 FONT$ = "32, 0, 0, 14, 1, 9"
18 ' ウィンドウを作成
19 GUCREATE WINDOW 1, "ウィンドウ", 1, FONT$,,,
20
21 FONT$ = "32, 0, 0, 14, 1, 8"
22 ' ウィンドウにテキストを配置
23 GUCREATE TEXTBOX 1, 10, "テキスト", FONT$, "100, 20", "20, 20", ""
24 ' ウィンドウにボタンを配置
25 GUCREATE BUTTON 1, 20, "ボタン", FONT$, "100, 20", "20, 80", ""
26
27 ' ウィンドウの右上の「x」印をクリックした際のイベントルーチン
28 ON GUEVT WINDOW 1 CALL EVT_WINDOW
29 ' ボタンがクリックされた際のイベントルーチン(EVT_BUTTON)を登録
30 ON GUEVT BUTTON 20 CALL EVT_BUTTON
31 ' 登録したイベントルーチンを有効にする
32 GUEVT WINDOW 1 ON
33 GUEVT BUTTON 20 ON
34
35 ' ウィンドウを表示
36 GUSHOW 1
37
38 ' GUI メインイベントループを開始 - ウィンドウを閉じるまでコマンドから戻ってこない
39 GUMAIN LOOP

```

40	
41	' ウィンドウを削除する
42	GUDELETE WINDOW 1
43	
44	? "bye. bye"
45	END

このサンプルを実行すると、テキストボックスとボタンが付いたウィンドウが一つ開きます。利用者は、テキストボックスに任意の文字列を記入し、ボタンをクリックすると、コンソールに例えば以下のようなメッセージが表示されます。

ボタンが押された:20 [BUTTON] 0
テキストの内容:テキスト
ボタンが押された:20 [BUTTON] 0
テキストの内容:テキスト 123

ウィンドウの右上の「×」印をクリックすると、ウィンドウが閉じて、サンプルの実行が終了します。

次に、サンプルコードの重要な所を、順に説明します。



### 2.1.1 GUIを使う宣言(OPTION GUI)

サンプルコードの1行目に「OPTION GUI」が記述されています。

```
OPTION GUI      ' GUI を使う事を宣言
```

GUIコマンドを利用するプログラムでは、必ず「OPTION GUI」を記述してください。

AJANは、この呼び出しがある事で、内部でGUI処理に必要な初期化を実施します。

### 2.1.2 ウィンドウの作成とGUI部品の貼り付け(GUCREATE WINDOW)

GUIアプリケーションは、ウィンドウを使って対話的に処理を行う必要があるため、ウィンドウを作成する必要があります。

これは、サンプルコードの19行目の「GUCREATE WINDOW」で行っています。

```
FONT$ = "32, 0, 0, 14, 1, 9"
' <フォント名>VL ゴシック <斜体>なし<太字>なし<サイズ>14<文字色>BLACK <背景色>GRAY

' ウィンドウを作成
GUCREATE WINDOW 1, "ウィンドウ", 1, FONT$,,,
' <ウィンドウ番号>1 <タイトル>"ウィンドウ" <最小/最大/閉じるボタン>表示
```

ウィンドウを作成するだけでは、利用者との対話ができないので、テキストボックスやボタンなどのGUI部品を、ウィンドウに貼り付ける必要があります。

これは、サンプルコードの21から25行の「GUCREATE TEXTBOX」および「GUCREATE BUTTON」で行っています。

```
FONT$ = "32, 0, 0, 14, 1, 8"
' ウィンドウにテキストを配置
GUCREATE TEXTBOX 1, 10, "テキスト", FONT$, "100, 20", "20, 20", ""
' ウィンドウにボタンを配置
GUCREATE BUTTON 1, 20, "ボタン", FONT$, "100, 20", "20, 80", ""
```

作成するウィンドウや貼り付けるGUI部品は、一意に認識する番号を割り当てる必要があります。これらの番号は、プログラムを作る方自身が、自分で定義して割り振り、それをプログラムコードに記述します。

例えば、ウィンドウの場合、「GUCREATE WINDOW 1, ...」と記述する事で、作成するウィンドウのウィンドウ番号として「1」を定義しています。

テキストボックスでは、「GUCREATE TEXTBOX 1, 10, ...」と記述する事で、作成するテキストボックスのテキストボックス番号として「10」を定義し、ウィンドウ番号「1」に貼り付けるよう指示しています。

### 2.1.3 ウィンドウの表示(GUSHOW)

作成したウィンドウは、そのままでは画面に表示されないので、表示するよう指示します。  
これは、サンプルコードの36行目の「GUSHOW」で行っています。

```
GUSHOW 1
```

### 2.1.4 イベントルーチンの登録(ON GUEVT CALL, GUEVT ON)

ここまでの処理で、ウィンドウの作成と表示はできますが、このままでは利用者からの操作に対して、何も対応できません。

そのため、ボタンをクリックしたり、テキストを入力した際に、反応して処理を行う イベントルーチンを記述して登録します。

イベントルーチンが記述されている箇所は、サンプルコードの3行から13行の、ユーザー定義のサブルーチン「EVT\_WINDOW」および「EVT\_BUTTON」です。

記述したイベントルーチンの登録は、サンプルコードの27行から30行の「ON GUEVT CALL」で行っています。

```
' ウィンドウの右上の「×」印をクリックした際のイベントルーチン
ON GUEVT WINDOW 1 CALL EVT_WINDOW
' ボタンがクリックされた際のイベントルーチン(EVT_BUTTON)を登録
ON GUEVT BUTTON 20 CALL EVT_BUTTON
```

例えば、「ON GUEVT BUTTON 20 CALL EVT\_BUTTON」は、ボタン番号「20」のボタンに対して、ボタンがクリックされると、ユーザー定義の「EVT\_BUTTON」 イベントルーチンを呼び出すよう登録します。

イベントルーチンを登録しただけでは、GUI部品の操作に対して応答しません。

登録したイベントルーチンを「有効」にする必要があります。

イベントルーチンの「有効」は、サンプルコードの31行から33行の「GUEVT ON」で行っています。

```
' 登録したイベントルーチンを有効にする
GUEVT WINDOW 1 ON
GUEVT BUTTON 20 ON
```

登録したイベントルーチンを「有効」にする事で、ボタンをクリックした際に別のテキストボックスの、文字列の内容を読み取ったり、ウィンドウの右上の「×」印をクリックした際にアプリケーションを終了したり。といった事ができるようになります。

### 2.1.5 GUIメインイベントループを回す(GUMAIN LOOP)

ここまでで、ウィンドウの作成と表示、イベントルーチンの登録と有効など、GUIアプリケーションとして必要な準備作業がすべて整いました。

GUIアプリケーションとして対話処理を行うようにするために、OSに依頼して、GUIのメインイベントループ処理を実行します。

GUIのメインイベントループは、サンプルコードの39行の「GUMAIN LOOP」で行っています。

```
GUMAIN LOOP
```

「GUMAIN LOOP」を呼び出すと、OS内部でGUIのメインイベントループが回ります。

以降、ボタンをクリックした際のイベントルーチンの呼び出しは、このメインイベントループから呼ばれます。

ウィンドウを閉じたり、次項の「GUMAIN QUIT」を呼び出して、GUIメインイベントループを終了させない限り、「GUMAIN LOOP」は抜け出せません。

### 2.1.6 GUIメインイベントループを終了させる(GUMAIN QUIT)

GUIアプリケーションを終了させるには、GUIメインイベントループを終了させる呼び出しを行います。

GUIメインイベントループの終了は、サンプルコードの6行目の「GUMAIN QUIT」で行っています。

```
GUMAIN QUIT()
```

「GUMAIN QUIT」を呼び出すと、「GUMAIN LOOP」内で回っている、GUIメインイベントループが終了し、次行以降の後片付けを行って、アプリケーションは終了します。

## 2.2 高度な使い方

### 2.2.1 自分でGUIメインイベントループを作って回す(GUMAIN ONELOOP)

「GUMAIN LOOP」を呼び出すと、OS内部でGUIメインイベントループが回る。と説明しました。これとは別に、自分自身でGUIメインイベントループを回す事が可能です。例えば、以下のように。

```
' 通常の GUI メインイベントループ呼び出し
GUMAIN LOOP
↓
' 自分自身で GUI メインイベントループを回す
DO WHILE TRUE
    TMP = GUMAIN ONELOOP(0)
LOOP
```

この関数を応用すると、例えば 約3秒後に、GUIメインイベントループを自動的に終了する。といった記述が簡単にできます。

```
ST = CLOCK
DO WHILE LOOP
    TMP = GUMAIN ONELOOP(0)
    IF (CLOCK - ST) > 3.0 THEN
        EXIT DO
    END IF
LOOP
```



1回の呼び出しで、OSから要求のあったGUI描画に必要な大量の処理要求の1つを消化します。このため、円滑に描画を行うには、GUMAIN ONELOOP は、頻繁に呼び出す事が肝要です。

## 2.3 注意と参考情報

ここでは、GUIコマンドを使用する際の諸注意および参考情報を紹介します。

### 2.3.1 注意：GUIコマンドは、メインスレッドでのみ使う事

GUIコマンドの呼び出しや実行は、メインスレッドでのみ使えます。

「ATTACH THREAD」で作成したスレッド処理や、一部の割り込み命令（「ON TIMER CALL」など）の処理ルーチンでは、呼び出さないでください。



この制限事項は、GUIコマンドが利用している、OSライブラリ (Gtk+) の制限事項に由来します。

### 2.3.2 参考：端末からGUIアプリを呼び出すと、警告メッセージが出力される

端末から、例えばGUIサンプル:BUTTON.AJNをコンパイルした GUIアプリケーションを実行すると、以下のような警告メッセージが出ます。

```
$ ./BUTTON
```

```
** (BUTTON:26553): WARNING **: Couldn't register with accessibility bus: Did not receive
a reply. Possible causes include: the remote application did not send a reply, the message
bus security policy blocked the reply, the reply timeout expired, or the network connection
was broken.
```

```
部品番号:1  部品名:WINDOW  内部イベント種別:0
```

```
...
```

これは、GUIが使用しているライブラリの制約に依存するものです。

どうしても気になるようでしたら、「**NO\_AT\_BRIDGE=1**」を環境変数に付加します。

例えば以下のようにします。

```
$ NO_AT_BRIDGE=1 ./BUTTON
```

```
部品番号:1  部品名:WINDOW  内部イベント種別:0
```

```
...
```

### 2.3.3 注意：AJAN統合開発環境上でデバッグ時の、トレースバーの表示挙動

AJAN統合開発環境のデバッグモードで、中断、自動トレース、ステップ・イン、ステップ・オーバーなどを用いると、実行行にトレースバーが表示されますが、GUIアプリケーションは、何も処理を行っていない場合は「GUMAIN LOOP(P. 54)」を常に実行している関係上、デバッグ実行で以下の条件を満たした場合、基本的に「GUMAIN LOOP(P. 54)」にトレースバーが表示されます。

- ・ 何も操作を行っていない状態でデバッグ実行を中断する。
- ・ 「GUMAIN LOOP(P. 54)」でトレースバーが表示されている状態で、ステップ・イン、ステップ・オーバー等を行った場合。
- ・ 「ON GUEVT CALL(P. 66)」で、処理ルーチン(特に END SUBなど)を抜けた時

実際には、OS内部でGUIイベントループが回って処理を続けており、別のイベントによる処理ルーチンが走る時や、ウィンドウを閉じて イベントループを終了する時に、次の実行行にトレースバーが移ります。

### 2.3.4 注意：スーパーユーザで実行する際の、日本語入力の制限

GUIアプリケーションをスーパーユーザで実行すると、テキストエリアやテキストボックスなど文字列を入力するとき、使用する日本語入力システムの仕様により、日本語入力モードへの切り替え機能が働かない事がある事を確認しています。

(使用する日本語入力システムの組み合わせでは、動作するものもあるようです)

コンパイルした実行ファイルに対して、端末から呼び出すとき、以下のように環境変数を付加してやる事で、スーパーユーザ実行下で、日本語入力ができる事を確認しています。

```
# ↓以下は、スーパーユーザ下で実行した場合の例です。(赤文字が重要)
root@xxx:/root# env GTK_IM_MODULE=xim QT_IM_MODULE=xim <実行ファイルパス>
```

### 2.3.5 注意：スーパーユーザで実行する際に「uid mismatch」警告文字

「2.3.4 注意：スーパーユーザで実行する際の、日本語入力の制限」で紹介したように、環境変数への設定で、日本語入力に用いる日本語入力システムを任意に変更できます。

日本語入力システムに「uim」を指定して、スーパーユーザで起動した時、テキストボックスなどの表示時に、日本語入力システムが標準エラー出力に対して「uid mismatch」という警告文字を出力する場合があります。

AJAN IDEは、ログイン時の環境変数を使用するため、「スーパーユーザで実行」時に、コンソールウィンドウに同じ警告文字が出力される場合があります。

### 2.3.6 注意：画面無しの状態では、GUIアプリは動かない

画面無しの状態では、表示する画面が無いので、当然 GUIアプリは動きません。

画面無しの状態とは、例えば、Webサーバーの環境下で動かそうとしたり、デーモンと呼ばれるサービスプログラムの環境下で動かしたり、ログインしていない状態などです。

また、OSを起動する際、single user mode で選んで起動すると、昔のDOS画面のような 黒の背景に白字のみで、左上に「アプリケーション」の選択メニューが全く無い画面で起動しますが、この時も GUIを表示する対象が存在しないので動きません。

他に、画面が表示されていても、GUIアプリに 画面に表示する為の情報が正しく渡らないと、やはりGUIアプリは動きません。



例えば、以下のように GUIアプリを動かす際、「DISPLAY」環境変数を空にして動かそうとすると、「Could not connect: 接続を拒否されました」というようなエラーメッセージが出ます。

```
$ DISPLAY='' ./GUI アプリへのパス名
Unable to init server: Could not connect: 接続を拒否されました

(...): Gtk-WARNING **: cannot open display
```

## 第3章 リファレンス

使用できるGUI(グラフィカル・ユーザー・インタフェース)コマンドの使い方について記載します。

	制限事項については、「注意」に記載しています。
	使用例は動作を保証するものではありません。 実際の使い方は各種サンプルプログラムを参照してください。

### 3.1 コマンド一覧

コマンド名	機能
GUIに関する関数・命令	
OPTION GUI	GUIアプリケーションを実行することを宣言します。
GUCREATE WINDOW	ウィンドウを作成します。
GUCREATE BUTTON	指定したウィンドウにボタンを配置します。
GUUPD BUTTON	ボタンの表示内容を更新します。
GUCREATE LABEL	指定したウィンドウにラベルを配置します。
GUUPD LABEL	ラベルの表示内容を更新します。
GUCREATE TEXTBOX	指定したウィンドウにテキストボックスを配置します。
GUUPD TEXTBOX	テキストボックスの表示内容を更新します。
GUCREATE TEXTAREA	指定したウィンドウにテキストエリアを配置します。
GUUPD TEXTAREA	テキストエリアの表示内容を更新します。
GUCREATE COMBOBOX	指定したウィンドウにコンボボックスを配置します。
GUADD COMBOBOX	指定したコンボボックスにリストを追加します。
GUUPD COMBOBOX	コンボボックスの選択内容を更新します。
GUCREATE RADIOBTN	指定したウィンドウにラジオボタンの領域を作成します。
GUADD RADIOBTN	ラジオボタンの領域にラジオボタンを追加します。
GUUPD RADIOBTN	ラジオボタンの選択内容を更新します。
GUCREATE CHECKBOX	指定したウィンドウにチェックボックスを配置します。
GUUPD CHECKBOX	チェックボックスの選択状態を更新します。
GUCREATE IMAGE	指定したウィンドウに画像データを貼り付けます。
GUUPD IMAGE	イメージの内容を更新します。
GUUPD IMAGE FIXED	イメージの横幅、縦幅を更新します。
GUUPD IMAGE ROTATE	イメージの描画角度を更新します。
GUCREATE SCALE	指定したウィンドウにスケールを配置します。
GUUPD SCALE	スケールの内容を更新します。
GUCREATE SWITCH	指定したウィンドウにスイッチを配置します。
GUUPD SWITCH	スイッチの内容を更新します。
GUCREATE CALENDAR	指定したウィンドウにカレンダーを配置します。
GUUPD CALENDAR	カレンダーの年月日を更新します。
GUCREATE VIDEO	指定したウィンドウにビデオ描画を配置します。
GUUPD VIDEO	ビデオに対して制御を指示します。
GUSHOW	指定したウィンドウを表示します。



コマンド名	機能
GUMAIN LOOP	GUIアプリケーションを動かすためのGUIメインイベントループを動かします。
GUMAIN ONELOOP	GUIアプリケーションを動かすためのGUIメインイベントループを、1回だけ動かします。
GUMAIN QUIT	GUIメインイベントループを終了させます。
GUDSP	指定したGUI部品を表示あるいは非表示にします。
GUGET\$	指定したGUI部品の表示内容を取得します。
GUGET\$ VIDEO	指定したビデオ部品の情報を取得します。
GUSIZE	指定したGUI部品のサイズを変更します。
GUFONT	指定したGUI部品のフォント情報を変更します。
GUTIP	指定したGUI部品のツールチップを変更します。
GUDELETE	指定したウィンドウを削除します。
GUMOVE WINDOW	ウィンドウを移動します。
GUMSGBOX	メッセージボックスを表示します。
GUMOUSE	マウスの情報を取得します。
GUEVT ON	GUI部品のイベント割り込みを許可します。
GUEVT OFF	GUI部品のイベント割り込みを無効にします。
ON GUEVT CALL	GUI部品のイベントが発生した時に、呼び出されるサブルーチンを定義します。
GUDIALOGBOX OPEN\$	ファイルを開くためのダイアログボックスを表示し、選択したファイルパスを取得します。
GUDIALOGBOX SAVE\$	ファイルを保存するためのダイアログボックスを表示し、選択したファイルパスを取得します。
GUDIALOGBOX DIRECTORY\$	ディレクトリを選択するためのダイアログボックスを表示し、選択したディレクトリパスを取得します。
GUNOTIFY SEND	指定したメッセージを、デスクトップ通知します。
グラフに関する関数・命令	
GUCREATE BARCHART	指定したウィンドウに棒グラフを配置します。
GUADD BARCHART VALUE	棒グラフに値を追加します。
GUUPD BARCHART VALUE	棒グラフの値を更新します。
GUADD BARCHART TITLE	棒グラフにタイトルを追加します。
GUUPD BARCHART LABEL	棒グラフの軸ラベルを更新します。
GUADD BARCHART DATALABEL	棒グラフにデータラベルを追加します。
GUUPD BARCHART DATALABEL	棒グラフのデータラベルを更新します。
GUADD BARCHART RANGE	棒グラフに軸の範囲を追加します。
GUUPD BARCHART RANGE	棒グラフの軸の範囲を更新します。
GUUPD BARCHART GRID	棒グラフの目盛り線を更新します。
GUUPD BARCHART DRAW	棒グラフのグラフ描画を実行します。
GUDSP BARCHART	棒グラフを表示あるいは非表示にします。
GUCREATE LINEGRAPH	指定したウィンドウに折れ線グラフを配置します。
GUADD LINEGRAPH VALUE	折れ線グラフに値を追加します。
GUUPD LINEGRAPH VALUE	折れ線グラフの値を更新します。または、値のデータを全クリアします。
GUADD LINEGRAPH TITLE	折れ線グラフにタイトルを追加します。
GUUPD LINEGRAPH LABEL	折れ線グラフの軸ラベルを更新します。
GUADD LINEGRAPH DATALABEL	折れ線グラフにデータラベルを追加します。
GUUPD LINEGRAPH DATALABEL	折れ線グラフのデータラベルを更新します。
GUADD LINEGRAPH RANGE	折れ線グラフに軸の範囲を追加します。
GUUPD LINEGRAPH RANGE	折れ線グラフの各軸の範囲を更新します。

コマンド名	機能
GUADD LINEGRAPH GUAGE	折れ線グラフに任意の補助線を追加します。
GUUPD LINEGRAPH GUAGE	折れ線グラフに任意の補助線を上書き更新します。
GUUPD LINEGRAPH GRID	折れ線グラフの目盛り線のデータを更新します。
GUUPD LINEGRAPH DRAW	折れ線グラフのグラフ描画を実行します。
GUDSP LINEGRAPH	折れ線グラフを表示あるいは非表示にします。
GUCREATE PIECHART	指定したウィンドウに円グラフを配置します。
GUADD PIECHART VALUE	円グラフに値を追加します。
GUUPD PIECHART VALUE	円グラフの値を更新します。
GUADD PIECHART TITLE	円グラフにタイトルを追加します。
GUADD PIECHART DATALABEL	円グラフにデータラベルを追加します。
GUUPD PIECHART DATALABEL	円グラフのデータラベルを更新します。
GUUPD PIECHART DRAW	円グラフのグラフ描画を実行します。
GUDSP PIECHART	円グラフを表示あるいは非表示にします。
グラフィックに関する関数・命令	
GUCREATE PIC	指定したウィンドウに描画領域を配置します。
GUGCLS	描画領域に対して、クリアします。
GUGCOLOR	描画領域に対して、前景色および背景色を設定します。
GUGLINE	描画領域に対して、線分を描画します。
GUGRECTANGLE	描画領域に対して、矩形を描画します。
GUGCIRCLE	描画領域に対して、円を描画します。
GUGARC	描画領域に対して、円弧を描画します。
GUGPSET	描画領域に対して、点を描画します。
GUGDRAWTEXT	描画領域に対して、テキスト文字を描画します。
GUGDRAWFILE	描画領域に対して、画像ファイルの画像を描画します。
GUGSAVEFILE	描画領域の全てないしは指定範囲の画像を、画像ファイルに保存します。
GUGDATAGET	描画領域の全てないしは指定範囲の画像情報を、配列データに取得します。
GUGDATASET	描画領域に対して、画像情報の配列データを元に、画像データとして直接書き込みます。
GUGDRAWBLIT	描画領域の特定範囲の画像を、別の描画領域に転送して貼り付けます。
画像処理に関する関数・命令	
IMG_READ	ファイルから画像を読み込みます。
IMG_WRITE	指定したファイルに画像を保存します。
IMG2GUG_DATA	IMG_READで得た画像データを、GUGDATASETで使用する画像データに変換します。
GUG2IMG_DATA	GUGDATAGETで得た画像データを、IMG_WRITE等で使用する画像データに変換します。
IMG_CONVERT_COLOR	画像の色空間を変換します。
IMG_RESIZE	画像の大きさを変換します。
IMG_BINARIZE	画像を2値化します。
IMG_BINARIZE_AUTO	画像を2値化します。しきい値は入力画像から自動で求めます。
IMG_FIND_CONTOURS	2値化画像から輪郭を抽出します。
IMG_ERODE	画像に対して収縮処理を実行します。
IMG_DILATE	画像に対して膨張処理を実行します。
IMG_MEAN_FILTER	平均フィルタを使用して画像を平滑化します。
IMG_GAUSSIAN_FILTER	ガウシアンフィルタを使用して画像を平滑化します。

コマンド名	機能
IMG_BILATERAL_FILTER	バイラテラルフィルタを使用して画像を平滑化します。
IMG_SOBEL_FILTER	ソーベルフィルタを使用して画像からX方向またはY方向のエッジを抽出します。
IMG_LAPLACIAN_FILTER	ラプラシアンフィルタを使用して画像からエッジを抽出します。
IMG_SHARP_FILTER	鮮鋭化フィルタを使用して、エッジが強調された画像を得ます。
IMG_CANNY_EDGE	キャニーのエッジ検出を実行します。
IMG_GET_ROTATION_MATRIX	アフィン変換で画像を回転するために使用する変換行列を求めます。
IMG_WARP_AFFINE	画像に対して与えられた変換行列でアフィン変換を行います。
IMG_MATCH_TEMPLATE	入力画像とテンプレート画像の相関関数を計算します。
IMG_PHASE_CORRELATION	位相限定相関法を使用して、与えられた2枚の画像の位置ずれを推定します。
IMG_WARP_POLAR	画像に対して極座標変換を行います。

3.2 GUIに関する共通項目

GUIの部品生成コマンドには共通した項目があります。  
各コマンドには詳細を記載していないため、こちらの説明を参照してください。

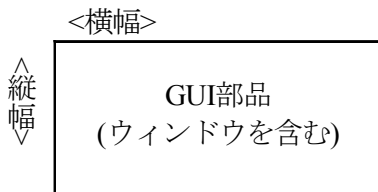
3.2.1 フォント設定

共通項目

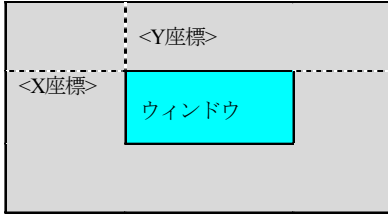
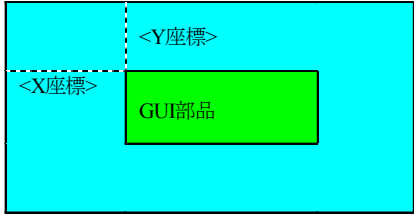
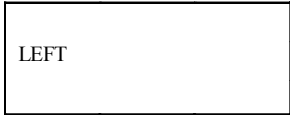

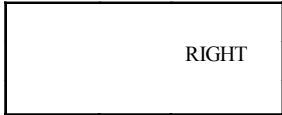
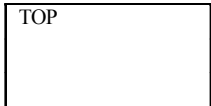


機能	GUI部品に表示される文字列のフォントを指定します。																																													
書式	<p>"&lt;フォント名&gt;, &lt;斜体&gt;, &lt;太字&gt;, &lt;サイズ&gt;, &lt;文字色&gt;, &lt;背景色&gt;" 1つの文字列でまとめてください。</p> <p>&lt;フォント名&gt; 表示する文字列のフォントを指定します。以下フォントを指定することができます。 フォント名の左に記載されている番号を入力してください。</p> <table><tr><td>1: Bitstream Charter</td><td>2: Century Shoolbook L</td><td>3: Courier 10 Pitch</td></tr><tr><td>4: DejaVu Sans</td><td>5: DejaVu Sans Mono</td><td>6: DejaVu Serif</td></tr><tr><td>7: Dingbats</td><td>8: FreeMono</td><td>9: FreeSans</td></tr><tr><td>10: FreeSerif</td><td>11: Gentium</td><td>12: Gentium Basic</td></tr><tr><td>13: Gentium Book Basic</td><td>14: GentiumAlt</td><td>15: IPA Pゴシック</td></tr><tr><td>16: IPA P明朝</td><td>17: IPAゴシック</td><td>18: IPA明朝</td></tr><tr><td>19: Liberation Mono</td><td>20: Liberation Sans</td><td>21: Liberation Serif</td></tr><tr><td>22: Nimbus Mono L</td><td>23: Nimbus Roman No9 L</td><td>24: Nimbus Sans L</td></tr><tr><td>25: Standard Symbols L</td><td>26: Symbol</td><td>27: URW Bookman L</td></tr><tr><td>28: URW Chancery L</td><td>29: URW Gothic L</td><td>30: URW Palladio L</td></tr><tr><td>31: VL Pゴシック</td><td>32: VL ゴシック</td><td>33: cmex10</td></tr><tr><td>34: cmmi10</td><td>35: cmr10</td><td>36: cmsy10</td></tr><tr><td>37: esint10</td><td>38: eufm10</td><td>39: msam10</td></tr><tr><td>40: unifont</td><td>41: wasy10</td><td>42: さざなみゴシック</td></tr><tr><td>43: さざなみ明朝</td><td></td><td></td></tr></table> <p>&lt;斜体&gt;, &lt;太字&gt; 表示する文字列を斜体、太字にする設定ができます。 0:設定無し 1:設定あり</p> <p>&lt;サイズ&gt; 表示する文字のサイズを指定します。</p>	1: Bitstream Charter	2: Century Shoolbook L	3: Courier 10 Pitch	4: DejaVu Sans	5: DejaVu Sans Mono	6: DejaVu Serif	7: Dingbats	8: FreeMono	9: FreeSans	10: FreeSerif	11: Gentium	12: Gentium Basic	13: Gentium Book Basic	14: GentiumAlt	15: IPA Pゴシック	16: IPA P明朝	17: IPAゴシック	18: IPA明朝	19: Liberation Mono	20: Liberation Sans	21: Liberation Serif	22: Nimbus Mono L	23: Nimbus Roman No9 L	24: Nimbus Sans L	25: Standard Symbols L	26: Symbol	27: URW Bookman L	28: URW Chancery L	29: URW Gothic L	30: URW Palladio L	31: VL Pゴシック	32: VL ゴシック	33: cmex10	34: cmmi10	35: cmr10	36: cmsy10	37: esint10	38: eufm10	39: msam10	40: unifont	41: wasy10	42: さざなみゴシック	43: さざなみ明朝		
1: Bitstream Charter	2: Century Shoolbook L	3: Courier 10 Pitch																																												
4: DejaVu Sans	5: DejaVu Sans Mono	6: DejaVu Serif																																												
7: Dingbats	8: FreeMono	9: FreeSans																																												
10: FreeSerif	11: Gentium	12: Gentium Basic																																												
13: Gentium Book Basic	14: GentiumAlt	15: IPA Pゴシック																																												
16: IPA P明朝	17: IPAゴシック	18: IPA明朝																																												
19: Liberation Mono	20: Liberation Sans	21: Liberation Serif																																												
22: Nimbus Mono L	23: Nimbus Roman No9 L	24: Nimbus Sans L																																												
25: Standard Symbols L	26: Symbol	27: URW Bookman L																																												
28: URW Chancery L	29: URW Gothic L	30: URW Palladio L																																												
31: VL Pゴシック	32: VL ゴシック	33: cmex10																																												
34: cmmi10	35: cmr10	36: cmsy10																																												
37: esint10	38: eufm10	39: msam10																																												
40: unifont	41: wasy10	42: さざなみゴシック																																												
43: さざなみ明朝																																														

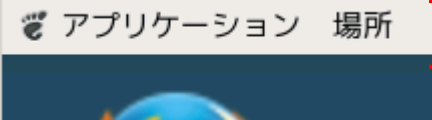
	<p>〈文字色〉, 〈背景色〉 表示する文字色、背景色を指定します。</p> <p>文字色、背景色の指定方法は以下の通りです。</p> <p>①色名による指定 色名で指定できる色は以下の色です。 色名の左に記載されている番号を入力してください。</p> <table><tr><td>1:BLACK</td><td>2:RED</td><td>3:GREEN</td><td>4:YELLOW</td></tr><tr><td>5:BLUE</td><td>6:PURPLE</td><td>7:CYAN</td><td>8:WHITE</td></tr><tr><td>9:GRAY</td><td>10:BROWN</td><td>11:ORANGE</td><td>12:PINK</td></tr><tr><td>13:VIOLET</td><td>14:GOLD</td><td>15:MAGENTA</td><td>16:SNOW</td></tr><tr><td>17:DARKBLUE</td><td>18:DARKCYAN</td><td>19:DARKGRAY</td><td>20:DARKGREEN</td></tr><tr><td>21:DARKMAGENTA</td><td>22:DARKORANGE</td><td>23:DARKRED</td><td>24:DARKSEAGREEN</td></tr><tr><td>25:DARKVIOLET</td><td>26:LIGHTBLUE</td><td>27:LIGHTCYAN</td><td>28:LIGHTGRAY</td></tr><tr><td>29:LIGHTGREEN</td><td>30:LIGHTPINK</td><td>31:LIGHTSEAGREEN</td><td>32:LIGHTSKYBLUE</td></tr><tr><td>33:LIGHTYELLOW</td><td>34:LIMEGREEN</td><td>35:DEEPPINK</td><td>36:DEEPSKYBLUE</td></tr><tr><td>37:NAVYBLUE</td><td>38:SKYBLUE</td><td>39:SEAGREEN</td><td>40:YELLOWGREEN</td></tr></table> <p>②カラーコードによる指定 カラーコードによって、詳細な色の指定をします。 先頭に#を付けた6桁の16進数を指定します。 (例) #FF3366</p> <p>デフォルト値 〈フォント名〉 = 32 (VL ゴシック) 〈斜体〉 = 0 (設定無し) 〈太字〉 = 0 (設定無し) 〈サイズ〉 = 10 〈文字色〉 = 1 (BLACK) 〈背景色〉 = 8 (WHITE)</p>	1:BLACK	2:RED	3:GREEN	4:YELLOW	5:BLUE	6:PURPLE	7:CYAN	8:WHITE	9:GRAY	10:BROWN	11:ORANGE	12:PINK	13:VIOLET	14:GOLD	15:MAGENTA	16:SNOW	17:DARKBLUE	18:DARKCYAN	19:DARKGRAY	20:DARKGREEN	21:DARKMAGENTA	22:DARKORANGE	23:DARKRED	24:DARKSEAGREEN	25:DARKVIOLET	26:LIGHTBLUE	27:LIGHTCYAN	28:LIGHTGRAY	29:LIGHTGREEN	30:LIGHTPINK	31:LIGHTSEAGREEN	32:LIGHTSKYBLUE	33:LIGHTYELLOW	34:LIMEGREEN	35:DEEPPINK	36:DEEPSKYBLUE	37:NAVYBLUE	38:SKYBLUE	39:SEAGREEN	40:YELLOWGREEN
1:BLACK	2:RED	3:GREEN	4:YELLOW																																						
5:BLUE	6:PURPLE	7:CYAN	8:WHITE																																						
9:GRAY	10:BROWN	11:ORANGE	12:PINK																																						
13:VIOLET	14:GOLD	15:MAGENTA	16:SNOW																																						
17:DARKBLUE	18:DARKCYAN	19:DARKGRAY	20:DARKGREEN																																						
21:DARKMAGENTA	22:DARKORANGE	23:DARKRED	24:DARKSEAGREEN																																						
25:DARKVIOLET	26:LIGHTBLUE	27:LIGHTCYAN	28:LIGHTGRAY																																						
29:LIGHTGREEN	30:LIGHTPINK	31:LIGHTSEAGREEN	32:LIGHTSKYBLUE																																						
33:LIGHTYELLOW	34:LIMEGREEN	35:DEEPPINK	36:DEEPSKYBLUE																																						
37:NAVYBLUE	38:SKYBLUE	39:SEAGREEN	40:YELLOWGREEN																																						
備 考	<ul style="list-style-type: none"><li>・サイズは、正の値を指定してください。</li><li>・サイズは、ポイント単位で管理されます。ポイント単位の値をピクセル単位に換算するには、以下の式で求める事ができます。 ピクセル単位の値 = ポイント単位の値 × ( 96.0 ÷ 72.0 )</li><li>・OSは、指定された通りのフォントサイズで表示しようとはしますが、その表示結果はOSおよびグラフィックドライバに依存します。</li></ul>																																								
注 意	<ul style="list-style-type: none"><li>・GUI部品のサイズに対して、あまりに大きいフォントサイズを指定すると、OSは何とかテキストを描画できるように、GUI部品の大きさを自動的に変更したり、テキストが描画し切れず、GUI部品の外にテキストを描画したりします。 これを避けるには、GUI部品のサイズがフォントサイズを上回るように値を調整してください。</li></ul>																																								
使用例1	<pre>GUCREATE WINDOW 1 FONT\$="32, 1, 1, 24, 2, 4" GUCREATE BUTTON 1, 10, "SAMPLE", FONT\$, SIZE\$, POSI\$, TIP\$ GUSHOW 1</pre> <p>変数FONT\$に 〈フォント名〉 VL ゴシック 〈斜体〉あり, 〈太字〉あり 〈サイズ〉 24 〈文字色〉 RED 〈背景色〉 YELLOW を設定します。</p>																																								
使用例2	<pre>GUCREATE WINDOW 1 GUCREATE BUTTON 1, 10, "SAMPLE", "32, 1, 1, 24, 2, 4", SIZE\$, POSI\$, TIP\$ GUSHOW 1</pre> <p>文字列は変数へ代入せず、直接引数として指定する事が出来ます。 使用例1と同じ事を直接引数に指定します。</p>																																								

## 3.2.2 サイズ設定

共通項目	
機能書式	<p>GUI部品に表示されるGUI部品の大きさを指定します。</p> <p>"&lt;横幅&gt;, &lt;縦幅&gt;" 1つの文字列でまとめてください。</p> <p>&lt;横幅&gt;, &lt;縦幅&gt; GUI部品の横幅と縦幅を指定します。</p> <ul style="list-style-type: none"> <li>・ウィンドウの場合 デフォルト値:&lt;横幅&gt;, &lt;縦幅&gt;ともにモニタ解像度の1/4</li> <li>・グラフ関係のGUI部品の場合 デフォルト値:&lt;横幅&gt;=100, &lt;縦幅&gt;=100</li> <li>・その他GUI部品の場合 デフォルト値:&lt;横幅&gt;=100, &lt;縦幅&gt;=20</li> </ul> <div style="text-align: center;"> <p>&lt;横幅&gt;</p>  <p>GUI部品 (ウィンドウを含む)</p> </div>
備考	<ul style="list-style-type: none"> <li>・横幅、縦幅は、正の整数値を与えてください。</li> <li>・GUI部品の大きさが、指定した値通りになるとは限りません。 その挙動はOSに依存します。例えば、1x1の大きさのボタンを作成しようとしても、OSは表示可能なサイズに自動調節します。</li> </ul>
注意	<ul style="list-style-type: none"> <li>・GUI部品のサイズに対して、あまりに大きいフォントサイズを指定すると、OSは何とかテキストを描画できるようにGUI部品の大きさを自動的に変更したり、テキストが描画し切れず、GUI部品の外にテキストを描画したりします。 これを避けるには、GUI部品のサイズがフォントサイズを上回るように、値を調整してください。</li> </ul>
使用例1	<pre>GUCREATE WINDOW 1 SIZE\$="100, 50" GUCREATE BUTTON 1, 10, "SAMPLE", FONT\$, SIZE\$, POSI\$, TIP\$ GUSHOW 1</pre> <p>変数SIZE\$に &lt;横幅&gt;100 &lt;縦幅&gt; 50 を設定します。</p>
使用例2	<pre>GUCREATE WINDOW 1 GUCREATE BUTTON 1, 10, "SAMPLE", FONT\$, "100, 50", POSI\$, TIP\$ GUSHOW 1</pre> <p>文字列は変数へ代入せず、直接引数として指定する事が出来ます。 使用例1と同じ事を直接引数に指定します。</p>

## 3.2.3 ポジション設定

共通項目	
機能書式	<p>GUI部品に表示されるGUI部品の表示場所を指定します。</p> <p>"&lt;X座標&gt;, &lt;Y座標&gt; [, &lt;横ポジション&gt;, &lt;縦ポジション&gt;]" 1つの文字列でまとめてください。</p> <p>&lt;X座標&gt;, &lt;Y座標&gt; ウィンドウ作成時に用いる場合と、GUI部品作成時に用いる場合で用途が異なります。</p> <p>ウィンドウの場合 画面上でのウィンドウの位置を指定します。</p> <p>GUI部品の場合 ウィンドウ上での位置を指定します。</p> <p>デフォルト値 &lt;X座標&gt;=0 &lt;Y座標&gt;=0</p> <div style="display: flex; justify-content: space-around; align-items: flex-start;"> <div style="text-align: center;"> <p>ウィンドウの場合 画面</p>  </div> <div style="text-align: center;"> <p>GUI部品の場合 ウィンドウ</p>  </div> </div> <p>&lt;横ポジション&gt; GUI部品上の文字列位置を指定します。ウィンドウ作成時には使用しません。 LEFT: 左揃え (デフォルト値) CENTER: 中央揃え RIGHT: 右揃え</p> <div style="display: flex; justify-content: space-around; align-items: flex-start;"> <div style="text-align: center;"> <p>LEFTの場合 GUI部品</p>  </div> <div style="text-align: center;"> <p>CENTERの場合 GUI部品</p>  </div> <div style="text-align: center;"> <p>RIGHTの場合 GUI部品</p>  </div> </div> <p>&lt;縦ポジション&gt; GUI部品上の文字列位置を指定します。ウィンドウ作成時には使用しません。 TOP: 上揃え (デフォルト値) MIDDLE: 上下中央揃え BOTTOM: 下揃え</p> <div style="display: flex; justify-content: space-around; align-items: flex-start;"> <div style="text-align: center;"> <p>TOPの場合 GUI部品</p>  </div> <div style="text-align: center;"> <p>MIDDLEの場合 GUI部品</p>  </div> <div style="text-align: center;"> <p>BOTTOMの場合 GUI部品</p>  </div> </div>

備 考	<ul style="list-style-type: none"> <li>・X座標、Y座標は、正の整数値を与えてください。</li> <li>・X、Y座標の指定に対して、実際にその位置に表示されるかは、実行時の状況に依存します。例えば、デスクトップ画面の上部パネルのバーがある場合、そのバーの縦幅分はLinuxのシステムが占有される為、配置できません。（自動的に下に配置される）</li> </ul>  <p>上部パネルのバー部分</p> <p>上部パネルのプロパティ設定の「自動的に隠す」などでバーを隠すと、配置が可能になります。</p> <ul style="list-style-type: none"> <li>・上と同様に、モニタの解像度以上の値を指定した場合、OSは画面内に表示できるように表示位置を自動調整します。</li> </ul>
使用例1	<pre>GUCREATE WINDOW 1 POSI\$="100, 200, CENTER, MIDDLE" GUCREATE BUTTON 1, 10, "SAMPLE", FONT\$, SIZE\$, POSI\$, TIP\$ GUSHOW 1</pre> <p>変数POSI\$に〈X座標〉 100 〈Y座標〉 200 〈横ポジション〉 CENTER 〈縦ポジション〉 MIDDLE を設定します。</p>
使用例2	<pre>GUCREATE WINDOW 1 GUCREATE BUTTON 1, 10, "SAMPLE", FONT\$, SIZE\$, "100, 200, CENTER, MIDDLE", TIP\$ GUSHOW 1</pre> <p>文字列は変数へ代入せず、直接引数として指定する事が出来ます。 使用例1と同じ事を直接引数に指定します。</p>

### 3.2.4 ツールチップ設定

共通項目	
機 能	マウスポインタがGUI部品上に来た際に、表示されるツールチップの内容を指定します。
書 式	<p>“〈表示内容〉”</p> <p>〈表示内容〉 表示したい文字列を指定します。 デフォルト値：空文字列</p>
備 考	・ ツールチップが見難くなるので、大量の文字列を与えないでください。
使用例1	<pre>GUCREATE WINDOW 1 TIP\$="TipData" GUCREATE BUTTON 1, 10, "SAMPLE", FONT\$, SIZE\$, POSI\$, TIP\$ GUSHOW 1</pre> <p>ボタンに対して〈表示内容〉 TipDataを設定します。</p>
使用例2	<pre>GUCREATE WINDOW 1 GUCREATE BUTTON 1, 10, "SAMPLE", FONT\$, SIZE\$, POSI\$, "TipData" GUSHOW 1</pre> <p>文字列は変数へ代入せず、直接引数として指定する事が出来ます。 使用例1と同じ事を直接引数に指定します。</p>



### 3.2.5 共通項目の省略について

フォント設定以降の引数は省略可能ですが、一つでも引数を指定する場合、全ての引数間のカンマが必要です。

<例>フォント設定以降を全て省略する場合

```
GUCREATE WINDOW 1, "ウィンドウ", 1
```

<例>フォント設定だけを指定する場合

```
GUCREATE WINDOW 1, "ウィンドウ", 1, FONT$,,,
```

<例>サイズ設定だけを指定する場合

```
GUCREATE WINDOW 1, "ウィンドウ", 1,, SIZE$,,,
```

引数を省略した箇所は、デフォルトが使用されます。

## 3.3 GUIに関する関数・命令

### 3.3.1 コマンド個別説明

#### 3.3.1.1 OPTION GUI

命令	
機 能	GUIアプリケーションを実行することを宣言します。
書 式	OPTION GUI
備 考	GUIコマンドを組み合わせるGUIアプリケーションを作成する場合、プログラムの先頭部分に、この記述を付加してください。 OSは、内部でGUIを使用する為の初期化を実施します。
使用例	OPTION GUI  GUCREATE WINDOW 1 GUSHOW 1  GUIコマンドを呼び出す際には、必ず指定します。

## 3.3.1.2 GUCREATE WINDOW

命令		
機能	ウィンドウを作成します。	
書式	GUCREATE WINDOW <①ウィンドウ番号> [, <②タイトル>, <③最大化・最小化・閉じるボタン表示設定> [, [ <④フォント設定> ], [ <⑤サイズ設定> ], [ <⑥ポジション設定> ], [ <⑦ツールチップ設定> ] ] ]	
	※フォント設定以降の引数を省略する場合、各引数間のカンマは必要です。	
パラメータ	①	<div> <div>&lt;ウィンドウ番号&gt;</div> <div>数値</div> </div> 作成するウィンドウの番号を1～15の範囲で指定します。
	②	<div> <div>&lt;タイトル&gt;</div> <div>文字列</div> </div> 指定した文字列をウィンドウのタイトルバーに表示します。
	③	<div> <div>&lt;最大化・最小化・閉じるボタン表示設定&gt;</div> <div>数値</div> </div> ウィンドウの最大化・最小化・閉じるボタン表示設定を指定します。 0：非表示 1：表示(デフォルト値) 2：閉じるボタンのみ非表示 ※上記の数値以外の値を指定した際の挙動は未定義です。
	④	<div> <div>&lt;フォント設定&gt;</div> <div>文字列</div> </div> GUI部品に表示される文字列のフォントを指定します。(参照：P. 20)
	⑤	<div> <div>&lt;サイズ設定&gt;</div> <div>文字列</div> </div> GUI部品に表示されるGUI部品の大きさを指定します。(参照：P. 22)
	⑥	<div> <div>&lt;ポジション設定&gt;</div> <div>文字列</div> </div> GUI部品に表示されるGUI部品の表示場所を指定します。(参照：P. 23)
	⑦	<div> <div>&lt;ツールチップ設定&gt;</div> <div>文字列</div> </div> マウスポインタがGUI部品上にきた際に、表示されるツールチップの内容を指定します。(参照：P. 24)
	<b>備考</b> <ul style="list-style-type: none"> <li>・ウィンドウをGUDELETEコマンドやマウス等で閉じた時、ウィンドウ番号を使って生成した、その他のGUI部品(GUCREATE BUTTONなど)も、自動的に破棄されます。</li> </ul>	
	<b>使用例1</b> <pre>GUCREATE WINDOW 1 GUSHOW 1</pre> <ウィンドウ番号> 1 でウィンドウを作成します。	
	<b>使用例2</b> <pre>GUCREATE WINDOW 1, "SAMPLE", 0 GUSHOW 1</pre> <ウィンドウ番号> 1 のウィンドウに<タイトル> SAMPLE<最大化・最小化・閉じるボタン> 非表示で作成します。	
	<b>使用例3</b> <pre>GUCREATE WINDOW 1, "SAMPLE", 0, FONT\$, SIZE\$, POSI\$, TIP\$ GUSHOW 1</pre> <ウィンドウ番号> 1 のウィンドウに<タイトル> SAMPLE<最大化・最小化・閉じるボタン> 非表示で作成します。 共通項目でフォントやサイズ、位置、ツールチップの表示内容を設定しています。	

## 3.3.1.3 GUCREATE BUTTON

命令		
機能	指定したウィンドウにボタンを配置します。	
書式	<p>GUCREATE BUTTON &lt;①ウィンドウ番号&gt;, &lt;②ボタン番号&gt;, &lt;③文字列&gt; [, [ &lt;④フォント設定&gt; ], [ &lt;⑤サイズ設定&gt; ], [ &lt;⑥ポジション設定&gt; ], [ &lt;⑦ツールチップ設定&gt; ] ]</p> <p>※ フォント設定以降の引数を省略する場合、各引数間のカンマは必要です。</p> <div style="display: flex; align-items: center; justify-content: center;"> <div style="border: 1px solid black; padding: 10px; margin: 10px;"> <div style="color: blue; font-weight: bold; margin-bottom: 5px;">ウィンドウ</div> <div style="border: 1px solid black; padding: 5px; margin: 5px;"> <div style="color: blue; font-weight: bold; margin-bottom: 5px;">ボタン</div> <div style="border: 1px solid black; padding: 2px; display: inline-block;">SAMPLE</div> </div> </div> </div>	
パラメータ	①	<ウィンドウ番号> 数値
	ボタンを配置するウィンドウの番号を指定します。	
	②	<ボタン番号> 数値
	作成するボタンの番号を指定します。 1～10000の整数値を指定してください。	
	③	<文字列> 文字列
	ボタン内に表示したい文字列を指定します。	
	④	<フォント設定> 文字列
	GUI部品に表示される文字列のフォントを指定します。(参照：P. 20)	
備考	⑤	<サイズ設定> 文字列
	GUI部品に表示されるGUI部品の大きさを指定します。(参照：P. 22)	
	⑥	<ポジション設定> 文字列
	GUI部品に表示されるGUI部品の表示場所を指定します。(参照：P. 23)	
	⑦	<ツールチップ設定> 文字列
	マウスポインタがGUI部品上にきた際に、表示されるツールチップの内容を指定します。(参照：P. 24)	
	<ul style="list-style-type: none"> <li>ボタンの文字列に、「_ (アンダーバー)」の後に英数文字を設定すると、ALT+&lt;指定した英数文字&gt; を押すことで、ボタンを押す事ができます。 これを、アクセラレータキーと呼びます。</li> <li>ボタンの文字列に、単に「_」 (アンダーバー)を表記したい場合は、「__」 と2回続けてください。</li> </ul>	
使用例	使用例1	<p>GUCREATE WINDOW 1 GUCREATE BUTTON 1, 10, "SAMPLE" GUSHOW 1</p> <p>&lt;ウィンドウ番号&gt; 1 のウィンドウに &lt;ボタン番号&gt; 10 &lt;文字列&gt; SAMPLE のボタンを配置します。</p>
	使用例2	<p>GUCREATE WINDOW 1 GUCREATE BUTTON 1, 10, "SAMPLE", FONT\$, SIZE\$, POSI\$, TIP\$ GUSHOW 1</p> <p>&lt;ウィンドウ番号&gt; 1 のウィンドウに &lt;ボタン番号&gt; 10 &lt;文字列&gt; SAMPLEのボタンを配置します。 共通項目でボタンの色や配置座標、サイズ、ツールチップでの表示内容を設定しています。</p>

3.3.1.4 GUUPD BUTTON

命令			
機能	ボタンの表示内容を更新します。		
書式	GUUPD BUTTON <①ボタン番号>, <②文字列>		
パラメータ	①	<ボタン番号>	数値
	表示内容を更新するボタンの番号を指定します。		
	②	<文字列>	文字列
	更新を行う文字列を指定します。		
使用例	GUCREATE WINDOW 1 GUCREATE BUTTON 1, 10, "TEST" GUUPD BUTTON 10, "SAMPLE" GUSHOW 1  <部品番号> 10 のボタンの内容を<文字列> SAMPLE に更新します。		

### 3.3.1.5 GUCREATE LABEL

命令		
機能	指定したウィンドウにラベルを配置します。	
書式	<p>GUCREATE LABEL &lt;①ウィンドウ番号&gt;, &lt;②ラベル番号&gt;, &lt;③文字列&gt; [, [ &lt;④フォント設定&gt; ], [ &lt;⑤サイズ設定&gt; ], [ &lt;⑥ポジション設定&gt; ], [ &lt;⑦ツールチップ設定&gt; ] ]</p> <p>※ フォント設定以降の引数を省略する場合、各引数間のカンマは必要です。</p> <div style="display: flex; align-items: center;"> <div style="border: 1px solid black; padding: 10px; margin: 10px;"> <p style="color: blue;">ウィンドウ</p> <div style="border: 1px solid black; padding: 10px; margin: 10px;"> <p style="color: blue;">ラベル</p> <p>SAMPLE</p> </div> </div> </div>	
パラメータ	①	<p style="text-align: center;">&lt;ウィンドウ番号&gt;</p> <p>ラベルを配置するウィンドウの番号を指定します。</p>
	②	<p style="text-align: center;">&lt;ラベル番号&gt;</p> <p>作成するラベルの番号を指定します。 1～10000の値を指定してください。</p>
	③	<p style="text-align: center;">&lt;文字列&gt;</p> <p>ラベル内に表示したい文字列を指定します。</p>
	④	<p style="text-align: center;">&lt;フォント設定&gt;</p> <p>GUI部品に表示される文字列のフォントを指定します。(参照：P. 20)</p>
	⑤	<p style="text-align: center;">&lt;サイズ設定&gt;</p> <p>GUI部品に表示されるGUI部品の大きさを指定します。(参照：P. 22)</p>
	⑥	<p style="text-align: center;">&lt;ポジション設定&gt;</p> <p>GUI部品に表示されるGUI部品の表示場所を指定します。(参照：P. 23)</p>
	⑦	<p style="text-align: center;">&lt;ツールチップ設定&gt;</p> <p>マウスポインタがGUI部品上に来た際に、表示されるツールチップの内容を指定します。(参照：P. 24)</p>
使用例1	<p>GUCREATE WINDOW 1 GUCREATE LABEL 1, 10, "SAMPLE" GUSHOW 1</p> <p>&lt;ウィンドウ番号&gt; 1 のウィンドウに&lt;ラベル番号&gt; 10 &lt;文字列&gt; SAMPLE のラベルを配置します。</p>	
使用例2	<p>GUCREATE WINDOW 1 GUCREATE LABEL 1, 10, "SAMPLE", FONT\$, SIZE\$, POSI\$, TIP\$ GUSHOW 1</p> <p>&lt;ウィンドウ番号&gt; 1 のウィンドウに&lt;ラベル番号&gt; 10 &lt;文字列&gt; SAMPLE のラベルを配置します。 共通項目でフォントやサイズ、位置、ツールチップの表示内容を設定しています。</p>	

3.3.1.6 GUUPD LABEL

命令			
機能	ラベルの表示内容を更新します。		
書式	GUUPD LABEL <①ラベル番号>, <②文字列>		
パラメータ	①	<ラベル番号>	数値
	表示内容を更新するラベルの番号を指定します。		
	②	<文字列>	文字列
	更新を行う文字列を指定します。		
使用例	GUCREATE WINDOW 1 GUCREATE LABEL 1, 10, "TEST" GUUPD LABEL 10, "SAMPLE" GUSHOW 1  <部品番号> 10 のラベルの内容を<文字列> SAMPLE に更新します。		

## 3.3.1.7 GUCREATE TEXTBOX

命令			
機能 書式	指定したウィンドウにテキストボックスを配置します。		
	<p>GUCREATE TEXTBOX &lt;①ウィンドウ番号&gt;, &lt;②テキストボックス番号&gt;, &lt;③文字列&gt; [, [ &lt;④フォント設定&gt; ], [ &lt;⑤サイズ設定&gt; ], [ &lt;⑥ポジション設定&gt; ], [ &lt;⑦ツールチップ設定&gt; ] ]</p> <p>※ フォント設定以降の引数を省略する場合、各引数間のカンマは必要です。</p> <div style="display: flex; align-items: center;"> <div style="margin-right: 10px;">ウィンドウ</div> <div style="border: 1px solid black; padding: 10px; text-align: center;"> <div style="margin-bottom: 5px;">テキストボックス</div> <div style="border: 1px solid black; width: 100px; height: 20px; margin: 0 auto;"></div> </div> </div>		
パラ メータ	①	<ウィンドウ番号>	数値
	テキストボックスを配置するウィンドウの番号を指定します。		
	②	<テキストボックス番号>	数値
	作成するテキストボックスの番号を指定します。 1～10000の値を指定してください。		
	③	<文字列>	文字列
	テキストボックスへ初期状態として入力する文字列を指定します。		
	④	<フォント設定>	文字列
	GUI部品に表示される文字列のフォントを指定します。(参照：P. 20)		
	⑤	<サイズ設定>	文字列
	GUI部品に表示されるGUI部品の大きさを指定します。(参照：P. 22)		
	⑥	<ポジション設定>	文字列
	GUI部品に表示されるGUI部品の表示場所を指定します。(参照：P. 23)		
	⑦	<ツールチップ設定>	文字列
	マウスポインタがGUI部品上に来た際に、表示されるツールチップの内容を指定します。 (参照：P. 24)		
使用例1	<p>GUCREATE WINDOW 1 GUCREATE TEXTBOX 1, 10, "SAMPLE" GUSHOW 1</p> <p>&lt;ウィンドウ番号&gt; 1 のウィンドウに &lt;テキストボックス番号&gt; 10 &lt;文字列&gt; SAMPLEのテキストボックスを配置します。</p>		
使用例2	<p>GUCREATE WINDOW 1 GUCREATE TXETBOX 1, 10, "SAMPLE", FONT\$, SIZE\$, POSI\$, TIP\$ GUSHOW 1</p> <p>&lt;ウィンドウ番号&gt; 1 のウィンドウに &lt;テキストボックス番号&gt; 10 &lt;文字列&gt; SAMPLEのテキストボックスを配置します。 共通項目でフォントやサイズ、位置、ツールチップの表示内容を設定しています。</p>		

3.3.1.8 GUUPD TEXTBOX

命令			
機能	テキストボックスの表示内容を更新します。		
書式	GUUPD TEXTBOX <①テキストボックス番号>, <②文字列>		
パラメータ	①	<テキストボックス番号>	数値
	表示内容を更新するテキストボックスの番号を指定します。		
	②	<文字列>	文字列
	更新を行う文字列を指定します。		
使用例	GUCREATE WINDOW 1 GUCREATE TEXTBOX 1, 10, "TEST" GUUPD TEXTBOX 10, "SAMPLE" GUSHOW 1  <部品番号> 10 のテキストボックスの内容を<文字列> SAMPLE に更新します。		



## 3.3.1.9 GUCREATE TEXTAREA

命令			
機能	指定したウィンドウにテキストエリアを配置します。 テキストボックスとテキストエリアの違いは、テキストエリアは自動でスクロールが付きます。		
書式	<p>GUCREATE TEXTAREA &lt;①ウィンドウ番号&gt;, &lt;②テキストエリア番号&gt;, &lt;③文字列&gt; [, [, &lt;④フォント設定&gt; ], [, &lt;⑤サイズ設定&gt; ], [, &lt;⑥ポジション設定&gt; ], [, &lt;⑦ツールチップ設定&gt; ] ]</p> <p>※ フォント設定以降の引数を省略する場合、各引数間のカンマは必要です。</p> <div style="display: flex; align-items: center;"> <div style="margin-right: 20px;">ウィンドウ</div> <div style="border: 1px solid black; padding: 10px; text-align: center;"> <div style="margin-bottom: 5px;">テキストエリア</div> <div style="border: 1px solid black; width: 150px; height: 60px; margin: 0 auto; position: relative;"> <div style="position: absolute; right: -20px; top: 0; bottom: 0; text-align: center;"> <div style="border-top: 1px solid black; border-bottom: 1px solid black; width: 10px; height: 10px; margin: 2px auto;">△</div> <div style="border-top: 1px solid black; border-bottom: 1px solid black; width: 10px; height: 10px; margin: 2px auto;">▽</div> </div> </div> </div> </div>		
パラメータ	①	<ウィンドウ番号>	数値
	テキストエリアを配置するウィンドウの番号を指定します。		
	②	<テキストエリア番号>	数値
	作成するテキストエリアの番号を指定します。 1～10000の値を指定してください。		
	③	<文字列>	文字列
	テキストエリアへ初期状態として入力する文字列を指定します。		
	④	<フォント設定>	文字列
	GUI部品に表示される文字列のフォントを指定します。(参照：P. 20)		
	⑤	<サイズ設定>	文字列
	GUI部品に表示されるGUI部品の大きさを指定します。(参照：P. 22)		
	⑥	<ポジション設定>	文字列
	GUI部品に表示されるGUI部品の表示場所を指定します。(参照：P. 23)		
	⑦	<ツールチップ設定>	文字列
	マウスポインタがGUI部品上に来た際に、表示されるツールチップの内容を指定します。 (参照：P. 24)		
使用例1	<p>GUCREATE WINDOW 1 GUCREATE TEXTAREA 1, 10, "SAMPLE" GUSHOW 1</p> <p>&lt;ウィンドウ番号&gt; 1 のウィンドウに &lt;テキストエリア番号&gt; 10 &lt;文字列&gt; SAMPLEのテキストエリアを配置します。</p>		
使用例2	<p>GUCREATE WINDOW 1 GUCREATE TEXTAREA 1, 10, "SAMPLE", FONT\$, SIZE\$, POSI\$, TIP\$ GUSHOW 1</p> <p>&lt;ウィンドウ番号&gt; 1 のウィンドウに &lt;テキストエリア番号&gt; 10 &lt;文字列&gt; SAMPLEのテキストエリアを配置します。 共通項目でフォントやサイズ、位置、ツールチップの表示内容を設定しています。</p>		

## 3.3.1.10 GUUPD TEXTAREA

命令			
機 能	テキストエリアの表示内容を更新します。		
書 式	GUUPD TEXTAREA <①テキストエリア番号>, <②文字列>		
パラ メータ	①	<テキストエリア番号>	数値
	表示内容を更新するテキストエリアの番号を指定します。		
②	<文字列>		文字列
	更新を行う文字列を指定します。		
使用例	GUCREATE WINDOW 1 GUCREATE TEXTAREA 1, 10, "TEST" GUUPD TEXTAREA 10, "SAMPLE" GUSHOW 1  <部品番号> 10 のテキストエリアの内容を<文字列> SAMPLE に更新します。		

## 3.3.1.11 GUCREATE COMBOBOX

命令			
機 能	指定したウィンドウにコンボボックスを配置します。		
書 式	GUCREATE COMBOBOX <①ウィンドウ番号>, <②コンボボックス番号> [, [ <③フォント設定> ], [ <④サイズ設定> ], [ <⑤ポジション設定> ], [ <⑥ツールチップ設定> ] ]  ※ フォント設定以降の引数を省略する場合、各引数間のカンマは必要です。 <div style="display: flex; align-items: center;"> <span style="color: blue; margin-right: 10px;">ウィンドウ</span> <div style="border: 1px solid black; padding: 10px; text-align: center;"> <span style="color: blue;">コンボボックス</span>  <div style="border: 1px solid black; display: inline-block; width: 100px; height: 20px; position: relative;"> <div style="position: absolute; right: -10px; top: 50%; transform: translateY(-50%); font-size: 10px;">▼</div> </div> </div> </div>		
パラ メータ	①	<ウィンドウ番号>	数値
	コンボボックスを配置するウィンドウの番号を指定します。		
	②	<コンボボックス番号>	数値
	作成するコンボボックスの番号を指定します。 1～10000の値を指定してください。		
	③	<フォント設定>	文字列
	GUI部品に表示される文字列のフォントを指定します。(参照：P. 20)		
	④	<サイズ設定>	文字列
	GUI部品に表示されるGUI部品の大きさを指定します。(参照：P. 22)		
⑤	<ポジション設定>		文字列
	GUI部品に表示されるGUI部品の表示場所を指定します。(参照：P. 23)		
⑥	<ツールチップ設定>		文字列
	マウスポインタがGUI部品上にきた際に、表示されるツールチップの内容を指定します。 (参照：P. 24)		
注 意	文字色と背景色の指定はできません。		
使用例1	GUCREATE WINDOW 1 GUCREATE COMBOBOX 1, 10 GUSHOW 1  <ウィンドウ番号> 1 のウィンドウに <コンボボックス番号> 10のコンボボックスを配置します。		

使用例2	<p>GUCREATE WINDOW 1          GUCREATE COMBOBOX 1, 10, FONT\$, SIZE\$, POSI\$, TIP\$          GUSHOW 1</p> <p>&lt;ウィンドウ番号&gt; 1 のウィンドウに &lt;コンボボックス番号&gt; 10のコンボボックスを配置します。</p> <p>共通項目でフォントやサイズ、位置、ツールチップの表示内容を設定しています。</p>
------	--

### 3.3.1.12 GUADD COMBOBOX

命令		
機能	指定したコンボボックスにリストを追加します。	
書式	GUADD COMBOBOX <①コンボボックス番号>, <②リスト名> [, <③初期状態> ]	
パラメータ	①	<コンボボックス番号> 数値 リスト名を追加するコンボボックスの番号を指定します。
	②	<リスト名> 文字列 コンボボックスに追加するリスト名を文字列で指定します。
	③	<初期状態> 数値 表示直後のコンボボックスの選択内容を指定できます。 0:選択無し(デフォルト値) 1:初期表示選択
備考	<ul style="list-style-type: none"><li>初期表示選択されたリストが複数ある場合、最後に選択された内容が表示されます。すべて0で選択された場合、1番最初に追加された内容が表示されます。</li></ul> <div><div>CCC</div><div>▼</div><div>AAA</div><div>BBB</div><div>CCC</div></div> <ul style="list-style-type: none"><li>実行前にコンボボックスの生成 (→「3.3.1.11 GUCREATE COMBOBOX」) が必要です。</li></ul>	
使用例	GUCREATE WINDOW 1 GUCREATE COMBOBOX 1, 20 GUADD COMBOBOX, 20, "AAA" GUADD COMBOBOX, 20, "BBB", 0 GUADD COMBOBOX, 20, "CCC", 1 GUSHOW 1  <コンボボックス番号> 20 のコンボボックスへ <リスト名> "AAA", "BBB", "CCC" の3つを追加し、"CCC"を初期選択に設定します。	

### 3.3.1.13 GUUPD COMBOBOX

<b>命令</b>			
<b>機能</b>	コンボボックスの選択項目を更新します。		
<b>書式</b>	GUUPD COMBOBOX SELECT <①コンボボックス番号>, <②値のインデックス番号または値>		
<b>パラメータ</b>	①	< <b>コンボボックス番号</b> >	数値
	表示内容を更新するコンボボックスの番号を指定します。		
	②	< <b>値のインデックス番号または値</b> >	数値／文字列
<p>選択する値を指定します。          値のインデックス番号を1はじまりの数値で指定します。または値を文字列で指定します。</p>			
<b>使用例</b>	<pre>GUCREATE WINDOW 1 GUCREATE COMBOBOX 1, 20 GUADD COMBOBOX 20, "AAA" GUADD COMBOBOX 20, "BBB" GUADD COMBOBOX 20, "CCC" GUUPD COMBOBOX SELECT 10, 2 GUSHOW 1</pre> <p>部品番号 20 のコンボボックスへ リスト "AAA","BBB","CCC"の3つを追加し、"BBB"を選択状態に設定します。</p>		

## 3.3.1.14 GUCREATE RADIOBTN

命令		
機能	指定したウィンドウにラジオボタンの領域を作成します。	
書式	GUCREATE RADIOBTN <①ウィンドウ番号>, <②ラジオボタン番号>, <③追加位置> [, [ <④フォント設定> ], [ <⑤サイズ設定> ], [ <⑥ポジション設定> ], [ <⑦ツールチップ設定> ] ]	
	※ フォント設定以降の引数を省略する場合、各引数間のカンマは必要です。	
パラメータ	①	<ウィンドウ番号> 数値
	ラジオボタンを表示するウィンドウの番号を指定します。	
	②	<ラジオボタン番号> 数値
	作成するラジオボタンの領域の番号を指定します。 1～10000の値を指定してください。	
	③	<追加位置> 数値
	ラジオボタンの追加を行う位置を指定します。 0: 右側に追加 1: 下側に追加	
	④	<フォント設定> 文字列
	GUI部品に表示される文字列のフォントを指定します。(参照 : P. 20)	
	⑤	<サイズ設定> 文字列
使用例1	GUI部品に表示されるGUI部品の大きさを指定します。(参照 : P. 22)	
	⑥	<ポジション設定> 文字列
	GUI部品に表示されるGUI部品の表示場所を指定します。(参照 : P. 23)	
	⑦	<ツールチップ設定> 文字列
	マウスポインタがGUI部品上に来た際に、表示されるツールチップの内容を指定します。 (参照 : P. 24)	
	GUCREATE WINDOW 1	
	GUCREATE RADIOBTN 1, 10, 0	
	GUSHOW 1	
	<ウィンドウ番号> 1 のウィンドウに<ラジオボタン番号> 10 <追加位置> 0のラジオボタンを配置します。	
使用例2	GUCREATE WINDOW 1	
	GUCREATE RADIOBTN 1, 10, 0, FONT\$, SIZE\$, POSI\$, TIP\$	
	GUSHOW 1	
	<ウィンドウ番号> 1 のウィンドウに<ラジオボタン番号> 10 <追加位置> 0のラジオボタンを配置します。	
	共通項目でフォントやサイズ、位置、ツールチップの表示内容を設定しています。	

## 3.3.1.15 GUADD RADIOBTN

命令		
機能	ラジオボタンの領域にラジオボタンを追加します。	
書式	GUADD RADIOBTN <①ラジオボタン番号>, <②リスト名> [, <③初期状態> ]	
パラメータ	①	<div> <div>&lt;ラジオボタン番号&gt;</div> <div>数値</div> </div>
	リスト名の追加を行うラジオボタンの領域の番号を指定します。	
	②	<div> <div>&lt;リスト名&gt;</div> <div>文字列</div> </div>
	ラジオボタンに追加するリスト名を文字列で指定します。	
	③	<div> <div>&lt;初期状態&gt;</div> <div>数値</div> </div>
	表示直後のラジオボタンのボタン状態を指定できます。	
	0:選択無し(デフォルト値)	
	1:初期選択	
備考	<p>・初期選択された処理が複数ある場合、最後に選択された内容にチェックが入ります。すべて0で選択された場合、1番最初に追加された内容にチェックが入ります。</p> <div> <div>ウィンドウ</div> <div> <div>ラジオボタン</div> <div>○AAA</div> <div>○BBB</div> <div>◎CCC</div> </div> </div> <p>・実行前にラジオボタンの生成 (→「3.3.1.14 GUCREATE RADIOBTN」) が必要です。</p>	
使用例	<pre>GUCREATE WINDOW 1 GUCREATE RADIOBTN 1, 20 GUADD RADIOBTN 20, "AAA" GUADD RADIOBTN 20, "BBB", 1 GUADD RADIOBTN 20, "CCC", 0 GUSHOW 1</pre> <p>&lt;ラジオボタン番号&gt; 20 のラジオボタン領域へ &lt;リスト名&gt;"AAA", "BBB", "CCC"の3つを追加し、"BBB"を初期選択に設定します。</p>	

## 3.3.1.16 GUUPD RADIOBTN

命令		
機 能	ラジオボタンの表示内容を更新します。	
書 式	GUUPD RADIOBTN CHECK <①ラジオボタン番号>, <②値のインデックス番号または値>	
パラ メータ	①	<ラジオボタン番号> 数値
	表示内容を更新するラジオボタンの番号を指定します。	
パラ メータ	②	<値のインデックス番号または値> 数値／文字列
	選択する値を指定します。 値のインデックス番号を1はじまりの数値で指定します。または値を文字列で指定します。	
使用例	<pre>GUCREATE WINDOW 1 GUCREATE RADIOBTN 1, 20 GUADD RADIOBTN 20, "AAA" GUADD RADIOBTN 20, "BBB" GUADD RADIOBTN 20, "CCC" GUUPD RADIOBTN CHECK 10, 2 GUSHOW 1</pre> <p>ラジオボタン番号 20 のラジオボタン領域へ ラジオボタン"AAA", "BBB", "CCC"の3つを追加し、"BBB"を選択状態に設定します。</p>	

## 3.3.1.17 GUCREATE CHECKBOX

命令		
機 能	指定したウィンドウにチェックボックスを配置します。	
書 式	GUCREATE CHECKBOX <①ウィンドウ番号>, <②チェックボックス番号>, <③文字列> [, <④初期状態> [, [, <⑤フォント設定> ], [, <⑥サイズ設定> ], [, <⑦ポジション設定> ], [, <⑧ツールチップ設定> ] ] ]	
パラ メータ	※ フォント設定以降の引数を省略する場合、各引数間のカンマは必要です。 <div style="display: flex; align-items: center; margin-top: 10px;"> <div style="margin-right: 20px; color: blue;">ウィンドウ</div> <div style="border: 1px solid black; padding: 10px; text-align: center;"> <div style="color: blue; margin-bottom: 5px;">チェックボックス</div> <input type="checkbox"/>SAMPLE </div> </div>	
	①	<ウィンドウ番号> 数値
パラ メータ	チェックボックスを配置するウィンドウの番号を指定します。	
	②	<チェックボックス番号> 数値
パラ メータ	作成するチェックボックスの番号を指定します。 1～10000の値を指定して下さい。	
	③	<文字列> 文字列
パラ メータ	チェックボックスの右隣に指定した文字列を表示します。	
	④	<初期状態> 数値
パラ メータ	表示直後のチェックボックスの状態を指定します。 0:チェック無し(デフォルト値) 1:チェックあり	
	⑤	<フォント設定> 文字列
パラ メータ	GUI部品に表示される文字列のフォントを指定します。(参照：P. 20)	

	⑥	＜サイズ設定＞	文字列
	GUI部品に表示されるGUI部品の大きさを指定します。(参照:P. 22)		
	⑦	＜ポジション設定＞	文字列
	GUI部品に表示されるGUI部品の表示場所を指定します。(参照:P. 23)		
	⑧	＜ツールチップ設定＞	文字列
	マウスポインタがGUI部品上に来た際に、表示されるツールチップの内容を指定します。(参照:P. 24)		
備 考	<ul style="list-style-type: none"> <li>・チェックボックスの文字列に、「_(アンダーバー)」の後に英数文字を設定すると、ALT+＜指定した英数文字＞を押すことで、チェックボックスを押す事ができます。これを、アクセラレータキーと呼びます。</li> <li>・チェックボックスの文字列に、単に「_」を表記したい場合は、「__」と2回続けてください。</li> </ul>		
使用例1	GUCREATE WINDOW 1 GUCREATE CHECKBOX 1, 10, "SAMPLE" GUSHOW 1  ＜ウィンドウ番号＞ 1 のウィンドウに ＜チェックボックス番号＞ 10 ＜文字列＞ SAMPLEのチェックボックスを配置します。		
使用例2	GUCREATE WINDOW 1 GUCREATE CHECKBOX 1, 10, " SAMPLE ", 1 GUSHOW 1  ＜ウィンドウ番号＞ 1 のウィンドウに ＜チェックボックス番号＞ 10 ＜文字列＞ SAMPLE ＜初期状態＞チェックありのチェックボックスを配置します。		
使用例3	GUCREATE WINDOW 1 GUCREATE CHECKBOX 1, 10, " SAMPLE ", 1, FONT\$, SIZE\$, POSI\$, TIP\$ GUSHOW 1  ＜ウィンドウ番号＞ 1 のウィンドウに ＜チェックボックス番号＞ 10 ＜文字列＞ SAMPLE ＜初期状態＞チェックありのチェックボックスを配置します。 共通項目でフォントやサイズ、位置、ツールチップの表示内容を設定しています。		

### 3.3.1.18 GUUPD CHECKBOX

命令			
機 能	チェックボックスの選択状態を更新します。		
書 式	GUUPD CHECKBOX CHECK <①チェックボックス番号>, <②チェックの値>		
パラ メータ	①	＜チェックボックス番号＞	数値
	表示内容を更新するチェックボックスの番号を指定します。		
	②	＜チェックの値＞	数値
使用例	GUCREATE WINDOW 1 GUCREATE CHECKBOX 1, 10, 0 GUUPD CHECKBOX CHECK 10, 1 GUSHOW 1		
	＜部品番号＞ 10 のチェックボックスのチェックをチェックありに更新します。		



## 3.3.1.19 GUCREATE IMAGE

命令		
機能	指定したウィンドウに画像イメージを貼り付けます。	
書式1	<p>GUCREATE IMAGE &lt;①ウィンドウ番号&gt;, &lt;②イメージ番号&gt;, &lt;③画像ファイルパス&gt; [, [ &lt;⑥フォント設定&gt; ], [ &lt;⑦サイズ設定&gt; ], [ &lt;⑧ポジション設定&gt; ], [ &lt;⑨ツールチップ設定&gt; ] ]</p> <p>指定したウィンドウに、画像ファイルで指定した画像イメージを貼り付けます。  ※ フォント設定以降の引数を省略する場合、各引数間のカンマは必要です。  ※ サイズ設定の設定値より、画像の大きさが優先されます。</p>	
書式2	<p>GUCREATE IMAGE &lt;①ウィンドウ番号&gt;, &lt;②イメージ番号&gt;, &lt;③画像ファイルパス&gt;, FIXED, &lt;④横幅&gt;, &lt;④縦幅&gt; [, [ &lt;⑥フォント設定&gt; ], [ &lt;⑦サイズ設定&gt; ], [ &lt;⑧ポジション設定&gt; ], [ &lt;⑨ツールチップ設定&gt; ] ]</p> <p>指定したウィンドウに、画像ファイルで指定した画像イメージの横幅と縦幅を指定して貼り付けます。  ※ フォント設定以降の引数を省略する場合、各引数間のカンマは必要です。</p>	
書式3	<p>GUCREATE IMAGE &lt;①ウィンドウ番号&gt;, &lt;②イメージ番号&gt;, &lt;③画像ファイルパス&gt;, ROTATE, &lt;⑤角度&gt; [, [ &lt;⑥フォント設定&gt; ], [ &lt;⑦サイズ設定&gt; ], [ &lt;⑧ポジション設定&gt; ], [ &lt;⑨ツールチップ設定&gt; ] ]</p> <p>指定したウィンドウに、画像ファイルで指定した画像イメージを回転させつつ貼り付けます。  ※ フォント設定以降の引数を省略する場合、各引数間のカンマは必要です。  ※ サイズ設定の設定値より、画像の大きさが優先されます。</p>	
パラメータ	①	<ウィンドウ番号> 数値
	画像イメージを表示するウィンドウの番号を指定します。	
	②	<イメージ番号> 数値
	表示する画像イメージの番号を指定します。1～10000の値を指定して下さい。	
	③	<画像ファイルパス> 文字列
	表示する画像ファイルのパスを指定します。 指定できるファイルはJPEG, PNG, GIF形式です。	
	④	<横幅>, <縦幅> 数値
	書式2の時、画像の大きさを横幅、縦幅に強制します。	
	⑤	<角度> 数値
	書式3の時、画像を指定角度に回転します。 0度、90度、180度、270度が指定できます。	
	⑥	<フォント設定> 文字列
	GUI部品に表示される文字列のフォントを指定します。(参照：P. 20)	
	⑦	<サイズ設定> 文字列
	GUI部品に表示されるGUI部品の大きさを指定します。(参照：P. 22)	
	⑧	<ポジション設定> 文字列
	GUI部品に表示されるGUI部品の表示場所を指定します。(参照：P. 23)	
	⑨	<ツールチップ設定> 文字列
	マウスポインタがGUI部品上に来た際に、表示されるツールチップの内容を指定します。 (参照：P. 24)	
備考	・FIXEDとROTATEの同時適用は行えません。	
注意	アニメーションGIFを表示する際、FIXEDやROTATEを使用するとアニメーションが停止します	
使用例1	<p>GUCREATE WINDOW 1</p> <p>GUCREATE IMAGE 1, 10, "sample.png"</p> <p>GUSHOW 1</p>	

	<p>&lt;ウィンドウ番号&gt; 1 のウィンドウに&lt;イメージ番号&gt; 10の画像ファイルsample.pngを貼り付けます。</p>
使用例2	<p>GUCREATE WINDOW 1 GUCREATE IMAGE 1, 10, "sample.png", FONT\$, SIZE\$, POSI\$, TIP\$ GUSHOW 1</p> <p>&lt;ウィンドウ番号&gt; 1 のウィンドウに&lt;イメージ番号&gt; 10の画像ファイルsample.pngを貼り付けます。 共通項目でフォントやサイズ、位置、ツールチップの表示内容を設定しています。</p>
使用例3	<p>GUCREATE WINDOW 1 GUCREATE IMAGE 1, 10, "sample.png", FIXED, 100, 50 GUSHOW 1</p> <p>&lt;ウィンドウ番号&gt; 1 のウィンドウに&lt;イメージ番号&gt; 10の画像ファイルsample.pngを貼り付けます。 FIXEDを指定することで&lt;横幅&gt; 100 &lt;縦幅&gt; 50で大きさを強制します。</p>
使用例4	<p>GUCREATE WINDOW 1 GUCREATE IMAGE 1, 10, "sample.png", FIXED, 100, 50, FONT\$, SIZE\$, POSI\$, TIP\$ GUSHOW 1</p> <p>&lt;ウィンドウ番号&gt; 1 のウィンドウに&lt;イメージ番号&gt; 10の画像ファイルsample.pngを貼り付けます。 FIXEDを指定することで&lt;横幅&gt; 100 &lt;縦幅&gt; 50で大きさを強制します。 共通項目でフォントやサイズ、位置、ツールチップの表示内容を設定しています。</p>
使用例5	<p>GUCREATE WINDOW 1 GUCREATE IMAGE 1, 10, "sample.png", ROTATE, 90 GUSHOW 1</p> <p>&lt;ウィンドウ番号&gt; 1 のウィンドウに&lt;イメージ番号&gt; 10の画像ファイルsample.pngを貼り付けます。 ROTATEを指定することで&lt;角度&gt; 90度 で画像を回転します。</p>
使用例6	<p>GUCREATE WINDOW 1 GUCREATE IMAGE 1, 10, "sample.png", ROTATE, 90, FONT\$, SIZE\$, POSI\$, TIP\$ GUSHOW 1</p> <p>&lt;ウィンドウ番号&gt; 1 のウィンドウに&lt;イメージ番号&gt; 10の画像ファイルsample.pngを貼り付けます。 ROTATEを指定することで&lt;角度&gt; 90度 で画像を回転します。 共通項目でフォントやサイズ、位置、ツールチップの表示内容を設定しています。</p>

## 3.3.1.20 GUUPD IMAGE

命令			
機 能	画像イメージの内容を、指定した画像ファイルに更新します。		
書 式	GUUPD IMAGE <①イメージ番号>, <②画像ファイルパス>		
パラ メータ	①	<イメージ番号>	数値
	表示内容を更新する部品の番号を指定します。		
パラ メータ	②	<画像ファイルパス>	文字列
	表示する画像ファイルのパスを指定します。 指定できるファイルはJPEG, PNG, GIF形式です。		
使用例	GUCREATE WINDOW 1 GUCREATE IMAGE 1, 10, "sample1.png" GUUPD IMAGE 10, "sample2.png" GUSHOW 1  <イメージ番号> 10の画像を画像ファイルsample2.pngで更新します。		

## 3.3.1.21 GUUPD IMAGE FIXED



命令			
機 能	画像イメージの横幅、縦幅を更新します。		
書 式	GUUPD IMAGE FIXED <①イメージ番号>, <②横幅>, <③縦幅>		
パラ メータ	①	<イメージ番号>	数値
	表示内容を更新する部品の番号を指定します。		
パラ メータ	②	<横幅>, <縦幅>	数値
	画像の大きさを横幅、縦幅に強制します。		
使用例	GUCREATE WINDOW 1 GUCREATE IMAGE 1, 10, "sample.png" GUUPD IMAGE FIXED 10, 100, 50 GUSHOW 1  <イメージ番号> 10 に設定した画像ファイルを<横幅> 100 <縦幅> 50で大きさを強制します。		

## 3.3.1.22 GUUPD IMAGE ROTATE

命令			
機 能	画像イメージの描画角度を更新します。		
書 式	GUUPD IMAGE ROTATE <①イメージ番号>, <②角度>		
パラ メータ	①	<イメージ番号>	数値
	表示内容を更新する部品の番号を指定します。		
パラ メータ	②	<角度>	数値
	画像を指定角度に回転します。 0度、90度、180度、270度が指定できます。		
使用例	GUCREATE WINDOW 1 GUCREATE IMAGE 1, 10, "sample.png" GUUPD IMAGE ROTATE 10, 90 GUSHOW 1  <イメージ番号> 10 に設定した画像ファイルを<角度> 90度 で回転します。		

### 3.3.1.23 GUCREATE SCALE

命令								
機能	指定したウィンドウにスケールを配置します。							
書式	GUCREATE SCALE <①ウィンドウ番号>, <②スケール番号>, <③設定文字列> [, [, <④フォント設定> ], [, <⑤サイズ設定> ], [, <⑥ポジション設定> ], [, <⑦ツールチップ設定> ] ]							
パラメータ	※ フォント設定以降の引数を省略する場合、各引数間のカンマは必要です。							
	①	<ウィンドウ番号> 数値						
	スケールを配置するウィンドウの番号を指定します。							
	②	<スケール番号> 数値						
	作成するスケールの番号を指定します。 1~10000の値を指定してください。							
	③	<設定文字列> 文字列						
	スケールの初期値と最小値、最大値、ステップ値をカンマ区切りで指定します。							
	設定文字列を文字列配列で指定すると、添字1以降にオプションを指定できます。設定できるオプション設定は以下の通りです。 書式は使用例 3 を参照してください。							
	<table><tr><th>オプション文字列</th><th>効果</th></tr><tr><td>"INVERT"</td><td>スケールのゲージの方向を反転します。</td></tr><tr><td>"POS=&lt;位置&gt;"</td><td>スケールの現在値を表示する位置を、任意の場所に設定できます。 設定は、"TOP"(上)、"BOTTOM"(下)、 "LEFT"(左)、"RIGHT"(右)です。</td></tr></table>		オプション文字列	効果	"INVERT"	スケールのゲージの方向を反転します。	"POS=<位置>"	スケールの現在値を表示する位置を、任意の場所に設定できます。 設定は、"TOP"(上)、"BOTTOM"(下)、 "LEFT"(左)、"RIGHT"(右)です。
	オプション文字列	効果						
"INVERT"	スケールのゲージの方向を反転します。							
"POS=<位置>"	スケールの現在値を表示する位置を、任意の場所に設定できます。 設定は、"TOP"(上)、"BOTTOM"(下)、 "LEFT"(左)、"RIGHT"(右)です。							
④	<フォント設定> 文字列							
GUI部品に表示される文字列のフォントを指定します。(参照：P. 20)								
⑤	<サイズ設定> 文字列							
GUI部品に表示されるGUI部品の大きさを指定します。(参照：P. 22)								
横幅が長い設定にすると、スケールは横に配置されます。 縦幅が長い設定にすると、スケールは縦に配置されます。								
⑥	<ポジション設定> 文字列							
GUI部品に表示されるGUI部品の表示場所を指定します。(参照：P. 23)								
⑦	<ツールチップ設定> 文字列							
マウスポインタがGUI部品上に来た際に、表示されるツールチップの内容を指定します。 (参照：P. 24)								
注 意	・スケールに指定する<設定文字列>は初期値が必須項目です。最小値と最大値、ステップ値は省略できます。デフォルト値は以下の通りです。 初期値：必須、最小値：0、最大値：100、ステップ値：1 ・スケールの<設定文字列>で指定するステップ数は、キーボードの矢印キーで、移動する距離です。							
使用例1	GUCREATE WINDOW 1 GUCREATE SCALE 1, 10, "50,-100,100,10" GUSHOW 1  <ウィンドウ番号>1 のウィンドウに<スケール番号>10、初期値50 最小値-100 最大値100 ステップ10のスケールを配置します。 下図のようなスケールが表示されます。							

	
使用例2	<p>GUCREATE WINDOW 1  GUCREATE SCALE 1, 10, "50,-100,100,10", , SIZE\$, POSI\$, TIP\$  GUSHOW 1</p> <p>&lt;ウィンドウ番号&gt; 1 のウィンドウに&lt;スケール番号&gt; 10、初期値50 最小値-100 最大値100 ステップ10のスケールを配置します。  共通項目でサイズや位置、ツールチップの表示内容を設定しています。</p>
使用例3	<p>GUCREATE WINDOW 1  GUCREATE SCALE 1, 10, [ "50,-100,100,10"; "INVERT" ]  GUSHOW 1</p> <p>使用例1に対して、オプション設定 "INVERT" (反転) を指定した例です。  下図のようなスケールが表示されます。</p> 

### 3.3.1.24 GUUPD SCALE

命令			
機能	スケールの現在値を更新します。		
書式	GUUPD SCALE <①スケール番号>, <②更新値>		
パラメータ	①	<スケール番号>	数値
	表示内容を更新するスケールの番号を指定します。		
	②	<更新値>	数値
更新を行う値を数値で指定します。			
使用例	<p>GUCREATE WINDOW 1  GUCREATE SCALE 1, 10, "50,-100,100,10"  GUUPD SCALE 10, 100  GUSHOW 1</p> <p>&lt;部品番号&gt; 10 のスケールの現在値を100に更新します。</p>		

## 3.3.1.25 GUCREATE SWITCH

命令		
機能	指定したウィンドウにスイッチを配置します。	
書式	GUCREATE SWITCH <①ウィンドウ番号>, <②スイッチ番号>, <③初期状態> [, [, <④フォント設定> ], [, <⑤サイズ設定> ], [, <⑥ポジション設定> ], [, <⑦ツールチップ設定> ] ]	
	※ フォント設定以降の引数を省略する場合、各引数間のカンマは必要です。	
パラメータ	①	数値
	<ウィンドウ番号> スイッチを配置するウィンドウの番号を指定します。	
	②	数値
	<スイッチ番号> 作成するスイッチの番号を指定します。 1～10000の値を指定してください。	
	③	数値
	<初期状態> スイッチの初期状態を指定します。 0: 表示はオフ 1: 表示はオン	
	④	文字列
	<フォント設定> GUI部品に表示される文字列のフォントを指定します。(参照：P. 20)	
	⑤	文字列
	<サイズ設定> GUI部品に表示されるGUI部品の大きさを指定します。(参照：P. 22)	
	⑥	文字列
	<ポジション設定> GUI部品に表示されるGUI部品の表示場所を指定します。(参照：P. 23)	
	⑦	文字列
	<ツールチップ設定> マウスポインタがGUI部品上にきた際に、表示されるツールチップの内容を指定します。 (参照：P. 24)	
使用例1	GUCREATE WINDOW 1 GUCREATE SWITCH 1, 10, 0 GUSHOW 1  <ウィンドウ番号> 1 のウィンドウに<スイッチ番号> 10 <初期状態> 0のスイッチを配置します。	
使用例2	GUCREATE WINDOW 1 GUCREATE SWITCH 1, 10, 0, , SIZE\$, POSI\$, TIP\$ GUSHOW 1  <ウィンドウ番号> 1 のウィンドウに<スイッチ番号> 10 <初期状態> 0のスイッチを配置します。 共通項目でサイズや位置、ツールチップの表示内容を設定しています。	

## 3.3.1.26 GUUPD SWITCH

命令			
機 能	スイッチのオン/オフを更新します。		
書 式	GUUPD SWITCH <①スイッチ番号>, <②更新値>		
パラ メータ	①	<スイッチ番号>	数値
	表示内容を更新するスイッチの番号を指定します。		
	②	<更新値>	数値
	以下の値でオン/オフを指定します 0: オフ 1: オン		
使用例	GUCREATE WINDOW 1 GUCREATE SWITCH 1, 10, 0 GUUPD SWITCH 10, 1 GUSHOW 1  <部品番号> 10 のスイッチをオンに更新します。		

## 3.3.1.27 GUCREATE CALENDAR

命令		
機能	指定したウィンドウにカレンダーを配置します。	
書式	GUCREATE CALENDAR <①ウィンドウ番号>, <②カレンダー番号>, <③年月日初期設定値> [, [ <④フォント設定> ], [ <⑤サイズ設定> ], [ <⑥ポジション設定> ], [ <⑦ツールチップ設定> ] ]  ※ フォント設定以降の引数を省略する場合、各引数間のカンマは必要です。	
パラメータ	①	<ウィンドウ番号> カレンダーを配置するウィンドウの番号を指定します。
	②	<カレンダー番号> 作成するカレンダーの番号を指定します。 1~10000の値を指定して下さい。
	③	<年月日初期設定値> カレンダーの初期年月日を指定します。 "年/月/日"の文字列形式で与えてください。
	④	<フォント設定> GUI部品に表示される文字列のフォントを指定します。(参照 : P. 20)
	⑤	<サイズ設定> GUI部品に表示されるGUI部品の大きさを指定します。(参照 : P. 22)
	⑥	<ポジション設定> GUI部品に表示されるGUI部品の表示場所を指定します。(参照 : P. 23)
	⑦	<ツールチップ設定> マウスポインタがGUI部品上に来た際に、表示されるツールチップの内容を指定します。 (参照 : P. 24)
使用例1	GUCREATE WINDOW 1 GUCREATE CALENDAR 1, 2, "2021/01/01" GUSHOW 1  <ウィンドウ番号> 1 のウィンドウに <カレンダー番号> 2 <初期年月日> "2021/01/01"を指定して配置します。	
使用例2	GUCREATE WINDOW 1 GUCREATE CALENDAR 1, 2, "2021/01/01", FONT\$, SIZE\$, POSI\$, TIP\$ GUSHOW 1  <ウィンドウ番号> 1 のウィンドウに <カレンダー番号> 2 <初期年月日> "2021/01/01"のカレンダーを配置します。 共通項目でフォントやサイズ、位置、ツールチップの表示内容を設定しています。	



## 3.3.1.28 GUUPD CALENDAR

命令		
機 能	カレンダーの年月日を更新します。	
書 式	GUUPD CALENDAR <①カレンダー番号>, <②年月日設定値>	
パラ メータ	①	<div> <div>&lt;カレンダー番号&gt;</div> <div>数値</div> </div>
	表示内容を更新するカレンダーの番号を指定します。	
	②	<div> <div>&lt;年月日設定値&gt;</div> <div>文字列</div> </div>
カレンダーの年月日を指定します。 "年/月/日"の文字列形式で与えてください。		
使用例	GUCREATE WINDOW 1 GUCREATE CALENDAR 1, 10, "2021/01/01" GUUPD CALENDAR 10, "2022/03/26" GUSHOW 1  <部品番号> 10 のカレンダーを"2022/03/26"に更新します。	

## 3. 3. 1. 29 GUCREATE VIDEO

命令		
機 能	指定したウィンドウにビデオ描画を配置します。	
書 式	GUCREATE VIDEO <①ウィンドウ番号>, <②ビデオ番号> [, [ <③フォント設定> ], [ <④サイズ設定> ], [ <⑤ポジション設定> ], [ <⑥ツールチップ設定> ] ]	
	※ フォント設定以降の引数を省略する場合、各引数間のカンマは必要です。	
パラ メータ	①	<ウィンドウ番号> 数値
	ビデオを配置するウィンドウの番号を指定します。	
	②	<ビデオ番号> 数値
	作成するビデオの番号を指定します。 1以上の値を指定してください。	
	③	<フォント設定> 文字列
	GUI部品に表示される文字列のフォントを指定します。(参照：P. xx)	
	④	<サイズ設定> 文字列
	GUI部品に表示されるGUI部品の大きさを指定します。(参照：P. xx)	
	⑤	<ポジション設定> 文字列
	GUI部品に表示されるGUI部品の表示場所を指定します。(参照：P. xx)	
	⑥	<ツールチップ設定> 文字列
	マウスポインタがGUI部品上に来た際に、表示されるツールチップの内容を指定します。 (参照：P. xx)	
備 考	<ul style="list-style-type: none"> <li>ビデオ番号は、管理し難くなるので、10000を超える値を指定しないでください。</li> <li>本コマンドは、libvlcライブラリを使用しています。</li> </ul>	
使用例	GUCREATE WINDOW 1 GUCREATE VIDEO 1, 10 GUSHOW 1  <ウィンドウ番号> 1 のウィンドウに <ビデオ番号> 10を指定して配置します。	

## 3.3.1.30 GUUPD VIDEO

命令			
機能	ビデオに対して制御を指示します。		
書式	GUUPD VIDEO <①ビデオ番号>, <②制御コマンド> [, <③コマンド引数> ]		
パラメータ	①	<ビデオ番号>	数値
	制御を行うビデオ番号を指定します。		
	②	<制御コマンド>	文字列
	ビデオに対する制御の指示を、文字列で指定します。		
	制御コマンド	コマンド引数	説明
	OPEN	文字列	再生したいビデオファイルのパスを指定します。 文字列配列にすると、添字：1以降は、ビデオファイルに対するオプション指定となります。 (指定可能なオプションは、libvlcを参照)
	PLAY	なし	OPEN後に呼び出すと、ビデオを再生します。
	PAUSE	なし	再生中に呼び出すと、一時停止します。
	STOP	なし	再生中に呼び出すと、停止します。
	SEEK_POS	数値	一時停止中に呼び出すと、再生位置を変更できます。 0から1までの間で指定します。
	SEEK_MS	数値	一時停止中に呼び出すと、再生位置を変更できます。 0からms単位で指定します。
	PREV_CHAPTER	なし	前のキャプターがあれば、再生位置を移動します。
	NEXT_CHAPTER	なし	次のキャプターがあれば、再生位置を移動します。
	RATE	数値	再生速度を変更します。 1が等倍。2が2倍。といった風に指定します。
	VOLUME	数値	ボリュームを変更します。 db単位で指定します。
	③	<コマンド引数>	値
	制御コマンドに対する、引数を指定します。		
備考	<ul style="list-style-type: none"> <li>OPENやPLAY時、存在しないビデオファイルを指定しても、内部のOS側ライブラリがエラーメッセージを出力しますが、AJANはエラーで停止しません。</li> <li>各制御コマンドの呼び出しは同期しません。呼び出しの後に、内部のOS側ライブラリが遅延して動作します。 このため、制御コマンドを続けて呼び出した時、動作が意図通りになるかは、保証されません。</li> <li>各制御コマンドの呼び出し時、内部のOS側ライブラリより、警告的なメッセージが出力される場合があります。</li> </ul>		
注意	・ビデオ領域の生成（→「3.3.1.29 GUCREATE VIDEO」）が必要です。		
使用例1	FNAME\$ = "file:///usr/share/interface/AJANPro/web_samples/WEB/static/fa.mp4" GUUPD VIDEO 10, "OPEN", FNAME\$ ' ビデオファイルをオープンして再生します。		
使用例2	FNAME\$ = "file:///usr/share/interface/AJANPro/web_samples/WEB/static/fa.mp4" GUUPD VIDEO 10, "OPEN", [ FNAME\$; "input-repeat=100" ] ' ビデオファイルをオープンする際に、オプションを追加指定します。 ' オプションには「input-repeat=100」を指定して、100回繰り返し再生させます。		
使用例3	' 仮に、USB Webカメラを接続しており、 ' Linuxコマンドで、/dev/video0 デバイスが認識されていると仮定します。 ' 以下は、Linuxコマンドの呼び出し確認事例です。 \$ v4l2-ctl --list-devices USB_Camera (usb-0000:00:14.0-1): /dev/video0  ' 上で認識された Webカメラのデバイスを指定して、オープンする事例です。		

	GUUPD VIDEO 10, "OPEN", "v4l2:///dev/video0"
--	--

## 3.3.1.31 GUSHOW

命令			
機 能	指定したウィンドウを表示します。		
書 式	GUSHOW <①ウィンドウ番号>		
パラ メータ	①	<ウィンドウ番号>	数値
	画面に表示するウィンドウの番号を指定します。		
使用例	GUCREATE WINDOW 1		
	GUSHOW 1		
	<ウィンドウ番号> 1 のウィンドウを表示します。		

## 3.3.1.32 GUMAIN LOOP

命令	
機 能	GUIアプリケーションを動かすためのGUIメインイベントループを動かします。 「GUMAIN QUIT」が呼び出されるまで、このコマンドから戻ってきません。
書 式	GUMAIN LOOP
備 考	<ul style="list-style-type: none"> <li>GUIによるウィンドウを作成して表示した後、GUIのイベント処理を行うために、このコマンドを呼び出してください。</li> <li>OSは、この呼び出しの間、内部で GUIの描画やボタン押しなどのイベント処理を、終了指示があるまで、実行ループし続けます。</li> <li>このイベントループの終了は、「GUMAIN QUIT」を用います。</li> </ul>
使用例	GUCREATE WINDOW 1 ' ウィンドウの作成 GUSHOW 1 ' ウィンドウの表示 GUMAIN LOOP ' GUIアプリケーションとして動作  ウィンドウを作成から表示した後、GUIアプリケーションとして動作させます

## 3.3.1.33 GUMAIN ONELOOP

関数			
機 能	GUIアプリケーションを動かすためのGUIメインイベントループを、1回だけ動かします。 このコマンドは実行後に即座に戻ってきます。		
書 式	<(戻り値)ステータス> = GUMAIN ONELOOP( <①イベント待ち> )		
戻り値	戻り値	<ステータス>	数値
	TRUEを返します。		
パラメータ	①	<イベント待ち>	数値
	TRUEを指定すると、描画処理などのイベントがあるまで待機します。 FALSEを指定すると、描画処理などのイベントがあれば実行して戻り、無ければそのまま戻ります。		
備 考	<ul style="list-style-type: none"> <li>「GUMAIN LOOP」は、「GUMAIN ONELOOP」を繰り返し実行しているのと、ほぼ同じ事を行います。</li> <li>1回の呼び出しで、OSから要求のあったGUI描画に必要な大量の処理要求の1つを消化します。このため、円滑に描画を行うには、「GUMAIN ONELOOP」を頻繁に呼び出す事が肝要です。</li> </ul>		
使用例	GUCREATE WINDOW 1 ' ウィンドウの作成 GUSHOW 1 ' ウィンドウの表示 DO WHILE TRUE TMP = GUMAIN ONELOOP(TRUE) LOOP  ウィンドウを作成して表示した後、GUIアプリケーションとして動作させます		

## 3.3.1.34 GUMAIN QUIT

命令	
機 能	GUIメインイベントループを終了させます。
書 式	GUMAIN QUIT
備 考	・ この呼び出し (GUMAIN QUIT) を行くと、「GUMAIN LOOP」は、GUIメインイベントループを終了して戻ります。
注 意	・ 「GUMAIN ONELOOP」使用時は、「GUMAIN QUIT」の呼び出しは機能しません。
使用例	<p>GUMAIN LOOPでGUIメインイベントループを回している中、GUMAIN QUITを呼び出して、イベントループを終了させるコード断片の事例です。</p> <pre>' WINDOWのイベントコールバック処理 SUB WINDOW_EVENT(NUM, NAME\$, EVT)   ' GUMAIN LOOPを終了させる   GUMAIN QUIT() END SUB</pre> <p>' WINDOWのイベント登録 ON GUEVT WINDOW 1 CALL WINDOW_EVENT GUEVT WINDOW 1 ON</p> <p>' GUIメインイベントループを開始 - ウィンドウを閉じるまで戻ってこない GUMAIN LOOP</p>

## 3.3.1.35 GUDSP

命令			
機 能	指定したGUI部品を表示あるいは非表示にします。		
書 式	GUDSP <①部品名> <②部品番号>, <③表示状態> [, <④編集可能状態> ]		
パラ メータ	①	<部品名>	キーワード
	表示あるいは非表示にする部品名を指定します。 ※指定できるGUI部品は以下のものです。 BUTTON CALENDAR CHECKBOX COMBOBOX IMAGE LABEL RADIOBTN SCALE SWITCH TEXTAREA TEXTBOX		
	②	<部品番号>	数値
	表示あるいは非表示にする部品の番号を指定します。		
	③	<表示状態>	真偽値
	GUI部品の表示状態を指定できます。 FALSE:非表示 TRUE:表示		
	④	<編集可能状態>	真偽値
	GUI部品の編集ができるかどうかを指定できます。 FALSE:編集不可 TRUE:編集可(デフォルト値)		
備 考	本機能の呼び出しは、ウィンドウの表示（→「3.3.1.31 GUSHOW」）後に行ってください。ウィンドウの表示時、全てGUI部品は表示状態です。		
使用例1	GUCREATE WINDOW 1 GUCREATE BUTTON 1, 10, "SAMPLE", FONT\$, SIZE\$, POSI\$, TIP\$ GUSHOW 1 GUDSP BUTTON 10, FALSE  <部品番号> 10 のボタンを非表示にします。		
使用例2	GUCREATE WINDOW 1 GUCREATE BUTTON 1, 10, "SAMPLE", FONT\$, SIZE\$, POSI\$, TIP\$ GUSHOW 1 GUDSP BUTTON 10, TRUE, FALSE  <部品番号> 10 のボタンを表示したまま、押せないようにします。		



## 3.3.1.36 GUGET\$

関数																									
機 能	指定したGUI部品の表示内容を取得します。																								
書 式	〈(戻り値)情報値〉 = GUGET\$ 〈①部品名〉 ( 〈②部品番号〉 )																								
戻り値	戻り値	〈情報値〉	文字列																						
	〈部品名〉に対応した、GUI部品の情報を取得します。																								
パラ メータ	①	〈部品名〉	キーワード																						
	表示内容を取得する部品名を指定します。																								
	<table><tr><th>部品名</th><th>動作</th></tr><tr><td>BUTTON</td><td>ボタン内に表示している文字列を取得します。</td></tr><tr><td>CALENDAR</td><td>カレンダーに設定した年月日を”yyyy/mm/dd”形式の文字列で取得します。</td></tr><tr><td>CHECKBOX</td><td>チェックボックスの状態を取得します。</td></tr><tr><td>COMBOBOX</td><td>選択されているリスト名の文字列を取得します。</td></tr><tr><td>LABEL</td><td>ラベル内に表示している文字列を取得します。</td></tr><tr><td>RADIOBTN</td><td>選択されているリスト名の文字列を取得します。</td></tr><tr><td>SCALE</td><td>スケール設定した値の文字列を取得します。</td></tr><tr><td>SWITCH</td><td>スイッチの状態(”1” / ”0”) の文字列を取得します。</td></tr><tr><td>TEXTAREA</td><td>テキストエリアの表示内容の文字列を取得します。</td></tr><tr><td>TEXTBOX</td><td>テキストボックスの表示内容の文字列を取得します。</td></tr></table>			部品名	動作	BUTTON	ボタン内に表示している文字列を取得します。	CALENDAR	カレンダーに設定した年月日を”yyyy/mm/dd”形式の文字列で取得します。	CHECKBOX	チェックボックスの状態を取得します。	COMBOBOX	選択されているリスト名の文字列を取得します。	LABEL	ラベル内に表示している文字列を取得します。	RADIOBTN	選択されているリスト名の文字列を取得します。	SCALE	スケール設定した値の文字列を取得します。	SWITCH	スイッチの状態(”1” / ”0”) の文字列を取得します。	TEXTAREA	テキストエリアの表示内容の文字列を取得します。	TEXTBOX	テキストボックスの表示内容の文字列を取得します。
	部品名	動作																							
	BUTTON	ボタン内に表示している文字列を取得します。																							
	CALENDAR	カレンダーに設定した年月日を”yyyy/mm/dd”形式の文字列で取得します。																							
	CHECKBOX	チェックボックスの状態を取得します。																							
	COMBOBOX	選択されているリスト名の文字列を取得します。																							
	LABEL	ラベル内に表示している文字列を取得します。																							
	RADIOBTN	選択されているリスト名の文字列を取得します。																							
	SCALE	スケール設定した値の文字列を取得します。																							
	SWITCH	スイッチの状態(”1” / ”0”) の文字列を取得します。																							
TEXTAREA	テキストエリアの表示内容の文字列を取得します。																								
TEXTBOX	テキストボックスの表示内容の文字列を取得します。																								
②	〈部品番号〉	数値																							
表示内容を取得する部品の番号を指定します。																									
使用例1	GUCREATE WINDOW 1 GUCREATE TEXTBOX 1, 10, ”HELLO”, FONT\$, SIZE\$, POSI\$, TIP\$ A\$ = GUGET\$ TEXTBOX(10)																								
	〈部品番号〉 10 のテキストボックスから入力内容を取得し、変数A\$へ代入します。																								
使用例2	GUCREATE WINDOW 1 GUCREATE CHECKBOX 1, 10, ”SAMPLE”, 0, FONT\$, SIZE\$, POSI\$, TIP\$ A\$ = GUGET\$ CHECKBOX(10)																								
	〈部品番号〉 10 のチェックボックスから状態を取得し、変数A\$へ代入します。 チェックボックスが選択状態の時は 1 未選択の時は 0 が代入されます。																								

3.3.1.37 GUGET\$ VIDEO

関数		
機 能	指定したビデオ部品の情報を取得します。	
書 式	<(戻り値)情報値> = GUGET\$ VIDEO ( <①ビデオ番号>, <②情報ID> )	
戻り値	戻り値	<情報値> 文字列 部品名に対応した、GUI部品の情報を取得します。
パラ メータ	①	<ビデオ番号> 数値 情報を取得するビデオ番号を指定します。
	②	<情報ID> 文字列 ビデオから取得する情報IDの指示を、文字列で指定します。
	情報ID	説明
	IS_PLAY	ビデオが再生中なら"1"を、停止中なら"0"が得られます。
	SEEK_POS	現在の再生位置を、0から1までの値で得られます。
	SEEK_MS	現在の再生位置を、ms単位で得られます。
	STATE	現在の再生状態を、以下の値から得られます。 "0" ... 何もし "1" ... オープン中 "2" ... バッファリング中 "3" ... 再生中 "4" ... 一時停止した "5" ... 停止した "6" ... 最後まで再生した "7" ... エラーが起きた
	VOLUME	現在のボリュームを、db単位で得られます。
	INFO	再生するビデオの情報(ファイル名など)が得られます。
注 意	・ビデオ領域の生成 (→「3.3.1.29 GUCREATE VIDEO」) が必要です。	
使用例	FNAME\$ = "file:///usr/share/interface/AJANPro/web_samples/WEB/static/fa.mp4" GUUPD VIDEO 10, "OPEN", FNAME\$  DO WHILE TRUE PRINT GUGET\$ VIDEO(10, "STATE") F = GUMAIN ONELOOP(0) LOOP ' ビデオファイルをオープンして再生する指示を出した後、 ' STATEで、再生状態が、どのように変化するか確認します。	

## 3.3.1.38 GUSIZE

命令			
機 能	指定したGUI部品のサイズを変更します。		
書 式	GUSIZE <①部品名> <②部品番号>, <③サイズ設定>		
パラ メータ	①	<部品名>	キーワード
	サイズ情報を変更する部品名を指定します。 以下に、指定可能なGUI部品名を列举します。 BUTTON CALENDAR CHECKBOX COMBOBOX IMAGE LABEL RADIOBTN SCALE SWITCH TEXTAREA TEXTBOX WINDOW		
	②	<部品番号>	数値
	サイズ設定を変更する部品の番号を指定します。		
	③	<サイズ設定>	文字列
GUI部品の大きさを、“<横幅>, <縦幅>” というような形式で指定します。 			

## 3.3.1.39 GUFONT

命令			
機能	指定したGUI部品のフォント情報を変更します。		
書式	GUFONT <①部品名> <②部品番号>, <③フォント設定>		
パラメータ	①	<部品名>	キーワード
	フォント情報を変更する部品名を指定します。 以下に、指定可能なGUI部品名を列挙します。 BUTTON CALENDAR CHECKBOX COMBOBOX LABEL RADIOBTN SCALE SWITCH TEXTAREA TEXTBOX		
	②	<部品番号>	数値
	フォント情報を変更する部品の番号を指定します。		
	③	<フォント設定>	文字列
	GUI部品に表示される文字列のフォントを指定します。(参照 : P. 20)		
使用例	GUFONT BUTTON 10, "32, 1, 1, 20, 8, 7"  部品番号10のボタンのフォント設定を変更します。		

## 3.3.1.40 GUTIP

命令			
機能	指定したGUI部品のツールチップを変更します。		
書式	GUTIP <①部品名> <②部品番号>, <③ツールチップ設定>		
パラメータ	①	<部品名>	キーワード
	ツールチップ情報を変更する部品名を指定します。 以下に、指定可能なGUI部品名を列挙します。 BUTTON CALENDAR CHECKBOX COMBOBOX IMAGE LABEL RADIOBTN SCALE SWITCH TEXTAREA TEXTBOX		
	②	<部品番号>	数値
	ツールチップを変更する部品の番号を指定します。		
	③	<ツールチップ設定>	文字列
	マウスポインタがGUI部品上に来た際に、表示されるツールチップの内容を指定します。(参照 : P. 24)		
使用例	GUTIP BUTTON 10, "Hello world"  部品番号10のボタンのツールチップ設定を変更します。		

## 3.3.1.41 GUDELETE

命令			
機 能	指定したウィンドウを削除します。		
書 式	GUDELETE WINDOW <①部品番号>		
パラ メータ	①	<部品番号>	数値
	削除するウィンドウ番号を指定します。		
注 意	<ul style="list-style-type: none"> <li>削除するウィンドウ番号は、GUCREATE WINDOW コマンドで生成されており、閉じる等で破棄されてない必要があります。</li> <li>有効でないウィンドウ番号を指定すると、エラーとなります。</li> </ul>		
使用例	GUCREATE WINDOW 1 GUSHOW 1 GUDELETE WINDOW 1  <部品番号> 1 のウィンドウを削除します。		

## 3.3.1.42 GUMOVE WINDOW

命令			
機 能	ウィンドウを移動します。		
書 式	GUMOVE WINDOW <①部品番号>, <②ポジション設定>		
パラ メータ	①	<部品番号>	数値
	移動する部品の番号を指定します。 ここでは、ウィンドウ番号を指定してください。		
	②	<ポジション設定>	文字列
	GUI部品に表示されるGUI部品の表示場所を指定します。(参照：P. 23)		
使用例	POSI\$ = "100, 50" GUMOVE WINDOW 1, POSI\$  ウィンドウを100, 50の位置に移動します。		

## 3.3.1.43 GUMSGBOX

関数		
機能書式	メッセージボックスを表示します。  <(戻り値)ステータス> = GUMSGBOX( <①ウィンドウ番号>, <②メッセージ内容> [, <③ボタン種類>, <④タイトル名> [, [ <⑤フォント設定> ], [ <⑥サイズ設定> ], [ <⑦ポジション設定> ], [ <⑧ツールチップ設定> ] ] )  ※フォント設定以降の引数を省略する場合、各引数間のカンマは必要です。	
戻り値	戻り値	

	<p>&lt;ウィンドウ番号&gt; 1 のウィンドウに &lt;メッセージ内容&gt; HELLO と書かれた、OKボタンの付いたメッセージボックスを表示します。</p>
使用例2	<p>RET = GUMSGBOX(1, " HELLO", 1, "タイトル")</p> <p>&lt;ウィンドウ番号&gt; 1 のウィンドウに &lt;メッセージ内容&gt; HELLO と書かれた、OKボタンとキャンセルボタンの付いたメッセージボックスを表示します。</p>
使用例3	<p>RET = GUMSGBOX(1, " HELLO", 1, "タイトル", FONT\$, SIZE\$, POSI\$, TIP\$)</p> <p>&lt;ウィンドウ番号&gt; 1 のウィンドウに &lt;メッセージ内容&gt; HELLO と書かれた、OKボタンとキャンセルボタンの付いたメッセージボックスを表示します。 共通項目でフォントやサイズ、位置、ツールチップの表示内容を設定しています。</p>

### 3.3.1.44 GUMOUSE

関数																					
機 能	マウスの情報を取得します。																				
書 式	<(戻り値)ステータス> = GUMOUSE( <①情報番号> )																				
戻り値	戻り値	<ステータス>	数値																		
	<情報番号>で指定した、マウスの特定の情報を返します。																				
パラメータ	①	<情報番号>	数値																		
	マウスから取得する情報を番号で指定します。 ボタンの状態は、押されている場合 1(TRUE)を、押されていない場合0(FALSE)を返します。																				
	<table><tr><th>情報番号</th><th>動作</th></tr><tr><td>0</td><td>左ボタンが押されているか</td></tr><tr><td>1</td><td>右ボタンが押されているか</td></tr><tr><td>2</td><td>中央ボタンが押されているか</td></tr><tr><td>3</td><td>Shiftキーが押されているか</td></tr><tr><td>4</td><td>Ctrlキーが押されているか</td></tr><tr><td>5</td><td>Altキーが押されているか</td></tr><tr><td>6</td><td>水平位置(X座標)</td></tr><tr><td>7</td><td>垂直位置(Y座標)</td></tr></table>			情報番号	動作	0	左ボタンが押されているか	1	右ボタンが押されているか	2	中央ボタンが押されているか	3	Shiftキーが押されているか	4	Ctrlキーが押されているか	5	Altキーが押されているか	6	水平位置(X座標)	7	垂直位置(Y座標)
情報番号	動作																				
0	左ボタンが押されているか																				
1	右ボタンが押されているか																				
2	中央ボタンが押されているか																				
3	Shiftキーが押されているか																				
4	Ctrlキーが押されているか																				
5	Altキーが押されているか																				
6	水平位置(X座標)																				
7	垂直位置(Y座標)																				
使用例	PRINT GUMOUSE(0)  マウスの左ボタンが押されているか、状態を表示します。																				

3.3.1.45 GUEVT ON

命令			
機能	GUI部品のイベント割り込みを許可します。 イベント割り込みのサブルーチン指定は、「ON GUEVT CALL」で定義します。		
書式	GUEVT <①部品名> <②部品番号> ON		
パラメータ	①	<部品名>	キーワード
	イベント割り込みを許可する部品名を指定します。 ※指定できるGUI部品は以下のものです。 BUTTON CALENDAR CHECKBOX COMBOBOX LABEL RADIOBTN SCALE SWITCH TEXTAREA TEXTBOX WINDOW		
	②	<部品番号>	数値
イベント割り込みを許可する部品番号を指定します。			
使用例	ボタンイベントを登録し、イベントの割り込みを許可します。 プログラムの終了前にイベント割り込みを無効にします。  SUB BUTTON_EVENT (NUM, NAME\$, EVT) ? GUGET\$ BUTTON (NUM) END SUB  ON GUEVT BUTTON 10 CALL BUTTON_EVENT GUEVT BUTTON 10 ON  GUEVT BUTTON 10 OFF END		



## 3.3.1.46 GUEVT OFF

命令			
機 能	GUI部品のイベント割り込みを無効にします。 イベント割り込みのサブルーチン指定は、「ON GUEVT CALL」で定義します。		
書 式	GUEVT <①部品名> <②部品番号> OFF		
パラ メータ	①	<部品名>	キーワード
	イベント割り込みを無効にする部品名を指定します。 ※指定できるGUI部品は以下のものです。 BUTTON CALENDAR CHECKBOX COMBOBOX LABEL RADIOBTN SCALE SWITCH TEXTAREA TEXTBOX WINDOW		
	②	<部品番号>	数値
	イベント割り込みを無効にする部品番号を指定します。		
使用例	ボタンイベントを登録し、イベントの割り込みを許可します。 プログラムの終了前にイベント割り込みを無効にします。  <pre> SUB BUTTON_EVENT (NUM, NAME\$, EVT)   ? GUGET\$ BUTTON (NUM) END SUB  ON GUEVT BUTTON 10 CALL BUTTON_EVENT GUEVT BUTTON 10 ON  GUEVT BUTTON 10 OFF END </pre>		

## 3.3.1.47 ON GUEVT CALL

命令			
機能	GUI部品のイベントが発生した時に、呼び出されるサブルーチンを定義します。 この割り込みは、本命令の呼び出し後の「GUEVT ON」で有効となります。		
書式	ON GUEVT <①部品名> <②部品番号> CALL <③サブルーチン名>		
パラ メータ	①	<部品名>	キーワード
	処理ルーチンを登録する部品名を指定します。 ※指定できるGUI部品は以下のものです。 BUTTON CALENDAR CHECKBOX COMBOBOX LABEL RADIOBTN SCALE SWITCH TEXTAREA TEXTBOX WINDOW		
	②	<部品番号>	数値
処理ルーチンを登録する部品番号を指定します。			
パラ メータ	③	<サブルーチン名>	名前
	イベントに対応したサブルーチン名を指定してください。 サブルーチンは、以下の定義に従います。  SUB サブルーチン名(<部品番号>, <部品名>, <内部イベント種別>) 処理内容 END SUB  部品番号：イベントが発生した部品番号 部品名：イベントが発生した部品名の文字列 内部イベント種別：将来拡張用(予約)  イベント要因は「<部品名とイベント名の対応一覧>(P. 67)」を参照してください。		
注意	<ul style="list-style-type: none"> <li>・「GUEVT ON」で有効にしている間は、本命令を使用することはできません。一旦「GUEVT OFF」で割り込みを無効にしてから使用してください。</li> <li>・本命令を呼び出し後、「GUEVT ON」で割り込みが有効になります。</li> <li>・CALENDAR部品で、月をまたぐように日付ボタンを押した際（例：6月が表示されている状態で7月や5月の日付をクリックする）、2回分イベントが通知されます。</li> <li>・その他の割り込みコマンドと併用した際の割り込み発生順番は、割り込み処理のサブルーチンの登録順です。</li> </ul>		
使用例	<pre> SUB BUTTON_EVENT(NUM, NAME\$, EVT)     ? GUGET\$ BUTTON(NUM) END SUB  ON GUEVT BUTTON 10 CALL BUTTON_EVENT GUEVT BUTTON 10 ON  ボタンイベントへBUTTON_EVENTサブルーチンを登録します。           </pre>		

## &lt;部品名とイベントの対応一覧&gt;

## &lt;部品名とイベント名の対応一覧&gt;

部品名	イベント要因
BUTTON	ボタンをクリックした際にイベントが発生します。
CALENDAR	日付を選択変更した際にイベントが発生します。
CHECKBOX	チェックボックスのチェックが変化した際にイベントが発生します。
COMBOBOX	コンボボックスの項目が選択された際にイベントが発生します。
LABEL	マウスをクリックした際にイベントが発生します。
RADIOBTN	ラジオボタンが選択された際にイベントが発生します。
SCALE	スケールのスライダーが変化した際にイベントが発生します。
SWITCH	スイッチのオン / オフが変化した際にイベントが発生します。
TEXTBOX	Enterキーを押して入力を確認した際にイベントが発生します。
TEXTAREA	キー入力が行われた際にイベントが発生します。
WINDOW	閉じるボタンをクリックされた際にイベントが発生します。

## 3.3.1.48 GUDIALOGBOX OPEN\$

関数		
機 能	ファイルを開くためのダイアログボックスを表示し、選択したファイルパスを取得します。	
書 式	<(戻り値)ファイルパス> = GUDIALOGBOX OPEN\$(<①ウィンドウ番号>, <②ディレクトリ> [, <③タイトル>, ( <④フィルタ説明>, <④フィルタ拡張子> ), <⑤ファイル名>, <⑥説明文> ] )	
戻り値	戻り値	<ファイルパス> 文字列
	表示したダイアログボックスで選択したファイルパスが得られます。	
パラ メータ	①	<ウィンドウ番号> 数値
	親に設定するウィンドウ番号を指定します。 0を指定すると親が無い状態でダイアログボックスを表示します	
	②	<ディレクトリ> 文字列
	最初に表示するディレクトリを指定します。	
	③	<タイトル> 文字列
	タイトルを指定します。 省略するとタイトルは「開く」が設定されます。	
	④	<フィルタ説明>, <フィルタ拡張子> 文字列
	表示するファイルに対するフィルタを丸カッコで挟んで指定します。 フィルタの説明と拡張子を文字列で指定します。 フィルタを複数指定する場合は配列にて指定します。 省略するとすべてのファイルが表示される状態となります。	
	⑤	<ファイル名> 文字列
	ファイル名入力フォームへ最初に表示する名前を指定します。 省略するとファイル名は入力されません。	
	⑥	<説明文> 文字列
	ダイアログボックスへ表示したい任意の文字列を指定します。 省略すると説明文は表示されません。	
備 考	<ul style="list-style-type: none"> <li>親ウィンドウを設定すると、ダイアログボックスが表示されている間は親のウィンドウを操作することが出来ません。</li> <li>フィルタを複数指定した場合に、説明と拡張子の配列要素数は同一にしてください。</li> <li>存在しない&lt;ディレクトリ&gt;を指定すると、エラーを検知します。</li> <li>&lt;フィルタ拡張子&gt;の指定は、Linuxのlsコマンドでのワイルドカード指定と同じです。 例えば、「*.txt」とすると、拡張子が「.txt」のファイル全てにマッチします。 大文字・小文字関係なくマッチさせたい場合は、「*. [tT][xX][tT]」と記述します。</li> <li>&lt;ファイル名&gt;で指定したファイル名の拡張子と、&lt;フィルタ拡張子&gt;で指定した拡張子を一致させるようにしてください。拡張子が一致しない場合、ダイアログボックス中の、拡張子のリスト選択は不定です。</li> <li>&lt;ファイル名&gt;に、ディレクトリ名を付加しないでください。付加した場合の動作は不定です。</li> <li>存在しない&lt;ファイル名&gt;を指定した場合、ダイアログボックスの場所欄に表示されるファイル名は、空欄にクリアされます。</li> <li>ダイアログ表示中に、「Cancel」などでファイルパス選択をキャンセルした場合、得られる文字列は空文字列です。</li> </ul>	
注 意	<ul style="list-style-type: none"> <li>ウィンドウ番号：0で親が無い状態でダイアログを表示させると、「Gtk-Message: GtkDialog mapped without a transient parent. This is discouraged.」のような警告が出力されますが、使用には差し支えありません。</li> </ul>	
使用例1	PATH\$ = GUDIALOGBOX OPEN\$(0, "/home/user")  <ウィンドウ番号> 0で親ウィンドウなし、<ディレクトリ> /home/user を最初に表示するディレクトリとして、ファイルを開くためのダイアログボックスを表示します。 選択したファイルパスを変数 PATH\$ へ代入します。	
使用例2	GUCREATE WINDOW 1	

	<p>GUSHOW 1 PATH\$ = GUDIALOGBOX OPEN\$(1, "/home/user")</p> <p>〈ウィンドウ番号〉1でウィンドウ1を親、〈ディレクトリ〉/home/user を最初に表示するディレクトリとして、ファイルを開くためのダイアログボックスを表示します。 選択したファイルパスを変数 PATH\$ へ代入します。</p>
使用例3	<p>PATH\$ = GUDIALOGBOX OPEN\$(0, "/home/user", "開くファイルを選択", ("すべてのファイル", "*"), "log.txt", "TEST")</p> <p>〈ウィンドウ番号〉0で親ウィンドウなし、〈ディレクトリ〉/home/user、〈タイトル〉開くファイルを選択、〈フィルタ説明〉"すべてのファイル"、〈フィルタ拡張子〉"*"、〈ファイル名〉log.txt、〈説明文〉TESTで、ファイルを開くためのダイアログボックスを表示します。 選択したファイルパスを変数 PATH\$ へ代入します。</p>
使用例4	<p>DIM A\$(1) DIM B\$(1) GUCREATE WINDOW 1 GUSHOW 1 A\$ = "すべてのファイル", "テキストファイル" B\$ = "*", "*.txt" PATH\$ = GUDIALOGBOX OPEN\$(1, "/home/user", "開くファイルを選択", (A\$, B\$), "log.txt", "TEST")</p> <p>〈ウィンドウ番号〉1でウィンドウ1を親、〈ディレクトリ〉/home/user、〈タイトル〉開くファイルを選択、フィルタを複数設定、〈ファイル名〉log.txt、〈説明文〉TESTで、ファイルを開くためのダイアログボックスを表示します。 選択したファイルパスを変数 PATH\$ へ代入します。</p>

## 3.3.1.49 GUDIALOGBOX SAVE\$

関数		
機 能	ファイルを保存するためのダイアログボックスを表示し、選択したファイルパスを取得します。	
書 式	<(戻り値)ファイルパス> = GUDIALOGBOX SAVE\$(<①ウィンドウ番号>, <②ディレクトリ> [, <③タイトル>, ( <④フィルタ説明>, <④フィルタ拡張子> ), <⑤ファイル名>, <⑥説明文> ] )	
パラ メータ	戻り値	<div> <div>&lt;ファイルパス&gt;</div> <div>文字列</div> </div> 表示したダイアログボックスで選択したファイルパスが得られます。
	①	<div> <div>&lt;ウィンドウ番号&gt;</div> <div>数値</div> </div> 親に設定するウィンドウ番号を指定します。 0を指定すると親が無い状態でダイアログボックスを表示します
	②	<div> <div>&lt;ディレクトリ&gt;</div> <div>文字列</div> </div> 最初に表示するディレクトリを指定します。
	③	<div> <div>&lt;タイトル&gt;</div> <div>文字列</div> </div> タイトルを指定します。 省略するとタイトルは「保存」が設定されます。
	④	<div> <div>&lt;フィルタ説明&gt;, &lt;フィルタ拡張子&gt;</div> <div>文字列</div> </div> 表示するファイルに対するフィルタを丸カッコで挟んで指定します。 フィルタの説明と拡張子を文字列で指定します。 フィルタを複数指定する場合は配列にて指定します。 省略するとすべてのファイルが表示される状態となります。
	⑤	<div> <div>&lt;ファイル名&gt;</div> <div>文字列</div> </div> ファイル名入力フォームへ最初に表示する、存在するファイル名を指定します。 省略するとファイル名は入力されません。
	⑥	<div> <div>&lt;説明文&gt;</div> <div>文字列</div> </div> ダイアログボックスへ表示したい任意の文字列を指定します。 省略すると説明文は表示されません。
	備 考	<ul style="list-style-type: none"> <li>・親ウィンドウを設定すると、ダイアログボックスが表示されている間は親のウィンドウを操作することが出来ません。</li> <li>・フィルタを複数指定した場合に、説明と拡張子の配列要素数は同一にしてください。</li> <li>・存在しない&lt;ディレクトリ&gt;を指定すると、エラーを検知します。</li> <li>・&lt;フィルタ拡張子&gt;の指定は、Linuxのlsコマンドでのワイルドカード指定と同じです。 例えば、「*.txt」とすると、拡張子が「.txt」のファイル全てにマッチします。 大文字・小文字関係なくマッチさせたい場合は、「*. [tT] [xX] [tT]」と記述します。</li> <li>・&lt;ファイル名&gt;で指定したファイル名の拡張子と、&lt;フィルタ拡張子&gt;で指定した拡張子を一致させるようにしてください。拡張子が一致しない場合、ダイアログボックス中の、拡張子のリスト選択は不定です。</li> <li>・&lt;ファイル名&gt;に、ディレクトリ名を付加しないでください。付加した場合の動作は不定です。</li> <li>・&lt;ファイル名&gt;には、存在するファイル名を指定してください。存在しないファイル名を指定すると、ダイアログのパス欄には表示されません。</li> <li>・ダイアログ表示中に、「Cancel」などでファイルパス入力をキャンセルした場合、得られる文字列は空文字列です。</li> </ul>
注 意	<ul style="list-style-type: none"> <li>・ウィンドウ番号：0で親が無い状態でダイアログを表示させると、「Gtk-Message: GtkDialog mapped without a transient parent. This is discouraged.」のような警告が出力されますが、使用には差し支えありません。</li> </ul>	
使用例1	PATH\$ = GUDIALOGBOX SAVE\$(0, "/home/user")  <ウィンドウ番号> 0で親ウィンドウなし、<ディレクトリ> /home/user を最初に表示するディレクトリとして、ファイルを保存するためのダイアログボックスを表示します。 選択したファイルパスを変数PATH\$へ代入します。	

使用例2	<p>GUCREATE WINDOW 1 GUSHOW 1 PATH\$ = GUDIALOGBOX SAVE\$(1, "/home/user")</p> <p>〈ウィンドウ番号〉 1でウィンドウ1を親、〈ディレクトリ〉 /home/user を最初に表示するディレクトリ、ファイルを保存するためのダイアログボックスを表示します。 選択したファイルパスを変数 PATH\$ へ代入します。</p>
使用例3	<p>PATH\$ = GUDIALOGBOX SAVE\$(0, "/home/user", "保存するファイルを指定", ("すべてのファイル", "*"), "log.txt", "TEST")</p> <p>〈ウィンドウ番号〉 0で親ウィンドウなし、〈ディレクトリ〉 /home/user、〈タイトル〉 保存するファイルを指定、〈フィルタ説明〉 "すべてのファイル"、〈フィルタ拡張子〉 "*"、〈ファイル名〉 log.txt、〈説明文〉 TESTで、ファイルを保存するためのダイアログボックスを表示します。 選択したファイルパスを変数 PATH\$ へ代入します。</p>
使用例4	<p>DIM A\$(1) DIM B\$(1) GUCREATE WINDOW 1 GUSHOW 1 A\$ = "すべてのファイル", "テキストファイル" B\$ = "*", "*.txt" PATH\$ = GUDIALOGBOX SAVE\$(1, "/home/user", "保存するファイルを指定", (A\$, B\$), "log.txt", "TEST")</p> <p>〈ウィンドウ番号〉 1でウィンドウ1を親、〈ディレクトリ〉 /home/user、〈タイトル〉 保存するファイルを指定、〈フィルタ説明／拡張子〉フィルタを複数設定、〈ファイル名〉 log.txt、〈説明文〉 TESTで、ファイルを開くためのダイアログボックスを表示します。 選択したファイルパスを変数 PATH\$ へ代入します。</p>

## 3.3.1.50 GUDIALOGBOX DIRECTORY\$


関数			
機 能	ディレクトリを選択するためのダイアログボックスを表示し、選択したディレクトリパスを取得します。		
書 式	<(戻り値)ディレクトリパス> = GUDIALOGBOX DIRECTORY\$(<①ウィンドウ番号>, <②ディレクトリ> [, <③タイトル>, <④説明文> ] )		
戻り値	戻り値	<ディレクトリパス>	文字列
	表示したダイアログボックスで選択したディレクトリパスが得られます。		
パラメータ	①	<ウィンドウ番号>	数値
	親に設定するウィンドウ番号を指定します。 0を指定すると親が無い状態でダイアログボックスを表示します		
	②	<ディレクトリ>	文字列
	最初に表示するディレクトリを指定します。		
	③	<タイトル>	文字列
	タイトルを指定します。 省略するとタイトルは「ディレクトリを開く」が設定されます。		
	④	<説明文>	文字列
	ダイアログボックスへ表示したい任意の文字列を指定します。 省略すると説明文は表示されません。		
備 考	<ul style="list-style-type: none"> <li>親ウィンドウを設定すると、ダイアログボックスが表示されている間は親のウィンドウを操作することが出来ません。</li> <li>存在しない&lt;ディレクトリ&gt;を指定すると、エラーとなります。</li> <li>ダイアログ表示中に、「Cancel」などでディレクトリパス入力をキャンセルした場合、得られる文字列は空文字列です。</li> </ul>		
注 意	<ul style="list-style-type: none"> <li>ウィンドウ番号：0で親が無い状態でダイアログを表示させると、「Gtk-Message: GtkDialog mapped without a transient parent. This is discouraged.」のような警告が出力されますが、使用には差し支えありません。</li> </ul>		
使用例1	PATH\$ = GUDIALOGBOX DIRECTORY\$(0, "/home/user")  <ウィンドウ番号> 0で親ウィンドウなし、<ディレクトリ> /home/user を最初に表示するディレクトリとして、ディレクトリを選択するためのダイアログボックスを表示します。 選択したディレクトリパスを変数 PATH\$ へ代入します。		
使用例2	GUCREATE WINDOW 1 GUSHOW 1 PATH\$ = GUDIALOGBOX DIRECTORY\$(1, "/home/user")  <ウィンドウ番号> 1でウィンドウ1を親、<ディレクトリ> /home/user を最初に表示するディレクトリとして、ディレクトリを選択するためのダイアログボックスを表示します。 選択したディレクトリパスを変数 PATH\$ へ代入します。		
使用例3	PATH\$ = GUDIALOGBOX DIRECTORY\$(0, "/home/user", "ディレクトリを指定", "TEST")  <ウィンドウ番号> 0で親ウィンドウなし、<ディレクトリ> /home/user、<タイトル> ディレクトリを指定、<説明文> TESTで、ディレクトリを選択するためのダイアログボックスを表示します。 選択したディレクトリパスを変数 PATH\$ へ代入します。		
使用例4	GUCREATE WINDOW 1 GUSHOW 1 PATH\$ = GUDIALOGBOX DIRECTORY\$(1, "/home/user", "ディレクトリを指定", "TEST")  <ウィンドウ番号> 1でウィンドウ1を親、<ディレクトリ> /home/user、<タイトル> ディレクトリを指定、<説明文> TESTで、ディレクトリを選択するためのダイアログボックスを表示します。		



	選択したディレクトリパスを変数 PATH\$ へ代入します。
--	--------------------------------

### 3.3.1.51 GUNOTIFY SEND

命令

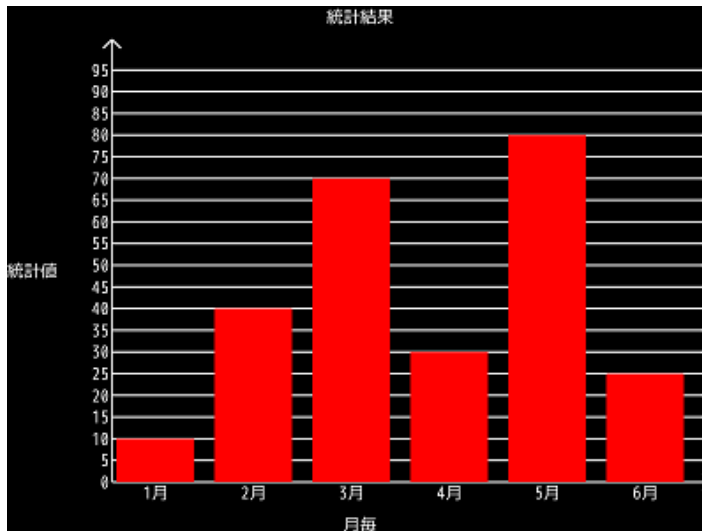
機能書式	指定したメッセージを、デスクトップ通知します。										
	GUNOTIFY SEND <①タイトル>, <②本文> [, <③オプション配列> ]										
パラメータ	①	<タイトル>	文字列								
	デスクトップ通知する際のタイトルを指定します。										
	②	<本文>	文字列								
	デスクトップ通知する際の本文を指定します。										
	③	<オプション配列>	文字列								
デスクトップ通知する際に、細かい挙動をオプション設定します。											
<table><tr><th>オプション</th><th>挙動</th></tr><tr><td>"ICON=&lt;文字列&gt;"</td><td>アイコン名を指定します。 指定可能なアイコン名は、「Icon Naming Specification」 &lt;<a href="https://specifications.freedesktop.org/icon-naming-spec/icon-naming-spec-latest.html">https://specifications.freedesktop.org/icon-naming-spec/icon-naming-spec-latest.html</a>&gt; を参考ください。 何も指定しないと、アイコンは無しです。</td></tr><tr><td>"TIMEOUT_MS=&lt;数値&gt;"</td><td>デスクトップ通知されるメッセージボックスの表示時間をms単位で指定します。 何も指定しないと、OSに任せられます。</td></tr><tr><td>"IMAGE=&lt;文字列&gt;"</td><td>アイコンの代わりとなる画像ファイルのパスを指定します。 "ICON"指定時は、このオプションを指定しないでください。 このオプションを指定した時、実際の表示はアイコンと同じ大きさまで縮小されます。</td></tr></table>				オプション	挙動	"ICON=<文字列>"	アイコン名を指定します。 指定可能なアイコン名は、「Icon Naming Specification」 < <a href="https://specifications.freedesktop.org/icon-naming-spec/icon-naming-spec-latest.html">https://specifications.freedesktop.org/icon-naming-spec/icon-naming-spec-latest.html</a> > を参考ください。 何も指定しないと、アイコンは無しです。	"TIMEOUT_MS=<数値>"	デスクトップ通知されるメッセージボックスの表示時間をms単位で指定します。 何も指定しないと、OSに任せられます。	"IMAGE=<文字列>"	アイコンの代わりとなる画像ファイルのパスを指定します。 "ICON"指定時は、このオプションを指定しないでください。 このオプションを指定した時、実際の表示はアイコンと同じ大きさまで縮小されます。
オプション	挙動										
"ICON=<文字列>"	アイコン名を指定します。 指定可能なアイコン名は、「Icon Naming Specification」 < <a href="https://specifications.freedesktop.org/icon-naming-spec/icon-naming-spec-latest.html">https://specifications.freedesktop.org/icon-naming-spec/icon-naming-spec-latest.html</a> > を参考ください。 何も指定しないと、アイコンは無しです。										
"TIMEOUT_MS=<数値>"	デスクトップ通知されるメッセージボックスの表示時間をms単位で指定します。 何も指定しないと、OSに任せられます。										
"IMAGE=<文字列>"	アイコンの代わりとなる画像ファイルのパスを指定します。 "ICON"指定時は、このオプションを指定しないでください。 このオプションを指定した時、実際の表示はアイコンと同じ大きさまで縮小されます。										
使用例	GUNOTIFY SEND "Interfaceより", "こんにちはAJANです", [ "ICON=dialog-information"; "TIMEOUT_MS=3000" ]										
画面右上に、下図のようなデスクトップ通知のメッセージが表示されます。											
											

## 3.4 グラフに関する関数・命令

### 3.4.1 グラフに関する個別説明

#### 3.4.1.1 棒グラフ (BARChart)

棒グラフコマンドは、以下のようなグラフ図像を描画する機能です。



プログラムコード例：

```
OPTION GUI      ' GUIを使用する際に必須です

' ウィンドウを作成する
FONT$ = "32, 0, 0, 14, 8, 1"
SIZE$ = "800, 600"
POSI$ = "0, 0"
TIP$ = "ウィンドウ"
GUCREATE WINDOW 1, "グラフ", 1, FONT$, SIZE$, POSI$, TIP$

DIM ARY(5)
ARY = [ 10; 40; 70; 30; 80; 25 ]
DIM LBL$(5)
LBL$ = [ "1月"; "2月"; "3月"; "4月"; "5月"; "6月" ]

' 棒グラフ領域を追加
GUCREATE BARChart 1, 10, FONT$, SIZE$, POSI$, TIP$
' 棒グラフのタイトルを設定
GUADD BARChart TITLE 10, "統計結果"
' 棒グラフの縦軸名、横軸名を追加
GUUPD BARChart LABEL 10, "統計値", "月毎"
' 棒グラフにデータを登録する
GUADD BARChart VALUE 10, (ARY), (2)
' 棒グラフにデータラベルを登録する
GUADD BARChart DATALABEL 10, (LBL$)
' 棒グラフの描画
GUUPD BARChart DRAW 10

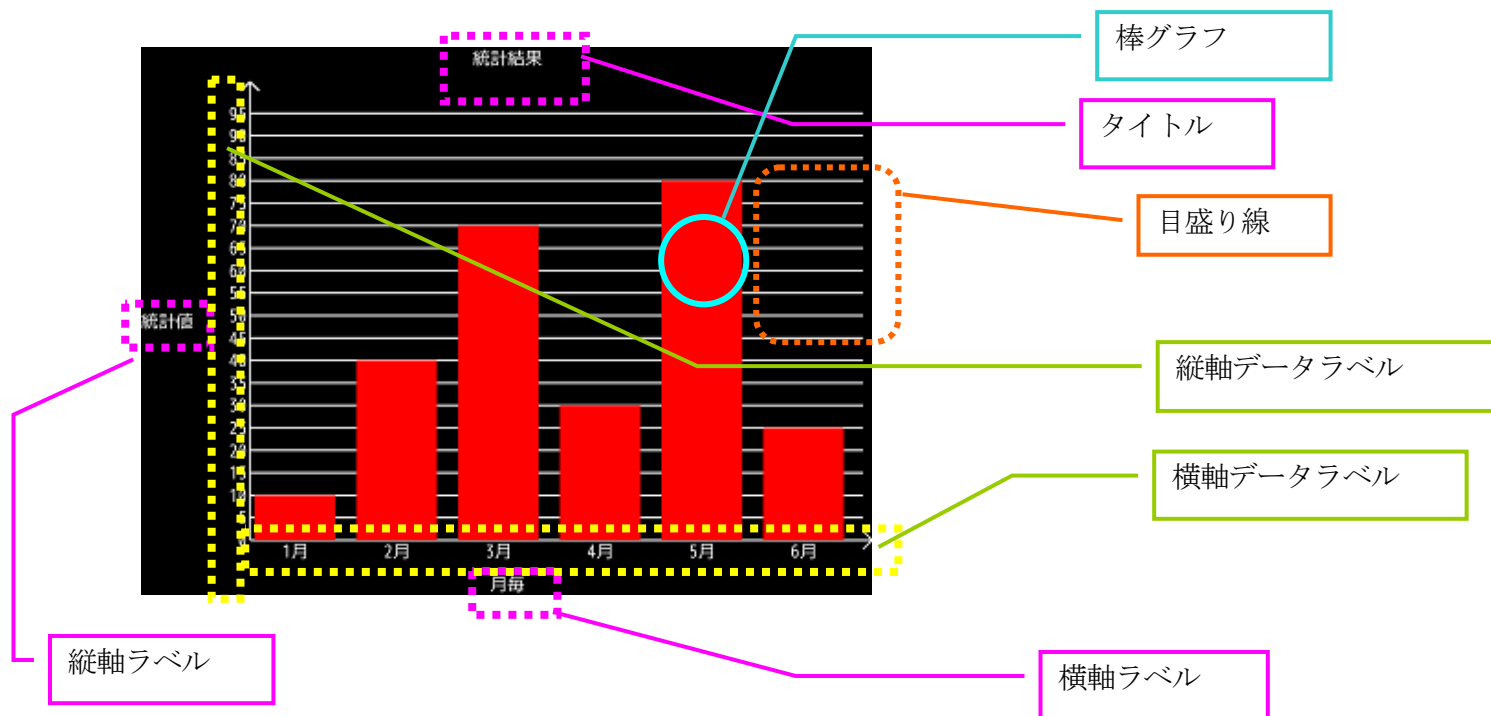
' ウィンドウの表示
GUSHOW 1

GUMAIN LOOP      ' メインイベントループが回ります
```

棒グラフの描画には、複数のコマンドを使用します。  
GUCREATE BARCHARTコマンドでグラフを作成した後、GUADD BARCHART VALUEコマンドなどでデータの追加  
や、GUUPD BARCHART RANGEコマンドなどで細かい設定を行い、  
GUUPD BARCHART DRAWコマンドで、グラフを描画します。

グラフの作成	GUCREATE BARCHART コマンド
↓	
データの追加	GUADD BARCHART VALUE コマンド
データの上書き	GUUPD BARCHART VALUE コマンド
データラベルの追加	GUADD BARCHART DATALABEL コマンド
データラベルの上書き	GUUPD BARCHART DATALABEL コマンド
表示範囲の設定	GUADD BARCHART RANGE コマンド
表示範囲の上書き設定	GUUPD BARCHART RANGE コマンド
タイトルの設定	GUADD BARCHART TITLE コマンド
ラベルの設定	GUUPD BARCHART LABEL コマンド
目盛り間隔の設定	GUUPD BARCHART GRID コマンド
↓	
グラフの描画	GUUPD BARCHART DRAW コマンド

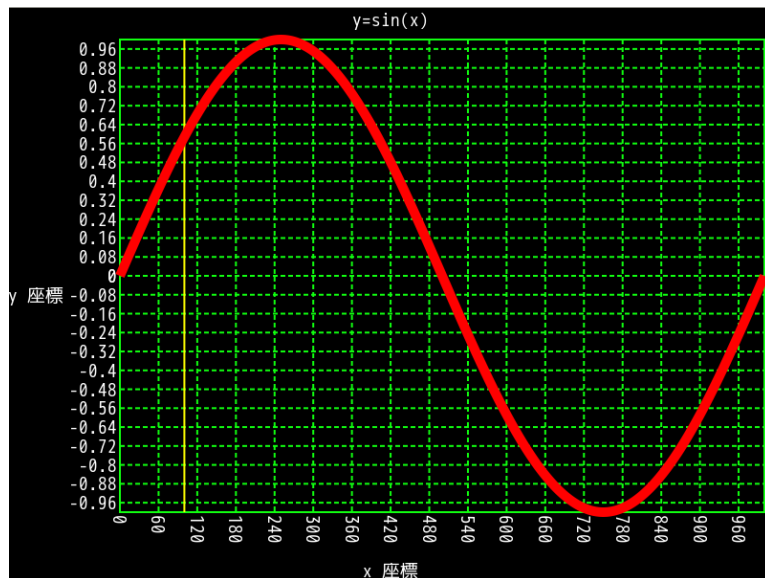
次に棒グラフの各部名称と、それを指定する為のコマンドの対応について簡単に説明します。



名称	解説
棒グラフ	1次元の数値配列を指定する事で、左(配列の添字の小さい方)から右(配列の添字の大きい方)に要素毎に、数値を縦の棒グラフとして並べて描画します。 グラフの追加は、GUADD BARCHART VALUE で、 グラフの更新は、GUUPD BARCHART VALUE で行います。
目盛り線	棒グラフを読み取りやすくするために、一定間隔毎に目盛り線を描画します。 棒グラフ描画時に、自動的に目盛り線が描画されます。 GUUPD BARCHART GRID で任意に指定することもできます。
縦軸データラベル	棒グラフの数値配列の各数値が、グラフの縦軸に相当します。 縦軸に対して目盛り線毎に、数値をデータラベルとして表示する事で、読み取りやすいよう表示します。 GUUPD BARCHART LABEL で縦軸ラベルを描画するように指定すると、同時にこのラベルが表示されるようになります。何も指定しない場合は、データラベルは表示されません。
横軸データラベル	GUUPD BARCHART LABEL または GUUPD BARCHART DATALABEL で、棒グラフの数値配列の添字が、グラフの横軸に表示されます。 何も指定しない場合は、データラベルは表示されません。
タイトル	グラフに対するタイトル名を表示できます。 GUADD BARCHART TITLE で指定可能です。何も指定しない場合は、タイトルは表示されません。
縦軸ラベル、横軸ラベル	グラフに対する横軸および縦軸を解説する為のラベルを表示できます。 GUUPD BARCHART LABEL で指定可能です。何も指定しない場合は、ラベルは表示されません。

### 3.4.1.2 折れ線グラフ (LINEGRAPH)

折れ線グラフコマンドは、以下のようなグラフ画像を描画する機能です。



プログラムコード例：

```
PTION GUI      ' GUI を使用する際に必須です

PI = 3.14159265359
' サインカーブのデータを配列に取得する
DIM SIN_VALUE(1000)
FOR I = 0 TO 1000
    SIN_VALUE(I) = SIN(2*PI*(I/1000))
NEXT I

' ウィンドウを作成する
FONT$ = "32, 0, 0, 14, 8, 1"
SIZE$ = "800, 600"
POSI$ = "0, 0"
TIP$ = "ウィンドウ"
GUCREATE WINDOW 1, "グラフ", 1, FONT$, SIZE$, POSI$, TIP$
' 折れ線グラフ領域を追加
GUCREATE LINEGRAPH 1, 10, FONT$, SIZE$, POSI$, TIP$
' 折れ線グラフのタイトルを設定
GUADD LINEGRAPH TITLE 10, "y=sin(x)"
' 折れ線グラフの縦軸名、横軸名を追加
GUUPD LINEGRAPH LABEL 10, "y 座標", "x 座標"
' 折れ線グラフにデータを登録する
GUADD LINEGRAPH VALUE 10, (SIN_VALUE), 2, 10
' 折れ線グラフに補助線を追加する
GUADD LINEGRAPH GUAGE 10, [1, 100, 4]

' グラフの描画
GUUPD LINEGRAPH DRAW 10
' ウィンドウの表示
GUSHOW 1

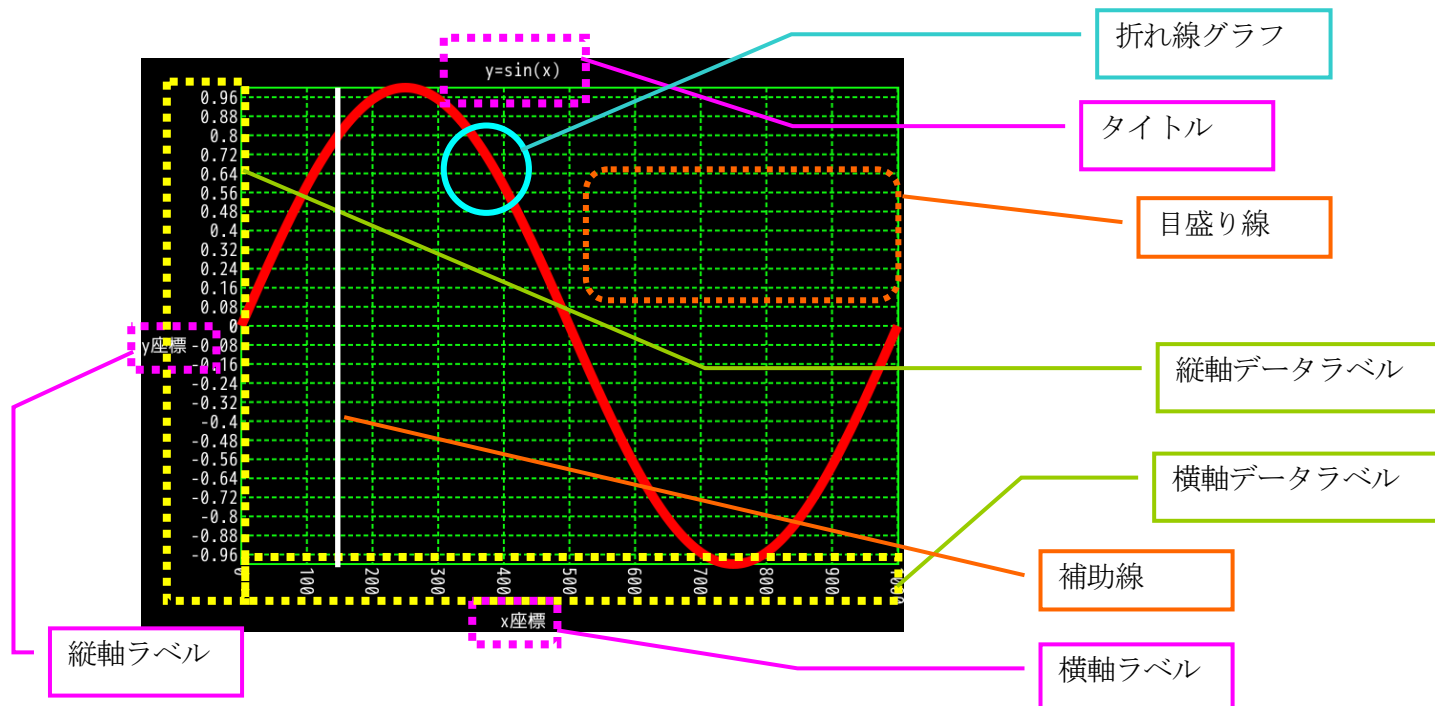
GUMAIN LOOP      ' メインイベントループが回ります
```

折れ線グラフの描画には、複数のコマンドを使用します。

GUCREATE LINEGRAPHコマンドでグラフを作成した後、GUADD LINEGRAPH VALUEコマンドなどでデータの追加や、GUUPD LINEGRAPH RANGEコマンドなどで細かい設定を行い、GUUPD LINEGRAPH DRAWコマンドで、グラフを描画します。

グラフの作成	GUCREATE LINEGARPH コマンド
↓	
データの追加	GUADD LINEGRAPH VALUE コマンド
データの更新	GUUPD LINEGRAPH VALUE コマンド
データラベルの追加	GUADD LINEGRAPH DATALABEL コマンド
データラベルの更新	GUUPD LINEGRAPH DATALABEL コマンド
表示範囲の追加	GUADD LINEGRAPH RANGE コマンド
表示範囲の更新	GUUPD LINEGRAPH RANGE コマンド
タイトルの追加	GUADD LINEGRAPH TITLE コマンド
ラベルの更新	GUUPD LINEGRAPH LABEL コマンド
目盛り間隔の更新	GUUPD LINEGRAPH GRID コマンド
補助線の更新	GUUPD LINEGRAPH GUAGE コマンド
↓	
グラフの描画	GUUPD LINEGRAPH DRAW コマンド

次に折れ線グラフの各部名称と、それを指定する為のコマンドの対応について簡単に説明します。

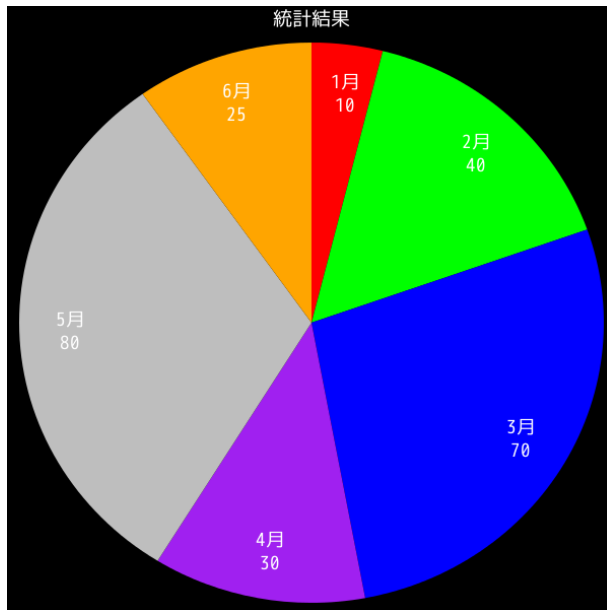


名称	解説
折れ線グラフ	<p>1 次元の数値配列を指定する事で、左(配列の添字の小さい方)から右(配列の添字の大きい方)に折れ線グラフを描画します。</p> <p>グラフの追加は、GUADD LINEGRAPH VALUE で、 グラフの更新は、GUUPD LINEGRAPH VALUE で行います。</p> <p>GUADD / GUUPD LINEGRAPH VALUE で、同じ要素数の 1 次元配列を複数列挙指定すると、複数の折れ線グラフを描画できます。</p>
目盛り線	<p>折れ線グラフを読み取りやすくするために、一定間隔毎に点線付きの目盛り線を描画します。</p> <p>折れ線グラフ描画時に、ある程度自動的に目盛り線が描画されます。</p> <p>GUUPD LINEGRAPH GRID で任意の間隔に指定することもできます。</p>
補助線	<p>折れ線グラフのみに描画できる垂直線および水平線で、折れ線グラフの任意の位置の数値を読み取りやすくする補助に使えます。</p>
縦軸データラベル	<p>GUUPD LINEGRAPH LABEL で縦軸ラベルを描画するように指定すると、折れ線グラフの数値配列の各数値を、グラフの縦軸に表示します。</p> <p>何も指定しない場合は、データラベルは表示されません。</p>
横軸データラベル	<p>GUUPD LINEGRAPH LABEL で横軸ラベルを描画するように指定すると、折れ線グラフの数値配列の添字を表示します。</p> <p>何も指定しない場合は、データラベルは表示されません。</p> <p>また、配列の添字ではなく別のデータラベルを表示したい場合は、GUADD LINEGRAPH DATALABEL および GUUPD LINEGRAPH DATALABEL により指定できます。</p> <p>任意の文字を表示できるため、「IDX10」「IDX50」、…という具合に文字列付きのデータラベルを表示する事ができます。</p>
タイトル	<p>GUADD LINEGRAPH TITLE でグラフに対するタイトル名を表示できます。</p> <p>何も指定しない場合は、タイトルは表示されません。</p>
縦軸ラベル、横軸ラベル	<p>GUUPD LINEGRAPH LABEL でグラフに対する横軸および縦軸を解説する為のラベルを表示できます。何も指定しない場合は、ラベルは表示されません。</p> <p>ラベル表示を有効にすると自動的にデータラベルも表示されます。</p>



### 3.4.1.3 円グラフ (PIECHART)

円グラフは、以下のようなグラフ図像を描画する機能です。



プログラムコード例：

```
OPTION GUI      ' GUIを使用する際に必須です

' ウィンドウを作成する
FONT$ = "32, 0, 0, 14, 8, 1"
SIZE$ = "800, 600"
POSI$ = "0, 0"
TIP$ = "ウィンドウ"
GUCREATE WINDOW 1, "グラフ", 1, FONT$, SIZE$, POSI$, TIP$

DIM ARY(5)
ARY = [ 10; 40; 70; 30; 80; 25 ]
DIM COL(5)
COL = [ 2; 3; 5; 6; 9; 11 ]
DIM LBL$(5)
LBL$ = [ "1月"; "2月"; "3月"; "4月"; "5月"; "6月" ]

' 円グラフ領域を追加
GUCREATE PIECHART 1, 10, FONT$, SIZE$, POSI$, TIP$
' 円グラフのタイトルを設定
GUADD PIECHART TITLE 10, "統計結果"
' 円グラフにデータを登録する
GUADD PIECHART VALUE 10, ( ARY ), COL
' 円グラフにデータラベルを登録する
GUADD PIECHART DATALABEL 10, (LBL$)
' 円グラフの描画
GUUPD PIECHART DRAW 10

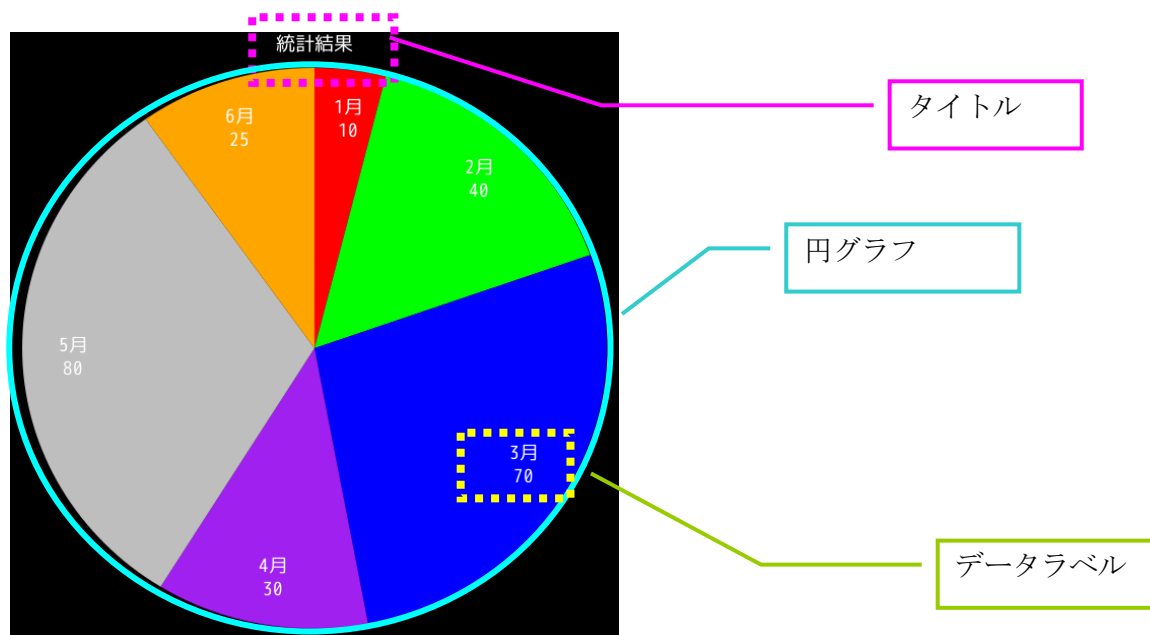
' ウィンドウの表示
GUSHOW 1

GUMAIN LOOP      ' メインイベントループが回ります
```

円グラフの描画には、複数のコマンドを使用します。  
GUCREATE PIECHARTコマンドでグラフを作成した後、GUADD PIECHART VALUEコマンドなどでデータの追加  
や、GUUPD PIECHART TITLEコマンドなどで細かい設定を行い、  
GUUPD PIECHART DRAWコマンドで、グラフを描画します。

グラフの作成	GUCREATE PIECHART コマンド
↓	
データの追加	GUADD PIECHART VALUE コマンド
データの更新	GUUPD PIECHART VALUE コマンド
データラベルの追加	GUADD PIECHART DATALABEL コマンド
データラベルの更新	GUUPD PIECHART DATALABEL コマンド
タイトルの追加	GUADD PIECHART TITLE コマンド
↓	
グラフの描画	GUUPD PIECHART DRAW コマンド

次に円グラフの各部名称と、それを指定する為のコマンドの対応について簡単に説明します。



名称	解説
円グラフ	1次元の数値配列を指定する事で、円の上部(配列の添字の小さい方)から時計回り(配列の添字の大きい方)に要素毎に、数値を比率でパイ状に区切られた円グラフとして描画します。 数値配列の追加は、GUADD PIECHART VALUE で、 数値配列の更新は、GUUPD PIECHART VALUEで行います。
データラベル	GUADD PIECHART DATALABEL で円グラフを構成する数値配列の各データ領域に対して、具体的な数値と項目説明を表示することができます。 GUUPD PIECHART DATALABEL で項目説明を更新することもできます。 何も指定しないと、データラベルは表示されません。
タイトル	GUADD PIECHART TITLE でグラフに対するタイトル名を表示できます。 何も指定しないと、タイトルは表示されません。

### 3.4.2 コマンド個別説明

#### 3.4.2.1 GUCREATE BARCHART

命令		
機能	指定したウィンドウに棒グラフを配置します。	
書式	GUCREATE BARCHART <①ウィンドウ番号>, <②棒グラフ番号> [, [ <③フォント設定> ], [ <④サイズ設定> ], [ <⑤ポジション設定> ], [ <⑥ツールチップ設定> ] ]	
	※ フォント設定以降の引数を省略する場合、各引数間のカンマは必要です。	
パラメータ	①	<ウィンドウ番号> 数値
	棒グラフを配置するウィンドウの番号を指定します。	
	②	<棒グラフ番号> 数値
	作成する棒グラフの番号を指定します。 1～10000の値を指定してください。	
	③	<フォント設定> 文字列
	GUI部品に表示される文字列のフォントを指定します。(参照：P. 20)	
	④	<サイズ設定> 文字列
	GUI部品に表示されるGUI部品の大きさを指定します。(参照：P. 22)	
	⑤	<ポジション設定> 文字列
	GUI部品に表示されるGUI部品の表示場所を指定します。(参照：P. 23)	
	⑥	<ツールチップ設定> 文字列
使用例1	マウスポインタがGUI部品上に来た際に、表示されるツールチップの内容を指定します。 (参照：P. 24)	
使用例2	GUCREATE WINDOW 1 GUCREATE BARCHART 1, 10	
	<ウィンドウ番号> 1 のウィンドウに <グラフ番号> 10の棒グラフを配置します。	
	GUCREATE WINDOW 1 GUCREATE BARCHART 1, 10, FONT\$, SIZE\$, POSI\$, TIP\$	
	<ウィンドウ番号> 1 のウィンドウに <グラフ番号> 10の棒グラフを配置します。 共通項目でフォントやサイズ、位置、ツールチップの表示内容を設定しています。	

## 3.4.2.2 GUADD BARCHART VALUE

命令		
機 能	棒グラフに値を追加します。	
書 式	GUADD BARCHART VALUE <①棒グラフ番号>, ( <②値配列> ) [, ( <③色配列> ) ]	
パラ メータ	①	数値
	<棒グラフ番号>	
	項目追加を行う棒グラフ番号を指定します。	
	②	配列
	<値配列>	
	描画する数値の1次元配列を丸カッコで挟んで指定します。	
	③	配列
	<色配列>	
	線の色を丸カッコで挟んで指定します。	
	<値配列>で指定した本数と同数の要素数を持つ配列へ、各色の数値を与えてください。	
	色番号は「フォント設定」の文字色の項を参照してください。	
備 考	・グラフ領域の生成 (→「3.4.2.1 GUCREATE BARCHART」) が必要です。	
使用例	<p>GUCREATE WINDOW 1  GUCREATE BARCHART 1, 10, , "400,300",,  GUADD BARCHART TITLE 10, "タイトル"  LBL\$ = ["1月"; "2月"; "3月"]  GUADD BARCHART DATALABEL 10, (LBL\$)  VAL1 = [10; 30; 20]  GUADD BARCHART VALUE 10, (VAL1), (2)  GUADD BARCHART RANGE 10, 0, 50  GUUPD BARCHART DRAW 10  GUSHOW 1</p> <p>&lt;ウィンドウ番号&gt; 1 のウィンドウに対して棒グラフを配置します。  タイトルの追加、データラベルの追加、縦軸の最大/最小値の追加、値の追加を行いグラフを描画します。</p>	

## 3.4.2.3 GUUPD BARCHART VALUE

命令		
機 能	棒グラフの値を更新します。	
書 式	GUUPD BARCHART VALUE <①棒グラフ番号>, ( <②値配列> ) [, ( <③色配列> ) ]	
パラ メータ	①	数値
	<棒グラフ番号>	
	項目追加を行う棒グラフ番号を指定します。	
	②	配列
	<値配列>	
	描画する数値の1次元配列を丸カッコで挟んで指定します。	
	③	配列
	<色配列>	
	線の色を丸カッコで挟んで指定します。	
	色番号は「3.2.1 フォント設定」の文字色の項を参照してください。	
備 考	・グラフ領域の生成 (→「3.4.2.1 GUCREATE BARCHART」) が必要です。	
使用例	<p>GUCREATE WINDOW 1  GUCREATE BARCHART 1, 10, , "400,300",,  ...  GUUPD BARCHART DRAW 10  GUSHOW 1  ...  LBL\$ = ["4月"; "5月"; "6月"]  GUUPD BARCHART DATALABEL 10, (LBL\$)  GUUPD BARCHART GRID 10, 10  GUUPD BARCHART LABEL 10, "縦軸", "横軸"</p>	

	<p>GUUPD BARCHART RANGE 10, 0, 30</p> <p>VAL1 = [30; 10; 20]</p> <p>GUUPD BARCHART VALUE 10, (VAL1), (2)</p> <p>GUUPD BARCHART DRAW 10</p> <p>描画した棒グラフに対して各種データを更新します。</p>
--	--

### 3.4.2.4 GUADD BARCHART TITLE

命令		
機 能	棒グラフにタイトルを追加します。	
書 式	GUADD BARCHART TITLE <①棒グラフ番号>, <②タイトル名> [, <③フォント設定> ]	
パラ メータ	①	<棒グラフ番号> 数値
	項目追加を行う棒グラフ番号を指定します。	
	②	<タイトル名> 文字列
	タイトルとして表示する文字列を指定します。	
	③	<フォント設定> 文字列
	GUI部品に表示される文字列のフォントを指定します。(参照 : P. 20)	
備 考	・ グラフ領域の生成 (→「3. 4. 2. 1 GUCREATE BARCHART」) が必要です。 ・ <フォント設定>を指定すると、任意のフォント名およびサイズ、文字色の文字をラベル名に指定できます。(背景色は反映されません)省略時、「GUCREATE BARCHART」で指定した設定が引き継がれます。	
使用例	GUCREATE WINDOW 1 GUCREATE BARCHART 1, 10,, "400,300",, GUADD BARCHART TITLE 10, "タイトル" LBL\$ = ["1月"; "2月"; "3月"] GUADD BARCHART DATALABEL 10, (LBL\$) VAL1 = [10; 30; 20] GUADD BARCHART VALUE 10, (VAL1), (2) GUADD BARCHART RANGE 10, 0, 50 GUUPD BARCHART DRAW 10 GUSHOW 1  <ウィンドウ番号> 1 のウィンドウに対して棒グラフを配置します。 タイトルの追加、データラベルの追加、縦軸の最大/最小値の追加、値の追加を行いグラフを描画します。	

## 3.4.2.5 GUUPD BARCHART LABEL

命令		
機能	棒グラフの軸ラベルを更新します。	
書式	GUUPD BARCHART LABEL <①棒グラフ番号>, <②縦軸名>, <②横軸名> [, <③フォント設定> ]	
パラメータ	①	数値
	<棒グラフ番号> 項目追加を行う棒グラフ番号を指定します。	
	②	文字列
	<縦軸名>, <横軸名> グラフに対する縦軸/横軸を解説する為のラベルを文字列で指定します。	
	③	文字列
	<フォント設定> GUI部品に表示される文字列のフォントを指定します。(参照 : P. 20)	
備考	<ul style="list-style-type: none"> <li>・グラフ領域の生成 (→「3.4.2.1 GUCREATE BARCHART」) が必要です。</li> <li>・&lt;フォント設定&gt;を指定すると、任意のフォント名およびサイズ、文字色の文字をラベル名に指定できます。(背景色は反映されません)</li> </ul> 省略時、GUCREATE BARCHART で指定した設定が引き継がれます。	
使用例	<pre> GUCREATE WINDOW 1 GUCREATE BARCHART 1, 10, , "400, 300", , ... GUUPD BARCHART DRAW 10 GUSHOW 1 ... LBL\$ = ["4月"; "5月"; "6月"] GUUPD BARCHART DATALABEL 10, (LBL\$) GUUPD BARCHART GRID 10, 10 GUUPD BARCHART LABEL 10, "縦軸", "横軸" GUUPD BARCHART RANGE 10, 0, 30 VAL1 = [30; 10; 20] GUUPD BARCHART VALUE 10, (VAL1), (2) GUUPD BARCHART DRAW 10           </pre> 描画した棒グラフに対して各種データを更新します。	

## 3.4.2.6 GUADD BARCHART DATALABEL

命令			
機 能	棒グラフにデータラベルを追加します。		
書 式	GUADD BARCHART DATALABEL <①棒グラフ番号>, ( <②横軸データラベル配列> )		
パラ メータ	①	<棒グラフ番号>	数値
	項目追加を行う棒グラフ番号を指定します。		
	②	<横軸データラベル配列>	配列
	横軸に対して任意に表示したいデータラベル文字列の1次元配列を丸カッコで挟んで指定します。		
備 考	・グラフ領域の生成 (→「3.4.2.1 GUCREATE BARCHART」) が必要です。		
使用例	<p>GUCREATE WINDOW 1  GUCREATE BARCHART 1, 10, , "400,300",,  GUADD BARCHART TITLE 10, "タイトル"  LBL\$ = ["1月"; "2月"; "3月"]  GUADD BARCHART DATALABEL 10, (LBL\$)  VAL1 = [10; 30; 20]  GUADD BARCHART VALUE 10, (VAL1), (2)  GUADD BARCHART RANGE 10, 0, 50  GUUPD BARCHART DRAW 10  GUSHOW 1</p> <p>&lt;ウィンドウ番号&gt; 1 のウィンドウに対して棒グラフを配置します。  タイトルの追加、データラベルの追加、縦軸の最大/最小値の追加、値の追加を行いグラフを描画します。</p>		



## 3.4.2.7 GUUPD BARCHART DATALABEL

命令			
機 能	棒グラフのデータラベルを更新します。		
書 式	GUUPD BARCHART DATALABEL <①棒グラフ番号>, ( <②横軸データラベル配列> )		
パラ メータ	①	<棒グラフ番号>	数値
	項目追加を行う棒グラフ番号を指定します。		
	②	<横軸データラベル配列>	配列
	横軸に対して任意に表示したいデータラベル文字列の1次元配列を丸カッコで挟んで指定します。		
備 考	・グラフ領域の生成 (→「3.4.2.1 GUCREATE BARCHART」) が必要です。		
使用例	GUCREATE WINDOW 1 GUCREATE BARCHART 1, 10, , "400,300",, ... GUUPD BARCHART DRAW 10 GUSHOW 1 ... LBL\$ = ["4月"; "5月"; "6月"] GUUPD BARCHART DATALABEL 10, (LBL\$) GUUPD BARCHART GRID 10, 10 GUUPD BARCHART LABEL 10, "縦軸", "横軸" GUUPD BARCHART RANGE 10, 0, 30 VAL1 = [30; 10; 20] GUUPD BARCHART VALUE 10, (VAL1), (2) GUUPD BARCHART DRAW 10  描画した棒グラフに対して各種データを更新します。		

## 3.4.2.8 GUADD BARCHART RANGE

命令			
機 能	棒グラフに軸の範囲を追加します。		
書 式	GUADD BARCHART RANGE <①棒グラフ番号>, <②縦軸最小値>, <②縦軸最大値>		
パラ メータ	①	<棒グラフ番号>	数値
	項目追加を行う棒グラフ番号を指定します。		
	②	<縦軸最大値>, <縦軸最小値>	数値
	縦軸の最大値/最小値を指定します。		
備 考	・グラフ領域の生成 (→「3.4.2.1 GUCREATE BARCHART」) が必要です。		
使用例	GUCREATE WINDOW 1 GUCREATE BARCHART 1, 10, , "400,300",, GUADD BARCHART TITLE 10, "タイトル" LBL\$ = ["1月"; "2月"; "3月"] GUADD BARCHART DATALABEL 10, (LBL\$) VAL1 = [10; 30; 20] GUADD BARCHART VALUE 10, (VAL1), (2) GUADD BARCHART RANGE 10, 0, 50 GUUPD BARCHART DRAW 10 GUSHOW 1  <ウィンドウ番号> 1 のウィンドウに対して棒グラフを配置します。 タイトルの追加、データラベルの追加、縦軸の最大/最小値の追加、値の追加を行いグラフ		

	を描画します。
--	---------

3.4.2.9 GUUPD BARCHART RANGE

命令			
機 能	棒グラフの軸の範囲を更新します。		
書 式	GUUPD BARCHART RANGE <①棒グラフ番号>, <②縦軸最小値>, <②縦軸最大値>		
パラ メータ	①	<棒グラフ番号>	数値
	項目追加を行う棒グラフ番号を指定します。		
	②	<縦軸最大値>, <縦軸最小値>	数値
縦軸の最大値/最小値を指定します。			
備 考	・ グラフ領域の生成 (→「3.4.2.1 GUCREATE BARCHART」) が必要です。		
使用例	GUCREATE WINDOW 1 GUCREATE BARCHART 1, 10,, "400,300",, ... GUUPD BARCHART DRAW 10 GUSHOW 1 ... LBL\$ = ["4月"; "5月"; "6月"] GUUPD BARCHART DATALABEL 10, (LBL\$) GUUPD BARCHART GRID 10, 10 GUUPD BARCHART LABEL 10, "縦軸", "横軸" GUUPD BARCHART RANGE 10, 0, 30 VAL1 = [30; 10; 20] GUUPD BARCHART VALUE 10, (VAL1), (2) GUUPD BARCHART DRAW 10  描画した棒グラフに対して各種データを更新します。		

## 3.4.2.10 GUUPD BARCHART GRID

命令			
機能	棒グラフの目盛り線を更新します。		
書式	GUUPD BARCHART GRID <①棒グラフ番号>, <②縦軸目盛り間隔>		
パラメータ	①	<棒グラフ番号>	数値
	項目追加を行う棒グラフ番号を指定します。		
	②	<縦軸目盛り間隔>	数値
	縦軸の目盛り線の間隔値を数値で指定します。 正数値を指定してください。		
備考	・グラフ領域の生成 (→「3.4.2.1 GUCREATE BARCHART」) が必要です。		
使用例	GUCREATE WINDOW 1 GUCREATE BARCHART 1, 10, , "400, 300", , ... GUUPD BARCHART DRAW 10 GUSHOW 1 ... LBL\$ = ["4月"; "5月"; "6月"] GUUPD BARCHART DATALABEL 10, (LBL\$) GUUPD BARCHART GRID 10, 10 GUUPD BARCHART LABEL 10, "縦軸", "横軸" GUUPD BARCHART RANGE 10, 0, 30 VAL1 = [30; 10; 20] GUUPD BARCHART VALUE 10, (VAL1), (2) GUUPD BARCHART DRAW 10  描画した棒グラフに対して各種データを更新します。		

## 3.4.2.11 GUUPD BARCHART DRAW

命令			
機能	棒グラフのグラフ描画を実行します。		
書式	GUUPD BARCHART DRAW <①棒グラフ番号>		
パラメータ	①	<棒グラフ番号>	数値
	項目追加を行う棒グラフ番号を指定します。		
備考	・グラフ領域の生成 (→「3.4.2.1 GUCREATE BARCHART」) が必要です。		
使用例	GUCREATE WINDOW 1 GUCREATE BARCHART 1, 10, , "400, 300", , ... GUUPD BARCHART DRAW 10 GUSHOW 1 ... LBL\$ = ["4月"; "5月"; "6月"] GUUPD BARCHART DATALABEL 10, (LBL\$) GUUPD BARCHART GRID 10, 10 GUUPD BARCHART LABEL 10, "縦軸", "横軸" GUUPD BARCHART RANGE 10, 0, 30 VAL1 = [30; 10; 20] GUUPD BARCHART VALUE 10, (VAL1), (2) GUUPD BARCHART DRAW 10  描画した棒グラフに対して各種データを更新します。		

## 3.4.2.12 GUDSP BARCHART

命令			
機 能	棒グラフを表示あるいは非表示にします。		
書 式	GUDSP BARCHART <①棒グラフ番号>, <②表示状態> [, <③編集可能状態> ]		
パラ メータ	①	<棒グラフ番号>	数値
	表示あるいは非表示にする棒グラフの番号を指定します。		
	②	<表示状態>	真偽値
	GUI部品の表示状態を指定できます。 FALSE: 非表示 TRUE: 表示		
	③	<編集可能状態>	真偽値
	GUI部品の編集ができるかどうかを指定できます。 FALSE: 編集不可 TRUE: 編集可(デフォルト値)		
備 考	本機能の呼び出しは、ウィンドウの表示 (→「3.3.1.31 GUSHOW」) 後に行ってください。ウィンドウの表示時、全てGUI部品は表示状態です。		
使用例	GUCREATE WINDOW 1 GUCREATE BARCHART 1, 10 GUDSP BARCHART 10, 0  <部品番号> 10 の棒グラフを非表示にします。		

## 3.4.2.13 GUCREATE LINEGRAPH

命令		
機能	指定したウィンドウに折れ線グラフを配置します。	
書式	GUCREATE LINEGRAPH <①ウィンドウ番号>, <②折れ線グラフ番号> [, [ <③フォント設定> ], [ <④サイズ設定> ], [ <⑤ポジション設定> ], [ <⑥ツールチップ設定> ] ]  ※ フォント設定以降の引数を省略する場合、各引数間のカンマは必要です。	
パラメータ	①	<ウィンドウ番号> 折れ線グラフを配置するウィンドウの番号を指定します。 数値
	②	<折れ線グラフ番号> 作成する折れ線グラフの番号を指定します。 1～10000の値を指定してください。 数値
	③	<フォント設定> GUI部品に表示される文字列のフォントを指定します。(参照：P. 20) 文字列
	④	<サイズ設定> GUI部品に表示されるGUI部品の大きさを指定します。(参照：P. 22) 文字列
	⑤	<ポジション設定> GUI部品に表示されるGUI部品の表示場所を指定します。(参照：P. 23) 文字列
	⑥	<ツールチップ設定> マウスポインタがGUI部品上に来た際に、表示されるツールチップの内容を指定します。 (参照：P. 24) 文字列
使用例1	GUCREATE WINDOW 1 GUCREATE LINEGRAPH 1, 10  <ウィンドウ番号> 1 のウィンドウに <グラフ番号> 10の折れ線グラフを配置します。	

## 3.4.2.14 GUADD LINEGRAPH VALUE

命令		
機 能	折れ線グラフに値を追加します。	
書 式	GUADD LINEGRAPH VALUE <①折れ線グラフ番号>, ( <②値配列> [, <②値配列> ...] ) [, <③線の色配列>, <④線の太さ配列> ]	
パラ メータ	①	<折れ線グラフ番号> 項目追加を行う折れ線グラフ番号を指定します。           数値
	②	<値配列> 描画する数値の1次元値配列を丸カッコで挟んで指定します。           配列
	③	<線の色配列> 線の色を指定します。 <値配列>で指定した本数と同数の要素数を持つ配列へ、各色の数値を与えてください。 色番号は「フォント設定」の文字色の項を参照してください。           配列
	④	<線の太さ配列> 線の太さを指定します。 <値配列>で指定した本数と同数の要素数を持つ配列へ、各線の太さの数値を与えてください。           配列
備 考	<ul style="list-style-type: none"> <li>・グラフ領域の生成 (→「3.4.2.13 GUCREATE LINEGRAPH」) が必要です。</li> <li>・グラフを実際に描画するには、「GUUPD LINEGRAPH DRAW」が必要です。</li> <li>それ以外の呼び出しは、内部に対する設定で実際の描画には反映されません。</li> </ul>	
使用例	<pre> GUCREATE WINDOW 1 GUCREATE LINEGRAPH 1, 10,, "400, 300",, GUADD LINEGRAPH TITLE 10, "タイトル" LBL\$ = ["1月"; "2月"; "3月"] GUADD LINEGRAPH DATALABEL 10, (LBL\$) VAL1 = [10; 30; 20] VAL2 = [20; 10; 30] CL = [2; 5] LI = [3; 6] GUADD LINEGRAPH VALUE 10, (VAL1, VAL2), CL, LI GUADD LINEGRAPH RANGE 10, 0, 50 GUUPD LINEGRAPH DRAW 10 GUSHOW 1 </pre> <p>           &lt;ウィンドウ番号&gt; 1 のウィンドウに対して折れ線グラフを配置します。            タイトルの追加、データラベルの追加、縦軸の最大/最小値の追加、値の追加を行いグラフを描画します。         </p>	

## 3.4.2.15 GUUPD LINEGRAPH VALUE

命令		
機能	折れ線グラフの値を更新します。(書式1) 折れ線グラフの値を全クリアします。(書式2)	
書式1	GUUPD LINEGRAPH VALUE <①折れ線グラフ番号>, ( <②値配列> [, <②値配列> ... ] ) [, <③線の色配列>, <④線の太さ配列> ]  折れ線グラフ用の、値のデータを更新します。	
書式2	GUUPD LINEGRAPH VALUE <①折れ線グラフ番号>  折れ線グラフ用の、値のデータを全クリアします。	
パラメータ	①	<div>&lt;折れ線グラフ番号&gt;</div> <div>数値</div> <div>項目追加を行う折れ線グラフ番号を指定します。</div>
	②	<div>&lt;値配列&gt;</div> <div>配列</div> <div>描画する数値の1次元配列を丸カッコで挟んで指定します。 複数の線を描画したい場合は、値配列を複数準備し引数に指定します。</div>
	③	<div>&lt;線の色配列&gt;</div> <div>配列</div> <div>線の色を指定します。 &lt;値配列&gt;で指定した本数の要素数の配列内に、各色の数値を与えてください。 色番号は「3.2.1 フォント設定」の文字色の項を参照してください。</div>
	④	<div>&lt;線の太さ配列&gt;</div> <div>配列</div> <div>線の太さを指定します。 &lt;値配列&gt;で指定した本数と同数の要素数を持つ配列へ、各線の太さの数値を与えてください。</div>
備考	<ul style="list-style-type: none"> <li>・グラフ領域の生成 (→「3.4.2.13 GUCREATE LINEGRAPH」) が必要です。</li> <li>・グラフを実際に描画するには、「GUUPD LINEGRAPH DRAW」が必要です。</li> </ul> それ以外の呼び出しは、内部に対する設定で実際の描画には反映されません。	
使用例	<pre> GUCREATE WINDOW 1 GUCREATE LINEGRAPH 1, 10,, "400,300",, ... GUUPD LINEGRAPH DRAW 10 GUSHOW 1 ... LBL\$ = ["4月"; "5月"; "6月"] GUUPD LINEGRAPH DATALABEL 10, (LBL\$) GUUPD LINEGRAPH GRID 10, 10, 1 GUUPD LINEGRAPH LABEL 10, "縦軸", "横軸" GUUPD LINEGRAPH RANGE 10, 0, 30 VAL1 = [30; 10; 20] VAL2 = [20; 30; 10] CL = [2; 5] LI = [3; 6] GUUPD LINEGRAPH VALUE 10, (VAL1, VAL2), CL, LI GUUPD LINEGRAPH DRAW 10 </pre> 描画した棒グラフに対して各種データを更新します。	

## 3.4.2.16 GUADD LINEGRAPH TITLE

命令		
機能	折れ線グラフにタイトルを追加します。	
書式	GUADD LINEGRAPH TITLE <①折れ線グラフ番号>, <②タイトル名> [, <③フォント設定> ]	
パラメータ	①	<折れ線グラフ番号> 数値
	項目追加を行う折れ線グラフ番号を指定します。	
	②	<タイトル名> 文字列
	タイトルとして表示する文字列を指定します。	
	③	<フォント設定> 文字列
	GUI部品に表示される文字列のフォントを指定します。(参照 : P. 20)	
備考	<ul style="list-style-type: none"> <li>・グラフ領域の生成 (→「3.4.2.13 GUCREATE LINEGRAPH」) が必要です。</li> <li>・グラフを実際に描画するには、「GUUPD LINEGRAPH DRAW」が必要です。それ以外の呼び出しは、内部に対する設定で実際の描画には反映されません。</li> <li>・&lt;フォント設定&gt;を指定すると、任意のフォント名およびサイズ、文字色の文字をラベル名に指定できます。(背景色は反映されません)省略時、「GUCREATE LINEGRAPH」で指定した設定が引き継がれます。</li> </ul>	
使用例	<pre> GUCREATE WINDOW 1 GUCREATE LINEGRAPH 1, 10,, "400,300",, GUADD LINEGRAPH TITLE 10, "タイトル" LBL\$ = ["1月"; "2月"; "3月"] GUADD LINEGRAPH DATALABEL 10, (LBL\$) VAL1 = [10; 30; 20] VAL2 = [20; 10; 30] CL = [2; 5] LI = [3; 6] GUADD LINEGRAPH VALUE 10, (VAL1, VAL2), CL, LI GUADD LINEGRAPH RANGE 10, 0, 50 GUUPD LINEGRAPH DRAW 10 GUSHOW 1 </pre> <p>&lt;ウィンドウ番号&gt; 1 のウィンドウに対して折れ線グラフを配置します。        タイトルの追加、データラベルの追加、縦軸の最大/最小値の追加、値の追加を行いグラフを描画します。</p>	



## 3.4.2.17 GUUPD LINEGRAPH LABEL

命令			
機能	折れ線グラフの軸ラベルを更新します。		
書式	GUUPD LINEGRAPH LABEL <①折れ線グラフ番号>, <②縦軸ラベル名>, <②横軸ラベル名> [, <③フォント設定> ]		
パラメータ	①	<折れ線グラフ番号>	数値
	項目追加を行う折れ線グラフ番号を指定します。		
	②	<縦軸ラベル名>, <横軸ラベル名>	文字列
	項目に表示する文字列を指定します。		
	③	<フォント設定>	文字列
	GUI部品に表示される文字列のフォントを指定します。(参照 : P. 20)		
備考	<ul style="list-style-type: none"> <li>・グラフ領域の生成 (→「3.4.2.13 GUCREATE LINEGRAPH」) が必要です。</li> <li>・グラフを実際に描画するには、「GUUPD LINEGRAPH DRAW」が必要です。それ以外の呼び出しは、内部に対する設定で実際の描画には反映されません。</li> <li>・&lt;フォント設定&gt;を指定すると、任意のフォント名およびサイズ、文字色の文字をラベル名に指定できます。(背景色は反映されません)</li> </ul> 省略時、「GUCREATE LINEGRAPH」で指定した設定が引き継がれます。		
使用例	<pre> GUCREATE WINDOW 1 GUCREATE LINEGRAPH 1, 10,, "400, 300",, ... GUUPD LINEGRAPH DRAW 10 GUSHOW 1 ... LBL\$ = ["4月"; "5月"; "6月"] GUUPD LINEGRAPH DATALABEL 10, (LBL\$) GUUPD LINEGRAPH GRID 10, 10, 1 GUUPD LINEGRAPH LABEL 10, "縦軸", "横軸" GUUPD LINEGRAPH RANGE 10, 0, 30 VAL1 = [30; 10; 20] VAL2 = [20; 30; 10] CL = [2; 5] LI = [3; 6] GUUPD LINEGRAPH VALUE 10, (VAL1, VAL2), CL, LI GUUPD LINEGRAPH DRAW 10  描画した棒グラフに対して各種データを更新します。 </pre>		

## 3.4.2.18 GUADD LINEGRAPH DATALABEL

命令			
機能	折れ線グラフにデータラベルを追加します。		
書式	GUADD LINEGRAPH DATALABEL <①折れ線グラフ番号>, ( <②横軸データラベル配列> )		
パラメータ	①	<折れ線グラフ番号>	数値
	項目追加を行う折れ線グラフ番号を指定します。		
	②	<横軸データラベル配列>	配列
	横軸に対して任意に表示したいデータラベル文字列の1次元配列を丸カッコで挟んで指定します。		
備考	<ul style="list-style-type: none"> <li>・グラフ領域の生成 (→「3.4.2.13 GUCREATE LINEGRAPH」) が必要です。</li> <li>・グラフを実際に描画するには、「GUUPD LINEGRAPH DRAW」が必要です。</li> </ul> それ以外の呼び出しは、内部に対する設定で実際の描画には反映されません。		
使用例	<pre> GUCREATE WINDOW 1 GUCREATE LINEGRAPH 1, 10,, "400,300",, GUADD LINEGRAPH TITLE 10, "タイトル" LBL\$ = ["1月"; "2月"; "3月"] GUADD LINEGRAPH DATALABEL 10, (LBL\$) VAL1 = [10; 30; 20] VAL2 = [20; 10; 30] CL = [2; 5] LI = [3; 6] GUADD LINEGRAPH VALUE 10, (VAL1, VAL2), CL, LI GUADD LINEGRAPH RANGE 10, 0, 50 GUUPD LINEGRAPH DRAW 10 GUSHOW 1 </pre> <ウィンドウ番号> 1 のウィンドウに対して折れ線グラフを配置します。 タイトルの追加、データラベルの追加、縦軸の最大/最小値の追加、値の追加を行いグラフを描画します。		

## 3.4.2.19 GUUPD LINEGRAPH DATALABEL

命令			
機 能	折れ線グラフのデータラベルを更新します。		
書 式	GUUPD LINEGRAPH DATALABEL <①折れ線グラフ番号>, ( <②横軸データラベル配列> )		
パラ メータ	①	<折れ線グラフ番号>	数値
	項目追加を行う折れ線グラフ番号を指定します。		
	②	<横軸データラベル配列>	配列
	横軸に対して任意に表示したいデータラベル文字列の1次元配列を丸カッコで挟んで指定します。		
備 考	<ul style="list-style-type: none"> <li>・グラフ領域の生成 (→「3.4.2.13 GUCREATE LINEGRAPH」) が必要です。</li> <li>・グラフを実際に描画するには、「GUUPD LINEGRAPH DRAW」が必要です。</li> </ul> それ以外の呼び出しは、内部に対する設定で実際の描画には反映されません。		
使用例	<pre> GUCREATE WINDOW 1 GUCREATE LINEGRAPH 1, 10,, "400,300",, ... GUUPD LINEGRAPH DRAW 10 GUSHOW 1 ... LBL\$ = ["4月"; "5月"; "6月"] GUUPD LINEGRAPH DATALABEL 10, (LBL\$) GUUPD LINEGRAPH GRID 10, 10, 1 GUUPD LINEGRAPH LABEL 10, "縦軸", "横軸" GUUPD LINEGRAPH RANGE 10, 0, 30 VAL1 = [30; 10; 20] VAL2 = [20; 30; 10] CL = [2; 5] LI = [3; 6] GUUPD LINEGRAPH VALUE 10, (VAL1, VAL2), CL, LI GUUPD LINEGRAPH DRAW 10  描画した棒グラフに対して各種データを更新します。 </pre>		

## 3.4.2.20 GUADD LINEGRAPH RANGE

命令			
機能	折れ線グラフに軸の範囲を追加します。		
書式	GUADD LINEGRAPH RANGE <①折れ線グラフ番号>, <②縦軸最小値>, <②縦軸最大値> [, <③横軸最小値>, <③横軸最大値> ]		
パラメータ	①	<折れ線グラフ番号>	数値
	項目追加を行う折れ線グラフ番号を指定します。		
	②	<縦軸最大値>, <縦軸最小値>	数値
	縦軸の最大値/最小値を指定します。		
	③	<横軸最大値>, <横軸最小値>	数値
	横軸の最大値/最小値を指定します。		
	省略した場合はグラフに描画する値によって自動計算します。		
備考	<ul style="list-style-type: none"> <li>・グラフ領域の生成 (→「3.4.2.13 GUCREATE LINEGRAPH」) が必要です。</li> <li>・グラフを実際に描画するには、「GUUPD LINEGRAPH DRAW」が必要です。それ以外の呼び出しは、内部に対する設定で実際の描画には反映されません。</li> <li>・「GUUPD LINEGRAPH VALUE」などで、値のデータを追加していない時(即ち空の時)、縦軸の範囲を設定しても、情報が不足している為に描画されません。</li> </ul>		
使用例	<pre> GUCREATE WINDOW 1 GUCREATE LINEGRAPH 1, 10,, "400,300",, GUADD LINEGRAPH TITLE 10, "タイトル" LBL\$ = ["1月"; "2月"; "3月"] GUADD LINEGRAPH DATALABEL 10, (LBL\$) VAL1 = [10; 30; 20] VAL2 = [20; 10; 30] CL = [2; 5] LI = [3; 6] GUADD LINEGRAPH VALUE 10, (VAL1, VAL2), CL, LI GUADD LINEGRAPH RANGE 10, 0, 50 GUUPD LINEGRAPH DRAW 10 GUSHOW 1 </pre> <p>&lt;ウィンドウ番号&gt; 1 のウィンドウに対して折れ線グラフを配置します。  タイトル追加、データラベルの追加、縦軸の最大/最小値の追加、値の追加を行いグラフを描画します。</p>		

## 3.4.2.21 GUUPD LINEGRAPH RANGE

命令			
機 能	折れ線グラフの各軸の範囲を更新します。		
書 式	GUUPD LINEGRAPH RANGE <①折れ線グラフ番号>, <②縦軸最小値>, <②縦軸最大値> [, <③横軸最小値>, <③横軸最大値> ]		
パラ メータ	①	<折れ線グラフ番号>	数値
	項目追加を行う折れ線グラフ番号を指定します。		
	②	<縦軸最大値>, <縦軸最小値>	数値
	縦軸の最大値/最小値を指定します。		
	③	<横軸最大値>, <横軸最小値>	数値
	横軸の最大値/最小値を指定します。 省略した場合はグラフに描画する値によって自動計算します。		
備 考	<ul style="list-style-type: none"> <li>・グラフ領域の生成 (→「3.4.2.13 GUCREATE LINEGRAPH」) が必要です。</li> <li>・グラフを実際に描画するには、「GUUPD LINEGRAPH DRAW」が必要です。</li> <li>それ以外の呼び出しは、内部に対する設定で実際の描画には反映されません。</li> <li>・「GUUPD LINEGRAPH VALUE」などで、値のデータを追加していない時(即ち空の時)、縦軸の範囲を設定しても、情報が不足している為に描画されません。</li> </ul>		
使用例	<pre> GUCREATE WINDOW 1 GUCREATE LINEGRAPH 1, 10,, "400, 300",, ... GUUPD LINEGRAPH DRAW 10 GUSHOW 1 ... LBL\$ = ["4月"; "5月"; "6月"] GUUPD LINEGRAPH DATALABEL 10, (LBL\$) GUUPD LINEGRAPH GRID 10, 10, 1 GUUPD LINEGRAPH LABEL 10, "縦軸", "横軸" GUUPD LINEGRAPH RANGE 10, 0, 30 VAL1 = [30; 10; 20] VAL2 = [20; 30; 10] CL = [2; 5] LI = [3; 6] GUUPD LINEGRAPH VALUE 10, (VAL1, VAL2), CL, LI GUUPD LINEGRAPH DRAW 10  描画した折れ線グラフに対して各種データを更新します。 </pre>		

## 3.4.2.22 GUADD LINEGRAPH GUAGE

命令		
機 能	折れ線グラフに、任意の補助線を追加します。	
書 式	GUADD LINEGRAPH GUAGE <①折れ線グラフ番号>, <②補助線配列>	
パラ メータ	①	<div> <div>&lt;折れ線グラフ番号&gt;</div> <div>数値</div> </div> 項目追加を行う折れ線グラフ番号を指定します。
	②	<div> <div>&lt;補助線配列&gt;</div> <div>配列</div> </div> 任意の補助線を引く為の情報を、配列形式で渡します。 配列は、3の要素の倍数で指定します。以下のように指定します。
	配列の 添字	項目
	(0)	線の方向 線を引く方向を指定します。 1 : 垂直線を引きます。 2 : 水平線を引きます。
	(1)	値 線を引く位置を指定します。 垂直線を引く時、x座標の位置に相当します。 水平線を引く時、y座標の位置に相当します。
	(2)	色番号 色番号を指定します。 GUIヘルプの「フォント設定」の<文字色>と同じ指定です。
...		
(0) ~ (2)の繰り返しです。		
備 考	<ul style="list-style-type: none"> <li>グラフ領域の生成 (→「3.4.2.13 GUCREATE LINEGRAPH」) が必要です。</li> <li>グラフを実際に描画するには、「GUUPD LINEGRAPH DRAW」が必要です。</li> </ul> それ以外の呼び出しは、内部に対する設定で実際の描画には反映されません。	
使用例	DIM ARY_X(2) ARY_X = [ 1; 100; 2 ] ' 垂直線、100番目の位置、赤(2)を引く GUADD LINEGRAPH GUAGE 10, ARY_X	

### 3.4.2.23 GUUPD LINEGRAPH GUAGE

命令		
機能	折れ線グラフに、任意の補助線を上書き更新します。	
書式	GUUPD LINEGRAPH GUAGE <①折れ線グラフ番号>, <②補助線配列>	
パラメータ	①	<div> <div>&lt;折れ線グラフ番号&gt;</div> <div>数値</div> </div> 項目更新を行う折れ線グラフ番号を指定します。
	②	<div> <div>&lt;補助線配列&gt;</div> <div>配列</div> </div> 任意の補助線を引く為の情報を、配列形式で渡します。 配列は、3の要素の倍数で指定します。以下のように指定します。
	配列の添字	項目
	(0)	線の方向 線を引く方向を指定します。 1：垂直線を引きます。 2：水平線を引きます。
	(1)	値 線を引く位置を指定します。 垂直線を引く時、x座標の位置に相当します。 水平線を引く時、y座標の位置に相当します。
	(2)	色番号 色番号を指定します。 GUIヘルプの「フォント設定」の<文字色>と同じ指定です。
...		
(0)～(2)の繰り返しです。		
備考	<ul style="list-style-type: none"> <li>グラフ領域の生成（→「3.4.2.13 GUCREATE LINEGRAPH」）が必要です。</li> <li>グラフを実際に描画するには、「GUUPD LINEGRAPH DRAW」が必要です。</li> </ul> それ以外の呼び出しは、内部に対する設定で実際の描画には反映されません。	
使用例	DIM ARY_X(2) ARY_X = [ 2; 150; 5 ] ' 水平線、150番目の位置、青(5)に更新する GUUPD LINEGRAPH GUAGE 10, ARY_X	

## 3. 4. 2. 24 GUUPD LINEGRAPH GRID

命令			
機 能	折れ線グラフの目盛り線を更新します。		
書 式	GUUPD LINEGRAPH GRID <①折れ線グラフ番号> [, <②縦軸目盛り間隔> [, <②横軸目盛り間隔> [, <③縦軸目盛り間隔の基準点> [, <③横軸目盛り間隔の基準点> ] ] ] ]		
パラ メータ	①	<折れ線グラフ番号>	数値
	項目追加を行う折れ線グラフ番号を指定します。		
	②	<縦軸目盛り間隔>, <横軸目盛り間隔>	数値
	縦軸および横軸の目盛り線の間隔値を数値で指定します。 指定するときは、正数値を指定してください。 省略すると、グラフ描画関数の判断に任されます。		
	③	<縦軸目盛り間隔の基準点>, <横軸目盛り間隔の基準点>	数値
	縦軸および横軸の目盛り線を描画する為の基準点を数値で指定します。 基準点から指定した目盛り間隔で、目盛り線を描画します。 省略すると、グラフ描画関数の判断に任されます。		
備 考	<ul style="list-style-type: none"> <li>・グラフ領域の生成（→「3. 4. 2. 13 GUCREATE LINEGRAPH」）が必要です。</li> <li>・グラフを実際に描画するには、「GUUPD LINEGRAPH DRAW」が必要です。</li> </ul> それ以外の呼び出しは、内部に対する設定で実際の描画には反映されません。		
使用例	<pre> GUCREATE WINDOW 1 GUCREATE LINEGRAPH 1, 10,, "400, 300",, ... GUUPD LINEGRAPH DRAW 10 GUSHOW 1 ... LBL\$ = ["4月"; "5月"; "6月"] GUUPD LINEGRAPH DATALABEL 10, (LBL\$) GUUPD LINEGRAPH GRID 10, 10, 10 GUUPD LINEGRAPH LABEL 10, "縦軸", "横軸" GUUPD LINEGRAPH RANGE 10, 0, 30 VAL1 = [30; 10; 20] VAL2 = [20; 30; 10] CL = [2; 5] LI = [3; 6] GUUPD LINEGRAPH VALUE 10, (VAL1, VAL2), CL, LI GUUPD LINEGRAPH DRAW 10  描画した折れ線グラフに対して各種データを更新します。 </pre>		



## 3.4.2.25 GUUPD LINEGRAPH DRAW

命令			
機能	折れ線グラフのグラフ描画を実行します。		
書式	GUUPD LINEGRAPH DRAW <①折れ線グラフ番号>		
パラメータ	①	<折れ線グラフ番号>	数値
	グラフ描画を実行する折れ線グラフ番号を指定します。		
備考	<ul style="list-style-type: none"> <li>・グラフ領域の生成（→「3.4.2.13 GUCREATE LINEGRAPH」）が必要です。</li> <li>・グラフを描画する為に情報が不足していると、グラフは描画されません。</li> </ul> <p>「GUUPD LINEGRAPH VALUE」や「GUADD LINEGRAPH VALUE」でグラフの表示に必要なデータを与える必要があります。</p> <p>あるいは、「GUUPD LINEGRAPH GRID」と「GUUPD LINEGRAPH RANGE」で縦軸および横軸の範囲を設定すると、目盛り線の描画は可能です。</p>		
使用例	<pre> GUCREATE WINDOW 1 GUCREATE LINEGRAPH 1, 10,, "400,300",, ... GUUPD LINEGRAPH DRAW 10 GUSHOW 1 ... LBL\$ = ["4月"; "5月"; "6月"] GUUPD LINEGRAPH DATALABEL 10, (LBL\$) GUUPD LINEGRAPH GRID 10, 10, 1 GUUPD LINEGRAPH LABEL 10, "縦軸", "横軸" GUUPD LINEGRAPH RANGE 10, 0, 30 VAL1 = [30; 10; 20] VAL2 = [20; 30; 10] CL = [2; 5] LI = [3; 6] GUUPD LINEGRAPH VALUE 10, (VAL1, VAL2), CL, LI GUUPD LINEGRAPH DRAW 10 </pre> <p>描画した折れ線グラフに対して各種データを更新します。</p>		

## 3.4.2.26 GUDSP LINEGRAPH

命令		
機 能	折れ線グラフを表示あるいは非表示にします。	
書 式	GUDSP LINEGRAPH <①折れ線グラフ番号>, <②表示状態> [, <③編集可能状態> ]	
パラ メータ	①	数値
	<折れ線グラフ番号> 表示あるいは非表示にする折れ線グラフの番号を指定します。	
	②	真偽値
	<表示状態> GUI部品の表示状態を指定できます。 FALSE: 非表示 TRUE: 表示	
	③	真偽値
	<編集可能状態> GUI部品の編集ができるかどうかを指定できます。 FALSE: 編集不可 TRUE: 編集可(デフォルト値)	
備 考	本機能の呼び出しは、ウィンドウの表示 (→「3.3.1.31 GUSHOW」) 後に行ってください。ウィンドウの表示時、全てGUI部品は表示状態です。	
使用例	GUCREATE WINDOW 1 GUCREATE LINEGRAPH 1, 10 GUDSP LINEGRAPH 10, 0  <部品番号> 10 の折れ線グラフを非表示にします。	

## 3.4.2.27 GUCREATE PIECHART

命令		
機能	指定したウィンドウに円グラフを配置します。	
書式	GUCREATE PIECHART <①ウィンドウ番号>, <②円グラフ番号> [, [ <③フォント設定> ], [ <④サイズ設定> ], [ <⑤ポジション設定> ], [ <⑥ツールチップ設定> ] ]	
	※ フォント設定以降の引数を省略する場合、各引数間のカンマは必要です。	
パラメータ	①	<ウィンドウ番号> 数値
	円グラフを配置するウィンドウの番号を指定します。	
	②	<円グラフ番号> 数値
	作成する円グラフの番号を指定します。 1～10000の値を指定してください。	
	③	<フォント設定> 文字列
	GUI部品に表示される文字列のフォントを指定します。(参照：P. 20)	
	④	<サイズ設定> 文字列
	GUI部品に表示されるGUI部品の大きさを指定します。(参照：P. 22)	
	⑤	<ポジション設定> 文字列
	GUI部品に表示されるGUI部品の表示場所を指定します。(参照：P. 23)	
	⑥	<ツールチップ設定> 文字列
	マウスポインタがGUI部品上に来た際に、表示されるツールチップの内容を指定します。 (参照：P. 24)	
使用例1	GUCREATE WINDOW 1 GUCREATE PIECHART 1, 10  <ウィンドウ番号> 1 のウィンドウに <グラフ番号> 10の円グラフを配置します。	

## 3.4.2.28 GUADD PIECHART VALUE

命令			
機 能	円グラフに値を追加します。		
書 式	GUADD PIECHART VALUE <①円グラフ番号>, ( <②値配列> ) [, <③色配列> ]		
パラ メータ	①	<円グラフ番号>	数値
	項目追加を行う円グラフ番号を指定します。		
	②	<値配列>	配列
	描画する数値の1次元配列を丸カッコで挟んで指定します。		
	③	<色配列>	配列
	線の色を指定します。 <値配列>で指定した数と同数の要素数を持つ配列へ、各色の数値を与えてください。 色番号は「3.2.1 フォント設定」の文字色の項を参照してください。		
備 考	・ グラフ領域の生成 (→「3.4.2.27 GUCREATE PIECHART」) が必要です。		
使用例	<p>GUCREATE WINDOW 1  GUCREATE PIECHART 1, 10, , "300, 300", ,  GUADD PIECHART TITLE 10, "タイトル"  LBL\$ = ["1月"; "2月"; "3月"]  GUADD PIECHART DATALABEL 10, (LBL\$)  VAL1 = [10; 30; 20]  CL = [2; 3; 4]  GUADD PIECHART VALUE 10, (VAL1), CL  GUUPD PIECHART DRAW 10  GUSHOW 1</p> <p>&lt;ウィンドウ番号&gt; 1 のウィンドウに対して円グラフを配置します。  タイトルの追加、データラベルの追加、値の追加を行いグラフを描画します。</p>		

## 3.4.2.29 GUUPD PIECHART VALUE

命令		
機 能	円グラフの値を更新します。	
書 式	GUUPD PIECHART VALUE <①円グラフ番号>, ( <②値配列> ) [, <③線の色配列> ]	
パラ メータ	①	数値
	<円グラフ番号> 項目追加を行う円グラフ番号を指定します。	
	②	配列
	<値配列> 描画する数値の1次元配列を丸カッコで挟んで指定します。	
	③	配列
	<線の色配列> 線の色を指定します。 <値配列>で指定した数と同数の要素数を持つ配列へ、各色の数値を与えてください。 色番号は「3.2.1 フォント設定」の文字色の項を参照してください。	
備 考	グラフ領域の生成 (→「3.4.2.27 GUCREATE PIECHART」) が必要です。	
使用例	GUCREATE WINDOW 1 GUCREATE PIECHART 1, 10, , "300, 300", , ... GUUPD PIECHART DRAW 10 GUSHOW 1 ... LBL\$ = ["4月"; "5月"; "6月"] GUUPD PIECHART DATALABEL 10, (LBL\$) VAL1 = [30; 10; 20] CL = [2; 3; 4] GUUPD PIECHART VALUE 10, (VAL1), CL GUUPD PIECHART DRAW 10  描画した円グラフに対して各種データを更新します。	

## 3.4.2.30 GUADD PIECHART TITLE

命令		
機 能	円グラフにタイトルを追加します。	
書 式	GUADD PIECHART TITLE <①円グラフ番号>, <②タイトル名> [, <③フォント設定> ]	
パラ メータ	①	数値
	<円グラフ番号> 項目追加を行う円グラフ番号を指定します。	
	②	文字列
	<タイトル名> タイトルとして表示する文字列を指定します。	
	③	文字列
	<フォント設定> GUI部品に表示される文字列のフォントを指定します。(参照 : P. 20)	
備 考	<ul style="list-style-type: none"> <li>・グラフ領域の生成 (→「3.4.2.27 GUCREATE PIECHART」) が必要です。</li> <li>・&lt;フォント設定&gt;を指定すると、任意のフォント名およびサイズ、文字色の文字をラベル名に指定できます。(背景色は反映されません)省略時、「GUCREATE PIECHART」で指定した設定が引き継がれます。</li> </ul>	
使用例	<pre>GUCREATE WINDOW 1 GUCREATE PIECHART 1, 10,, "300, 300",, GUADD PIECHART TITLE 10, "タイトル" LBL\$ = ["1月"; "2月"; "3月"] GUADD PIECHART DATALABEL 10, (LBL\$) VAL1 = [10; 30; 20] CL = [2; 3; 4] GUADD PIECHART VALUE 10, (VAL1), CL GUUPD PIECHART DRAW 10 GUSHOW 1</pre> <p>&lt;ウィンドウ番号&gt; 1 のウィンドウに対して円グラフを配置します。 タイトルの追加、データラベルの追加、値の追加を行いグラフを描画します。</p>	

## 3.4.2.31 GUADD PIECHART DATALABEL

命令		
機 能	円グラフにデータラベルを追加します。	
書 式	GUADD PIECHART DATALABEL <①円グラフ番号>, ( <②データラベル配列> )	
パラ メータ	①	数値
	<円グラフ番号>	
	項目追加を行う円グラフ番号を指定します。	
パラ メータ	②	配列
	<データラベル配列>	
	グラフの値に対する補足説明の内容を文字列配列で丸カッコで挟んで指定します。	
備 考	・グラフ領域の生成 (→「3.4.2.27 GUCREATE PIECHART」) が必要です。	
使用例	<pre>GUCREATE WINDOW 1 GUCREATE PIECHART 1, 10,, "300, 300",, GUADD PIECHART TITLE 10, "タイトル" LBL\$ = ["1月"; "2月"; "3月"] GUADD PIECHART DATALABEL 10, (LBL\$) VAL1 = [10; 30; 20] CL = [2; 3; 4] GUADD PIECHART VALUE 10, (VAL1), CL GUUPD PIECHART DRAW 10 GUSHOW 1</pre> <p>&lt;ウィンドウ番号&gt; 1 のウィンドウに対して円グラフを配置します。 タイトルの追加、データラベルの追加、値の追加を行いグラフを描画します。</p>	

## 3.4.2.32 GUUPD PIECHART DATALABEL

命令		
機 能	円グラフのデータラベルを更新します。	
書 式	GUUPD PIECHART DATALABEL <①円グラフ番号>, ( <②データラベル配列> )	
パラ メータ	①	数値
	<円グラフ番号>	
	項目追加を行う円グラフ番号を指定します。	
パラ メータ	②	配列
	<データラベル配列>	
	グラフの値に対する補足説明の内容を文字列配列で丸カッコで挟んで指定します。	
備 考	グラフ領域の生成 (→「3.4.2.27 GUCREATE PIECHART」) が必要です。	
使用例	<pre>GUCREATE WINDOW 1 GUCREATE PIECHART 1, 10,, "300, 300",, ... GUUPD PIECHART DRAW 10 GUSHOW 1 ... LBL\$ = ["4月"; "5月"; "6月"] GUUPD PIECHART DATALABEL 10, (LBL\$) VAL1 = [30; 10; 20] CL = [2; 3; 4] GUUPD PIECHART VALUE 10, (VAL1), CL GUUPD PIECHART DRAW 10</pre> <p>描画した円グラフに対して各種データを更新します。</p>	

## 3.4.2.33 GUUPD PIECHART DRAW

命令			
機 能	円グラフのグラフ描画を実行します。		
書 式	GUUPD PIECHART DRAW <①円グラフ番号>		
パラ メータ	①	<円グラフ番号>	数値
	項目追加を行う円グラフ番号を指定します。		
備 考	グラフ領域の生成 (→「3.4.2.27 GUCREATE PIECHART」) が必要です。		
使用例	GUCREATE WINDOW 1 GUCREATE PIECHART 1, 10, , "300, 300", , ... GUUPD PIECHART DRAW 10 GUSHOW 1 ... LBL\$ = ["4月"; "5月"; "6月"] GUUPD PIECHART DATALABEL 10, (LBL\$) VAL1 = [30; 10; 20] CL = [2; 3; 4] GUUPD PIECHART VALUE 10, (VAL1), CL GUUPD PIECHART DRAW 10  描画した円グラフに対して各種データを更新します。		

## 3.4.2.34 GUDSP PIECHART

命令			
機 能	円グラフを表示あるいは非表示にします。		
書 式	GUDSP PIECHART <①円グラフ番号>, <②表示状態> [, <③編集可能状態> ]		
パラ メータ	①	<円グラフ番号>	数値
	表示あるいは非表示にする円グラフの番号を指定します。		
	②	<表示状態>	真偽値
	GUI部品の表示状態を指定できます。 FALSE:非表示 TRUE:表示		
備 考	③	<編集可能状態>	真偽値
	GUI部品の編集ができるかどうかを指定できます。 FALSE:編集不可 TRUE:編集可(デフォルト値)		
使用例	GUCREATE WINDOW 1 GUCREATE PIECHART 1, 10 GUDSP PIECHART 10, 0  <部品番号> 10 の円グラフを非表示にします。		



## 3.5 グラフィック描画に関する関数・命令

### 3.5.1 グラフィック描画に関する個別説明

グラフィック描画コマンドを用いると、ウィンドウ内に描画領域を設定する事で、その描画領域の中で、自由に線や円、画像などを書き込む事が可能です。

グラフィック描画を行う事で、下図のような自由な描画が可能です。



プログラムコード例：

OPTION GUI       ' GUI を使用する際に必須です

SIZE\$ = "400,200"

' ウィンドウを作成

GUCREATE WINDOW 1, "ウィンドウ", 1,,SIZE\$,,

' グラフィック描画領域を作成

GUCREATE PIC 1, 10,,SIZE\$,,

' 線分を描画

FOR X=0 TO 400 STEP 5

    C = INT(RND() \* 7) + 1

    GUGLINE PIC 10, (X, 0)-(400-X, 200), C

NEXT X

' 円を描画

FOR CNT=0 TO 50

    X = RND() \* 200

    Y = RND() \* 200

    R = INT(RND() \* 50) + 10

    C = INT(RND() \* 7) + 1

    GUGCOLOR PIC 10, C

    GUGCIRCLE PIC 10, (X, Y), R

NEXT CNT

' 画像を描画

FOR CNT=0 TO 10

    X = (RND() \* 200) + 200

    Y = (RND() \* 50) + 50

    GUGDRAWFILE PIC 10, (X, Y), "/usr/share/icons/hicolor/64x64/apps/interface-logo.png"

NEXT CNT

' テキスト「Hello AJAN」を描画

GUGDRAWTEXT PIC 10, (15, 13), "Hello AJAN", "6, 0, 0, 48, 1, 0"

GUGDRAWTEXT PIC 10, (10, 10), "Hello AJAN", "6, 0, 0, 48, 2, 0"

' ウィンドウを表示 GUSHOW 1
' GUI メインイベントループを開始 GUMAIN LOOP

グラフィック描画を行うGUI部品は、「GUCREATE PIC」で、ウィンドウ内に配置する事ができます。

その後は、「GUCREATE PIC」 を呼び出した時に指定した、描画領域番号を使って、線分(GUGLINE) や円(GUGCIRCLE)を描いたり、画像(GUGDRAWFILE)を描画したり、文字(GUGDRAWTEXT)を書き込んだり、部分画像のコピー(GUGDRAWBLIT)などができます。

グラフィック描画の各コマンドの座標は、描画領域の左上を原点(0, 0) として、サイズ分までを描画可能な領域とします。

描画コマンドの呼び出し毎に、描画内容は上書きされます。(油絵でキャンバスに絵を描くようなイメージ)  
「GUGCLS」で、描画領域に対してクリアできます。

3.5.2 コマンド個別説明

3.5.2.1 GUCREATE PIC

命令			
機能	指定したウィンドウに描画領域を配置します。		
書式	GUCREATE PIC <①ウィンドウ番号>, <②描画領域番号> [, [ <③フォント設定> ], [ <④サイズ設定> ], [ <⑤ポジション設定> ], [ <⑥ツールチップ設定> ] ]		
パラメータ	※ フォント設定以降の引数を省略する場合、各引数間のカンマは必要です。		
	①	<ウィンドウ番号>	数値
	グラフを配置するウィンドウの番号を指定します。		
	②	<描画領域番号>	数値
	作成する描画領域の番号を指定します。 1~10000の値を指定してください。		
	③	<フォント設定>	文字列
	GUI部品に表示される文字列のフォントを指定します。(参照：P. 20)		
	④	<サイズ設定>	文字列
	GUI部品に表示されるGUI部品の大きさを指定します。(参照：P. 22)		
	⑤	<ポジション設定>	文字列
	GUI部品に表示されるGUI部品の表示場所を指定します。(参照：P. 23)		
	⑥	<ツールチップ設定>	文字列
使用例	マウスポインタがGUI部品上に来た際に、表示されるツールチップの内容を指定します。 (参照：P. 24)		
	GUCREATE WINDOW 1 GUCREATE PIC 1, 10		
	<ウィンドウ番号> 1 のウィンドウに <描画領域番号> 10の描画領域を配置します。		

## 3.5.2.2 GUGCLS

命令			
機 能	描画領域に対して、クリアします。		
書 式	GUGCLS <①部品名> <②部品番号>		
パラ メータ	①	<部品名>	キーワード
	対象の部品名を指定します。 PIC(描画領域)		
	②	<部品番号>	数値
	描画対象の部品の番号を指定します。		
備 考	<ul style="list-style-type: none"> <li>・「GUCREATE PIC」で作成した描画領域に対して有効です。</li> <li>・呼び出し時、前景色が黒、背景色が白に、初期化されます。</li> </ul>		
使用例	GUCREATE PIC 1, 10 GUGCLS PIC 10		

## 3.5.2.3 GUGCOLOR

命令		
機 能	描画領域に対して、前景色および背景色を設定します。	
書 式	GUGCOLOR <①部品名> <②部品番号>, <③前景色> [, <④背景色> ]	
パラ メータ	①	<div> <div>&lt;部品名&gt;</div> <div>キーワード</div> </div> 描画対象の部品名を指定します。 PIC(描画領域)
	②	<div> <div>&lt;部品番号&gt;</div> <div>数値</div> </div> 描画対象の部品の番号を指定します。
	③	<div> <div>&lt;前景色&gt;</div> <div>数値</div> </div> 描画対象の前景色を指定します。 指定可能な組み合わせは、「フォント設定」の<文字色>の表を参照してください。
	④	<div> <div>&lt;背景色&gt;</div> <div>数値</div> </div> 描画対象の背景色を指定します。 指定可能な組み合わせは、「フォント設定」の<文字色>の表を参照してください。 省略すると、背景色は変更されません。
	備 考 ・「GUCREATE PIC」で作成した描画領域に対して有効です。	
使用例	GUCREATE PIC 1, 10 GUGCOLOR PIC 10, 2	

## 3.5.2.4 GUGLINE

命令		
機 能	描画領域に対して、線分を描画します。	
書 式	GUGLINE <①部品名> <②部品番号>, ( <③始点X座標>, <③始点Y座標> ) - ( <④終点X座標>, <④終点Y座標> ) [, <⑤前景色> ]	
パラ メータ	①	<div> <div>&lt;部品名&gt;</div> <div>キーワード</div> </div> 描画対象の部品名を指定します。 PIC(描画領域)
	②	<div> <div>&lt;部品番号&gt;</div> <div>数値</div> </div> 描画対象の部品の番号を指定します。
	③	<div> <div>&lt;始点X座標&gt;, &lt;始点Y座標&gt;</div> <div>数値</div> </div> 描画対象に対して、線分を描画する開始点の位置を丸カッコで挟んで指定します。
	④	<div> <div>&lt;終点X座標&gt;, &lt;終点Y座標&gt;</div> <div>数値</div> </div> 描画対象に対して、線分を描画する終了点の位置を丸カッコで挟んで指定します。
	⑤	<div> <div>&lt;前景色&gt;</div> <div>数値</div> </div> 線分を描画する際の、色を指定します。 省略すると、「GUGCOLOR」で指定した色が使用されます。
	備 考 ・「GUCREATE PIC」で作成した描画領域に対して有効です。	
使用例	GUCREATE PIC 1, 10 GUGLINE PIC 10, (5, 5) - (20, 30), 3	

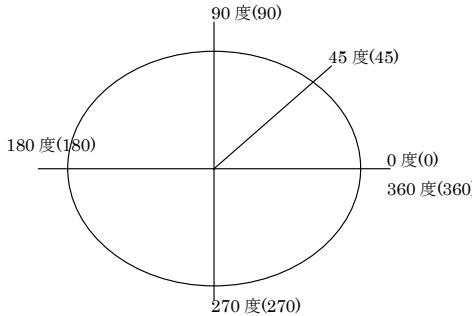
## 3.5.2.5 GUGRECTANGLE

命令			
機 能	描画領域に対して、矩形を描画します。		
書 式	GUGRECTANGLE <①部品名> <②部品番号>, ( <③始点X座標>, <③始点Y座標> ) - ( <④終点X座標>, <④終点Y座標> ) [, <⑤塗りつぶしスイッチ> [, <⑥前景色> ] ]		
パラ メータ	①	<部品名>	キーワード
	描画対象の部品名を指定します。 PIC(描画領域)		
	②	<部品番号>	数値
	描画対象の部品の番号を指定します。		
	③	<始点X座標>, <始点Y座標>	数値
	描画対象に対して、矩形を描画する開始点の位置を丸カッコで挟んで指定します。		
	④	<終点X座標>, <終点Y座標>	数値
	描画対象に対して、矩形を描画する終了点の位置を丸カッコで挟んで指定します。		
⑤	<塗りつぶしスイッチ>		真偽値
	TRUE : 矩形領域内を <前景色> で塗りつぶします。		
	FALSE(または省略) : 矩形描画のみです。		
⑥	<前景色>		数値
	矩形を描画する際の色を指定します。 省略すると、「GUGCOLOR」で指定した色が使用されます。		
備 考	・「GUCREATE PIC」で作成した描画領域に対して有効です。		
使用例	GUCREATE PIC 1, 10 GUGRECTANGLE PIC 10, (20, 20) - (40, 40), FALSE, 4		

3.5.2.6 GUGCIRCLE

命令			
機能	描画領域に対して、円を描画します。		
書式	GUGCIRCLE <①部品名> <②部品番号>, ( <③中心X座標>, <③中心Y座標> ), <④半径> [, <⑤前景色> [, <⑥開始角度> [, <⑥終了角度> [, <⑦塗りつぶしスイッチ> ] ] ] ]		
パラメータ	①	<部品名>	キーワード
	描画対象の部品名を指定します。 PIC(描画領域)		
	②	<部品番号>	数値
	描画対象の部品の番号を指定します。		
	③	<中心X座標>, <中心Y座標>	数値
	描画対象に対して、円を描画する中心点の位置を丸カッコで挟んで指定します。		
	④	<半径>	数値
	描画対象に対して、円を描画する半径を指定します。		
	⑤	<前景色>	数値
	円を描画する際の、色を指定します。 省略すると、「GUGCOLOR」で指定した色が使用されます。		
	⑥	<開始角度>, <終了角度>	数値
	円を描画する開始角度と終了角度を指定します。 角度の指定と関係性は、下図の通りです。		
	省略すると、開始角度：0、終了角度：360として扱われます。		
	⑦	<塗りつぶしスイッチ>	真偽値
	TRUE：円領域内を <前景色> で塗りつぶします。 FALSE(または省略)：円描画のみです。		
備考	・「GUCREATE PIC」で作成した描画領域に対して有効です。		
使用例	GUCREATE PIC 1, 10 GUGCIRCLE PIC 10, (50, 50), 30, 4		

## 3.5.2.7 GUGARC

命令		
機能書式	描画領域に対して、円弧を描画します。 GUGARC <①部品名> <②部品番号>, ( <③始点X座標>, <③始点Y座標> ) - ( <④横幅>, <④縦幅> ) [, <⑤塗りつぶしスイッチ> [, <⑥前景色> [, <⑦開始角度> [, <⑦終了角度> ] ] ] ]	
パラメータ	①	<部品名> 描画対象の部品名を指定します。 PIC(描画領域)
	②	<部品番号> 描画対象の部品の番号を指定します。
	③	<始点X座標>, <始点Y座標> 描画対象に対して、円弧を描画する開始点の位置を丸カッコで挟んで指定します。
	④	<横幅>, <縦幅> 描画対象に対して、円弧を描画する開始点から横幅および縦幅を丸カッコで挟んで指定します。
	⑤	<塗りつぶしスイッチ> TRUE：円弧領域内を <前景色> で塗りつぶします。 FALSE(または省略)：円弧描画のみです。
	⑥	<前景色> 円弧を描画する際の色を指定します。 省略すると、「GUGCOLOR」で指定した色が使用されます。
	⑦	<開始角度>, <終了角度> 円弧を描画する開始角度と終了角度を指定します。 角度の指定と関係性は、下図の通りです。
		
省略すると、開始角度：0、終了角度：360として扱われます。		
備考	・「GUCREATE PIC」で作成した描画領域に対して有効です。	
使用例	GUCREATE PIC 1, 10 GUGARC PIC 10, (50, 50)-(40, 40), TRUE, 2	

## 3.5.2.8 GUGPSET

命令			
機 能	描画領域に対して、点を描画します。		
書 式	GUGPSET <①部品名> <②部品番号>, ( <③X座標>, <③Y座標> ) [, <④前景色> ]		
パラ メータ	①	<部品名>	キーワード
	描画対象の部品名を指定します。 PIC(描画領域)		
	②	<部品番号>	数値
	描画対象の部品の番号を指定します。		
	③	<X座標>, <Y座標>	数値
備 考	描画対象に対して、点を描画する位置を丸カッコで挟んで指定します。		
	④	<前景色>	数値
	点を描画する際の、色を指定します。 省略すると、「GUGCOLOR」で指定した色が使用されます。		
使用例	GUCREATE PIC 1, 10 GUGPSET PIC 10, (5, 150), 5		



## 3.5.2.9 GUGDRAWTEXT

命令		
機能	描画領域に対して、テキスト文字を描画します。	
書式1	GUGDRAWTEXT <①部品名> <②部品番号>, ( <③始点X座標>, <③始点Y座標> ), <⑤文字列> [ , <⑥フォント設定> [ , <⑦角度> ] ]	
	始点座標から、テキスト文字を描画します。	
書式2	GUGDRAWTEXT <①部品名> <②部品番号>, ( <③始点X座標>, <③始点Y座標> ) - ( <④横幅>, <④縦幅> ), <⑤文字列> [ , <⑥フォント設定> [ , <⑦角度> ] ]	
	始点座標から横幅と縦幅の中に納まるよう、テキスト文字を描画します。	
パラメータ	①	<部品名> キーワード
	描画対象の部品名を指定します。 PIC(描画領域)	
	②	<部品番号> 数値
	描画対象の部品の番号を指定します。	
	③	<始点X座標>, <始点Y座標> 数値
	描画対象に対して、テキスト文字を描画する開始点の位置を丸カッコで挟んで指定します。	
	④	<横幅>, <縦幅> 数値
	書式2の時、テキスト文字を描画する横幅 および 縦幅の範囲を丸カッコで挟んで指定します。	
	⑤	<文字列> 文字列
	テキスト文字を描画する文字列を指定します。	
	⑥	<フォント設定> 文字列
	テキスト文字を描画する際の、文字列のフォントを指定します。(参照:P.20)	
	GUGDRAWTEXT時のみ、背景色に「0(透過)」を指定できます。	
	⑦	<角度> 数値
	テキスト文字を描画する際、指定角度に回転して描画します。	
備考	・「GUCREATE PIC」で作成した描画領域に対して有効です。	
使用例	GUCREATE PIC 1, 10,, "200,200",, GUGRECTANGLE PIC 10, (0, 0)-(200, 200), TRUE, 2  ' 文字色:1(黒)、背景色:3(緑)で、「hello」と描画します。 GUGDRAWTEXT PIC 10, (10, 10), "hello", "32, 0, 0, 14, 1, 3" ' 文字色:1(黒)、背景色:0(透過)で、「AJAN」と描画します。 GUGDRAWTEXT PIC 10, (10, 40), "AJAN", "32, 0, 0, 14, 1, 0" ' 横幅:100、縦幅:10の大きさで、「ようこそ」と描画します。 GUGDRAWTEXT PIC 10, (10, 80)-(100, 10), "ようこそ"	

## 3.5.2.10 GUGDRAWFILE

命令			
機能	描画領域に対して、画像ファイルの画像を描画します。		
書式	GUGDRAWFILE <①部品名> <②部品番号>, ( <③X座標>, <③Y座標> ), <④画像ファイルパス>		
パラメータ	①	<部品名>	キーワード
	描画対象の部品名を指定します。 PIC(描画領域)		
	②	<部品番号>	数値
	描画対象の部品の番号を指定します。		
	③	<X座標>, <Y座標>	数値
備考	描画対象に対して、画像ファイルの画像を描画する位置を丸カッコで挟んで指定します。		
	④	<画像ファイルパス>	文字列
	画像を描画する画像ファイルのパスを指定します。 指定可能な画像ファイルの形式は、おおよそ以下の通りです。 PNG、JPG、GIF、webp など。		
使用例	・「GUCREATE PIC」で作成した描画領域に対して有効です。 GUCREATE PIC 1, 10 GUGDRAWFILE PIC 10, (60, 60), "hoge.PNG"		

## 3.5.2.11 GUGSAVEFILE

命令		
機 能	描画領域の全てないしは指定範囲の画像を、画像ファイルに保存します。	
書 式 1	GUGSAVEFILE <①部品名> <②部品番号>, <③画像ファイルパス>  描画領域全てを、画像ファイルとして保存します。	
書 式 2	GUGSAVEFILE <①部品名> <②部品番号>, ( <④X座標>, <④Y座標> ) - ( <⑤横幅>, <⑤縦幅> ), <③画像ファイルパス>  描画領域の指定範囲を、画像ファイルとして保存します。	
パラ メータ	①	<部品名> キーワード 描画対象の部品名を指定します。 PIC(描画領域)
	②	<部品番号> 数値 描画対象の部品の番号を指定します。
	③	<画像ファイルパス> 文字列 画像を保存する画像ファイルのパスを指定します。 ファイルパス中の拡張子により、指定可能な画像ファイルの形式は、おおよそ以下の通りです。(拡張子から類推できない場合、PNG画像として保存します) .PNG、.JPG、.BMPなど。
	④	<X座標>, <Y座標> 数値 描画対象に対して、画像ファイルとして保存する画像の位置を丸カッコで挟んで指定します。
	⑤	<横幅>, <縦幅> 数値 描画対象に対して、画像ファイルとして保存する画像の横幅と縦幅を丸カッコで挟んで指定します。
	・ 「GUCREATE PIC」で作成した描画領域に対して有効です。	
備 考		
使用例	GUGSAVEFILE PIC 10, (0, 0) - (100, 50), "hoge.png" ' 10番の描画領域の、0,0 から 100, 50 までの画像を、"hoge.png" というファイル名で、PNG形式で保存します。	

## 3.5.2.12 GUGDATAGET

関数					
機能	描画領域の全てないしは指定範囲の画像情報を、配列データに取得します。				
書式1	<(戻り値)画像情報配列> = GUGDATAGET <①部品名> ( <②部品番号> )  描画領域全てを、画像情報として得られます。				
書式2	<(戻り値)画像情報配列> = GUGDATAGET <①部品名> ( <②部品番号>, <③X座標>, <③Y座標>, <④横幅>, <④縦幅> )  描画領域の指定範囲を、画像情報として得られます。				
戻り値	戻り値	<画像情報配列>		配列	
	画像情報のデータを、単精度整数の2次元配列形式として得られます。 1次元目が縦幅、2次元目が横幅です。				
	画像の1ドットの色情報が、1要素の値として格納されます。 色情報と値の関係は、下図のようにビット単位で割り当てられています。				
	bit	31～24	23～16	15～8	7～0
	意味	アルファ値(透過値) (0～255 に割り当て)	赤 (red) 値 (0～255 に割り当て)	緑 (green) 値 (0～255 に割り当て)	青 (blue) 値 (0～255 に割り当て)
パラメータ	①	<部品名>		キーワード	
	描画対象の部品名を指定します。 PIC(描画領域)				
	②	<部品番号>		数値	
	描画対象の部品の番号を指定します。				
	③	<X座標>, <Y座標>		数値	
描画対象に対して、画像情報として得る画像の位置を指定します。					
④	<横幅>, <縦幅>		数値		
描画対象に対して、画像情報として得る横幅と縦幅を指定します。					
備考	・「GUCREATE PIC」で作成した描画領域に対して有効です。				
使用例	LIST ARY ARY = GUGDATAGET PIC (10, 10, 10, 50, 30) PRINT "縦幅="; LDIM(ARY, 1) PRINT "横幅="; LDIM(ARY, 2) ' 10番の描画領域の、10,10 から 50, 30 までの画像情報を、配列形式で得ます。				

## 3.5.2.13 GUGDATASET

命令		
機 能	描画領域に対して、画像情報の配列データを元に、画像データとして直接書き込みます。	
書 式1	GUGDATASET <①部品名> <②部品番号>, ( <③X座標>, <③Y座標> ) - ( <④横幅>, <④縦幅> ), <⑤画像情報配列>	
書 式2	GUGDATASET <①部品名> <②部品番号>, ( <③X座標>, <③Y座標> ), <⑤画像情報配列>	
パラ メータ	①	<部品名>      キーワード
	描画対象の部品名を指定します。 PIC(描画領域)	
	②	<部品番号>      数値
	描画対象の部品の番号を指定します。	
	③	<X座標>, <Y座標>      数値
	描画対象に対して、画像情報を書き込む 画像の位置を丸カッコで挟んで指定します。	
	④	<横幅>, <縦幅>      数値
	描画対象に対して、画像情報を書き込む 横幅と縦幅を丸カッコで挟んで指定します。 書式2の時、2次元の画像情報配列で、1次元目を縦幅、2次元目を横幅に置き換えます。	
	⑤	<画像情報配列>      配列
描画対象に画像データとして書き込む為の画像情報を、単精度整数の配列で与えます。 データの形式は、「GUGDATAGET」を参照してください。		
備 考	<ul style="list-style-type: none"> <li>・「GUCREATE PIC」で作成した描画領域に対して有効です。</li> <li>・画像領域に書き込む際、元の画像との論理演算は行われず、上書きされます。 (色情報のアルファ値は、考慮されません)</li> </ul>	
使用例	<pre> LIST ARY ARY = GUGDATAGET PIC (10, 10, 10, 50, 30) PRINT "縦幅="; LDIM(ARY, 1) PRINT "横幅="; LDIM(ARY, 2)  ARY = DIMOR(ARY, &amp;HFF00) ' 色情報に、緑の情報をORする  GUGDATASET PIC 11, (0, 0) - (50, 30), ARY ' 10番の描画領域の、0,0 から 50, 30 までの画像情報を、配列形式で得た後、 ' 緑の情報を付加して、11番の画像領域に書き込みます。 </pre>	

### 3.5.2.14 GUGDRAWBLIT

命令		
機能	描画領域の特定範囲の画像を、別の描画領域に転送して貼り付けます。	
書式	GUGDRAWBLIT <①転送先部品名> <②転送先部品番号>, ( <③転送先X座標>, <③転送先Y座標> ) [ - ( <④転送先横幅>, <④転送先縦幅> ) ], <⑤転送元部品名> <⑥転送元部品番号>, ( <⑦転送元X座標>, <⑦転送元Y座標> ) - ( <⑧転送元横幅>, <⑧転送元縦幅> ) [ , <⑨論理演算オプション> ]	
パラメータ	①	<転送先部品名> キーワード 転送先 描画対象の部品名を指定します。 PIC(描画領域)
	②	<転送先部品番号> 数値 転送先 描画対象の部品の番号を指定します。
	③	<転送先X座標>, <転送先Y座標> 数値 転送先 描画対象に対して、画像データを描画する位置を丸カッコで挟んで指定します。
	④	<転送先横幅>, <転送先縦幅> 数値 転送先 描画対象に対して、画像データを描画する際の横幅と縦幅を丸カッコで挟んで指定します。 指定しないと、元の画像データの横幅と縦幅が採用されます。
	⑤	<転送元部品名> キーワード 転送元 描画対象の部品名を指定します。 PIC(描画領域)
	⑥	<転送元部品番号> 数値 転送元 描画対象の部品の番号を指定します。
	⑦	<転送元X座標>, <転送元Y座標> 数値 転送元 描画対象に対して、画像データを取得する位置を丸カッコで挟んで指定します。
	⑧	<転送元横幅>, <転送元縦幅> 数値 転送元 描画対象に対して、画像データを取得する際の横幅と縦幅を丸カッコで挟んで指定します。
	⑨	<論理演算オプション> 文字列 "SOURCE" を指定するか 省略してください。
備考	・「GUCREATE PIC」で作成した描画領域に対して有効です。	
使用例	GUGDRAWBLIT PIC 10, (10, 10), PIC 11, (0, 0) - (50, 50) ' 11番の描画領域 0, 0 の座標から 50, 50 の大きさの画像データを、10番の描画領域 10, 10の位置に転送する例です。	

## 3.6 画像処理に関する関数・命令



本章で紹介するコマンドは、Ver1.10より提供されました。

### 3.6.1 コマンド個別説明

#### 3.6.1.1 用語説明

用語	説明
カーネル	本章で説明するカーネルとは、画像のフィルタ処理に使用する係数です。
カーネルサイズ	対象画素を処理する際に使用する周囲 $n \times n$ の画素サイズです。

#### 3.6.1.2 IMG\_READ

関数													
機 能	ファイルから画像を読み込みます。												
書 式	<戻り値>読み込んだ画像 = IMG_READ( <①ファイル名>, <②カラータイプ> )												
戻り値	戻り値	<読み込んだ画像>	配列										
	ファイルから読み込んだ画像が得られます。 「画像の縦サイズ x 画像の横サイズ x チャンネル数」の3次元配列が得られます。 モノクロ画像として読み込んだ場合は「画像の縦サイズ x 画像の横サイズ」の2次元配列が得られます。												
パラ メータ	①	<ファイル名>	文字列										
	読み込む画像のファイル名を指定します。 読み込みできるファイル形式は、以下の通りです。												
	<table><tr><th>ファイル形式</th><th>拡張子</th></tr><tr><td>Windows bitmap</td><td>.bmp, .dib</td></tr><tr><td>JPEG file</td><td>.jpg, .jpeg, .jpe</td></tr><tr><td>Portable Network Graphics</td><td>.png</td></tr><tr><td>TIFF file</td><td>.tif</td></tr></table>			ファイル形式	拡張子	Windows bitmap	.bmp, .dib	JPEG file	.jpg, .jpeg, .jpe	Portable Network Graphics	.png	TIFF file	.tif
	ファイル形式	拡張子											
	Windows bitmap	.bmp, .dib											
	JPEG file	.jpg, .jpeg, .jpe											
Portable Network Graphics	.png												
TIFF file	.tif												
	②	<カラータイプ>	数値										
	画像のカラータイプを選択します。												
	<table><tr><th>設定値</th><th>動作</th></tr><tr><td>0</td><td>常にカラー画像として読み込みます。 読み込んだ画像のチャンネル数は 3 になります。 チャンネルはR(赤), G(緑), B(青)の順になります。</td></tr><tr><td>1</td><td>常にモノクロ画像として読み込みます。 読み込んだ画像のチャンネル数は 1 になります。</td></tr><tr><td>2</td><td>ファイルの画像と同じカラータイプになります。</td></tr></table>			設定値	動作	0	常にカラー画像として読み込みます。 読み込んだ画像のチャンネル数は 3 になります。 チャンネルはR(赤), G(緑), B(青)の順になります。	1	常にモノクロ画像として読み込みます。 読み込んだ画像のチャンネル数は 1 になります。	2	ファイルの画像と同じカラータイプになります。		
	設定値	動作											
	0	常にカラー画像として読み込みます。 読み込んだ画像のチャンネル数は 3 になります。 チャンネルはR(赤), G(緑), B(青)の順になります。											
1	常にモノクロ画像として読み込みます。 読み込んだ画像のチャンネル数は 1 になります。												
2	ファイルの画像と同じカラータイプになります。												
備 考	・読み込んだ画像の、各要素の値の範囲は0～255の整数です。												
使用例	' ファイル名SAMPLE.JPGから画像を読み込みます。 LIST IMAGE IMAGE = IMG_READ("SAMPLE.JPG", 0)												

## 3.6.1.3 IMG\_WRITE

関数													
機 能		指定したファイルに画像を保存します。											
書 式		<(戻り値)保存結果> = IMG_WRITE( <①ファイル名>, <②保存する画像> [, <③オプション> ] )											
戻り値	戻り値	<保存結果>	数値										
	正常に画像を保存できた場合は0が得られます。 それ以外の場合は1以上が得られます。												
パラ メータ	①	<ファイル名>	文字列										
	画像を保存するファイル名を指定します。 保存形式は拡張子から自動で判別します。 書き込みできるファイル形式は、以下の通りです。												
	<table><tr><th>ファイル形式</th><th>拡張子</th></tr><tr><td>Windows bitmap</td><td>.bmp, .dib</td></tr><tr><td>JPEG file</td><td>.jpg, .jpeg, .jpe</td></tr><tr><td>Portable Network Graphics</td><td>.png</td></tr><tr><td>TIFF file</td><td>.tif</td></tr></table>			ファイル形式	拡張子	Windows bitmap	.bmp, .dib	JPEG file	.jpg, .jpeg, .jpe	Portable Network Graphics	.png	TIFF file	.tif
	ファイル形式	拡張子											
	Windows bitmap	.bmp, .dib											
	JPEG file	.jpg, .jpeg, .jpe											
	Portable Network Graphics	.png											
	TIFF file	.tif											
	②	<保存する画像>	配列										
	保存する画像を指定します。 画素値は0～255の範囲内である必要があります。 0以下の値は黒、255以上の値は白として保存されます。 色の並びはR(赤), G(緑), B(青)の順で並んでいる必要があります。												
③	<オプション>	文字列											
画像を保存する際のオプションを文字列で指定します。 形式は"オプション1:設定値1;オプション2:設定値2;..."です。													
<table><tr><th>オプション</th><th>設定値</th></tr><tr><td>JPEG_QUALITY</td><td>jpegの保存品質を指定します。0から100までの値が指定できます。値が高いほどクオリティが高くなります。(デフォルト値: 95)</td></tr></table>			オプション	設定値	JPEG_QUALITY	jpegの保存品質を指定します。0から100までの値が指定できます。値が高いほどクオリティが高くなります。(デフォルト値: 95)							
オプション	設定値												
JPEG_QUALITY	jpegの保存品質を指定します。0から100までの値が指定できます。値が高いほどクオリティが高くなります。(デフォルト値: 95)												
使用例	' IMAGEをSAMPLE. JPGに保存します。 RESULT = IMG_WRITE("SAMPLE. JPG", IMAGE, "" ) ? RESULT												



## 3.6.1.4 IMG2GUG\_DATA

関数			
機 能	IMG_READで得た画像データを、GUGDATASETで使用する画像データに変換します。		
書 式	<(戻り値)出力画像> = IMG2GUG_DATA( <①入力画像> )		
戻り値	戻り値	<出力画像>	配列
	GUGDATASETで使用する画像データ(2次元配列)に変換します。		
パラ メータ	①	<入力画像>	配列
	IMG_READで得た画像データ(3次元配列)を指定します。		
備 考	・ 入力画像は、グレイ画像とカラー画像 両方対応します。		
使用例	<pre> ' IMG_READで得た画像データを、GUGDATASETで書く画像データに変換します。 LIST ARY%, ARY2% ARY% = IMG_READ("interface.png", 2) ARY2% = IMG2GUG_DATA(ARY%) ' この前で、GUGの初期化が行われている前提です。 GUGDATASET PIC 10, (0, 0), ARY2%</pre>		

## 3.6.1.5 GUG2IMG\_DATA

関数			
機 能	GUGDATAGETで得た画像データを、IMG_WRITE等で使用する画像データに変換します。		
書 式	<(戻り値)出力画像> = GUG2IMG_DATA( <①入力画像> )		
戻り値	戻り値	<出力画像>	配列
	IMG_WRITEで使用する画像データ(3次元配列)に変換します。		
パラ メータ	①	<入力画像>	配列
	GUGDATAGETで得た画像データ(2次元配列)を指定します。		
備 考	・ 出力画像は、常にカラー画像になります。		
使用例	<pre> ' GUGDATAGETで得た画像データを、IMG_WRITEで使用する画像データに変換します。 LIST ARY%, ARY2% ' この前で、GUGの初期化が行われている前提です。 ARY% = GUGDATAGET PIC(10, 0, 0, 50, 50) ARY2% = GUG2IMG_DATA(ARY%) PRINT IMG_WRITE("tmp.png", ARY2%)</pre>		

3.6.1.6 IMG\_CONVERT\_COLOR

関数		
機 能	画像の色空間を変換します。	
書 式	<(戻り値)出力画像> = IMG_CONVERT_COLOR( <①入力画像>, <②入力画像の色空間>, <②出力画像の色空間> )	
戻り値	戻り値	<出力画像> 配列 色空間を変換した画像が得られます。
パラ メータ	①	<入力画像> 配列 色空間を変換する画像を指定します。
	②	<入力画像の色空間>, <出力画像の色空間> 数値 入力画像と出力画像の色空間を指定します。
	設定値	色空間 説明
	0	RGB チャンネルがR, G, Bの順で並んでいる画像です。
	1	BGR チャンネルがB, G, Rの順で並んでいる画像です。
	2	GRAY モノクロ画像です。
使用例	' 読み込んだSAMPLE.JPGのカラー画像をモノクロ画像に変換します。 LIST IMG_RGB, IMG_GRAY IMG_RGB = IMG_READ("SAMPLE.JPG", 0) IMG_GRAY = IMG_CONVERT_COLOR(IMG_RGB, 0, 2)	

## 3.6.1.7 IMG\_RESIZE

関数		
機能	画像の大きさを変換します。	
書式	<(戻り値) 出力画像> = IMG_RESIZE( <①入力画像>, <②変換先の縦サイズ>, <③変換先の横サイズ>, <④ピクセル値の補間方法> )	
戻り値	戻り値	<div> <div>&lt;出力画像&gt;</div> <div>配列</div> </div> 大きさを変換した画像が得られます。
パラメータ	①	<div> <div>&lt;入力画像&gt;</div> <div>配列</div> </div> 大きさを変換する画像を指定します。
	②	<div> <div>&lt;変換先の縦サイズ&gt;</div> <div>数値</div> </div> 変換先画像の縦サイズを指定します。1以上の整数を指定します。
	③	<div> <div>&lt;変換先の横サイズ&gt;</div> <div>数値</div> </div> 変換先画像の横サイズを指定します。1以上の整数を指定します。
	④	<div> <div>&lt;ピクセル値の補間方法&gt;</div> <div>数値</div> </div> ピクセル値の補間方法を指定します。
	設定値	補間方法
	0	最近傍補間
	1	バイリニア補間
	2	バイキュービック補間
	3	面積平均法で補間
	4	ランチョス法で補間
使用例	' 面積平均法で画素値を補間した50x50の画像が得られます。 LIST IMAGE_SMALL, IMAGE_BIG IMAGE_BIG = IMG_READ("BIG_IMAGE.JPG", RGB) IMAGE_SMALL = IMG_RESIZE(IMAGE_BIG, 50, 50, 3)	

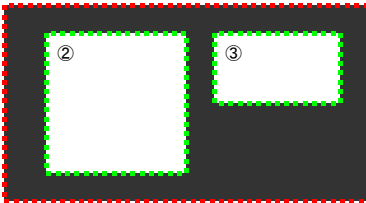
3.6.1.8 IMG\_BINARIZE

関数		
機能	画像を2値化します。 ある画素の値が与えられたしきい値より大きければ 画素の最大値 を割り当て、 そうでなければ 0 を割り当てます。	
書式	<(戻り値)2値化画像> = IMG_BINARIZE( <①入力画像>, <②しきい値>, <③画素の最大値> [, <④反転フラグ> ] )	
戻り値	戻り値	< 2値化画像> 配列
	2値化した画像が得られます。入力画像と同じ形式の配列が得られます。	
パラメータ	①	<入力画像> 配列
	2値化する画像を指定します。グレースケール画像である必要があります。	
	②	<しきい値> 数値
	2値化に使用するしきい値を指定します。	
	③	<画素の最大値> 数値
	2値化した画像における画素値の最大値を指定します。	
	④	<反転フラグ> 真偽値
	FALSE: 与えられたしきい値より大きい値を持つ画素に 画素の最大値 を割り当て、 そうでない画素に 0 を割り当てます。(デフォルト) TRUE: 与えられたしきい値より大きい値を持つ画素に 0 を割り当て、 そうでない画素に 画素の最大値 を割り当てます。	
使用例	' しきい値120でSAMPLE.JPGの画像を2値化します。 LIST IMAGE IMAGE = IMG_READ("SAMPLE.JPG") IMAGE = IMG_BINARIZE (IMAGE, 120, 255)	

## 3.6.1.9 IMG\_BINARIZE\_AUTO

関数			
機能	画像を2値化します。しきい値は入力画像から自動で求めます。 ある画素の値が与えられたしきい値より大きければ 画素の最大値 を割り当て、 そうでなければ 0 を割り当てます。		
書式	<(戻り値)2値化画像> = IMG_BINARIZE_AUTO( <①入力画像>, <②最大値> [, <③反転フラグ> ] )		
戻り値	戻り値	< 2値化画像>	配列
	2値化した画像が得られます。入力画像と同じ形式の配列が得られます。		
パラメータ	①	<入力画像>	配列
	2値化する画像を指定します。グレースケール画像である必要があります。		
	②	<最大値>	数値
	2値化した画像における画素値の最大値を指定します。		
	③	<反転フラグ>	真偽値
	FALSE: 与えられたしきい値より大きい値を持つ画素に 画素の最大値 を割り当て、 そうでない画素に 0 を割り当てます。(デフォルト) TRUE: 与えられたしきい値より大きい値を持つ画素に 0 を割り当て、 そうでない画素に 画素の最大値 を割り当てます。		
使用例	' 入力画像から自動的に求めたしきい値でSAMPLE. JPGの画像を2値化します。 LIST IMAGE IMAGE = IMG_READ("SAMPLE. JPG") IMAGE = IMG_BINARIZE_AUTO(IMAGE, 255)		

3.6.1.10 IMG\_FIND\_CONTOURS

関数												
機能	2値化画像から輪郭を抽出します。											
書式	<(戻り値)輪郭情報> = IMG_FIND_CONTOURS( <①2値化画像>, <②輪郭の抽出方法>, <③輪郭の近似方法> )											
戻り値	戻り値	<輪郭情報>	配列									
	抽出された輪郭情報は、以下の構造体型の配列が得られます。											
	<div>DEFINE STRUCT IMG_CONTOURS_HIERARCHY LIST CONTOURS        ’ 輪郭の頂点の一覧 PARENT_INDEX        ’ 一つ上の階層にある輪郭で、最初の輪郭の添字 CHILD_INDEX        ’ 一つ下の階層にある輪郭で、最初の輪郭の添字 PREV_INDEX        ’ 同じ階層にある輪郭で、一つ前の輪郭の添字 NEXT_INDEX        ’ 同じ階層にある輪郭で、一つ後の輪郭の添字 END STRUCT</div>											
	添字が存在しない場合は負の値が得られます。 以下の図で輪郭と階層について説明します。											
	<div>① </div>											
輪郭とは、異なる色の境界に沿って連続した線のことです。この図では、赤の点線で示した輪郭①と緑の点線で示した輪郭②と③が輪郭になります。輪郭②と③は輪郭①の中にあり、入れ子の構造になっています。このように入れ子になった輪郭では、外側にある輪郭①を親と呼び、内側にある輪郭②と③を子と呼びます。輪郭②の中にさらに輪郭があれば、②の中にある輪郭から見て②は親になります。このような親子関係を階層として記録します。階層情報を見ることによって、それぞれの輪郭の親子関係を知ることが出来ます。												
パラメータ	①	<2値化画像>	配列									
	輪郭を抽出する2値化画像を指定します。											
	②	<輪郭の抽出方法>	数値									
	輪郭の抽出方法を指定します。											
	<table><tr><th>設定値</th><th>抽出方法</th></tr><tr><td>0</td><td>親子関係を無視して全ての輪郭を抽出します。 PARENT_INDEXとCHILD_INDEXは存在しないため負の値になります。</td></tr><tr><td>1</td><td>最も外側にある輪郭のみ検出します。 PARENT_INDEXとCHILD_INDEXは存在しないため負の値になります。</td></tr><tr><td>2</td><td>全ての輪郭を抽出し、2つの階層に分けます。外側の輪郭は階層1、穴などの内側の輪郭は階層2に属します。階層2の中に輪郭がある場合、その輪郭は階層1となり、その中の穴は階層2になります。</td></tr><tr><td>3</td><td>全ての輪郭を抽出し、完全な階層構造を保持します。最も外側にある輪郭は階層1、その内側にある輪郭は階層2となり、内側の輪郭になるほど階層が増えます。</td></tr></table>			設定値	抽出方法	0	親子関係を無視して全ての輪郭を抽出します。 PARENT_INDEXとCHILD_INDEXは存在しないため負の値になります。	1	最も外側にある輪郭のみ検出します。 PARENT_INDEXとCHILD_INDEXは存在しないため負の値になります。	2	全ての輪郭を抽出し、2つの階層に分けます。外側の輪郭は階層1、穴などの内側の輪郭は階層2に属します。階層2の中に輪郭がある場合、その輪郭は階層1となり、その中の穴は階層2になります。	3
設定値	抽出方法											
0	親子関係を無視して全ての輪郭を抽出します。 PARENT_INDEXとCHILD_INDEXは存在しないため負の値になります。											
1	最も外側にある輪郭のみ検出します。 PARENT_INDEXとCHILD_INDEXは存在しないため負の値になります。											
2	全ての輪郭を抽出し、2つの階層に分けます。外側の輪郭は階層1、穴などの内側の輪郭は階層2に属します。階層2の中に輪郭がある場合、その輪郭は階層1となり、その中の穴は階層2になります。											
3	全ての輪郭を抽出し、完全な階層構造を保持します。最も外側にある輪郭は階層1、その内側にある輪郭は階層2となり、内側の輪郭になるほど階層が増えます。											
	③	<輪郭の近似方法>	数値									
	輪郭の近似方法を指定します。											
	<table><tr><th>設定値</th><th>近似方法</th></tr></table>			設定値	近似方法							
設定値	近似方法											

	<table border="1"> <tr> <td>0</td><td>輪郭上の全ての点を保持します。</td></tr> <tr> <td>1</td><td>輪郭を圧縮します。直線的に連続して並んでいる点は除去されます。</td></tr> <tr> <td>2</td><td>Teh-Chin アルゴリズムを使って輪郭を圧縮します。 (k-cosineを使用)</td></tr> <tr> <td>3</td><td>Teh-Chin アルゴリズムを使って輪郭を圧縮します。 (1-curvatureを使用)</td></tr> </table>	0	輪郭上の全ての点を保持します。	1	輪郭を圧縮します。直線的に連続して並んでいる点は除去されます。	2	Teh-Chin アルゴリズムを使って輪郭を圧縮します。 (k-cosineを使用)	3	Teh-Chin アルゴリズムを使って輪郭を圧縮します。 (1-curvatureを使用)
0	輪郭上の全ての点を保持します。								
1	輪郭を圧縮します。直線的に連続して並んでいる点は除去されます。								
2	Teh-Chin アルゴリズムを使って輪郭を圧縮します。 (k-cosineを使用)								
3	Teh-Chin アルゴリズムを使って輪郭を圧縮します。 (1-curvatureを使用)								
備 考	<ul style="list-style-type: none"> <li>構造体の定義は、GUI002.AJNにて定義されています。</li> <li>プログラムの先頭で「INCLUDE "GUI002.AJN"」を記述してください。</li> </ul>								
使用例	<p>' 完全な階層構造を保持しながら全ての輪郭を抽出します。</p> <pre>LIST IMAGE IMAGE = IMG_READ("SAMPLE.JPG", 1) ' SAMPLE.JPGから画像を読み込みます IMAGE = IMG_BINARIZE(IMAGE, 120, 255) ' しきい値120で画像を2値化します  LIST STRUCT IMG_CONTOURS_HIERARCHY CONTOURS_HIERARCHY CONTOURS_HIERARCHY = IMG_FIND_CONTOURS(IMAGE, 3, 1)</pre>								

3.6.1.11 IMG\_ERODE

関数			
機 能	画像に対して収縮処理を実行します。		
書 式	<(戻り値) 出力画像> = IMG_ERODE( <①入力画像>, <②カーネルサイズ> )		
戻り値	戻り値	<出力画像>	配列
	収縮処理を実行した結果が得られます。入力画像と同じ形式の配列が得られます。		
パラ メータ	①	<入力画像>	配列
	収縮処理を実行する画像を指定します。		
	②	<カーネルサイズ>	数値
	収縮処理で使用するカーネルサイズを指定します。		
	1以上の奇数を指定します。		
使用例	' 2値化したSAMPLE.JPGの画像に対して、カーネルサイズ3で収縮処理を実行します。 LIST IMAGE IMAGE = IMG_READ("SAMPLE.JPG", 1) IMAGE = IMG_BINARIZE (IMAGE, 120, 255) IMAGE = IMG_ERODE (IMAGE, 3)		



## 3.6.1.12 IMG\_DILATE

関数			
機 能	画像に対して膨張処理を実行します。		
書 式	<(戻り値) 出力画像> = IMG_DILATE( <①入力画像>, <②カーネルサイズ> )		
戻り値	戻り値	<出力画像>	配列
	膨張処理を実行した結果が得られます。入力画像と同じ形式の配列が得られます。		
パラ メータ	①	<入力画像>	配列
	膨張処理を実行する画像を指定します。		
	②	<カーネルサイズ>	数値
	膨張処理で使用するカーネルサイズを指定します。 1以上の奇数を指定します。		
使用例	' 2値化したSAMPLE.JPGの画像に対して、カーネルサイズ3で膨張処理を実行します。 LIST IMAGE IMAGE = IMG_READ("SAMPLE.JPG", 1) IMAGE = IMG_BINARIZE(IMAGE, 120, 255) IMAGE = IMG_DILATE(IMAGE, 3)		

3.6.1.13 IMG\_MEAN\_FILTER

関数			
機 能	平均フィルタを使用して画像を平滑化します。 画像内のノイズを軽減することができます。ただし、エッジなどもぼやけてしまいます。		
書 式	<(戻り値)出力画像> = IMG_MEAN_FILTER( <①入力画像>, <②カーネルサイズ> [, <③外挿方法> ] )		
戻り値	戻り値	<出力画像>	配列
	平均フィルタを使用して平滑化した画像が得られます。 入力画像と同じ形式の配列が得られます。		
パラメータ	①	<入力画像>	配列
	平滑化したい画像を指定します。		
	②	<カーネルサイズ>	数値
	カーネルの大きさを指定します。1以上の整数を指定します。 カーネルサイズを大きくするほど、平滑化の効果が強まります。		
	③	<外挿方法>	数値
	画像の外側を処理する際に画素の値を何にするかを指定します。 例に示している画素値の外挿はカーネルサイズ7を使用した場合です。		
設定値		外挿方法	
0		<div>外挿                      画像                      外挿</div> <div>1 1 1 1 2 3 4 5 5 5 5</div> <div>画像の一番端の画素値を繰り返しコピーします。</div>	
1		<div>外挿                      画像                      外挿</div> <div>3 2 1 1 2 3 4 5 5 4 3</div> <div>画像端の画素値を反転してコピーします。</div>	
2		<div>外挿                      画像                      外挿</div> <div>4 3 2 1 2 3 4 5 4 3 2</div> <div>画像端の画素値を反転してコピーしますが、一番端の画素値はコピーされません。(デフォルト)</div>	
使用例	' カーネルサイズ5の平均フィルタを使用して画像を平滑化します。 LIST IMAGE IMAGE = IMG_READ("SAMPLE.JPG", 1) IMAGE = IMG_MEAN_FILTER(IMAGE, 5)		

## 3. 6. 1. 14 IMG\_GAUSSIAN\_FILTER

関数		
機能	ガウシアンフィルタを使用して画像を平滑化します。 平均フィルタを使用した場合と比較してより自然に平滑化されます。 ただし、エッジもぼやけてしまいます。	
書式	〈(戻り値) 出力画像〉 = IMG_GAUSSIAN_FILTER( 〈①入力画像〉, 〈②カーネルサイズ〉, 〈③標準偏差〉 [, 〈④外挿方法〉 ] )	
パラメータ	戻り値	<div> <div>〈出力画像〉</div> <div>配列</div> </div> ガウシアンフィルタを使用して平滑化した画像が得られます。 入力画像と同じ形式の配列が得られます。
	①	<div> <div>〈入力画像〉</div> <div>配列</div> </div> 平滑化したい画像を指定します。
	②	<div> <div>〈カーネルサイズ〉</div> <div>数値</div> </div> カーネルの大きさを指定します。カーネルサイズが大きいほど、平滑化の効果が強まります。 正の奇数を指定します。または、0を指定することも可能です。 0を指定した場合は、標準偏差からカーネルサイズを計算します。
	③	<div> <div>〈標準偏差〉</div> <div>数値</div> </div> ガウシアンフィルタの標準偏差を指定します。 標準偏差が大きいほど、平滑化の効果が強まります。 0より大きい実数を指定します。または、0を指定することも可能です。 0を指定した場合は、カーネルサイズから標準偏差を計算します。
	④	<div> <div>〈外挿方法〉</div> <div>数値</div> </div> 画像の外側を処理する際に画素の値を何にするかを指定します。 設定値は「IMG_MEAN_FILTER」を参照してください。
備考	・カーネルサイズと標準偏差の両方に0を指定することは出来ません。	
使用例	・カーネルサイズ5、標準偏差1.4のガウシアンフィルタを使用して画像を平滑化します。 <pre> LIST IMAGE IMAGE = IMG_READ("SAMPLE.JPG", 1) IMAGE = IMG_GAUSSIAN_FILTER(IMAGE, 5, 1.4) </pre>	

## 3.6.1.15 IMG\_BILATERAL\_FILTER

関数		
機能	バイラテラルフィルタを使用して画像を平滑化します。 エッジを保存しつつ平滑化を行うことができます。 ただし、標準偏差のパラメータによってはエッジがぼやけることがあります。	
書式	〈(戻り値)出力画像〉 = IMG_BILATERAL_FILTER( 〈①入力画像〉, 〈②カーネルサイズ〉, 〈③色空間の標準偏差〉, 〈④座標空間の標準偏差〉 [, 〈⑤外挿方法〉 ] )	
戻り値	戻り値	〈出力画像〉 配列
	バイラテラルフィルタを使用して平滑化した画像が得られます。 入力画像と同じ形式の配列が得られます。	
パラメータ	①	〈入力画像〉 配列
	平滑化したい画像を指定します。	
	②	〈カーネルサイズ〉 数値
	カーネルの大きさを指定します。1以上の整数を指定します。 カーネルサイズを大きくするほど、平滑化の効果が強まります。	
	③	〈色空間の標準偏差〉 数値
	0より大きい実数を指定します。 このパラメータを大きくするほど、ノイズを除去する効果が強まります。 ただし、エッジがなまりやすくなります。	
	④	〈座標空間の標準偏差〉 数値
	0より大きい実数を指定します。 このパラメータを大きくするほど、より遠くの画素値を使って平滑化を行います。	
	⑤	〈外挿方法〉 数値
使用例	画像の外側を処理する際に画素の値を何にするかを指定します。 設定値は「IMG_MEAN_FILTER」を参照してください。	
	<pre> ' カーネルサイズ5、色空間の標準偏差1.4、座標空間の標準偏差2の ' バイラテラルフィルタを使用して画像を平滑化します。 LIST IMAGE IMAGE = IMG_READ("SAMPLE.JPG", 1) IMAGE = IMG_BILATERAL_FILTER(IMAGE, 5, 1.4, 2) </pre>	

### 3.6.1.16 IMG\_SOBEL\_FILTER

関数								
機 能	ソーベルフィルタを使用して画像からX方向またはY方向のエッジを抽出します。							
書 式	〈(戻り値)出力画像〉 = IMG_SOBEL_FILTER( 〈①入力画像〉, 〈②カーネルサイズ〉, 〈③エッジ抽出方向〉 [, 〈④外挿方法〉 ] )							
戻り値	戻り値	〈出力画像〉 配列 ソーベルフィルタを使用して入力画像からエッジを抽出した画像が得られます。 入力画像と同じ形式の配列が得られます。						
	①	〈入力画像〉 配列 エッジを抽出したい画像を指定します。						
パラメータ	②	〈カーネルサイズ〉 数値 カーネルの大きさを指定します。1～7までの奇数を指定します。 カーネルサイズが大きいほど、より広範囲の画素を参照しながらエッジを抽出します。						
	③	〈エッジ抽出方向〉 数値 エッジを抽出する方向を指定します。						
	<table><tr><td>設定値</td><td>説明</td></tr><tr><td>0</td><td>X方向のエッジを抽出します。</td></tr><tr><td>1</td><td>Y方向のエッジを抽出します。</td></tr></table>		設定値	説明	0	X方向のエッジを抽出します。	1	Y方向のエッジを抽出します。
	設定値	説明						
	0	X方向のエッジを抽出します。						
	1	Y方向のエッジを抽出します。						
④	〈外挿方法〉 数値 画像の外側を処理する際に画素の値を何にするかを指定します。 設定値は「IMG_MEAN_FILTER」を参照してください。							
使用例								
’ カーネルサイズ3のソーベルフィルタを使用して、X方向およびY方向のエッジを抽出します LIST IMAGE IMAGE = IMG_READ(“SAMPLE.JPG”, 1)  LIST EDGE_X, EDGE_Y EDGE_X = IMG_SOBEL_FILTER(IMAGE, 3, 0) EDGE_Y = IMG_SOBEL_FILTER(IMAGE, 3, 1)								

## 3.6.1.17 IMG\_LAPLACIAN\_FILTER

関数			
機 能	ラプラシアンフィルタを使用して画像からエッジを抽出します。 ラプラシアンフィルタでは、方向に依存しないエッジが直接得られます。		
書 式	〈(戻り値)出力画像〉 = IMG_LAPLACIAN_FILTER( 〈①入力画像〉, 〈②カーネルサイズ〉 [, 〈③外挿方法〉 ] )		
戻り値	戻り値	〈出力画像〉	配列
	ラプラシアンフィルタを使用して入力画像からエッジを抽出した画像が得られます。 入力画像と同じ形式の配列が得られます。		
パラ メータ	①	〈入力画像〉	配列
	エッジを抽出したい画像を指定します。		
	②	〈カーネルサイズ〉	数値
	カーネルの大きさを指定します。1～31の奇数を指定します。 カーネルサイズが大きいほど、より広範囲の画素を参照しながらエッジを抽出します。		
	③	〈外挿方法〉	数値
	画像の外側を処理する際に画素の値を何にするかを指定します。 設定値は「IMG_MEAN_FILTER」を参照してください。		
使用例	' カーネルサイズ5のラプラシアンフィルタを使用してエッジを抽出します LIST IMAGE IMAGE = IMG_READ("SAMPLE.JPG", 1) IMAGE = IMG_LAPLACIAN_FILTER(IMAGE, 5)		

## 3.6.1.18 IMG\_SHARP\_FILTER

関数		
機 能	鮮鋭化フィルタを使用して、エッジが強調された画像を得ます。	
書 式	<(戻り値) 出力画像> = IMG_SHARP_FILTER( <①入力画像>, <②カーネルサイズ>, <③強調度> [, <④外挿方法> ] )	
戻り値	戻り値	<出力画像> 配列
	エッジが強調された画像が得られます。 入力画像と同じ形式の配列が得られます。	
パラ メータ	①	<入力画像> 配列
	エッジを強調したい画像を指定します。	
	②	<カーネルサイズ> 数値
	カーネルの大きさを指定します。正の奇数を指定します。 カーネルサイズが大きいほど、より広範囲の画素を参照しながらエッジを強調します。	
	③	<強調度> 数値
	正の数値を指定します。 大きな値にするほどエッジが強調されます。	
	④	<外挿方法> 数値
	画像の外側を処理する際に画素の値を何にするかを指定します。 設定値は「IMG_MEAN_FILTER」を参照してください。	
使用例	' カーネルサイズ5の鮮鋭化フィルタを使用してエッジを強調します。 LIST IMAGE IMAGE = IMG_READ("SAMPLE.JPG", 1) IMAGE = IMG_SHARP_FILTER(IMAGE, 5)	

## 3.6.1.19 IMG\_CANNY\_EDGE

関数			
機 能	キャニーのエッジ検出を実行します。		
書 式	<(戻り値)エッジ画像> = IMG_CANNY_EDGE ( <①入力画像>, <②下側しきい値>, <②上側しきい値> [, <③フィルタサイズ>[, <④L2フラグ> ] ] )		
戻り値	戻り値	<エッジ画像>	配列
	エッジ検出結果が得られます。 入力画像と同じ形式の配列が得られます。		
パラ メータ	①	<入力画像>	配列
	エッジを検出する画像を指定します。		
	②	<下側しきい値>, <上側しきい値>	数値
	下側しきい値と上側しきい値には、画素値の微分値のしきい値を指定します。 画素値の微分値が下側しきい値以下の場合にはエッジではないと判定し、上側しきい値以上の場合はエッジであると判定します。 微分値が下側しきい値と上側しきい値の間にある場合は、周りの画素値を考慮してエッジであるかを判定します。		
	③	<フィルタサイズ>	数値
使用例	画素値の微分値を計算するために内部でソーベルフィルタを使用しています。 フィルタサイズでは使用するソーベルフィルタの大きさを指定します。(デフォルト: 3) 1～7の奇数を指定します。		
	④	< L2フラグ>	真偽値
	勾配の大きさの計算方法を指定します。  TRUE: より正確なL2ノルムを使用します。L2ノルム : $\sqrt{dx^2 + dy^2}$  FALSE: 計算が少ないL1ノルムを使用します。L1ノルム : $ dx  +  dy $ (デフォルト)		
使用例	' SAMPLE.JPGからエッジを抽出します。		
	LIST IMAGE		
	IMAGE = IMG_READ("SAMPLE.JPG", 1)		
	LIST EDGE		
使用例	EDGE = IMG_CANNY_EDGE (IMAGE, 100, 200)		



## 3.6.1.20 IMG\_GET\_ROTATION\_MATRIX

関数			
機 能	アフィン変換で画像を回転するために使用する変換行列を求めます。		
書 式	<(戻り値) 変換行列> = IMG_GET_ROTATION_MATRIX( <①回転の中心のx座標>, <②回転の中心のy座標>, <③回転角度> [, <④画像の大きさの倍率> ] )		
戻り値	戻り値	<変換行列>	配列
	2×3の変換行列が得られます。		
パラ メータ	①	<回転の中心のx座標>	数値
	回転の中心のx座標を指定します。		
	②	<回転の中心のy座標>	数値
	回転の中心のy座標を指定します。		
	③	<回転角度>	数値
	画像を回転する角度を指定します。		
	④	<画像の大きさの倍率>	数値
	画像の大きさの倍率を指定します。 デフォルトの値は1です。		
使用例	' 座標(100, 100)を中心に、画像を左周りに20度回転して2倍に拡大する変換行列が得られる LIST ROTATION_MATRIX ROTATION_MATRIX = IMG_GET_ROTATION_MATRIX(100, 100, 20, 2)		

## 3.6.1.21 IMG\_WARP\_AFFINE

関数			
機能	画像に対して与えられた変換行列でアフィン変換を行います。		
書式	<(戻り値)変換画像> = IMG_WARP_AFFINE( <①入力画像>, <②変換行列>, <③出力画像の縦サイズ>, <④出力画像の横サイズ>, <⑤画素値の補間方法> [, <⑥逆変換フラグ> ] )		
戻り値	戻り値	<変換画像>	配列
アフィン変換した結果が得られます。			
パラメータ	①	<入力画像>	配列
	アフィン変換する画像を指定します。		
	②	<変換行列>	配列
	2×3の変換行列を指定します。		
	③	<出力画像の縦サイズ>	数値
	出力画像の縦サイズを指定します。1以上の整数を指定します。		
	④	<出力画像の横サイズ>	数値
	出力画像の横サイズを指定します。1以上の整数を指定します。		
	⑤	<画素値の補間方法>	数値
	画素値の補間方法を指定します。		
	設定値は「IMG_RESIZE」を参照してください。		
	⑥	<逆変換フラグ>	真偽値
	TRUE: 逆変換を実行します。		
	FALSE: 通常通りのアフィン変換を行います。(デフォルト)		
使用例	<pre> LIST IMAGE IMAGE = IMG_READ("SAMPLE.JPG", 1)          ' 画像を読み込みます。  LIST ROTATION_MATRIX ROTATION_MATRIX = IMG_GET_ROTATION_MATRIX(100, 100, 20, 1)  H = LDIM(IMAGE, 1) W = LDIM(IMAGE, 2)  ' 中心座標(100, 100)で左に20度回転した画像が得られます。 IMAGE = IMG_WARP_AFFINE(IMAGE, ROTATION_MATRIX, H, W, 0) </pre>		

## 3.6.1.22 IMG\_MATCH\_TEMPLATE

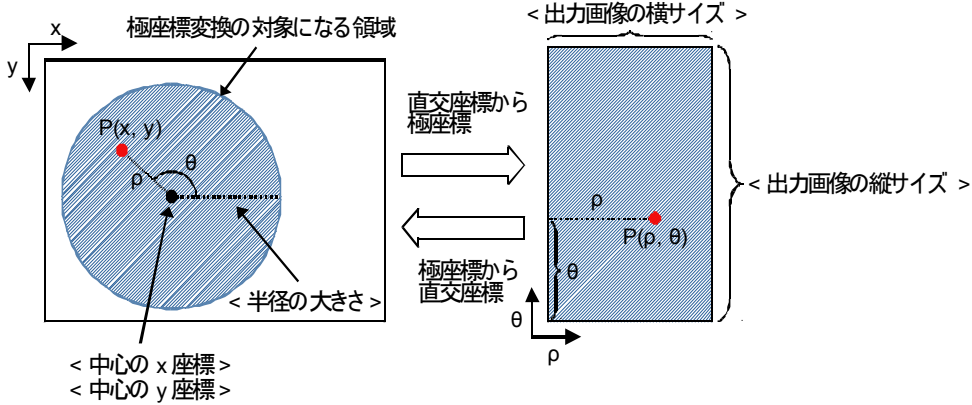
関数		
機能	入力画像とテンプレート画像の相関関数を計算します。	
書式	〈(戻り値) 計算結果〉 = IMG_MATCH_TEMPLATE( 〈①入力画像〉, 〈②テンプレート画像〉, 〈③相関関数〉 )	
戻り値	戻り値	〈計算結果〉 配列
	相関関数の計算結果が得られます。 入力画像とテンプレート画像の配列の大きさをそれぞれ(H, W)、(h, w)とすると、計算結果の配列の大きさは、「(H - h + 1) × (W - w + 1) × 1」の単精度実数型の3次元配列になります。	
パラメータ	①	〈入力画像〉 配列
	テンプレート画像を探索する画像を指定します。	
	②	〈テンプレート画像〉 配列
	テンプレート画像を指定します。 テンプレートの縦サイズと横サイズは入力画像よりも小さい必要があります。	
	③	〈相関関数〉 数値
相関関数として使用する関数を指定します。 以下の説明で使用している数式について、テンプレート画像の位置(x, y)における画素の値をT(x, y)とおき、テンプレート画像と重ね合わせた入力画像の対応する画素値をI(x, y)とおきます。		
設定値	相関関数	説明
0	SSD	輝度値の差の2乗和を計算します。 $R = \sum_{x,y} (I(x, y) - T(x, y))^2$ 画像同士の疑似度が大きいほど値は小さくなります。
1	正規化SSD	正規化したSSDを計算します。 $R = \frac{\sum_{x,y} (I(x, y) - T(x, y))^2}{\sqrt{\sum_{x,y} I(x, y)^2 \cdot T(x, y)^2}}$
2	相互相関	輝度値の積の総和を計算します。 $R = \sum_{x,y} (I(x, y) \cdot T(x, y))$ 画像同士の疑似度が大きいほど値は大きくなります。
3	正規化相互相関(NCC)	正規化した相互相関を計算します。 $R = \frac{\sum_{x,y} (I(x, y) \cdot T(x, y))}{\sqrt{\sum_{x,y} I(x, y)^2 \cdot T(x, y)^2}}$
4	相関係数	輝度値の平均を引いてから相互相関を計算します。 $R = \sum_{x,y} ((I(x, y) - \bar{I})(T(x, y) - \bar{T}))$ $\bar{I} = \frac{\sum_{x',y'} I(x', y')}{wh}, \bar{T} = \frac{\sum_{x',y'} T(x', y')}{wh}$ 画像同士の疑似度が大きいほど値は大きくなります。

	5	正規化相関係数 (ZNCC)	正規化した相関係数を計算します。 $R = \frac{\sum_{x,y} ((I(x,y) - \bar{I})(T(x,y) - \bar{T}))}{\sqrt{\sum_{x,y} (I(x,y) - \bar{I})^2 (T(x,y) - \bar{T})^2}}$ $\bar{I} = \frac{\sum_{x',y'} I(x',y')}{wh}, \bar{T} = \frac{\sum_{x',y'} T(x',y')}{wh}$	
使用例	' 正規化相関係数を使用して、IMAGE1とIMAGE2の疑似度を計算します。 LIST IMAGE1, IMAGE2 IMAGE1 = IMG_READ("SAMPLE1", 1)    ' テンプレート画像を探索する画像です IMAGE2 = IMG_READ("SAMPLE2", 2)    ' テンプレート画像です  LIST RESULT RESULT = IMG_MATCH_TEMPLATE (IMAGE1, IMAGE2, 5)			

## 3. 6. 1. 23 IMG\_PHASE\_CORRELATION

関数			
機 能	位相限定相関法を使用して、与えられた2枚の画像の位置ずれを推定します。		
書 式	<(戻り値)結果> = IMG_PHASE_CORRELATION( <①画像1>, <①画像2> )		
戻り値	戻り値	<結果>	構造体
	<p>推定された2枚の画像のずれは、以下の構造体で得られます。</p> <pre> DEFINE STRUCT IMG_POC_RESULT     X      ' x座標のずれ     Y      ' y座標のずれ     E      ' 推定値の強度 END STRUCT </pre> <p>推定値の強度Eは、0から1の値が得られます。</p>		
パラメータ	①	<画像1>, <画像2>	配列
	<p>推定に使用する画像を指定します。</p> <p>どちらの画像も縦サイズと横サイズ、チャンネル数が等しい必要があります。</p> <p>カラー画像を入力した場合はRGB画像としてグレースケールへ変換した後に、計算を行います。</p>		
備 考	<ul style="list-style-type: none"> <li>構造体の定義は、GUI002.AJNにて定義されています。</li> <li>プログラムの先頭で「INCLUDE "GUI002.AJN"」を記述してください。</li> </ul>		
使用例	<pre> LIST IMAGE1, IMAGE2 IMAGE1 = IMG_READ("SAMPLE1.JPG", 1) IMAGE2 = IMG_READ("SAMPLE2.JPG", 1)  STRUCT IMG_POC_RESULT RESULT RESULT = IMG_PHASE_CORRELATION(IMAGE1, IMAGE2)  ? "x座標のずれ: "; RESULT.X ? "y座標のずれ: "; RESULT.Y ? "推定値の強度: "; RESULT.E </pre>		

3.6.1.24 IMG\_WARP\_POLAR

関数		
機能	画像に対して極座標変換を行います。	
書式	<(戻り値) 出力画像> = IMG_WARP_POLAR( <①入力画像>, <②中心のx座標>, <③中心のy座標>, <④出力画像の縦サイズ>, <⑤出力画像の横サイズ>, <⑥半径の大きさ>, <⑦画素値の補間方法> [, <⑧対数フラグ> [, <⑨逆変換フラグ> ] ] )	
戻り値	戻り値	<出力画像> 配列
極座標に変換した画像が得られます。		
パラメータ	①	<入力画像> 配列
	極座標変換する画像を指定します。	
	②	<中心のx座標> 数値
	対象領域の中心のx座標を指定します。	
	③	<中心のy座標> 数値
	対象領域の中心のy座標を指定します。	
	④	<出力画像の縦サイズ> 数値
	出力画像の縦サイズを指定します。1以上の整数を指定します。 縦サイズは円周の分解能にあたります。 0を指定した場合、出力画像の縦サイズは <出力画像の横サイズ> × $\pi$ になります。 出力画像の横サイズも0を指定した場合、出力画像の縦サイズは <半径の大きさ> × $\pi$ になります。	
	⑤	<出力画像の横サイズ> 数値
	出力画像の横サイズを指定します。1以上の整数を指定します。 横サイズは円の半径の分解能にあたります。 0を指定した場合、縦サイズは <半径の大きさ> と等しくなります。	
	⑥	<半径の大きさ> 数値
	対象領域の半径の大きさを指定します。	
	⑦	<画素値の補間方法> 数値
	画素値の補間方法を指定します。 設定値は「IMG_RESIZE」を参照してください。	
	⑧	<対数フラグ> 真偽値
	TRUE: 対象領域の中心からの距離を対数スケールに変換した極座標変換を行います。 FALSE: 対象領域の中心からの距離をそのまま使用した極座標変換を行います。(デフォルト)	
	⑨	<逆変換フラグ> 真偽値
	TRUE: 極座標から直角座標に変換します。 FALSE: 直角座標から極座標に変換します。(デフォルト)	
備考	<p>・それぞれの引数は以下の画像と対応します。</p>  <p>&lt; 中心の x 座標 &gt; &lt; 中心の y 座標 &gt;</p>	
使用例	<pre>' 画像に対して中心座標(X_CENTER, Y_CENTER)と半径Rで極座標変換を行います。 LIST IMAGE IMAGE = IMG_READ("SAMPLE.JPG", 0) ' 画像を読み込みます</pre>	

	<pre>Y = LDIM(IMAGE, 1)    ' Y座標 X = LDIM(IMAGE, 2)    ' X座標  X_CENTER = X / 2      ' 画像の中心のX座標 Y_CENTER = Y / 2      ' 画像の中心のY座標  R = SQR(X_CENTER * X_CENTER + Y_CENTER * Y_CENTER) ' 半径の大きさ  LIST POLAR_IMAGE POLAR_IMAGE = IMG_WARP_POLAR(IMAGE, X_CENTER, Y_CENTER, X, Y, R, 0)</pre>
--	--

## 第4章 サンプルプログラム

サンプルプログラムは「/usr/share/interface/AJANPro/samples/GUI/」に格納されています。

AJAN統合開発環境を起動すると、左ペインのエクスプローラウィンドウ内の「Samples/GUI/」に、ファイルが取り込まれて配置されます。

### 4.1 サンプルプログラム

#	ファイル名	内容
GUI		
1	BUTTON. AJN	ウィンドウにボタンを配置し、ボタンをクリックした際に、呼び出されるサブルーチン(BUTTON_EVENT)を登録します。 ボタンをクリックすると、クリックしたボタンの情報が表示されます。 ウィンドウの閉じるボタンをクリックすると終了します
2	CALENDAR. AJN	ウィンドウにカレンダーを配置し、カレンダーの日付を選択した際に、呼び出されるサブルーチン(CALENDAR_EVENT)を登録します。 カレンダーの日付をクリックすると、クリックしたカレンダーの情報(GUGET\$ CALENDAR 関数で日付も)が表示されます。 ウィンドウの閉じるボタンをクリックすると終了します
3	CHECKBOX. AJN	ウィンドウにチェックボックスを配置し、チェックボックスをクリックした際に、呼び出されるサブルーチン(CHECKBOX_EVENT)を登録します。 チェックボックスをクリックすると、クリックしたチェックボックスの情報(GUGET\$ CHECKBOX 関数でチェック内容も)が表示されます。 ウィンドウの閉じるボタンをクリックすると終了します
4	COMBOBOX. AJN	ウィンドウにコンボボックスを配置し、コンボボックスのリストを選択した際に、呼び出されるサブルーチン(COMBOBOX_EVENT)を登録します。 コンボボックスのリストを選択すると、選択したコンボボックスの情報(GUGET\$ COMBOBOX 関数で選択したリストの値も)が表示されます。 ウィンドウの閉じるボタンをクリックすると終了します
5	IMAGE. AJN	ウィンドウに画像イメージを配置し、画像を表示します。 表示する画像は、通常の画像の貼り付け、画像サイズを指定して貼り付け、画像を回転して貼り付け、の3パターンを用意しています。 ウィンドウの閉じるボタンをクリックすると終了します。
6	LABEL. AJN	ウィンドウにラベルを配置し、ラベルをクリックした際に、呼び出されるサブルーチン(LABEL_EVENT)を登録します。 配置されるラベルは、ラベルの文字列を左寄せ、中央寄せ、右寄せの、3パターンを用意しています。 ラベルをクリックすると、クリックしたラベルの情報(GUGET\$ LABEL 関数でラベル文字列も)が表示されます。 ウィンドウの閉じるボタンをクリックすると終了します。
7	RADIOBTN. AJN	ウィンドウにラジオボタンを配置し、ラジオボタンの項目を選択した際に、呼び出されるサブルーチン(RADIOBTN_EVENT)を登録します。 ラジオボタンの項目を選択すると、選択したラジオボタンの情報(GUGET\$ RADIOBTN 関数で選択したテキスト内容も)が表示されます。 ウィンドウの閉じるボタンをクリックすると終了します。
8	SCALE. AJN	ウィンドウにスケールを配置し、スケールのゲージをスライドして変化した際に、呼び出されるサブルーチン(SCALE_EVENT)を登録します。 スケールのゲージをスライドさせると、スライドさせたスケールの情報(GUGET\$ SCALE 関数でゲージの設定値も)が表示されます。 ウィンドウの閉じるボタンをクリックすると終了します。
9	SWITCH. AJN	ウィンドウにスイッチを配置し、スイッチをクリックしてオン / オフした



#	ファイル名	内容
		際に、呼び出されるサブルーチン(SWITCH_EVENT)を登録します。 スイッチをクリックしてオン / オフすると、クリックしたスイッチの情報(GUGET\$ SWITCH 関数でスイッチの状態も)が表示されます。 ウィンドウの閉じるボタンをクリックすると終了します。
10	TEXTAREA. AJN	ウィンドウにテキストエリアとボタンを配置し、テキストエリアでキー入力を行った際に呼び出されるサブルーチン(TEXTAREA_EVENT)を登録します。 また、ボタンをクリックする際に呼び出されるサブルーチン(BUTTON_EVENT)を登録します。 テキストエリアでキー入力すると、キー入力したテキストエリアの情報が表示されます。 ボタンをクリックすると、GUGET\$ TEXTAREA 関数で、テキストエリアのテキスト内容が表示されます。 ウィンドウの閉じるボタンをクリックすると終了します。
11	TEXTBOX. AJN	ウィンドウにテキストボックスを配置し、テキストボックスでEnter キーを押した際に、呼び出されるサブルーチン(TEXTBOX_EVENT)を登録します。 テキストボックスで文字列を入力してEnter キーを押すと、Enter キーを押したテキストボックスの情報(GUGET\$ TEXTBOX 関数でテキスト内容も)が表示されます。 ウィンドウの閉じるボタンをクリックすると終了します。
12	GUDSP. AJN	ボタンをクリックする事で、ラベルの表示状態を切り替えて変更します。 表示状態の切り替えは、GUDSP 命令で行います。 ウィンドウの閉じるボタンをクリックすると終了します
13	GUMAIN. AJN	ボタンサンプル(BUTTON. AJN)の構成で、GUMAIN ONELOOP 関数を使って、自分でGUI メインイベントループを回して制御する例です。 ウィンドウにボタンを配置し、ボタンをクリックした際に、呼び出されるサブルーチン(BUTTON_EVENT)を登録します。 GUMAIN LOOP 命令を使っていたのを、GUMAIN ONELOOP 関数を繰り返し呼び出す事で、自分でGUI メインイベントループを回しています。 ウィンドウの閉じるボタンをクリックすると、GUMAIN QUIT 命令で終了指示を出していたのを、フラグ(IS_EXIT 変数)を立てて、自前のDO WHILE - LOOP を終了しています。
14	GUMOUSE. AJN	実行時、マウスの情報(左ボタンが押されているか否か)を、取得して表示します。
15	GUMOVE. AJN	ウィンドウとボタンを生成し、ボタンをクリックするとウィンドウを移動させます。 GUMOVE WINDOW 命令で、ウィンドウを、指定位置に移動します。 ウィンドウの閉じるボタンをクリックすると終了します。
16	GUMSGBOX. AJN	ウィンドウを表示した後、メッセージボックスを表示します。 メッセージボックスのOK ボタンをクリックすると、終了します。
17	GUUPD. AJN	ウィンドウにラベルとボタンを配置し、ボタンをクリックした際に、呼び出されるサブルーチン(BUTTON_EVENT)を登録します。 ボタンをクリックすると、GUUPD LABEL 命令で、ラベルの表示内容を更新します。 ウィンドウの閉じるボタンをクリックすると終了します。
18	BARChart. AJN	ウィンドウに棒グラフを描画します。 ウィンドウの閉じるボタンをクリックすることで終了します。
19	BARChart_FULL. AJN	ウィンドウに棒グラフとコンボボックスを配置して表示します。 コンボボックスを切り替えると、表示するグラフ設定を更新して再描画します。 棒グラフを再描画する際、縦軸の範囲や目盛りの設定など、様々な設定を行っています。 ウィンドウの閉じるボタンをクリックすると終了します。
20	LINEGRAPH. AJN	ウィンドウに波形グラフを配置して表示します。

#	ファイル名	内容
		波形グラフには、正弦、余弦のグラフを重ねて表示します。 ウィンドウの閉じるボタンをクリックすると終了します。
21	LINEGRAPH_AND_GUMAIN_ONELoop.AJN	ウィンドウに波形グラフを配置して表示します。 GUMAIN ONELoop 関数を使って、描画ループを回しつつ、逐次グラフを描画し続ける事で、あたかもグラフがアニメーションしているかのように見せます。 ウィンドウの閉じるボタンをクリックすると終了します。
22	LINEGRAPH_FULL.AJN	ウィンドウに波形グラフ、ボタン、スイッチを配置して表示します。 波形グラフには、正弦、余弦のグラフを重ねて表示します。 ボタンをクリックすると、グラフの表示範囲を切り替えて表示します。 スイッチをクリックすると、正弦のみのグラフ、正弦と余弦のグラフというように、切り替えて表示します。 ウィンドウの閉じるボタンをクリックすると終了します。
23	LINEGRAPH_PERFORMANCE.AJN	折れ線グラフの描画に掛かる時間のぶれを、簡単にパフォーマンス計測する事例です。 1200x800の画面サイズの折れ線グラフ (LINEGRAPH) に対して、凡そ、9ms 程度で、100件データ追加の描画繰り返しを行い、描画に掛かる時間が想定外 over (10ms 以上) したら、掛かった時間を表示します。
24	PIECHART.AJN	ウィンドウに円グラフを配置して表示します。 人口の男女比を円グラフにして表示します。 ウィンドウの閉じるボタンをクリックすると終了します。
25	PIECHART_FULL.AJN	ウィンドウに円グラフとコンボボックスを配置して表示します。 人口の男女比を、円グラフにして表示します。 (世界の統計と日本の統計を、コンボボックスで切り替え可能です) コンボボックスを切り替えると、表示するグラフの内容を切り替えて再描画します。 ウィンドウの閉じるボタンをクリックすると終了します。
26	GUDIALOGBOX_OPEN.AJN	ウィンドウにボタンとラベルを配置し、ボタンをクリックした際に、呼び出されるサブルーチン (BUTTON_EVENT) を登録します。 ボタンをクリックすると、ファイルを開くためのダイアログボックスを表示し、選択したファイルパスを取得して、ラベルに更新表示します。 ウィンドウの閉じるボタンをクリックすると終了します。
27	GUDIALOGBOX_SAVE.AJN	ウィンドウにボタンとラベルを配置し、ボタンをクリックした際に、呼び出されるサブルーチン (BUTTON_EVENT) を登録します。 ボタンをクリックすると、ファイルを保存するためのダイアログボックスを表示し、選択したファイルパスを取得して、ラベルに更新表示します。 ウィンドウの閉じるボタンをクリックすると終了します。
28	GUDIALOGBOX_DIRECTORY.AJN	ウィンドウにボタンとラベルを配置し、ボタンをクリックした際に、呼び出されるサブルーチン (BUTTON_EVENT) を登録します。 ボタンをクリックすると、ディレクトリを選択するためのダイアログボックスを表示し、選択したファイルパスを取得して、ラベルに更新表示します。 ウィンドウの閉じるボタンをクリックすると終了します。
29	GUG_SIMPLE.AJN	GUG 系のグラフィック描画コマンドを、何度も呼び出して描画される過程を見せるデモサンプルです。 ウィンドウにグラフィック描画領域を配置して表示します。 その後、グラフィック描画コマンドで、各種描画を行います。 GUGPSET 命令で点を描画します。GUGDRAWFILE 命令で画像を描画します。 GUGLINE 命令で線分を描画します。GUGRECTANGLE 命令で矩形を描画します。 GUGARC 命令で円弧を描画します。GUGCIRCLE 命令で円を描画します。 GUGDRAWTEXT で文字を描画します。 各命令を呼び出す都度、待ち時間を挿入して、描画過程が見られるようにしています。
30	VIDEO.AJN	動画を再生します。ボタンで再生、一時停止、停止、再生位置の制御がで

#	ファイル名	内容
		きます。
31	IMG_SIMPLE.AJN	IMG 系の画像処理コマンドを使用するデモサンプルです。 左に元画像を表示する領域、右に IMG 関数を呼び出した結果を出力する領域を用意します。 中央のラジオボタンのメニューを選択すると、IMG 関数を呼び出して、結果を右の領域に表示します。 左上のボタンで、変換元となる画像を、オープンダイアログから選択できます。

## 第5章 索引

<b>G</b>			
GUADD BARCHART DATALABEL	88	GUFONT	60
GUADD BARCHART RANGE	89	GUG2IMG_DATA	129
GUADD BARCHART TITLE	86	GUGARC	119
GUADD BARCHART VALUE	85	GUGCIRCLE	118
GUADD COMBOBOX	35	GUGCLS	115
GUADD LINEGRAPH DATALABEL	98	GUGCOLOR	116
GUADD LINEGRAPH GUAGE	102	GUGDATAGET	124
GUADD LINEGRAPH RANGE	100	GUGDATASET	125
GUADD LINEGRAPH TITLE	96	GUGDRAWBLIT	126
GUADD LINEGRAPH VALUE	94	GUGDRAWFILE	122
GUADD PIECHART DATALABEL	111	GUGDRAWTEXT	121
GUADD PIECHART TITLE	110	GUGET\$	57
GUADD PIECHART VALUE	108	GUGET\$ VIDEO	58
GUADD RADIOBTN	38	GUGLINE	116
GUCRATE BUTTON	27	GUGPSET	120
GUCREATE BARCHART	84	GUGRECTANGLE	117
GUCREATE CALENDAR	48	GUGSAVEFILE	123
GUCREATE CHECKBOX	39	GUMAIN LOOP	54
GUCREATE COMBOBOX	34	GUMAIN ONELOOP	54
GUCREATE IMAGE	41	GUMAIN QUIT	55
GUCREATE LABEL	29	GUMOUSE	63
GUCREATE LINEGRAPH	93	GUMOVE WINDOW	61
GUCREATE PIC	114	GUMSGBOX	62
GUCREATE PIECHART	107	GUNOTIFY SEND	74
GUCREATE RADIOBTN	37	GUSHOW	53
GUCREATE SCALE	44	GUSIZE	59
GUCREATE SWITCH	46	GUTIP	60
GUCREATE TEXTAREA	33	GUUPD BARCHART DATALABEL	89
GUCREATE TEXTBOX	31	GUUPD BARCHART DRAW	91
GUCREATE VIDEO	50	GUUPD BARCHART GRID	91
GUCREATE WINDOW	26	GUUPD BARCHART LABEL	87
GUDELETE	61	GUUPD BARCHART RANGE	90
GUDIALOGBOX DIRECTORY\$	72	GUUPD BARCHART VALUE	85
GUDIALOGBOX OPEN\$	68	GUUPD BUTTON	28
GUDIALOGBOX SAVE\$	70	GUUPD CALENDAR	49
GUDSP	56	GUUPD CHECKBOX	40
GUDSP BARCHART	92	GUUPD IMAGE	43
GUDSP LINEGRAPH	106	GUUPD IMAGE FIXED	43
GUDSP PIECHART	112	GUUPD IMAGE ROTATE	43
GUEVT OFF	65	GUUPD LABEL	30
GUEVT ON	64	GUUPD LINEGRAPH DATALABEL	99
		GUUPD LINEGRAPH DRAW	105
		GUUPD LINEGRAPH GRID	104

GUUPD LINEGRAPH GUAGE	103	IMG_ERODE	136
GUUPD LINEGRAPH LABEL	97	IMG_FIND_CONTOURS	134
GUUPD LINEGRAPH RANGE	101	IMG_GAUSSIAN_FILTER	139
GUUPD LINEGRAPH VALUE	95	IMG_GET_ROTATION_MATRIX	145
GUUPD PIECHART DATALABEL	111	IMG_LAPLACIAN_FILTER	142
GUUPD PIECHART DRAW	112	IMG_MATCH_TEMPLATE	147
GUUPD PIECHART VALUE	109	IMG_MEAN_FILTER	138
GUUPD RADIOBTN	39	IMG_PHASE_CORRELATION	149
GUUPD SCALE	45	IMG_READ	127
GUUPD SWITCH	47	IMG_RESIZE	131
GUUPD TEXTAREA	34	IMG_SHARP_FILTER	143
GUUPD TEXTBOX	32	IMG_SOBEL_FILTER	141
GUUPD VIDEO	51	IMG_WARP_AFFINE	146
<i>I</i>		IMG_WARP_POLAR	150
		IMG_WRITE	128
		IMG2GUG_DATA	129
IMG_BILATERAL_FILTER	140	<i>O</i>	
IMG_BINARIZE	132		
IMG_BINARIZE_AUTO	133		
IMG_CANNY_EDGE	144	ON GUEVT CALL	66
IMG_CONVERT_COLOR	130	OPTION GUI	25
IMG_DILATE	137		

## 第6章 重要な情報

---

### 保証の内容と制限

弊社は本ドキュメントに含まれるソースプログラムの実行が中断しないこと、またはその実行に誤りが無いことを保証していません。

本製品の品質や使用に起因する、性能に起因するいかなるリスクも使用者が負うものとします。

弊社はドキュメント内の情報の正確さに万全を期しています。万一、誤記または誤植などがあつた場合、弊社は予告無く改訂する場合があります。ドキュメントまたはドキュメント内の情報に起因するいかなる損害に対しても弊社は責任を負いません。

ドキュメント内の図や表は説明のためであり、ユーザ個別の応用事例により変化する場合があります。

### 著作権、知的所有権

弊社は本製品に含まれるおよび本製品に対する権利や知的所有権を保持しています。

本製品はコンピュータ ソフトウェア、映像/音声(例えば図、文章、写真など)を含んでいます。

### 医療機器/器具への適用における注意

弊社の製品は人命に関わるような状況下で使用される機器に用いられる事を目的として設計、製造された物では有りません。

弊社の製品は人体の検査などに使用するに適する信頼性を確保する事を意図された部品や検査機器と共に設計された物では有りません。

医療機器、治療器具などの本製品の適用により、製品の故障、ユーザ、設計者の過失などにより、損傷/損害を引き起こす場合が有ります。

### 複製の禁止

弊社の許可なく、本ドキュメントの全て、または一部に関わらず、複製、改変などを行うことはできません。

### 責任の制限

弊社は、弊社または再販売者の予見の有無にかかわらず発生したいかなる特別損害、偶発的損害、間接的な損害、重大な損害について、責任を負いません。

本製品(ハードウェア、ソフトウェア)のシステム組み込み、使用、ならびに本製品から得られる結果に関する一切のリスクについては、本製品の使用者に帰属するものとします。

本製品に含まれる不都合、あるいは本製品の供給(納期遅延)、性能もしくは使用に起因する付帯的損害もしくは間接的損害に対して、弊社に全面的に責がある場合でも、弊社はその製品に対する改良(有償サービスの利用)、代品交換までとし、製品の予防交換並びに、代金減額等、金銭面での賠償の責任は負わないものとします。

本製品は、日本国内仕様です。

### 商標/登録商標

本書に掲載されている会社名、製品名は、それぞれ各社の商標または登録商標です。

## 改訂履歴

Ver.	年 月	改 訂 内 容
0.90	2019年10月	新規作成
1.00	2022年1月	誤記修正。 コマンド追加。GUNOTIFY SEND, GUCREATE VIDEO, GUUPD VIDEO, GUGET\$ VIDEO サンプル追加。VIDEO.AJN
1.10	2023年3月	画像処理コマンドの追加。 コマンド追加。GUUPD CALENDAR

このマニュアルは、製品の改良その他により将来予告なく改訂しますので、予めご了承ください。