GUIオブジェクト

- 画面に情報を表示したり、プログラムを対話的に操作するためのオブジェクトです。
- プログラムでは、大きさと位置でGUIオブジェクトの大きさと位置を指定して使ってください。
- 位置を指定しない場合には、直前に作ったGUIオブジェクトの右隣に配置され、画面の右端を越えたり次の行を実行することで下の行の左端に移動します。ただし、画面の大きさは実行する環境によって異なりますので、できるだけ「大きさ」と「位置」を指定してください。
- GUIオブジェクトの背景色（塗る）はMacでは対応していません。
- GUIオブジェクトを参照するとオブジェクト内の文字列が返ります。テキストエリアとリストは行が空白で連結されます。選択メニューは選択されている要素が返ります。

- ラベル、ボタン、選択メニューの文字列中にHTMLを記述できます（Java版のみ）
  - 記述できるHTMLは、W3CのHTML3.2相当です。ただしaタグによるリンクなど、一部の機能は利用できません。HTMLの意味や文法については、関連する書籍などを参照してください。
  - 文字列の先頭は“<html>”で始める必要があります。“”による引用は記述できません。<font color=“red”>は、<font color=red>と記述します。
  - 利用できる主なタグを示します。
    - 本文(body)、見出し(h1-h6)、段落(p, br, hr)、箇条書き(ul, ol, dl, lib)、表組(table, tr, th, td)、文字(b, font)、画像(img)
  - 利用例は付属のサンプルプログラム(html.dtl)をご覧ください。
  - HTMLを用いた画像は、タートルや図形オブジェクトと重なっても衝突を起こしません。そのため、背景画像の表示に適しています。imgタグのほか、bodyタグのbackground属性で繰り返し表示することも可能です。
  - （例） ローカルのlarge.jpgという画像を表示します。

```
ラベル＝"<html><img src=file:large.jpg></html>" 作る。
```

- （例） 文字の背景に画像を表示します。

```
{ラベル＝"<html><body
background=http://dolittle.eplang.jp/image/pukiwiki.png>あいうえおかきくけこさしすせそ<br><br><br><br><br></body></html>" 作る。
```

GUIオブジェクトの共通の説明

- 以下に、GUIオブジェクトに共通の命令を示します。「ボタン1」という名前のボタンオブジェクトの例を示しています。次の1行に続けて書いて実行してください。

```
ボタン1＝ボタン＝"ABC" 作る。
```

- **次の行**: 直前に作成したGUIオブジェクトの次の行に配置します。
  - （例）オブジェクトを直前のGUIオブジェクトの次の行に表示ます。

```
ボタン1！次の行。
```

- **位置**: 表示位置を指定します。
  - （例）オブジェクトを画面の「（100, 100）」の座標の位置に動かします。

```
ボタン1！100　100　位置。
```

- **移動する**: 右にx歩、上にy歩動きます。
  - （例）**オブジェクト**を画面で「右に0、上に100」だけ動かします。

```
ボタン1！0　100　移動する。
```

- **大きさ**: 大きさを指定します。
  - （例）**オブジェクト**を「横100、縦50」の大きさにします。

```
ボタン1！100　50　大きさ。
```

- **幅？**: 横幅を調べます。
  - （例）**オブジェクト**の横幅を表示します。

```
ラベル！（ボタン1！幅？）作る。
```

- **高さ？**: 高さを調べます。
  - （例）**オブジェクト**の高さを表示します。

```
ラベル！（ボタン1！高さ？）作る。
```

- **文字サイズ**: 表示する文字の大きさを指定します。標準は24ポイントです。リストとスライダー以外のGUIオブジェクトに共通です。
  - （例）**オブジェクト**に表示する文字サイズを「16」に設定します。

```
ボタン1！16　文字サイズ。
```

- **塗る**: 色を色オブジェクトか三原色で指定します。リスト以外のGUIオブジェクトに共通です。
  - （例）**オブジェクト**の色を設定します。

```
ボタン1！（水）塗る。
```

- （例）**オブジェクト**の色を設定します。

```
ボタン1！255　128　255　塗る。
```

- （例）**オブジェクト**の色を設定します。

```
ボタン1□"#FF88FF"□塗る。
```

- （例）**オブジェクト**の色を設定します。

```
ボタン1□"#F8F"□塗る。
```

- **文字色**: 文字の色を色オブジェクトか三原色で指定します。リスト以外のGUIオブジェクトに共通です。
  - （例）**オブジェクト**の文字色を設定します。

```
ボタン1！（緑）文字色。
```

- （例）**オブジェクト**の文字色を設定します。

```
ボタン1！0　128　0　文字色。
```

- （例）**オブジェクト**の文字色を設定します。

```
ボタン1！"#008800"　文字色。
```

- （例）オブジェクトの文字色を設定します。

```
ボタン1！"#080"　文字色。
```

- **消える**：画面から消えます。
  - （例）オブジェクトを画面から消します。

```
ボタン1！消える。
```

- **現れる**：画面に現れます。消えた姿を戻すときに使います□□Bit Arrow版のみ）
  - （例）消えているオブジェクトを画面に表示します。「消える」の例に続いて実行してください。

```
ボタン1！現れる。
```

ボタン

- 画面に表示されるGUI部品です。初期サイズは150×45です。
- GUIオブジェクトに共通の説明は「GUIオブジェクトの共通の説明」を参照してください。
- ボタンが押されると**動作**というメソッドが実行されます。
  - （例）ボタン1＝ボタン！”挨拶”作る。

```
ボタン1：動作＝「ラベル！"こんにちは" 作る」。
```

- ボタンの生成時に**ショートカットキー**を指定することができます。
- 以下の「作る」以外の例では、「ボタン1」を作ってから実行してください。

- **作る**：新しいボタンを作ります。1個目のパラメータにはボタンに表示するラベルを指定します。
  - （例）「実行」と書かれたボタンを作ります。

```
ボタン1＝ボタン！"実行"　作る。
ボタン1：動作＝「ラベル！"こんにちは" 作る」。
```

2個目のパラメータにショートカットキーを指定できます。キーの文字列は**sec□keylist**「ショートカットキー一覧」を参照してください。次の例では上向き矢印キーを押すとボタンの動作が実行されます。

- （例）上向き矢印キーを押すと実行されるボタンを作ります。

```
ボタン1＝ボタン！"実行"□"UP"□作る。
ボタン1：動作＝「ラベル！"こんにちは" 作る」。
```

- **読む**：文字を読んで返します。
  - （例）ボタンの文字列を表示します。

```
ラベル！（ボタン1！読む）作る。
```

- **書く**：文字を書きます。
  - （例）ボタンに「あいうえお」という文字を書きます。

```
ボタン1！"あいうえお"　書く。
```

- **増やす**：表示されている数を増やします。増やす数を省略すると1だけ増えます。
  - （例）ボタンに表示されている値を1増やします。

ボタン1！　増やす。

- （例）**ボタン**に表示**されている**値を**10**増やします。

ボタン1！10　増やす。

- **減らす**: 表示**されている**数を減らします。減らす数を省略すると**1**だけ減ります。
  - （例）**ボタン**に表示**されている**値を**1**減らします。

ボタン1！　減らす。

- （例）**ボタン**に表示**されている**値を**10**減らします。

ボタン1！10　減らす。

フィールド

- 画面に表示**される**GUI部品**です**。文字の表示や入力に使**います**。初期**サイズは150×45です**。
- GUI**オブジェクト**に共通の説明**は**「GUI**オブジェクト**の共通の説明」を参照**してください**。
- **リターンキー** が押されると**動作**というメソッドが実行され、**フィールド**の値がパラメータとして渡されます(Java版のみ)
- 以下の「作る」以外の例**では**、「**フィールド1**」を作ってから実行**してください**。

- **作る**: 新**しいフィールド**を作ります。
  - （例）**フィールド**を作ります。**パラメータ**に初期値を指定することもできます。

フィールド1＝フィールド！作る。

- **読む**: 文字を読んで返します。
  - （例）**フィールド**の文字を表示します。

ラベル！（フィールド1！読む）作る。

- **書く**: 文字を書きます。
  - （例）**フィールド**に「**あいうえお**」という文字を書きます。

フィールド1！"あいうえお"　書く。

- **クリア**: 空にします。
  - （例）**フィールド**の文字を消します。

フィールド1！クリア。

- **増やす**: 表示**されている**数を増やします。増やす数を省略すると**1**だけ増えます。
  - （例）**フィールド**に表示**されている**値を**1**増やします。

フィールド1！増やす。

- （例）**フィールド**に表示**されている**値を**10**増やします。

フィールド1！10　増やす。

- **減らす**: 表示**されている**数を減らします。減らす数を省略すると**1**だけ減ります。
  - （例）**フィールド**に表示**されている**値を**1**減らします。

```
フィールド1！減らす。
```

- （例）フィールドに表示されている値を10減らします。

```
フィールド1！10　減らす。
```

- 以下はJava版のみ
  - **フォーカス**：フィールドが画面で選択された状態にします。
  - （例）フィールドを画面でフォーカスします。

```
フィールド1！フォーカス。
```

ラベル

- 画面に表示されるGUI部品です。文字の表示に使います。中に1つの文字列を入れられます。
- GUIオブジェクトに共通の説明は「GUIオブジェクトの共通の説明」を参照してください。
- 大きさは、表示する文字列によって自動的に設定されます。
- 以下の「作る」以外の例では、「ラベル1」を作ってから実行してください。

- **作る**：新しいラベルを作ります。
  - （例）「あいうえお」と書かれたラベルを作ります。

```
ラベル1＝ラベル！"あいうえお"　作る。
```

- **書く**：文字を書きます。
  - （例）ラベルに「かきくけこ」という文字を書きます。

```
ラベル1！"かきくけこ"　書く。
```

- **増やす**：表示されている数を増やします。増やす数を省略すると1だけ増えます。
  - （例）ラベルに表示されている値を1増やします。

```
ラベル1！　増やす。
```

- （例）ラベルに表示されている値を10増やします。

```
ラベル1！10　増やす。
```

- **減らす**：表示されている数を減らします。減らす数を省略すると1だけ減ります。
  - （例）ラベルに表示されている値を1減らします。

```
ラベル1！　減らす。
```

- （例）ラベルに表示されている値を10減らします。

```
ラベル1！10　減らす。
```

- **変身する**：ラベルに文字の代りに画像を表示します。
  - 画像を利用する場合は、画像ファイル（png, jpg, gifのいずれか）をドリトル本体（dolittle.jar）と同じディレクトリに置いてください。
    - 上記以外のフォルダにある画像を表示したい場合は、画像へのパスを指定してください。
  - （例）ラベルに「99.png」という名前の画像を表示します。

ラベル1□"99.png"□変身する。

選択メニュ□□Java版のみ）

- 画面に表示されるGUI部品です。選択肢を表示し、そこから選択できます。初期サイズは150×45です。
- GUIオブジェクトに共通の説明は「GUIオブジェクトの共通の説明」を参照してください。
- 選択肢が選ばれると**動作**というメソッドが実行され、選ばれた選択肢の文字列と何番目が選ばれたかを表す番号がパラメータとして渡されます。
- 以下の「作る」以外の例では、「メニュー1」を作ってから実行してください。

- **作る**：新しい選択メニューを作ります。
  - （例）「+」と「-」という選択肢で選択メニューを作ります。選択肢を選ぶと、その文字列が表示されます。

```
メニュー1＝選択メニュー！"＋" "－" 作る。
ラベル1＝ラベル！作る。
メニュー1：動作□□□x| ラベル1□□x□書く」。
```

- **書く**：メニューに文字列を書きます。新しい選択肢として追加されます。
  - （例）選択メニューに「+」と「-」という選択肢を追加します。

```
メニュー1！"＋" "－" 書く。
```

- **何番目？**：選ばれている選択肢の番号を1から始まる整数で返します。「書く」の例に続けて実行してください。
  - （例）選択メニューの1個目の選択肢が選ばれたときに、「かめた！100 歩く」を実行します。

```
「（メニュー1！何番目？） ==1」！なら「かめた！100　歩く」実行。
```

- **読む**：メニューの文字列を読みます。メニューの番号を1から始まる整数で指定します。
  - （例）メニューの1個目の選択肢の文字を読みます。「書く」の例に続けて実行してください。

```
ラベル！（メニュー1！1　読む）作る。
```

スライダ□□Java版のみ）

- 画面に表示されるGUI部品です。初期サイズは300×45です。
- GUIオブジェクトに共通の説明は「GUIオブジェクトの共通の説明」を参照してください。
- つまみを動かして0～100の範囲で値を変えられます。値が変わると**動作**というメソッドが実行され、スライダーの値がパラメータとして渡されます。
- スライダーの生成時に**ショートカットキー**を指定することができます。
- 以下の「作る」以外の例では、「スライダー1」を作ってから実行してください。

- **作る**：新しいスライダーを作ります。
  - （例）スライダーを作ります。

```
スライダー1＝スライダー！作る。
```

- （例）初期値を80に設定しながら、スライダーを作ります。

```
スライダー2＝スライダー！80 作る。
```

値を減少/増加させるショートカットキーを指定できます。キーの文字列はsec□keylist 「ショートカットキー一覧」を参照してください。次の例では左右の矢印キーを押すとスライダーのバーが動きます。

- （例）左右の矢印キーで操作できるスライダーを作ります。値が画面に表示されます。

```
スライダー1＝スライダ□□"LEFT"□"RIGHT"□作る。
ラベル1＝ラベル！作る。
スライダー1：動作□□□x|　ラベル1□□x□書く」。
```

- **値？**: スライダーの値を得ます。0～100の値が返ります。
  - （例）スライダーの値を表示します。

```
ラベル1＝ラベル！作る。
スライダー1：動作＝「ラベル1！（スライダー1！値？）書く」。
```

- **値**: スライダーに値を設定します。0～100の値を指定します。
  - （例）スライダーに「50」を設定します。

```
スライダー1！50　値。
```

- **横向き**: 横長のスライダーにします。
  - （例）スライダーを横向きにします。

```
スライダー1！横向き。
```

- **縦向き**: 縦長のスライダーにします。
  - （例）スライダーを縦向きにします。

```
スライダー1！縦向き。
```

- **文字出す**: 目盛ラベルを表示します。10ごとの目盛ラベルが表示されます。
  - （例）スライダーに目盛ラベルを表示します。

```
スライダー1！文字出す。
```

- **文字消す**: 目盛ラベルを表示しません。
  - （例）スライダーから目盛ラベルを消します。

```
スライダー1！文字消す。
```

- **から**: 値の範囲を設定します。
- **まで**: 値の範囲を設定します。
  - （例）スライダーの値の範囲を-50から50に設定します。

```
スライダー1！-50　から　50　まで。
```

テキストエリア

- 画面に表示されるGUI部品です。複数行の文字の表示や入力に使います。
- Java版

    - GUIオブジェクトに共通の説明は「GUIオブジェクトの共通の説明」を参照してください。
    - 初期サイズは180×135です。
- Bit Arrow版
    - 初期サイズは3文字×2行です。「書く」などで文字が追加されると大きさが変化します。ただし、「行数」または「列数」が指定された場合は大きさは変化しません。
    - 大きさが変化すると、隣り合った他のGUI部品の位置も変化します。ただし、「位置」または「移動する」が指定されたGUI部品の位置は変化しません。
- 以下の「作る」以外の例では、「エリア1」を作ってから実行してください。

- **作る**: 新しいテキストエリアを作ります。引数は1行ずつ追加されます。配列を指定すると要素が各行に追加されます。
    - （例）テキストエリアを作ります。

```
エリア1＝テキストエリア！作る。
```

- （例）テキストエリアを作ります□□abc□が1行目に、□def□が2行目に追加されます。

```
エリア1＝テキストエリア□"abc"□"def"□作る。
```

- （例）テキストエリアを作ります。配列１の要素が各行に追加されます。

```
エリア1＝テキストエリア！（配列１）作る。
```

- **書く**: 文字を書きます。最後の文字の後に追加されます。
    - （例）テキストエリアに「“ あいうえお ”」という文字列を追加します。

```
エリア1！"あいうえお"　書く。
```

- **改行**: 改行します。
    - （例）テキストエリアに「“ あいうえお ”」という文字列を追加して改行します。

```
エリア1！"あいうえお"　書く　改行。
```

- **読む**: テキストエリアの内容を文字列で返します。
    - （例）テキストエリアの内容を1行で表示します。

```
ラベル！（エリア1！読む）作る。
```

- **上書き**: テキストエリアの内容を削除して新しい内容で書き換えます。
    - （例）テキストエリアの内容を新しい内容で書き換えます。

```
エリア1□"ABC" "DEF" 上書き。
```

- **クリア**: 空にします。
    - （例）テキストエリアのすべての要素を削除します。

```
エリア1！クリア。
```

- **読むだけ**: 画面から編集できないようにします。
    - （例）テキストエリアを画面から書けないようにします。

```
エリア1！読むだけ。
```

- **読み書き**: 画面から編集できるようにします。
    - （例）テキストエリアを画面から書けるようにします。

```
エリア1！読み書き。
```

- **自動改行**:「書く」の後で自動的に改行するようにします。（いいえ）を指定すると自動改行しないようにします。
  - （例）テキストエリアを自動的に改行するようにします。

```
エリア1！自動改行。
```

- （例）テキストエリアを自動的に改行しないようにします。

```
エリア1！（いいえ）自動改行。
```

- Bit Arrow版
  - **行数**: テキストエリアの縦の行数を指定します。内容が行数を超えた場合はスクロールバーが表示されます。

```
エリア1！3　行数。
```

- **列数**: テキストエリアの横の列数を指定します。内容が列数を超えた場合は次の行に続けて表示されます。

```
エリア1！10　列数。
```

リスト

- 画面に表示されるGUI部品です。初期サイズは150×90です。
- GUIオブジェクトに共通の説明は「GUIオブジェクトの共通の説明」を参照してください。
- 次の点を除いて、テキストエリアと機能は同じです。
  - 表示専用です。
  - 「書く」で自動的に改行されます。
  - 背景色は白です。
- 以下の「作る」以外の例では、「リスト1」を作ってから実行してください。

- **作る**: 新しいリストを作ります。
  - （例）リストを作ります。

```
リスト1＝リスト！作る。
```

- **書く**: 文字を書きます。新しい行として追加されます。
  - （例）リストに「“ あいうえお ”」という文字列を追加します。

```
リスト1！"あいうえお"　書く。
```

- **読む**: 文字を読んで返します。行を1から始まる整数で指定します。行を省略した場合や存在しない行を指定した場合は要素が連結された文字列が返ります。
  - （例）リストの1個目の要素を読みます。

```
ラベル！（リスト1！1　読む）作る。
```

- **クリア**: 空にします。
  - （例）リストのすべての要素を削除します。

```
リスト1！クリア。
```

ショートカットキー一覧

- ボタンやスライダーなどのGUI部品をキーボードから操作するための**ショートカットキー**の一覧です。
- これらの文字は、各種のキーボードで共通に使えます。

意味	記号
英字	“A”, “B”, …, “Z”
数字	“1”, “2”, …, “0”
ファンクション	“F1”, “F2”, …, “F12”
エスケープ（ESC）	“ESCAPE”
マイナス（-）	“MINUS”
バックスラッシュ（¥）	“BACK_SLASH”
開き括弧（[）	“OPEN_BRACKET”
閉じ括弧（]）	“CLOSE_BRACKET”
セミコロン（;）	“SEMICOLON”
コンマ（,）	“COMMA”
ピリオド（.）	“PERIOD”
スラッシュ（/）	“SLASH”
エンタ（ENTER）	“ENTER”
ホーム（HOME）	“HOME”
エンド（END）	“END”
ページアップ（PageUp）	“PAGE_UP”
ページダウン（PageDown）	“PAGE_DOWN”
カーソル上（↑）	“UP”
カーソル下（↓）	“DOWN”
カーソル左（←）	“LEFT”
カーソル右（→）	“RIGHT”

From:
https://dolittle.eplang.jp/ - **プログラミング言語「ドリトル」**

Permanent link:
https://dolittle.eplang.jp/ref_gui331?rev=1580447374

Last update: **2020/01/31 14:09**