# GTK基本情報

## 基本情報

総合デスクトップ環境GNOME用のツールキット等として広く利用。（例えばUbuntuはGNOME）

最初はLinuxでの画像編集ソフトGIMPを実装する為に開発されたウィジェットだった為，特にLinuxとの関係が強い。

以前はGTK+と呼ばれたが，2019年にGTKに改名された。

元々がC言語で，様々な言語版がある。

C++ 　(gtkmm) ，C#　(Gtk#)，Perl 　(gtk2-perl) ，Python　(PyGTK) ，Java　(Java-GNOME)

他にもJavaScript、Vala。非公式にFortran (gtk-fortran)、Ruby (Ruby/Gtk2) ，PHP (PHP-GTK)、Pascal、Lua、Haskell、FreeBASIC

GTKは以下のコンポーネントからなる

GLib GTKの移植性を高めるための関数をまとめたライブラリ

GDK ウィンドウシステムに対する軽量なWrapper。ウィンドウの生成や描画機能を持つ

GTK GDKの上に構築されるウィジェット（ボタンやスクロールバーなど個々のGUI部品）のセット

実行環境の確認

gcc -dumpversion # gccのバージョン確認

g++ -dumpversion # g++ のバージョン確認

## インストール

Linuxのディストリビューションによって異なるとの事

Ubuntu：

sudo apt install libgtk-3-dev

CentOS：

yum install gtk2 # 最初から入っている可能性が高い　rpm -q gtk2で確認できる。

yum install gtk2-devel

## 注意点

日本語を表示するためにはソースファイルの文字コードを UTF-8 としておく必要がある。

## ツール

Gtkはクセがあり，習得まで時間がかかるらしい。Gladeなどのツールを用いる方が一般的らしい。

但し，本資料はあくまでgtkの内容のみを書き留める事にする。

### Glade

sudo apt install glade

## 資料

GTK入門：<http://home.a00.itscom.net/hatada/cpp/gtk01.html>

WikiBooks：<https://ja.wikibooks.org/wiki/GTKプログラミング>

# gtk/gtk.h

## 標準関数など

公式情報：

<https://developer.gnome.org/gtk3/stable/gtk3-General.html>

### 関数

#### gtk\_init

構文：void gtk\_init (int \*argc, char \*\*\*argv);

GTK+の機能を使用するまえに実行する初期化関数。

int main(int argc, char \*argv[])

{

gtk\_init(&argc,&argv); # コマンドライン引数をそのまま渡すらしい。

// 本処理

}

#### gtk\_main\_quit

構文：void gtk\_main\_quit (void);

メインループの中核部での実行returnの呼び出し

#### gtk\_main

構文：void gtk\_main (void);

gtk\_main\_quitが呼び出されるまでメインループを実行する。

gtk\_mainは入れ子呼び出しが可能。（？？？）

### 関数

#### gtk\_widget\_show

構文：gtk\_widget\_show (GtkWidget \*widget);

gtk\_widget\_show\_all()

gtk\_main()

### マクロ

#### g\_signal\_connect

定義：#define g\_signal\_connect(instance, detailed\_signal, c\_handler, data)

### オブジェクトマクロ（？）

型変換専用のマクロだろうか，と思われるものがいくつかある。

#### GTK\_WIDGET

Widgetオブジェクト

label1 = GTK\_WIDGET(gtk\_builder\_get\_object(builder, “label1”));

#### G\_CALLBACK

## GtkBuilder

XML UL定義からインターフェイスを構築する。

### 関数

#### gtk\_builder\_connect\_signals

構文：void gtk\_builder\_connect\_signals (GtkBuilder \*builder, gpointer user\_data);

明示的にBuilderオブジェクトに追加されたシンボルを追加する。

gtk\_builder\_connect\_signals\_full関数の簡易版。

gtk\_builder\_connect\_signals(builder, NULL);

#### gtk\_builder\_get\_object

構文： GObject \* gtk\_builder\_get\_object (GtkBuilder \*builder, const gchar \*name);

name で指定したIDを持つオブジェクトを返す。

GtkWidget \*label1;

label1 = GTK\_WIDGET(gtk\_builder\_get\_object(builder, "label1"));

#### gtk\_builder\_get\_objects

構文：GSList \* gtk\_builder\_get\_objects (GtkBuilder \*builder);

#### gtk\_builder\_new

## GtkWidget

≒コントロール。

### GtkWidget クラス

#### 関数

##### gtk\_builder\_new\_from\_file

XMLファイルからGtkBuilder UI定義を読み込む

構文：gtk\_builder\_new\_from\_file (const gchar \*filename);

GtkBuilder \*builder;

builder = gtk\_builder\_new\_from\_file("*glade\_xml\_file.glade*");

### GtkLabel クラス

## GtkWindow

継承関係：GObject：GInitiallyUnowned：GtkWidget：GtkContainer：GtkBin：GtkWindow

<https://developer.gnome.org/gtk3/unstable/GtkWindow.html#GtkWindowType>

### 関数

#### gtk\_window\_new

構文：GtkWidget\* gtk\_window\_new(GtkWindowType type)

（例）

GtkWidget \*wnd; // ウィジェットを格納するためのポインタ変数を宣言。

gtk\_init(&argc, &argv); // GTKの初期化

wnd = gtk\_window\_new(GTK\_WINDOW\_TOPLEVEL); // ウィンドウを作成

#### gtk\_window\_set\_title

GtkWindowのタイトルを設定する。

構文：void gtk\_window\_set\_title (GtkWindow \*window, const gchar \*title);

### 列挙

#### GtkWindowType

GTK\_WINDOW\_TOPLEVEL ダイアログのような通常のウィンドウ

GTK\_WINDOW\_POPUP ツールチップのような特別なウィンドウ

## その他のクラス

### GError クラス

### GCallback クラス

## 参考資料

GNOME Developer <https://developer.gnome.org/gtk3/>

メモ

終了の仕方

gtk\_widget\_destroy(lookup\_widget(GTK\_WIDGET(button),"window1"));

gtk\_main\_quit();

どちらでも良いらしい