**ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH**

**TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN**

**KHOA KHOA HỌC VÀ KĨ THUẬT THÔNG TIN**



**BÁO CÁO ĐỒ ÁN**

**KỸ THUẬT LẬP TRÌNH PYTHON**

**Đề tài:**

**TÌM HIỂU, SO SÁNH CÁC LIB CHUYÊN VỀ GUI: TKINTER, PYGOBJECT, PYQT, PYSIDE, KIVY, WXPYTHON**

GVHD: ThS. Nguyễn Thanh Sơn

Sinh viên thực hiện: Nguyễn Thị Ngọc Đang MSSV: 20520007

Tp. Hồ Chí Minh, 12/2022

**NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN**

*……., ngày……...tháng……năm 2023*

**Người nhận xét**

*(Ký tên và ghi rõ họ tên****)***

**LỜI MỞ ĐẦU**

Sau đây, tôi sẽ trình bày về khái niệm, so sánh các lib chuyên về GUI: Tkinter, Pygobject, Pyqt, Pyside, Kivy, WxPython như sau:

* Chương 1: Tìm hiểu về GUI và GUI trong Python
* Chương 2: Tìm hiểu GUI trong Python về: tkInter, PyGObject, PyQt, PySide, Kivy, wxPython
* Chương 3: So sánh GUI trong Python về: tkInter, PyGObject, PyQt, PySide, Kivy, wxPython
* Chương 4: Kết luận

**MỤC LỤC**

[LỜI MỞ ĐẦU 3](#_Toc123916091)

[Chương 1: KHÁI NIỆM VỀ GUI, GUI trong PYTHON 5](#_Toc123916092)

[1.1 GUI 5](#_Toc123916093)

[là 6 **Python GUI Framework** để người dùng Python có thể làm app giao diện một cách hiệu quả và nhanh chóng nhất 9](#_Toc123916094)

[Chương 2: 5](#_Toc123916092)

[Chương 3: 5](#_Toc123916092)

[Chương 4: KẾT LUẬN 11](#_Toc123916095)

[2.1 Ưu điểm 11](#_Toc123916096)

[2.2 Nhược điểm 11](#_Toc123916097)

[2.3 Hướng phát triển 11](#_Toc123916098)

[TÀI LIỆU THAM KHẢO 11](#_Toc123916099)

# KHÁI NIỆM VỀ GUI, GUI PYTHON

## GUI

**KHÁI NIỆM**

**GUI**

GUI viết tắt là Graphical User Interface – Giao diện đồ họa người dùng. Giao diện này được phát triển bởi Alan Kay, Douglas Engelbart và một số nhà nghiên cứu khác vào tại Xerox PARC. Hiện nay gần như các hệ điều hành máy tính đều sử dụng giao diện đồ họa này.

GUI giúp người dùng dễ dàng hơn trong quá trình giao tiếp với thiết bị điện tử thông minh, bằng việc sử dụng hình ảnh hoặc chữ viết thay vì mất công gõ các đoạn văn bản dài dòng.

GUI đang được ứng dụng cực kỳ phổ biến, từ các ứng dụng online, chương trình máy tính, hệ điều hành, ….

GUI được tạo bởi 2 thành phần chính là Thành phần cấu trúc và Thành phần tương tác.

Thành phần cầu trúc:

Windows: là nơi người dùng có thể tương tác với máy tính bằng cách click vào các biểu tượng. Nó có thể được hiển thị, ẩn đi hoặc di chuyển đến các vị trí bất kỳ theo mục đích của người sử dụng. Một số loại cửa sổ làm việc thường gặp như:

* Cửa sổ trình duyệt: cửa sổ trình duyệt Web, ,,,,
* Cứa sổ con: cửa sổ bật lên trên Internex

**GUI PYTHON**

**Python GUI** (Python Graphic User Interface) hay còn gọi là Giao diện đồ hoạ vốn từng là một phần không thể thiếu khi nói về lập trình. Dù học ngôn ngữ lập trình nào, người ta cũng nghĩ tới chuyện "làm sao để có giao diện đồ hoạ".

Tuy nhiên khi thời đại tất cả mọi thứ đều chuyển lên web, thì web/app mobile trở thành giao diện để tương tác với người dùng, chứ không phải các phần mềm có giao diện chạy trên máy tính như trước kia.

Giờ đây mọi người đều có thể nghe nhạc trên web, xem film trên web, chơi game trên web, soạn thảo văn bản trên web… khó còn ứng dụng nào không đưa lên web nữa. Vậy nên về mặt "sự nghiệp", có vẻ như bạn nên đầu tư vào kỹ năng làm web thay vì học để tạo một app trên desktop như cách đây chục năm.

Hiện có 6 thư viện tốt nhất trong Python GUI là:

1. Thư viện Tkinter
2. Thư viện Kivy
3. Thư viện PyGObject
4. Thư viện PyQt
5. Thư viện PySide
6. Thư viện wxPython

#### **Thư viện Tkinter**

Tkinter thường được đóng gói với Python, và nó là **Python GUI** Framework tiêu chuẩn của Python. Nó phổ biến vì sự đơn giản và giao diện người dùng đồ họa, mã nguồn mở và có sẵn theo Python License.

Một trong những ưu điểm của việc chọn Tkinter là vì nó được cung cấp theo Python mặc định, nên có rất nhiều tài nguyên, cả code và sách tham khảo dành cho người mới.

 Ngoài ra, với cộng đồng lâu đời và năng động, có nhiều người có thể sẵn sàng giúp bạn trong trường hợp bạn mới bắt đầu học, rất nhiều lỗi bạn có thể tìm ra cách sửa chữa ngay lập tức.  
Link tải framework [Tkinter](https://docs.python.org/3/library/tk.html)

**Thư viện Kivy**

 Kivy là một Python GUI Framework tăng tốc OpenGL ES 2 để tạo giao diện người dùng mới. Kivy chạy trên Linux, Windows, OS X, Android, iOS và Raspberry Pi. Bạn có thể chạy cùng một mã nguồn trên tất cả các nền tảng được hỗ trợ.

 Nó có thể sử dụng nguyên bản hầu hết các đầu vào, giao thức và thiết bị bao gồm WM\_Touch, WM\_Pen, Mac OS X Trackpad và Magic Mouse, Mtdev, Linux Kernel HID, TUIO. Một mô phỏng chuột cảm ứng đa điểm.  
Kivy được sử dụng miễn phí 100%, theo giấy phép MIT (bắt đầu từ 1.7.2) và LGPL 3 cho các phiên bản trước. Bộ công cụ được phát triển, hỗ trợ chuyên nghiệp. Bạn có thể sử dụng nó trong một sản phẩm thương mại.  
Framework Kivy ổn định và có tài liệu API tốt, cùng với hướng dẫn lập trình kèm theo để giúp bạn bắt đầu nhanh chóng.  
Link tải framework [Kivy](https://kivy.org/#home)

**Thư viện PyGobject**

PyGObject là một mô-đun mở rộng của Python. Nó cung cấp quyền truy cập tốt và nhất quán vào toàn bộ nền tảng phần mềm Gnome với sự trợ giúp của GObject Introspection. Nó hỗ trợ đầy đủ GObject Introspection và các tính năng của nó như gọi lại, phân lớp phụ, hỗ trợ GVariant, bao đóng, v.v. Nó có sẵn cho các hệ điều hành như Windows, macOS, Linux, v.v. Nó hỗ trợ Python3 trở lên. Trong bài viết này, chúng ta sẽ xem quy trình từng bước để cài đặt PyGObject cho Python trong Windows.

# PyGObject

**PyGObject**là một phần mềm thư viện nguồn mở và hoàn toàn miễn phí được viết bằng Python và được thiết kế cho các nhà phát triển muốn có một bộ sưu tập các ràng buộc Python cho Hệ thống đối tượng GLib (GObject) và thư viện GLib. trong ngôn ngữ lập trình Python. Mục tiêu chính của thư viện PyGObject là hỗ trợ đầy đủ sự quan tâm của GObject và tất cả các tính năng của nó, bao gồm cả callbacks, closures, GVariant support, etc.

Một bộ hoàn chỉnh các ràng buộc Python

PyGObject là một tập hợp khá đầy đủ các ràng buộc Python, rất hữu ích và có thể được sử dụng để viết các chương trình đơn giản và phức tạp (xem thư mục ví dụ trong tarball cho một số ví dụ ngẫu nhiên, nhưng đơn giản của các chương trình bạn có thể viết bằng thư viện này ).

Dưới mui xe và sẵn có

Thư viện được viết hoàn toàn bằng ngôn ngữ lập trình Python, có nghĩa là nó có thể được cài đặt dễ dàng trên bất kỳ hệ điều hành GNU / Linux nào có sẵn Python. Bởi vì điều này, nó có thể được tải xuống từ Softoware như là một kho lưu trữ nguồn phổ quát, đòi hỏi bạn phải cấu hình và biên dịch chương trình trước khi cài đặt.

Yêu cầu bắt buộc

Điều quan trọng cần đề cập ở đây là để biên dịch thư viện PyGObject trên hệ điều hành dựa trên hạt nhân Linux của bạn, bạn sẽ có trình biên dịch C được hỗ trợ (GCC hoặc MSVC), Python 2.7 hoặc cao hơn, GLib và Gio 2.38.0 hoặc cao hơn, gobject-introspection 1.38.0 hoặc cao hơn, cũng như thư viện libffi, là tùy chọn tại thời điểm này.

Hướng dẫn cài đặt

Để cài đặt PyGObject, bạn sẽ phải thực thi & ldquo; ./ configure --prefix = && make && make install & rdquo; lệnh trong trình mô phỏng thiết bị đầu cuối, xem xét thực tế là bạn đã đăng nhập với tư cách là người quản trị hệ thống (root). Trên một số bản phân phối GNU / Linux, bạn có thể dễ dàng cài đặt thư viện từ kho phần mềm mặc định của nó. Cả kiến ​​trúc 32 bit và 64 bit đều được hỗ trợ tại thời điểm này.

**Có gì mới**trong bản phát hành này:

* Khắc phục Gio.Application bị rò rỉ trong trường hợp không xử lý tín hiệu được thiết lập trước.
* Cảnh báo quan trọng của Squash khi sử dụng mảng làm giá trị băm.

**Tính năng mới**trong phiên bản:

* Hoàn nguyên & quot; setup.py: Đồng thời đặt setup\_requires để yêu cầu pycairo & quot; (Christoph Reiter)
* setup.py: Cũng đặt setup\_requires để yêu cầu pycairo (Christoph Reiter)
* setup.py: Cung cấp dự phòng os.path.samefile cho Python 2 trong Windows (Christoph Reiter)
* Thêm tài liệu dựa trên nhân sư (Christoph Reiter) (# 791448)
* PKG-INFO: Hoàn nguyên tên quay lại PyGObject (Christoph Reiter)
* setup.py: Rework pycairo discovery để không sử dụng pkg-config (Christoph Reiter)
* setup.py: Khắc phục lệnh distcheck trên Windows (Christoph Reiter)
* setup.py: Xóa các trình phân loại khác nhau và url tải xuống không được pypi (Christoph Reiter) chấp nhận
* bướu phiên bản (Christoph Reiter)

**Tính năng mới**trong phiên bản 3.26.1:

* đối tượng pygobject: Khắc phục Python GC thu thập chu kỳ ref quá sớm (Christoph Reiter) (# 731501)
* Sửa lỗi truy cập bộ nhớ không khởi tạo tiềm năng trong suốt GC (Daniel Colascione) (# 786872)
* kiểm tra: hoàn nguyên các phần của bài kiểm tra trước đó vì nó bị hỏng trên các bản dựng 32 bit (Christoph Reiter) (# 786948)
* cờ: Thêm testcase cho lỗi 786948 (Christoph Reiter) (# 786948)
* sửa lỗi tràn tiềm năng khi marshalling cờ từ giao diện py (Philippe Renon) (# 786948)
* to\_py\_array: Xử lý đúng các mục mảng enum (Christoph Reiter) (# 788890)
* đóng cửa: Khắc phục sự truy cập chưa được ký và out-of-bounds (James Clarke) (# 788894)
* xây dựng: Khắc phục sự cố không cài đặt tệp .egg-info (Christoph Reiter) (# 777719)
* configure.ac: phiên bản bump thành 3.26.1 (Christoph Reiter)

**Tính năng mới**trong phiên bản 3.26.0:

* configure.ac: phiên bản trước khi phát hành sẽ tăng lên 3.26.0 (Christoph Reiter)
* đóng: im lặng cảnh báo trình biên dịch mới (Christoph Reiter)
* kiểm tra: bỏ qua một số thử nghiệm không thành công trong Windows với Python 3.6 (Christoph Reiter)
* kiểm tra: pyflakes / pep8 fixes (Christoph Reiter)
* kiểm tra: Khắc phục thử nghiệm cairo với pycairo & gt; = 1.13 (Christoph Reiter)
* Đảm bảo thông tin phiên bản được chuyển tới require\_version là một chuỗi. (Benjamin Berg) (# 781582)
* configure.ac: phiên bản hậu phát hành bị bướu lên 3.25.2 (Christoph Reiter)

**Tính năng mới**trong phiên bản 3.25.1:

* configure.ac: phiên bản tiền phát hành gặp phải 3.24.0 (Christoph Reiter)

**Có gì mới**trong phiên bản 3.24.0:

* configure.ac: phiên bản tiền phát hành gặp phải 3.24.0 (Christoph Reiter)

**Tính năng mới**trong phiên bản 3.22.0:

* configure.ac: phiên bản tiền phát hành gặp phải 3.22.0 (Christoph Reiter)

**Tính năng mới**trong phiên bản 3.20.1 / 3.22.0 Beta 2:

* Cho phép cài đặt bằng pip (Mathieu Bridon) (# 767988)
* Bỏ qua một thử nghiệm với glib cũ hơn (Christoph Reiter) (# 740301)
* Sửa thử nghiệm bằng Python 3.1 / 3.2 (Arfrever Frehtes Taifersar Arahesis, Christoph Reiter) (# 740324)
* kiểm tra: Sử dụng địa điểm kwarg để xác nhậnAlmostEqual (Arfrever Frehtes Taifersar Arahesis, Christoph Reiter) (# 740337)
* In ngoại lệ nếu marshalling một đối số tín hiệu không thành công (Christoph Reiter) (# 748198)
* ghi đè: cho phép viết tắt trình tự treemodel (Marinus Schraal) (# 766580)
* Loại bỏ pygobject-external.h (Christoph Reiter) (# 767084)
* Loại bỏ pygobject-private.h và đổi tên pygobject.c thành pygobject-object.c (Christoph Reiter) (# 767084)
* Hợp nhất pyglib-private.h thành pyglib.h (Christoph Reiter) (# 767084)
* Loại bỏ pygi.h và pygi-private.h (Christoph Reiter) (# 767084)
* configure.ac: phiên bản hậu phát hành gặp 3.21.1 (Simon Feltman)

**Tính năng mới**trong phiên bản 3.18.2:

* configure.ac: phiên bản phát hành của phiên bản là 3,18 .2

**Tính năng mới**trong phiên bản 3.18 Beta 1:

* Cho phép chuyển danh sách unicode sang thuộc tính GStrv trên Python 2 (Christoph Reiter) (# 744745)
* Tránh một sự im lặng dài để int cắt ngắn (Rui Matos) (# 749698)
* Xử lý marshalling gtype (Mathieu Bridon) (# 749696)
* pygi-foreign-cairo.c: sửa lỗi bao gồm cho py3cairo.h (Daniel Hahler) (# 746742)
* kiểm tra: Tắt tiếng các thông báo lỗi và cảnh báo khác nhau (Christoph Reiter) (# 751156)
* Khắc phục sự cố hồi quy khi xdg-user-dirs chưa được cài đặt (Christoph Reiter) (# 751299)
* Kiểm tra rõ ràng nếu ghi đè tồn tại thay vì ImportError (Garrett Regier) (# 749532)

**Tính năng mới**trong phiên bản 3.16.2:

* ghi đè: Cung cấp thuộc tính \_overrides\_module

**Tính năng mới**trong phiên bản 3.16.0:

* : Không sử dụng thuộc tính ghi đè không được chấp nhận
* Thêm GLib.MINFLOAT v.v. và đánh dấu GObject.G\_MINFLOAT v.v. không được chấp nhận
* Phát ra PyGIDeprecationCảnh báo khi truy cập các thuộc tính ghi đè không được chấp nhận
* Thêm tên vùng và tên vùng chứa vào tất cả các cảnh báo / thông báo lỗi
* kiểm tra: Thêm kiểm tra cho GIRepository.UnionInfo.get\_size ()
* Tránh trùng lặp tên tệp khi marshalling từ Python thành C

**Tính năng mới**trong phiên bản 3.14.0:

* configure.ac: phiên bản trước phát hành là 3.14.0 (Simon Feltman)

**Có gì mới**trong phiên bản 3.14 RC:

* kiểm tra: Thêm kiểm tra cho Gio.Application.add\_main\_option ( ) (Simon Feltman)
* kiểm tra: Chia nhỏ các trường hợp kiểm tra khác nhau (Simon Feltman) (# 735193)
* Sửa lỗi đọc không hợp lệ trong mã dọn dẹp đối số (Simon Feltman)
* Khắc phục sự cố quản lý bộ nhớ với các đối số cấu trúc cho các tín hiệu (Simon Feltman) (# 736175)

**Có gì mới**trong phiên bản 3.12.2:

* Bản sửa lỗi PEP8 (Simon Feltman)
* Python 3.4 thực hiện các bản sửa lỗi (Simon Feltman) (# 730411)

**Tính năng mới**trong phiên bản 3.11.5:

* tái cấu trúc bộ nhớ cache: Di chuyển tất cả trình sắp xếp bộ nhớ cache vào tệp dựa trên loại (Simon Feltman) (# 709700)
* kiểm tra: Thêm kiểm tra cho cấu trúc có khung được sở hữu được chuyển vào trong cuộc gọi lại (Mike Gorse) (# 722899)
* build: Thêm tùy chọn cấu hình - không có cấu hình chung cho các nhà duy trì gói (Patrick Welche) (# 721646)
* demo: Thêm trình diễn thực hiện giao diện TreeModel (Simon Feltman)
* bản dựng: Đặt lại PLATFORM\_VERSION thành 3.0 (Colin Walters)
* kiểm tra: Chỉ chạy PyFlakes và PEP8 trên SUBDIRS (Simon Feltman)
* Kết hợp các mô-đun PyGLib và PyGObject tĩnh vào PyGI (Simon Feltman) (# 712197)
* Thêm kiểm tra đối số dữ liệu người dùng gọi lại với các đối số sau (Martin Pitt) (# 722104)

**Tính năng mới**trong phiên bản 3.11.4:

* ghi đè: Sửa \_\_repr\_\_ cho các cấu trúc Gdk khác nhau (Simon Feltman)
* Thêm phương thức thành viên enum và cờ (Simon Feltman) (# 693099)
* python.m4: g / c JD\_PYTHON\_CHECK\_VERSION (Patrick Welche) (# 721662)
* Hỗ trợ tạo liên minh với PyGIStruct (Simon Feltman)
* tài liệu: Liệt kê các hàm tạo trong chuỗi đối tượng và struct doc (Simon Feltman) (# 708060)
* tài liệu: Khắc phục đối số độ dài mảng bỏ qua với các đối số trước đó
* tài liệu: Thêm giá trị trả về và bỏ qua các đối số ẩn trong các hàm (Simon Feltman) (# 697356)
* tài liệu: Bỏ qua chiều dài mảng tiềm ẩn args khi tạo chuỗi tài liệu chức năng (Simon Feltman) (# 697356)
* gtk-demo: Thêm bản trình diễn CSS (Gian Mario Tagliaretti) (# 719722)
* xây dựng: Tránh xung đột giữa gi / types.py và stdlib (Colin Watson) (# 721025)

**Có gì mới**trong phiên bản 3.11.3:

* Thay thế việc sử dụng PyGIBoxed\_Type bằng PyGIStruct\_Type (Simon Feltman) (# 581525)

**Tính năng mới**trong phiên bản 3.11.2:

* gkt-demo: Thay đổi sổ tay thông tin / nguồn chính thành GtkStack (Simon Feltman)
* Thêm cảnh báo không dùng nữa và ghi đè trình khởi tạo lớp làm sạch (Simon Feltman) (# 705810)
* Khắc phục phương thức dir cho static GParamSpec trong Python 3 (Simon Feltman)
* Xóa kiểm tra đối số quá mức đối với userdata gọi lại (Simon Feltman) (# 711173)

**Có gì mới**trong phiên bản 3.10.2:

* Khắc phục sự cố an toàn chủ đề bằng cách luôn bật GIL ( Simon Feltman) (# 709223, # 710447)

**Tính năng mới**trong phiên bản 3.11.1:

* Khắc phục các vấn đề về an toàn của người dùng bằng cách luôn bật GIL (Simon Feltman) (# 709223)
* Thêm chế độ dọn dẹp GLib.MainLoop SIGINT phù hợp (Simon Feltman) (# 710978)
* tài liệu: Thêm giá trị từ khóa Không có cho chú thích không cho phép (Simon Feltman) (# 640812)
* Xóa ghi đè để hỗ trợ chức năng tín hiệu GObject trước 3.10 (Simon Feltman)
* Thêm chủ đề\_init trở lại làm yêu cầu đối với các bản repo không phải Python (Simon Feltman) (# 710447)
* Thêm phương thức dir vào bộ truy cập đạo cụ GObject (Simon Feltman) (# 705754)
* Xóa PyGObjectWeakRef bây giờ rằng g\_binding\_unbind tồn tại (Simon Feltman) (# 699571)
* Khắc phục nhiều rò rỉ bộ nhớ rò rỉ (Simon Feltman) (# 693402, # 709397)
* Thêm hỗ trợ cho các đối số dữ liệu người dùng biến (Simon Feltman) (# 640812)
* Bump glib và g-i phụ thuộc để ổn định mới nhất. (Martin Pitt)

**PyGObject** là một gói Python cung cấp các ràng buộc cho các thư viện dựa trên [GObject](https://developer.gnome.org/gobject/stable/) như [GTK](https://www.gtk.org/) , [GStreamer](https://gstreamer.freedesktop.org/) , [WebKitGTK](https://webkitgtk.org/) , [GLib](https://developer.gnome.org/glib/stable/) , [GIO](https://developer.gnome.org/gio/stable/) và nhiều thứ khác.

Nó hỗ trợ Linux, Windows và macOS, đồng thời hoạt động với **Python 3.7+** và **PyPy3** . PyGObject, bao gồm cả tài liệu này, được cấp phép theo **LGPLv2.1+** .

Nếu bạn muốn viết một ứng dụng Python cho [GNOME](https://www.gnome.org/) hoặc một ứng dụng Python GUI bằng GTK, thì PyGObject là lựa chọn phù hợp. Để biết thêm thông tin về các thư viện cụ thể, hãy xem “ [Hướng dẫn Python GTK 3](https://python-gtk-3-tutorial.readthedocs.io/) ” và “ [Tham khảo API Python GI](https://lazka.github.io/pgi-docs) ”.

**import** gi

gi**.**require\_version**(**"Gtk"**,** "3.0"**)**

**from** gi.repository **import** Gtk

window **=** Gtk**.**Window**(**title**=**"Hello World"**)**

window**.**show**()**

window**.**connect**(**"destroy"**,** Gtk**.**main\_quit**)**

Gtk**.**main**()**

# Làm thế nào nó hoạt động?[ℑ](https://pygobject.readthedocs.io/en/latest/#how-does-it-work)

PyGObject sử dụng [glib](https://developer.gnome.org/glib/stable/) , [gobject](https://developer.gnome.org/gobject/stable/) , [girepository](https://developer.gnome.org/gi/stable/) , [libffi](https://sourceware.org/libffi/) và các thư viện khác để truy cập thư viện C (libgtk-3.so) kết hợp với siêu dữ liệu bổ sung từ tệp typelib đi kèm (Gtk-3.0.typelib) và tự động cung cấp giao diện Python dựa trên thông tin đó.

# Ai đang sử dụng PyGObject?[ℑ](https://pygobject.readthedocs.io/en/latest/#who-is-using-pygobject)

* [Anaconda](https://fedoraproject.org/wiki/Anaconda) - một chương trình cài đặt được sử dụng bởi Fedora, RHEL và những người khác
* [D-Feet](https://wiki.gnome.org/action/show/Apps/DFeet) - trình gỡ lỗi D-Bus dễ sử dụng
* [Gaphor](https://gaphor.org/) - một công cụ mô hình hóa đơn giản
* [Girens](https://gitlab.gnome.org/tijder/girens) - ứng dụng khách Plex để phát phim, chương trình TV và nhạc từ thư viện Plex của bạn
* [Âm nhạc Gnome](https://wiki.gnome.org/Apps/Music) - trình phát nhạc cho Gnome
* [GNOME Tweak Tool](https://wiki.gnome.org/action/show/Apps/GnomeTweakTool) - công cụ tùy chỉnh các tùy chọn nâng cao của GNOME 3
* [Gramps](https://gramps-project.org/) - một chương trình phả hệ
* [Lollypop](https://wiki.gnome.org/Apps/Lollypop) - một máy nghe nhạc hiện đại
* [Meld](http://meldmerge.org/) - một công cụ hợp nhất và khác biệt trực quan
* [MyPaint](http://mypaint.org/) - một công cụ nhanh nhẹn, không bị phân tâm và dễ dàng dành cho các họa sĩ kỹ thuật số
* [Nicotine+](https://nicotine-plus.org/) - ứng dụng khách đồ họa cho mạng chia sẻ tệp Soulseek
* [Orca](https://wiki.gnome.org/Projects/Orca) - trình đọc màn hình linh hoạt và có thể mở rộng
* [Pithos](https://pithos.github.io/) - khách hàng của Pandora Radio
* [Pitivi](http://www.pitivi.org/) - trình chỉnh sửa video mã nguồn mở và miễn phí
* [Quod Libet](https://quodlibet.readthedocs.io/) - trình quản lý / trình phát thư viện nhạc
* [Terminator](https://gnome-terminator.org/) - Tương lai robot của thiết bị đầu cuối
* [Transmageddon](http://www.linuxrising.org/) - bộ chuyển mã video

Các ứng dụng hoặc thư viện sau đây sử dụng PyGObject cho các tính năng tùy chọn, chẳng hạn như phần bổ trợ hoặc phần phụ trợ tùy chọn:

* [củ cải đường](http://beets.io/) - trình quản lý thư viện nhạc và trình gắn thẻ MusicBrainz
* [gedit](https://wiki.gnome.org/Apps/Gedit) - trình soạn thảo văn bản Gnome
* [matplotlib](http://matplotlib.org/) - thư viện vẽ đồ thị 2D python
* [Totem](https://wiki.gnome.org/Apps/Videos) - trình phát video cho Gnome

**Thư viện PyQt**

 Qt là một trong những ràng buộc Python đa nền tảng được ưa chuộng triển khai thư viện Qt cho **Framework Python GUI** phát triển ứng dụng Qt (thuộc sở hữu của Nokia).

 Hiện tại, Qt có sẵn cho Unix / Linux, Windows, Mac OS X và Sharp Zaurus. Nó kết hợp những gì tốt nhất của Python và Qt và tùy thuộc vào từng lập trình viên để quyết định tạo một chương trình bằng cách viết code hay sử dụng Qt Designer để tạo các hộp thoại trực quan.

 Qt có sẵn trong cả giấy phép thương mại cũng như GPL. Mặc dù một số tính năng có thể không có trong phiên bản miễn phí, nhưng nếu ứng dụng của bạn là mã nguồn mở thì bạn có thể sử dụng nó theo giấy phép miễn phí.  
Link tải framework [Qt](https://riverbankcomputing.com/news)

**Thư viện WxPython**

 WxPython là một trình bao bọc mã nguồn mở cho thư viện Python GUI đa nền tảng WxWidgets (trước đó được gọi là WxWindows) và được triển khai như một mô-đun mở rộng Python.

Với WxPython, bạn có thể tạo các ứng dụng gốc cho Windows, Mac OS và Unix.

[Tham khảo thêm cách cài đặt python trên Windows, Mac OS và Unix..](https://mcivietnam.com/blog-detail/cach-cai-python/)

Nếu bạn mới bắt đầu phát triển các ứng dụng trong thư viện WxPython, đây là một hướng dẫn đơn giản mà bạn có thể tham khảo: https://zetcode.com/wxpython/  
Link tải framework [WxPython](https://www.wxpython.org/)

**Thư viện PySide**

 PySide là một dự án phần mềm mã nguồn mở cung cấp các ràng buộc Python cho Qt Framework.

 Qt là một ứng dụng đa nền tảng và **Python GUI Framework,** cho phép các lập trình viên viết ứng dụng một lần và triển khai chúng trên nhiều hệ điều hành mà không cần viết lại mã nguồn.

 Trong khi Python là một ngôn ngữ lập trình hiện đại, năng động với một cộng đồng lập trình viên khổng lồ. Kết hợp sức mạnh của Qt và Python, PySide cung cấp Qt Framework phong phú cho các lập trình viên sử dụng Python phát triển ứng dụng GUI nhanh chóng trên tất cả các hệ điều hành chính.  
Link tải framework [Pyside](https://pypi.org/project/PySide/)

**PyGUI**

 PyGUI là một **Python GUI Framework** đa nền tảng ứng cho Unix, Macintosh và Windows. So với một số GUI Framework khác, cho đến nay, PyGUI là đơn giản nhất và nhẹ nhất, vì API hoàn toàn đồng bộ với Python.

 PyGUI chèn rất ít code giữa nền tảng GUI và ứng dụng Python, do đó giao diện của ứng dụng thường hiển thị GUI tự nhiên của nền tảng.  
Link tải framework [PyGUI](https://www.cosc.canterbury.ac.nz/greg.ewing/python_gui/version/Doc/index.html)

### là 6 **Python GUI Framework** để người dùng [Python](https://mcivietnam.com/lap-trinh-python/) có thể làm app giao diện một cách hiệu quả và nhanh chóng nhất

TkInter trong Python

Để chạy được code tkInter Python thì cài thêm phần mềm hỗ trợ anaconda



Code cửa sổ hiển thị

Graphical user interface

Description automatically generated

Tạo Label

Graphical user interface, text, application

Description automatically generated

Màu chữ và nền chữ Label

A picture containing graphical user interface

Description automatically generated

Tạo nút button

A picture containing chart

Description automatically generated

Đổi màu button

Application

Description automatically generated with medium confidence

Hiện thông báo khi click vào button

Graphical user interface, text, application

Description automatically generated

Click button

Graphical user interface, text, application

Description automatically generated

Câu lệnh Entry

Graphical user interface, text, application

Description automatically generated

Liên kết Label, Button, Entry tạo thành chương trình

Text

Description automatically generated

Graphical user interface, application

Description automatically generated

A picture containing logo

Description automatically generated

Graphical user interface, text

Description automatically generated with medium confidence

Graphical user interface, text, application

Description automatically generated

# KẾT LUẬN

## Ưu điểm

## Nhược điểm

## Hướng phát triển

# TÀI LIỆU THAM KHẢO

1. **https://pygobject.readthedocs.io/en/latest/**
2. [https://](https://play.google.com/store/apps/details?id=com.grabtaxi.passenger&hl=en&gl=US)
3. **https://www.geeksforgeeks.org/**