**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: Code GUI trong Python về: tkInter, PyGObject, PyQt, PySide, Kivy, wxPython
* Chương 4: Kết luận

**MỤC LỤC**

[LỜI MỞ ĐẦU 3](#_Toc123916091)

[Chương 1: TÌM HIỂU VỀ GUI, GUI trong PYTHON 5](#_Toc123916092)

1. GUI 6-7
2. GUI Python 7

Chương 2: Tìm hiểu GUI trong Python về: tkInter, PyGObject, PyQt, PySide, Kivy, wxPython 8

1. Thư viện TkInter 8
2. Thư viện PyQt 8-9
3. Thư viện Kivy 9
4. Thư viện PyGObject 10
5. Thư viện PySide 10-11
6. Thư viện wxPython 11

Chương 3: Code GUI trong Python về: tkInter, PyGObject, PyQt, PySide, Kivy, wxPython 12

1. Thiết kế giao diện TkInter 12-16
2. Thiết kế giao diện PyQt 17-19
3. Thiết kế giao diện Kivy 19-22
4. Thiết kế giao diện PyGObject
5. Thiết kế giao diện PySide
6. Thiết kế giao diện wxPython

Chương 4: KẾT LUẬN 25

TÀI LIỆU THAM KHẢO 26

# KHÁI NIỆM VỀ GUI, GUI PYTHON

## GUI

* 1. **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:
* Cửa sổ làm việc 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 Web, cửa sổ bật lên trên Internet,…
* Menu: là thành phần thường thấy của GUI, và ở đây người dùng có thể thực hiện lệnh để tương tác với máy tính thông qua danh sách các lựa chọn mà phần mềm cung cấp cho người dùng.
* Biểu tượng (Icon): Thông thường, các biểu tượng sẽ được hiển thị dưới dạng hình ảnh, giúp cho người dùng có thể tương tác nhanh với máy tính như mở tài liệu, khởi chạy ứng dụng nào đấy. Trong một số trường hợp, người dùng sẽ có thể tìm kiếm tệp thông qua biểu tượng của ứng dụng, từ đó sẽ tiết kiệm thời gian.
* Widget: là một thành phần thuộc các ứng dụng, mà ở đây người dùng có thể thực hiện một lệnh tương tác cụ thể với ứng dụng đấy.
* **Thành phần tương tác:**

**Con trỏ: đây là thành phần giúp người dùng định hướng được vị trí người dùng sẽ tương tác, chẳng hạn như vị trí mà con trỏ chuột người dùng click hay vị trí mà người dùng sẽ nhập liệu các ký tự.**

**Thao tác chọn: Người dùng có thể thực hiện thao tác chọn với các thành phần có trên cửa sổ làm việc. Tương tác chọn có thể được thực hiện bởi cả chuột, bàn phím, bút cảm ứng,...**

**Thao tác kéo thả: Thông thường, đây là thao tác được sử dụng để tương tác với các tệp hay hình ảnh trong cửa sổ làm việc.**

* 1. **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 người dùng nên đầu tư vào kỹ năng làm web thay vì học để tạo một app trên desktop như cách đây chục năm.

# TÌM HIỂU VỀ GUI PYTHON VỚI 6 THƯ VIỆN THÔNG DỤNG

Hiện có 6 thư viện thông dung trong Python GUI là:

1. Thư viện Tkinter
2. Thư viện PyQt
3. Thư viện Kivy
4. Thư viện PyGObject
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 người dùng trong trường hợp người dùng mới bắt đầu học, rất nhiều lỗi người dùng có thể tìm ra cách sửa chữa ngay lập tức.

Tkinter chạy trên hệ điều hành phổ biến: Windows, Ubuntu, OSX/MacOS.

1. **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 người dùng là mã nguồn mở thì người dùng có thể sử dụng nó theo giấy phép miễn phí.

1. **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. Người dùng 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. Người dùng 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 người dùng bắt đầu nhanh chóng.

1. **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**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.

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 người dùng có thể viết bằng thư viện này).

1. **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.

1. **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, người dùng có thể tạo các ứng dụng gốc cho Windows, Mac OS và Unix.

**BẢNG SO SÁNH 6 THƯ VIỆN CHẠY TRÊN HỆ ĐIỀU HÀNH**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Thư viện** | **Tkinter** | **PyQt** | **Kivy** | **PyGobject** | **PySide** | **wxPython** |
| Hệ diều hành |  |  |  |  |  |  |
| Windows | x | x | x | x | x | X |
| Mac OS/OS X | x | x | x | x | x | X |
| Unix |  | x |  |  |  | x |
| Linux |  | x | x | x | x |  |
| Android |  | x |  |  |  |  |
| iOS |  | x | x |  |  |  |
| Raspberry Pi |  |  | x |  |  |  |
| Sharp Zaurus |  | x |  |  |  |  |
| Ubuntu | x |  |  |  |  |  |

# CODE GUI VỚI PYTHON VỀ 6 THƯ VIỆN HIỆN HÀNH

1. **TkInter trong Python**

A picture containing logo

Description automatically generated

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

Graphical user interface, text

Description automatically generated with medium confidence

Graphical user interface, text, application

Description automatically generated

1. **Thư viện PyQt**

Cài đặt PyQt

Text

Description automatically generated

Dùng phần mềm hỗ trợ Qt Designer

Graphical user interface, application, Word

Description automatically generated

Tạo label, text, push button, đổi tên nhãn, nút

Graphical user interface, application

Description automatically generated

Sau đó kết nối với phần mềm hỗ trợ Pycharm

Lưu Qt vào chung đường dẫn với Pycharm

Graphical user interface, text, application

Description automatically generated

Tạo form trong Pycharm .py để import form Qt

Code kế thừa để hiển thị màn hình từ Qt

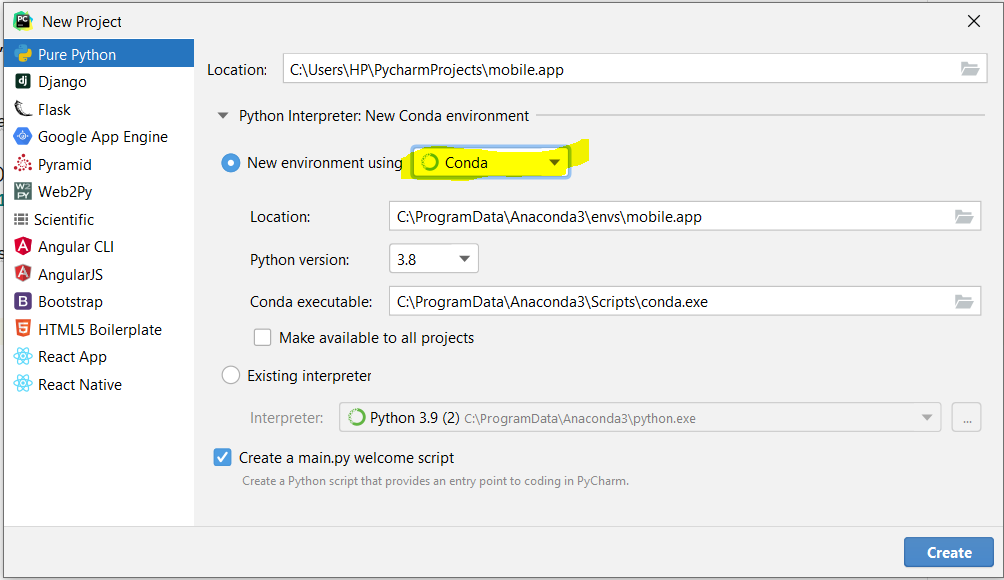
Graphical user interface, text, application

Description automatically generated

1. **Thư viện Kivy**

Sử dụng chương trình phụ hỗ trợ Pycharm

Tạo project mobile.app



Tiến hành cài đặt Kivy. Kiểm tra Terminal phần cài đặt Kivy

Graphical user interface, text, application

Description automatically generated

Kiểm tra xem đã import Kivy thành công chưa

Graphical user interface, application

Description automatically generated

Phần code cơ bản của Kivy:

Graphical user interface, text, application

Description automatically generated

Graphical user interface, application, Word

Description automatically generated

Graphical user interface, text, application

Description automatically generated

Chuyển đổi 2 button bên phải – bên trái

Graphical user interface, application

Description automatically generated

1. **Thư viện PyGObject**
2. **Thư viện PySide**
3. **Thư viện wxPython**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Bảng chi tiết dùng phần mềm phụ hỗ trợ** | | | | | | |
|  | Anaconda | Pycharm | Qt |  |  |  |
| Tkinter | x |  |  |  |  |  |
| PyQt |  | x | x |  |  |  |
| Kivy |  | x |  |  |  |  |
| PyGObject |  |  |  |  |  |  |
| PySide |  |  |  |  |  |  |
| wxPython |  |  |  |  |  |  |

# KẾT LUẬN

6 **Python GUI Framework** trên giúp người dùng có thể Python làm app giao diện một cách hiệu quả và nhanh chóng nhất. Người dùng có thể lựa chọn bất kỳ thư viện nào phù hợp với mục đích của người dùng để xây dựng một app giao diện hoàn chỉnh. Chúc các người dùng hoàn thành được một app giao diện hoàn chỉnh của riêng mình.

# TÀI LIỆU THAM KHẢO

1. **https://pygobject.readthedocs.io/en/latest/**
2. [**https://www.geeksforgeeks.org/**](https://www.geeksforgeeks.org/)
3. **https://kivy.org/doc/stable/installation/installation-windows.html**