BDRPC187**BỘ GIÁO DỤC VÀ ĐÀO TẠO**

TRƯỜNG ĐẠI HỌC SÀI GÒN



KHOA CÔNG NGHỆ THÔNG TIN

**ĐỀ TÀI**

**TÌM HIỂU CÔNG NGHỆ LẬP TRÌNH PYTHON**

**CROSS-PLATFORM FRAMEWORK TKINTER, PYQT5**

**Học phần : Các công nghệ lập trình hiện đại**

**Nhóm :** 17

**Sinh viên thực hiện :** Lý Hiên Ninh – 3119410295

Giảng viên hướng dẫn : Phạm Thi Vương

**Tháng 9 năm 2023**

MỤC LỤC

[CHƯƠNG I. NGÔN NGỮ PYTHON 4](#_Toc151405642)

[**1. Các cú pháp 4**](#_Toc151405643)

[1.1. Biến và kiểu dữ liệu 4](#_Toc151405644)

[1.2. Câu lệnh điều kiện 4](#_Toc151405645)

[1.3. Vòng lặp 4](#_Toc151405646)

[1.4. Hàm 4](#_Toc151405647)

[1.5. Chuỗi 5](#_Toc151405648)

[1.6. Danh sách 5](#_Toc151405649)

[**2. Ưu - nhược điểm 6**](#_Toc151405650)

[2.1. Ưu điểm 6](#_Toc151405651)

[2.2. Nhược điểm 6](#_Toc151405652)

[CHƯƠNG II. FRAMEWORK TKINTER 7](#_Toc151405653)

[**1. Widget 7**](#_Toc151405654)

[**2. Phương thức bố cục 8**](#_Toc151405655)

[Phương thức pack() 8](#_Toc151405656)

[Phương thức grid() 8](#_Toc151405657)

[Phương thức place() 9](#_Toc151405658)

[**3. Ưu - nhược điểm 9**](#_Toc151405659)

[3.1. Ưu điểm 9](#_Toc151405660)

[3.2. Nhược điểm 10](#_Toc151405661)

[CHƯƠNG III. API 10](#_Toc151405662)

[**1. Tổng quát 10**](#_Toc151405663)

[**2. Cách hoạt động 11**](#_Toc151405664)

[**3. Các loại API 12**](#_Toc151405665)

[**3.1. Dựa trên phân ngành nhỏ trong ngành công nghệ thông tin 12**](#_Toc151405666)

[**3.2. Dựa trên quyền hạn truy cập 12**](#_Toc151405667)

[**4. Ưu - nhược điểm 12**](#_Toc151405668)

[4.1. Ưu điểm 12](#_Toc151405669)

[4.2. Nhược điểm 13](#_Toc151405670)

[**5. Tính bảo mật cho API 13**](#_Toc151405671)

[CHƯƠNG IV. ÁP DỤNG GUI FRAMEWORK VÀO ĐỀ TÀI 14](#_Toc151405672)

[1.1. Cài đặt môi trường 14](#_Toc151405673)

[1.2. Thực hiện chức năng 15](#_Toc151405674)

[**1.2.1. API 15**](#_Toc151405675)

[**1.2.2. Tkinter 15**](#_Toc151405676)

[**Tìm kiếm kỹ năng 15**](#_Toc151405677)

[**Tìm kiếm vật phẩm 18**](#_Toc151405678)

[**1.2.3. PyQt5 21**](#_Toc151405679)

[**Tìm kiếm vật phẩm 21**](#_Toc151405680)

[CHƯƠNG V. SO SÁNH GIỮA 2 GUI FRAMEWORK TKINTER VÀ PYQT5 23](#_Toc151405681)

# **CHƯƠNG I. NGÔN NGỮ PYTHON**

## **1. Các cú pháp**

### **1.1. Biến và kiểu dữ liệu**

* Khai báo biến: **variable\_name = value**
* Các kiểu dữ liệu cơ bản: **int, float, str, bool, list, tuple, set, dict.**
* Ép kiểu (type casting): **int(), float(), str(), list(), tuple(), set(), dict().**

### **1.2. Câu lệnh điều kiện**

**if, elif, else** để kiểm tra điều kiện.

Ví dụ :

**if** condition:

# code nếu condition là True

**elif** another\_condition:

# code nếu another\_condition là True

**else**:

# code nếu cả hai điều kiện đều False

### **1.3. Vòng lặp**

* Vòng lặp **for** dùng để lặp qua một dãy.
* Vòng lặp **while** dùng để lặp cho đến khi một điều kiện không còn đúng.

Ví dụ :

**for** item **in** iterable:

# code thực hiện trên mỗi phần tử của iterable

**while** condition:

# code trong vòng lặp while

### **1.4. Hàm**

* Định nghĩa hàm: **def function\_name(parameters):**
* Gọi hàm: **result = function\_name(arguments**)

Ví dụ :

**def** **add**(a, b):

**return** a + b

result = add(**3**, **5**)

### **1.5. Chuỗi**

* Định nghĩa chuỗi: **'Hello, world!' hoặc "Hello, world!"**
* Cắt chuỗi: **string[start:end]**
* Định dạng chuỗi: **f"Hello, {name}"**

### **1.6. Danh sách**

* **List**
* Định nghĩa list: **[item1, item2, item3]**
* Truy cập phần tử: **my\_list[index]**
* Thêm và xóa phần tử: **append(), insert(), remove(), pop()**
* **Tuples**
* Định nghĩa tuple: **(item1, item2, item3)**
* Tương tự list nhưng không thể thay đổi giá trị sau khi tạo.
* **Dictionary**
* Định nghĩa dictionary: **{'key1': value1, 'key2': value2}**
* Truy cập giá trị: **my\_dict['key']**
* Thêm và xóa phần tử: **my\_dict['new\_key'] = value, del my\_dict['key']**
* **Sets**
* Định nghĩa set: **{item1, item2, item3}**
* Thêm và xóa phần tử: **add(), remove(), discard()**

## **2. Ưu - nhược điểm**

### **2.1. Ưu điểm**

* **Dễ Đọc và Dễ Viết:** Python có cú pháp đơn giản và gần giống ngôn ngữ tự nhiên, giúp mã nguồn dễ đọc, dễ viết và dễ hiểu. Điều này làm cho Python rất thích hợp cho người mới học lập trình.
* **Thư Viện và Frameworks Đa Dạng:** Python có một hệ sinh thái phong phú của các thư viện và frameworks. Ví dụ, Django và Flask cho phát triển web, NumPy và pandas cho xử lý dữ liệu, TensorFlow và PyTorch cho học máy, Matplotlib cho vẽ đồ thị, và nhiều thư viện khác.
* **Tương Tác:** Python hỗ trợ mô hình tương tác thông qua trình thông dịch (interpreter) nơi bạn có thể thử nghiệm và kiểm tra mã một cách nhanh chóng. Jupyter Notebook là một ví dụ tiêu biểu cho môi trường này.
* **Tích Hợp Dễ Dàng:** Python dễ dàng tích hợp với các ngôn ngữ khác như C, C++, và Java. Điều này cho phép bạn sử dụng các thư viện chuyên biệt khi cần.
* **Hỗ Trợ Đa Nền Tảng:** Python là một ngôn ngữ đa nền tảng, có sẵn trên nhiều hệ điều hành như Windows, macOS, và Linux.
* **Mã Mở và Miễn Phí:** Python là một ngôn ngữ mã nguồn mở và hoàn toàn miễn phí. Bạn có thể sử dụng, phân phối và sửa đổi mã nguồn Python mà không cần trả bất kỳ phí nào.

### **2.2. Nhược điểm**

* **Ghi Chú Chậm (Global Interpreter Lock - GIL):** GIL là một khóa tự động bên trong CPython (phiên bản Python chính thống) để đảm bảo rằng chỉ có một luồng được thực thi tại một thời điểm.
* **Khoản Thời Gian Khởi Động Chậm:** Ứng dụng Python thường có khoản thời gian khởi động chậm hơn so với các ngôn ngữ khác, do việc tải và biên dịch mã nguồn.
* **Kích Thước Ứng Dụng Lớn:** Ứng dụng Python có thể có kích thước lớn hơn so với các ứng dụng tương tự viết bằng C++ hoặc Go, do sự bao gồm của trình thông dịch Python và thư viện chuẩn.
* **Khó Sử Dụng cho Ứng Dụng Mobile:** Python không phải là ngôn ngữ phát triển chính cho ứng dụng di động và có hạn chế trong việc phát triển ứng dụng di động chuyên nghiệp.

# **CHƯƠNG II. FRAMEWORK TKINTER**

## **1. Widget**

|  |  |
| --- | --- |
| **Button** | Button được sử dụng để thêm nhiều nút khác nhau vào ứng dụng python. |
| **Canvas** | Canvas được sử dụng để vẽ các hình trên cửa sổ. |
| **Checkbutton** | Checkbutton được sử dụng để hiển thị CheckButton trên cửa sổ. |
| **Entry** | Entry được sử dụng để hiển thị trường văn bản một dòng cho người dùng. Nó thường được sử dụng để nhập các giá trị của người dùng. |
| **Frame** | Frame có thể được định nghĩa là một vùng chứa mà có thể chứa một hoặc nhiều widget khác. |
| **Label** | Label là một văn bản được sử dụng để hiển thị một số thông báo hoặc thông tin cho các widget khác. |
| **Listbox** | ListBox được sử dụng để hiển thị danh sách các tùy chọn cho người dùng. |
| **Menubutton** | Menubutton được sử dụng để hiển thị các mục menu cho người dùng. |
| **Menu** | Menu được sử dụng để thêm các mục menu cho người dùng. |
| **Message** | Message được sử dụng để hiển thị hộp tin nhắn cho người dùng. |
| **Radiobutton** | Người dùng được cung cấp các tùy chọn khác nhau và người dùng chỉ có thể chọn một tùy chọn trong số đó. |
| **Scale** | Nó được sử dụng để cung cấp thanh trượt cho người dùng. |
| **Scrollbar** | Nó cung cấp thanh cuộn cho người dùng để người dùng có thể cuộn cửa sổ lên và xuống. |
| **Text** | Nó khác với Entry vì nó cung cấp một trường văn bản nhiều dòng cho người dùng để người dùng có thể viết văn bản và chỉnh sửa văn bản bên trong nó. |
| **PanedWindow** | Nó giống như một widget container chứa các ô ngang hoặc dọc. |
| **LabelFrame** | LabelFrame là một widget vùng chứa hoạt động như một container. |
| **MessageBox** | Nó được sử dụng để hiển thị hộp thông báo trong các ứng dụng desktop. |

## **2. Phương thức bố cục**

### **Phương thức pack()**

Phương thức pack() được sử dụng để tổ chức widget theo khối.

Cú pháp : **widget.pack(options)**

Các tùy chọn ( options ) :

* **expand :** Nếu expand là true thì tiện ích con sẽ mở rộng để lấp đầy khoảng trống.
* **Fill :** Xác định xem widget con có lấp đầy bất kỳ không gian thừa nào do trình đóng gói phân bổ cho nó hay không hoặc giữ các kích thước tối thiểu của riêng nó: NONE(mặc định), X (chỉ điền theo chiều ngang), Y (chỉ điền theo chiều dọc) hay BOTH (điền theo cả chiều ngang và chiều dọc ).
* **side :** Nó giúp xác định vị trí của widget so với widget cha.

### **Phương thức grid()**

Trình quản lý layout grid() sắp xếp các widget ở dạng bảng. Chúng ta có thể chỉ định các hàng và cột.

Cú pháp : **widget.grid(options)**

Các tùy chọn ( options ) :

* **Column:** Số cột mà widget con sẽ được đặt. Cột ngoài cùng bên trái được biểu thị bằng 0.
* **Columnspan:** Chiều rộng của widget con. Nó đại diện cho số cột mà đến đó, cột được mở rộng.
* **ipadx, ipady:** Nó đại diện cho số lượng pixel cho đường viền của gidget.
* **padx, pady:** Nó đại diện cho số lượng pixel bên ngoài đường viền của gidget.
* **row:** Số hàng mà widget con sẽ được đặt. Hàng trên cùng được biểu thị bằng 0.
* **rowspan:** Chiều cao của widget con, tức là số hàng mà tiện ích con được mở rộng.
* **Sticky:** Nếu ô lớn hơn một widget, thì Sticky được sử dụng để chỉ định vị trí của widget bên trong ô. Nó có thể là sự ghép nối của các chữ cái sticky đại diện cho vị trí của widget. Nó có thể là N, E, W, S, NE, NW, NS, EW, ES.

### **Phương thức place()**

Trình quản lý layout place() sắp xếp các widget theo các tọa độ x và y.

Cú pháp : **widget.place(options)**

Các tùy chọn ( options ) :

* **Anchor:** Nó thể hiện vị trí chính xác của widget con trong container. Giá trị mặc định là NW (góc trên bên trái).
* **bordermode:** Giá trị mặc định của kiểu đường viền là INSIDE đề cập đến việc bỏ qua giá trị gốc bên trong đường viền. Tùy chọn còn lại là OUTSIDE.
* **height, width:** Nó đề cập đến chiều cao và chiều rộng của widget tính bằng pixel.
* **relheight, relwidth:** Chiều cao và chiều rộng có giá trị trong khoảng 0,0 và 1,0.
* **relx, rely:** Có giá trị trong khoảng 0,0 và 1,0, là độ lệch theo hướng ngang và dọc.
* **x, y:** Nó đề cập đến độ lệch ngang và dọc theo pixel.

## **3. Ưu - nhược điểm**

### **3.1. Ưu điểm**

* **Dễ Học và Sử Dụng:** Tkinter có cú pháp đơn giản và dễ đọc, điều này làm cho nó thích hợp cho người mới học lập trình GUI. Một số widgets cơ bản có thể được tạo và tùy chỉnh một cách dễ dàng.
* **Widgets Đa Dạng:** Tkinter cung cấp một loạt các widgets như cửa sổ, nút, hộp văn bản, danh sách, menu, và nhiều widgets khác để bạn có thể sử dụng trong ứng dụng của mình. Điều này giúp bạn xây dựng giao diện người dùng phong phú và đa dạng.
* **Tương Tác Tốt Với Python:** Tkinter tích hợp tốt với Python và cho phép bạn tạo ứng dụng có tính tương tác cao. Bạn có thể dễ dàng thực hiện xử lý sự kiện như bấm nút, nhập liệu từ bàn phím, và nhiều sự kiện khác.
* **Đa Nền Tảng:** Tkinter được tích hợp trong Python và hỗ trợ trên nhiều hệ điều hành, bao gồm Windows, macOS và Linux. Điều này đảm bảo rằng ứng dụng bạn phát triển có thể chạy trên nhiều nền tảng mà không cần thay đổi mã nguồn.

### **3.2. Nhược điểm**

* **Đa Nền Tảng :** Tkinter được tích hợp trong Python và hỗ trợ trên nhiều hệ điều hành, bao gồm Windows, macOS và Linux. Điều này đảm bảo rằng ứng dụng bạn phát triển có thể chạy trên nhiều nền tảng mà không cần thay đổi mã nguồn.
* **Hiệu Suất Hạn Chế :** Tkinter có thể không thích hợp cho các ứng dụng đòi hỏi hiệu suất cao, đặc biệt là khi có nhiều phần tử đồ họa hoặc xử lý số lớn. Tkinter có một số hạn chế về hiệu suất do việc sử dụng GIL (Global Interpreter Lock) của Python.
* **Khả Năng Tùy Chỉnh Khó Khăn** : Một số tùy chỉnh giao diện phức tạp hơn trong Tkinter so với các thư viện GUI khác, đặc biệt là trong việc tạo các phần tử giao diện tùy chỉnh.
* **Không Phù Hợp Cho Ứng Dụng Lớn và Phức Tạp :** Tkinter thường không được sử dụng cho các ứng dụng lớn và phức tạp như các ứng dụng đồ họa 3D hoặc ứng dụng chơi video game.

# **CHƯƠNG III. API**

## **1. Tổng quát**

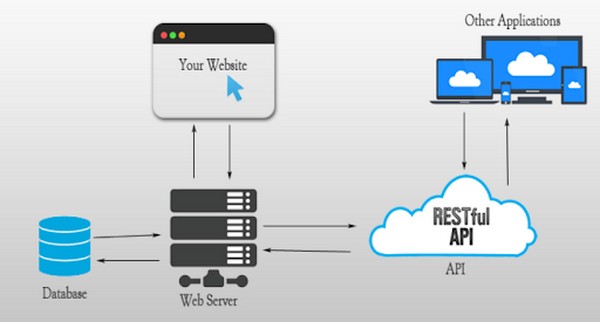
API được viết tắt bởi cụm từ tiếng Anh Application Programming Interface, tức giao diện lập trình ứng dụng. Đây là phương tiện cho hai hoặc nhiều ứng dụng trao đổi, tương tác với nhau, tạo ra tương tác giữa người dùng với ứng dụng hiệu quả và tiện lợi hơn.

Với API, các lập trình viên có thể tiếp cận, truy xuất dữ liệu từ máy chủ thể hiện chúng trên ứng dụng phần mềm hoặc website của mình một cách dễ dàng hơn.

Cơ chế API cho phép 2 phần mềm sử dụng các tập hợp giao thức hoặc tập hợp định nghĩa để giao tiếp với nhau. Bạn có thể hình dung định nghĩa này thông qua ví dụ trực quan về thời tiết. Chằng hạn như khi bạn nhận được các thông báo về thời tiết của ngày hôm nay thì có nghĩa là phần mềm dự báo thời tiết trên điện thoại của bạn đang dùng API để kết nối với cơ quan dự báo thời tiết trung ương để truyền đến bạn các tin tức đó.

## **2. Cách hoạt động**

Hiểu một cách đơn giản, API là giao diện cho phép ứng dụng này giao tiếp với các ứng dụng khác thông qua một hoặc nhiều câu lệnh khác nhau. Những lệnh này có thể được gửi, định dạng và truy xuất dữ liệu thông qua API khác với API SOAP hoặc REST, nhưng vẫn sẽ tuân thủ theo một số quy định và quy luật chung.



Có 4 cách để API hoạt động, nó phụ thuộc vào lý do và thời điểm tạo ra chúng :

* **API SOAP:** Đây là một trong những API “đời đầu”, nó dùng giao thức truy cập đối tượng đơn giản. Lúc này, máy chủ sẽ trao đổi thông điệp thông qua XML với máy khách.
* **API RPC:** Các API còn được biết đến với tên gọi là Lệnh gọi thủ tục từ xa. API này buộc máy chủ phải hoàn thành được một thủ tục bắt buộc thì máy khách mới có thể được nhận được kết quả.
* **API Websocket:** Đây là một phiên bản API web hiện đại chuyển dữ liệu nhờ vào việc sử dụng các đối tượng JSON. Máy khách và máy chủ có thể trao đổi thông tin 2 chiều nhờ vào API này.
* **API REST:** Hiện nay thì API REST được cho là linh hoạt nhất và cũng được sử dụng rộng rãi nhất. Với loại API này, máy khách sẽ gửi đến một yêu cầu dưới dạng dữ liệu. Sau đó máy chủ sẽ dùng các dữ liệu này để thực hiện các hàm nội bộ và cho ra các dữ liệu đầu ra rồi gửi lại cho máy khách.

Tên của một số phương thức API chính là các từ mô tả hành động của API. Một số hành động có thể xảy ra như :

* GET
* POST
* PUT
* DELETE

## **3. Các loại API**

#### **3.1. Dựa trên phân ngành nhỏ trong ngành công nghệ thông tin**

* **API trên nền tảng web:** còn có tên gọi khác là web API. Loại này hiện đang được sử dụng thông dụng nhất trong lập trình web hiện nay.
* **API trên hệ điều hành:** Khái niệm này còn xuất hiện trước cả API trên nền tảng web.
* **API của thư viện phần mềm hay framework:** Loại này mô tả và quy định các hành động mong muốn mà các thư viện cung cấp. Một API có thể có nhiều cách triển khai khác nhau.

#### **3.2. Dựa trên quyền hạn truy cập**

* **API mở (open API):** Là loại API công khai, không có hạn chế nào khi truy cập các loại API này vì chúng có sẵn công khai.
* **API đối tác (Partner API):** Bạn cần được cấp quyền hoặc giấy phép cụ thể khi muốn truy cập loại API này vì chúng không có sẵn công khai.
* **API nội bộ (Internal API):** Loại này còn được gọi là API riêng tư, chỉ những hệ thống nội bộ mới có thể sử dụng loại này.

## **4. Ưu - nhược điểm**

### **4.1. Ưu điểm**

* API được tích hợp trên hầu hết các ứng dụng của desktop, mobile và các ứng dụng trên website.
* Có sự linh hoạt trong việc định dạng dữ liệu khác nhau khi được trả về client.
* API là một công cụ có mã nguồn mở, việc này giúp bạn có thể sử dụng API mọi lúc bất kể đâu với điều kiện là phải có kết nối với internet. Giúp hỗ trợ chức năng RESTful.
* Hỗ trợ giao tiếp hai chiều và được xác nhận trong những giao dịch khác nhau. Điều này giúp đảm bảo được độ tin cậy cao.
* API hỗ trợ đầy đủ các thành phần MVC từ Model Binder, Action cho đến Unit Test,…

### **4.2. Nhược điểm**

* API chỉ hỗ trợ mặc định POST, GRT chứ chưa hoàn toàn như là RESTful service thông thường;
* Bạn cần có một kiến thức chuyên sâu thì việc sử dụng API mới thật sự hiệu quả;
* Để vận hành và phát triển API tốn rất nhiều thời gian cũng cũng như chi phí;
* Hệ thống bảo mật sẽ bị tấn công nếu chủ sở hữu chưa giới hạn kỹ các điều kiện trên hệ thống.

## **5. Tính bảo mật cho API**

**Một sốt vấn đề phổ biến nhất liên quan đến bảo mật API**

* **SQL Injection**

SQL Injection hay Injection là một lỗi nghiêm trọng khá thường gặp. Những kẻ tấn công mạng sẽ lợi dụng một số lỗ hổng của quy trình kiểm tra dữ liệu trong ứng dụng web để chuyển đến hệ thống quản lý cơ sở dữ liệu. Hướng khắc phục phổ biến nhất để thực hiện là dùng Regular Expression để loại bỏ các ký tự lạ, ký tự không phải số.

* **Spam request**

Các request ở chế độ công khai thường dễ gặp phải tình trạng spam. Chẳng hạn những trường hợp mà người dùng chỉ cần nhập tên và password để đăng ký tài khoản, có thể có hoặc không có bước xác thực. Cách khắc phục đơn giản nhất là làm những request của bạn trở nên phức tạp hơn.

**Giải pháp bảo mật hệ thống thông tin cho API**

* Hãy kiểm tra thẩm quyền người dùng, xác thực ứng dụng thật cẩn thận

Ngoài xác thực người sử dụng cuối cùng, bạn cũng cần chú ý đến quá trình xác thực ứng dụng.

* Mã hóa dữ liệu được truyền đi từ nền tảng

Sử dụng chứng chỉ SSL là giải pháp đơn giản nhất để bạn làm được việc này.

* Tránh dùng mật khẩu cố định, dạng nhúng hoặc quá dễ đoán

Vì lười biếng, nhiều người dùng thường lựa chọn mật khẩu cố định rất dễ đoán. Thậm chí họ còn sử dụng mật khẩu, thông tin được lưu cục bộ trên nhiều ứng dụng di động.

* Sử dụng chữ ký số

Vì lười biếng, nhiều người dùng thường lựa chọn mật khẩu cố định rất dễ đoán. Thậm chí họ còn sử dụng mật khẩu, thông tin được lưu cục bộ trên nhiều ứng dụng di động.

# **CHƯƠNG IV. ÁP DỤNG GUI FRAMEWORK VÀO ĐỀ TÀI**

**CÀI ĐẶT TRÊN WINDOWS**

### **1.1. Cài đặt môi trường**

* **Python 3.10**

Bạn download file cài đặt Python tại: [**https://www.python.org/downloads/windows/**](https://www.python.org/downloads/windows/)

* **PIP**

Phiên bản: Windows 7, Windows 8.1 và Windows 10.

* [get-pip.py](https://quantrimang.com/url?q=aHR0cHM6Ly9ib290c3RyYXAucHlwYS5pby9nZXQtcGlwLnB5)
* [get-pip.py (đối với Python **3.2)**](https://quantrimang.com/url?q=aHR0cHM6Ly9ib290c3RyYXAucHlwYS5pby8zLjIvZ2V0LXBpcC5weQ%3D%3D)

**Bước 1:** Tải xuống tập lệnh phía trên bằng cách nhấp chuột phải vào liên kết, chọn **Save As… (Lưu liên kết thành…)** vào lưu file vào thư mục bạn muốn.

**Bước 2:** Mở **Command Prompt** và điều hướng đến tệp get-pip.py.

**Bước 3:**Chạy lệnh sau:

python get-pip.py

* **PIL**

pip install Pillow

* **XAMPP**

**Bước 1:** Download XAMPP : <https://www.apachefriends.org/download.html>

**Bước 2:** Nhấp vào file có đuôi .exe trên file tải xuống.

**Bước 3:** Trên cửa sổ Set up, tích chọn các phần mềm mà bạn muốn cài đặt. Nếu bạn muốn cài WordPress trên XAMPP, các phần mềm bắt buộc phải chọn là MySQL, Apache, PHPMyAdmin. Sau khi chọn xong, nhấn Next.

**Bước 4:** Chọn thư mục cài đặt và nhấn Next.

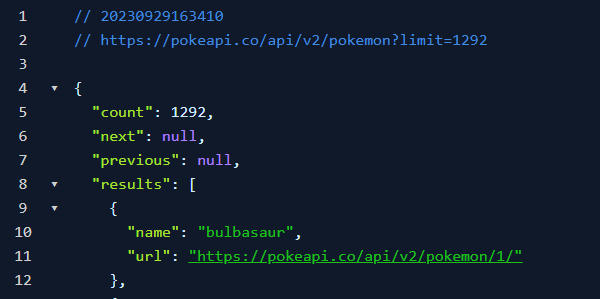
**Bước 5:** Chờ vài phút để cài đặt, sau khi cài đặt hoàn tất nhấn finish để kết thúc.

### **1.2. Thực hiện chức năng**

#### **1.2.1. API**

Đề tài sử dụng API của PokeAPI : https://pokeapi.co/api/v2/pokemon/ditto



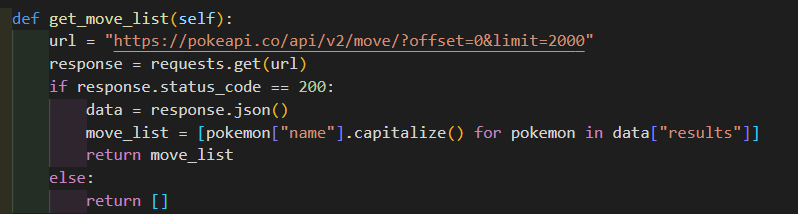
  
Trang API

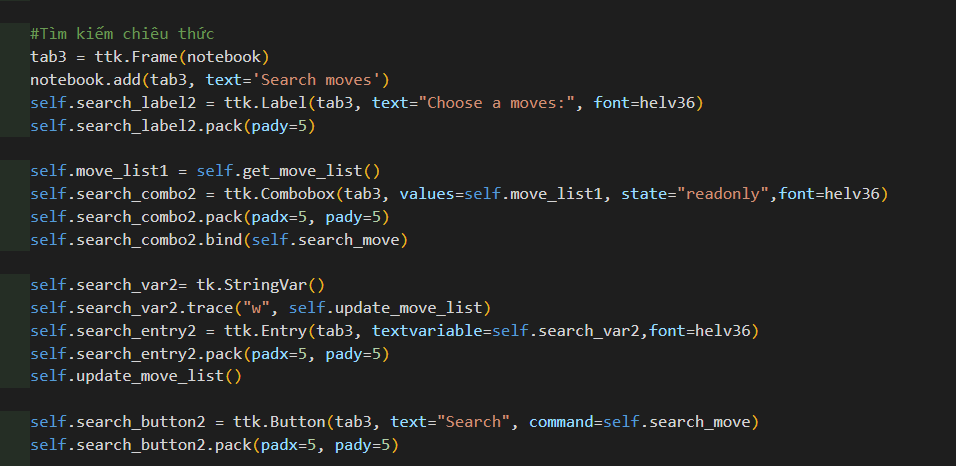
#### **1.2.2. Tkinter**

##### **Tìm kiếm kỹ năng**

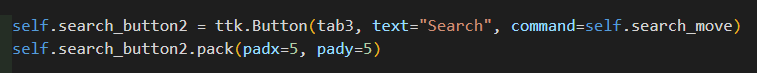
Thực hiện tìm kiếm kỹ năng cụ thể, cho biết được thông tin của kỹ năng đó như :

* **Type :** Loại kĩ năng.
* **Damage class :** Loại sát thương.
* **Accuracy :** Độ chính xác.
* **Power :** Sức mạnh.
* **PP :** Số lần sử dụng.
* **Effect :** hiệu ứng gây ra.

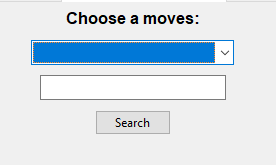
  
Trả kết quả các dữ liệu về move\_list( danh sách tất cả các kĩ năng trên API )

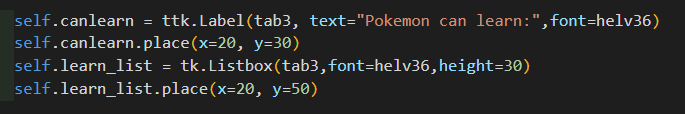


Giao diện combobox dựa trên kết quả từ search\_var2

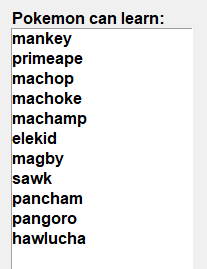


Thực hiện thao tác ấn nút tìm kiếm

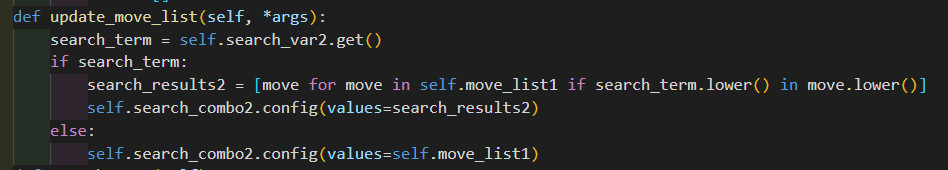




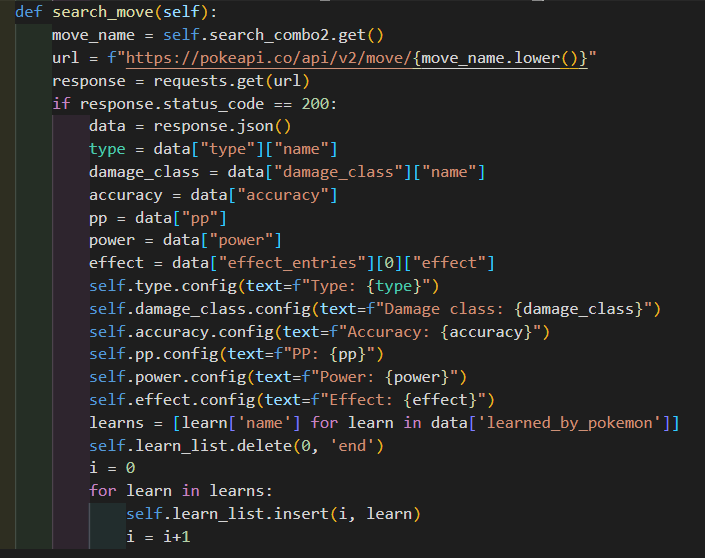
Hiển thị dữ liệu các Pokemon có thể học được kĩ năng này



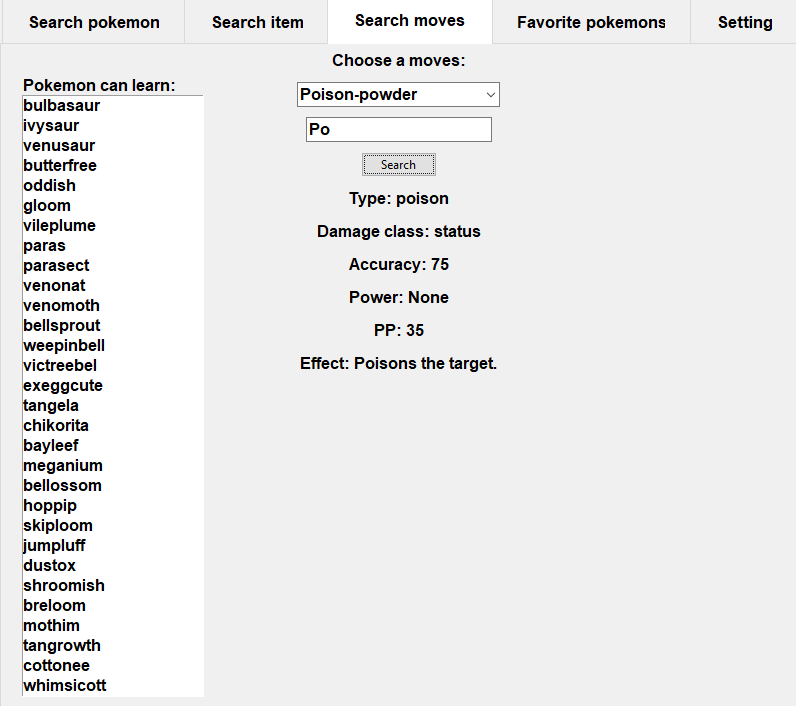
Giao diện sidebar hiển thị các Pokemon có thể học được kỹ năng này



Cập nhật lại danh sách các kĩ năng được lọc theo từ khóa đã nhập để hiển thị lên giao diện



Thực hiện xử lý dữ liệu đã lấy về được và hiển thị lên giao diện ứng dụng



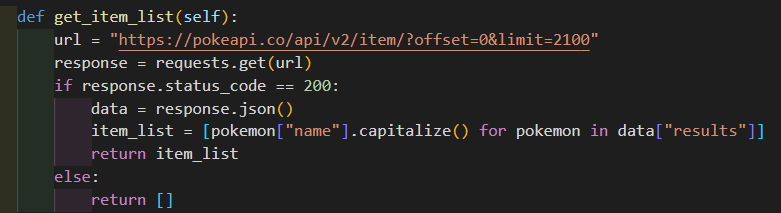
Giao diện tìm kiếm thành công một kĩ năng

##### **Tìm kiếm vật phẩm**

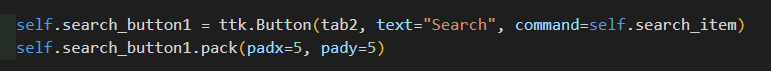
Giao diện thực hiện tìm các vật phẩm trong game chẳng hạn như quả cầu pokemon, vật phẩm tiêu thụ,…

Khi tìm kiếm thành công sẽ cho biết các thông tin cụ thể như :

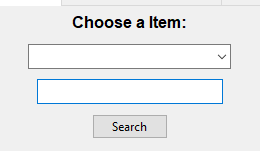
* **item name :** tên vật phẩm.
* **effect :** Hiệu ứng khi sử dụng.



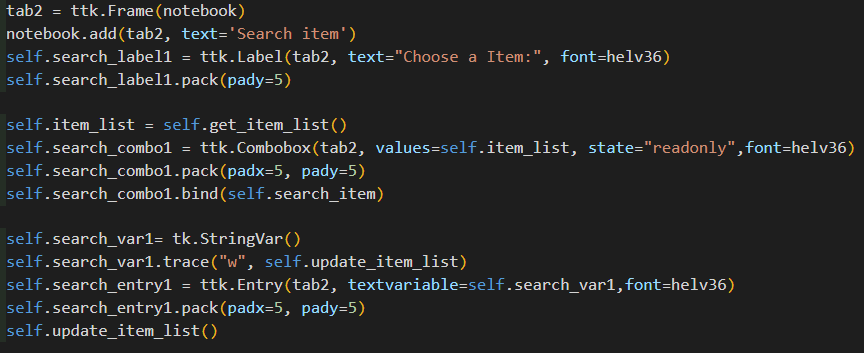
Trả kết quả các dữ liệu về item\_list( danh sách tất cả các vật phẩm trên API )



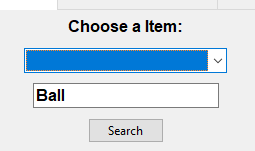
Thực hiện thao tác ấn nút tìm kiếm



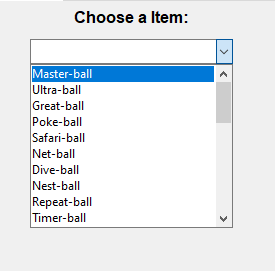
Khu vực tìm kiếm vật phẩm



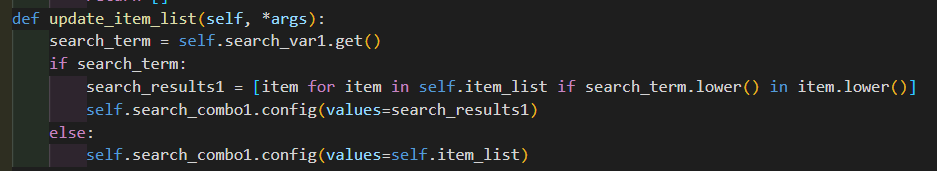
Giao diện combobox dựa trên kết quả từ search\_var2

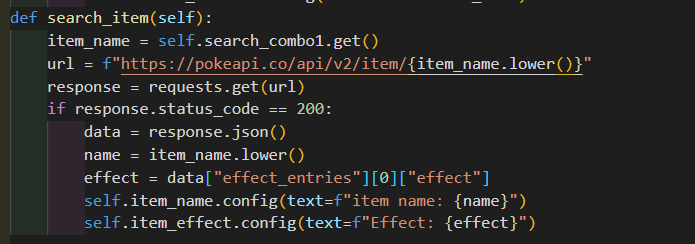


Khi thực hiện tìm kiếm bằng từ khóa nhập vào

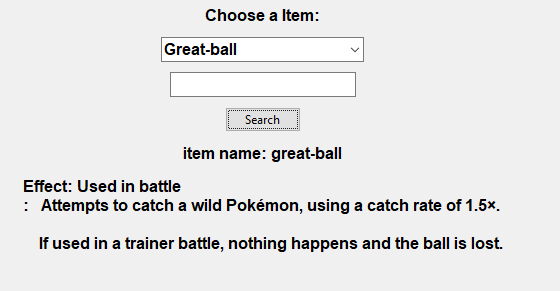


Hiển thị các vật phẩm có chứa từ khóa đã nhập

  
Cập nhật lại danh sách các vật phẩm được lọc theo từ khóa đã nhập để hiển thị lên giao diện



Thực hiện xử lý dữ liệu được lấy về từ API để hiển thị lên giao diện ứng dụng



Giao diện khi tìm kiếm thành công

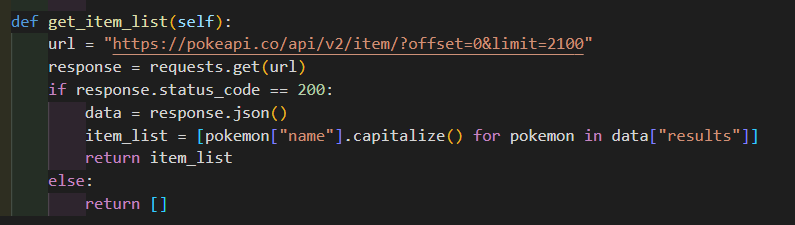
#### **1.2.3. PyQt5**

##### **Tìm kiếm vật phẩm**

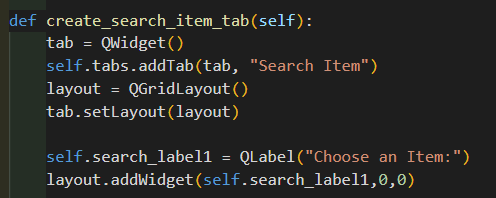
Giao diện thực hiện tìm các vật phẩm trong game chẳng hạn như quả cầu pokemon, vật phẩm tiêu thụ,…

Khi tìm kiếm thành công sẽ cho biết các thông tin cụ thể như :

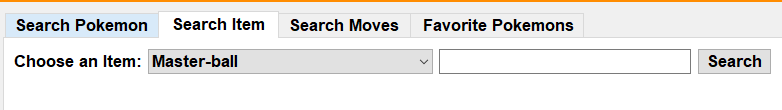
* **item name :** tên vật phẩm.
* **effect :** Hiệu ứng khi sử dụng.

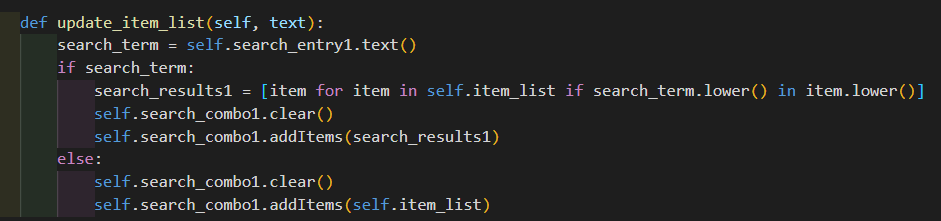


Trả kết quả các dữ liệu về item\_list( danh sách tất cả các vật phẩm trên API )

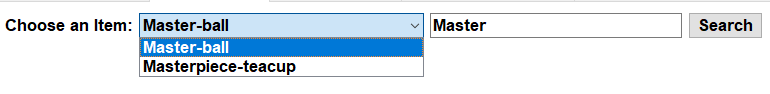


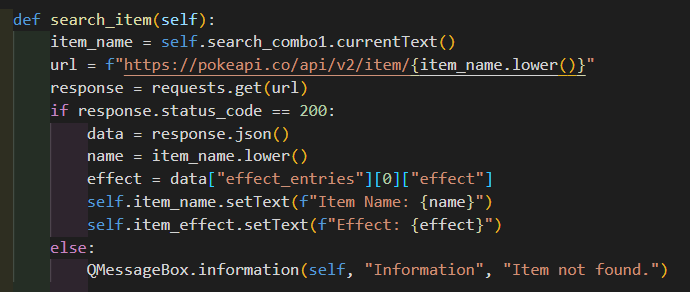
Tìm kiếm vật phẩm

  
Kết quả giao diện tìm kiếm vật phẩm

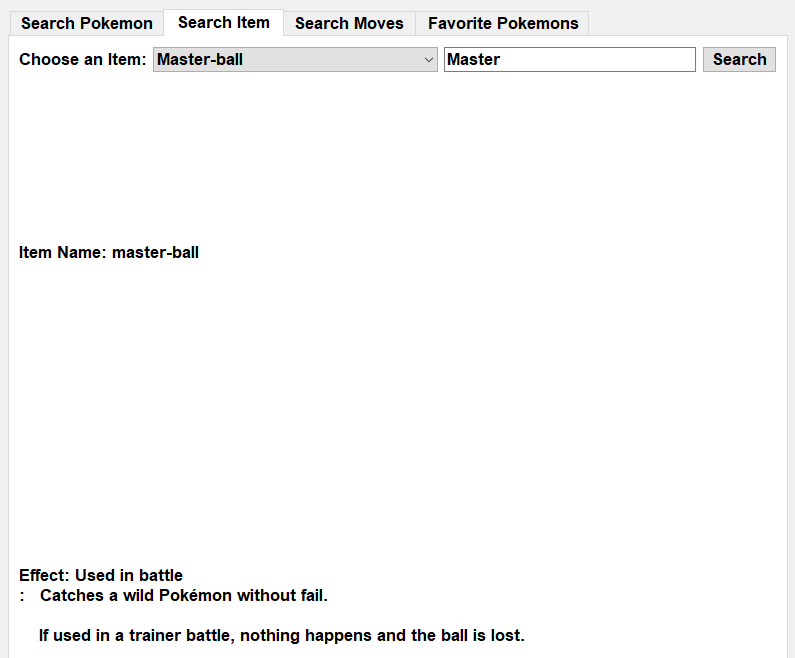


Khi thực hiện tìm từ khóa trên thanh tìm kiếm thì sẽ cập nhật trên combobox

  
Giao diện minh họa cho việc tìm kiếm theo tên



Lấy dữ liệu JSON từ API về và xử lý lên giao diện

  
Giao diện tìm kiếm vật phẩm

# **CHƯƠNG V. SO SÁNH GIỮA 2 GUI FRAMEWORK TKINTER VÀ PYQT5**

Cả Tkinter và PyQt đều là các khung GUI phổ biến cho Python. Cái nào tốt hơn phụ thuộc vào nhu cầu và sở thích cụ thể của bạn.

Sau khi thực hiện đề tài cài đặt các ứng dụng và thông qua cả nắm bắt phần khái niệm tổng quát của 2 GUI framework này, nhóm chúng em xin tóm tắt tổng quát về 2 framework này theo bảng sau đây :

|  |  |  |
| --- | --- | --- |
|  | **Tkinter** | **PyQt5** |
| **Thư viện và tích hợp** | Là một thư viện GUI mặc định đi kèm với Python. Không cần cài đặt bổ sung, nó sẵn có trong hầu hết các phiên bản Python. | Là một thư viện bên ngoài và cần được cài đặt trước. PyQt5 là một liên kết Python cho Qt, một khung làm việc đa nền tảng mạnh mẽ. |
| **Giao diện người dùng** | Giao diện người dùng Tkinter có thể trông đơn giản hơn so với PyQt5 và có ít tính năng mặc định. | PyQt5 cung cấp nhiều tính năng mạnh mẽ hơn và cho phép bạn tạo giao diện đẹp và phức tạp hơn với nhiều tùy chọn tùy chỉnh. |
| **Khả năng tùy chỉnh** | Tkinter cho phép bạn tùy chỉnh giao diện, nhưng các tùy chọn có thể hạn chế hơn so với PyQt5. | PyQt5 có khả năng tùy chỉnh cao hơn, với nhiều tùy chọn và tính năng tạo giao diện linh hoạt. |
| **Hiệu năng** | Tkinter có hiệu năng tốt cho các ứng dụng đơn giản. Tuy nhiên, hiệu năng có thể giảm đi khi ứng dụng trở nên phức tạp. | PyQt5 có hiệu năng tốt và ổn định, đặc biệt cho các ứng dụng đa nhiệm và phức tạp. |

**Chi tiết về các widget và phương thức giữa 2 framework**

|  |  |  |
| --- | --- | --- |
| **Button** | **Tkinter** | Tkinter cung cấp widget **Button** để tạo nút. Có thể tùy chỉnh màu sắc, kích thước và sự kiện click. |
| **PyQt5** | PyQt5 có widget **QPushButton** tương tự. Nó cũng hỗ trợ nhiều tính năng tùy chỉnh và biểu diễn nút. |
| **Label** | **Tkinter** | Tkinter sử dụng widget **Label** để hiển thị văn bản hoặc hình ảnh. Có thể tùy chỉnh font, màu sắc và vị trí. |
| **PyQt5** | PyQt5 cung cấp widget **QLabel**, với khả năng hiển thị văn bản và hình ảnh. Nó có nhiều tính năng tùy chỉnh như chế độ căn chỉnh và hiệu ứng. |
| **Entry** | **Tkinter** | Tkinter sử dụng widget **Entry** để nhập liệu văn bản. Có thể đặt giá trị mặc định và xử lý sự kiện. |
| **PyQt5** | PyQt5 có widget **QLineEdit**. Nó hỗ trợ nhiều tính năng như giới hạn độ dài, kiểm tra định dạng và các sự kiện liên quan đến việc nhập liệu. |
| **Frame** | **Tkinter** | Tkinter cung cấp widget **Frame** để nhóm và tổ chức các widget khác. |
| **PyQt5** | PyQt5 có **QFrame** tương tự với nhiều tính năng nhóm và vẽ đường viền. |
| **Pack /Place /Grid** | **Tkinter** | Tkinter sử dụng các phương thức **pack(), place(), và grid()** để quản lý việc bố trí các widget trên cửa sổ. |
| **PyQt5** | PyQt5 không có các phương thức tương tự. Thay vào đó, nó sử dụng quy trình layout manager mạnh mẽ để tự động bố trí các widget. |
| **Event Handling** | **Tkinter** | Tkinter sử dụng **bind()** để liên kết sự kiện với các hàm xử lý sự kiện. |
| **PyQt5** | PyQt5 sử dụng **connect()** để kết nối sự kiện với các hàm xử lý. Cũng hỗ trợ các decorator để đơn giản hóa việc xử lý sự kiện. |
| **Widget Styling** | **Tkinter** | Tkinter hỗ trợ tùy chỉnh giao diện thông qua các thuộc tính như **bg** (background), **fg** (foreground), **font**,... |
| **PyQt5** | PyQt5 sử dụng CSS để tùy chỉnh giao diện, cho phép bạn áp dụng kiểu dáng linh hoạt hơn. |
| **MessageBox và Dialogs** | **Tkinter** | Tkinter sử dụng **tkinter.messagebox,** **tkinter.simpledialog** để hiển thị hộp thoại thông báo và hộp thoại đơn giản. |
| **PyQt5** | PyQt5 sử dụng **QMessageBox** và các lớp khác như **QInputDialog** để tạo các hộp thoại thông báo và nhập liệu. |

Cuối cùng, lựa chọn giữa Tkinter và PyQt5 phụ thuộc vào mục tiêu của ứng dụng, trình độ kỹ năng của bạn, và yêu cầu cụ thể của dự án. Tkinter thích hợp cho các ứng dụng đơn giản và để làm quen với lập trình đồ họa, trong khi PyQt5 là một lựa chọn mạnh mẽ cho các ứng dụng phức tạp và yêu cầu tính tùy chỉnh cao hơn.