1. **Installation von erforderlichen Bibliotheken**

Das Programm verwendet Tkinter und Pillow (PIL)

pip install pillow, pip install tkinter im terminal eingeben

1. **Importierte Module im Code**

**import random**

* Funktion: Wird verwendet, um ein zufälliges Meme auszuwählen. Das heißt, bei jedem Klick auf "Zeige Meme" wird ein zufälliges Bild aus einem Ordner angezeigt**.**

**import os**

* Funktion: Wird verwendet, um mit dem Dateisystem zu interagieren. Damit kannst du die Dateipfade von Memes durchsuchen, sie öffnen und anzeigen.

**import tkinter as tk**

* Funktion: Wird verwendet, um die Benutzeroberfläche (GUI) zu erstellen. Tkinter ermöglicht es, Fenster, Buttons und Textfelder zu erstellen, die mit dem Benutzer interagieren**.**

**from PIL import Image, ImageTk**

* Funktion: PIL (Python Imaging Library) wird verwendet, um Bilder zu öffnen, zu bearbeiten und im richtigen Format (für Tkinter) anzuzeigen. Image öffnet und bearbeitet das Bild, während ImageTk es in ein Format umwandelt, das von Tkinter verstanden wird (für die Anzeige in der GUI).

**3. Vorbereitung der Meme-Dateien:**

Du hast mehrere Ordner auf deinem Computer, in denen Memes gespeichert sind. Diese Memes sind nach verschiedenen Kategorien unterteilt, z.B.:

**CATEGORIES**: Ein Dictionary

* Tiere (mit Unterkategorien wie Hunde, Katzen, Wildtiere)
* Emotionen (z.B. Glücklich, Traurig, Verwirrt)
* Gaming, IT, Kochen, Musik, usw.

Du hast diese Ordner organisiert und sicher gestellt, dass die Dateien korrekt benannt sind, damit sie später leichter gefunden werden können.

**4. Erstellen der Benutzeroberfläche (GUI):**

Du hast eine grafische Benutzeroberfläche (GUI) mit **Tkinter** erstellt. Tkinter ist eine Python-Bibliothek, die es dir ermöglicht, Fenster und Buttons zu erstellen, mit denen Benutzer interagieren können.

* Du hast ein Hauptfenster erstellt, das die Anwendung darstellt.
* Im Hauptfenster hast du ein Textfeld (für die **Suchfunktion**) und Buttons hinzugefügt. Die Buttons zeigen verschiedene **Kategorien** und **Unterkategorien** der Memes an.
* **root**: Das Hauptfenster der Tkinter-Anwendung.
* **search\_var und search\_entry**: Ein Textfeld, in das der Benutzer eine Suchanfrage eingeben kann, um nach Memes zu suchen.
* **search\_button**: Ein Button, der die Suchfunktion auslöst

**5. Erstellen der Meme-Kategorien:**

Du hast ein Dictionary erstellt, das alle Kategorien und Unterkategorien von Memes enthält. Zum Beispiel:

* **Tiere** -> Unterkategorien wie Hunde, Katzen, Vögel usw.
* **Emotionen** -> Unterkategorien wie Glücklich, Traurig, Verwirrt.

Dadurch wird es dem Programm möglich, schnell und einfach zu wissen, wo sich die Memes in deinem System befinden und diese später anzuzeigen.

**6. Funktionen im Detail:**

**show\_random\_meme:**

Diese Funktion zeigt ein zufällig ausgewähltes Meme aus einem gegebenen Ordner an.

* Wenn eine **Unterkategorie** übergeben wird, wird der entsprechende Ordner aus dem Dictionary CATEGORIES ausgewählt.
* **random.choice(memes)** wird verwendet, um zufällig ein Meme aus dem Ordner auszuwählen.
* Das Bild wird geöffnet, auf 500x500 Pixel skaliert und im Tkinter-Canvas angezeigt.

**show\_categories:**

Diese Funktion zeigt alle Hauptkategorien an (z.B. "Tiere", "Emotionen", "Gaming").

* Für jede Kategorie wird ein Button erzeugt. Wenn der Benutzer auf eine Kategorie klickt, wird die Funktion show\_subcategories aufgerufen (falls es Unterkategorien gibt) oder show\_random\_meme direkt ausgeführt.

**show\_subcategories:**

Diese Funktion zeigt die Unterkategorien für eine ausgewählte Hauptkategorie an.

* Für jede Unterkategorie wird ein Button erzeugt, der wiederum ein zufälliges Meme aus der gewählten Unterkategorie anzeigt.

**search\_memes:**

Die Funktion wird ausgeführt, wenn der Benutzer im Suchfeld nach einem Meme sucht.

* Die Eingabe wird in Kleinbuchstaben konvertiert, um die Suche zu erleichtern.
* Der Code durchläuft alle Kategorien und Unterkategorien, um nach Dateinamen zu suchen, die den Suchbegriff enthalten.
* Wenn Ergebnisse gefunden werden, werden sie als Buttons angezeigt, die das entsprechende Meme öffnen.

**show\_meme:**

Diese Funktion zeigt das Meme an, das der Benutzer durch die Suchergebnisse ausgewählt hat.

* Es prüft, ob die Datei existiert, und öffnet dann das Bild im Tkinter-kompatiblen Format.

**7. Meme zufällig auswählen:**

Wenn der Benutzer auf eine Kategorie klickt (z.B. **Tiere**), zeigt das Programm zufällig ein Meme aus dieser Kategorie an. Das passiert folgendermaßen:

* Das Programm schaut in den entsprechenden Ordner (z.B. "Tiere\Hunde").
* Es wählt zufällig eine Datei (ein Meme) aus diesem Ordner aus.
* Das Meme wird dann im Hauptfenster angezeigt.

**8. Unterkategorien anzeigen:**

Wenn du auf eine Hauptkategorie klickst (z.B. **Tiere**), zeigt das Programm dir zunächst alle **Unterkategorien** (wie Hunde, Katzen, Maus). Wenn du eine Unterkategorie auswählst (z.B. **Hunde**), wird das Programm ein zufälliges Meme aus diesem Bereich anzeigen.

**7.** **Zugriff auf Dateien:**

* Der Code verwendet **os.listdir()**, um die Dateien aus den Ordnern zu laden und nach Memes zu suchen.
* **os.path.join()** wird verwendet, um den vollständigen Pfad zu einem Meme zu erstellen

**9. Suchfunktion hinzufügen:**

Du hast auch eine **Suchfunktion** eingebaut. Damit kann der Benutzer nach einem bestimmten Meme suchen, indem er ein **Suchwort** in ein Textfeld eingibt (z.B. "Hund").

* Das Programm durchsucht dann alle Meme-Dateien, ob der Dateiname des Memes das Suchwort enthält.
* Wenn es passende Ergebnisse gibt, werden diese als **Buttons** angezeigt, und der Benutzer kann darauf klicken, um das Meme zu sehen.

**10. Meme-Anzeige:**

Wenn der Benutzer ein Meme auswählt, wird dieses Meme in einem speziellen Bereich der Benutzeroberfläche angezeigt. Du hast dafür **PIL (Pillow)** verwendet, um das Bild im richtigen Format anzuzeigen und es auf die passende Größe zu skalieren.

* **frame**: Ein Container, in dem die Buttons für die Kategorien und Suchergebnisse angezeigt werden.
* **meme\_label**: Ein Label, auf dem das ausgewählte Meme angezeigt wird.

**11. Zurück-Button:**

Du hast dafür gesorgt, dass der Benutzer jederzeit zum vorherigen Menü zurückkehren kann, indem er auf einen **Zurück-Button** klickt. Dieser Button bringt den Benutzer entweder zu den Unterkategorien oder zurück zu den Hauptkategorien.

**12. Fehlerbehandlung:**

Du hast auch dafür gesorgt, dass das Programm dem Benutzer eine klare Fehlermeldung gibt, falls etwas schiefgeht:

* Wenn keine Memes in einem Ordner gefunden werden.
* Wenn keine Suchergebnisse zu einem bestimmten Suchbegriff vorhanden sind.
* Wenn ein Meme nicht mehr existiert oder gelöscht wurde.

**13. Zusammenfassung des Arbeitsablaufs:**

* Der Benutzer öffnet die Anwendung und sieht ein Hauptmenü mit Kategorien.
* Der Benutzer klickt auf eine Kategorie oder Unterkategorie, um ein zufälliges Meme anzusehen.
* Der Benutzer kann auch nach einem Meme suchen, indem er ein Stichwort eingibt.
* Gefundene Memes werden als Buttons angezeigt. Wenn der Benutzer auf einen Button klickt, wird das Meme angezeigt.
* Der Benutzer kann jederzeit zum vorherigen Bildschirm zurückkehren, um eine andere Kategorie oder Unterkategorie auszuwählen.