

```
#ショートカットランチャーアプリ

#①必要なライブラリのインポート処理
from tkinter import Tk, Button
import subprocess

#②アプリケーションを開いた時の動作
def open_application(path):
    subprocess.Popen(path) # 指定されたパスのアプリを開く
    root.destroy()        # アプリケーションを開いたときにアプリを終了する

#③メインウィンドウ(root)の設定
root = Tk()                # tkinterでウィンドウの作成
root.title("Launcher")     # タイトル名
root.geometry("300x150")   # windowサイズ

#④ショートカット設定 変更する場合 ボタン名 + パスで変更可
shortcut = {
    "GoogleCrome": "C:\\Program Files\\Google\\Chrome\\Application\\chrome.exe",
    "メモ帳": "C:\\Windows\\notepad.exe",
    "コマンドプロンプト": "C:\\WINDOWS\\system32\\conhost.exe",
    "コントロールパネル": "C:\\WINDOWS\\system32\\control.exe",
    "エクスプローラー": "C:\\Windows\\explorer.exe",
    "vscode": "C:\\Users\\7d03\\AppData\\Local\\Programs\\Microsoft VS
Code\\Code.exe"
}

#⑤ボタン設定
row = 1 # 行
col = 0 # 列
#shortcutに入っている各項目についてnameを取り出して繰り返す
for name, path in shortcut.items():
    # lambdaに7行目のopen_application関数を代入
    btn = Button(root, text=name, command=lambda p=path: open_application(p))
    # 指定位置にボタンを配置
    btn.grid(column=col, row=row, sticky="NSEW")
    # 3行の表示なので3行目以上になった場合次の2列目に移行
    col += 1
    if col > 2: # もし3列より大きいなら
        col = 0 # カラム位置の初期配置に
        row += 1 # +1して2列目に移行

#⑥各ボタンがwindowサイズに応じて形状変化するように設定
root.grid_columnconfigure(0, weight=1)
root.grid_columnconfigure(1, weight=1)
root.grid_columnconfigure(2, weight=1)
root.grid_rowconfigure(0, weight=0)
root.grid_rowconfigure(1, weight=1)
root.grid_rowconfigure(2, weight=1)

#⑦勝手に終了しないようにループさせる
root.mainloop()
```

コード解説

①必要なライブラリのインポート処理 from tkinter import Tk, Button: tkinterモジュールからTkクラスとButtonクラスをインポートします。 import subprocess: サブプロセスを実行するためのモジュールをインポートします。

・tkinter pythonに標準で付属しているGUIライブラリ ・ subprocess Pythonプログラムから他のプログラムやシステムコマンドを実行するために必要になる。

②アプリケーションを開いた時の動作 def open_application(path):: open_application関数を定義。この関数は、指定されたパスのアプリを開くために使用されます。 subprocess.Popen(path): subprocessモジュールを使用して、指定されたパスのアプリを開きます。 root.destroy(): アプリケーションを開いたときにアプリを終了します。

③メインウィンドウ(root)の設定 root = Tk(): Tkクラスのインスタンスを作成し、メインウィンドウ(root)を定義します。 root.title("Launcher"): ウィンドウのタイトルの設定。 root.geometry("300x150"): ウィンドウのサイズの設定。

④ショートカット設定 変更する場合 ボタン名+パスで変更可 shortcut = { ... }: ショートカットの名前とパスを辞書型で定義。

⑤ボタン設定 for name, path in shortcut.items(): shortcut辞書の各要素について、名前とパスを取得します。 btn = Button(root, text=name, command=lambda p=path: open_application(p)): 各アプリのショートカットのボタンを作成し、クリックされたときに対応するアプリを開くように設定します。 btn.grid(column=col, row=row, sticky="NSEW"): グリッドレイアウトを使用してボタンを指定位置に配置します。 root.grid_columnconfigure(0, weight=1): ウィンドウの列のサイズが変更されたときにウィジェットがウィンドウに対してどのように拡大/縮小されるかを設定します。

⑥各ボタンがwindowサイズに応じて形状変化するように設定 root.grid_rowconfigure(1, weight=1): ウィンドウの行のサイズが変更されたときにウィジェットがウィンドウに対してどのように拡大/縮小されるかを設定します。

⑦勝手に終了しないようにループさせる root.mainloop(): イベントループを開始し、ウィンドウを表示します。