

Setup database

import random

def create\_username(first\_name, last\_name):

"""

Tạo username gồm chữ cái đầu của tên và họ, viết thường

"""

username = first\_name[0] + last\_name

username = username.lower()

return username

def create\_password():

"""

Tạo mật khẩu gồm 4 chữ số ngẫu nhiên

"""

password = str(random.randint(1000, 9999))

return password

def create\_database(customers):

"""

Tạo một cơ sở dữ liệu dưới dạng từ điển với username làm khóa và password làm giá trị

Tham số:

customers : list

Danh sách các danh sách con, mỗi danh sách con chứa tên và họ của khách hàng

Giá trị trả về:

db : dict

Cơ sở dữ liệu (từ điển) username -> password

n\_customers : int

Số lượng khách hàng trong cơ sở dữ liệu

"""

db = {}

# Lặp qua từng khách hàng trong danh sách customers

for khach in customers:

# Tạo username từ tên và họ

username = create\_username(khach[0], khach[1])

# Tạo password ngẫu nhiên

password = create\_password()

# Lưu username và password vào cơ sở dữ liệu

db[username] = password

n\_customers = len(db)

return db, n\_customers

chạy hàm

{

"cells": [

{

"cell\_type": "code",

"execution\_count": 4,

"metadata": {},

"outputs": [

{

"name": "stdout",

"output\_type": "stream",

"text": [

"1.\n",

"jgelb\n",

"\n",

"2.\n",

"jgelb\n",

"\n",

"3.\n",

"jgelb\n",

"\n",

"4.\n",

"jgelb\n",

"\n",

"5.\n",

"4695\n",

"\n",

"6.\n",

"{'mlopez': '4965', 'jsmith': '7208', 'mseid': '9799'}\n",

"Tong so khach hang: 3\n"

]

}

],

"source": [

"print(\"1.\")\n",

"import setup\_database\n",

"\n",

"username = setup\_database.create\_username(\"John\", \"Gelb\")\n",

"print(username)\n",

"\n",

"\n",

"print(\"\\n2.\")\n",

"import setup\_database as sdb\n",

"\n",

"username = sdb.create\_username(\"John\", \"Gelb\")\n",

"print(username)\n",

"\n",

"\n",

"print(\"\\n3.\")\n",

"from setup\_database import create\_username\n",

"\n",

"username = create\_username(\"John\", \"Gelb\")\n",

"print(username)\n",

"\n",

"\n",

"print(\"\\n4.\")\n",

"from setup\_database import create\_username as cu\n",

"\n",

"# gọi trực tiếp hàm đó ra và cho nó tên \"cu\"\n",

"username = cu(\"John\", \"Gelb\")\n",

"print(username)\n",

"\n",

"\n",

"print(\"\\n5.\")\n",

"import setup\_database\n",

"\n",

"password = setup\_database.create\_password()\n",

"print(password)\n",

"\n",

"\n",

"print(\"\\n6.\")\n",

"import setup\_database\n",

"\n",

"customers = [[\"Maria\", \"Lopez\"], [\"Julia\", \"Smith\"], [\"Mohammed\", \"Seid\"]]\n",

"\n",

"db, n = setup\_database.create\_database(customers)\n",

"\n",

"print(db)\n",

"print(\"Tong so khach hang: \", n)\n"

]

}

],

"metadata": {

"kernelspec": {

"display\_name": "base",

"language": "python",

"name": "python3"

},

"language\_info": {

"codemirror\_mode": {

"name": "ipython",

"version": 3

},

"file\_extension": ".py",

"mimetype": "text/x-python",

"name": "python",

"nbconvert\_exporter": "python",

"pygments\_lexer": "ipython3",

"version": "3.13.5"

}

},

"nbformat": 4,

"nbformat\_minor": 2

}

print("33. Quà sinh nhật")

print("1. Đọc tệp .txt")

def read\_txt(file\_name\_in):

"""

Đọc một tệp .txt có mỗi dòng là một số

và trả về danh sách các số đó.

"""

numbers = []

try:

with open(file\_name\_in, "r") as file:

for line in file:

line = line.strip() # Xóa khoảng trắng và ký tự xuống dòng

if line: # Bỏ qua dòng trống

try:

numbers.append(float(line))

except ValueError:

print(f"Bỏ qua dòng không hợp lệ: '{line}'")

except FileNotFoundError:

print(f" Không tìm thấy tệp '{file\_name\_in}'.")

return numbers

purchases = read\_txt("purchases.txt")

print(purchases)

print("\n2. Phân tích các số")

def calculate\_stats(numbers):

"""

Tính giá trị nhỏ nhất, lớn nhất và tổng của danh sách số.

"""

if not numbers:

return None, None, 0 # Tránh lỗi nếu danh sách rỗng

min\_val = min(numbers)

max\_val = max(numbers)

total = sum(numbers)

return min\_val, max\_val, total

# --- Gọi hàm và in kết quả ---

mi, ma, to = calculate\_stats(purchases)

print("Giá trị nhỏ nhất:", mi)

print("Giá trị lớn nhất:", ma)

print("Giá trị tổng:", to)

print("\n3. Lưu phân tích")

def write\_txt(file\_name\_out, min\_val, max\_val, total):

"""

Ghi kết quả thống kê (min, max, sum) ra tệp văn bản.

"""

with open(file\_name\_out, "w", encoding="utf-8") as file:

file.write(f"Giá trị nhỏ nhất: {min\_val}\n")

file.write(f"Giá trị lớn nhất: {max\_val}\n")

file.write(f"Giá trị tổng: {total}\n")

print(f" Đã ghi kết quả vào tệp '{file\_name\_out}' thành công!")

write\_txt("purchases\_out.txt", mi, ma, to)