# Gerekli Python modülleri içe aktarılıyor

import os # Dosya ve klasör işlemleri için temel modül (genel amaçlı)

import shutil # Dosya taşıma, kopyalama gibi işlemler için kullanılır

from pathlib import Path # Modern, okunabilir ve güvenli dosya yolu yönetimi sağlar

# ------------------------------------------------------------

# DOSYA KATEGORİLERİNİN TANIMLANMASI

# ------------------------------------------------------------

# Burada bir sözlük (dictionary) oluşturuluyor.

# Her anahtar bir klasör adını (kategori adını),

# her değer ise o kategoriye ait dosya uzantılarının listesini temsil ediyor.

# Böylece hangi dosya tipinin hangi klasöre taşınacağını önceden tanımlıyoruz.

DOSYA\_KATEGORILERI = {

"Dokumanlar": [".docx", ".doc", ".pdf", ".txt", ".md"], # Belgeler

"Tablolar": [".xlsx", ".xls", ".csv"], # Elektronik tablolar

"Sunumlar": [".pptx", ".ppt"], # Sunum dosyaları

"Resimler": [".png", ".jpg", ".jpeg", ".gif"], # Görseller

"Loglar": [".log"], # Kayıt dosyaları

"Kodlar": [".py", ".java", ".cpp", ".c", ".js", ".html", ".css"], # Kaynak kod dosyaları

"Arsivler": [".zip", ".rar", ".tar", ".gz"], # Sıkıştırılmış dosyalar

}

# ------------------------------------------------------------

# FONKSİYON: klasor\_duzenle()

# ------------------------------------------------------------

def klasor\_duzenle(hedef\_klasor):

"""

Bu fonksiyon, belirtilen klasörün içindeki dosyaları tarar

ve her dosyayı uzantısına göre uygun alt klasöre taşır.

- Eğer dosya uzantısı tanımlıysa, o kategoriye taşınır.

- Eğer hiçbir kategoriyle eşleşmiyorsa, 'Diger' adlı klasöre taşınır.

Amaç: Dağınık bir klasörü, dosya türlerine göre düzenlemek.

"""

# Path sınıfı kullanılarak hedef klasörün yolu temsil edilir

yol = Path(hedef\_klasor)

# Eğer belirtilen klasör yoksa, kullanıcı uyarılır ve işlem durdurulur

if not yol.exists():

print(f"❌ {hedef\_klasor} klasörü bulunamadı.")

return

# --------------------------------------------------------

# KLASÖRDEKİ TÜM ELEMANLARI (dosya veya klasörleri) DOLAŞ

# --------------------------------------------------------

for dosya in yol.iterdir():

# Sadece dosyalar üzerinde işlem yapılır, klasörler atlanır

if dosya.is\_file():

tasindi\_mi = False # Bu değişken, dosya bir kategoriye taşındı mı diye kontrol için kullanılır

# --------------------------------------------------------

# Her kategori ve uzantı listesi üzerinden kontrol yapılır

# --------------------------------------------------------

for kategori, uzantilar in DOSYA\_KATEGORILERI.items():

# Eğer dosyanın uzantısı (suffix) bu kategoride tanımlıysa

if dosya.suffix.lower() in uzantilar:

# O kategoriye ait klasör yolu oluşturulur (örneğin .../Resimler)

kategori\_klasoru = yol / kategori

# Eğer bu klasör yoksa, oluşturulur (exist\_ok=True varsa hata vermez)

kategori\_klasoru.mkdir(exist\_ok=True)

# Dosya, bulunduğu yerden yeni klasöre taşınır

shutil.move(str(dosya), kategori\_klasoru / dosya.name)

# Bilgilendirme mesajı ekrana yazdırılır

print(f"📂 {dosya.name} → {kategori}/ klasörüne taşındı")

# Dosya taşındı, o yüzden bu kontrol True olur

tasindi\_mi = True

break # Artık diğer kategorilere bakmaya gerek yok

# --------------------------------------------------------

# Eğer hiçbir kategoriyle eşleşmediyse

# dosya "Diger" adlı klasöre taşınır

# --------------------------------------------------------

if not tasindi\_mi:

diger\_klasoru = yol / "Diger"

diger\_klasoru.mkdir(exist\_ok=True)

shutil.move(str(dosya), diger\_klasoru / dosya.name)

print(f"📂 {dosya.name} → Diger/ klasörüne taşındı")

# ------------------------------------------------------------

# ANA ÇALIŞMA BLOĞU

# ------------------------------------------------------------

if \_\_name\_\_ == "\_\_main\_\_":

# Program doğrudan çalıştırıldığında bu blok devreye girer.

# (Başka bir Python dosyasına modül olarak dahil edilirse çalışmaz.)

# Düzenlenecek klasörün yolu burada belirtilir.

# Bu örnekte masaüstü klasörü hedef olarak seçilmiştir.

hedef = r"C:\Users\ibrah\Desktop\"

# Fonksiyon çağrılarak dosya düzenleme işlemi başlatılır

klasor\_duzenle(hedef)

# İşlem tamamlandığında kullanıcıya bilgi verilir

print("✅ Klasör düzenleme tamamdır!")