# Gerekli modüller içe aktarılıyor

import os # Dosya yolları, yeniden adlandırma gibi işlemler için

from pathlib import Path

import re # Regular expression (düzenli ifade) işlemleri için — metin temizleme, desen arama vb.

def isim\_temizle(dosya\_adi):

"""

Bu fonksiyon, bir dosya adını standart bir forma dönüştürür.

Amacı, dağınık ve düzensiz dosya adlarını profesyonel, okunabilir ve tutarlı hale getirmektir.

Fonksiyon şu işlemleri yapar:

1. Tüm harfleri küçültür. (Büyük/küçük harf farkını ortadan kaldırır)

2. Boşluk karakterlerini alt çizgi (\_) ile değiştirir.

3. 'final', 'kopya', 'taslak' gibi gereksiz kelimeleri siler.

4. Art arda gelen birden fazla alt çizgiyi teke düşürür.

5. Başta veya sonda kalan alt çizgileri temizler.

"""

# Dosya adını (adı + uzantı) olarak ikiye ayırır

# Örnek: "Rapor Taslak.docx" → ad = "Rapor Taslak", uzanti = ".docx"

ad, uzanti = os.path.splitext(dosya\_adi)

# Ad kısmındaki tüm karakterleri küçük harfe dönüştür

ad = ad.lower()

# Tüm boşlukları alt çizgi ile değiştir

# r"\s+" → bir veya birden fazla boşluk karakteri anlamına gelir

ad = re.sub(r"\s+", "\_", ad)

# Dosya adında geçen "final", "kopya" veya "taslak" kelimelerini tamamen kaldır

# örnek: "rapor\_final.docx" → "rapor.docx"

ad = re.sub(r"(final|kopya|taslak)+", "", ad)

# Art arda gelen fazla alt çizgileri teke düşür

# örnek: "rapor\_\_son" → "rapor\_son"

ad = re.sub(r"\_+", "\_", ad)

# Başta veya sonda kalan gereksiz alt çizgileri temizle

# örnek: "\_rapor\_taslak\_" → "rapor\_taslak"

ad = ad.strip("\_")

# Temizlenmiş dosya adını, uzantısı küçük harfe çevrilmiş haliyle geri döndür

return ad + uzanti.lower()

def toplu\_yeniden\_adlandir(klasor\_yolu):

"""

Bu fonksiyon, verilen klasörün içindeki tüm dosyaları tek tek tarar

ve her birinin adını isim\_temizle() fonksiyonundan geçirerek yeniden adlandırır.

Yani klasördeki dosya adlarını topluca düzenler.

"""

# Path nesnesiyle klasörü temsil et

klasor = Path(klasor\_yolu)

# Eğer klasör gerçekten mevcut değilse kullanıcıyı uyar ve işlemi sonlandır

if not klasor.exists():

print(f"❌ {klasor\_yolu} klasörü bulunamadı.")

return

# Klasörün içindeki her dosya veya alt klasörü sırayla dolaş

for dosya in klasor.iterdir():

# Sadece dosyalar üzerinde işlem yap (klasörleri atla)

if dosya.is\_file():

# Dosyanın yeni, temizlenmiş ismini oluştur

yeni\_isim = isim\_temizle(dosya.name)

# Yeni dosya yolunu oluştur (aynı klasör içinde)

yeni\_yol = klasor / yeni\_isim

# Eğer dosya adında değişiklik gerekiyorsa (yani aynı değilse)

if dosya.name != yeni\_isim:

# Dosyayı yeni adıyla yeniden adlandır

dosya.rename(yeni\_yol)

# Ekrana bilgi mesajı ver: hangi dosya nasıl değişti

print(f"🔄 {dosya.name} → {yeni\_isim} olarak değiştirildi")

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

# ANA ÇALIŞMA BLOĞU

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

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

# Program doğrudan çalıştırıldığında bu blok devreye girer.

# (Bir modül olarak başka bir dosyada içe aktarılırsa bu kısım çalışmaz.)

# Düzenlenecek klasör yolu burada tanımlanır.

# Örneğin masaüstündeki dosyalar üzerinde işlem yapılacaktır.

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

# Fonksiyon çağrılarak, hedef klasördeki dosyalar yeniden adlandırılır.

toplu\_yeniden\_adlandir(hedef\_klasor)

# İşlem tamamlandığında kullanıcıya bilgi mesajı verilir.

print("✅ Toplu yeniden adlandırma tamamdır!")