

تحليل السكريبت exportbookmarkstourl.py وخطة تطويره السكريبت يحول ملف HTML للمفضلة إلى مجموعة من ملفات url . في مجلد محدد. الميزات الحالية: تنظيم احترافي: الكود منظم في دوال مستقلة. دعم الطرفية: يستخدم argparse لقبول المسار من سطر الأوامر. تسجيل الأحداث: يسجل العمليات في ملف exportlog.txt. استخدام BeautifulSoup: لاستخراج الروابط من HTML. نقاط القوة: توثيق جيد باستخدام docstrings. إعدادات مجمعة في بداية السكريبت. التعامل مع الأخطاء. خطة التوسعة المقترحة: 1. تصنيف حسب المجلدات: استخراج أسماء المجلدات من HTML. 2. تقرير إحصائي: txt يحتوي على: عدد الروابط لكل مجلد. أكثر النطاقات تكراراً. استبعاد الروابط غير المرغوبة (فارغة، 5. تصدير إلى CSV أو JSON: حفظ قائمة المفضلات في ملف bookmarks.csv أو bookmarks. يتضمن الاسم، الرابط، والمجلد. اقتراحات إضافية: إزالة التكرار: التحقق من وجود الملف قبل إنشائه. حفظ وقت التصدير: إضافة طابع زمني. إعدادات خارجية: قراءة المسارات من ملف config.json. ضغط النتائج: إنشاء ملف ZIP. مثال لمخلص النتائج: 535 رابط معطل أو غير صالح إجمالي 1010 رابط تم فحصهم خيلنا نبسطها ونضيف ملاحظات تذكيرة فحسب اريد جزء التفاعل مع المستخدم يكن بالانجليزي نظرا لعدم دعم بورشيل وسي ام دي للعربي ايضا استخدام ترميز صحيح لدعم العربي التعليقات بالعربي داخل السكريبت للمطور اضافة وصف دقيق للسكريبت وماذا يفعل لتفادي النسيان والتساؤل عن دي سكريبت ايه ممكن نستخدم علامة الاقتباس الثلاثية التي تدعم التعامل مع اسطر لان شكل الشباك مش حلو في حال كان الوصف طويل تصدير النتائج في سجل لوج عرض ملخص مختصر للمستخدم في الطرفية او بورشيل مثلا تم نسخ عدد 3 ملف وتجاهل 2 وعدم وجود 4 مع التركيز على التنظيم، التوثيق، سهولة الصيانة، والتعامل مع الأخطاء. النقاط الرئيسية: التوثيق (Documentation): استخدام سلاسل نصية ثلاثية (لوصف وظيفة السكريبت، المدخلات/المخرجات، ومكان الإعدادات. الإعدادات (Configuration): تجميع المتغيرات الهامة (المسارات، أسماء الملفات) في بداية السكريبت أو في ملفات خارجية (JSON/TXT) لتسهيل التعديل. التعامل مع الأخطاء (Error Handling): استخدام try/except لتوقع ومعالجة الأخطاء المحتملة، مع توفير رسائل خطأ واضحة. تفاعل المستخدم (User Interaction): استخدام اللغة الإنجليزية في رسائل الطرفية، وتقديم ملخص سريع للعمليات المنجزة. التسجيل (Logging): إنشاء ملف log. تنظيم الكود (Code Structure): تقسيم الكود إلى دوال (functions) لتحسين القراءة وإعادة الاستخدام. إنشاء مجلدات الواجهة تلقائياً. إرسال إشعارات بعد الانتهاء. دعم خيار --help باستخدام argparse. الأفكار المتكررة/المدمجة: تم دمج نقاط الوصف والتوثيق في قسم واحد. تم تجميع نقاط الإخراج وتفاعل المستخدم في قسم واحد يركز على التواصل الفعال. تم دمج نقاط الترميز والتعليقات في قسم واحد يتعلق بوضوح الكود. التركيز على اللغة العربية في التعليقات مفيد للمطورين العرب. الأمثلة العملية (خاصة مثال بايثون) تعزز الفهم. التأكيد على التعامل مع الأخطاء والسيناريوهات المتوقعة يعكس التفكير الاحترافي. تحليل محتوى حول كتابة سكريبت احترافي الموضوع الرئيسي: تقديم إرشادات ومعايير لكتابة سكريبت احترافية، مع التركيز على التنظيم، التوثيق، سهولة الصيانة، والتعامل مع الأخطاء. النقاط الرئيسية: التوثيق (Documentation): استخدام سلاسل نصية ثلاثية (لوصف وظيفة السكريبت، المدخلات/المخرجات، ومكان الإعدادات. الإعدادات (Configuration): تجميع المتغيرات الهامة (المسارات، أسماء الملفات) في بداية السكريبت أو في ملفات خارجية (JSON/TXT) لتسهيل التعديل. التعامل مع الأخطاء (Error Handling): استخدام try/except لتوقع ومعالجة الأخطاء المحتملة، مع توفير رسائل خطأ واضحة. التسجيل (Logging): إنشاء ملف log.txt لتسجيل تفاصيل العمليات، تنظيم الكود (Code Structure): تقسيم الكود إلى دوال (functions) لتحسين القراءة وإعادة الاستخدام. إنشاء مجلدات الواجهة تلقائياً. التقييم: الأمثلة العملية (خاصة مثال بايثون) تعزز الفهم. التأكيد على التعامل مع الأخطاء والسيناريوهات المتوقعة يعكس التفكير الاحترافي. تلخيص الدردشة فسكريبت وهمي لتطبيق كل النقاط في الدردشة سألخص الدردشة في سكريبت وهمي يجمع كل النقاط التي ناقشناها. هذا السكريبت سيكون بمثابة نموذج تطبيقي للمبادئ التي تحدثنا عنها، والتعامل الاحترافي مع الإدخال والإخراج. --- مع تعليقات توضيحية لكل جزء. """ import json import argparse # ضع جميع المتغيرات الهامة هنا لسهولة التعديل DESTINATION\_PATH = "C:\\Users\\User\\My Documents\\Destination" txt # مثال على استخدام ملف JSON لبيانات خارجية "r", "json" load(f) ALLOWED\_FILE\_TYPES = config\_data.get("file\_types", []) # التعامل مع سيناريو عدم وجود الملف print("Warning: config. Using default file types. ALLOWED\_FILE\_TYPES = [". txt", ". def log\_event(message, log\_path = os.path.join(os.datetime.strftime("%Y-%m-%d %H:%M:%S") with

