**Creating a Directory If It Doesn’t Exist**

Zanim zaczniemy robić cokolwiek z paragonami, chcemy mieć przetworzony katalog, aby je przenieść, abyśmy nie próbowali przetwarzać tego samego paragonu dwa razy. Nasz skrypt może być wystarczająco inteligentny, aby utworzyć ten katalog dla nas, jeśli nie istnieje, gdy uruchamiamy skrypt po raz pierwszy. Użyjemy funkcji os.mkdir; jeśli katalog już istnieje, możemy złapać zgłoszony błąd OSE:

import os

try:

os.mkdir("./processed")

except OSError:

print("'processed' directory already exists")

**Zbieranie wpływów do przetworzenia**

Z powłoki możemy zbierać pliki na podstawie wzorców i to jest przydatne. Dla naszych celów chcemy uzyskać każdy pokwitowanie z nowego katalogu, który pasuje do tego wzoru:

receipt-[0-9]\*.json

Ten wzór przekłada się na paragon-, po którym następuje dowolna liczba cyfr, a kończąc na rozszerzenie pliku .json. Dokładny wynik możemy osiągnąć za pomocą funkcji glob.glob.

receipts = glob.glob('./new/receipt-[0-9]\*.json')

subtotal = 0.0

Część przetwarzania paragonów będzie wymagać zsumowania wszystkich wartości, więc uruchomimy nasz skrypt z sumą częściową wynoszącą 0,0.

**Reading JSON**

Pozostała część naszego skryptu wymagać będzie od nas:

1. Iteruj rachunki

2. Odczytanie JSON każdego paragonu

3. Sumowanie wartości wpływów

4. Przeniesienie każdego pliku potwierdzenia do przetworzonego katalogu po jego zakończeniu

Użyliśmy funkcji json.dump do zapisania pliku JSON i możemy użyć przeciwnej funkcji json.load, aby odczytać plik JSON. Zawartość pliku zostanie przekształcona w słownik, do którego możemy uzyskać dostęp do jego kluczy. Dodamy wartość do podsumy przed ostatecznym przeniesieniem pliku za pomocą shutil.move. Oto nasz ostatni skrypt:

import glob

import os

import shutil

import json

try:

os.mkdir("./processed")

except OSError:

print("'processed' directory already exists")

# Get a list of receipts

receipts = glob.glob('./new/receipt-[0-9]\*.json')

subtotal = 0.0

for path in receipts:

with open(path) as f:

content = json.load(f)

subtotal += float(content['value'])

name = path.split('/')[-1]

destination = f"./processed/{name}"

shutil.move(path, destination)

print(f"moved '{path}' to '{destination}'")

print("Receipt subtotal: $%.2f" % subtotal)

Dodajmy pliki, które nie pasują do naszego wzorca, do nowego katalogu przed uruchomieniem naszego skryptu:

touch new/receipt-other.json new/receipt-14.txt new/random.txt

Wreszcie uruchommy nasz skrypt dwa razy i zobaczmy, co otrzymamy:

$ python3.6 process\_receipts.py

moved './new/receipt-0.json' to './processed/receipt-0.json'

moved './new/receipt-1.json' to './processed/receipt-1.json'

moved './new/receipt-2.json' to './processed/receipt-2.json'

moved './new/receipt-3.json' to './processed/receipt-3.json'

moved './new/receipt-4.json' to './processed/receipt-4.json'

moved './new/receipt-5.json' to './processed/receipt-5.json'

moved './new/receipt-6.json' to './processed/receipt-6.json'

moved './new/receipt-7.json' to './processed/receipt-7.json'

moved './new/receipt-8.json' to './processed/receipt-8.json'

moved './new/receipt-9.json' to './processed/receipt-9.json'

Receipt subtotal: $6932.04

$ python3.6 process\_receipts.py

'processed' directory already exists

Receipt subtotal: $0.00