cleanup\_files.py

*"""  
CP1404/CP5632 Practical  
Demos of various os module examples  
"""*import shutil  
import os  
from os import path  
import re  
  
  
def main():  
 print("Starting directory is: {}".format(os.getcwd()))  
  
 os.chdir('Lyrics/Christmas')  
  
 print("Files in {}:\n{}\n".format(os.getcwd(), os.listdir('.')))  
  
 if path.exists("temp"):  
 pass  
 else:  
 os.mkdir('temp')  
  
 for filename in os.listdir('.'):  
 if os.path.isdir(filename):  
 continue  
 else:  
 new\_name = get\_fixed\_filename(filename)  
 print("Renaming {} to {}".format(filename, new\_name))  
  
  
def get\_fixed\_filename(filename):  
 *"""Return a 'fixed' version of filename."""* return\_name = ""  
 # print("Given Name: {}".format(filename))  
 new\_name = re.split('(?=[A-Z])', filename.replace(".TXT", ".txt"))  
 for word in new\_name:  
 return\_name = return\_name + " " + word.title()  
 # print("Camel Name: {}".format(return\_name))  
 new\_name = return\_name.replace(" ", "\_").replace("\_\_", "\_").replace("(\_", "(").replace(".Txt", ".txt")  
 new\_name = new\_name[1:]  
 return new\_name  
  
  
main()

os\_demos.py

*"""  
CP1404/CP5632 Practical  
Demos of various os module examples  
"""*import shutil  
import os  
from os import path  
  
def main():  
 *"""Demo os module functions."""* print("Starting directory is: {}".format(os.getcwd()))  
  
 # Change to desired directory  
 os.chdir('Lyrics/Christmas')  
  
 # Print a list of all files in current directory  
 print("Files in {}:\n{}\n".format(os.getcwd(), os.listdir('.')))  
  
 # Make a new directory  
 # The next time you run this, it will crash if the directory exists  
 if path.exists("temp"):  
 pass  
 else:  
 os.mkdir('temp')  
  
 # Loop through each file in the (current) directory  
 for filename in os.listdir('.'):  
 # Ignore directories, just process files  
 if os.path.isdir(filename):  
 continue  
  
 new\_name = get\_fixed\_filename(filename)  
 print("Renaming {} to {}".format(filename, new\_name))  
  
 # Option 1: rename file to new name - in place  
 # os.rename(filename, new\_name)  
  
 # Option 2: move file to new place, with new name  
 # shutil.move(filename, 'temp/' + new\_name)  
  
  
def get\_fixed\_filename(filename):  
 *"""Return a 'fixed' version of filename."""* new\_name = filename.replace(" ", "\_").replace(".TXT", ".txt")  
 return new\_name  
  
  
def demo\_walk():  
 *"""Process all subdirectories using os.walk()."""* os.chdir('Lyrics')  
 for directory\_name, subdirectories, filenames in os.walk('.'):  
 print("Directory:", directory\_name)  
 print("\tcontains subdirectories:", subdirectories)  
 print("\tand files:", filenames)  
 print("(Current working directory is: {})".format(os.getcwd()))  
 # os.rename(filenames, "NewName")  
  
  
# main()  
demo\_walk()

sort\_files\_1.py

import os  
import shutil  
from os import path  
  
  
def main():  
 os.chdir('FilesToSort')  
 list\_of\_dirs = ['xlsx', 'xls', 'txt', 'png', 'jpg', 'gif', 'docx', 'doc']  
 for direct in list\_of\_dirs:  
 if path.exists(direct):  
 pass  
 else:  
 os.mkdir(direct)  
 for filename in os.listdir('.'):  
 if os.path.isdir(filename):  
 continue  
 else:  
 for direct in list\_of\_dirs:  
 if filename[-3:] == direct[-3:]:  
 shutil.move(filename, direct)  
 # or os.rename() (but more typing expensive)  
  
  
main()

sort\_files\_2.py

import os  
import shutil  
from os import path  
  
  
def main():  
 os.chdir('FilesToSort')  
 list\_of\_dirs = ['xlsx', 'xls', 'txt', 'png', 'jpg', 'gif', 'docx', 'doc']  
 user\_responses = []  
 for direct in list\_of\_dirs:  
 user\_responses.append(input("What category would you like to sort {} files into?".format(direct)))  
 for direct\_path in list(dict.fromkeys(user\_responses)):  
 if path.exists(direct\_path):  
 pass  
 else:  
 os.mkdir(direct\_path)  
 for filename in os.listdir('.'):  
 for direct in list\_of\_dirs:  
 if filename[-3:] == direct[-3:]:  
 shutil.move(filename, user\_responses[list\_of\_dirs.index(direct)])  
  
  
main()