os\_demos.py

*"""  
CP1404/CP5632 Practical  
Demos of various os module examples  
"""***import** shutil  
**import** os  
  
  
**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  
 #* ***TODO: Use exception handling to avoid the crash (just pass)* try**:  
 os.mkdir(**'temp'**)  
 **except** FileExistsError:  
 **pass** *# 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))  
  
 *#* ***TODO: Try these options one at a time*** *# 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()))  
  
 *#* ***TODO: add a loop to rename the files* for** filename **in** filenames:  
 initial\_name = os.path.join(directory\_name, filename)  
 new\_name = os.path.join(directory\_name, get\_fixed\_filename(filename))  
 os.rename(initial\_name, new\_name)  
  
  
*# main()*demo\_walk()

cleanup\_files.py

*"""  
CP1404/CP5632 Practical  
Demos of various os module examples  
"""***import** shutil  
**import** os  
  
  
**def** main():  
 *"""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()))  
  
 *#* ***TODO: add a loop to rename the files* for** filename **in** filenames:  
 print(filename)  
 print(**"Rename {} to {}"**.format(filename, get\_fixed\_filename(filename)))  
 initial\_name = os.path.join(directory\_name, filename)  
 new\_name = os.path.join(directory\_name, get\_fixed\_filename(filename))  
 os.rename(initial\_name, new\_name)  
  
  
**def** get\_fixed\_filename(filename):  
 *"""Return a 'fixed' version of filename."""* **for** i **in** range(len(filename)-1):  
 **if** filename[i] == **' '**:  
 filename[i + 1].upper() *#upper the first letter of word* **if** filename[i].islower() **and** filename[i + 1].isupper():  
 filename = filename[:i + 1] + **'\_'** + filename[i + 1:] *# add '\_' between 2 words* new\_name = filename.replace(**" "**, **"\_"**).replace(**".TXT"**, **".txt"**) *# add '\_' between 2 words* **return** new\_name  
  
  
main()

sort\_files\_1.py

*"""  
sort these files from FilesToSort into subdirectories for each extension.  
"""***import** os  
**import** shutil  
**def** main():  
 *"""move files into subfolders with the same name as their extension."""* os.chdir(**"FilesToSort"**)  
 **for** filename **in** os.listdir(**'.'**):  
 **if** os.path.isdir(filename):  
 **continue** file\_extension=filename.split(**'.'**)[-1] *#split the file name and its extension by the '.'* **try**:  
 os.mkdir(file\_extension)  
 **except** FileExistsError: *#if the extension dir have been made already* **pass** shutil.move(filename, **'{}/{}'**.format(file\_extension,filename))  
  
main()

sort\_files\_2.py

*"""  
Let the user categorise different extensions as the program encounters these, then move them all into those subdirectories.  
"""***import** os  
**import** shutil  
  
  
**def** main():  
 *"""move files into subfolders where users want to put them on based on the extension."""* extension\_subfolders = {}  
 os.chdir(**"FilesToSort"**)  
 **for** filename **in** os.listdir(**'.'**):  
 **if** os.path.isdir(filename):  
 **continue** file\_extension = filename.split(**'.'**)[-1] *# split the file name and its extension by the '.'* **if** file\_extension **not in** extension\_subfolders.keys():  
 extension\_category = input(**"What category would you like to sort {} files into? "**.format(file\_extension))  
 extension\_subfolders[  
 file\_extension] = extension\_category *# add the extension\_category as a value and file\_extension as a key on extension\_subfolder* **try**:  
 os.mkdir(extension\_category)  
 **except** FileExistsError: *# if the extension category dir have been made already* **pass** shutil.move(filename, **'{}/{}'**.format(extension\_subfolders[file\_extension], filename))  
  
  
main()