Python Directory and Folder Operations

Fan Wang

2020-05-24

Contents

1 Folder Operations

Go to the **RMD**, **PDF**, or **HTML** version of this file. Go back to fan's Python Code Examples Repository (bookdown site).

1.1 New Folder and Files

1. create a folder and subfolder
2. create two files in the new folder

```
import pathlib

# folder root
srt_folder = "_folder/"

# new folder
srt_subfolder = srt_folder + "fa/"
# new subfolder
srt_subfolder = srt_subfolder + "faa/"
# generate folders recursively
pathlib.Path(srt_subfolder).mkdir(parents=True, exist_ok=True)

# Open new file
fl_tex_contents_aa = open(srt_subfolder + "file_a.txt", 'w')
# Write to File
fl_tex_contents_aa.write('contents of file a')
```

```
## 18
```

```
fl_tex_contents_aa.close()

# Open another new file and save
```

```
fl_tex_contents_ab = open(srt_subfolder + "file_b.txt", 'w')
# Write to File
fl_tex_contents_ab.write('contents of file b')
```

```
## 18
```

```
fl_tex_contents_ab.close()
```

Generate more folders without files:

```
# generate folders recursively
pathlib.Path("_folder/fb/fba/").mkdir(parents=True, exist_ok=True)
# generate folders recursively
pathlib.Path("_folder/fc/").mkdir(parents=True, exist_ok=True)
# generate folders recursively
pathlib.Path("_folder/fd/").mkdir(parents=True, exist_ok=True)
```

1.2 Copy a File from One Folder to Another

Move the two files from *_folder/fa/faa* to *_folder/faa* as well as to *_folder/fb/faa. *Use* shutil.copy2* so that more metadata is copied over. But *copyfile* is faster.

- How do I copy a file in Python?

Moving one file:

```
import shutil
# Faster method
shutil.copyfile('_folder/fa/faa/file_a.txt', '_folder/fb/file_a.txt')
# More metadat copied, and don't need to specify name
```

```
## '_folder/fb/file_a.txt'
```

```
shutil.copy2('_folder/fa/faa/file_a.txt', '_folder/fb/fba')
```

```
## '_folder/fb/fba\\file_a.txt'
```

1.3 Copy Folder to Multiple Destimations

Move Entire Folder, How do I copy an entire directory of files into an existing directory using Python?:

```
from distutils.dir_util import copy_tree

# Move contents from fa/faa/ to fc/faa
srt_curroot = '_folder/fa/'
srt_folder = 'faa/'
srt_newroot = '_folder/fc/'

# Full source and destination
srt_sourc = srt_curroot + srt_folder
srt_desct = srt_newroot + srt_folder

# Check/Create new Directory
pathlib.Path(srt_desct).mkdir(parents=True, exist_ok=True)

# Move
copy_tree(srt_sourc, srt_desct)
```

```
## ['_folder/fc/faa/file_a.txt', '_folder/fc/faa/file_b.txt']
```

Move contents to multiple destinations:

```
from distutils.dir_util import copy_tree
# Check/Create new Directory
pathlib.Path('_folder/fd/faa/fa_images').mkdir(parents=True, exist_ok=True)
pathlib.Path('_folder/fd/faa/fb_images').mkdir(parents=True, exist_ok=True)
pathlib.Path('_folder/fd/faa/fc_images').mkdir(parents=True, exist_ok=True)
pathlib.Path('_folder/fd/faa/fz_img').mkdir(parents=True, exist_ok=True)
pathlib.Path('_folder/fd/faa/fz_other').mkdir(parents=True, exist_ok=True)

# Move
copy_tree('_folder/fa/faa/', '_folder/fd/faa/fa_images')
```

```
## ['_folder/fd/faa/fa_images\\file_a.txt', '_folder/fd/faa/fa_images\\file_b.txt']
```

```
copy_tree('_folder/fa/faa/', '_folder/fd/faa/fb_images')
```

```
## ['_folder/fd/faa/fb_images\\file_a.txt', '_folder/fd/faa/fb_images\\file_b.txt']
```

```
copy_tree('_folder/fa/faa/', '_folder/fd/faa/fc_images')
```

```
## ['_folder/fd/faa/fc_images\\file_a.txt', '_folder/fd/faa/fc_images\\file_b.txt']
```

```
copy_tree('_folder/fa/faa/', '_folder/fd/faa/fz_img')
```

```
## ['_folder/fd/faa/fz_img\\file_a.txt', '_folder/fd/faa/fz_img\\file_b.txt']
```

```
copy_tree('_folder/fa/faa/', '_folder/fd/faa/fz_other')
# Empty Folder
```

```
## ['_folder/fd/faa/fz_other\\file_a.txt', '_folder/fd/faa/fz_other\\file_b.txt']
```

```
pathlib.Path('_folder/fd/faa/fd_images').mkdir(parents=True, exist_ok=True)
pathlib.Path('_folder/fd/faa/fe_images').mkdir(parents=True, exist_ok=True)
```

1.4 Search for Files in Folder

Find the total number of files in a folder.

```
import pathlib
# the number of files in folder found with search critiera
st_fle_search = '*.txt'
ls_spn = [Path(spn).stem for spn in Path('_folder/fd/faa/fa_images').rglob(st_fle_search)]
print(ls_spn)

# count files in a non-empty folder
```

```
## ['file_a', 'file_b']
```

```
srn = '_folder/fd/faa/fa_images'
[spn for spn in Path(srn).rglob(st_fle_search)]
```

```
## [WindowsPath('_folder/fd/faa/fa_images/file_a.txt'), WindowsPath('_folder/fd/faa/fa_images/file_b.tx
```

```
bl_folder_is_empty = len([spn for spn in Path(srn).rglob(st_fle_search)])>0
print(bl_folder_is_empty)

# count files in an empty folder
```

```
## True
```

```
srn = '_folder/fd/faa/fd_images'
[spn for spn in Path(srn).rglob(st_fle_search)]
```

```
## []
```

```
bl_folder_is_empty = len([spn for spn in Path(srn).rglob(st_fle_search)])>0
print(bl_folder_is_empty)
```

```
## False
```

1.5 Search for Folder Names

- python search for folders containing strings

Search for folders with certain search word in folder name, and only keep if folder actually has files.

```
import os

# get all folder names in folder
ls_spt = os.listdir('_folder/fd/faa/')
print(ls_spt)

# Select only subfolder names containing _images
```

```
## ['fa_images', 'fb_images', 'fc_images', 'fd_images', 'fe_images', 'fz_img', 'fz_other', '_img']
```

```
srt = '_folder/fd/faa/'
st_search = '_images'
ls_srt_found = [srt + spt
                for spt in os.listdir(srt)
                if st_search in spt]
print(ls_srt_found)
```

```
## ['_folder/fd/faa/fa_images', '_folder/fd/faa/fb_images', '_folder/fd/faa/fc_images', '_folder/fd/faa
```

1.6 Find Non-empty Folders by Name

Search:

1. Get subfolders in folder with string in name
2. Only collect if there are files in the subfolder

```
import pathlib

# Select only subfolder names containing _images
srt = '_folder/fd/faa/'
# the folder names must contain _images
st_srt_srh = '_images'
# there must be files in the folder with this string
st_fle_srh = '*.txt'

# All folders that have String
ls_srt_found = [srt + spt
                for spt in os.listdir(srt)
                if st_srt_srh in spt]
print(ls_srt_found)
```

```
# All folders that have String and that are nonempty
```

```
## ['_folder/fd/faa/fa_images', '_folder/fd/faa/fb_images', '_folder/fd/faa/fc_images', '_folder/fd/faa/
```

```
ls_srt_found = [srt + spt
                for spt in os.listdir(srt)
                if ((st_srt_srh in spt)
                    and
                    (len([spn for spn
                          in Path(srt + spt).rglob(st_fle_srh)])>0)) ]
print(ls_srt_found)
```

```
## ['_folder/fd/faa/fa_images', '_folder/fd/faa/fb_images', '_folder/fd/faa/fc_images']
```

1.7 Found Folders to new Folder

1. Search for subfolders by folder name string in a folder
2. Select nonempty subfolders
3. Move nonsempty subfolders to one new folder
4. Move this single combination folder

The results here are implemented as function in the pyfan package: fp_agg_move_subfiles.

```
import pathlib
import os
import shutil
from distutils.dir_util import copy_tree

# 1 Define Parameters

# Select only subfolder names containing _images
srt = '_folder/fd/faa/'
# the folder names must contain _images
st_srt_srh = '_images'
# there must be files in the folder with this string
st_fle_srh = '*.txt'

# new aggregating folder name
srt_agg = '_img'

# folders to move aggregation files towards
ls_srt_dest = ['_folder/fd/faa/', '_folder/']

# delete source
bl_delete_source = False

# 2 Gather Folders
ls_ls_srt_found = [[srt + spt, spt]
                   for spt in os.listdir(srt)
                   if ((st_srt_srh in spt)
                       and
                       (len([spn for spn
                             in Path(srt + spt).rglob(st_fle_srh)])>0)) ]
print(ls_ls_srt_found)
```

```
# 3 Loop over destination folders, loop over source folders
```

```
## [['_folder/fd/faa/fa_images', 'fa_images'], ['_folder/fd/faa/fb_images', 'fb_images'], ['_folder/fd/
```

```
for srt in ls_srt_dest:

  # Move each folder over
  for ls_srt_found in ls_ls_srt_found:

    # Paths
    srt_source = ls_srt_found[0]
    srt_dest = os.path.join(srt, srt_agg, ls_srt_found[1])

    # dest folders
    pathlib.Path(srt_dest).mkdir(parents=True, exist_ok=True)

    # move
    copy_tree(ls_srt_found[0], srt_dest)

# 4. Delete Sources
```

```
## ['_folder/fd/faa/_img\\fa_images\\file_a.txt', '_folder/fd/faa/_img\\fa_images\\file_b.txt']
## ['_folder/fd/faa/_img\\fb_images\\file_a.txt', '_folder/fd/faa/_img\\fb_images\\file_b.txt']
## ['_folder/fd/faa/_img\\fc_images\\file_a.txt', '_folder/fd/faa/_img\\fc_images\\file_b.txt']
## ['_folder/_img\\fa_images\\file_a.txt', '_folder/_img\\fa_images\\file_b.txt']
## ['_folder/_img\\fb_images\\file_a.txt', '_folder/_img\\fb_images\\file_b.txt']
## ['_folder/_img\\fc_images\\file_a.txt', '_folder/_img\\fc_images\\file_b.txt']
```

```
if bl_delete_source:
  for ls_srt_found in ls_ls_srt_found:
    shutil.rmtree(ls_srt_found[0])
```