Python Searching for Programs, Reading and Writing to File Examples

Fan Wang

2020-05-09

Contents

1 Check, Read, Write and Convert Files

Go to the **RMD**, **PDF**, or **HTML** version of this file. Go back to Python Code Examples Repository (bookdown site) or the pyfan Package (API).

1.1 Check if where a Program is.

Suppose we want to generate a commandline call from within python and want to start it in bash. Calling something like "C:/Program Files/Git/git-bash.exe -c COMMANDS". However, depending on the computer that is been used, the git-bash command might be in different spots. How do we find the right path to the *git-bash* file. Accomplish this by using *shutil.which*, which can find the path to different commands, including the *git* command. Given the path that we found, the *git-bash.exe* file is stored in the *Git* folder, two levels up. So we will use pathlib to get to the correct path location. To be safe, go up one level, and then two levels to search for *git-bash.exe*.

First, find the path to the git exe command:

```
# Imports
import os
import shutil
# cmd
st_cmd = 'git'
# Using shutil.which() method to find local path to the *git* command
spn_path_to_git = shutil.which(st_cmd)
# Print result
print(f'{spn_path_to_git=}')
```

```
## spn_path_to_git='G:\\ProgramData\\Git\\cmd\\git.EXE'
```

Second, find the parent and grandparent folders:

```
from pathlib import Path
# Get the parent folder 2 levels up
```

```
srt_path_git_parent_folder = Path(spn_path_to_git).parents[0]
srt_path_git_grandparent_folder = Path(spn_path_to_git).parents[1]
# Print
print(f'{srt_path_git_parent_folder=}')
```

```
## srt_path_git_parent_folder=WindowsPath('G:/ProgramData/Git/cmd')
```

```
print(f'{srt_path_git_grandparent_folder=}')
# Search for for the git-bash.exe file in parent and then in the grandparent folder.
```

```
## srt_path_git_grandparent_folder=WindowsPath('G:/ProgramData/Git')
```

Third, search inside parent folder first, and then grand until find the path to *git-bash.exe*. Will put all three steps code together:

```
# required packages
import shutil
from pathlib import Path
# find path to git
st_cmd = 'git'
spn_path_to_git = shutil.which(st_cmd)
# find path to git-bash.exe
spn_path_to_gitbash = ''
for it_up_iter in [0,1]:
    # up-tier folder
    srt_path_git_up_folder = Path(spn_path_to_git).parents[it_up_iter]
    # search
    # get file names in folders (not recursively)
    ls_spn_found_git_bash = [spn_file for spt_srh in [srt_path_git_up_folder]
                             for spn_file in Path(spt_srh).glob('git-bash.exe')]
    # if found, length of ls of founds files must be 1
    if len(ls_spn_found_git_bash) == 1:
        spn_path_to_gitbash = ls_spn_found_git_bash[0]
        break

if spn_path_to_gitbash == '':
    raise NameError(f'failed to find git-bash, {spn_path_to_git=}')
else:
    print(f'Found git-bash: {spn_path_to_gitbash} by searching around {spn_path_to_git=}')
```

```
## Found git-bash: G:\ProgramData\Git\git-bash.exe by searching around spn_path_to_git='G:\\ProgramData
```

1.2 Generate a tex file

Will a bare-bone tex file with some texts inside, save inside the *_file* subfolder.

First, create the text text string, note the the linebreaks utomatically generate linebreaks, note that slash need double slash:

```
# Create the Tex Text
# Note that trible quotes begin first and end last lines
stf_tex_contents = """\\documentclass[12pt,english]{article}
\\usepackage[bottom]{footmisc}
\\usepackage[urlcolor=blue]{hyperref}
\\begin{document}
\\title{A Latex Testing File}
\\author{\\href{http://fanwangecon.github.io/}{Fan Wang} \\thanks{See information \\href{https://fanwang
```

```
\\maketitle
Ipsum information dolor sit amet, consectetur adipiscing elit. Integer Latex placerat nunc orci.
\\paragraph{\\href{https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3140132}{Data}}
Village closure information is taken from a village head survey.\\footnote{Generally students went to s
\\end{document}"""
# Print
print(stf_tex_contents)
```

```
## \documentclass[12pt,english]{article}
## \usepackage[bottom]{footmisc}
## \usepackage[urlcolor=blue]{hyperref}
## \begin{document}
## \title{A Latex Testing File}
## \author{\href{http://fanwangecon.github.io/}{Fan Wang} \thanks{See information \href{https://fanwang
## \maketitle
## Ipsum information dolor sit amet, consectetur adipiscing elit. Integer Latex placerat nunc orci.
## \paragraph{\href{https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3140132}{Data}}
## Village closure information is taken from a village head survey.\footnote{Generally students went to
## \end{document}
```

Second, write the contents of the file to a new tex file stored inside the *_file* subfolder of the directory:

```
# Relative file name
srt_file_tex = "_file/"
sna_file_tex = "test_fan"
srn_file_tex = srt_file_tex + sna_file_tex + ".tex"
# Open new file
fl_tex_contents = open(srn_file_tex, 'w')
# Write to File
fl_tex_contents.write(stf_tex_contents)
# print
```

```
## 617
```

```
fl_tex_contents.close()
```

1.3 Replace Strings in a tex file

Replace a set of strings in the file just generated by a set of alternative strings.

```
# Open file Get text
fl_tex_contents = open(srn_file_tex)
stf_tex_contents = fl_tex_contents.read()
print(srn_file_tex)

# define new and old
```

```
## _file/test_fan.tex
```

```
ls_st_old = ['information', 'Latex']
ls_st_new = ['INFOREPLACE', 'LATEX']

# zip and loop and replace
for old, new in zip(ls_st_old, ls_st_new):
  stf_tex_contents = stf_tex_contents.replace(old, new)
print(stf_tex_contents)
```

```
# write to file with replacements
```

```
## \documentclass[12pt,english]{article}
## \usepackage[bottom]{footmisc}
## \usepackage[urlcolor=blue]{hyperref}
## \begin{document}
## \title{A LATEX Testing File}
## \author{\href{http://fanwangecon.github.io/}{Fan Wang} \thanks{See INFOREPLACE \href{https://fanwang
## \maketitle
## Ipsum INFOREPLACE dolor sit amet, consectetur adipiscing elit. Integer LATEX placerat nunc orci.
## \paragraph{\href{https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3140132}{Data}}
## Village closure INFOREPLACE is taken from a village head survey.\footnote{Generally students went to
## \end{document}
```

```
sna_file_edited_tex = "test_fan_edited"
srn_file_edited_tex = srt_file_tex + sna_file_edited_tex + ".tex"
fl_tex_ed_contents = open(srn_file_edited_tex, 'w')
fl_tex_ed_contents.write(stf_tex_contents)
```

```
## 617
```

```
fl_tex_ed_contents.close()
```

1.4 Convert Tex File to Pandoc and Compile Latex

Compile tex file to pdf and clean up the extraneous pdf outputs. See ff_pdf_gen_clean.

```
import subprocess
import os

# Change to local directory so path in tex respected.
os.chdir("C:/Users/fan/py4econ/support/inout")

# Convert tex to pdf
subprocess.call(['C:/texlive/2020/bin/win32/xelatex.exe', '-output-directory',
                 srt_file_tex, srn_file_edited_tex], shell=False)
# Clean pdf extraneous output
```

```
## 0
```

```
ls_st_remove_suffix = ['aux','log','out','bbl','blg']
for st_suffix in ls_st_remove_suffix:
    srn_cur_file = srt_file_tex + sna_file_edited_tex + "." + st_suffix
    if (os.path.isfile(srn_cur_file)):
        os.remove(srt_file_tex + sna_file_edited_tex + "." + st_suffix)
```

Use pandoc to convert tex file

```
import subprocess

# md file name
srn_file_md = srt_file_tex + "test_fan_edited.md"
# Convert tex to md
subprocess.call(['pandoc', '-s', srn_file_tex, '-o', srn_file_md])
# Open md file
```

```
## 0
```

```
fl_md_contents = open(srn_file_md)
print(fl_md_contents.read())
```

```
## ---
## author:
## - "[Fan Wang](http://fanwangecon.github.io/) [^1]"
## title: A Latex Testing File
## ---
##
## Ipsum information dolor sit amet, consectetur adipiscing elit. Integer
## Latex placerat nunc orci.
##
## #### [Data](https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3140132)
##
## Village closure information is taken from a village head survey.[^2]
##
## [^1]: See information
##     [Tex4Econ](https://fanwangecon.github.io/Tex4Econ/) for more.
##
## [^2]: Generally students went to schools.
```

1.5 Search for Files with Suffix in Several Folders

- python search all files in folders with suffix

Search for files in several directories that have a particular suffix. Then decompose directory into subcomponents.

Search file inside several folders (not recursively in subfolders):

```
from pathlib import Path

# directories to search in
ls_spt_srh = ["C:/Users/fan/R4Econ/amto/",
              "C:/Users/fan/R4Econ/function/"]

# get file names in folders (not recursively)
ls_spn_found = [spn_file for spt_srh in ls_spt_srh
                         for spn_file in Path(spt_srh).glob('*.Rmd')]
for spn_found in ls_spn_found:
  print(spn_found)
```

```
## C:\Users\fan\R4Econ\amto\main.Rmd
## C:\Users\fan\R4Econ\function\main.Rmd
```

Search file recursively in all subfolders of folders:

```
from pathlib import Path

# directories to search in
ls_spt_srh = ["C:/Users/fan/R4Econ/amto/array/",
              "C:/Users/fan/R4Econ/amto/list"]

# get file names recursively in all subfolders
ls_spn_found = [spn_file for spt_srh in ls_spt_srh
                         for spn_file in Path(spt_srh).rglob('*.R')]
for spn_found in ls_spn_found:
  drive, path_and_file = os.path.splitdrive(spn_found)
  path_no_suffix = os.path.splitext(spn_found)[0]
  path_no_file, file = os.path.split(spn_found)
  file_no_suffix = Path(spn_found).stem
  print('file:', file, '\ndrive:', drive,
        '\nfile no suffix:', file_no_suffix,
        '\nfull path:', spn_found,
        '\npt no fle:', path_no_file,
        '\npt no suf:', path_no_suffix, '\n')
```

```
## file: fs_ary_basics.R
## drive: C:
## file no suffix: fs_ary_basics
## full path: C:\Users\fan\R4Econ\amto\array\htmlpdfr\fs_ary_basics.R
## pt no fle: C:\Users\fan\R4Econ\amto\array\htmlpdfr
## pt no suf: C:\Users\fan\R4Econ\amto\array\htmlpdfr\fs_ary_basics
##
## file: fs_ary_generate.R
## drive: C:
## file no suffix: fs_ary_generate
## full path: C:\Users\fan\R4Econ\amto\array\htmlpdfr\fs_ary_generate.R
## pt no fle: C:\Users\fan\R4Econ\amto\array\htmlpdfr
## pt no suf: C:\Users\fan\R4Econ\amto\array\htmlpdfr\fs_ary_generate
##
## file: fs_ary_mesh.R
## drive: C:
## file no suffix: fs_ary_mesh
## full path: C:\Users\fan\R4Econ\amto\array\htmlpdfr\fs_ary_mesh.R
## pt no fle: C:\Users\fan\R4Econ\amto\array\htmlpdfr
## pt no suf: C:\Users\fan\R4Econ\amto\array\htmlpdfr\fs_ary_mesh
##
## file: fs_ary_string.R
## drive: C:
## file no suffix: fs_ary_string
## full path: C:\Users\fan\R4Econ\amto\array\htmlpdfr\fs_ary_string.R
## pt no fle: C:\Users\fan\R4Econ\amto\array\htmlpdfr
## pt no suf: C:\Users\fan\R4Econ\amto\array\htmlpdfr\fs_ary_string
##
## file: fs_listr.R
## drive: C:
## file no suffix: fs_listr
## full path: C:\Users\fan\R4Econ\amto\list\htmlpdfr\fs_listr.R
## pt no fle: C:\Users\fan\R4Econ\amto\list\htmlpdfr
## pt no suf: C:\Users\fan\R4Econ\amto\list\htmlpdfr\fs_listr
##
## file: fs_lst_basics.R
## drive: C:
## file no suffix: fs_lst_basics
## full path: C:\Users\fan\R4Econ\amto\list\htmlpdfr\fs_lst_basics.R
## pt no fle: C:\Users\fan\R4Econ\amto\list\htmlpdfr
## pt no suf: C:\Users\fan\R4Econ\amto\list\htmlpdfr\fs_lst_basics
```