

```

import os
import sublime import sublime_plugin

PDF_BASE_PATH="/home/jankincai/boolean/pdf/pro_comment/"

MD_BASE_PATH="/home/jankincai/boolean/docs/pro_comment/"

def run_pandoc():

for directory, dirs, files in os.walk(MD_BASE_PATH):

for file in files:

if ".md" in file:

path = os.path.join(directory, file)

pdf = file.replace(".md", ".pdf")

cmd = f'pandoc {path} -o {PDF_BASE_PATH}{pdf} -latex-engine=xelatex -V mainfont="Noto Sans CJK JP"'

print(cmd)

os.system(cmd)

PLUGIN_NAME = "Pandoc"

settings=sublime.load_settings("Pandoc.sublime-settings") output_suffix=settings.get("output_suffix",
".pdf") latex_engine=settings.get("latex_engine", "xelatex") mainfont=settings.get("mainfont",
"Noto Sans CJK JP")

def View(): """ Get current active window view """

return Window().active_view()

def Window(): """ """

```

```

return sublime.active_window()

class PandocCommand(sublime_plugin.TextCommand):
    def run(self, edit, *args, **kwargs):
        spath = self.view.file_name()

        path, suffix = os.path.splitext(spath)

        print(output_suffix)

        opath = "{}{}".format(path, output_suffix)

        cmd = "pandoc {} -o {} --latex-engine={} -V mainfont='{}'".format(
            spath,
            opath,
            latex_engine,
            mainfont,
        )

        os.system(cmd)

class PandocToPdfCommand(sublime_plugin.WindowCommand):
    def run(self, *args, **kwargs):
        print(dir(self))
        print(dir(self.window))
        # print(dir(self.window.active_view()))
        print(self.name())
        print(self.window.project_file_name())
        print(View().file_name())
        self.window.show_input_panel("File",
        View().file_name(), None, None, None)

```