Table of Contents

-- *coding:utf-8* --

Markdown Literary Programming

Author: Lin Pengcheng

Blog: Markdown Literary Programming

Note

Subdirectorys and file of mlt_base_dir

- out (Changeable)

- js

- image

- css

- MultiMarkdown (Optional)

- CuteMarked (Optional)

- ChromePortable (Optional)

- MLT_common_head.md

out_dir can be separated from mlt_base_dir(public resource directory),

but must be on the same disk, in “mlt_common_head.md” file,

the path starts with the root path, for example:

```
/path_to/mlt_base_dir/js/prismjs/prism.js
```

Python lib:

markdown or mistune does not support mermaid

Viewer

Cutemarked (viewer = 1)

- it is the only native Markdown viewer that supports JS, image, HTML.
- it has an outline view of Markdown.
- It exports HTML in chrome is not good to display.

MultiMarkdown + Chrome (viewer = 2)

- MultiMarkdown support TOC.
- MultiMarkdown does not support the GitHub style

code block block and Mermaid charts,

I made a correction in py.

- Multimarkdown faster and lighter than pandoc.
- When converted to HTML, it is automatically open with Chrome,

and the display works well.

Markdownviewer++ (viewer = 3)

- It exports HTML in chrome is displayed very well
- fast, Its viewer is only supported by plain text markdown LivePreview.
- its viewer no syntax highlighted, no mermaid, no image displayed

Config

```
viewer = 2

is_clean_mlt_out_dir = False
```

```
is_delete_generate_file = False

mlt_base_dir = notepad.getNppDir() + "\\MarkdownLiteraryProgramming"

# mlt_out_dir = mlt_base_dir + "\\out"

mlt_out_dir = mlt_base_dir

cutemarked_path = mlt_base_dir + "\\cutemarked\\cutemarked.exe"

multimarkdown_path = mlt_base_dir + "\\MultiMarkdown\\bin\\multimarkdown.exe"

chrome_path = "D:\\PortableApps\\GoogleChromePortable\\GoogleChromePortable.exe"
```

End Config

Code

Init

```
import os

import re

import time

notepad.save()

current_full_path = notepad.getCurrentFilename()

(current_dir, current_filename) = os.path.split(current_full_path)

if not os.path.exists(mlt_out_dir):

   os.makedirs(mlt_out_dir)
```

get Common Head Text

```
common_head_txt = ""

common_head_file = mlt_base_dir + "\\MLP_common_head.md"

if os.path.exists(common_head_file):

   f= open(common_head_file,'r')

   common_head_txt = f.read()

   f.close()

replace_txt = ""

if mlt_out_dir == mlt_base_dir + "\\out":

   replace_txt = ".."
```

```
elif mlt_out_dir == mlt_base_dir:
   replace_txt = "."
else: # remove disk, unix style
   replace_txt = re.sub('[a-z,A-Z]{1}:', "", mlt_base_dir)
   replace_txt = replace_txt.replace('\\','/')
common_head_txt = re.sub('%mlt_base_dir%', replace_txt, common_head_txt,0,re.M)
```

Convert Markdown Literary Programming to Markdown

```
dest_text = editor.getText()
# lisp or null (default as lisp)
if editor.getLexerLanguage() == 'lisp' or editor.getLexerLanguage() == 'null':
   dest_text = re.sub(r'^;', "", dest_text,0,re.M)
elif editor.getLexerLanguage() == 'python' or editor.getLexerLanguage() == 'r':
   dest_text = re.sub(r'^#', "", dest_text,0,re.M)
elif editor.getLexerLanguage() == 'cpp':
   dest_text = re.sub(r'//', "", dest_text,0,re.M)
else:
   dest_text = dest_text
# dest_text = common_head_txt + markdown.markdown(dest_text)
# dest_text = common_head_txt + mistune.markdown(dest_text)
```

merge Common Head Text and Markdown

```
md_body_begin = '\r\n\r\n<article class="markdown-body">\r\n\r\n'
md_body_end   = '\r\n\r\n</article>'
TOC = "# Table of Contents \r\n\r\n{{TOC}}\r\n\r\n"
if viewer == 2:
   dest_text = common_head_txt + md_body_begin + TOC + dest_text +md_body_end
else:
   dest_text = common_head_txt + md_body_begin + dest_text + md_body_end
```

View

CuteMarked

```
if viewer == 1:

    # Method 1: create a md file, to open by cutemarked.

    if not os.path.exists(cutemarked_path):

        cutemarked_url = "https://github.com/cloose/CuteMarkEd"

        msg = "CuteMarked.exe does not exists! \r\nvar cutemarked_path: " + 
cutemarked_path + "\r\nIt can be download from " + cutemarked_url

        notepad.messageBox(msg, "Warning", 0)

    dest_full_path = mlt_out_dir + "\\" + current_filename + ".md"

    f= open(dest_full_path,'w')

    f.write(dest_text)

    f.close()

    cmd = cutemarked_path + " " + dest_full_path

    # os.system(cmd)

    os.popen(cmd)

    if is_delete_generate_file:

        os.remove(dest_full_path)
```

MultiMarkdown+Chrome

```
elif viewer == 2:

          # Method 2: MultiMarkdown+Chrome

    if not os.path.exists(multimarkdown_path):

        multimarkdown_url = "https://github.com/fletcher/MultiMarkdown-6"

        msg = "multimarkdown.exe does not exists! \r\nvar multimarkdown_path: " + 
multimarkdown_path + "\r\nIt can be download from " + multimarkdown_url

        notepad.messageBox(msg, "Warning", 0)

    if not os.path.exists(chrome_path):

        chrome_url = "https://portableapps.com/apps/internet/google_chrome_portable"

        msg = "Chrome.exe(or GoogleChromePortable.exe) does not exists! \r\nvar 
chrome_path: " + chrome_path + "\r\nIt can be download from " + chrome_url

        notepad.messageBox(msg, "Warning", 0)
    md_full_path = mlt_out_dir + "\\" + current_filename + ".md"

    f= open(md_full_path,'w')

    f.write(dest_text)

    f.close()

    html_full_path = mlt_out_dir + "\\" + current_filename + ".html"

    cmd = multimarkdown_path + " --full --to=html --output=" + html_full_path + " " + md_full_path

    # cmd = multimarkdown_path + " --full " + md_full_path

    os.popen(cmd)

    f= open(html_full_path,'r')

    html_txt = f.read()

    f.close()

    html_txt = re.sub(r'<pre><code class="', r'<pre><code class="language-', html_txt, 0, re.M)

    html_txt = re.sub(r'<pre><code class="language-mermaid">', r'<pre><code class="mermaid">', html_txt, 0, re.M)

    html_txt = html_txt.replace("\r\n\r\n","\r\n")

    f= open(html_full_path,'w')

    f.write(html_txt)

    f.close()

    cmd = chrome_path + " " + html_full_path

    os.popen(cmd)

    if is_delete_generate_file:

        time.sleep(15)

        os.remove(md_full_path)

        os.remove(html_full_path)
```

MarkdownViewer++Chrome

```
else:

    # Method 3: MarkdownViewer++Chrome

    editor2= notepad.new()

    editor.setText(dest_text)
    # dest_full_path = mlt_out_dir + current_filename + ".md"

    # notepad.saveAs(dest_full_path)

    notepad.runPluginCommand('MarkdownViewer++','MarkdownViewer++')
```

Clean mlt_out_dir

```
def del_file(path):

    ls = os.listdir(path)

    for i in ls:

        c_path = os.path.join(path, i)

        if os.path.isdir(c_path):

            del_file(c_path)

        else:

            os.remove(c_path)
if is_clean_mlt_out_dir:

    if viewer == 2:

        time.sleep(15)

    del_file(mlt_out_dir)
```

End Code