Contents

1 Identity2

1.1 Code

> This site is a working demonstration of taking a standard gemini capsule and auto generating and distributing derived MD, Web, LaTeX, PDF, ePub, etc.

If you can improve on my ugly Python and the HTML template please share your ideas, and although the Eisvogel works magically I am sure something simple and elegant can devised for this specific task.

I am interested in futhering and promoting gemtext encoding /gemini as an integral part for the complete heirachy of longtext publishing.

This site is processed thru Gemtext to Markdown, Markdown to Html, Markdown to LaTeX, Latex to PDF, PDF to print. If that all works ok finalstep of Htm to eBooks should be trivial.

Gemtext encoding is identical to COMMONMARK standard markdown except for the URL encoding, so the following converts it to commonmark.

```
[Python3]
modif = ''
with open(fi, 'r') as i_file:
   while True:
       line = i_file.readline()
       if not line:
           break
       if">"<a href="" in line:</a>

           line">= re.sub('<a href="', '<a href="', line)</a>
           line = re.sub('.gmi  +', '.html">', line)
           line = re.sub('  +', '">', line)
           line = re.sub('\n', '</a>', line)
           line = re.sub('gemini://', 'https://', line)
           modif = f'{modif} {line}\n\n'
       else:
           modif = f'{modif} {line}'
mdfile.write(modif)
mdfile.close()
```

With the execption that the html encoded url is ignored by pandoc in the following

```
[Python3]
md = re.sub('.gmi', '.md', fi)
texdir = re.sub('gemini', 'gemini-tex', tex)
title = re.sub('.gmi', '', file)
os.system(
  f'pandoc --standalone --from=commonmark --metadata title="{title}" '
   f'--template tmplt.html {md} -o {htm}'
)
```

With tmplt.html being something like the following

```
[html]
<!doctype html>
<html lang="en">
  <head>
    <meta charset="utf-8">
    <meta name="date" content='$date-meta$'>
    <title>Identity2 $title$</title>
<style>
body{margin-left: 4em; background-color: #ffffff;
 background-image: url("i/Noisy-Grid.jpg");background-repeat;}

h1  {color: #003c32; margin-top: -0.3em; margin-left: -2em;
     padding-left: 5em; padding-bottom: 0.5em;
     padding-top: 0.6em; padding-left: -1em;
     background-color: #ddd;}

h2  {color: #003c32; margin-top: .3em; padding-bottom: 0.2em;
     padding-top: .5em; padding-left: -1em;
     background-color: rgba(255,255,255,0.33);}

h3  {color: #aad7dc;}

ul  {color: #804000;}

li  {color: #804000;}

blockquote {
    font: 14px/22px normal helvetica, sans-serif; margin-top: 10px;
    margin-bottom: 10px; margin-left: 30px; padding-left: 15px;
    border-left: 6px solid #aad7dc;}

p   {padding-left: 1em;padding-top: .6em;}

.language-ascii-art2 {color: #003c32; display: inline-block;
    text-shadow: 0 0 5px rgba(100,100,100,0.5);
    background-color: rgba(255,255,255,0.33);}}
</style>
  </head>
  <body>
$body$
  </body>
</html>
```

I place this html on a standard web server, by automated script of course.

Similarly:

```
[Python3]
os.system(
   f'pandoc -N --pdf-engine=xelatex  --template '
   f'eisvogel -f commonmark --toc {md} -o {pdf}')
```

Pandoc using XeTeX (as its unicode) and the eisvogel template generates a standalone PDF of every page using this template.

It will be trivial to concantonate all the pages into a book in the same script iterator, you need to be sure headings levels are consistent so that the pdf’s generated table of contents makes sense.

Since this is an instruction site the properly formatted and paginated pdf allows users to print instructions cleanly.

This demonstration can inform creators on the effect of site design on repurposed documents and gemini server builders on possible feaatures that can be incorporated as server features.

□ □ □ □ □ □