

```
1| #!/usr/bin/env raku
2|
3| # Weave a Semi-literate file into Text, Markdown, HTML, etc.
4| # © 2023 Shimon Bollinger. All rights reserved.
5| # Last modified: Wed 06 Sep 2023 03:28:29 PM EDT
6| # Version 0.0.1
7|
```

# Weave a semi-literate program into Text, Markdown, etc. format

```
8| #| Weave Markdown documentation from Raku code
9| sub MAIN($input-file,
10|         Bool :l(:$line-numbers) = True;
11|         #= Should line numbers be added to the embeded code?
12|         Str :f(:$format) is copy = 'markdown';
13|         #= The output format for the woven file.
14|         Str :o(:$output-file);
15|         #= The name of the output file. Defaults to stdout.
16|         Bool :v(:$verbose) = True;
17|         #= verbose will print diagnostics and debug prints to $*ERR
18|     ) {
19|     my Str $extension;
20|     my Str @options;
21|     my Bool $no-output-file = False;
22|
23|     note "Input Format => $format" if $verbose;
24|     $format .= trim;
25|     given $format {
26|         when /^:i ^ markdown | md $ / {
27|             $format = 'MarkDown2';
28|             $extension = 'md';
29|         };
30|         when /^:i ^ [[plain][\-|_]?]? t[e]?xt $ / {
31|             $format = 'Text';
32|             $extension = 'txt';
33|         }
34|         when /^:i ^ [s]?htm[l]? $/ {
35|             $format = 'HTML2';
36|             $extension = 'html';
37|         } # end of when /^:i html 2? $/
38|
39|         when /^:i ^ pdf $ / {
40|             $format = 'PDF';
41|             $extension = '.pdf';
42|             @options = "--save-as=$output-file" if $output-file;
43|             $no-output-file = True;
44|         }
45|
46|         when /^:i ^ pdf[\-|_]?lite $ / {
47|             $format = 'PDF::Lite';
48|             $extension = '.pdf';
49|             @options = "--save-as=$output-file" if $output-file;
50|             $no-output-file = True;
51|         }
52|
53|         default {
54|             $extension = $format;
55|         } # end of default
56|
57|
58|     } # end of given $output-format
59|     note "Weave Format => $format" if $verbose;
60|     my Str $f = "Pod::To::$format";
```

```
61|     try require ::($f);
62|     if ::($f) ~~ Failure {
63|         die "$format is not a supported output format"
64|     } # end of if ::("Pod::To::$_") ~~ Failure
65|
66|     my Str $woven = weave($input-file, :$format, :$line-numbers);
67|
68|     my ($pod-file, $fh) = tempfile(suffix => '.p6');
69|
70|     $pod-file.IO.spurt: $woven;
71|
72|     my $output-file-handle = $output-file      ??
73|                             open(:w, $output-file) !!
74|                             $*OUT
75|                             unless $no-output-file;
76|
77|     run $*EXECUTABLE,
78|         "--doc=$format",
79|         $pod-file,
80|         @options,
81|         :out($output-file-handle);
82|
83| } # end of sub MAIN($input-file,
84|
```