

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

```

3| sub MAIN($input-file,
4|         Bool :l(:$line-numbers) = True;
5|         Str :f(:$format) is copy = 'markdown';
6|         Str :o(:$output-file);
7|         Bool :v(:$verbose) = True;
8|     ) {
9|     my Str $extension;
10|    my Str @options;
11|    my Bool $no-output-file = False;
12|
13|    note "Input Format => $format" if $verbose;
14|    $format .= trim;
15|    given $format {
16|        when /:i ^ markdown | md $ / {
17|            $format = 'MarkDown2';
18|            $extension = 'md';
19|        };
20|        when /:i ^ [[plain][\-|_]]? t[e]?xt $ / {
21|            $format = 'Text';
22|            $extension = 'txt';
23|        }
24|        when /:i ^ [s]?htm[l]? $/ {
25|            $format = 'HTML2';
26|            $extension = 'html';
27|        }
28|        when /:i ^ pdf $ / {
29|            $format = 'PDF';
30|            $extension = '.pdf';
31|            @options = "--save-as=$output-file" if $output-file;
32|            $no-output-file = True;
33|        }
34|        when /:i ^ pdf[\-|_]?lite $ / {
35|            $format = 'PDF::Lite';
36|            $extension = '.pdf';
37|            @options = "--save-as=$output-file" if $output-file;
38|            $no-output-file = True;
39|        }
40|        when /:i ^ pod 6? $/ {
41|            $format = 'Pod6';
42|            $extension = '.rakudoc';
43|        }
44|
45|        default {
46|            $extension = $format;
47|        }
48|
49|    }
50|    my Str $woven = weave($input-file, :$line-numbers);
51|
52|    my $output-file-handle = $output-file ??
53|                            open(:w, $output-file) !!
54|                            $*OUT
55|                            unless $no-output-file;

```

```
56|
57|     if $format eq 'Pod6' {
58|         $output-file-handle.spurt: $woven;
59|         return;
60|     }
61|
62|     note "Weave Format => $format" if $verbose;
63|     my Str $f = "Pod::To::$format";
64|     try require ::($f);
65|     if ::($f) ~~ Failure {
66|         die "$format is not a supported output format"
67|     }
68|
69|     my ($pod-file, $fh) = tempfile(suffix => '.rakudoc', :!unlink);
70|     note "Temp file: $pod-file" if $verbose;
71|
72|     $pod-file.IO.spurt: $woven;
73|
74|     run $*EXECUTABLE,
75|         "--doc=$format",
76|         $pod-file,
77|         @options,
78|         :out($output-file-handle);
79|
80| }
81|
```