```
1| use v6.d;
2|
3| use File::Temp;
4| use Semi::Literate;
```

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

```
  5|  sub MAIN($input-file,
  6|          Bool :l(:$line-numbers)  = True;
  7|          Str :f(:$format) is copy = 'markdown';
  8|          Str :o(:$output-file);
  9|          Bool :v(:$verbose) = True;
 10|      ) {
 11|      my Str $extension;
 12|      my Str @options;
 13|      my Bool $no-output-file = False;
 14|
 15|      note "Input Format =>  $format" if $verbose;
 16|      $format .= trim;
 17|      given $format {
 18|          when  /:i ^ markdown | md $ / {
 19|              $format    = 'MarkDown2';
 20|              $extension = 'md';
 21|          };
 22|          when  /:i ^ [[plain][\-|_]?]? t[e]?xt $ / {
 23|              $format    = 'Text';
 24|              $extension = 'txt';
 25|          }
 26|          when  /:i ^ [s]?htm[l]? $/ {
 27|              $format    = 'HTML2';
 28|              $extension = 'html';
 29|          }
 30|
 31|          when /:i ^ pdf $ / {
 32|              $format = 'PDF';
 33|              $extension = '.pdf';
 34|              @options = "--save-as=$output-file" if $output-file;
 35|              $no-output-file = True;
 36|          }
 37|
 38|          when /:i ^ pdf[\-|_]?lite  $ / {
 39|              $format = 'PDF::Lite';
 40|              $extension = '.pdf';
 41|              @options = "--save-as=$output-file" if $output-file;
 42|              $no-output-file = True;
 43|          }
 44|
 45|          default {
 46|              $extension = $format;
 47|          }
 48|
 49|
 50|      }
 51|      note "Weave Format =>  $format" if $verbose;
 52|      my Str $f = "Pod::To::$format";
 53|      try require ::($f);
 54|      if ::($f) ~~ Failure {
 55|          die "$format is not a supported output format"
 56|      }
 57|
```

```
58|     my Str $woven = weave($input-file, :$format, :$line-numbers);
59|
60|     my ($pod-file, $fh) = tempfile(suffix =>  '.p6');
61|
62|     $pod-file.IO.spurt: $woven;
63|
64|     my $output-file-handle = $output-file              ??
65|                                 open(:w, $output-file) !!
66|                                   $*OUT
67|                              unless $no-output-file;
68|
69|     run $*EXECUTABLE,
70|         "--doc=$format",
71|         $pod-file,
72|         @options,
73|         :out($output-file-handle);
74|
75| }
76|
77|
```