```

# Weave a Semi-literate file into Text, Markdown, HTML, etc.
# © 2023 Shimon Bollinger. All rights reserved.
# Last modified: Wed 06 Sep 2023 03:28:29 PM EDT
# Version 0.0.1

```

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

```
sub MAIN($input-file,
         Bool :l(:$line-numbers)  = True;
            #= Should line numbers be added to the embeded code?
         Str :f(:$format) is copy = 'markdown';
            #= The output format for the woven file.
         Str :o(:$output-file);
            #= The name of the output file.  Defaults to stdout.
         Bool :v(:$verbose) = True;
            #= verbose will print diagnostics and debug prints to $*ERR
    ) {
    my Str $extension;
    my Str @options;
    my Bool $no-output-file = False;

    note "Input Format =>  $format" if $verbose;
    $format .= trim;
    given $format {
        when  /:i ^ markdown | md $ / {
            $format    = 'MarkDown2';
            $extension = 'md';
        };
        when  /:i ^ [[plain][\-|_]?]? t[e]?xt $ / {
            $format    = 'Text';
            $extension = 'txt';
        }
        when  /:i ^ [s]?htm[l]? $/ {
            $format    = 'HTML2';
            $extension = 'html';
        } # end of when  /:i html 2? $/

        when /:i ^ pdf $ / {
            $format = 'PDF';
            $extension = '.pdf';
            @options = "--save-as=$output-file" if $output-file;
            $no-output-file = True;
        }

        when /:i ^ pdf[\-|_]?lite  $ / {
            $format = 'PDF::Lite';
            $extension = '.pdf';
            @options = "--save-as=$output-file" if $output-file;
            $no-output-file = True;
        }

        default {
            $extension = $format;
        } # end of default


    } # end of given $output-format
    note "Weave Format =>  $format" if $verbose;
    my Str $f = "Pod::To::$format";
    try require ::($f);
    if ::($f) ~~ Failure {
        die "$format is not a supported output format"
    } # end of if ::("Pod::To::$_") ~~ Failure

    my Str $woven = weave($input-file, :$format, :$line-numbers);

    my ($pod-file, $fh) = tempfile(suffix =>  '.p6');

    $pod-file.IO.spurt: $woven;

    my $output-file-handle = $output-file              ??
                                open(:w, $output-file) !!
                                $*OUT
                            unless $no-output-file;

    run $*EXECUTABLE,
        "--doc=$format",
        $pod-file,
        @options,
        :out($output-file-handle);

} # end of sub MAIN($input-file,

```