

jodliterate Group

John D. Baker

May 25, 2020

Contents

jodliterate Overview	2
jodliterate Interface	2
jodliterate and JOD	2
Running jodliterate	3
jodliterate Source Code	6
=: Index	52

jodliterate Overview

jodliterate is a J utility script that generates [literate](#) documents directly from JOD dictionary groups.

See the following for details:

1. [Using jodliterate notebook](#).
2. [Using jodliterate PDF](#).

jodliterate Interface

<code>THISPANDOC</code>	<i>full pandoc path - use (pandoc) if on shell path</i>
<code>grplit</code>	<i>make latex for group (y)</i>
<code>ifacesection</code>	<i>interface section summary string</i>
<code>setjodliterate</code>	<i>prepare for processing</i>

jodliterate and JOD

jodliterate makes some assumptions about the J code and document text it processes. It assumes:

1. All source code is stored in JOD dictionaries.
2. jodliterate document fragments are either:
 - Markdown group headers: `2 9 disp 'groupname'`
 - \LaTeX suffixed macros: `4 disp 'groupname','_oview_tex'`

Running jodliterate

jodliterate runs on Windows, Linux and Mac versions of J.¹

To use jodliterate you must:

1. Install JOD. [JOD](#) is a J addon. It is usually installed with [pacman](#).
2. Install JODSOURCE. [JODSOURCE](#) is also a J addon. Install it with [pacman](#). It contains the JOD dictionaries required to build JOD. JODSOURCE also contains jodliterate.
3. Install a current version of [pandoc](#). pandoc version 2.9.1.1 and beyond supports J syntax highlighting. Prior versions required modification. The following blog posts provide more information:
 - (a) [More J Pandoc Syntax HighLighting](#)
 - (b) [Pandoc based J Syntax Highlighting](#)
 - (c) [Semi-Literate JOD](#).
4. Build jodliterate. jodliterate is a J group. JOD makes J scripts from groups. To make jodliterate do:

```
require 'general/jod'  
od ;:'joddev jod utils' [ 3 od ''  
mls 'jodliterate'
```

5. Open the JOD dictionaries containing the group you want to document.

```
od ;:'joddev jod utils' [ 3 od ''
```

¹jodliterate will run on J 8.01 and beyond.

6. Load `jodliterate`. After making `jodliterate` it can be loaded like any J script.

```
load 'jodliterate'
```

7. Set a working directory. `jodliterate` generates \LaTeX files. All such files are written to the directory specified by `setjodliterate`.

```
setjodliterate '' NB. current JOD put dictionary document directory
```

```
setjodliterate 'c:\temp' NB. windows
```

```
setjodliterate '/home/john/temp' NB. linux
```

8. Run `grplit` on the group you want to document. `grplit` generates \LaTeX files in the working directory. The root \LaTeX file is given the group name.

```
grplit 'jodliterate' NB. makes jodliterate.tex
```

9. Use \LaTeX to compile the files generated by `grplit`.

A batch `bat` script is written to the `setjodliterate` directory that shows the sequence of \LaTeX compilation commands.

Files may be processed with `pdflatex`, `xelatex` or `lualatex`. Your choice will be dictated by the presence of Unicode characters. For more details see the preamble file `JODLiteratePreamble.tex` in the working directory.

There are some \LaTeX settings you may want to adjust.

1. Edit `JODLiteratePreamble.tex` to change the number of index columns. The default is four but this may result in long names running together. To decrease or increase index columns change:

```
\begin{multicols}{4}[\section*{\indexname}]
```

2. `jodliterate` wraps long source code lines. Wrapping is controlled by `WRAPLIMIT`. The default is 110 characters. Change this value if you change font or page size. Wrapped lines are preceded with the string `WRAPLEAD` with default `>. .>`.

jodliterate Source Code

*NB.*jodliterate s-- generates literate source code documents directly from JOD groups.*
NB.
NB. verbatim: see the following blog posts and github files
NB.
NB. <https://analyzethedatanotthedrivel.org/2012/10/01/semi-literate-jod/>
NB. <https://analyzethedatanotthedrivel.org/2020/02/19/more-j-pandoc-syntax-highlighting/>
NB. <https://github.com/bakerjd99/jacks/blob/master/jodliterate/UsingJodliterate.pdf>
NB. <https://github.com/bakerjd99/jacks/blob/master/jodliterate/Using%20jodliterate.ipynb>
NB.
NB. interface word(s):
NB. -----
NB. THISPANDOC - full pandoc path - use pandoc only if on shell path
NB. grplit - make latex for group (y)
NB. ifacesection - interface section summary string
NB. ifc - format interface comment text
NB. setjodliterate - prepare LaTeX processing - sets directory - writes preamble
NB.
NB. author: John D. Baker
NB. created: 2012oct01
NB. -----
NB. 12oct03 (x) grplit argument added to suppress root tex overwrites
NB. 12oct04 group IFACEWORDSgroupname hyperlinked
NB. 12oct05 replaced ;: parsing with (wfl) - handles bad j code
NB. 12oct08 added error handling - replaced (write) with (writeas)
NB. 12oct11 adjusted LaTeX preamble - changing monofonts

NB. 12oct12 added (sbtokens) - useful for analyzing code text
NB. 12oct17 added (wrapvrblong) - long source lines now wrapped
NB. 13dec29 added to (jacks) GitHub repository
NB. 20may07 adjusted word formation (wfl) for J 9.01
NB. 20may08 updated for current (pandoc) versions

```
coclass 'ajodliterate'  
coinsert 'ijod'
```

*NB.*dependents*
NB. declared global here to avoid confusing LaTeX names with J names
NB. ()=: JLTITLETEX JLOVIEWTEX JLBUILDTTEX JLGRPLITTEX JODLiteratePreamble*

NB. Roger Hui's word formation state machine - similiar to ;: but
NB. parses text with LFs, retains whitespace and handles open quotes.
NB.
NB. verbatim: note difference
NB.
NB. wfl'+/ i. 23 5, ''OPEN QUOTE'
NB. ;:'+/ i. 23 5, ''OPEN QUOTE'

NB. hide script locals !()=. mfl sfl*
mfl=. 256\$0 NB. X other
mfl=. 1 (9,a.i.' ') }mfl NB. S whitespace (space and horizontal tab)
mfl=. 2 (,(a.i.'Aa')+/i.26) }mfl NB. A A-Z a-z excluding N B
mfl=. 3 (a.i.'N') }mfl NB. N the letter N
mfl=. 4 (a.i.'B') }mfl NB. B the letter B

```

mfl=. 5 (a.i.'0123456789_') }mfl NB. 9 digits and _
mfl=. 6 (a.i.'.') }mfl NB. D .
mfl=. 7 (a.i.':') }mfl NB. C :
mfl=. 8 (a.i.'''') }mfl NB. Q quote
mfl=. 9 (13) }mfl NB. CR
mfl=. 10 (10) }mfl NB. LF

sfl=. _2]\ "1 }.".;._2 (0 : 0)
' X S A N B 9 D C Q CR LF ' ]0
1 1 12 1 2 1 3 1 2 1 6 1 1 1 1 7 1 10 1 1 1 NB. 0 initial
1 2 12 2 2 2 3 2 2 2 6 2 1 0 1 0 7 2 10 2 1 2 NB. 1 other
1 2 12 2 2 0 2 0 2 0 2 0 1 0 1 0 7 2 10 2 1 2 NB. 2 alp/num
1 2 12 2 2 0 2 0 4 0 2 0 1 0 1 0 7 2 10 2 1 2 NB. 3 N
1 2 12 2 2 0 2 0 2 0 2 0 5 0 1 0 7 2 10 2 1 2 NB. 4 NB
9 0 9 0 9 0 9 0 9 0 9 0 1 0 1 0 9 0 10 2 1 2 NB. 5 NB.
1 4 13 0 6 0 6 0 6 0 6 0 6 0 1 0 7 4 10 2 1 2 NB. 6 num
7 0 7 0 7 0 7 0 7 0 7 0 7 0 8 0 10 2 1 2 NB. 7 '
1 2 11 2 2 2 3 2 2 2 6 2 1 2 1 2 7 0 10 2 1 2 NB. 8 ''
9 0 9 0 9 0 9 0 9 0 9 0 9 0 9 0 10 2 1 2 NB. 9 comment
1 2 11 2 2 2 4 2 2 2 6 2 1 2 1 2 7 2 10 2 11 0 NB. 10 CR
1 2 11 2 2 2 4 2 2 2 6 2 1 2 1 2 7 2 10 2 1 2 NB. 11 CRLF
1 2 12 0 2 2 3 2 2 2 6 0 1 2 1 2 7 2 10 2 1 2 NB. 12 space
1 2 13 0 2 2 3 2 2 2 6 0 1 2 1 2 7 2 10 2 1 2 NB. 13 space after num
)

NB. word formation for lines
wfl=: (0;sfl;mfl) & ;:

```



```
JLDIRECTORY=:  ''
NB.*enddependents

NB.<<~~~~ { .latex }

NB. group title and author - standard \maketitle
JLTITLETEX=:  0 : 0

% latex author and title
\author{~#~author~#~}
\title{\texttt{~#~group~#~} Group}
)

NB. group overview header
JLOVIEWTEX=:  0 : 0

% this jodliterate overview
\section{\texttt{~#~group~#~} Overview}
)

NB. latex group build script
JLBUILDTEX=:  0 : 0

rem sequence of latex commands that generate PDF
rem assumes latex exes are on the working path
```

```
setlocal
cd /d %~dp0
lualatex ~#~group~#~
makeindex ~#~group~#~
lualatex ~#~group~#~
lualatex ~#~group~#~
endlocal
)
```

NB. group root tex - columns may need adjusting

```
JLGRPLITTEX=: 0 : 0
```

```
% Main jodliterate (grplit) latex file. (grplit) generates "group"
% named versions of this file for each JOD group it processes.
```

```
\input{JODLiteratePreamble.tex}
```

```
\input{~#~group~#~title.tex}
```

```
\maketitle
```

```
\tableofcontents
```

```
\newpage
```

```
% commands for adjusting distance
```

```
% between columns and inserting a rule
```

```
%\setlength{\columnsep}{3em}
```

```
%\setlength{\columnseprule}{0.5pt}
```

```
%\twocolumn
```

```
\input{~#~group~#~oview.tex}

\newpage
%\onecolumn
\input{~#~group~#~code.tex}

\newpage
\phantomsection
\addcontentsline{toc}{section}{\texttt{=:} Index}
\printindex

\end{document}
)

NB. main jodliterate LaTeX preamble
JODLiteratePreamble=: 0 : 0

% jodliterate latex preamble.
%
% This file is a highly customized version of the preamble
% material generated by pandoc's -s option when producing
% .tex output. pandoc highlighting is overridden and
% the standard index is redefined.

\documentclass[12pt]{article}

\usepackage[landscape]{geometry}
```

```
\usepackage[headings]{fullpage}
\usepackage{lmodern}
\usepackage{amssymb,amsmath}
\usepackage{ifxetex,ifluatex}

% provides \textsubscript
\usepackage{fixltx2e}

% use microtype if available
\IfFileExists{microtype.sty}{\usepackage{microtype}}{}
\ifnum 0\ifxetex 1\fi\ifluatex 1\fi=0 % if pdftex
  \usepackage[utf8]{inputenc}
\else % if luatex or xelatex
  \usepackage{fontspec}
  \ifxetex
    \usepackage{xltxtra,xunicode}
  \fi
  \defaultfontfeatures{Mapping=tex-text,Scale=MatchLowercase}
  % replace EUROUC with unicode euro character
  % if you need this character - the presence of
  % this single character in the preamble forces use of xelatex, lualated
  %\newcommand{\euro}{EUROUC}
  % can set other monospace fonts if they're available
  % I rather like Source Code Pro see:
  % http://blogs.adobe.com/typblography/2012/09/source-code-pro.html
  %\setmonofont{FreeMono}
  %\setmonofont{Source Code Pro}
```

```
\fi

% Redefine labelwidth for lists; otherwise, the enumerate package will cause
% markers to extend beyond the left margin.
\makeatletter\AtBeginDocument{%
  \renewcommand{\@listi}
    {\setlength{\labelwidth}{4em}}
\makeatother
\usepackage{enumerate}

% tightlist command for list spacing
\providecommand{\tightlist}{%
  \setlength{\itemsep}{0pt}\setlength{\parskip}{0pt}}

% build document index
\usepackage{makeidx}

% colors
\usepackage{color}
\definecolor{shadecolor}{RGB}{248,248,248}
% j control structures
\definecolor{keywcolor}{rgb}{0.13,0.29,0.53}
% j explicit arguments x y m n u v
\definecolor{datacolor}{rgb}{0.13,0.29,0.53}
% j numbers - all types see j.xml
\definecolor{decvcolor}{rgb}{0.00,0.00,0.81}
\definecolor{basencolor}{rgb}{0.00,0.00,0.81}
```

```
\definecolor{floatcolor}{rgb}{0.00,0.00,0.81}
% j local assignments
\definecolor{charcolor}{rgb}{0.31,0.60,0.02}
\definecolor{stringcolor}{rgb}{0.31,0.60,0.02}
\definecolor{commentcolor}{rgb}{0.56,0.35,0.01}
% primitive adverbs and conjunctions
%\definecolor{othercolor}{rgb}{0.56,0.35,0.01}
\definecolor{othercolor}{RGB}{0,0,255}
% global assignments
\definecolor{alertcolor}{rgb}{0.94,0.16,0.16}
% primitive J verbs and noun names
\definecolor{funccolor}{rgb}{0.00,0.00,0.00}

\usepackage{fancyvrb}
\DefineShortVerb[commandchars=\\\{\}\]{\|}
\DefineVerbatimEnvironment{Highlighting}{Verbatim}{commandchars=\\\{\}\}
% Add ',fontsize=\small' for more characters per line

% pandoc generated syntax coloring commands - names
% are fixed in generated code but definitions may
% be set to any valid text formatting command
\usepackage{framed}
\newenvironment{Shaded}{}{}
\newcommand{\KeywordTok}[1]{\textcolor{keywcolor}{\textbf{#1}}}}
% works better with Source Code Pro
%\newcommand{\KeywordTok}[1]{\textcolor{keywcolor}{#1}}
\newcommand{\DataTypeTok}[1]{\textcolor{datacolor}{#1}}
```

```
%\newcommand{\DecValTok}[1]{\textcolor{decvcolor}{\{#1\}}}  
\newcommand{\DecValTok}[1]{\{#1\}}  
\newcommand{\BaseNTok}[1]{\textcolor{basencolor}{\{#1\}}}  
\newcommand{\FloatTok}[1]{\textcolor{floatcolor}{\{#1\}}}  
\newcommand{\CharTok}[1]{\textcolor{charcolor}{\textbf{\{#1\}}}}  
\newcommand{\StringTok}[1]{\textcolor{stringcolor}{\{#1\}}}  
\newcommand{\CommentTok}[1]{\textcolor{commentcolor}{\textit{\{#1\}}}}  
\newcommand{\OtherTok}[1]{\textcolor{othercolor}{\{#1\}}}  
\newcommand{\AlertTok}[1]{\textcolor{alertcolor}{\textbf{\{#1\}}}}  
\newcommand{\FunctionTok}[1]{\textcolor{funccolor}{\{#1\}}}  
\newcommand{\FunctionTok}[1]{\{#1\}}  
\newcommand{\RegionMarkerTok}[1]{\{#1\}}  
\newcommand{\ErrorTok}[1]{\textbf{\{#1\}}}  
\newcommand{\NormalTok}[1]{\{#1\}}  
  
% JOD oriented auxiliary commands for post processing pandoc generated latex  
\newenvironment{JODGroupHeader}{}{}  
\newenvironment{JODPostProcessor}{}{}  
  
\usepackage{fancyhdr}  
\pagestyle{fancy}  
  
\ifxetex  
  \usepackage[setpagesize=false, % page size defined by xetex  
             unicode=false,      % unicode breaks when used with xetex  
             xetex]{hyperref}  
\else
```

```
\usepackage[unicode=true]{hyperref}
\fi

\hypersetup{breaklinks=true,
            bookmarks=true,
            pdfauthor={},
            pdftitle={},
            colorlinks=true,
            urlcolor=blue,
            linkcolor=magenta,
            pdfborder={0 0 0}}
\setlength{\parindent}{0pt}
\setlength{\parskip}{6pt plus 2pt minus 1pt}
\setlength{\emergencystretch}{3em} % prevent overfull lines
\setcounter{secnumdepth}{0}

% reset latex index to use four columns - default is two
% which results in lots of wasted page space in landscape
% NOTE: adjust if index names run together
% from: http://www.latex-community.org/viewtopic.php?f=4&t=1735
\usepackage{multicol}
\makeatletter
\renewenvironment{theindex}
{
  \if@twocolumn
    \@restonecolfalse
  \else
    \@restonecoltrue
  \fi
}
```



```
\fi
\setlength{\columnseprule}{0pt}
\setlength{\columnsep}{35pt}
\begin{multicols}{4}[\section*{\indexname}]
\markboth{\MakeUppercase\indexname}%
          {\MakeUppercase\indexname}%
\thispagestyle{plain}
\setlength{\parindent}{0pt}
\setlength{\parskip}{0pt plus 0.3pt}
\relax
\let\item\@idxitem}%
{\end{multicols}\if@restonecol\onecolumn\else\clearpage\fi}
\makeatother

\makeindex

\begin{document}

)
NB.>>~~~~

NB.*end-header

NB. string marking start of LaTeX indexed word - see FAKETOKENS
BEGININDEX=: '\KeywordTok{=:=:}'

NB. marks start of JOD group header in pandoc latex
BEGINJODHEADER=: '\begin{JODGroupHeader}'
```

NB. marks start of JOD group postprocessor in pandoc latex

BEGINJODPOSTP=: '\begin{JODPostProcessor}'

NB. marks the start of J script text that is not J

BEGINNOTJ=: 'NB.<<~~~~'

NB. carriage return character

CR=: 13{a.

NB. default pandoc install location

DEFAULTPANDOC=: '"C:\Program Files\Pandoc\pandoc"'

NB. string marking end of LaTeX indexed word - see FAKETOKENS

ENDINDEX=: '\KeywordTok{=..=..}'

NB. marks end of JOD group header in pandoc latex

ENDJODHEADER=: '\end{JODGroupHeader}'

NB. marks end of JOD group postprocessor in pandoc latex

ENDJODPOSTP=: '\end{JODPostProcessor}'

NB. marks the end of J script text that is not J

ENDNOTJ=: 'NB.>>~~~~'

NB. 2 and 3 j (wfl) tokens - the trailing blank of (;1{FAKETOKENS) matters!

FAKETOKENS=: <;._1 '|=:=:|=..=.. '

NB. interface word list name prefix

IFACEWORDSPFX=: 'IFACEWORDS'

NB. interface words for (jodliterate) group

IFACEWORDSjodliterate=: <;._1 ' THISPANDOC grplit ifacesection ifc setjodliterate'

NB. jodliterate author - inserted in latex \author{}

JLAUTHOR=: 'John D. Baker'

NB. suffix of jodliterate code file

JLCODEFILE=: 'code.tex'

NB. markdown text string that marks where generated group interface inserted

JLINSERTIFACEMD=: '`{~{insert_interface_md_}~}`'

NB. suffix of jodliterate overview file

JLOVIEWFILE=: 'oview.tex'

NB. name suffix of markdown overview text

JLOVIEWSUFFIX=: '_oview_tex'

NB. suffix of jodliterate title file

JLTITLEFILE=: 'title.tex'

NB. temporary latex file

LATEXTMP=: 'jltemp.tex'

NB. line feed character

LF=: 10{a.

NB. marks start of J code for pandoc -- requires pandoc with j syntax coloring

MARKDOWNHEAD=: '~~~~ { .j }'

NB. marks end J code for pandoc

MARKDOWNTAIL=: '~~~~'

NB. temporary markdown file

MARKDOWNTMP=: 'jltmp.markdown'

NB. root words for (jodliterate) group

ROOTWORDSjodliterate=: <;._1 ' DEFAULTTPANDOC IFACEWORDSjodliterate ROOTWORDSjodliterate grplit sbtokens set
>..>jodliterate'

NB. full pandoc path - use (pandoc) if on shell path

THISPANDOC=: '"C:\Program Files\Pandoc\pandoc"'

NB. white space characters

WHITESPACE=: 10 13 9 32{a.

NB. wrapped line prefix

WRAPLEAD=: '>..>'

NB. maximum number of code listing characters - adjust for given LaTeX pagesize

WRAPLIMIT=: 110

NB. invalid j string starting wrapped line - exclude '=' - trailing blank matters

WRAPPREFIX=: ')=.)=. '

NB. pandoc LaTeX fragment from (WRAPPREFIX) - these strings must correspond

WRAPPREFIXTEX=: '\RegionMarkerTok{}}\KeywordTok{=.}\RegionMarkerTok{}}\KeywordTok{=.}'

NB. retains string after first occurrence of (x)

afterstr=:] }.~ #@[+ 1&(i.~)@([E.])

NB. trims all leading and trailing blanks

alltrim=:] #~ [: -. [: (*./\ . +. */.\) ' '&=

NB. retains string before first occurrence of (x)

beforestr=:] {.~ 1&(i.~)@([E.])

betweenstrs=: 4 : 0

*NB.*betweenstrs v-- select sublists between nonnested delimiters*

NB. discarding delimiters.

NB.

NB. dyad: blcl =. (clStart;clEnd) betweenstrs cl

NB. blnl =. (nlStart;nlEnd) betweenstrs nl

NB.

```
NB. ('start';'end') betweenstrs 'start yada yada end boo hoo start ahh end'
NB.
NB. NB. also applies to numeric delimiters
NB. (1 1;2 2) betweenstrs 1 1 66 666 2 2 7 87 1 1 0 2 2
```

```
's e'=. x
llst=. ((-#s) (|.!.0) s E. y) +. e E. y
mask=. ~:/\ llst
(mask#llst) <|.1 mask#y
)
```

```
NB. boxes open nouns
boxopen=: <^(L. = 0:)
```

```
changestr=: 4 : 0
```

```
NB.*changestr v-- replaces substrings - see long documentation.
NB.
NB. dyad: clReps changestr cl
NB.
NB. NB. first character delimits replacements
NB. '/change/becomes/me/ehh' changestr 'blah blah ...'
```

```
pairs=. 2 {."(1) _2 [\ <|.1 x      NB. change table
cnt=. _1 [ lim=. # pairs
while. lim > cnt=.>:cnt do.        NB. process each change pair
  't c'=. cnt { pairs              NB. /target/change
```

```

if. +./b=. t E. y do.          NB. next if no target
  r=. I. b                    NB. target starts
  'l q'=. #&> cnt { pairs      NB. lengths
  p=. r + 0,+/\(<:# r)$ d=. q - 1 NB. change starts
  s=. * d                      NB. reduce < and > to =
  if. s = _1 do.
    b=. 1 #~ # b
    b=. ((l * # r)$ 1 0 #~ q,l-q) (,r +/ i. l)} b
    y=. b # y
    if. q = 0 do. continue. end. NB. next for deletions
  elseif. s = 1 do.
    y=. y #~ >: d r} b        NB. first target char replicated
  end.
  y=. (c $~ q *# r) (,p +/i. q)} y NB. insert replacements
end.
end. y                          NB. altered string
)

```

NB. character table to newline delimited list

```
ctl=: }.@(@1&(",1)@(-.@(*./\."1@(&' ' @])))) # ,@((10{a.)&(",1)@]))
```

```
cutnestidx=: 4 : 0
```

*NB.*cutnestidx v-- cut list into nested runs and other.*

NB.

*NB. Nested runs are delimited by begin and end tags. This verb is
NB. oriented toward XML parsing where typical begin end tags are*

```

NB. <ul> </ul> and tags with attributes like: <hoo boy="2">
NB. </hoo>
NB.
NB. This verb can process numeric lists but care must be taken to
NB. insure the pad item (1{.0$y) does not match begin and end
NB. values.
NB.
NB. dyad: (ilIdx ;< blcl) =. (clStart;clEnd) cutnestidx cl
NB.       (ilIdx ;< blnl) =. (nlStart;nlEnd) cutnestidx nl
NB.
NB. xml=. 'yada <ol><li>one</li><ol><li>sub one</li></ol></ol> boo'
NB. ('<ol'; '</ol>') cutnestidx xml
NB.
NB. 88 99 cutnestidx (i.5),88,(10?10),99 88 5 5 5 5 5 99

if. #y do.
  's e'=. ,&.> x          NB. start end lists
  ut=. 1{.0$y             NB. padding
  assert. -.s -: e        NB. they must differ
  assert. -. (s -:ut) +. e -:ut
  sp=. s E. ut=.y,ut       NB. start mask

  NB. quit if no delimiters
  if. -.1 e. sp do. (i.0);<<y return. end.

  ep=. e E. ut            NB. end mask
  assert. (+/sp) = +/ep    NB. basic balance

```



```

dp=. sp + - ep          NB. start end marks
assert. 0 *./ . <: +/\ dp  NB. nested balance
ep=. I. _1=dp [ sp=. I. 1=dp  NB. start end indexes
ut=. +/\dp -. 0          NB. scanned marks
dp=. /:~ sp,ep          NB. all indexes
sp=. (firstones 1<:ut)#dp  NB. starts of nested
ep=. (#e)+(0=ut)#dp      NB. starts of other
dp=. /:~ ~.0,sp,ep      NB. cut starts
ut=. }: 1 dp} (>:#y)#0   NB. cut mask
(dp i. sp);<ut <;.1 y    NB. nest indexes cut list
else.
  (i.0);<<y             NB. empty arg result
end.
)

NB. double quotes - doubles internal " quotes like (quote)
dbquote=: '""&,@(&'""')@(#~ >:@(=&'""'))

NB. quote unquoted strings containing blanks: dbquoteuq 'c:\blanks in\paths bitch'
dbquoteuq=: ]`dbquote@.([[: -. '""'&-:@({: , {.)} *. ' ' e. ])

decomm=: 3 : 0

NB.*decomm v-- removes comments from j words. The (x) argument
NB. specifies whether all blank lines are removed or retained.
NB.
NB. monad: decomm ctWord

```

```
NB.
NB.  decomm jcr 'decomm'  NB. decomment self
NB.
NB. dyad:  pa decomm ctWord
NB.
NB.  1 decomm jcr 'decomm'  NB. remove blanks (default)
NB.  0 decomm jcr 'decomm'  NB. retain all blank lines

1 decomm y
:
NB. mask of unquoted comment starts
c=. ($y)$'NB.' E. ,y
c=. +./\"1 c > ~:/\"1 y e. ''''

NB. ,, work around for j8.05 bug - remove when fixed
y=. ,,y

NB. blank out comments
y=. ' ' (I. ,c)} y
y=. y $~ $c

NB. remove blank lines - default
if. x do. y #~ y +./ . ~: ' ' end.
)

NB. boxes UTF8 names
fboxname=: ([: < 8 u: >) ::]
```

NB. erase files - cl / blcl of path file names

```
ferase=: 1!:55 ::(_1:)@(fboxname&>)@boxopen
```

NB. 1 if file exists 0 otherwise

```
fexist=: 1:@(1!:4) ::0:@(fboxname&>)@boxopen
```

NB. 0's all but first 1 in runs of 1's - like (firstone) but differs for nulls

```
firstones=: > (0: , }:)
```

```
formifacetex=: 3 : 0
```

*NB.*formifacetex v-- formats hyperlinked and highlighted interface words.*

NB.

NB. monad: cl =. formifacetex blclIwords

NB. jod refs !()=. get*

```
head=. '\begin{Shaded}',LF,'\begin{Highlighting}[]',LF
```

```
tail=. '\end{Highlighting}',LF,'\end{Shaded}',LF
```

```
ctok=. '\CommentTok{'
```

```
ntok=. '\NormalTok{'
```

```
href=. '\hyperlink{'
```

NB. fetch current short descriptions !()=. WORD_ajod_ EXPLAIN_ajod_*

```
'rc tab'=. (WORD_ajod_,EXPLAIN_ajod_) get y
```

```
words=. 0 {"1 tab
```

```
desc=. 1 {"1 tab
```

NB. set hyperlinks on words - colors on comments

```
words=. (<href) ,&.> words ,&.> (<'{}',ntok) ,&.> (<"1 (>words),"1 ' ' ' ,&.> <'{}' '
desc=. (<ctok) ,&.> (alltrim&.> desc) ,&.> '}'
tex=. ;words ,&.> desc ,&.> LF
head,tex,tail
)
```

```
formtexindexes=: 3 : 0
```

*NB.*formtexindexes v-- format latex index commands from global marks.*

NB.

NB. monad: blcl =. formtexindexes blclMarked

NB. extract =: =. marked text

```
inames=. ;@('{}'&betweenstrs)&.> (-#ENDINDEX) }.&.> (#BEGININDEX) }.&.> y
```

NB. find any indirect ()=: and multiple '=: assignments

```
'pma pia'=. I.&.> <"1 ' ' ' ' =/ {:@(-.&' ' ')&> inames
```

NB. form latex index commands

```
indexes=. (<'\\AlertTok{=:}\\index{' ,&.> inames ,&.> (<'@\\texttt{' ,&.> inames ,&.> <'{}' ')
```

NB. replace indirect and multiple assignments with fixed proxies

```
indexes=. ((#pma) # <'\\AlertTok{=:}\\index{00multiple@\\texttt{' '...'=:}}') pma} indexes
```

```
indexes=. ((#pia) # <'\\AlertTok{=:}\\index{01indirect@\\texttt{(...)=:}}') pia} indexes
```

NB. adjust j locative chars _ they give latex indexing grief

NB. later versions of pandoc handle this case

```
NB. if. #pos=. I. ' ' &e.& > indexes do.  
NB.   indexes=. ('#_#\_' &change& > pos{indexes) pos} indexes  
NB. end.
```

```
indexes  
)
```

```
NB. size of file in bytes  
fsize=: 1!:4 ::(_1:)@(fboxname&>)@boxopen
```

```
NB. opens and catenates boxed lists on the last axis  
fuserows=: >@((>@[ , "1 >@]) /)
```

```
gbodylatex=: 3 : 0
```

```
NB.*gbodylatex v-- group body latex.  
NB.  
NB. monad: clTex =. gbodylatex clGroupname
```

```
if. #mtxt=. markdfrgroup y do. latexfrmarkd mtxt else. '' end.  
)
```

```
gheadlatex=: 3 : 0
```

```
NB.*gheadlatex v-- group header latex.  
NB.  
NB. monad: clTex =. gheadlatex clGroupname
```

```
if. #mtxt=. markdfgrhead y do.  
  BEGINJODHEADER,LF,(tlf latexfrmarkd mtxt),ENDJODHEADER,2#LF  
else.  
  ''  
end.  
)
```

```
gpostlatex=: 3 : 0
```

```
NB.*gpostlatex v-- group post processor latex.
```

```
NB.
```

```
NB. monad: clTex =. gpostlatex clGroupname
```

```
if. #mtxt=. markdfgrpost y do.  
  BEGINJODPOSTP,LF,(tlf latexfrmarkd mtxt),ENDJODPOSTP  
else.  
  ''  
end.  
)
```

```
grouplatex=: 3 : 0
```

```
NB.*grouplatex v-- group latex with pandoc syntax highlighting.
```

```
NB.
```

```
NB. monad: clTex =. grouplatex clGroupname
```

```
NB.
```

```
NB.  NB. requires open JOD dictionary with a 'jod' group
NB.  gtex=. grouplatex 'jod'
NB.
NB. dyad:  clTex =. paIndex grouplatex clGroupname
NB.
NB.  0 grouplatex 'jod' NB. do not replace marks with index

1 grouplatex y
:
NB. jod refs !(*)=. badrc_ajod_ grp jderr_ajod_
if. badrc_ajod_ gnames=. grp y do. gnames return. end.

ltx=. x indexwraplatex (gheadlatex ; gbodylatex ; gpostlatex) y
'\section{\texttt{'',(alltrim y),'} Source Code}',LF,LF,ltx
)

grplit=: 3 : 0

NB.*grplit v-- make latex for group (y).
NB.
NB. monad:  (paRc ; blclTeXfiles) =. grplit clGroupname
NB.
NB.  grplit 'jodliterate' NB. document self
NB.
NB. dyad:  (paRc ; blclTeXfiles) =. paOw grplit glGroupname
NB.
NB.  NB. do not overwrite root tex - allows for latex tweaking
NB.  0 grplit 'jodliterate'
```

```
1 grplit y
:
NB. jod refs !(*)=. badrc_ajod_ get grp jderr_ajod_ ok_ajod_
try.

if. 3~:(4!:(0) <'badrc_ajod_' do. 0;'!error: jod is not loaded' return. end.
if. 0=#JLDIRECTORY do. 0;'!error: working directory is not set' return. end.

NB. group must exist
if. badrc_ajod_ glist=. GROUP_ajod_ grp group=. y -. ' ' do. glist return. end.

NB. default overview
ohd=. ('/~#~group~#~/',alltrim y) changestr JLOVIEWTEX [ gdoc=. ''
iwords=. ifacewords group

NB. overview documents are either markdown/latex group headers or stored LaTeX macros
if. badrc_ajod_ gdoc=. MACRO_ajod_ get group,JLOVIEWSUFFIX do.
  NB. no stored LaTeX generate LaTeX from group document markdown/latex
  if. badrc_ajod_ gdoc=. (GROUP_ajod_,DOCUMENT_ajod_) get group do. gdoc return. end.
  if. #gdoc=. ;{:,>1{gdoc do.
    NB. insert interface md based on IFACEWORDSgroup
    if. +./JLINSERTIFACEMD E. gdoc do.
      gdoc=. ('/',JLINSERTIFACEMD,'/',ifacemarkd group) changestr gdoc
    end.
    gdoc=. latexfrmarkd gdoc
    ifstr=. ifacesection group
```



```
    if. (+./ifstr E. gdoc) *. (<IFACEWORDSPFX,group) e. glist do.
      gdoc=. iwords setifacelinks ifstr;gdoc
    end.
  end.
else.
  NB. stored macro LaTeX - no adjustments
  gdoc=. ;{:,>1{gdoc
end.

NB. root .tex file - gets group name
wdir=. JLDIRECTORY
jlroot=. wdir,group,'.tex'
if. chroot=. x -: 1 do.
  root=. ('/~#~group~#~/',group) changestr JLGRPLITTEX
  (toJ root) writeas jlroot
end.

NB. author title .tex file
agstrs=. ('/~#~author~#~/',(alltrim JLAUTHOR),'/~#~group~#~/',alltrim y
(toJ agstrs changestr JLTITLETEX) writeas jltitle=. wdir,group,JLTITLEFILE

NB. group overview .tex file
ohd=. ohd,LF,gdoc
(toJ ohd) writeas jloview=. wdir,group,JLOVIEWFILE

NB. group build batch script - latex utils that compile generated files
jlbuildtex=. ('/~#~group~#~/',alltrim y) changestr JLBUILDTEX
```

```
(toJ jlbuildtex) writeas jlbuildbat=. wdir,group, '.bat'

NB. group source code .tex - return file names
gltx=. grouplatex group
gltx=. iwords setifacetargs gltx
(toJ gltx) writeas jlcode=. wdir,group,JLCODEFILE
ok_ajod_ (-.chroot) }. jlroot;jlttitle;jloview;jlcode;jlbuildbat

catchd.
  0;!'error: (grplit) failure - last J error ->';13!:12 ''
end.
)

ifacemarkd=: 3 : 0

NB.*ifacemarkd v-- generate word interface markdown section.
NB.
NB. monad: clMd =. ifacemarkd clGroupName
NB.
NB. ifacemarkd 'jodliterate'

LF,'~~~~{ .j }',LF,(2 ifc y),LF,'~~~~',LF
)

ifacesection=: 3 : 0

NB.*ifacesection v-- interface section summary string.
```

```
NB.
NB. This verb produces the interface section summary string. For
NB. (jodliterate) to include an updated hyperlinked interface
NB. summary it must find this string in generated latex. Edit
NB. this verb if you change the section layout.
NB.
NB. monad: cl =. ifacesection clGroupname

'\subsection{\texttt{' ,y,'} Interface}'
)

ifacewords=: 3 : 0

NB.*ifacewords v-- return interface word list.
NB.
NB. Assume the interface is out of date fetch current definition
NB. from dictionary. We need the value not the storage
NB. representation so define it in the JOD scratch object.
NB.
NB. monad: blcl =. ifacewords clGroupname

NB. jod refs !(*)=. get
iname=. (IFACEWORDSPFX,y) -. ' '
(;SO__JODobj) get iname
iname=. iname,'__SO__JODobj'
words=. ". iname
words [ (4!:55) <iname
)
```

```
ifc=: 3 : 0
```

```
NB.*ifc v-- format interface comment text.
```

```
NB.
```

```
NB. Looks up interface words of a group and formats text for
```

```
NB. insertion into group headers and postprocessors.
```

```
NB.
```

```
NB. monad: ifc clGroupName
```

```
NB. dyad: iaOption ifc clGroupName
```

```
1 ifc y
```

```
:
```

```
NB. require 'jod' !(*)=. badrc_ajod_ get jderr_ajod_ badcl_ajod_ badil_ajod_
```

```
if. badcl_ajod_ y do. jderr_ajod_ 'invalid group name' return.
```

```
else.
```

```
iface=. 'IFACEWORDS',alltrim y
```

```
end.
```

```
x=. {. ,x [ msg=. 'invalid ifc options'
```

```
if. badil_ajod_ x do. jderr_ajod_ msg return. end.
```

```
if. -.x e. i.3 do. jderr_ajod_ msg return. end.
```

```
NB. set comment style (header, postprocessor)
```

```
cpx=. ; x { (<'NB. ' ;' - '),(<'NB. ' ;' NB. '),<' ;' NB. '
```

```
NB. define interface list in jod scratch locale
```

```
NB. !(*)=. SO__JODobj erase__SO__JODobj locsfx_ajod_ nl__SO__JODobj
```

```

if. badrc_ajod_ rc=. (;SO__JODobj) get iface do. rc return.
elseif. ilist=. ".iface , ;locsfx_ajod_ ;SO__JODobj
      erase__SO__JODobj nl__SO__JODobj i. 4
      badrc_ajod_ rc=. 0 8 get /:~ ~.ilist do. rc return.
elseif. 0=#txt=. >1{rc do. jderr_ajod_ 'no interface words' return.
elseif.do.
  ctl fuserows >&.> <"1 |: ((#txt)#,:cpx) ,&.> txt
end.
)

indexgrouptex=: 3 : 0

NB.*indexgrouptex v-- insert index commands in pandoc highlight group latex.
NB.
NB. dyad: cl =. clGroupName indexgrouptex clTex

'pos ltx'=. (BEGININDEX;ENDINDEX) cutnestidx y
if. #pos do. ; (formtexindexes pos{ltx} pos} ltx else. y end.
)

indexwraplatex=: 4 : 0

NB.*indexwraplatex v-- insert index commands and handle spurious blanks.
NB.
NB. dyad: clLatex =. paIndex indexwraplatex clLatex

ltx=. ]`indexgrouptex@.(1 -: x) ; tlf&.> y -. a:

```

NB. wrap prefix final LaTeX

```
wpfx=. '\AlertTok{' ,WRAPLEAD,'}'
```

NB. convert wrap marks to LaTeX fragments - handle trailing blank first

```
ltx=. ('#',WRAPPREFIXTEX,' ', '#',wpfx) changestr ltx
```

```
ltx=. ('#',WRAPPREFIXTEX,'#',wpfx) changestr ltx
```

NB. remove spurious normal token blanks

```
sprb=. wpfx, '\NormalTok{'
```

```
(' #',sprb,' #' ,sprb) changestr ltx
```

```
)
```

NB. standarizes J path delimiter to unix/linux forward slash

```
jpathsep=: '/'&(('\' I.@:= ]))
```

```
jtokenize=: 3 : 0
```

*NB.*jtokenize v-- tokenizes j text with (wfl).*

NB.

NB. Similar to (;:~>)@(<:~) but preserves whitespace and is

NB. able to parse invalid j text containing open quotes. When an

NB. open quote is encountered it is treated like an unterminated

NB. string.

NB.

NB. monad: bblcl =. jtokenize clJtext

NB.

NB. jtokenize 5!:5 <'jtokenize'

```
ct=. wfl y,LF
(ct -:&> <,LF) <|.2 ct
)
```

```
latexfrmarkd=: 3 : 0
```

*NB.*latexfrmarkd v-- latex from markdown using pandoc.*

NB.

NB. monad: clTex =. latexfrmarkd clMarkdown

NB. require 'task' !()=. shell*

```
if. #y do.
```

```
  ferase mrktmp=. JLDIRECTORY,MARKDOWNTMP
```

```
  ferase ltxtmp=. JLDIRECTORY,LATEXTMP
```

```
  (toJ y) writeas mrktmp
```

NB. highlighting style is overridden in latex preamble

```
  shell THISPANDOC,' --highlight-style=tango ',(dbquoteuq mrktmp),' -o ',dbquoteuq ltxtmp
```

```
  assert. 0 < fsize ltxtmp
```

```
  tex=. read ltxtmp
```

```
  tex [ ferase ltxtmp [ ferase mrktmp
```

```
else.
```

```
  y
```

```
end.
```

```
)
```

```
markdfrghead=: 3 : 0
```

```
NB.*markdfrghead v-- markdown text from group header.
NB.
NB. monad: cl =. markdfrghead clGroupname
NB.
NB.   mtxt=. markdfrghead 'jod'
NB.   (toHOST mtxt) write 'c:/temp/jodhdr.markdown'

NB. jod refs !(*)=. badrc_ajod_ get HEADEND_ajodmake_
if. badrc_ajod_ hdr=. 2 get y do. hdr return. end.
if. 0=#hdr=. ;1{,>1{hdr      do. '' return. end.
hdr=. hdr,LF,HEADEND_ajodmake_

NB. handle any non j code regions
'idx chd'=. (BEGINNOTJ;ENDNOTJ) cutnestidx hdr

if. #idx do.
  psj=. idx -.~ i.#chd
  chd=. (markgnonj&.> idx{chd) idx} chd
  chd=. (markgassign&.> psj{chd) psj} chd
  hdr=. ;chd
else.
  hdr=. markgassign hdr
end.

if. #hdr do. markdj hdr else. '' end.
)
```



```
markdfrgpost=: 3 : 0
```

```
NB.*markdfrgpost v-- markdown from group post processor.
```

```
NB.
```

```
NB. monad: clMarkdown =. markdfrgpost clGroupname
```

```
NB. jod refs !(*)=. get
```

```
'rc post'=. 2 {. 4 get 'POST_',y -. ' '
```

```
if. rc do. markdj markgassign ; {: , post else. '' end.
```

```
)
```

```
markdfrgroup=: 3 : 0
```

```
NB.*markdfrgroup v-- markdown text from group.
```

```
NB.
```

```
NB. monad: cl =. markdfrgrp clGroupname
```

```
NB.
```

```
NB. txt=. markdfrgroup 'jod'
```

```
NB. (toHOST txt) write 'c:/temp/jcode.markdown'
```

```
NB. jod refs !(*)=. badrc_ajod_ get gdeps grp
```

```
if. badrc_ajod_ gnl=. grp y do. gnl return. end.
```

```
if. badrc_ajod_ gdp=. gdeps y do. gdp return. end.
```

```
if. #gnl=. (gnl -. gdp) -. a: do. markdfrwords gnl else. '' end.
```

```
)
```

```
markdfrwords=: 3 : 0
```

```
NB.*markdfrwords v-- markdown text from word list.
NB.
NB. This verb takes a blcl of JOD word names and returns a UTF-8
NB. encoded cl of word source code in markdown format. Markdown
NB. is a simple but versatile text markup format that is almost
NB. ideal for documenting program source code, see:
NB.
NB. http://daringfireball.net/projects/markdown/
NB.
NB. monad: clMarkdown =. markdfrwords blclWords
NB.
NB.   markdfrwords ;:'go ahead mark us up'
NB.
NB.   NB. markdown text from JOD group words
NB.   mtxt=. markdfrwords }. grp 'jod'

NB. require 'jod' !(*)=. WORD_ajod_ NVTABLE_ajod_ badrc_ajod_ get wttxt__MK__JODobj
if. badrc_ajod_ src=. (WORD_ajod_,NVTABLE_ajod_) get y do. src return. end.

NB. commented source code (name,source) table.
if. badrc_ajod_ src=. 0 0 1 wttxt__MK__JODobj >1{src do. src
else.
  src=. markgassign&.> {:"1 >1{src
  NB. similar to (markdj) but faster here
  utf8 ; (<LF,MARKDOWNHEAD,LF) ,&.> src ,&.> <LF,MARKDOWNTAIL,LF
end.
)
```

```
markdj=: 3 : 0
```

```
NB.*markdj v-- mark j code for markdown.
```

```
NB.
```

```
NB. monad: clM =. markdj clJ
```

```
utf8 (LF,MARKDOWNHEAD,LF),(tlf y),MARKDOWNTAIL,LF
)
```

```
markgassign=: 3 : 0
```

```
NB.*markgassign v-- mark j code for latex indexing.
```

```
NB.
```

```
NB. This verb tokenizes j code and replaces all global  
NB. assignments with syntactically incorrect j strings that will  
NB. be transformed by pandoc into easily located latex strings  
NB. that will then be converted by a post pandoc processor into  
NB. valid latex index commands. This works because regex based  
NB. pandoc coloring does not "understand" j's parsing rules.
```

```
NB.
```

```
NB. monad: cl =. markgassign clJcode
```

```
NB.
```

```
NB. jcode=. 'markgassign=: ' , 5!:5 <'markgassign'
```

```
NB. markgassign jcode
```

```
if. 0=#jcode=. y -. CR do. y return. end.
```

```
jcode=. WRAPLIMIT wrapvrblong jcode
```

```
jtokens=. jtokenize jcode
```

```

NB. only interested in global assignment lines
if. #gix=. I. ; (<'=:') e. L: 1 jtokens do.
  jgl=. gix{jtokens
  jshp=. $jat=. >jgl
  jix=. I. jat = <'=: ' [ jat=. ,jat
  NB. extract global assignments
  NB. ignoring interleaving blanks
  jat2=. (jat -.&.> ' ') -. a:
  anames=. (<:I.(<'=:') -:&> jat2){jat2
  NB. (0{FAKETOKENS) and (1{FAKETOKENS) are invalid in j
  faketoks=. (0{FAKETOKENS) ,&.> anames ,&.> 1{FAKETOKENS
  jat=. <"1 jshp $ faketoks jix} jat
  jat=. (#&> jgl) {.&.> jat
  NB. adjust last LF
  (-LF={:y) }. ;;jat gix} jtokens
else.
  y
end.
)

```

```
markgnonj=: 3 : 0
```

*NB.*markgnonj v-- mark non j code region global assignments.*

NB.

NB. Non J code is often inserted in J scripts as character nouns

NB. using explicit multi-line '0 : 0' definitions. This verb

NB. marks the assigned noun name. Only '=: 0 : 0' will be found

```
NB. and marked.
NB.
NB. verbatim:
NB.
NB. IamFound =: 0 : 0
NB. .... non j code ...
NB. )
NB.
NB. monad: cl =. markgnonj clNonj

ct=. <.2 tlf y
mrk=. '=:0:0'
pos=. I. mrk&~:~> (-#mrk)&{.> ct -.> <WHITESPACE
ct=. ;(LF ,>~ markgassign.> pos{ct) pos} ct
(-LF={:y) }. ct
)

NB. reads a file as a list of bytes
read=: 1!:1&[<@.(32>@.(3!:0)))

NB. trim right (trailing) blanks
rtrim=: ] #~ [: -. [: *./\ . ' ' " _ = ]

NB. blcl of nonempty noncomment J cl tokens
sbtokens=: a: -.~ (<13 10 9{a.) -.>~ [: alltrim.> [: wfl [: ctl [: decomm [: ];_1 (10{a.) , (13{a.) -.~ ]

setifacelinks=: 4 : 0
```

```
NB.*setifacelinks v-- set hyperref links in any overview
NB. interface words section.
NB.
NB. dyad: cl =. blcliwords setifacelinks (clIfstr ; clTex)

'ifstr tex'=. y
rmrk=. '\end{Shaded}'
head=. ifstr&beforestr tex
tail=. ifstr&afterstr tex

if. +./rmrk E. tail do.
  ifbk=. formifacetex x
  tail=. rmrk&afterstr tail
  head,ifstr,(2#LF),ifbk,tail
else.
  tex
end.
)

setifacetargs=: 4 : 0

NB.*setifacetargs v-- set hyperlink targets in group latex.
NB.
NB. dyad: cl =. blcliwords setifacetargs clTex

targs=. (<'\\NormalTok{' ) ,&.> x ,&.> <'\\AlertTok{=:}\\index'
rstrs=. (<'\\hypertarget{' ) ,&.> x ,&.> (<'{'\\NormalTok{' ) ,&.> x ,&.> <''}\\AlertTok{=:}\\index'
chgs=. ;(<'#' ,&.> targs) ,&.> '<' ,&.> rstrs
```

```
chgs changestr y
)

setjodliterate=: 3 : 0

NB.*setjodliterate v-- prepare LaTeX processing - sets out directory writes preamble.
NB.
NB. monad: (paRc ; clDir) =. setjodliterate clWorkingDir | zl
NB.
NB.   setjodliterate 'c:\temp'           NB. windows
NB.   setjodliterate '/home/john/temp'   NB. linux
NB.
NB.   NB. use the current JOD put dictionary document directory
NB.   setjodliterate ''

try.

if. 3~:(4!:0) <'badrc_ajod_' do. 0; '!error: jod is not loaded' return. end.
if. 0 = #DPATH__ST__JODobj do. 0; '!error: no open jod dictionaries' return. end.

NB. if the path is empty use the current put dictionary document directory !(*)=. dob
if. 0 e. $y do. y=. DOC__dob [ dob=: {:{.DPATH__ST__JODobj end.

NB. profile (*)=. IFWIN
JLDIRECTORY_ajodliterate=: jpathsep`winpathsep@.(IFWIN) tslash2 y

NB. write main latex preamble once per directory
preamble=. 'JODLiteratePreamble.tex'
```

```
if. -.fexist JLDIRECTORY,preamble do.
  (toJ JODLiteratePreamble) writeas JLDIRECTORY,preamble
end.
1;JLDIRECTORY

catchd.
  0;!'error: (setjodliterate) failure - last J error ->';13!:12 ''
end.
)

NB. appends trailing line feed character if necessary
tlf=: ] , ((10{a.})"_ = {:) }. (10{a.})"_

NB. converts character strings to J delimiter LF
toJ=: ((10{a.}) I.@(e.&(13{a.}))@] } )@:(#~ -.@((13 10{a.})&E.@,))

NB. appends trailing / iff last character is not \ or /
tslash2=: ([: - '\/' e.~ {:) }. '/' ,~ ]

NB. character list to UTF-8
utf8=: 8&u:

NB. standardizes path delimiter to windows back \ slash
winpathsep=: '\ '&(( '/' I.@:= ]))

wrapvrblong=: 3 : 0
```



```
NB.*wrapvrblong v-- wraps verbatim text lines with length > (x).
NB.
NB. Wraps lines with length > (x) and prefixes each wrapped line
NB. with the syntactically invalid j string ')=.)=.' (WRAPPREFIX)
NB. This string is transformed by pandoc into an easily found
NB. sequence of LaTeX commands.
NB.
NB. monad: cl =. wrapvrblong clTxt
NB. dyad: cl =. iaLength wrapvrblong clTxt
```

```
WRAPLIMIT wrapvrblong y
```

```
:
```

```
NB. always trim trailing blanks
```

```
ct=. <@rtrim;._2 tlf y -. CR
```

```
NB. only wrap lines exceeding limit
```

```
if. #pos=. 1. x < #&> ct do.
```

```
  wlen=. x-#WRAPLEAD
```

```
  wt=. (-wlen) (<\)&.> pos{ct
```

```
  slen=. 1&,@:<:@#&.> wt
```

```
NB. lead wrapped lines with prefix
```

```
wt=. (slen #&.> <(<''),<LF,WRAPPREFIX) ,.&.> wt
```

```
wt=. a: -.~ L: 1 ,&.> wt
```

```
NB. last wrapped line LF terminated
```

```
wt=. wt , L: 1 <LF
```

```
nwpos=. (i.#ct) -. pos
```

```
ct=. ((nwpos{ct) ,&.> LF) nwpos} ct
```

```
;;wt pos} <"0 ct
else.
  (-LF~::~:y) }. ; ct ,&.> LF
end.
)

NB. write file as list of bytes - throws unambiguous error on failure
writeas=: (1!:2 ]`<@.(32&>@3!:0))) ::([: 'cannot write file'&(13!:8) 1:)

NB.POST_jodliterate post processor (-.)=:

smoutput IFACE=: (0 : 0)
NB. (jodliterate) interface word(s):
NB. -----
NB. THISPANDOC      NB. full pandoc path - use (pandoc) if on shell path
NB. grplit          NB. make latex for group (y)
NB. ifacesection    NB. interface section summary string
NB. ifc              NB. format interface comment text
NB. setjodliterate  NB. prepare LaTeX processing - sets out directory writes preamble
)

cocurrent 'base'
coinsert 'ajodliterate'

(3 : 0) ''
if. +./@('pandoc'&E.) panver=. ;0{ <;._2 tlf (shell THISPANDOC,' --version') -. CR do.
  smoutput 'NOTE: adjust pandoc path if version ('panver,') is not >= 2.9.1.1'
```

```
else.  
  smoutput 'ERROR: pandoc not set - adjust THISPANDOC'  
  smoutput 'THISPANDOC_ajodliterate=: ''pandoc'' NB. use when proper pandoc on path'  
end.  
)
```

Index

afterstr, 21
alltrim, 21

beforestr, 21
BEGININDEX, 17
BEGINJODHEADER, 17
BEGINJODPOSTP, 18
BEGINNOTJ, 18
betweenstrs, 21
boxopen, 22

changestr, 22
CR, 18
ctl, 23
cutnestidx, 23

dbquote, 25
dbquoteuq, 25
decomm, 25
DEFAULTPANDOC, 18
dob, 47

ENDINDEX, 18
ENDJODHEADER, 18
ENDJODPOSTP, 18
ENDNOTJ, 18

FAKETOKENS, 18
fboxname, 26

ferase, 27
fexist, 27
firstones, 27
formifacetex, 27
formtexindexes, 28
fsize, 29
fuserows, 29

gbodylatex, 29
gheadlatex, 29
gpostlatex, 30
grouplatex, 30
grplit, 31

IFACE, 50
ifacemarkd, 34
ifacesection, 34
ifacewords, 35
IFACEWORDSjodliterate,
19
IFACEWORDSPFX, 19
ifc, 36
indexgrouptex, 37
indexwraplatex, 37

JLAUTHOR, 19
JLBUILDTEX, 9
JLCODEFILE, 19

JLDIRECTORY, 9
JLDIRECTORY_ajodliterate_,
47
JLGRPLITTEX, 10
JLINSERTIFACEMD, 19
JLOVIEWFILE, 19
JLOVIEWSUFFIX, 19
JLOVIEWTEX, 9
JLTITLEFILE, 19
JLTITLETEX, 9
JODLiteratePreamble, 11
jpathsep, 38
jtokenize, 38

latexfrmarkd, 39
LATEXTMP, 19
LF, 20

markdfrghead, 39
markdfrgpost, 41
markdfrgroup, 41
markdfrwords, 41
markdj, 43
MARKDOWNHEAD, 20
MARKDOWNTAIL, 20
MARKDOWNTMP, 20
markgassign, 43

markgnonj, 44

read, 45
ROOTWORDSjodliterate,
20
rtrim, 45

sbtokens, 45
setifacelinks, 45
setifacetargs, 46
setjodliterate, 47

THISPANDOC, 20
tlf, 48
toJ, 48
tslash2, 48

utf8, 48

wfl, 8
WHITESPACE, 20
winpathsep, 48
WRAPLEAD, 20
WRAPLIMIT, 21
WRAPPREFIX, 21
WRAPPREFIXTEX, 21
wrapvrblong, 48
writeas, 50