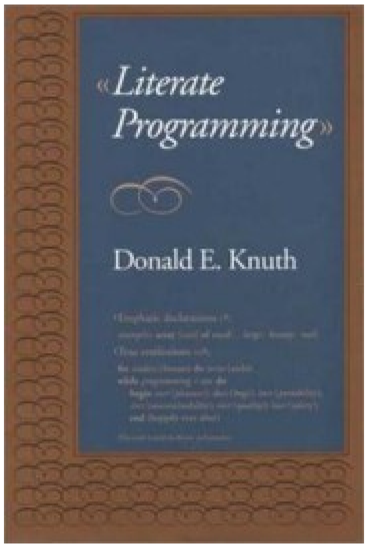
Using jodliterate

May 18, 2020

Using `jodliterate`

The JODSOURCE addon, (a part of the JOD system), contains a handy *literate programming* tool that enables the generation of *beautiful* J source code documents.

The *Bible*, *Koran* and *Bhagavad Gita* of Literate Programming is Donald Knuth's masterful tome of the same name.



Knuth applied Literate Programming to his own TEX systems and produced what many consider enduring masterpieces of program documentation.

`jodliterate` is certainly not worthy of TEX level accolades but with a little work it's possible to produce fine documents. This J kernel notebook outlines how you can install and use `jodliterate`. Jupyter notebooks are typically executed but to accomdate J users that do hot have Jupyter this notebook is also available on GitHub as a static PDF document.

Notebook Preliminaries

```
[1]: NB. show J kernel version
     9!:14 ''
```

```
j901/j64avx2/windows/release-e/commercial/www.jsoftware.com/2020-01-29T11:17:19
```

[2]:
```
NB. load JOD in a clear base locale
load 'general/jod' [ clear ''

NB. The distributed JOD profile automatically RESETME's.
NB. To safely use dictionaries with many J tasks they must
NB. be READONLY. To prevent opening the same put dictionary
NB. READWRITE comment out (dpset) and restart this notebook.
dpset 'RESETME'

NB. Converting Jupyter notebooks to LaTeX is
NB. simplified by ASCII box characters.
portchars ''

NB. Verb to show large boxed displays in
NB. the notebook without ugly wrapping.
sbx_ijod_=: ' ... ' ,"1~ 75&{."1@":
```

Installing `jodliterate`

To use `jodliterate` you need to:

1. Install a current version of J.
2. Install the J addons JOD, JODSOURCE and JODDOCUMENT.
3. Build the JOD development dictionaries from JODSOURCE.
4. Install a current version of pandoc.
5. Install a current version of TEX and LATEX.
6. Make the `jodliterate` script.
7. Run `jodliterate` on a JOD *group* with pandoc compatible markdown or LATEX document fragments.
8. Compile the LATEX files of the previous step to produce a PDF.

When presented with long lists of program prerequistes my impluse is to *run!* Life is too short for configuration wars. Everything should be easy. Installing `jodliterate` requires more work than phone apps but compared to enterprise installations setting up `jodliterate` is trivial. We'll go through it step by step.

Step 1: Install a current version of J

J is freely availabe at jsoftware.com. J installation instructions can be found on the J Wiki on this page.

Follow the appropriate instructions for your OS.

Note: JOD runs on Windows, Linux and MacOS versions of J, hence these are the only platforms that currently support `jodliterate`.

Step 2: Install the J addons JOD, JODSOURCE and JODDOCUMENT

After installing J install the J addons. J addons are installed with the J package manager pacman. Pacman has three IDE flavors: a command line flavor and two GUI flavors. The GUI flavors depend on JQT or JHS. The GUI flavors of pacman are only available on some versions of J whereas the command line version is part of the base J install and is available on all platforms.

I install all the addons. I recommend that you do the same.

JOD depends on some J modules like `jfiles`, `regex` and `task` that are sometimes distributed as addons. If you install all addons JOD's modules and dependents are both installed.

Installing addons with command line `pacman` Start J and do:

[3]:
```
NB. install J addons using the command line version of pacman

load 'pacman'    NB. load pacman jpkg services
```

[4]:
```
'help' jpkg ''   NB. what can you do for me?
```

```
Valid options are:
 history, install, manifest, remove, reinstall, search,
 show, showinstalled, shownotinstalled, showupgrade,
 status, update, upgrade

https://code.jsoftware.com/wiki/JAL/Package_Manager/jpkg
```

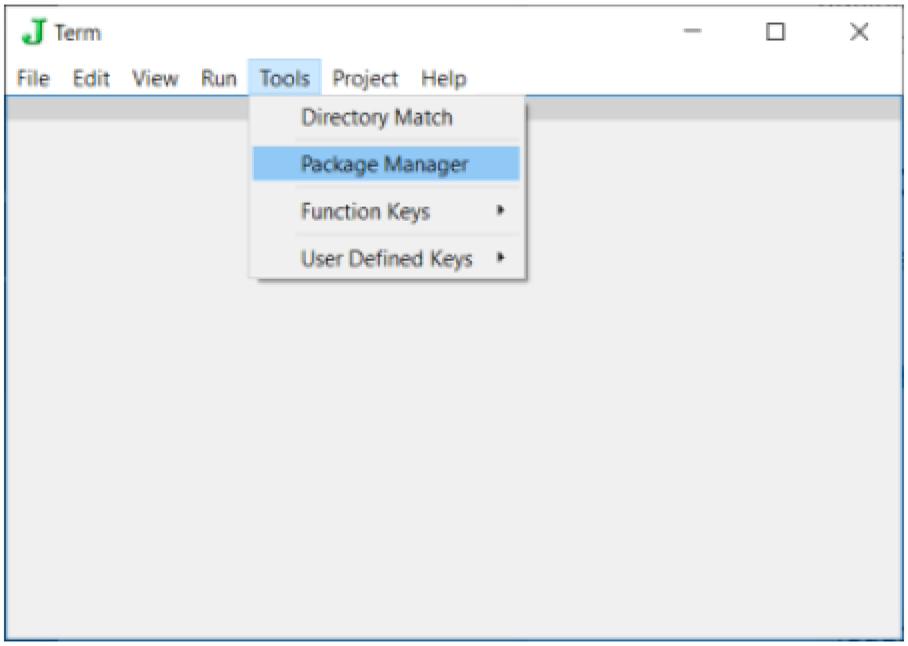
[5]:
```
NB. install all addons - see https://code.jsoftware.com/wiki/Pacman for details

NB. 'install' jpkg '*'  NB. uncomment to run if you have not installed addons
```

[6]:
```
5 {. 'showinstalled' jpkg '' NB. first five installed addons
```

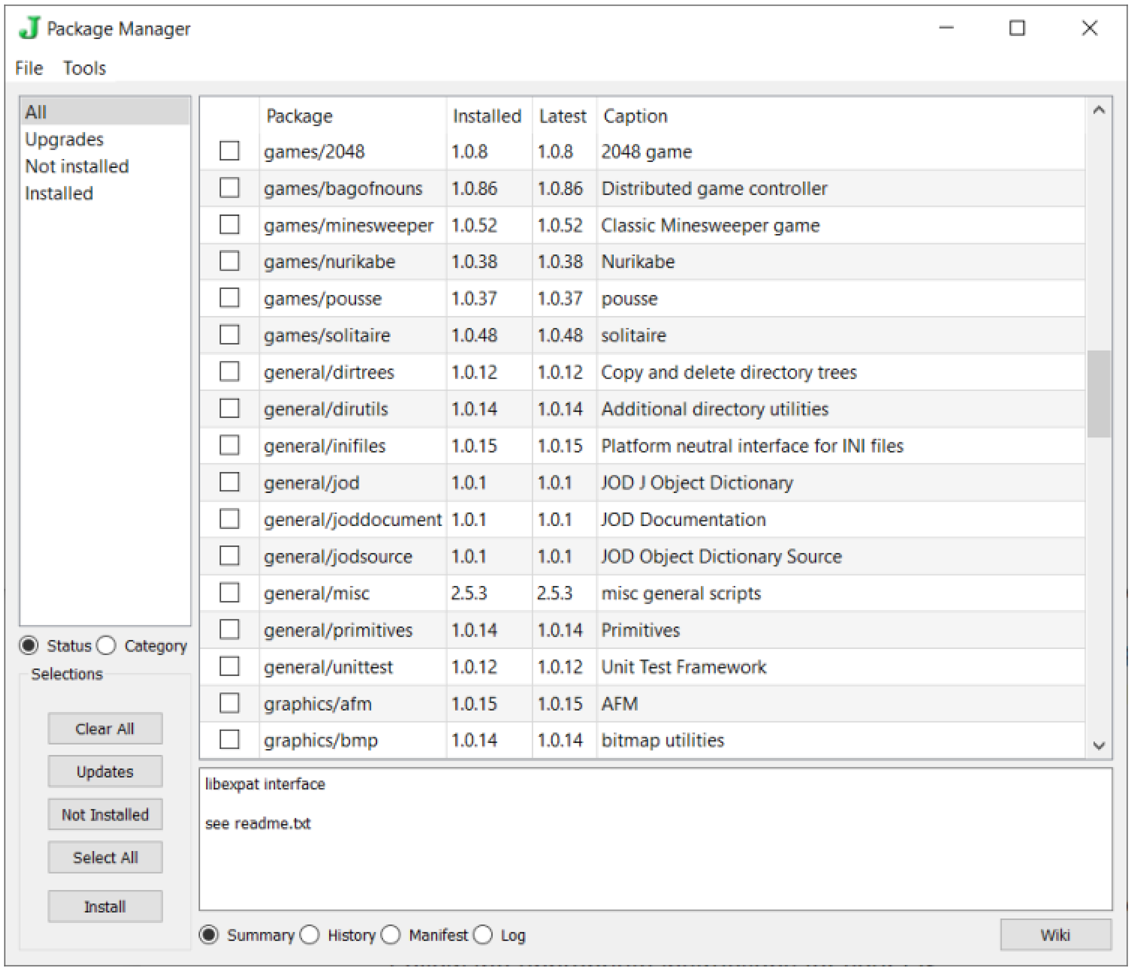
```
+---------+------+------+---------------------------------------+
|api/expat|1.0.11|1.0.11|libexpat                               |
+---------+------+------+---------------------------------------+
|api/gles |1.0.31|1.0.31|Modern OpenGL API                      |
+---------+------+------+---------------------------------------+
|api/java |1.0.2 |1.0.2 |api: Java to J shared library          |
+---------+------+------+---------------------------------------+
|api/jc   |1.0.1 |1.0.1 |api: J to C library (streams/fds/system)|
+---------+------+------+---------------------------------------+
|api/jni  |1.0.15|1.0.15|JNI                                    |
+---------+------+------+---------------------------------------+
```

[7]:
```
'showupgrade' jpkg ''  NB. list addons with updates
```

Installing addons with JQT GUI `pacman` I mostly use the Windows JQT version of pacman to install and maintain J addons. You can find pacman on the tools menu.



pacman shows all available addons and provides tools for installing, updating and removing them.



[● Status] ○ Category

	Package	Installed	Latest	Caption
☐	games/2048	1.0.8	1.0.8	2048 game
☐	games/bagofnouns	1.0.86	1.0.86	Distributed game controller
☐	games/minesweeper	1.0.52	1.0.52	Classic Minesweeper game
☐	games/nurikabe	1.0.38	1.0.38	Nurikabe
☐	games/pousse	1.0.37	1.0.37	pousse
☐	games/solitaire	1.0.48	1.0.48	solitaire
☐	general/dirtrees	1.0.12	1.0.12	Copy and delete directory trees
☐	general/dirutils	1.0.14	1.0.14	Additional directory utilities
☐	general/inifiles	1.0.15	1.0.15	Platform neutral interface for INI files
☐	general/jod	1.0.1	1.0.1	JOD J Object Dictionary
☐	general/joddocument	1.0.1	1.0.1	JOD Documentation
☐	general/jodsource	1.0.1	1.0.1	JOD Object Dictionary Source
☐	general/misc	2.5.3	2.5.3	misc general scripts
☐	general/primitives	1.0.14	1.0.14	Primitives
☐	general/unittest	1.0.12	1.0.12	Unit Test Framework
☐	graphics/afm	1.0.15	1.0.15	AFM
☐	graphics/bmp	1.0.14	1.0.14	bitmap utilities

libexpat interface

see readme.txt

[● Summary] ○ History ○ Manifest ○ Log

The GUI version is easy to use. Press the `Select All` button and then press the `Install` button to install all the addons. To update addons select the `Upgrades` menu and select the addons you want to update.

Step 3: Build the JOD development dictionaries from JODSOURCE

JOD source code is distributed in the form of JOD dictionary dumps. Dictionary dumps are large J scripts that serialize JOD dictionaries. Dumps contain everything stored in dictionaries. You will find source code, binary data, test scripts, documentation, build macros and more in typical JOD dictionaries.

`jodliterate` is stored as a JOD dictionary group. A dictionary group is simply a collection of J words with an optional *header* and *post-processor*. JOD generates J scripts from groups. Before we can *make* `jodliterate` we must load the JOD developement dictionaries. The JODSOURCE addon includes a J script that loads development dictionaries.

Again, start J and do:

[8]:
```
require 'general/jod'
```

[9]:
```
NB. set a JODroot user folder
NB. if not set /jod/ is the default

NB. use paths for your OS
UserFolders_j_=: UserFolders_j_ , 'JODroot';'c:/jodtest/joddicts'

sbx UserFolders_j_
```

```
+-----------+-------------------------------------------------------------- …
|Demos      |c:/j64/j901/addons/demos                                       …
+-----------+-------------------------------------------------------------- …
|Projects   |c:/users/john.baker/j901-user/projects                         …
+-----------+-------------------------------------------------------------- …
|User       |c:/users/john.baker/j901-user                                  …
+-----------+-------------------------------------------------------------- …
|JOD        |c:/users/john.baker/onedrive - jackson companies/jod           …
+-----------+-------------------------------------------------------------- …
|JODDUMPS   |c:/users/john.baker/onedrive - jackson companies/jod/joddumps  …
+-----------+-------------------------------------------------------------- …
|JODPRVDUMPS|c:/users/john.baker/onedrive - jackson companies/wd/jacksons/g …
+-----------+-------------------------------------------------------------- …
|JODSOURCE  |c:/jodtest/labtesting                                          …
+-----------+-------------------------------------------------------------- …
|JODTEST    |c:/jodtest/test                                                …
+-----------+-------------------------------------------------------------- …
|JACK       |c:/users/john.baker/onedrive - jackson companies/wd/jacksons/g …
+-----------+-------------------------------------------------------------- …
|BIDATA     |c:/bidata                                                      …
+-----------+-------------------------------------------------------------- …
```

```
|JODroot    |c:/jodtest/joddicts                                                 …
+-----------+------------------------------------------------------------------- …
```

[10]:
```
NB. list registered JOD dictionaries
NB. joddev, jod, utils must not be on list

od ''
```

```
+-+----+---+----+--------+-----------+---+------+----+---------+-----+
|1|docs|gps|imex|jacksons|jacksonsdev|jod|joddev|play|smugpyter|utils|
+-+----+---+----+--------+-----------+---+------+----+---------+-----+
```

[11]:
```
NB. uncomment the next line and run to load JOD developement dictionaries
NB. WARNING: do not do this if you have already build the development␣
 ↪dictionaries

NB. 0!:0<jpath'~addons/general/jodsource/jodsourcesetup.ijs'
```

[12]:
```
NB. list dictionaries with locations - joddev, jod, utils should exist
sbx 4 od ''
```

```
+-+------------------------------------------------------------------------ …
|1|+-----------+----------------------------------------------------------- …
| ||docs       |c:/users/john.baker/onedrive - jackson companies/jod/docs/  …
| |+-----------+----------------------------------------------------------- …
| ||gps        |c:/users/john.baker/onedrive - jackson companies/jod/gps/   …
| |+-----------+----------------------------------------------------------- …
| ||imex       |c:/users/john.baker/onedrive - jackson companies/jod/imex/  …
| |+-----------+----------------------------------------------------------- …
| ||jacksons   |c:/users/john.baker/onedrive - jackson companies/jod/jackso …
| |+-----------+----------------------------------------------------------- …
| ||jacksonsdev|c:/users/john.baker/onedrive - jackson companies/jod/jackso …
| |+-----------+----------------------------------------------------------- …
| ||jod        |c:/users/john.baker/onedrive - jackson companies/jod/jod/   …
| |+-----------+----------------------------------------------------------- …
| ||joddev     |c:/users/john.baker/onedrive - jackson companies/jod/joddev …
| |+-----------+----------------------------------------------------------- …
| ||play       |c:/users/john.baker/onedrive - jackson companies/jod/play/  …
| |+-----------+----------------------------------------------------------- …
| ||smugpyter  |c:/users/john.baker/onedrive - jackson companies/jod/smugpy …
| |+-----------+----------------------------------------------------------- …
| ||utils      |c:/users/john.baker/onedrive - jackson companies/jod/utils/ …
| |+-----------+----------------------------------------------------------- …
+-+------------------------------------------------------------------------ …
```

Step 4: Install a current version of `pandoc`

pandoc is easily one of the most useful markup utilities on the intertubes. If you routinely deal with markup formats like markdown, XML, LaTeX, json and you aren't using pandoc you are working too hard.

Be lazy! Install pandoc.

`jodliterate` uses the `task` addon to *shell out* to pandoc. Versions of pandoc after `2.9.1.1` support J syntax high-lighting.

[13]:
```
NB. show pandoc version from J - make sure you are running
NB. a recent version of pandoc. There may be different
NB. versions in many locations on various systems.

THISPANDOC=: '"C:\Users\john.baker\AppData\Local\Pandoc\pandoc"'
shell THISPANDOC,' --version'
```

```
pandoc 2.9.1.1
Compiled with pandoc-types 1.20, texmath 0.12, skylighting 0.8.3
Default user data directory: C:\Users\john.baker\AppData\Roaming\pandoc
Copyright (C) 2006-2019 John MacFarlane
Web:  https://pandoc.org
This is free software; see the source for copying conditions.
There is no warranty, not even for merchantability or fitness
for a particular purpose.
```

[14]:
```
NB. make sure your version of pandoc supports J syntax-highlighting

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

NB. J is on the supported languages list
(<;._2 tlf (shell THISPANDOC,' --list-highlight-languages') -. CR) e.~ <,'j'
```

```
1
```

Step 5: Install a current version of TeX and LaTeX

`jodliterate` uses LaTeX to compile PDF documents. When `jodliterate` runs it writes a LaTeX preamble file `JODLiteratePreamble.tex` to the output directory set by `setjodliterate`. It's a good idea to review this file to get an idea of the LaTeX packages used. It's possible that some of these packages are not in your LaTeX distribution and will have to be installed.

To ease the burden of LaTeX package maintenance I use freely available TeX versions that automatically install missing packages.

1. On Windows I use MiKTeX
2. On other platforms I use TeXLive

If your system automatically installs packages the first time you compile `jodliterate` it might fetch missing packages from The Comprehensive TEX Archive Network (CTAN). It may be necessary to reprocess your files a few times to insure all the required packages are downloaded and installed.

Step: 6 Make the `jodliterate` script

Once the JOD developement dictionaries are built (Step 3) making `jodliterate` is easy. Start J and do:

[15]:
```
require 'general/jod'

NB. open dictionaries
od ;:'imex joddev jod utils' [ 3 od ''
```

```
+-+-----------------------+----+------+---+-----+
|1|opened (rw/ro/ro/ro) ->|imex|joddev|jod|utils|
+-+-----------------------+----+------+---+-----+
```

[16]:
```
NB. generate jodliterate
sbx mls 'jodliterate'
```

```
+-+--------------------+--------------------------------------------------- …
|1|load script saved ->|c:/users/john.baker/onedrive - jackson companies/jo …
+-+--------------------+--------------------------------------------------- …
```

`mls` creates a standard J load script. Once generated this script can be loaded with the standard J `load` utility. You can test this by restarting J without JOD and loading `jodliterate`.

[17]:
```
NB. load generated script
load 'jodliterate'
```

```
NB. (jodliterate) interface word(s):
NB. --------------------------------
NB. THISPANDOC      NB. full pandoc executable path - use 'pandoc' only if on
shell path
NB. grplit          NB. make latex for group (y)
NB. ifacesection    NB. interface section summary string
NB. setjodliterate  NB. prepare for processing

NOTE: adjust pandoc path if version (pandoc 2.9.1.1) >= 2.9.1.1
```

Step 7: Run `jodliterate` on a JOD group with pandoc compatible markdown or LaTeX document fragments

This sounds a lot worse than it is. There is a group in `utils` called `sunmoon` that has an interesting *pandoc compatible document fragment.*

Start J and do:

[18]:
```
require 'general/jod'

od 'utils' [ 3 od ''
```

```
+-+--------------+-----+
|1|opened (ro) ->|utils|
+-+--------------+-----+
```

[19]:
```
NB. list words in the (sunmoon) group
80 list }. grp 'sunmoon'
```

```
IFACEWORDSsunmoon NORISESET         ROOTWORDSsunmoon  arctan
calmoons          cos               fromjulian        moons
round             sin               sunriseset0       sunriseset1
tabit             tan               today             yeardates
```

[20]:
```
NB. display short word explanations for (sunmoon)
sbx hlpnl }. grp 'sunmoon'
```

```
+-----------------+------------------------------------------------------- …
|IFACEWORDSsunmoon|interface words (IFACEWORDSsunmoon) group                 …
|NORISESET        |indicates sun never rises or sets in (sunriseset0) and ( …
|ROOTWORDSsunmoon |root words (ROOTWORDSsunmoon) group                       …
|arctan           |arc tangent                                              …
|calmoons         |calendar dates of new and full moons                     …
|cos              |cosine radians                                           …
|fromjulian       |converts Julian day numbers to dates, converse (tojulian …
|moons            |times of new and full moons for n calendar years         …
|round            |round (y) to nearest (x) (e.g. 1000 round 12345)         …
|sin              |sine radians                                             …
|sunriseset0      |computes sun rise and set times - see group documentatio …
|sunriseset1      |computes sun rise and set times - see group documentatio …
|tabit            |promotes only atoms and lists to tables                  …
|tan              |tan radians                                              …
|today            |returns todays date                                      …
|yeardates        |returns all valid dates for n calendar years             …
+-----------------+------------------------------------------------------- …
```

[21]:
```
NB. display part of the (sunmoon) group document header
NB. this is pandoc compatible markdown - note the inclusion
NB. of LaTeX commands - pandoc allows mixtures of markdown and LaTeX
3000 {. 2 9 disp 'sunmoon'
```

```
`sunmoon` is a collection of basic astronomical algorithms
The key verbs are `moons`, `sunriseset0` and `sunriseset1.`
All of these verbs were derived from BASIC programs published
in *Sky & Telescope* magazine in the 1990's. The rest of
the verbs in `sunmoon` are mostly date and trigonometric
utilities.

\subsection{\texttt{sunmoon} Interface}

~~~~ { .j }
  calmoons      NB. calendar dates of new and full moons
```

```
  moons         NB. times of new and full moons for n calendar years
  sunriseset0   NB. computes sun rise and set times - see group documentation
  sunriseset1   NB. computes sun rise and set times - see group documentation
~~~~

\subsection{\textbf\texttt{sunriseset0} \textsl{v--} sunrise and sunset times}

This  verb has been adapted from a BASIC program submitted by
Robin  G.  Stuart  *Sky & Telescope's*  shortest  sunrise/set
program  contest. Winning  entries were listed  in the  March
1995 Astronomical Computing column.

The  J version of this algorithm has been vectorized.  It can
compute any number of sunrise and sunset times in one call.

The `(y)` argument is a `5*n` floating point table where:

    0{ is latitude in degrees with northern latitudes positive.
    1{ is longitude in degrees with western longitudes negative.
    2{ is western time zones expressed as positive whole hours.
    3{ is the month number.
    4{ is the day number.

The result is a numeric table with four rows.  To handle the cases
when the sun never rises or sets the first two elements of the
corresponding result columns are:

    0{ is NORISESET an invalid hour indicating no rise or set
    1{ is 0 when the sun never rises
    1{ is 1 when the sun never sets

Warning: this algorithm breaks for latitudes close
to the South pole.

The original BASIC code has been slightly modified
to use control structures in place of GOTO's and line numbers.

Adapted from:

~~~~ { .c .numberLines startFrom="1"}
/* Sunrise/set by R. G. Stuart,  Mexico City, Mexico */
PI = 3.14159265#: DR = PI / 180: RD = 1 / DR
INPUT "Lat, Long (deg)"; B5, L5
INPUT "Time zone (hrs)"; H
B5 = DR * B5
INPUT "Month, day"; M, D
N = INT(275 * M / 9) - 2 * INT((M + 9) / 12) + D - 30
L0 = 4.8771 + .0172 * (N + .5 - L5 / 360)
```

```
C = .03342 * SIN(LO + 1.345)
C2 = RD * (ATN(TAN(LO + C)) - ATN(.9175 * TAN(LO + C)) - C)
SD = .3978 * SIN(LO + C): CD = SQR(1 - SD * SD)
SC = (SD * SIN(B5) + .0145) / (COS(B5) * CD)
IF ABS(SC) <= 1 THEN
  C3 = RD * ATN(SC / SQR(1 - SC * SC))
  R1 = 6 - H - (L5 + C2 + C3) / 15
  HR = INT(R1): MR = INT((R1 - HR) * 60)
  PRINT USING "Sunrise at ##:##"; HR; MR
  S1 = 18 - H - (L5 + C2 - C3) / 15
  HS = INT(S1): MS = INT((S1 - HS) * 60)
  PRINT USING "Sunset at ##:##"; HS; MS
ELSEIF SC > 1 THEN
  PRINT "Sun up all day"
ELSEIF SC < -1 THEN
  PRINT "Sun down all day"
END IF
END
~~~~

~~~~ { .j }
monad: ntRiseset =. sunriseset0 flBLHMD

  NB. rise and set times at Dog Lake today (daylight savings)
  td=. (44 + 19%60),(- 76 + 21%60), 4 , }. today 0
```

[22]:
```
NB. run jodliterate on (sunmoon)
require 'jodliterate'

NB. set the output directory - when running in Jupyter
NB. use a subdirectory of your notebook directory.
setjodliterate 'C:\Users\john.baker\bixml\grplit'
```

```
+-+---------------------------------+
|1|C:\Users\john.baker\bixml\grplit\|
+-+---------------------------------+
```

[23]:
```
NB. (grplit) returns a list of generated LaTeX and command
NB. files. The *.bat file compiles the generated LaTeX
,. grplit 'sunmoon'
```

```
+-----------------------------------------------+
|1                                              |
+-----------------------------------------------+
|C:\Users\john.baker\bixml\grplit\sunmoon.tex     |
+-----------------------------------------------+
|C:\Users\john.baker\bixml\grplit\sunmoontitle.tex|
+-----------------------------------------------+
```

```
|C:\Users\john.baker\bixml\grplit\sunmoonoview.tex|
+--------------------------------------------------+
|C:\Users\john.baker\bixml\grplit\sunmooncode.tex |
+--------------------------------------------------+
|C:\Users\john.baker\bixml\grplit\sunmoon.bat      |
+--------------------------------------------------+
```

Step 8: Compile the LaTeX files of the previous step to produce a PDF

```
[24]: _1000 {. shell 'C:\Users\john.baker\bixml\grplit\sunmoon.bat'
```

```
lmmono10-italic.otf><c:/users/j
ohn.baker/appdata/local/programs/miktex 2.9/fonts/opentype/public/lm/lmroman12-
italic.otf><c:/users/john.baker/appdata/local/programs/miktex 2.9/fonts/opentyp
e/public/lm/lmmonoslant10-regular.otf><c:/users/john.baker/appdata/local/progra
ms/miktex 2.9/fonts/opentype/public/lm/lmromanslant12-regular.otf><c:/users/joh
n.baker/appdata/local/programs/miktex 2.9/fonts/opentype/public/lm/lmmonolt10-b
old.otf><c:/users/john.baker/appdata/local/programs/miktex 2.9/fonts/opentype/p
ublic/lm/lmroman12-bold.otf><c:/users/john.baker/appdata/local/programs/miktex
2.9/fonts/opentype/public/lm/lmroman12-regular.otf><c:/users/john.baker/appdata
/local/programs/miktex 2.9/fonts/opentype/public/lm/lmroman17-regular.otf><c:/u
sers/john.baker/appdata/local/programs/miktex 2.9/fonts/opentype/public/lm/lmmo
no12-regular.otf>
Output written on sunmoon.pdf (22 pages, 110406 bytes).
Transcript written on sunmoon.log.

C:\Users\john.baker\bixml\grplit>endlocal
```

```
[25]: NB. display generated PDF
       shell 'C:\Users\john.baker\bixml\grplit\sunmoon.pdf'
```

```
[ ]:
```