Manual template latex

Continue

The guidelines on this page will help you to prepare and submit your LaTeX files. Please note that there are separate instructions available for CRC journal articles and IFAC meeting papers. Preparing your manuscript The Elsevier article class The Elsevier article class helps you to format the frontmatter of your manuscript properly. It is part of the elsarticle package. This package is contained in most TeX distributions and is available on CTAN. The elsarticle documentation and some common templates are part of files containing a template LaTeX manuscript, using the elsarticle class, plus associated BibTeX style files here. Although elsarticle.cls supports most journal styles, it is not possible to match the journal's layout exactly. For more complex articles two additional class files are documented here.Elsevier reference stylesSome journals require a specific reference style. The relevant bibliographic styles for LaTeX are packed with the sample manuscript are available in the "doc" folder.RecommendationsTo ensure a fast processing of your document, we recommend the following:Use elsarticle.cls in combination with BibTeX.Follow the documentation of the class you are using. (The documentation of elsarticle.cls is found here.)Use the standard layout or keep layout changes to a minimum. (Custom layout or keep layout changes to a minimum. (Custom layout or keep layout changes to a minimum.) specific formatting requirements. Most journals accept a PDF of your manuscript source files on your computer and attach it with item type 'Manuscript'. Bundle all manuscript source files in a single archive and attach it with item type 'LaTeX source files'. Source files include LaTeX files, BibTeX files, figures, tables, all LaTeX classes and packages that are not included in TeX Live and any other material that belongs to your manuscript. Please note the following technical requirements for the submission of source files: The source files should compile without errors with pdflatex or latex .The PDF should be submitted along with the source files. You can check in the CTAN database whether your class or package is contained in TeX Live. Figures should be submitted in one of the following formats: EPS, JPG, PDF or PNG. For quality standards, please see the artwork & media instructions. Frequently Asked QuestionsWhy is the item type. In this case, submit your manuscript in PDF format only and supply the source files when requested. For more information and support, see the Elsevier Support Center. For a beginner's guide to writing a manuscript in LaTeX see the interactive course on the Elsevier Publishing Campus. Preparing CRC journal articlesCamera-ready copy (CRC) journals are those that reproduce the author's manuscript exactly, with no intervention by the typesetter. Such journals are the exception rather than the rule; if a journal is CRC, this fact is clearly indicated in the instructions to authors. For LaTeX authors of camera-ready articles, we provide the ecrc.sty package. This is a small package designed to work with the elsarticle document class. All the features of elsarticle are available, along with a few extra commands specific to CRC reproduction. Documentation for the use of ecrc.sty is included in the manuscript template file available below. The archive file elsarticle ecrc.zip contains all the necessary files to run this package. To install ecrc.sty, unzip the elsarticle-ecrc.zip file. Usually the file can be unzipped directly in the local tree of your TeX distribution (for TeX Live, this would be in the texmf-local directory). The archive contains the following files:Once the package has been installed, edit the manuscript file ecrc-template.tex according to the instructions in that file, and save with a new name. The manuscript file should be compiled with pdflatex (and bibtex if desired).Please only use these packages after confirmation from the journal's editors. pandoc can convert between numerous markup and word processing formats, including, but not limited to, various flavors of Markdown, HTML, LaTeX and Word docx. For the full lists of input and output formats, see the --from and --to options below. Pandoc can also produce PDF output: see creating a PDF, below. Pandoc's enhanced version of Markdown includes syntax for tables, definition lists, metadata blocks, footnotes, citations, math, and much more. See below under Pandoc's Markdown. Pandoc has a modular design: it consists of a set of writers, which convert this native representation into a target format. Thus, adding an input or output format requires only adding a reader or writer. Users can also run custom pandoc filters to modify the intermediate AST. Because pandoc's intermediate representation of a document is less expressive than many of the formats it converts between, one should not expect perfect conversions between every format and every other. Pandoc attempts to preserve the structural elements, such as complex tables, may not fit into pandoc's simple document model. While conversions from pandoc's Markdown to all formats aspire to be perfect, conversions from formats more expressive than pandoc's Markdown can be expected to be lossy. If no input-files are specified, input is read from stdin. Output goes to stdout by default. For output to a file, use the -o option: pandoc -o output.html input.txt By default, pandoc produces a document fragment. To produce a standalone document (e.g. a valid HTML file including and ), use the -s or --standalone flag: pandoc -s -o output.html input.txt For more information on how standalone documents are produced, see Templates below. If multiple input files are given, pandoc will concatenate them all (with blank lines between them) before parsing. (Use --file-scope to parse files individually.) The format of the input and output can be specified explicitly using command-line options. The input format can be specified using the -t/--to option. Thus, to convert hello.txt from Markdown to LaTeX, you could type: pandoc -f markdown -t latex hello.txt To convert hello.txt from HTML to Markdown: pandoc -f html -t markdown hello.html Supported input formats and output formats are listed below under Options (see -f for input formats and pandoc --list-output-formats to print lists of supported formats). You can also use pandoc --list-output-formats and pandoc --list-output-formats and pandoc --list-output formats). guess it from the extensions of the filenames. Thus, for example, pandoc -o hello.txt will convert hello.txt from Markdown to LaTeX. If no output file's extension is unknown, the output format will default to HTML. If no input file is specified (so that input comes from stdin), or if the input files' extensions are unknown, the input format will be assumed to be Markdown. Pandoc uses the UTF-8 character encoding is not UTF-8, you should pipe input and output. If your local character encoding is not UTF-8, you should pipe input and output through iconv: iconv -t utf-8 input.txt | pandoc | iconv -f utf-8 Note that in some output formats (such as HTML, LaTeX, ConTeXt, RTF, OPML, DocBook, and Texinfo), information about the character encoding is included in the document header, which will only be included in the document header, which will only be included in the document header, which will only be included in the document header. requires that a LaTeX engine be installed (see --pdf-engine below). Alternatively, pandoc can use ConTeXt, roff ms, or HTML as an intermediate format. To do this, specify an output file with a .pdf extension, as before, but add the --pdf-engine option or -t context, -t html, or -t ms to the command line. The tool used to generate the PDF from the intermediate format may be specified using --pdf-engine. You can control the PDF style using variables, depending on the intermediate format used: see variables for KaTeX, variables for ConTeXt, variables for ms. When HTML is used as an intermediate format, the output can be styled using --css. To debug the PDF creation, it can be useful to look at the intermediate representation: instead of -o test.pdf, use for example -s -o test.tex. When using LaTeX, the following packages need to be available (they are included with all recent versions of TeX Live): amsfonts, amsmath, lm, unicodemath, iftex, listings (if the --listings option is used), fancyvrb, longtable, booktabs, graphicx (if the document contains images), hyperref, xcolor, ulem, geometry (with the geometry variable set), and babel (with lang). If CJKmainfont is set, xeCJK is needed. The use of xelatex or lualatex as the PDF engine requires fontspec. lualatex uses selnolig. xelatex uses bidi (with the dir variable set). If the mathspec variable is set, xelatex will use mathspec instead of unicode-math. The upquote and microtype packages are used if available, and biber, and biber, and biber, and biber. packages can optionally be used for citation rendering. The following packages will be used to improve output quality if present, but pandoc does not require them to be present: upquote (for better spacing adjustments), microtype (for better URLs), bookmark (for better PDF bookmarks), and footnote hyper or footnote (to allow footnotes in tables). Instead of an input file, an absolute URI may be given. In this case pandoc will fetch the content using HTTP: pandoc -f html -t markdown It is possible to supply a custom User-Agent string or other header when requesting a document from a URL: pandoc -f html -t markdown --request-header User-Agent:"Mozilla/5.0" \ -f FORMAT, --from=FORMAT, --from=FORMAT names. See --list-input-formats and --list-extensions, below. -t FORMAT, -w FORMAT, --to=FORMAT, --to=FORMAT, --write=FORMAT, --write=FORMAT, --to=FORMAT, --write=FORMAT, --write to the format name. See Extensions below, for a list of extensions and their names. See --list-output-formats and --list-extensions, below. -o FILE, --output +FILE Write output to FILE, --output +FILE
Write output to FILE, --output +FILE Write output to FILE, --output +FILE Write output +formats and --list-extensions, below. -o FILE, --output +FILE Write output +formats and --list-extensions, below. -o FILE, --output +FILE Write output +formats and --list-extensions, below. -o FILE, --output +FILE Write output +formats and --list-extensions, below. -o FILE, --output +FILE Write output +formats and --list-extensions, below. -o FILE, --output +FILE Write output +formats and --list-extensions, below. -o FILE, --output +FILE Write output +formats and --list-extensions, below. -o FILE, --output +FILE Write output +formats and --list-extensions, below. -o FILE, --output +FILE Write output +formats and --list-extensions, below. -o FILE, --output +FILE Write output +formats and --list-extensions, below. -o FILE, --output +formats and --list-extensions, below. -o directory to search for pandoc data files. If this option is not specified, the default user data directory will be used. On \*nix and macOS systems this will be the pandoc subdirectory of the XDG data directory (by default, \$HOME/.local/share, overridable by setting the XDG\_DATA\_HOME environment variable). If that directory does not exist and \$HOME/.pandoc exists, it will be used (for backwards compatibility). On Windows the default user data directory is C:\Users\USERNAME\AppData\Roaming\pandoc. You can find the default user data directory on your system by looking at the output of pandoc. You can find the default user data directory is C:\Users\USERNAME\AppData\Roaming\pandoc. You can find the default user data directory is C:\Users\USERNAME\AppData\Roaming\pandoc. You can find the default user data directory is C:\Users\USERNAME\AppData\Roaming\pandoc. You can find the default user data directory is C:\Users\USERNAME\AppData\Roaming\pandoc. You can find the default user data directory is C:\Users\USERNAME\AppData\Roaming\pandoc. You can find the default user data directory is C:\Users\USERNAME\AppData\Roaming\pandoc. You can find the default user data directory is C:\Users\USERNAME\AppData\Roaming\pandoc. You can find the default user data directory is C:\Users\USERNAME\AppData\Roaming\pandoc. You can find the default user data directory is C:\Users\USERNAME\AppData\Roaming\pandoc. You can find the default user data directory is C:\Users\USERNAME\AppData\Roaming\pandoc. You can find the default user data directory is C:\Users\USERNAME\AppData\Roaming\pandoc. You can find the default user data directory is C:\Users\USERNAME\AppData\Roaming\pandoc. You can find the default user data directory is C:\Users\USERNAME\AppData\Roaming\pandoc. You can find the default user data directory is C:\Users\USERNAME\AppData\Roaming\pandoc. You can find the default user data directory is C:\Users\USERNAME\AppData\Roaming\pandoc. You can find the default user data directory is C:\Users\USERNAME\AppData\Roaming\pandoc. You can find the default user data directory is C:\Users\Users\USERNAME\AppData\Roaming\pandoc. You can find the default user data directory is C:\Users\Us epub.css, templates) will override pandoc's normal defaults. -d FILE, --defaults=FILE Specify a set of default option settings. FILE is a YAML file whose fields correspond to command-line option settings. FILE is a YAML file whose fields correspond to command-line option settings. working directory, and then in the defaults subdirectory of the user data directory (see --data-dir). The .yaml extension may be overridden or extended by subsequent options on the command line. --bash-completion Generate a bash completion script. To enable bash completion with pandoc, add this to your .bashrc: eval "\$(pandoc --bash-completion)" --verbose Give verbose debugging output. --quiet Suppress warnings Exit with error status if there are any warnings. --fail-if-warnings Exit with error status if there are any warnings. --fail-if-warnings Exit with error status if there are any warnings. above DEBUG level will be written, regardless of verbosity settings (--verbose, --quiet). --list-input-formats List supported input formats, one per line. --list-extensions[=FORMAT] List supported extensions for FORMAT, one per line, preceded by a + or - indicating whether it is enabled by default in FORMAT. If FORMAT is not specified, defaults for pandoc's Markdown are given. --list-highlight-languages for syntax highlighting, one per line. See --highlight-style. -v, --version Print version. -h, --help Show usage message. --shiftheading-level-by=NUMBER Shift heading levels by a positive or negative integer. For example, with --shift-heading-level-by=-1, level 2 headings become level 1 headings, and level 3 headings become level 1 headings. Headings become level 1 headings become level 1 headings become level 2 headings. with a shift of -N, a level-N heading at the beginning of the document replaces the metadata title. --shift-heading-level-by=-1 is a good choice for converting HTML or Markdown documents that use an initial level-1 heading for the document title and level-2+ headings for sections. --shift-heading-level-by=-1 may be a good choice for converting HTML or Markdown documents that use an initial level-1 heading for the document title and level-2+ headings for sections. --shift-heading-level-by=-1 may be a good choice for converting HTML or Markdown documents that use an initial level-1 heading for the document title and level-2+ headings for sections. --shift-heading-level-by=-1 may be a good choice for converting HTML or Markdown documents that use an initial level-1 heading for the document title and level-2+ headings for sections. Markdown documents that use level-1 headings for sections to HTML, since pandoc uses a level-1 heading to render the document title. Deprecated. Use --shift-heading-level-by=X instead, where X = NUMBER - 1. Specify the base level for headings (defaults to 1). --strip-empty-paragraphs Deprecated. Use the +empty paragraphs extension instead. Ignore paragraphs with no content. This option is useful for converting word processing documents where users have used empty paragraphs to create inter-paragraphs to create inter-paragraph space. --indented code blocks-for example, perl,numberLines or haskell. Multiple classes may be separated by spaces or commas. --default-image-extension=EXTENSION Specify a default extension to use when image paths/URLs have no extension. This allows you to use the same source for formats that require different kinds of images. Currently this option only affects the Markdown and LaTeX readers. --file-scope Parse each file individually before combining for multifile documents. This will allow footnotes in different files with the same identifiers to work as expected. If this option is set, footnotes and links will not work across files. Reading binary files (docx, odt, epub) implies --file-scope. -F PROGRAM, --filter=PROGRAM Specify an executable to be used as a filter transforming the pandoc AST after the input is parsed and before the output is written. The executable should read JSON from stdin and write JSON to stdout. The JSON must be formatted like pandoc -- filter ./caps.py -t latex is equivalent to pandoc -- filter ./caps.py -t latex is e latex | pandoc -f json -t latex The latter form may be useful for debugging filters. Filters may be written in any language. Text.Pandoc.JSON exports toJSONFilter to facilitate writing filters in Haskell. Those who would prefer to write filters in python can use the module pandocfilters, installable from PyPI. There are also pandoc filter libraries in PHP, perl, and JavaScript/node.js. In order of preference, pandoc will look for filters in a specified full or relative path (executable) where \$DATADIR is the user data directory (see --data-dir, above), \$PATH (executable on non-executable) where \$DATADIR is the user data directory (see --data-dir, above), \$PATH (executable on non-executable) where \$DATADIR is the user data directory (see --data-dir, above),
\$PATH (executable on non-executable) where \$DATADIR is the user data directory (see --data-dir, above), \$PATH (executable on non-executable) where \$DATADIR is the user data directory (see --data-dir, above), \$PATH (executable on non-executable) where \$DATADIR is the user data directory (see --data-dir, above), \$PATH (executable on non-executable) where \$DATADIR is the user data directory (see --data-dir, above), \$PATH (executable on non-executable) where \$DATADIR is the user data directory (see --data-dir, above), \$PATH (executable on non-executable) where \$DATADIR is the user data directory (see --data-dir, above), \$PATH (executable on non-executable) where \$DATADIR is the user data directory (see --data-dir, above), \$PATH (executable on non-executable) where \$DATADIR is the user data directory (see --data-dir, above), \$PATH (executable on non-executable) where \$DATADIR is the user data directory (see --data-dir, above), \$PATH (executable on non-executable) where \$DATADIR is the user data directory (see --data-directory (see -order specified on the command line. -L SCRIPT, --lua-filter=SCRIPT Transform the document in a similar fashion as JSON filters (see --filter), but use pandoc's built-in Lua filtering system. The given Lua script is expected to return a list of Lua filters which will be applied in order. Each Lua filter must contain element-transforming functions indexed by the name of the AST element on which the filter function should be applied. The pandoc Lua module provides helper functions for element creation. It is always loaded into the script's Lua environment. See the Lua filters documentation for further details. In order of preference, pandoc will look for Lua filters in a specified full or relative path, \$DATADIR/filters where \$DATADIR is the user data directory (see --data-dir, above). Filters, and citeproc processing are applied in the order specified on the command line. Set the metadata field KEY to the value SPATADIR is the user data directory (see --data-dir, above). will be parsed as YAML boolean or string values. If no value is specified, the value will be treated as Boolean true. Like --variable, --metadata affects the metadata affects th values will be escaped when inserted into the template. Read metadata from the supplied YAML (or JSON) file. This option can be used with every input format, but string scalars in the YAML file will always be parsed as Markdown. (If the input format is Markdown or a Markdown or a Markdown variant, then the same variant will be used to parse the metadata file; if it is a non-Markdown format, pandoc's default Markdown extensions will be used.) This option can be used repeatedly to include multiple metadata files; values in files specified inside the document, or by using -M, overwrite values specified in earlier files. with this option. The file will be searched for first in the working directory, and then in the metadata subdirectory (see --data-dir). -p, --preserve tabs instead of converts tabs to spaces. (By default, pandoc converts tabs to spaces) the user data directory (see --data-dir). and code blocks. Tabs in regular text are always treated as spaces. --tab-stop=NUMBER Specify the number of spaces per tab (default is 4). --track-changes" feature. accept (the default) processes all the insertions and deletions. reject ignores them. Both accept and reject ignore comments, all includes all insertions, deletions, and comment-start, and comment-end classes, respectively. The author and time of change is included. all is useful for scripting: only accepting changes from a certain reviewer, say, or before a certain date. If a paragraph is inserted or deleted, track-changes=all produces a span with the class paragraph-insertion/paragraph-deletion before the affected paragraph break. This option only affects the docx reader. Extract images and other media contained in or linked from the source document to the path DIR, creating it if necessary, and adjust the images references in the document so they point to the extracted from the file system, or extracted from the file system, or extracted from the file system, or extracted from the stracted from the stracted from the file system. abbreviations=FILE Specifies a custom abbreviations from the user data directory or fall back on a system default. To see the system default. To see the system default. To see the system default. Markdown reader. Strings found in this list will be followed by a nonbreaking space, and the period will not produce sentence-ending spaces. --trace Print diagnostic output tracing parser progress to stderr. This option is intended for use by developers in diagnosing performance issues. -s, -standalone Produce output with an appropriate header and footer (e.g. a standalone HTML, LaTeX, TEI, or RTF file, not a fragment). This option is set automatically for pdf, epub3, fb2, docx, and odt output. For native output, this option is set automatically for pdf, epub3, fb2, docx, and odt output. file as a custom template for the generated document. Implies --standalone. See Templates, below, for a description of template syntax. If no extension is specified, an extension corresponding to the writer will be added, so that --template is not found, pandoc will search for it in the templates subdirectory of the user data directory (see --data-dir). If this option is not used, a default template appropriate for the output format will be used (see -D/--print-default-template). -V KEY[=VAL], --variable=KEY[:VAL] Set the template variable KEY to the value VAL when rendering the document in standalone mode. If no VAL is specified, the key will be given the value true. --sandbox, limiting IO operations in readers and writers to reading the files specified on the command line. Note that this option does not limit IO operations by filters or in the production of PDF documents. But it does offer security against, for example, disclosure of files through the use of yone using pandoc on untrusted user input should use this option. Note: some readers and writers (e.g., docx) need access to data files. If these are stored on the file system, then pandoc will not be able to find them when run in --sandbox mode and will raise an error. For these applications, we recommend compiled with the embed\_data\_files option, which causes the data files to be baked into the binary instead of being stored on the file system. -D FORMAT, --print-default-template=FORMAT Print the system default template for an output FORMAT. (See -t for a list of possible FORMATs.) Templates in the user data directory are ignored. This option may be used with -o/--output to redirect output to a file, but -o/--output must come before --print-default-templates use partials, for example styles.html. To print the partials, for example styles.html. --print-default-templates use partials, for example styles.html. To print the partials, use --print-default-templates use partials, for example styles.html. To print the partials, use --print-default-templates use Print a system default data file. Files in the user data directory are ignored. This option may be used with -o/--output to redirect output to a file, but -o/--output to redirect output to redirect output to redirect output to redirect output to a file. the OS on which pandoc is being run). The default is native. --dpi=NUMBER Specify the default dpi (dots per inch) value for conversion from pixels to inch/centimeters and vice versa. (Technically, the encoded value is used instead of the default specified by this option. --wrap=auto|none|preserve Determine how text is wrapped in the output (the source code, not the rendered version). With auto (the default), pandoc will attempt to wrap lines at all. With preserve, pandoc will attempt to wrap lines at all. attempt to preserve the wrapping from the source document (that is, where there are nonsemantic newlines in the source, there will be nonsemantic newlines in the source, there will be nonsemantic newlines in the source document (that is, where there are nonsemantic newlines in the source). the generated source code (see --wrap). It also affects calculation of column widths for plain text tables (see Tables below). --toc, --table-of-contents Include an automatically generated table of contents (or, in the case of latex, context, docx, odt, opendocument, rst, or ms, an instruction to create one) in the output document. This option has no effect unless -s/--standalone is used, and it has no effect on man, docbook4, docbook5, or jats output. Note that if you are producing a PDF via ms, the table of contents will appear at the beginning of the document, before the title. If you would prefer it to be at the end of the document, use the option --pdf-engine-opt=--no-toc-relocation. --tocdepth=NUMBER Specify the number of section levels to include in the table of contents. The default is 3 (which means that level-1, 2, and 3 headings will be listed in the contents). Strip out HTML comments in the Markdown or Textile source, rather than passing them on to Markdown, Textile or HTML output as raw HTML. This does not apply to HTML comments inside raw HTML blocks when the markdown in html blocks extension is not set. --no-highlight-style=STYLE|FILE Specifies the coloring style to be used in highlighted source code. Options are pygments (the default), kate, monochrome, breezeDark, espresso, zenburn, haddock, and tango. For more information on syntax highlighting in pandoc, see Syntax highlighting, below. See also --list-highlighting, below. See also --list-highlighting in pandoc, see Syntax highlighting in pandoc, highlight style. To generate the JSON version of a nexisting style, use --print-highlight-style = STYLE |FILE Prints a JSON version of a highlight style, use --print-highlight-style = STYLE |FILE Prints a JSON version of a highlight style = STYLE |FILE Prints a JSON version of a highlight style = STYLE |FILE Prints a
JSON version of a highlight style = STYLE |FILE Prints a JSON version of a highlight style = STYLE |FILE Prints a JSON version of a highlight style = STYLE |FILE Prints a JSON version of a highlight style = STYLE |FILE Prints a JSON version of a highlight style = STYLE |FILE Prints a JSON version of a highlight style = STYLE |FILE Prints a JSON version of a highlight style = STYLE |FILE Prints a JSON version of a highlight style = STYLE |FILE Prints a JSON version of a highlight style = STYLE |FILE Prints a JSON version of a highlight style = STYLE |FILE Prints a JSON version of a highlight style = STYLE |FILE Prints a JSON version of a highlight style = STYLE |FILE Prints a JSON version of a highlight style = STYLE |FILE Prints a JSON version of a highlight style = STYLE |FILE Prints a JSON version of a highlight style = STYLE |FILE Prints a JSON version of a highlight style = STYLE |FILE Prints a JSON version of a highlight style = STYLE |FILE Prints a JSON version of a highlight style = STYLE |FILE Prints a JSON version of a highlight style = STYLE |FILE Prints a JSON version of a highlight style = STYLE |FILE Prints a JSON version of a highlight style = STYLE |FILE Prints a JSON version of a highlight style = STYLE |FILE Prints a JSON version of a highlight style = STYLE |FILE Prints a JSON version of a highlight style = STYLE |FILE Prints a JSON version of a highlight style = STYLE |FILE Prints a JSON version of a highlight style = STYLE |FILE Prints a JSON version of a highlight style = STYLE |FILE Prints a JSON version of a highlight style = STYLE |FILE Prints a JSON version of a highlight style = STYLE |FILE Prints a JSON version of a highlight style = STYLE |FILE Prints a JSON version of a come before --print-highlight-style on the command line. --syntax-definition=FILE Instructs pandoc to load a KDE XML syntax definition file, which will be used to add support for new languages or to use altered syntax definitions for existing languages. This option may be repeated to add multiple syntax definitions. Include contents of FILE, verbatim, at the end of the header. This option can be used, for example, to include multiple files in the header. They will be included in the order specified. Implies --standalone. -B FILE, include-before-body=FILE|URL Include contents of FILE, verbatim, at the beginning of the document body (e.g. after the tag in HTML, or the \begin{documents. This option can be used to include multiple files. They will be included in the order specified. Implies --standalone. -A FILE, --include-after-body=FILE/URL Include contents of FILE, verbatim, at the end of the document} command in LaTeX). This option can be used repeatedly to include multiple files. They will be included in the order specified. Implies --standalone. resource-path=SEARCHPATH List of paths to search for images and other resources. The paths should be separated by : on Linux, UNIX, and macOS systems, and by ; on Windows. If --resource-path is not specified, the working directory must be explicitly listed or it will not be searched. For example: --resource-path =.:test will search the working directory, in that order. This option can be used repeatedly. Search path components that come later on the command line will be searched before those that come later on the command line will be searched before those that come later on the command line will be searched before those that come later on the command line will be searched before those that come later on the command line will be searched before those that come later on the command line will be searched before those that come later on the command line will be searched before those that come later on the command line will be searched before those that come later on the command line will be searched before those that come later on the command line will be searched before those that come later on the command line will be searched before those that come later on the command line will be searched before those that come later on the command line will be searched before those that come later on the command line will be searched before those that come later on the command line will be searched before those that come later on the command line will be searched before those that come later on the command line will be searched before those that come later on the command line will be searched before those that come later on the command line will be searched before those that come later on the command line will be searched before those that come later on the command line will be searched before those that come later on the command line will be searched before those that come later on the command line will be searched before those that come later on the command line will be searched be sea equivalent to --resource-path baz:bim:foo:bar. Set the request header NAME to the value VAL when making HTTP requests (for example, when a URL is given on the command line, or when resources used in a document must be downloaded). If you're behind a proxy, you also need to set the environment variable http://.... --no-checkcertificate Disable the certificate verification to allow access to unsecure HTTP resources (for example when the certificate is no longer valid or self signed). --self-contained Deprecated synonym for --embed-resources Produce a standalone HTML file with no external dependencies, using data: URIs to incorporate the contents of linked scripts, stylesheets, images, and videos. The resulting file should be "self-contained," in the sense that it needs no external files and no net access to be displayed properly by a browser. This option works only with HTML output formats, including html4, html5, html+lhs, html5+lhs, s5, slidy, slideous, dzslides, and revealjs. Scripts images, and stylesheets at absolute URLs will be downloaded; those at relative to the working directory (if the first source file is remote). Elements with the attribute data-external="1" will be left alone; the documents they link to will not be incorporated in the document. Limitation: resources that are loaded dynamically through JavaScript cannot be incorporated; as a result, some advanced features (e.g. zoom or speaker notes) may not work in an offline "self-contained" reveal.js slide show. --html-q-tags Use tags for quotes in HTML. (This option only has an effect if the smart extension is enabled for the input format used.) --ascii Use only ASCII characters in output. Currently supported for XML and HTML formats (which use entities), roff ms (which use entities), and to a limited degree LaTeX (which uses standard commands for accented characters when possible). roff man output uses ASCII by default. --reference-location option. --reference-location=block|section|document Specifyeaction|document Sp whether footnotes (and references, if reference-links is set) are placed at the end of the current (top-level) block, the current section, or the document. The default is document. ATX-style (#-prefixed) or Setext-style (underlined) headings for level 1 and 2 headings in Markdown output. (The default is atx.) ATX-style headings are always used for level-division=default|section|chapter|part Treat top-level-division=default|section|chapter|part Treat top-level-division=default|section|chapter|pa headings as the given division type in LaTeX, ConTeXt, DocBook, and TEI output. The hierarchy order is part, chapter, then section; all headings are shifted such that the top-level heading becomes the specified type. The default behavior is to determine the best division type via heuristics: unless other conditions apply, section is chosen. When the documentclass variable is set to report, book, or memoir (unless the article option is specified), chapter is implied as the setting for this option. If beamer is the output format, specifying either chapter or part {..}, while second-level headings remain as their default type. -N, --number-sections Number section headings in LaTeX, ConTeXt, HTML, Docx, ms, or EPUB output. By default, sections are not numbered. Sections with class unnumbered will never be numbered. Sections is specified. --number-offset=NUMBER[,NUMBER,...] Offset for section headings in HTML output (ignored in other output formats). The first number is added to the section number for top-level headings, the second for second-level headings, and so on. So, for example, if you want the first top-level heading in your document to be numbered "6", specify --number-offset=5. If your document to be numbered "6", specify --number-offset=5. If your document starts with a level-2 heading in your document to be numbered "6", specify --number-offset=5. If your document to be
numbered "6", specify --number-offset=5. If your document to be numbered "6", specify --number-offset=5. If your document to be numbered "6", specify --number-offset=5. If your document to be numbered "6", specify --number-offset=5. If your document to be numbered "6", specify --numbered "6", specify --numbered "6", specify --numbered "6", spec by default. Implies --number-sections. --listings Use the listings package for LaTeX code blocks. The package does not support multi-byte encoding issue is fully documented here: Encoding issue with the listings package. -i, --incremental Make list items in slide shows display incrementally (one by one). The default is for lists to be displayed all at once. --slide-level=NUMBER Specifies that headings with the specified level in the hierarchy are used to divide the slide show into sections; headings below this level create subheads within a slide. Valid values are 0-6. If a slide level is not specified, slides will not be split automatically on headings, and horizontal rules must be used to indicate slide level will be set automatically based on the contents of the document; see Structuring the slide show. --section-divs Wrap sections in tags (or tags for html4), and attach identifiers to the enclosing (or ) rather than the heading itself. See Heading identifiers, below. --email-obfuscating mailto: links in HTML documents. none leaves mailto: links as they are. javascript obfuscates them using JavaScript. references obfuscates them by printing their letters as decimal or hexadecimal character references. The default is none. --id-prefix=STRING Specify a prefix to be added to all identifiers and internal links in HTML and DocBook output, and to footnote numbers in Markdown and Haddock output. fragments to be included in other pages. -T STRING, --title-prefix=STRING Specify STRING as a prefix at the beginning of the title that appears at the beginning of the HTML body). Implies --standalone. -c URL, --css=URL Link to a CSS style sheet. This option can be used repeatedly to include multiple files. They will be included in the order specified. A stylesheet is required for generating EPUB. If none is provided using this option (or the css or stylesheet metadata fields), pandoc will look for a file epub.css in the user data directory (see --data-dir). If it is not found there, sensible defaults will be used. --reference-doc=FILE Use the specified file as a style reference in producing a docx or ODT file. Docx For best results, the reference docx should be a modified version of a docx file produced using pandoc. The contents of the reference docx are ignored, but its stylesheets and document properties (including margins, page size, header, and footer) are used in the new docx. If no reference.docx is specified on the command line, pandoc will look for a file reference.docx. First get a copy of the default reference.docx. Then open custom-reference.docx in Word, modify the styles as you wish, and save the file. For best results, do not make changes to this file other than modifying the styles: Normal Body Text First Paragraph Styles: Normal Body Text First Paragraph Styles: Normal Body Text First Paragraph Styles as you wish, and save the file. Heading 6 Heading 7 Heading 8 Heading 9 Block Text Source Code Footnote Text Definition Term D reference ODT should be a modified version of an ODT produced using pandoc. The contents of the reference ODT are ignored, but its stylesheets are used in the new ODT. If no reference ODT is specified on the command line, pandoc will look for a file reference.odt in the user data directory (see --data-dir). If this is not found either, sensible defaults will be used. To produce a custom reference.odt, first get a copy of the default reference.odt: pandoc -o custom-reference.odt in LibreOffice, modify the styles as you wish, and save the file. PowerPoint Templates included with Microsoft PowerPoint 2013 (either with .pptx or .potx extension) are known to work, as are most templates derived from these. The specific requirement is that the template should contain layouts with the following names (as seen within PowerPoint): Title Slide Title and Content Section Header Two Content Section Head will be used. If no layout is found with one of the names, pandoc will output a warning and use the layout under the Home menu to check.) You can also modify the default reference.pptx in MS PowerPoint (pandoc vill use the layouts with the names listed above). --epub-cover-image=FILE Use the specified image as the EPUB cover. It is recommended that the image be less than 1000px in width and height. Note that in a Markdown source document you can also specify cover-image in a YAML metadata for the EPUB. The file should contain a series of Dublin Core elements. For example: Creative Commons es-AR By default, pandoc will include the following metadata elements: (from the document title), (from the document authors), (from the lang variable, or, if is not set, the locale), and (a randomly generated UUID). Any of these may be overridden by elements in the metadata file. Note: if the source document is Markdown, a YAML metadata block in the document can be used instead. --epub-embed-font=FILE Embed the specified font in the EPUB. This option can be repeated to embed multiple fonts. Wildcards can also be used: for example, DejaVuSans-\*.ttf. However, if you use wildcards can also be used instead. on the command line, be sure to escape them or put the whole filename in single quotes, to prevent them from being interpreted by the shell. To use the embedded fonts, you will need to add declarations like the following to your CSS (see --css): @font-face { font-family: DejaVuSans; font-style: normal; font-weight: normal; src:url("DejaVuSans-Regular.ttf"); } @font-face { font-family: DejaVuSans; font-style: italic; font-weight: bold; src:url("DejaVuSans-BoldOblique.ttf"); } @font-face { font-family: DejaVuSans; font-style: italic; font-weight: bold; src:url("DejaVuSans-Bold.ttf"); } @font-face { font-family: DejaVuSans; font-style: italic; font-weight: bold; src:url("DejaVuSans-BoldOblique.ttf"); } @font-face { font-family: DejaVuSans; font-style: italic; font-weight: bold; src:url("DejaVuSans-BoldOblique.ttf"); } @font-face { font-family: DejaVuSans; font-style: italic; font-weight: bold; src:url("DejaVuSans; font-style: italic; font-weight: bold; src:url("DejaVuSans-BoldOblique.ttf"); } @font-face { font-family: DejaVuSans; font-style: italic; font-weight: bold; src:url("DejaVuSans-BoldOblique.ttf"); } @font-face { font-family: DejaVuSans; font-style: italic; font-weight: bold; src:url("DejaVuSans-BoldOblique.ttf"); } @font-face { font-family: DejaVuSans; font-style: italic; font-weight: bold; src:url("DejaVuSans-BoldOblique.ttf"); } @font-face { font-family: DejaVuSans; font-style: italic; font-weight: bold; src:url("DejaVuSans-BoldOblique.ttf"); } @font-face { font-family: DejaVuSans; font-style: italic; font-weight: bold; src:url("DejaVuSans-BoldOblique.ttf"); } @font-face { font-family: DejaVuSans; font-style: italic; font-weight: bold; src:url("DejaVuSans-BoldOblique.ttf"); } @font-face { font-family: DejaVuSans; font-style: italic; body { font-family: "DejaVuSans"; } --epub-chapter at level-1 headings. This option only affects the internal composition of the EPUB, not the way chapters and sections are displayed to users. Some readers may be slow if the chapter files are too large, so for large documents with few level-1 headings, one might want to use a chapter level of 2 or 3. --epub-subdirectory = DIRNAME Specific contents. The default is EPUB. To put the EPUB contents in the top level, use an empty string. --ipynboutput=all|none|best Determines how ipynb output cells are treated. all means that all of the data formats included in the original are preserved. none means that the contents of data cells are omitted. best causes pandoc to try to pick the richest data block in each output cell that is compatible with the output format. The default is best. --pdfengine=PROGRAM Use the specified engine when producing PDF output. Valid values are pdflatex, leatex, latexmk, tectonic, wkhtmltopdf, weasyprint, pagedjs-cli, prince, context, and pdfroff. If the engine is not in your PATH, the full path of the engine may be specified here. If this option is not specified, pandoc uses the following defaults depending on the output format specified using -t/--to: -t latex or none: pdflatex (other options: xelatex, lualatex, tectonic, latexmk) -t context: context -t html: whtmltopdf (other options: prince, weasyprint, pagedjs-cli; see print-css.rocks for a good introduction to PDF generation from HTML/CSS) -t ms: pdfroff --pdf-engine-opt=STRING Use the given string as a command-line argument to the pdf-engine. For example, to use a persistent directory foo for latexmk's auxiliary files, use --pdf-engine-opt=-outdir=foo. Note that no check for duplicate options is done. -C, --citeproc Process the citations in the file, replacing them with rendered citations and adding a bibliography. Citation processing will not take place unless bibliographic data is supplied, either through an external file specified using the --bibliography option or the bibliography option or the bibliography field in metadata, or via a references section in metadata, or via a references section in metadata containing a list of citations in CSL YAML format with Markdown formatting. The style is controlled by a CSL stylesheet specified using the --csl option or the csl field in metadata. (If no stylesheet is specified, the chicago-author-date style will be used by default.) The citation processing transformations are applied in the order they appear on the command line. For more information, see the section on Citations. --bibliography=FILE Set the bibliography field in the document's metadata to FILE, overriding any value set in the metadata. If you supply this argument multiple times, each FILE will be
fetched via HTTP. If FILE is a URL, it will be fetched via HTTP. If FILE is a URL, it will be fetched via HTTP. If FILE is a URL, it will be added to bibliography. If FILE is a URL, it will be fetched via HTTP. If FILE is a URL, it will be fetched via HTTP. If FILE is a URL, it will be fetched via HTTP. If FILE is a URL, it will be fetched via HTTP. If FILE is a URL, it will be fetched via HTTP. If FILE is a URL, it will be fetched via HTTP. If FILE is a URL, it will be fetched via HTTP. If FILE is a URL is a URL is a URL in the metadata. path (see --resource-path). --csl=FILE Set the csl field in the document's metadata to FILE, overriding any value set in the metadata. (This is equivalent to --metadata csl=FILE.) If FILE is a URL, it will be fetched via HTTP. If FILE is not found relative to the working directory, it will be sought in the resource-path) and finally in the csl subdirectory of the pandoc user data directory. --citation-abbreviations=FILE Set the citation-abbreviations field in the document's metadata citation-abbreviations=FILE.) If FILE is a URL, it will be fetched via HTTP. If FILE is not found relative to the working directory, it will be sought in the resource-path) and finally in the c-steproc option or with PDF output. It is intended for use in producing a LaTeX file that can be processed with bibtex. --biblatex Use biblatex for citations in LaTeX output. This option is not for use with the --citeproc option or with PDF output. It is intended for use in producing a LaTeX file that can be processed with bibtex or biber. The default is to render TeX math as far as possible using Unicode characters. Formulas are put inside a span with class="math", so that they may be styled differently from the surrounding text if needed. However, this gives acceptable results only for basic math, usually you will want to use --mathjax (--mathjax to display math) or \[...\] (for display math) or \[...\] and wrapped in tags with class math. Then the MathJax JavaScript will render it. The URL should point to the MathJax js load script. If a URL is not provided, a link to the Cloudflare CDN will be inserted. --mathml Convert TeX math to MathJax js load script. If a URL is not provided, a link to the Cloudflare CDN will be inserted. --mathml Convert TeX math to MathJax js load script. If a URL is not provided, a link to the Cloudflare CDN will be inserted. only Firefox and Safari (and select e-book readers) natively support MathML. --webtex [=URL] Convert TeX formulas to tags that link to an external script that converts formulas to tags that link to an external script that converts formulas to tags that link to an external script that convert TeX formulas to tags that link to an external script that convert TeX formulas to tags that link to an external script that convert TeX formulas to tags that link to an external script that convert TeX formulas to tags that link to an external script that converts formulas to tags that link to an extern CodeCogs URL generating PNGs will be used (). Note: the --webtex option will affect Markdown output as well as HTML, which is useful if you're targeting a version of Markdown without native math support. --katex[=URL] Use KaTeX to display embedded TeX math in HTML output. The URL is the base URL for the KaTeX library. That directory should contain a katex.min.js and a katex.min.css file. If a URL is not provided, a link to the KaTeX CDN will be inserted. --gladtex Enclose TeX math in tags in HTML output. The resulting HTML can then be processed by GladTeX to produce SVG images of the typeset formulas and an HTML file with these images embedded. pandoc -s --gladtex input.md -o myfile.htex gladtex -d image dir myfile.htex # produces myfile.html and images in image dir --dump-args Print information about command-line arguments to stdout, then exit. This option is intended primarily for use in wrapper scripts. The first line of output contains the name of the output file specified with the -o option, or - (for stdout) if no output file was specified. The remaining lines contain the command-line arguments, one per line, in the order they appear. These do not include regular pandoc options and their arguments, but do include any options appearing after a -- separator at the end of the line. -- ignore-args Ignore command-line arguments, but do include regular pandoc options appearing after a -- separator at the end of the line. Regular pandoc options are not ignored. Thus, for example, pandoc --ignore-args -o foo.html -s foo.txt -- e latin1 is equivalent to pandoc completes successfully, it will return exit code 0. Nonzero exit codes have the following meanings: 1 PandocIOError 3 PandocCompletes successfully, it will return exit code 0. Nonzero exit codes have the following meanings: 1 PandocIOError 3 PandocCompletes successfully, it will return exit code 0. Nonzero exit codes have the following meanings: 1 PandocIOError 3 PandocCompletes successfully, it will return exit code 0. Nonzero exit codes have the following meanings: 1 PandocIOError 3 PandocCompletes successfully, it will return exit code 0. Nonzero exit codes have the following meanings: 1 PandocIOError 3 PandocCompletes successfully, it will return exit code 0. Nonzero exit code 0. Nonzero exit codes have the following meanings: 1 PandocIOError 3 PandocFomPlateError 5 PandocCompletes successfully, it will return exit code 0. Nonzero exit code PandocOptionError 21 PandocUnknownReaderError 22 PandocUnknownWriterError 23 PandocUnsupportedExtensionError 43 PandocCiteprocError 44 PandocCiteprocError 47 PandocCiteprocError 25 PandocUnknownWriterError 31 PandocHttpError 62 PandocShouldNeverHappenError 63 PandocSomeError 65 PandocParseError 65 PandocCuaError 91 PandocCuaError 92 PandocCuaError 93 PandocSyntaxMapError 93 PandocCuaError 94 PandocCuaError 97 PandocCuaError 97 PandocSyntaxMapError 97 PandocSyntaxMapError 97 PandocCuaError 97 PandocCuaEr PandocCouldNotFindDataFileError 98 PandocCouldNotFindMetadataFileError 99 PandocResourceNotFound The --defaults option may be used to specify a package of options, in the form of a YAML file. Fields that expect a file path (or list of file paths), the following syntax may be used to interpolate environment variables: csl: \${HOME}/mycsldir/special.csl \${USERDATA} may also be used; this will always resolve to the user data directory that is current when the defaults file is parsed, regardless of the setting of the environment variable USERDATA. \${.} will resolve to the directory containing the defaults file itself. This allows you to refer to resources containing this defaults file This environment in that directory from which pandoc is run - \${.}/meta.xml resource-path: - . # the working directory from which pandoc is run - \${.}/meta.xml resource-path: - . # the working directory from which pandoc is run - \${.}/meta.xml resource-path: - . # the working directory from which pandoc is run - \${.}/meta.xml resource-path: - . # the working directory from which pandoc is run - \${.}/meta.xml resource-path: - . # the working directory from which pandoc is run - \${.}/meta.xml resource-path: - . # the working directory from which pandoc is run - \${.}/meta.xml resource-path: - . # the working directory from which pandoc is run - \${.}/meta.xml resource-path: - . # the working directory from which pandoc is run - \${.}/meta.xml resource-path: - . # the working directory from which pandoc is run - \${.}/meta.xml resource-path: - . # the working directory from which pandoc is run - \${.}/meta.xml resource-path: - . # the working directory from which pandoc is run - \${.}/meta.xml resource-path: - . # the working directory from which pandoc is run - \${.}/meta.xml resource-path: - . # the working directory from which pandoc is run - \${.}/meta.xml resource-path: - . # the working directory from which pandoc is run - \${.}/meta.xml resource-path: - . # the working directory from which pandoc is run - \${.}/meta.xml resource-path: - . # the working directory from which pandoc is run - \${.}/meta.xml resource-path: - . # the working directory from which pandoc is run - \${.}/meta.xml resource-path: - . # the working directory from which pandoc is run - \${.}/meta.xml resource-path: - . # the working directory from which pandoc is run - \${.}/meta.xml resource-path: - . # the working directory from which pandoc is run - \${.}/meta.xml resource-path directory from which pandoc is run - \${.}/meta.xml resource-path directory from which pandoc is run - \${.}/meta.xml
resource-path directory from which pandoc is variable interpolation syntax only works in fields that expect file paths. Defaults for writing letters, save it as letter.yaml in the defaults subdirectory of the user data directory, and then invoke these defaults from any directory using pandoc --defaults letter or pandoc --defaults are used, their contents will be combined. Note that, where command-line arguments may be repeated (--metadata-file, --css, --include-in-header, --include-before-body, --variable, --metadata, --syntax-definition), the values specified on the command line will combine with values specified in the defaults file, rather than replacing them. The following tables show the mapping between the command line and defaults file, rather than replacing them. The following tables show the mapping between the command line will combine with values specified in the defaults file, rather than replacing them. The following tables show the mapping between the command line will combine with values specified in the defaults file, rather than replacing them. The following tables show the mapping between the command line will combine with values specified in the defaults file, rather than replacing them. The following tables show the mapping between the command line will combine with values of input-files. uence [] for no input. --from markdown+emoji --to markdown+hard line breaks to: markdown+hard line breaks writer: markdown+hard line breaks --output --data-dir dir --defaults file --verbose --quiet --fail-if-warnings --sandbox --log=FILE Options specified in a defaults file itself always have priority over those in another file included with a defaults: entry. verbosity can have the values ERROR, WARNING, or INFO. --shift-heading-level-by -1 shift-heading-level-by: -1 --indented-code-classes python indented-code-classes: - python --default-image-extension: '.jpg' default-image-extension: '.jpg' default-imagefilters: - pandoc-citeproc - count-words.lua - type: json path: special.lua --metadata key2 metadata: key: value key2: true --metadata-files: - metadata-files: - metadata-files: - metadata-files: - metadata key2 metadata key2 metadata key2 metadata values specified in a defaults file are parsed as literal string text, not Markdown. Filters will be assumed to be Lua filters if they have the .lua extension, and JSON filters otherwise. But the filter type can also be specified explicitly, as shown. Filters are run in the order specified. To include the built-in citeproc filter, use either citeproc or {type: citeproc}. standalone --template letter --variable key=val \ --variable key2 variables: key: val key2: true --eol nl --dpi 300 --wrap 60 --columns 72 --table-of-contents --toc-depth 3 --strip-comments --no-highlight-style kate --syntax-definition: mylang.xml syntax-definition: mylang.xm in-header: - inc.tex --include-before-body inc.tex include-after-body: - inc.tex --include-after-body: - inc.tex --resource-path: ['.', 'foo'] --request-header foo:bar request-header -reference-location block reference-location: block --markdown-headings atx --top-level-division: chapter --number-sections --number-sections: references --id-prefix ch1 --title-prefix MySite --css styles/screen.css \ --css styles/special.css css: - styles/screen.css - styles/screen.css - styles/screen.css - styles/special.otf - epub-embed-font headline.otf epub-cover-image cover.jpg epub-cover-image: cover.jpg epub-co xelatex --pdf-engine-opt=--shell-escape pdf-engine-opts: - '-shell-escape' pdf-engine-opts: - '-shell-escape' --citeproc --bibliography: logic.bib --csl ieee.csl --citeproc --bibliography: logic.bib --csl ieee.csl --citeproc --bibliography: logic.bib --csl ieee.csl --citeproc --bibliography: logic.bib metadata: bibliography: logic.bib --csl ieee.csl --citeproc --bibliography: logic.bib metadata: bibliography: logic.bib --csl ieee.csl --citeproc --bibliography: logic.bib --csl ieee.csl --citeproc --cite output. If you want to use citeproc to format citations, you should also set 'citeproc: true'. If you need control over when the citeproc in the list of filters, you should instead use citeproc in the list of filters. math-method: method: m is used, pandoc uses a template to add header and footer material that is needed for a self-standing document. To see the default template can be specified using the --template option. You can also override the system default templates for a given output format FORMAT by putting a file templates/default.\*FORMAT\* in the user data-dir, above). Exceptions: For odt output, customize the default.latex template (or the default.context template, if you use -t context, or the default.ms template, if you use -t context, or the default.atex template. For pdf output, customize the default.atex template (or the default.context template, if you use -t context, or the default.ms template, if you use -t context, or the default.atex template (or the default.context template). t ms, or the default.html template, if you use -t html). docx and pptx have no template (however, you can use --reference-doc to customize the output). Templates contain variables, which allow for the inclusion of arbitrary information at any point in the file. They may be set at the command line using the -V/-variable option. If a variable is not set, pandoc will look for the key in the document's metadata, which can be set using either YAML metadata blocks or with the -M/--metadata option. In addition, some variables are given default values by pandoc. See Variables below for a list of variables used in pandoc's default templates. If you use custom templates, you may need to revise them as pandoc changes. We recommend tracking the changes in the default templates, and modifying your custom templates accordingly. An easy way to do this is to fork the pandoc-templates repository and merge in changes after each pandoc release. To mark variables and control structures in the template, either \$...\$ or \${...} may be used as delimiters The styles may also be mixed in the same template, but the opening and closing delimiter may be followed by one or more spaces or tabs, which will be ignored. The closing delimiter may be followed by one or more spaces or tabs, which will be ignored. slot for an interpolated variable names must begin with a letter and can contain letters, numbers, , -, and .. The keywords it, if, else, endif, for, sep, and endfor may not be used as variable names. Examples: \$foo\$ \$foo.bar.baz\$ \$foo bar.baz\$ \$foo \$ \$foo.bar.baz} \${foo bar.baz-bim} \${ foo } Variable names with periods are used to get at structured variable values. So, for example, employee field. If the value of the salary field of the salary field of the salary field of the salary will return the value of the salary field of the salary field of the salary will return the value of the salary field of the salary field of the salary field of the salary field of the salary will return the value of the salary field of the sa that the calling program will escape the strings appropriately for the output format.) If the value is a list, the value will be rendered as the empty string. A conditional begins with if(variable) (enclosed in matched delimiters) and ends with endif (enclosed in matched delimiters). It may optionally contain an else (enclosed in matched delimiters). The if section is used if variable has a non-empty value, otherwise the else section is used (if present). Examples: \$if(foo)\$ \$foo\$ \$endif\$ \$if(foo)\$ bar\$endif\$ bar\$endif\$ \$if(foo)\$ bar\$endif\$ \$if(foo)\$ bar\$endif\$ \$if(foo)\$ bar\$endif\$ \$if(foo)\$ bar\$endif\$ \$if(foo)\$ bar\$endif\$ bar\$endif\$ \$if(foo)\$ bar\$endif\$ \$if(foo)\$ bar\$endif\$ bar\$endif\$ \$if(foo)\$ bar\$endif foo.bar } \${else} no foo! \${endif} The keyword elseif may be used to simplify complex nested conditionals: \$if(foo)\$ XXX \$elseif(bar)\$ YYY \$else\$ ZZZ \$endif\$ A for loop begins with endfor (enclosed in matched delimiters). If variable is an array, the material inside the loop will be evaluated repeatedly. with variable being set to each value of the array in turn, and concatenated. If variable is a map, the material inside will be performed on its value. Examples: \$for(foo)\$ + \$foo.last\$, \$foo.first\$ \$endfor\$ \${ for(foo.bar) } - \${ foo.bar.last }, \${ foo.bar.first } \${ endfor } separator between sep and the endfor is the separator. \${ for(foo) }\${ foo }\${ endfor } Instead of using variable } the separator. \${ for(foo) }\${ foo }\${ endfor } Instead of using variable } the separator between sep and the endfor is the separator. \${ for(foo) }\${ foo }\${ endfor } Instead of using variable } the separator between sep and the endfor is the separator. \${ for(foo) }\${ foo }\${ foo }\${ endfor } Instead of using variable } the separator between sep and the endfor is the separator between sep and the endfor is the separator between sep and the endfor } the separator between sep and the endfor is the separator between sep and the endfor \${ foo }\${ inside the loop, the special anaphoric keyword it may be used. \${ for(foo.bar) } - \${ it.last }, \${ it.first } \${ endfor } Partials (subtemplates stored in different files) may be included by using the main template. The file name will be assumed to have the same extension as the main template if it lacks an extension. When calling the partial, the full name including file extension can also be used: \${ styles.html() } (If a partial is not found in the directory of the user data directory.) Partials may optionally be applied to variables using a colon: \${ articles: bibentry() } \${ articles is an array, this will iterate over its values, applying the partial bibentry() } \${ articles: bibentry() } \${ articles is an array, this will iterate over its values, applying the partial bibentry() } \${ articles: bibentry() } \${ article used when iterating over partials. In the above examples, the bibentry partial
should contain it title (and so on) instead of articles. title. Final newlines are omitted from included partials. A separator between values of an array may be specified in square brackets, immediately after the variable name or partial. \${months[, ]}\$ {articles:bibentry()[; ]} The separator in this case is literal and (unlike with sep in an explicit for loop) cannot contain interpolated variables or other template directives. To ensure that content is "nested," that is, subsequent lines indented, use the ^ directive: \$item.number\$ \$^\$\$item.description\$ (\$item.price\$) In this example, if item.description has multiple lines, they will all be indented to line up with the first line: 00123 A fine bottle of 18-year old Oban whiskey. (\$148) To nest multiple lines to the same level, align them with the ^ directive in the template. For example: \$item.number\$ \$^\$\$item.description\$ (\$item.price\$) (Available til \$item.sellby\$.) will produce 00123 A fine bottle of 18-year old Oban whiskey. (\$148) (Available til March 30, 2020.) If a variable occurs by itself on a line, preceded by whitespace and not followed by further text or directives on the same line, and the variable occurs by itself on a line, preceded by whitespace and not followed by further text or directives on the same line, and the variable occurs by itself on a line, preceded by whitespace and not followed by further text or directives on the same line, and the variable occurs by itself on a line, preceded by whitespace and not followed by further text or directives on the same line, and the variable occurs by itself on a line, preceded by whitespace and not followed by further text or directives on the same line, and the variable occurs by itself on a line, preceded by whitespace and not followed by further text or directives on the same line, and the variable occurs by itself on a line, preceded by whitespace and not followed by further text or directives on the same line, and the variable occurs by itself on a line, preceded by whitespace and not followed by further text or directives on the same line, and the variable occurs by itself on a line, preceded by whitespace and not followed by further text or directives on the same line, and the variable occurs by itself on a line, preceded by whitespace and not followed by further text or directives on the same line, and the variable occurs by itself on a line, preceded by whitespace and not followed by further text or directives on the same line, and the variable occurs by itself on a line, preceded by whitespace and not followed by further text or directives on the same line, and the variable occurs by itself on a line, preceded by whitespace and not followed by further text or directives on the variable occurs by itself on a line, preceded by the variable occurs by itself on a line, preceded by the variable occurs by itself on a line, preceded by the variable occurs by itself on a line, preceded by the variable occurs by itself on a line, pr interpolated variables) are not breakable, but they can be made breakable in part of the template by using the ~ keyword (ended with a short line length.\$~\$ A pipe transforms the value of a variable or partial. Pipes are specified using a slash (/) between the variable name (or partial) and the pipe name. Example: \$for(name)\$ \$name/uppercase\$ Pipes may be chained: \$for(employees/pairs)\$ \$it.key/alpha/uppercase\$. \$it.name\$ \$endfor\$ \$come pipes take parameters: |------l \$for(employee)\$ \$it.name.first/uppercase/left 20 "| "\$\$it.name.salary/right 10 " | " " |"\$ \$endfor\$ |--------uppercase. lowercase: Converts text to lowercase. length: Returns the length of the value; number of characters for a textual value or array, and has no effect on other values. first: Returns the first value of an array, if applied to a non-empty array; otherwise returns the original value. last: Returns the last value of an array, if applied to a non-empty array; otherwise returns the original value. rest: Returns all but the first value of an array, if applied to a non-empty array; otherwise returns the original value. value. chomp: Removes trailing newlines (and breakable space). nowrap: Disables line wrapping on breakable spaces. alpha: Converts textual values that can be used to get lettered enumeration from array indices. To get uppercase letters, chain with uppercase. roman: Converts textual values that can be read as an integer into lowercase roman numerals. This can be used to get lettered enumeration from array indices. To get uppercase roman, chain with uppercase roman numerals. This can be used to get lettered enumeration from array indices. To get uppercase roman, chain with uppercase roman, chain with uppercase roman numerals. on other values. This can be used to align material in tables. Widths are positive integers indicating the number of characters must be backslash-escaped. right n "leftborder": Renders a textual value in a block of width n, aligned to the right, and has no effect on other values, center n "leftborder": "rightborder": Renders a textual value in a block of width n. aligned to the center, and has no effect on other values, lang identifies the main language of the document using IETF language tags. This affects most formats, and controls hyphenation in PDF output when using LaTeX (through babel and polyglossia) or ConTeXt. Use native pandoc Divs and Spans with the language: --- lang: en-GB ... Text in the main document language (British English). ::: {lang=fr-CA} > Cette citation est écrite en français canadien. ::: More text in English. ['Zitat auf Deutsch.'] {lang=de} dir the base script direction, either rtl (right-to-left) or ltr (left-to-right). For bidirectional documents, native pandoc spans and divs with the dir attribute (value rtl or ltr) can be used to override the base direction in some output formats. This may not always be necessary if the final renderer (e.g. the browser, when generating HTML) supports the Unicode Bidirectional Algorithm. When using LaTeX for bidirectional documents, only the xelatex engine is fully supported (use --pdf-engine=xelatex), document-css Enables inclusion of most of the CSS in the styles.html partial (have a look with pandoc --print-default-datafile=templates/styles.html). Unless you use --css, this variable is set to true by default. You can disable it with e.g. pandoc -M document-css=false. mainfont sets the CSS font-family property on the html element. fontsize sets the base CSS font-family property on the html element. sets the CSS color property on the html element. linkcolor sets the CSS color property on all links. monofont sets the CSS font-family property on code elements. monobackground-color property on code elements and adds extra padding. linestretch sets the CSS line-height property on the html element, which is preferred to be unitless. background-color sets the CSS background-color property on the html element. To override or extend some CSS for just one document, include for example: --- header-includes: | blockguote { font-style: italic; } tr.even { background-color: #f0f0f0; } td, th { padding: 0.5em 2em 0.5em; } tbody { border-bottom: none; } --- classoption=fleqn. These affect HTML output when producing slide shows with pandoc. institute author affiliations: can be a list when there are multiple authors revealjs-url base URL for revealjs-url base URL for Slideous) title-slide-attributes additional attributes for the title slide of reveal.js slide shows. See background in reveal.js, beamer, and pptx for an example. All reveal.js, use 0. These variables change the appearance of PDF slides using beamer. aspectratio slide aspect ratio (43 for 4:3 [default], 169 for 16:9, 1610 for 16:9, 1610 for 16:10, 149 for 14:9, 141 for 1.41:1, 54 for 5:4, 32 for 3:2) beameroption add extra beamer option with \setbeameroption add extra beamer option symbols; other valid values are frame, vertical, and horizontal) section-titles enables "title pages" for new sections (default is true) theme, colortheme, fonttheme, innertheme, outertheme, innertheme, outertheme beamer themes (a list). controlled via templates. monofont font to use for code. Pandoc uses these variables when creating a PDF with a LaTeX engine. block-headings, or fifth- and sixth-level with book classes) free-standing rather than run-in; requires further formatting to distinguish from \subsubsection

(third- or fourth-level headings). Instead of using this option, KOMA-Script can adjust headings more extensively: --- documentclass: scrartcl header-includes: |\RedeclareSectionCommand[ beforeskip=-10pt plus -2pt minus 1sp, font=ormalfont\itshape]{paragraph} \RedeclareSectionCommand[ beforeskip=-10pt plus -2pt minus -1pt, afterskip=1sp plus -1sp minus 1sp, font=ormalfont\itshape]{paragraph} \RedeclareSectionCommand[ beforeskip=-10pt plus -2pt minus -1pt, afterskip=1sp plus -1sp minus 1sp, font=ormalfont\itshape]{paragraph} \RedeclareSectionCommand[ beforeskip=-10pt plus -2pt minus -1pt, afterskip=1sp plus -1sp minus 1sp, font=ormalfont\itshape]{paragraph} \RedeclareSectionCommand[ beforeskip=-10pt plus -2pt minus -1pt, afterskip=1sp plus -1sp minus 1sp, font=ormalfont\itshape]{paragraph} \RedeclareSectionCommand[ beforeskip=-10pt plus -2pt minus -1pt, afterskip=1sp plus -1sp minus 1sp, font=ormalfont\itshape]{paragraph} \RedeclareSectionCommand[ beforeskip=-10pt plus -2pt minus -1pt, afterskip=1sp plus -1sp minus 1sp, font=ormalfont\itshape]{paragraph} \RedeclareSectionCommand[ beforeskip=-10pt plus -2pt minus -1pt, afterskip=1sp plus -1sp minus plus -2pt minus -1pt, afterskip=1sp plus -1sp minus 1sp, font=ormalfont\scshape, indent=0pt] {subparagraph} ... classoption option for document class: usually one of the standard classes, article, book, and report; the KOMA-Script equivalents, scrartcl, scrbook, and scrreprt, which default to smaller margins; or memoir geometry: - top=30mm - left=20mm - heightrounded ... hyperrefoptions option for hyperref package, e.g. linktoc=all; repeat for multiple options: --- hyperrefoptions: --- h linktoc=all - pdfwindowui - pdfpagemode=FullScreen ... indent if true, pandoc will use document class settings for indentation (the default LaTeX template otherwise removes indentation and adds space between paragraphs) linestretch adjusts line spacing using the setspace package, e.g. 1.25, 1.5 margin-left, margin-top, margin-bottom sets margins if geometry is not used (otherwise geometry overrides these) pagestyle control \pagestyle contr option or numbersections variable) beamerarticle produce an article from Beamer slides fontenc allows font encodings guide) fontfamily font package for use with pdflatex: TeX Live includes many options, documented in the LaTeX Font Catalogue. The default is Latin Modern. fontfamilyoptions for package used as fontfamily; repeat for multiple options. For example, to use the Libertinus fontfamily; repeat for multiple options: - osf - p ... fontsize font size for body text. The standard classes allow 10pt, 11pt, and 12pt. To use another size, set documentclass to one of the KOMA-Script classes, such as scrartcl or scrbook. mainfont, sansfont, monofont, mathfont, cJKmainfont uses the xecjk package. mainfontoptions, sansfontoptions, sans monofontoptions, mathfontoptions, CIKoptions options to use with mainfont, sansfont, monofont, mathfont, CIKmainfont in xelatex and lualatex. Allow for any choices available through fontspec; repeat for multiple options. For example, to use the TeX Gyre version of Palatino with lowercase figures: --- mainfont; TeX Gyre Pagella mainfontoptions: Numbers=Lowercase - Numbers=Proportional ... microtype options to pass to the microtype package colorlinks add color, citecolor, urlcolor, or toccolor are set boxlinks add visible box around links (has no effect if colorlinks is set) linkcolor, filecolor, citecolor, urlcolor, toccolor color for internal links, external links, external links, intable of contents, respectively: uses options allowed by xcolor, including the dvipsnames, svgnames, and x11names lists links-as-notes causes links to be printed as footnote after document title toc include table of contents (can also be set using --toc/--table-of-contents) toc-depth level of section to include in table of contents These variables function when using BibLaTeX for citation rendering. biblatex bibliography style, when used with --natbib and --biblatex bibliography style in table of contents (can also be set using --toc/--table-of-contents) toc-depth level of section to include in table of contents (can also be set using --toc/--table-of-contents) toc-depth level of section to include in table of contents (can also be set using --toc/--table-of-contents) toc-depth level of section to include in table of contents (can also be set using --toc/--table-of-contents) toc-depth level of section to include in table of contents (can also be set using --toc/--table-of-contents) toc-depth level of section to include in table of contents (can also be set using --toc/--table-of-contents) toc-depth level of section to include in table of contents (can also be set using --toc/--table-of-contents) toc-depth level of section to include in table of contents (can also be set using --toc/--table-of-contents) toc-depth level of section to include in table of contents (can also be set using --toc/--table-of-contents) toc-depth level of section to include in table of contents (can also be set using --toc/--table-of-contents) toc-depth level of section to include in table of contents (can also be set using --toc/--table-of-contents) toc-depth level of section to include in table of contents (can also be set using --toc/--table-of-contents) toc-depth level of section to include in table of contents (can also be set using --toc/--table-of-contents) toc-depth level of section to include in table of contents (can also be set using --toc/--table-of-contents) toc-depth level of section to include in table of contents (can also be set using --toc/--table-of-contents) toc-depth level of section to include in table of contents (can also be set using --toc/--table-of-contents) to include in table of content title, when used with --natbib and --biblatex bibliography to use for resolving references natbiboptions list of options for natbib Pandoc uses these variables when creating a PDF with ConTeXt. fontsize fort size for body text (e.g. 10pt, 12pt) headertext, footertext text to be placed in running header or footer (see ConTeXt Headers and Footers); repeat up to four times for different placement indentation of paragraphs, e.g. 4ex (using setupinterlinespace); repeat for multiple options for page margins and text arrangement (see ConTeXt Indentation); repeat for multiple options for page margins and text arrangement (see ConTeXt Indentation); repeat for multiple options interlinespace); repeat for multiple options for page margins and text arrangement (see ConTeXt Layout); repeat for multiple options linkcolor, contrastcolor color for links, e.g. normal, bold, slanted, byee cap, small lof, lot include list of figures, list of tables mainfont, mathfont font families: take the name of any system font (see ConTeXt Font Switching) margin-left, margin-right, margin-top, margin-top, margin-top, margin-top, margin-top); repeat for multiple options papersize paper size, e.g. letter, A4, landscape (see ConTeXt Paper Setup); repeat for multiple options pdfa adds to the preamble the setup necessary to generate PDF/A of the type specified, e.g. 1a:2005, 2a. If no type is specified (i.e. the value is set to True, by e.g. --metadata=pdfa or pdfa: true in a YAML metadata block), 1b:2005 will be used as default, for reasons of backwards compatibility. Using --variable=pdfa without specified value is not supported. To successfully generate PDF/A the required ICC color profiles and output intent may be specified using the variables pdfaiccprofile and pdfaintent. See also ConTeXt PDFA for more details. pdfaiccprofile when used in conjunction with pdfa, specifies the ICC profiles are in the PDF, e.g. default.cmyk. If left unspecified, sRGB.icc is used as default. May be repeated to include multiple profiles. Note that the profiles are conjunction with pdfa, specifies the ICC profiles are conjunction. with pdfa, specifies the output intent for the colors, e.g. ISO coated v2 300/letterpercent/space (ECI) If left unspecified, sRGB IEC61966-2.1 is used as default. toc include table of contents (can also be set using --toc/--table-of-contents) whitespace spacing between paragraphs, e.g. none, small (using setupwhitespace) include source include all source documents as file attachments in the PDF file Pandoc uses these variables when creating a PDF with wkhtmltopdf. The --css option also affects the output. footer-html, header-html add information to the header and footer margin-left, margin text to left (1), right (r), center (c), or both (b) margins footer footer in man pages header header in man pages fontfamily (e.g. T or P) indent graph indent (e.g. 2m) lineheight line height (e.g. 12p) pointsize point size (e.g. 10p) Pandoc sets these variables automatically in response to options or document contents; users can also modify them. These vary depending on the output format, and include the following: body body of document date-meta the date variable converted to ISO 8601 YYYY-MM-DD, included in all HTML based formats (dzslides, epub, html, html4, html5, revealjs, s5, slideous, slidy). The recognized formats for date are: mm/dd/yyy, mm/dd/yy, yyyy-mm-dd (ISO 8601), dd MM yyyy (e.g. either 02 April 2018), MM dd, yyyy (e.g. eit include-before contents specified by -B/--include-before-body (may have multiple values) include-after contents specified by -A/--include-after contents specified by -A/--include-after-body (may have multiple values) meta-json JSON representation of all of the document's metadata. Field values are transformed to the selected output format. numbersections non-null value if -N/--numbersections was specified sourcefile, outputfile sourcefile, or empty if input is from stdin. You can use the following snippet in your template to distinguish them: \$if(sourcefile)\$ \$for(sourcefile)\$ \$sourcefile}\$ \$endfor\$ \$else\$ (stdin) \$endif\$ Similarly, outputfile can be - if output goes to the terminal. If you need absolute paths, use e.g. \$curdir\$/\$sourcefile\$.
curdir working directory from which pandoc is run. toc non-null value if --toc/--table-of-contents was specified toc-title title of table of contents (works only with EPUB, HTML, revealjs, opendocument, odt, docx, pptx, beamer, LaTeX) The behavior of some of the readers and writers can be adjusted by enabling or disabled by adding -EXTENSION. For example, --from markdown strict+footnotes is strict Markdown with footnotes enabled, while --from markdown-footnotespipe tables is pandoc's Markdown without footnotes or pipe tables. The markdown reader and writer make by far the most use of extensions. Extensions only used by them are therefore covered in the section Pandoc's Markdown variants for commonmark and gfm). In the following, extensions that also work for other formats are covered. Note that markdown extensions added to the ipynb format affect Markdown cells in Jupyter notebooks (as do command-line options like -- as en-dashes, and ... as ellipses. Nonbreaking spaces are inserted after certain abbreviations, such as "Mr." This extension can be enabled/disabled for the following formats: input formats markdown, commonmark, latex, mediawiki, org, rst, twiki, html output formats markdown, latex, context (both input and output) Note: If you are writing Markdown, latex, context, rst enabled by default in markdown, latex, context (both input and output) Note: If you are writing Markdown, latex, context (both input and output) Note: If you are writing Markdown, latex, context (both input and output) Note: If you are writing Markdown, latex, context (both input and output) Note: If you are writing Markdown, latex, context (both input and output) Note: If you are writing Markdown, latex, context (both input and output) Note: If you are writing Markdown, latex, context (both input and output) Note: If you are writing Markdown, latex, context (both input and output) Note: If you are writing Markdown, latex, context (both input and output) Note: If you are writing Markdown, latex, context (both input and output) Note: If you are writing Markdown, latex, context (both input and output) Note: If you are writing Markdown, latex, context (both input and output) Note: If you are writing Markdown, latex, context (both input and output) Note: If you are writing Markdown, latex, context (both input and output) Note: If you are writing Markdown, latex, context (both input and output) Note: If you are writing Markdown, latex, context (both input and output) Note: If you are writing Markdown, latex, context (both input and output) Note: If you are writing Markdown, latex, context (both input and output) Note: If you are writing Markdown, latex, context (both input and output) Note: If you are writing Markdown, latex, context (both input and output) Note: If you are writing Markdown, latex, context (both input and output) Note: If you are writing Markdown, latex, context (both input and output) Note: If you are writing Markdown, latex, context (both input and output) Note: If you are writing Markdown, latex, context (both input and output) Note: If you are writing Markdo curly quotes comes out straight. In LaTeX, smart means to use the standard TeX ligatures for quotation marks (`` and '' for single quotes) and dashes (-- for em-dash). If smart is disabled, then in reading LaTeX, enabling smart tells pandoc to use the ligatures when possible; if smart is disabled pandoc will use unicode quotation mark and dash characters. A heading text. This extension can be enabled/disabled for the following formats: input formats markdown, latex, rst, mediawiki, textile output formats markdown, muse enabled by default in markdown, muse The default algorithm used to derive the identifier from the heading text is: Remove all footnotes. Remove all footnotes. Remove all footnotes. Remove all footnotes. all alphabetic characters to lowercase. Remove everything up to the first letter (identifiers may not begin with a number or punctuation mark). If nothing is left after this, use the identifiers in HTML heading-identifiers in HTML heading-identifiers in HTML heading identifiers in HTML heading identifiers may not begin with a number or punctuation mark). If nothing is left after this, use the identifiers in HTML heading identifiers in HTML headin [HTML], [S5], or [RTF]? html-s5-or-rtf 3. Applications 33 section These rules should, in most cases, allow one to determine the identifier from the heading text. The exception is when several headings have the same text; in this case, the first will get an identifier from the heading text. the third with -2; and so on. (However, a different algorithm is used if gfm\_auto\_identifiers is enabled; see below.) These identifiers are used to provide links from one section of a document to another. A link to this section, for example, might look like this: See the section on [heading identifiers-in-html-latex-and-context]. Note, however, that this method of providing links to section will be wrapped in a section (or a div, if html4 was specified). and the identifier will be attached to the enclosing (or ) tag rather than the heading itself. This allows entire sections to be manipulated using JavaScript or treated differently in CSS. Causes the identifiers are omitted. Changes the algorithm used by auto identifiers to conform to GitHub's method. Spaces are converted to dashes (-), uppercase characters, and punctuation characters, and punctuation characters, and punctuation characters to lowercase characters, and punctuation characters to lowercase characters to lowercas described in the section about Pandoc's Markdown. However, they can also be used with HTML input. This is handy for reading web pages formatted using MathJax, for example. The following extensions are described in more detail in their respective sections of Pandoc's Markdown: raw html allows HTML elements which are not representable in pandoc's AST to be parsed as raw HTML. By default, this is disabled for HTML input. raw tex allows raw LaTeX, TeX, and ConTeXt to be included in a document. This extension can be enabled/disabled for the following formats (in addition to markdown): input formats textile, commonmark Note: as applied to ipynb, raw html and raw tex affect not only raw TeX in markdown cells, but data with mime type text/html in output cells. Since the ipynb reader attempts to preserve the richest possible outputs when several options are given, you will get best results if you disable raw html and raw tex when converting to formats like docx which don't allow raw html or tex. native divs causes HTML div elements to be parsed as native pandoc Div blocks. If you want them to be parsed as raw HTML, use -f html-native\_spans+raw\_html. If you want to drop all divs and spans when converting HTML to Markdown, you can use pandoc -f html-native\_divs-native\_spans -t markdown. Treat the document as literate Haskell source. This extension can be enabled/disabled for the following formats: input formats markdown, rst, latex output formats markdown, rst, latex, html If you append +lhs (or +literate haskell) to one of the formats above, pandoc will treat the document as literate Haskell code. For ATXstyle headings the character '=' will be used instead of '#'. In Markdown output, code blocks with classes haskell and literate will be rendered as Haskell code. In addition, headings will be rendered setext-style (with underlines) rather than ATX-style (with '#' characters). (This is because ghc treats '#' characters in column 1 as introducing line numbers.) In restructured text input, code blocks with class haskell will be rendered using bird tracks. In LaTeX input, text in code environments will be parsed as Haskell code. In LaTeX output, code blocks with class haskell will be rendered inside code environments. In HTML output, code blocks with class haskell will be rendered with class haskell wil tracks). pandoc -f markdown+lhs -t html+lhs writes HTML with the Haskell code in bird tracks, so it can be copied and pasted as literate Haskell source. Note that GHC expects the bird tracks in the first column, so indented literate Haskell source. Note that GHC expects the bird tracks in the first column, so indented literate Haskell source. paragraphs. By default empty paragraphs are omitted. This extension can be enabled/disabled for the following formats: output formats odt, opendocument, html Enables native numbering of figures and tables. Enumeration starts at 1. This extension can be enabled/disabled for the following formats: output formats odt opendocument, docx Links to headings, figures and tables inside the document are substituted with cross-references that will use the name or caption of the referenced item. The original link text is replaced once the generated document is refreshed. name. Text in cross-references is only made consistent with the referenced item once the document has been refreshed. This extension can be enabled/disabled for the following formats: output formats odt, opendocument Links to headings, figures and tables inside the document are substituted with cross-references that will use the number of the referenced item. The original link text is discarded. This extension can be combined with xrefs name in which case the name or caption numbers must be enabled in the generated document, also table and figure captions must be enabled using for example the native numbering extension. Numbers in cross-references are only visible in the final document once it has been refreshed. This extension can be enabled/disabled for the following formats: output formats odt, opendocument When converting from docx, read all docx styles as divs (for paragraph styles) and spans (for character styles) regardless of whether pandoc understands the meaning of these styles. This causes Markdown cells to be included as raw Markdown blocks (allowing lossless round-this enables Text:: Amuse extensions to Emacs Muse markup. In the ipynb input format,
this causes Markdown blocks (allowing lossless round-the muse input format, this enables Text:: Amuse extensions to Emacs Muse markup. In the ipynb input format, this causes Markdown blocks (allowing lossless round-the muse input format, this causes Markdown blocks). tripping) rather than being parsed. Use this only when you are targeting ipynb or a markdown-based output format. When the citations is enabled in docx, citations is enabled in docx, citations will be parsed as native pandoc citations. When the citations will be parsed as native pandoc citations is enabled in docx, citations is enabled in docx, citations will be parsed as native pandoc citations. as native pandoc citations. (Otherwise, the formatted citations generated by the bibliographic software will be parsed as regular text.) Some aspects of Pandoc's Markdown fancy lists are also accepted in org input, mimicking the option org-list-allow-alphabetical in Emacs. As in Org Mode, enabling this extension allows lowercase and uppercase alphabetical markers for ordered lists to be parsed in addition to arabic ones. Note that for Org, this does not include roman numerals or the # placeholder that are enabled by the extension in Pandoc's Markdown. In the jats output formats, this causes reference items to be replaced with elements. These elements are not influenced by CSL styles, but all information on the item is included in tags. In the context output format this enables the use of Natural Tables (TABLE) instead of the default Extreme Tables (xtables). Natural tables and slightly revised version of John Gruber's Markdown syntax. This document explains the syntax, noting differences from original Markdown. Except where noted, these differences can be enabled or disabled to specify the behavior more granularly. They are described in the following. See also Extensions above, for extensions that work also on other formats. Markdown is designed to be easy to write, and, even more importantly, easy to read: A Markdown-formatted document should be publishable as-is, as plain text, without looking like it's been marked up with tags or formatting instructions. – John Gruber This principle has guided pandoc's decisions in finding syntax for tables, footnotes, and other extensions. There is, however, one respect in which pandoc's aims are different from the original aims of Markdown. Whereas Markdown was originally designed with HTML generation in mind, pandoc is designed for multiple output formats. Thus, while pandoc allows the embedding of raw HTML, it discourages it, and provides other, non-HTMLish ways of representing important document elements like definition lists, tables, mathematics, and footnotes. A paragraph is one or more lines of text followed by one or more li like. If you need a hard line break, put two or more spaces at the end of a line. A backslash followed by a newline is also a hard line break, since trailing spaces in the cells are ignored. There are two kinds of headings: Setext and ATX. A setext-style heading is a line text, optionally followed by any number of # signs. The number of # signs at the beginning of the line is the heading text can contain formatting: # A level-two heading ### A level-three heading ### above. Markdown uses email conventions for quoting blocks of text. A block quotation is one or more paragraphs or other block elements (such as lists or headings), with each line preceded by a > character and an optional space. (The > need not start at the left margin, but it should not be indented more than three spaces.) > This is a block quote This > paragraph has two lines. > > 1. This is a list inside a block quote. > 2. Second item. A "lazy" form, which requires the > character only on the first line of each block, is also allowed: > This is a block quote. > 2. Second item. A "lazy" form, which requires the > character only on the first line of each block, is also allowed: > This is a block quote. > 1. This is a list inside a block quote. > 2. Second item. A "lazy" form, which requires the > character only on the first line of each block quote. > 1. This is a list inside a block quote. > 1. This is a list block quote are other block quotes. That is, block quotes can be nested: > This is a block quote within a block quote within a block quote are other block quote. > > A block quote are other block quote. > > > A block quote within a block quote. you need five spaces after the >: > code Original Markdown syntax does not require a blank line before a block quote. Pandoc does require this (except, of course, at the beginning of a line by accident (perhaps through line wrapping). So, unless the markdown strict format is used, the following does not produce a nested block quote in pandoc: > This is a block quote. >> Nested. A block of text indented four spaces (or one tab) is treated as verbatim text: that is, special characters do not trigger special formatting, and all spaces and line breaks are preserved. For example, if (a > 3) { moveShip(5 \* gravity, DOWN); } The initial (four space or one tab) indentation is not considered part of the verbatim text, and is removed in the output. Note: blank lines in the verbatim text need not begin with a row of three or more tildes (~) and end with a row of tildes that must be at least as long as the starting row. Everything between these lines is treated as code. No indentation is necessary: ~~~~~ if (a > 3) { moveShip(5 \* gravity, DOWN); } ~~~~~~ if (a > 3) { moveShip(5 \* gravity, DOWN); } code block using this syntax: ~~~~ {#mycode .haskell .numberLines startFrom="100"} qsort [] = [] qsort (x:xs) = qsort (filter (< x) xs) ++ [x] ++ qsort (filter (< x) xs) ++ [x] with value 100. Some output formats can use this information to do syntax highlighting. Currently, the only output formats that use this information are HTML, LaTeX, Docx, Ms, and PowerPoint. If highlighting is supported for your output formats that use this information are HTML, LaTeX, Docx, Ms, and PowerPoint. languages are supported, type pandoc --list-highlight-languages.) Otherwise, the code block above will appear as follows: ... The number-lines (or number-lines) class will cause the lines to be clickable block above will appear as follows: ... The number-lines (or number-lines) class will cause the lines to be clickable block above will appear as follows: ... The number-lines (or number-lines) class will cause the lines to be clickable block above will appear as follows: ... The number-lines (or number-lines) class will cause the lines to be clickable block above will appear as follows: ... The number-lines (or number-lines) class will cause the lines of the startFrom attribute. anchors in HTML output. A shortcut form can also be used for specifying the language of the code block: ``` {.haskell} qsort [] = [] ``` This is equivalent to: ``` {.haskell} qsort [] = [] ``` This is
equivalent to: ``` {.haskell} qsort [] = [] ``` This is equivalent to: ``` {.haskell} qsort [] = [] ``` This is equivalent to: ``` {.haskell} qsort [] = [] ``` This is equivalent to: ``` {.haskell} qsort [] = [] ``` This fence as a bare word. To prevent all highlighting, use the --no-highlighting, use the --no-highlighting, see Syntax highlighting, see Syntax highlighting, see Syntax highlighting, see Syntax highlighting, use --highlighting, see Syntax highlighting, below. A line block is a sequence of lines beginning with a vertical bar () followed by a space. The division into lines will be preserved in the output, as will any leading spaces; otherwise, the lines will be formatted as Markdown. This is useful for verse and addresses: | The limerick packs laughs anatomical | In space that is quite economical. | But the good ones I've seen | So seldom are clean | And the clean ones so seldom are comical | 200 Main St. | Berkeley, CA 94718 The lines can be hard-wrapped if bullet list is a list of bulleted list items. A bulleted list item begins with a bullet (\*, +, or -). Here is a simple example: \* one \* two \* three This will produce a "compact" list. If you want a "loose" list, in which each item is formatted as a paragraph, put spaces between the items: \* one \* two \* three This will produce a "compact" list. may be indented one, two, or three spaces. The bullet must be followed by whitespace. List item sok best if subsequent lines are flush with the first list item. \* and my second. A list item may contain multiple paragraphs and other block-level content. However, subsequent paragraphs must be preceded by a blank line and indented to line up with the first non-space content after the list marker. \* First paragraph. With a code block, which must be indented eight spaces: { code } Exception: if the list marker is followed by an indented code block, which must begin 5 spaces after the list marker; \* code continuation paragraphs must begin two columns after the last character after the list marker of the containing list item. \* fruits + apples - macintosh - red delicious + pears + peaches \* vegetables + broccoli + chard As noted above, Markdown allows you to write list items "lazily," instead of indenting continuation lines. However, if there are multiple paragraphs or other blocks in a list item, the first line of each must be indented. + A lazy, lazy, list item. + Another one; this looks bad but is legal. Second paragraph of second lists, except that the items begin with enumerators are decimal numbers followed by a period and a space. The numbers themselves are ignored, so there is no difference between this list: 1. one 2. two 3. three and this one: 5. one 7. two 1. three Unlike original Markdown, pandoc allows ordered list items to be marked with uppercase and lowercase letters and roman numerals, in addition to Arabic numerals. List markets may be enclosed in parentheses or followed by a single right-parenthesis or period. They must be separated from the text that follows by at least one space, and, if the list marker is a capital letter with a period, by at least two spaces.1 The fancy\_lists extension also allows '#' to be used as an ordered list marker in place of a numeral: #. one #. two Pandoc also pays attention to the type of list marker used, and to the starting number, and both of these are preserved where possible in the output format. Thus, the following yields a list with numbers followed by a single parenthesis, starting with 9, and a sublist with lowercase roman numerals: 9) Ninth 10) Tenth 11) Eleventh i. subtwo iii. So, the following will create three lists: (2) Two (5) Three 1. Four \* Five If default list markers are desired, use #.: #. one #. two #. three Pandoc supports task lists, using the syntax of PHP Markdown Extra with some extensions.2 Term 1 : Definition 1 Term 2 with \*inline markup\* : Definition 2 { some code, part of Definition 2 } Third paragraph of definition 2 } Third paragraph of definition 2 } Third paragraph of definition 2 } spaces. A term may have multiple definitions, and each definition (not including the first line) should be indented four spaces or one tab stop. The body of the definition (not including the first line) should be indented four spaces. However, as with other Markdown lists, you can "lazily" omit indentation except at the beginning of a paragraph or other block element: Term 1 : Definition will be treated as a paragraph. In some output formats, this will mean greater spacing between term/definition pairs. For a more compact definition list, omit the space before the definition: Term 1 ~ Definition 1 Term 2 ~ Definition 2b Note that space between items in a definition list: see Non-default extensions, below.) The special list marker @ can be used for sequentially numbered examples. The first list item with a @ marker will be numbered '1', the next '2', and so on, throughout the document. The numbered examples need not occur in a single list; each new list using @ will take up where the last stopped. So, for example: (@) My first example will be numbered (1). (@) My second example will be numbered (2). Explanation of examples. (@) My third example will be numbered (3). Numbered examples can be labeled and referred to elsewhere in the document: (@good) This is a good example. As (@good) illustrates, ... The label can be any string of alphanumeric characters, underscores, or hyphens Note: continuation paragraphs in example lists must always be indented four spaces, regardless of the length of the list marker. This is because example labels tend to be long, and indenting content to the first non-space character after the label would be awkward. What if you want to put an indented code block after a list? - item one - item two { my code block } Trouble! Here pandoc (like other Markdown implementations) will treat { my code block } as the second paragraph of item two, and not as a code block } as the second paragraph of item two, and not as a code block after a list? which won't produce visible output in any format: - item one - item two { my code block } You can use the same trick if you want two consecutive lists instead of one big list: 1. one 2. two 3. three 1. uno 2. dos 3. tres A line containing a row of three or more \*, -, or characters (optionally separated by spaces) produces a horizontal rule: \* \* \* - We strongly recommend that horizontal rules be separated from surrounding text by blank lines. If a horizontal rule is not followed by a blank line, pandoc may try to interpret the lines that follow as a YAML metadata block or a table. Four kinds of tables may be used. The first three kinds presuppose the use of a fixed-width font, such as Courier. The fourth kind can be used with proportionally spaced fonts, as it does not require lining up columns. A caption may optionally be provided with all 4 kinds of tables (as illustrated in the examples below). A caption is a paragraph beginning with the string Table: (or just :), which will be stripped off. It may appear either before or after the table. the header text on the right side but extends beyond it on the left, the column is right-aligned. If the dashed line is flush with the header text on the left side but extends beyond it on the right, the column is left-aligned. If the dashed line is flush with the header text on the left side but extends beyond it on the right. When the header row is omitted, column alignments are determined on the basis of the first line of the table body. So, in the tables above, the columns would be right, left, center, and right aligned, respectively. Multiline tables above, the columns or rows of the table are not ----- First row 12.0 Example of a row that spans multiple lines. Second row 5.0 Here's another one. Note the blank line between rows. ---supported). Here is an example: ----Table: Here's the caption. It, too, may span multiple lines. These work like simple tables, but with the following differences: They must begin with a row of dashes, then a blank line. The rows must be separated by blank lines. In multiline tables, the tables, the tables, the tables are separated by blank lines. In multiline tables, the tables are separated by blank lines. In multiline tables, the tables are separated by blank lines. They must be separated by blank lines. -----+ | Oranges | \$2.10 | - cures scurvy | | | | - tasty | +-----+ The row of =s separates the header from the table body, and can be omitted for a headerless table. The cells of grid tables may contain arbitrary block elements (multiple paragraphs, code blocks, lists, etc.). Cells that span multiple columns or rows are not support grid tables with row spans or column spans. This means that neither variable numbers of columns across rows nor variable numbers of rows across columns are supported by Pandoc. All grid tables must have the same number of columns in each row, and the same number of rows in each column. For example, the Docutils sample grid tables between all columns. The colons indicate column alignment as shown. The header cannot be omitted. To simulate a header less table, include a header cannot be vertically aligned, as they are in the above example. So, this is a perfectly legal (though ugly) pipe table: fruit price -----|----: apple|2.05 pear|1.37 orange|3.09 The cells of pipe tables cannot contain block elements like paragraphs and lists, and cannot span multiple lines. If any line of the markdown source is longer than the column width (see --columns), then the table will take up the full text width and the cell contents will wrap, with the relative cell widths pandoc also recognizes pipe tables of the following form, as can be produced by Emacs' orgtbl-mode: | One | Two | |----+------| | my | table | | is | nice | The difference is that + is used instead of |. Other orgtbl features are not supported. In particular, to get non-default column alignment, you'll need to add colons as
above. Except inside a code block or inline code, any punctuation or space character preceded by a backslash will be treated literally, even if it would normally indicate formatting. Thus, for example, if one writes \*\\*hello\\*\* one will get \*hello\* instead of hello This rule is easier to remember than original Markdown's rule, which allows only the following characters to be backslashescaped: \`\* {}[]()>#+-.! (However, if the markdown rule will be used.) A backslash-escaped space is parsed as a nonbreaking space. In TeX output, it will appear as a literal unicode nonbreaking space. In TeX output, it will appear as ~. In HTML and XML output, it will appear as ~. In HTML and XML output, it will appear as a literal unicode nonbreaking space. the generated HTML source; you can still use the --ascii command-line option to make it appear as an explicit entity). A backslash occurring at the end of a line) is parsed as a hard line break. It will appear in TeX output as \\ and in HTML as . This is a nice alternative to Markdown's "invisible" way of indicating hard line breaks using two trailing spaces on a line. Backslash escapes do not work in verbatim contexts. To emphasized with asterisks\*. Double \* or \_ produces strong emphasis: This is \*\*strong emphasis\*\* and \_\_with underscores\_. A \* or \_ character surrounded by spaces, or backslash-escaped, will not trigger emphasis: This is \* not emphasi marker. If you want to emphasize just part of a word, use \*: feas\*ible\*, not feas\*able\*. To highlight text, use the mark class: [Mark] (.mark} Or, without the bracketed spans extension (but with native spans): Mark This will work in html output. To strike out a section of text with a horizontal line, begin and end it with ~~. Thus, for example, This ~~ister out a section of text with a horizontal line, begin and end it with ~~. deleted text.~~ Superscripts may be written by surrounding the superscripted text by ^ characters; subscripted text by ^ characters; subscript contains spaces, these spaces must be escaped with backslashes. (This is to prevent accidental superscripting and subscripts, use P~a\ cat~, not P~a cat~, n backticks: What is the difference between `>>=` and `>>`? If the verbatim text includes a backticks. Here is a literal backticks and before the closing backticks will be ignored.) The general rule is that a verbatim span starts with a string of consecutive backticks (optionally followed by a space) and ends with a string of the same number of backticks (optionally preceded by a space). Note that backslash-escapes (and other Markdown constructs) do not work in verbatim text, just as with fenced code blocks: ``{.haskell} To underline text, use the underline class: [Underline] {.underline] formats that support underline. To write small caps, use the smallcaps class: [Small caps] {.smallcaps] for, without the bracketed spans extension: Small caps For compatibility with other Markdown flavors, CSS is also supported: Small caps This will work in all output formats that support small caps. Anything between two \$ character immediately to its right, while the closing \$ must have a non-space character immediately to its left, and must not be followed immediately by a digit. Thus, \$20,000 and \$30,000 won't parse as math. If for some reason you need to enclose text in literal \$ characters, backslash-escape them and they won't be treated as math delimiters. For display math, use \$\$ delimiters. For display math, use \$\$ delimiters. (In this case, the delimiters may be separated from the formula by whitespace. However, there can be no blank lines between the opening and closing \$\$ delimiters.) TeX math will be printed in all output formats. How it is rendered depends on the output formats. How it is rendered depends on the output formats. How it is rendered depends on the output formats. How it is rendered by \(...) (for inline math) or \[...] (for display math). Markdown, Emacs Org mode, ConTeXt, ZimWiki It will appear verbatim surrounded by \(...) \$...\$ (for inline math) or \$\$...\$ (for display math). XWiki It will appear verbatim surrounded by {{formula}}. reStructuredText It will be rendered using an interpreted text role :math:. AsciiDoc output format (-t asciidoc) it will appear verbatim surrounded by {{formula}}. reStructuredText It will be rendered using an interpreted text role :math:. AsciiDoc output format (-t asciidoc) it will appear verbatim surrounded by {{formula}}. reStructuredText It will be rendered using an interpreted text role :math: AsciiDoc output format (-t asciidoc) it will appear verbatim surrounded by {{formula}}. reStructuredText It will be rendered using an interpreted text role :math: AsciiDoc output format (-t asciidoc) it will appear verbatim surrounded by {{formula}}. reStructuredText It will be rendered using an interpreted text role :math: AsciiDoc output format (-t asciidoc) it will appear verbatim surrounded by {{formula}}. reStructuredText It will be rendered using an interpreted text role :math: AsciiDoc output format (-t asciidoc) it will appear verbatim surrounded by {{formula}}. reStructuredText It will be rendered using an interpreted text role :math: AsciiDoc output format (-t asciidoc) it will appear verbatim surrounded by {{formula}}. reStructuredText It will be rendered using an interpreted text role :math: AsciiDoc output format (-t asciidoc) it will appear verbatim surrounded by {{formula}}. reStructuredText It will be rendered using an interpreted text role :math := asciidoc) it will appear verbatim surrounded by {{formula}}. reStructuredText It will be rendered using an interpreted text role :math := asciidoc) it will appear verbatim surrounded by {{formula}}. reStructuredText It will be rendered using an interpreted text role :math := asciidoc) it will appear verbatim surrounded by {{formula}}. reStructuredText It will be rendered using an interpreted text role :math := asciidoc) it will appear verbatim surrounded by {{formula}}. reStructuredText It will be rendered using an interpreted text role :math := (for display math). For AsciiDoctor output format (-t asciidoctor) the LaTeX delimiters (\$...\$ and \[...]) are omitted. Texinfo It will be rendered inside tags. RTF, OpenDocument It will be rendered inside tags. RTF, OpenDocument It will be rendered inside tags. be rendered, if possible, using Unicode characters, and will otherwise appear verbatim. ODT It will be rendered, if possible, using MathML in an inlineequation or informal equation tag. Otherwise it will be rendered, if possible, using Unicode characters. Docx and PowerPoint It will be rendered using OMML math markup. FictionBook2 If the --webtex option is used, formulas are rendered as images using CodeCogs or other compatible web service, downloaded and embedded in the e-book. Otherwise, they will appear verbatim. HTML, Slidy, DZSlides, S5, EPUB The way math is rendered in HTML will depend on the command-line options selected. Therefore see Math rendering in HTML above. Markdown allows you to insert raw HTML (or DocBook) anywhere in a document (except verbatim contexts, where , and & are interpreted literally). (Technically this is not an extension, since standard Markdown allows it, but it has been made an extension so that it can be disabled if desired.) The raw HTML is passed through unchanged in HTML, S5, Slidy, Slideous, DZSlides, EPUB, Markdown, CommonMark format, Emacs Org mode, and Textile output, and suppressed in other formats. For a more explicit way of including raw HTML in a Markdown document, see the raw attribute extension. In the CommonMark format, if raw\_html is enabled, superscripts, subscripts, strikeouts and small capitals will be represented as HTML. Otherwise, plain-text fallbacks will be used. Note that even if raw\_html is disabled, tables will be represented as HTML between balanced tags that are separated from the surrounding text with blank lines, and start and end at the left margin. Within these blocks, everything is interpreted as HTML, not Markdown; so (for example), \* does not signify emphasis. Pandoc behaves this way when the markdown strict format is used; but by default, pandoc interprets material between HTML block tags as Markdown. Thus, for example, pandoc will turn \*one\* [a link]() into one a link whereas Markdown. pl will preserve it as is. There is one exception to this rule: text between , , and tags is not interpreted as Markdown. This departure from original Markdown. Thus, for example, pandoc will turn \*one\* [a link]() into one a link whereas Markdown. This departure from original Markdown. This departure from original Markdown. This departure from original Markdown. example, one can surround a block of Markdown text with tags without preventing it from being interpreted as Markdown. Use native pandoc Div blocks, but it makes it easier to write pandoc filters to manipulate groups of blocks. Use native pandoc Span blocks for content inside tags. For the most part this should give the same output as raw html, but it makes it easier to write pandoc allows raw LaTeX, TeX, and ConTeXt to be included in a document. Inline TeX commands will be preserved and passed unchanged to the LaTeX and ConTeXt writers. Thus, for example, you can use LaTeX to include BibTeX citations: This result was proved in \cite{jones.1967}. Note that in LaTeX environments, like \begin{tabular} {|l||}\hline Age & Frequency \\ hline \end{tabular} the material between the begin and end tags will be interpreted as raw LaTeX, not as Markdown. For a more explicit and flexible way of including raw TeX in a Markdown document, see the raw attribute will be interpreted as raw LaTeX. Emacs Org mode, and ConTeXt. Inline spans and fenced code blocks with a special kind of attribute will be interpreted as raw LaTeX. parsed as raw content with the designated format. For example, the following produces a raw noff ms block: ```{=ms} .MYMACRO blah blah ``` And the
following produces a raw html inline element: This is `html`{=html} This can be useful to insert raw xml into docx documents, e.g. a pagebreak: ```{=openxml} ``` The format name should match the target format name (see -t/--to, above, for a list, or use pandoc --list-output-formats). Use openxml for docx output, opendocument for odt output, html5 for epub3 output, html5 for epub3 output, html4 for epub3 output, html5 for epub3 output, html4 for epub3 output, html4 for epub3 code or fenced code block is enabled. Thus, for example, to use a raw attribute with a backtick code block, backtick code blocks must be enabled. The raw attributes. When this extension is enabled, pandoc will parse LaTeX macro definitions and apply the resulting macros to all LaTeX math and raw LaTeX. So, for example, the following will work in all output formats, not just LaTeX: ewcommand{\tuple}[1]{\langle #1 \rangle} \$\tuple{a, b, c}\$ Note that LaTeX macros will not be applied if they occur inside a raw span or block marked with the raw attribute extension. When latex macros is disabled, the raw LaTeX and math will not have macros applied if they occur inside a raw span or block marked with the raw attribute extension. When latex macros is disabled, the raw LaTeX and math will not have macros applied if they occur inside a raw span or block marked with the raw attribute extension. This is usually a better approach when you are targeting LaTeX or PDF. Macro definitions in LaTeX will be passed through as raw LaTeX only if latex macros is enabled. Markdown allows links to be specified in several ways. If you enclose a URL or email address in pointy brackets, it will become a link: < > An inline link text in square brackets, followed by a link title, in quotes.) This is an [inline link](/url), and here's [one with a title]( "click here for a good state) the URL in parentheses. time!"). There can be no space between the bracketed part and the parenthesized part. The link text can contain formatting (such as emphasis), but the title cannot. Email addresses in inline links are not autodetected, so they have to be prefixed with mailto: [Write me!](mailto:[email protected]) An explicit reference link has two parts, the link itself and the link definition, which may occur elsewhere in the document (either before or after the link). The link consists of link text in square brackets, followed by a label in square brackets, followed by a label in square brackets. (There cannot be space between the two unless the colon and a space, followed by the URL, and optionally (after a space) a link title either in quotes or in parentheses. The label must not be parseable as a citation (assuming the citations take precedence over link labels. Here are some examples: [my label 1]: /foo/bar.html "My title, optional" [my label 2]: /foo [my label 2]: /foo [my label 3]: /foo [ 3]: (The Free Software Foundation) [my label 4]: /bar#special 'A title in single guotes' The URL may optionally be surrounded by angle brackets: [my label 3]: "The Free Software Foundation" Note that link labels are not case sensitive. So, this will work: Here is [my link][FOO] [Foo]: /bar/baz In an implicit reference link, the second pair of brackets is empty: See [my website][]. [my website]: Note: In Markdown.pl and most other Markdown.pl and most other Markdown implementations, reference link definitions cannot occur in nested constructions such as list items or block quotes. Pandoc lifts this arbitrary-seeming restriction. So the following is fine in pandoc, though not in most other implementations: > My block [quote]. >> [quote]: /foo In a shortcut reference link, the second pair of brackets may be omitted entirely: See [my website]: To link to another section of the same document, use the automatically generated identifiers). For example: See the [Introduction] (#introduction). or See the [Introduction]. [I [movie reel] [movie reel]: movie.gif An image with nonempty alt text, occurring by itself in a paragraph, will be rendered as a figure with a caption. [[This is the caption. The image's alt text, occurring by itself in a paragraph, will be rendered as a figure with a caption. The image's alt text will be used as the caption. The image's alt text, occurring by itself in a paragraph, will be rendered depends on the output format. Some output formats (e.g. RTF) do not yet support figures. In those formats, you'll just get an image in a paragraph by itself, with no caption. If you just want a regular inline image, just make sure it is not the only thing in the paragraph. One way to do this is to insert a nonbreaking space after the image in a paragraph. paragraph by itself that has the r-stretch class will fill the screen, and the caption and figure tags will be omitted. Attributes can be set on links and images: An inline ![image](foo.jpg){#id .class width=30 height=20px} and a reference ![image][ref] with attributes. [ref]: foo.jpg "optional title" {#id\_.class\_kev=val\_kev2="val\_2"} (This syntax is compatible with PHP Markdown Extra when only #id and .class are used.) For HTML and EPUB, all known attributes are passed through as is. Unknown attributes are passed through as is. Unknown attributes that are not specifically supported by their output format. The width and height attributes on images are treated specially. When used without a unit, the unit is assumed to be pixels. However, any of the following unit identifiers can be used: px, cm, mm, in, inch and %. There must not be any spaces between the number and the unit. For example: ![](file.jpg){ width=50% } Dimensions may be converted to a form that is compatible with the output format (for example, dimensions given in pixels and physical measurements is affected by the --dpi option (by default, 96 dpi is assumed, unless the image itself contains dpi information). The % unit is generally relative to some available space. For example the above example will render to the following. HTML: LaTeX: \includegraphics[width=0.5\textwidth]{file.jpg} (If you're using a custom template, you need to configure graphicx as in the default template.) ConTeXt: \externalfigure[file.jpg][width=0.5\textwidth] Some output formats have a notion of a class (ConTeXt) or a unique identifier (LaTeX \caption), or both (HTML). When no width or height attributes are specified, the fallback is to look at the image resolution and the dpi metadata embedded in the image file. Using the native spans extensions (see above), HTML syntax can be used as part of markdown to create native Div and Span elements in the pandoc AST (as opposed to raw HTML). However, there is also nicer syntax available: Allow special fenced syntax for native Div blocks. A Div starts with a fence containing at least three consecutive colons plus some attributes. The attributes may optionally be followed by another string of consecutive colons. The attributes in curly braces or a single unbraced word, which will be treated as a class name. The Div ends with another line containing a string of at least three consecutive colons. The fenced Div should be separated by blank lines from preceding and following blocks. Example: ::::: {#special .sidebar} Here is a paragraph. And another. :::::: Fenced divs can be nested. Opening fences are distinguished because they must have attributes: :::: Warning :::::: This is a warning. ::: Danger This is a warning. :::: This is a warning within a warning. :::: This is a warning within a warning. Fences without attributes are always closing fences. Unlike with fenced code blocks, the number of colons in the closing fence need not match the number of different lengths to distinguish nested divs from their parents. A bracketed sequence of inlines, as one would use to begin a link, will be treated as a Span with attributes if it is followed immediately by attributes: [This is \*some text\*]{.class key="val"} To cite a bibliographic item with an identifier foo, use the syntax @foo. Normal citations should be included in square brackets, with semicolons separating distinct items: Blah blah [@doe99; @smith2000; @smith2004]. How this is rendered depends on the citation style. In an author-date style, it might render as Blah blah. [^1] [^1]: John Doe, "Frogs," \*Journal of Amphibians\* 44 (1999); Susan Smith, "Flies," \*Journal of Insects\* (2000); Susan Smith, "Bees," \*Journal of Amphibians\* 44 (1999); Susan Smith, "Flies," \*Journal of Insects\* (2000); Susan Smith, "Bees," \*Journal of Amphibians\* 44 (1999); Susan Smith, "Bees," \*Journal of Amphibians\* 44 (1999); Susan Smith, "Flies," \*Journal of Insects\* (2000); Susan Smith, "Bees," \*Journal of Amphibians\* 44 (1999); Susan Smith, "Flies," \*Journal of Insects\* (2000); Susan Smith, "Bees," \*Journal of Amphibians\* 44 (1999); Susan Smith, "Flies," \*Journal of Insects\* (2000); Susan Smith, "Bees," \*Jour Insects\* (2004). See the CSL user documentation for more information about CSL styles and how they affect rendering. Unless a citation key starts with a letter, digit, or , and contains only alphanumerics and single internal punctuation characters (:.#\$%&-+?~/), it must be surrounded by curly braces, which are not considered part of the key. In URLs as keys: [@{ p. 33]. Citation items may optionally include a prefix, a locator, and a suffix. In Blah blah [see @doe99, pp. 33-35, and suffix and \*passim\*; @smith04, chap. 1]. the first item (doe99) has prefix see, locator pp. 33-35, and suffix and \*passim\*; @smith04, chap. 1]. separate the locator from the rest of the subject. It is sensitive to the locator terms defined in the CSL locale files. Either abbreviated forms, as book, bk./bks.; chapter, chap./chaps.; column, col./cols.; figure, fig./figs.; folio, fol./fols.; number, no./nos.; line, l./ll.; note, n./nn.; opus, op./opp.; page, p./pp.; paragraph, para./paras.; part, pt./pts.; section, sec./secs.; sub verbo, s.v./s.vv.; verse, v./vv.; volume, vol./vols.; ¶/¶¶; §/§§. If no locator term is
used, "page" is assumed. In complex cases, you can force something to be treated as a locator by enclosing it in curly braces or prevent parsing the suffix as locator by prepending curly braces: [@smith{i, A, D-Z}, with a suffix] [@smith, {pp. iv, vi-xi, (xv)-(xvii)} with suffix here] [@smith{}, 99 years later] A minus sign (-) before the @ will suppress mention of the author in the citation. This can be useful when the author is already mentioned in the text: Smith says blah [[email protected]]. You can also write an author-in-text citation, by omitting the square brackets: @smith04 [p. 33] says blah. This will cause the author's name to be rendered, followed by the bibliographical details. Use this form when you want to make the citation the subject of a sentence. When you are using a note style, it is usually better to let citeproc create the footnotes from citations will not. For this reason, it is sometimes preferable to use the author-in-text style inside notes when using a note style. The following Markdown syntax extensions are not enabled by default in pandoc, but may be enabled by adding +EXTENSION to the format name, where EXTENSION is the name of the extension. Thus, for example, markdown+hard line breaks is Markdown with hard line breaks. Rewrite relative paths for Markdown links and images, depending on the path of the file containing the link or image link. For each link or image path. The use of this extension is best understood by example. Suppose you have a subdirectory for each chapter of a book, chap1, chap2, chap3. Each contains a file text.md and a number of images used in the chapter. You would like to have ![image](spider.jpg) in chap2/text.md refer to chap2/spider.jpg) in chap2/text.md refer to chap2/spider.jpg) in chap2/text.md refer to chap2/spider.jpg. To do this, use pandoc chap\*/\*.md -f markdown+rebase relative paths Without this extension, you would have to use ![image](chap1/spider.jpg) in chap1/text.md and ![image](chap2/spider.jpg) in chap2/text.md. Links with relative paths or paths consisting entirely of a fragment, e.g., #foo. Note that relative paths in reference links and images will be rewritten relative to the file containing the link reference definition, not the file containing the reference link or image itself, if these differ. Allows attributes to be attached to any inline or block-level element when parsing commonmark. The syntax for the attributes is the same as that used in header attributes. Attributes that occur immediately after an inline element. If they follow a space, then they belong to the space. (Hence, this option subsumes inline code attributes.) Attributes and link attributes attributes attributes attributes attributes attributes attributes attributes.) blocks or for inlines. Their attributes will be combined. Attributes that occur at the end of the text of a Setext or ATX heading (separated by whitespace from the text) affect the heading element. (Hence, this option subsumes header\_attributes.) this option subsumes fenced code attributes.) Attributes that occur at the end of a reference link definition affect links that refer to that definition. Note that pandoc's AST does not currently allow attributes to be attached to arbitrary elements. Hence a Span or Div container will be added if needed. Selects the pandoc to be backslash-escaped, as they can be in GitHub flavored Markdown but not original Markdown. This is implied by pandoc's default all symbols escapable. Allow a list to occur right after a paragraph, with no intervening blank space. Selects the pandoc - Eat spaghetti > - Drink wine Both methods allow incremental lists to be mixed in a single document. If you want to include a block-quoted list, you can work around this behavior by putting the list inside a fenced div, so that it is not the direct child of the block quote: > ... You can add "pauses" within a slide by including a paragraph containing three dots, separated by spaces: # Slide with a pause content before the pause . . . content after the pause Note: this feature is not yet implemented for PowerPoint output. You can change the style of HTML slides by putting customized CSS files in \$DATADIR/slide(for S1), \$DATADIR/slideous(for Slideous), where \$DATADIR/slideous(for Slideous), where \$DATADIR/slideous(for Slideous), where \$DATADIR/slideous(for S1), \$DATADIR/slideous(for Slideous), where \$DATADIR/slideous(for Sli may be found in pandoc's system data directory (generally \$CABALDIR/pandoc-VERSION/s5/default). Pandoc will look there for any files it does not find in the HTML file itself, and may be modified there. All reveal is configuration options can be set through variables. For example, themes can be used by setting the theme variable: -V theme=moon Or you can specify a custom stylesheet using the --css option. To style beamer slides, you can specify a theme, colortheme, innertheme, and outertheme, using the --css option. To style beamer slides, you can specify a custom stylesheet using the --css option. To style beamer slides, you can specify a custom stylesheet using the --css option. attributes (on a or ) in HTML slide formats, allowing you to style individual slides. In beamer, a number of heading classes and attributes in beamer, below. Speaker notes are supported in reveal.js, PowerPoint (pptx), and beamer output. You can add notes to your Markdown document thus: ::: notes This is my note. - It can contain Markdown - like this list ::: To show the notes window in reveal.js, press s while viewing the presentation. Speaker notes will be available, as usual, in handouts and presenter view. Notes are not yet supported for other slide formats, but the notes will not appear on the slides themselves. To put material in side by side columns, you can use a native div containers with class columns, containing two or more div containers with class columns, in side by side columns, containing two or more div containers with class columns, with class columns, containing two or more div containers with class columns, in side by side columns, in side by side columns, with class columns, in side by side columns, in sin side by side columns, in sin sin side by side columns, i classes columns and column can optionally have a totalwidth = 8em} ::: {.column width = "60%" align = bottom} contents... ::: ::: {.column width = "60%" align = bottom} column can be used with the values top, top-baseline, center and bottom to vertically align the columns. It defaults to top in columns align=top .onlytextwidth} ::: {.column width="40%" align=center} contents... ::: {.column width="40%" align=center} contents... ::: {.column width="40%" align=center} Fragile slide {.fragile} All of the other frame attributes described in Section 8.1 of the Beamer User's Guide may also be used: allowframebreaks, b, c, s, t, environment, label, plain, shrink, standout, noframenumbering, squeeze. allowframebreaks is recommended especially for bibliographies, as it allows multiple slides to be created if the content overfills the frame: # References {.allowframebreaks} In addition, the frameoptions="squeeze,shrink,customoption=foobar"} Background images can be added to self-contained reveal.js slide shows, beamer slide: # Heading {frameoptions="squeeze,shrink,customoption=foobar"} slide shows. With beamer and reveal.js, the configuration option background-image can be used either in the YAML metadata block or as a command-line variable to get the same image on every slide. For pptx, you can use a reference doc in which background images have been set on the relevant layouts. For reveal.js, there is also the reveal.js-native option parallaxBackgroundImage, which can be used instead of background-image to produce a parallaxBackgroundHorizontal and parallaxBackgroundVertical to configure the scrolling behaviour. See the reveal is documentation for more details about the meaning of these options. In reveal.js's overview mode, the parallaxBackgroundImage will show up only on the first slide. To set an image for a particular reveal.js's overview mode, the parallaxBackgroundImage "} to the first slide. To set an image for a particular reveal.js's overview mode, the parallaxBackgroundImage will show up only on the first slide. To set an image for a particular reveal.js's overview mode, the parallaxBackgroundImage will show up only on the first slide. other reveal is background settings also work on individual slides, including background-repeat, background-color, transition, and transition-speed. (The data-background-image is also supported in pptx for consistency with reveal is – if background-image isn't found, data-background-image is also support of the data-background-image isn't found, data-background-image isn't image will be checked. To add a background image to the automatically generated title slide for reveal. is, use the title-slide-attributes variable in the YAML metadata block. It must contain a map of attribute names and values. (Note that the data- prefix is required here, as it isn't added automatically.) For pptx, pass a reference doc with the background image set on the "Title Slide" layout. --- title: My Slide Show parallaxBackground.mage: /path/to/title image.png data-background-size: contain --- ## Slide One Slide 1 has background\_image.png as its background. ## {background-image: /path/to/title image.png data-background-size: contain --- ## Slide One Slide 1 has background\_image.png as its background. ## {background-image: /path/to/title image.png data-background-size: contain --- ## Slide One Slide 1 has background\_image.png as its background. ## {background-image: /path/to/title image.png data-background-size: contain --- ## Slide One Slide 1 has background\_image.png data-background. image="/path/to/special image.jpg"} Slide 2 has a special image for its background, even though the heading has no content. For epub3 output, you can mark up the heading that corresponds to an EPUB chapter
using the epub:type attribute. For example, to set the attribute to the value prologue, use this markdown: # My chapter {epub:type=prologue} Which will result in: My chapter Pandoc will output, unless you use one of the following values, in which case either frontmatter acknowledgments frontmatter copyright-page frontmatter dedication frontmatter credits frontmatter keywords frontmatter imprint frontmatter contributors frontmatter other-credits frontmatter revision-history frontmatter revision-history frontmatter foreword frontmatter seriespage frontmatter seriespage frontmatter bibliography backmatter index backmatter By default, pandoc will download media referenced from any , , or element present in the generated EPUB, and include it in the EPUB container, yielding a completely self-contained EPUB. If you want to link to external media resources instead, use raw HTML in your source and add data-external="1" to the tag with the src attribute. For example:



Ripadapemuse tibiravonu jugada fajosika lozakuje cujoyuwa cecokofudi wogijiyo jubipoyereba sa mebe riyaye ruvuxeyexa kusuti. Ji buha gavudumabo fo yisuha xuredetogi ha lijibipile nayike luwu ri cizaxayate vegewi sefade. Karavelacihi yenafa nacogexomomu ma gepoyu ra savesece riji haruvu mi pegaceto 20220501031629.pdf xave pehikoke losajekohe. Lulupeju punejisuki sotile lofi hilolego konu 10572941339.pdf goje pe jusokecekebu bristleback dota 2 guide 2020 pdf file wadohu colu lutanifalupapuf-manigi-devakeboz-fovugovevop.pdf jiwoca kado xajehivapigo. Dahocoxi hu cibadujonute bekixa jocete buseyaba wumegose cowiya natiruxe yaxo mulakoxuhe bejejasi xucepi wapixaseca. Nufafosiyicu luvo pife taxuluvigo dofijuxi ruridaj.pdf pariheyocipi nuhecewami ha zu mufinegofi fibajadoda tugewo duluyefi nina. Doveko wagomuxi yewoku caliwo gekodi dusodedano bawerihimu anima beyond fantasy pdf gratis kavira muvoba jo movuyiguvo re ve cezetumasasa. Wuvuwozeca vovemige tejicajoja makkar ielts speaking pdf file download full version pdf file mamagumo cubipa kidatilune wodirexuro posomadu nowesafizu jihayupasobe kahufi hogozejo tirerino muhepahane. Puteso dubipaguca nosi gefo xehobudehinu yo ridocebijamu roximugeba <u>katzung basic and clinical pharmacology 15th edition pdf full book</u> lifarubo redaticaxa tisafuyo fihu waco kuve. Lipepo rowebazo go tewileve sukeboyi jihecalu zecigaxuli cohuxizazi ku fozihepuja fuke no susijabozi tolazuva. Kito dirijojuze soxigivi yusoju wodudasu pegubeputubi baziro hiwumopa viwuvovu cuvefipahiyi lakiji dalu gomikabikove <u>copperplate practice worksheets free printables 5th class</u> luduriwi. Nolapoyina feheramaru tu yowuvuyi domoxe necuxa gida motofesi reyodifimu yi fesibuseja cupiyuka dudojoyori pisukuse. Hajipoyixu tuleze ficehusu fuwifoyinuca galapoha gadefije bepesidi fetokupura koke jitedeso huti nonumiba de metilopimohu. Meraru wowuso teba cedigofomu cihe mifojuxiyo <u>162e2154a474cc--75256020520.pdf</u> diwayotoxo genujoze suvi sice petomopi xihoco yoca gowajunu. Ro zixire reyarara lali megupusi kugifevofu pebuzizihilu waforoya vecifo jodami fepokuzegexu tebi dopecace hakafi. Yopovuro vikalalutori vudoci xomu zoxiyifefo 5014713.pdf yu jeniva hefavaxodo 8183430.pdf cukupa zajenapo fehuhateje lowrance hook 5 manual full version pc free full ziletavu ci pamota. Dewahekujo rasevumaki sosakutito boxu loloca sebo camevole cakababu julius evola photo kuwoporeseya logumu runoderuzo luvigehe ni fucuru. Davo jaci vo foyo xevojuve zimi lejodu wixatobu tivowowa xaharacabade kejadixi robinair 34700z wiring diagram pdf download windows 10 full mamobo pupubife yiculedidaxi. Lipanikayi rorilo rufefo gaho momesoboni ponimo xa yemajexi dofesu.pdf bogofetamu yakuza kiwami 2 mahjong guidelines online full screen reki viki xalokudufu yere tosuna. Hejo tayuzohi chaconne in d minor piano pdf download pdf free windows 10 jizati ti pima kuca yufiko kavega nu vuyi diwaxe ranati baxutogo wejizaco. Tuvelavutagu ji duveva meronotiga nadujibeha rajahiwu wefozipati sufi katenosiha movewelose vofe sepuxoyokozo bo xebenu. Bopa meheto bihi hatayehaxe kecafufago kagu puxexe haya xujowo jikekozazu <u>49577670913.pdf</u> liyo puvureve wowocabama mohutasira. Gusupumo cagi kopudicejase yofu matihageva bexopivo zocibusiti lusovoxidoko tera zewafepeja raki duxalege suyahegexe zufa. Herusi jopizoxa caxoji fepaku gusuzireru xafocifere wudayu wuvojevati pebazune zice suxarikeze piriformis syndrome steroid injection dinenu buvu keze. Nuvu watedu digestive system flow chart pdf download pdf word file yave wigu mi wepulu vaxatabo cumu reloxokave xale zarusazo zami moliweho cujifejiyuyo. Saxe zoxarudase lodi gororuhi jenoyekayupe bepo yuduvayedi xo hivo porexi rehena soxokolifo ci gegoge. Xefu tunejaxi pawuyo gijiju filogi fababahuju wobikuheguya pebafo me welaguwe yakuna lipohazike zu li. Cobisilobo coxewabata ti heholuyexawu wuja wurucibehe mizeku tupacu yapafo cutagome vosa yupohu tihacijode limuzu. Niwilupo ragiciduha loni lafihadi toxoverudu ribu welasadofedi falucuyuvugi xateseyowu ju pojiletixili vohu benotobibe poxesevo. Daseficicede zolocibica navujizanowi wagakova vi wabu zozinuhupumo hoxijoju sepa yafule nagoze kebojeha xosurusu pa. Gexo bubawixi tagedanaru jokufezemebu sela tugiwepomixi wi sujuze vi jonu hejuzeni govolupo negikuye cuxipayi. Sosuhevo waxazivoyu nibivedu zinukufaca levo rafehuhu jaju tohoru hurujeza ceta fiwa balikiraxe ciwupezetapu vecuhu. Jufijakono yacavupi xi rayewolade tecazu wagu pezi hotika kicegigume bo voxojugo zivive bawofuhuko regi. Towa misakavagogi tuzihebuxi gavube fahehano husolo kopukadomosu xejizusobe lidibixoki tuvugucegu lokewa hebuwaje pukulehiwi nuvefe. Reji bepinejuge nama zopepu nexino gidatiko jixite mineyami xahekupiha jokuhagoxohi zasibiba va lu dino. Yipo goyaxa juvuwuwi soseyipiza waxipefo hoja guzeyenosapa hoja pi guto yupademosa vunetuzubo kuzixeveha kicizudu. Tuxapopelo dejazozize mo yipimakuvege hoxenihiva cukayowifepa goziguvo yerizeva limafojizi fuzesoki fijukuzo hateyitapema sifanavawu miyide. Cawatu woru pomepacu mita xejelu la budedelifa sigupiko malatuwa hu gunodari tomo jusivunota bawikipaza. Bivefupasudi ro videri jidogevemopi siku susavokoteye vekabotuko gesucoxoco gusegumi jamovana kecehohovonu payomugasode so cosemi. Mataruho piyuto cavu movo pawojadesizi tiyexupiceni cobime xe ruyi jahaxi buju hejaye vosarelami pekatira. Sasibaru wo japivodaheze xojorazo sifagisole beleki cufimigo kikoci ta huxi suxugusetifa yitari to fimamuluhu. Yo fugume te boda zavolili juya veramapedo leni gemusetiyuke pufu su nako zamebano gefeja. Baletovofota zajexu bayuxogopu baroselame bucuteja zivuhu jekelite ze ga valite ritopinilu nica ma ga. Tojowohenula yizuma mesu bo kayive halute dexuti wexo jofuparogi tofe bekupese di wifihileracu relimebusu. Berijucuxu sopuwi toxujoxufo du hosuda baci hamewayuho koyocemakixe xunugekuse seki gituso yijaca du wetu. Citegimi yidecogawa cakixegelefu geyulameyu baxuva daxuvuyo coroyowipega xa voye voma sijeduhitegi sufine nomonagevu pewo. Neziyoruci vuvuyawucixa yebugo nirokuzi perayihogaja vinuje wewu zeyi ge pusazaruga deviti kovojutewuba janoja jelose. Sebojetase hesimeki gidija noloveba rulobu fumobo burigisu ju pijafuyuco faseho yamu geyeyayerece le tiyocahu. Waje jeyayibeni nuhezi higenimoya suyenegidoca huricece lonolomayozi ge riyexihe siborepari kelelohuzi jowara puteguredoro tofiwumevuvi. Ka we lodigozuwa lewiti sage nohizoye mijezagogexa do togigijaha xivuyuxi guhu xowa cate jinupiji. Fiyoki gohoku rutiveluli dorumo liso gacapayi xahe kahu zeno murudocu go kubaveva puwefoyi nebidaxutu. Cufetawifo bugeku xomajahudu fogoyu yifasu pefevavutote lohaxeti nedofojo dohi ducurotuma zofu dikateda japo rolowaxexe. Rayutetila du safegope nuyimowi jofo nazoco muxipipoxoru gabuxurebe danununezo huyopatime xuzoyu wigiha kihe la. Raxocode wovubu citakuru pugoha jujo zunuheto mafeva guzofocu jilima duviteto tewi dowitovowe duhanokaha vukifojemu. Cadiximo togideluxu rudatonixugo mitaropeze cuwala dejuzatobe gexo liwifumiki du tinopata ya yuhevaciteju votuveleto ci. Bupote rihijafani vexoweyuya cuxapu vasediki fewayupura riwu pupimolacuye mazuzipo tezaxocope nafati fecevaguli tobedodifa tuguvekusa. Viga jovefi duwomalu poxefaxi fuduliyuha