
innocnv-mintmod Documentation

Release 0.0.2

Innocampus

Sep 19, 2018

1	What is innoConv (mintmod)?	1
2	Table of contents	3
2.1	Installation	3
2.1.1	Prerequisites	3
2.1.2	Dependencies	3
2.1.2.1	Python interpreter	3
2.1.2.2	Pandoc	3
2.1.2.3	Virtual environment	3
2.2	How to use innoconv-mintmod	4
2.2.1	Command line arguments	4
2.2.1.1	Positional Arguments	4
2.2.1.2	Named Arguments	4
2.3	Converting legacy mintmod content	5
2.3.1	Adjust commands	5
2.3.1.1	Remove <code>\ifttm... \else... \fi</code> commands	5
2.3.1.2	Unwanted LaTeX commands	6
2.3.1.3	Including other modules	6
2.3.2	Replace strings	6
2.3.3	Clean up code	7
2.4	Architecture	7
2.4.1	The command line interface	7
2.4.2	panzer	7
2.4.3	A conversion run in a nutshell	8
2.5	Module overview	8
2.5.1	innoconv_mintmod.constants	8
2.5.2	innoconv_mintmod.errors	9
2.5.3	innoconv_mintmod_mintmod_filter	9
2.5.3.1	innoconv_mintmod_mintmod_filter.commands	9
2.5.3.2	innoconv_mintmod_mintmod_filter.elements	14
2.5.3.3	innoconv_mintmod_mintmod_filter.environments	14
2.5.3.4	innoconv_mintmod_mintmod_filter.filter_action	15
2.5.3.5	innoconv_mintmod_mintmod_filter.math	16
2.5.4	innoconv_mintmod.runner	16
2.5.5	innoconv_mintmod.utils	16
3	Indices and tables	19

CHAPTER 1

What is innoConv (mintmod)?

This package converts mintmod-flavoured LaTeX into Markdown.

It can be seen as a shim for `mintmod.tex` and handles important mintmod commands by translating them to regular Pandoc elements.

Technically spoken it's essentially a wrapper to [Pandoc](#).

Read more about innoconv-mintmods [architecture](#).

2.1 Installation

2.1.1 Prerequisites

innocnv-mintmod is mainly used on Linux machines. It might work on Mac OS, Windows/Cygwin/WSL. You are invited to share experiences in doing so.

2.1.2 Dependencies

The only dependencies you have to provide yourself is Pandoc and the Python interpreter.

All others can be installed into a [Virtual environment](#).

2.1.2.1 Python interpreter

While other versions of Python might work fine, innocnv-mintmod was tested with **Python 3.6**. Make sure you have it installed.

2.1.2.2 Pandoc

You need to make sure to have a recent version of the pandoc binary available in `PATH` (**Pandoc 2.2.1** at the time of writing). There are [several ways on installing Pandoc](#).

2.1.2.3 Virtual environment

Setup and activate a virtual environment in a location of your choice.

```
$ python3 -m venv venv
$ source venv/bin/activate
```

Install innocnv-mintmod in your virtual environment using pip.

```
$ pip install --process-dependency-links -e git+https://gitlab.tubit.tu-berlin.de/
↳innodoc/innocnv-mintmod.git#egg=innocnv-mintmod
```

If everything went fine you should now have access to the innocnv-mintmod command.

```
$ innocnv-mintmod
usage: innocnv-mintmod [-h] [-o OUTPUT_DIR_BASE]
                        [-f {latex+raw_tex,markdown}]
                        [-t {html5,json,latex,markdown,asciidoc}] [-l {de,en}]
                        [-d] [-i] [-r] [-s]
                        source
innocnv-mintmod: error: the following arguments are required: source
```

Congratulations!

2.2 How to use innocnv-mintmod

You can run the converter in your content directory.

```
$ innocnv-mintmod .
```

This will trigger the conversion for this folder.

2.2.1 Command line arguments

```
usage: innocnv_mintmod [-h] [-o OUTPUT_DIR_BASE]
                        [-f {latex+raw_tex,markdown}]
                        [-t {markdown,latex,asciidoc,html5,json}] [-l {de,en}]
                        [-d] [-i] [-r] [-s]
                        source
```

2.2.1.1 Positional Arguments

source	content directory or file
---------------	---------------------------

2.2.1.2 Named Arguments

-o, --output-dir-base	output base directory Default: “./innocnv_output”
-f, --from	Possible choices: latex+raw_tex, markdown input format Default: “latex+raw_tex”
-t, --to	Possible choices: markdown, latex, asciidoc, html5, json output format Default: “markdown”

- l, --language-code** Possible choices: de, en
two-letter language code
Default: “de”
- d, --debug** debug mode (output HTML and highlight unknown commands)
Default: False
- i, --ignore-exercises** don’t show logs for unknown exercise commands/envs
Default: False
- r, --remove-exercises** remove all exercise commands/envs
Default: False
- s, --split-sections** split sections
Default: True

2.3 Converting legacy mintmod content

In this chapter some findings are documented on how to prepare content so it can be read by the *innocnv-mintmod* command.

Note: It’s not a complete list and there might be things missing that need to be done in your specific case.

First of all make sure all content is *UTF-8 encoded*. If not, tools like *iconv* can be helpful.

2.3.1 Adjust commands

There are some mintmod commands Pandoc is not able to parse. You need to manually replace them throughout your project.

2.3.1.1 Remove \ifttm...\else...\fi commands

`mintmod_ifttm` can get rid of all `\ifttm` commands.

Usage:

```
$ mintmod_ifttm < file_in.tex > file_out.tex
```

Automate on many files:

```
$ find . -name '*.tex' | xargs -I % sh -c 'mintmod_ifttm < % > %_changed && mv %_
↪changed %'
```

Warning: The script cares only about `\ifttm...\else...\fi` with an `\else` command. There may be occurrences of `\ifttm...\fi` (without `\else`). You need to remove them manually!

2.3.1.2 Unwanted LaTeX commands

A couple of commands are superfluous or doesn't make sense in a web-first content publishing platform like innoDoc. So remove any occurrences of the following commands.

- `\input{mintmod.tex}`
- `\input{english.tex}`
- `\begin{document} \begin{document}`
- `\MPpragma{MathSkip}`
- `\Mtikzexternalize`
- `\relax`
- `\- (hyphenation)`
- `\pagebreak`
- `\newpage`
- `\MPrintIndex`
- `\relax`

Automate:

```
find . -type f -name '*.tex' -or -name '*.rtex' | xargs perl -i -pe 's/\\input
↪{mintmod(.tex|)}\\w*\\n//igs'
```

2.3.1.3 Including other modules

Pandoc doesn't understand `\IncludeModule`. Change these statements to proper LaTeX commands.

`\IncludeModule{folder}{file.tex} → \input{folder/file.tex}.`

2.3.2 Replace strings

There are a couple of special characters you need to replace yourself.

- `\ "a → ä`
- `\ "o → ö`
- `\ "u → ü`
- `\ "A → Ä`
- `\ "O → Ö`
- `\ "U → Ü`
- `\ "s → ß`
- `\ "S → ß`
- `{\ss} → ß`
- `\ss `` → ``ß`
- `\ss\ → ß`
- `\ss{ } → ß`

- `\ss` → ß
- `"a` → ä
- `"o` → ö
- `"u` → ü
- `"A` → Ä
- `"O` → Ö
- `"U` → Ü
- `"`` → „
- ```` → ”
- `' '` → “
- `" '` → ”

Automate:

```
find . -type f -name '*.tex' -or -name '*.rtex' | xargs sed -i 's/\\\"a/ä/g'
```

2.3.3 Clean up code

Remove unused files from your project and keep track of your changes using a VCS.

2.4 Architecture

This section gives an overview of innoconv-mintmods architecture.

2.4.1 The command line interface

The entry point is the command line tool `innoconv-mintmod`.

It calls `panzer` which is a Pandoc wrapper tool with the correct parameters.

Most of the magic happens in the package `MintmodFilterAction`.

It is implemented as a [Pandoc filter](#) and provides functions to deal with a number of special LaTeX mintmod commands Pandoc would otherwise just ignore.

All special commands are translated into primitives Pandoc knows already. Additionally information is encoded in attributes that are attached to the resulting elements.

The result of the `MintmodFilterAction` is a regular Pandoc AST that can be further processed by Pandoc output modules, thus be translated to Markdown, LaTeX, HTML and so forth.

2.4.2 panzer

`panzer` is a small wrapper script around Pandoc. It enriches Pandoc with several useful features that just happened to match this projects needs.

First of all it is possible to define profiles (called *styles* in panzer) that can already define parameters on how to run Pandoc.

Furthermore it can manage applied filters, run pre- and postprocessors etc.

You can find its configuration in the sub-directory `.panzer`.

2.4.3 A conversion run in a nutshell

TDB

2.5 Module overview

2.5.1 `innocnv_mintmod.constants`

Project constants are defined here.

`innocnv_mintmod.constants.INDEX_LABEL_PREFIX`

Element class for index labels

`innocnv_mintmod.constants.COMMANDS_IRREGULAR`

Math commands with irregular arguments, key=command-name, value=formatstring or value=dict (number of arguments, formatstring)

`innocnv_mintmod.constants.REGEX_PATTERNS`

Regular expressions

`innocnv_mintmod.constants.ELEMENT_CLASSES`

Element classes

`innocnv_mintmod.constants.MINTMOD_SUBJECTS`

Subjects as used in mintmod command `\MSetSubject`

`innocnv_mintmod.constants.LANGUAGE_CODES`

Supported language codes

`innocnv_mintmod.constants.DEFAULT_LANGUAGE_CODE`

Default language code

`innocnv_mintmod.constants.DEFAULT_OUTPUT_DIR_BASE`

Default innocnv output directory

`innocnv_mintmod.constants.DEFAULT_OUTPUT_FORMAT`

Default innocnv output format

`innocnv_mintmod.constants.OUTPUT_FORMAT_EXT_MAP`

mapping between output formats and file extensions

`innocnv_mintmod.constants.OUTPUT_FORMAT_CHOICES`

Output format choices

`innocnv_mintmod.constants.ROOT_DIR`

project root dir

`innocnv_mintmod.constants.PANZER_SUPPORT_DIR`

panzer support directory

`innocnv_mintmod.constants.ENCODING`

encoding used in this project

2.5.2 innocnv_mintmod.errors

Exceptions are defined here.

exception `innocnv_mintmod.errors.ParseError`
 Raised when a mintmod command cannot be parsed.

2.5.3 innocnv_mintmod_mintmod.mintmod_filter

This module handles mintmod LaTeX commands.

Commands and environments are defined in the classes `Commands` and `Environments`.

2.5.3.1 innocnv_mintmod.mintmod_filter.commands

Handle mintmod LaTeX commands.

Note: Provide a `handle_CMDNAME` function for handling `CMDNAME` command. You need to [slugify](#) the command name.

Example: `handle_msection` method will receive the command `\MSection`.

class `innocnv_mintmod.mintmod_filter.commands.Commands`

Handlers for commands are defined here.

Given the command:

```
\MSection{Foo}
```

The handler method `handle_msection` receives the following arguments:

- `cmd_args`: `['Foo']`
- `elem`: `panflute.base.Element`

handle_bigskip (`cmd_args`, `elem`)
 Handle `\bigskip` command.

handle_clearpage (`cmd_args`, `elem`)
 Handle `\clearpage` command.

handle_glqq (`cmd_args`, `elem`)
 Handle `\glqq` command.

handle_grqq (`cmd_args`, `elem`)
 Handle `\grqq` command.

handle_highlight (`cmd_args`, `elem`)
 Handle `highlight` command.

This seems to be some sort of formatting command. There's no documentation and it does nothing in the mintmod code. We just keep the information here.

handle_hspace (`cmd_args`, `elem`)
 Handle `\hspace` and `\hspace*` command.

handle_jhtmlhinweiseingabefunktionen (`cmd_args`, `elem`)
 Handle `\jHTMLHinweisEingabeFunktionen` command.

handle_jhtmlhinweiseingabefunktionenexp (*cmd_args, elem*)

Handle \jHTMLHinweisEingabeFunktionenExp command.

handle_mblank (*cmd_args, elem*)

Handle \MBlank command.

handle_mcontenttable (*cmd_args, elem*)

Handle \MContentTable command.

handle_mcopyrightcollection (*cmd_args, elem*)

Handle \MCopyrightCollection command.

handle_mdeclaresiteuxid (*cmd_args, elem*)

Handle \MDeclareSiteUXID command.

This command is used to embed IDs. This is not relevant anymore and becomes a no-op.

handle_medskip (*cmd_args, elem*)

Handle \medskip command.

handle_mentry (*cmd_args, elem*)

Handle \MEntry command.

This command creates an entry for the glossary/index.

handle_mequationitem (*cmd_args, elem*)

Handle \MEquationItem command.

handle_mextlink (*cmd_args, elem*)

Handle \MExtLink command.

This command inserts an external link.

handle_mformelzoomhint (*cmd_args, elem*)

Handle \MFormelZoomHint command.

handle_mglobalchaptertag (*cmd_args, elem*)

Handle \MGlobalChapterTag command.

handle_mglobalconftag (*cmd_args, elem*)

Handle \MGlobalConfTag command.

handle_mglobaldatatag (*cmd_args, elem*)

Handle \MGlobalDataTag command.

handle_mglobalfavotag (*cmd_args, elem*)

Handle \MGlobalFavoTag command.

handle_mgloballocationtag (*cmd_args, elem*)

Handle \MGlobalLocationTag command.

handle_mgloballogintag (*cmd_args, elem*)

Handle \MGlobalLoginTag command.

handle_mgloballogouttag (*cmd_args, elem*)

handle \MGlobalLogoutTag command.

handle_mglobalsearchtag (*cmd_args, elem*)

Handle \MGlobalSearchTag command.

handle_mglobalstart (*cmd_args, elem*)

Handle \MGlobalStart command.

handle_mglobalstesttag (*cmd_args, elem*)

Handle \MGlobalSTestTag command.

handle_mgraphics (*cmd_args, elem, add_desc=True*)

Handle \MGraphics.

Embed an image with title.

Example: MGraphics{img.png}{scale=1}{title}

handle_mgraphicssolo (*cmd_args, elem*)

Handle \MGraphicsSolo.

Embed an image without title. Uses filename as image title.

handle_mgroupbutton (*cmd_args, elem*)

Handle \MGroupButton command

handle_mindex (*cmd_args, elem*)

Handle \MIndex command.

This command creates an invisible entry for the glossary/index.

handle_minputhint (*cmd_args, elem*)

Handle \MInputHint command.

handle_mlabel (*cmd_args, elem*)

Handle \MLabel command.

Will search for the previous header element and update its ID to the ID defined in the \MLabel command.

The command can occur in an environment that is parsed by a subprocess. In this case there's no last header element. The process can't set the ID because it can't access the doc tree. Instead it replaces the \MLabel by an element that is found by the parent process using function `innocnv.utils.extract_identifier()`.

handle_mlcheckbox (*cmd_args, elem*)

Handle exercises defined by \MLCheckbox command

handle_mlfunctionquestion (*cmd_args, elem*)

Handle exercises defined by \MLFunctionQuestion command

handle_mlintervalquestion (*cmd_args, elem*)

Handle exercises defined by \MLIntervalQuestion command

handle_mlparsedquestion (*cmd_args, elem*)

Handle exercises defined by \MLParsedQuestion command

handle_mlquestion (*cmd_args, elem*)

Handle exercises defined by \MLQuestion command

handle_mlsimplifyquestion (*cmd_args, elem*)

Handle exercises defined by \MLSimplifyQuestion command

handle_mlspecialquestion (*cmd_args, elem*)

Handle exercises defined by \MLSpecialquestion command

handle_mmodstartbox (*cmd_args, elem*)

Handle \MModStartBox command.

This command displays a table of content for the current chapter. This is handled elsewhere and becomes a no-op.

handle_mnref (*cmd_args, elem*)

Handle \MNRef command.

This command inserts a section link.

handle_modsemph (*cmd_args, elem*)

Handle modsemph command.

handle_modstextbf (*cmd_args, elem*)

Handle modstextbf command.

handle_mpragma (*cmd_args, elem*)

Handle \MPpragma command.

This command was used to embed build time flags for mintmod. It becomes a no-op.

handle_mprintindex (*cmd_args, elem*)

Handle \MPrintIndex command.

Index will be printed automatically. It becomes a no-op.

handle_mpullsite (*cmd_args, elem*)

Handle \MPullSite command.

handle_mref (*cmd_args, elem*)

Handle \MRef command.

This command translates to \vref.

handle_msection (*cmd_args, elem*)

Handle \MSection command.

handle_msetsectionid (*cmd_args, elem*)

Handle \MSetSectionID command.

This command is used solely for tikz images. It becomes a no-op.

handle_msetsubject (*cmd_args, elem*)

Handle \MSetSubject{ } command.

Command defines the category.

handle_msref (*cmd_args, elem*)

Handle \MSRef command.

This command inserts a fragment-style link.

handle_msubject (*cmd_args, elem*)

Handle \MSubject{title} command.

Command defines the document title.

handle_msubsection (*cmd_args, elem*)

Handle \MSubsection

handle_msubsubsection (*cmd_args, elem*)

Handle \MSubsubsection

handle_msubsubsectionx (*cmd_args, elem*)

Handle \MSubsubsectionx command. Which will generate a level 3 header.

handle_msubsubsubsectionx (*cmd_args, elem*)

Handle \MSubsubsubsectionx command. Which will generate a level 4 header.

handle_mtikzauto (*cmd_args, elem*)

Handle \MTikzAuto command.

Create a CodeBlock with TikZ code.

handle_mtitle (*cmd_args, elem*)

Handle \MTitle command.

This is an equivalent to \subsubsection

handle_mugraphics (*cmd_args, elem*)

Handle \MUGraphics.

Embed an image with title.

handle_mugraphicssolo (*cmd_args, elem*)

Handle \MUGraphicsSolo.

Embed an image without title.

handle_mvideo (*cmd_args, elem*)

Handle \MVideo.

Just return a Link Element.

handle_mwatermarksettings (*cmd_args, elem*)

Handle \MWatermarkSettings command.

handle_myoutubevideo (*cmd_args, elem*)

Handle \MYoutubeVideo.

Just return a Link Element.

handle_mzahl (*cmd_args, elem*)

Handle \MZahl command.

This is a math command but in fact occurs also in text.

handle_mzxyzhltrrennzeichen (*cmd_args, elem*)

Handle \MZXYZhltrrennzeichen command.

It is transformed to a \decmarker command and later substituted by MathJax. This is already in math substitutions but as it occurs outside of math environments it's defined here too.

handle_newline (*cmd_args, elem*)

Handle newline command.

handle_newpage (*cmd_args, elem*)

Handle \newpage command.

A display related command. It becomes a no-op.

handle_noindent (*cmd_args, elem*)

Handle \noindent command.

handle_quad (*cmd_args, elem*)

Handle \quad command.

handle_smallskip (*cmd_args, elem*)

Handle \smallskip command.

handle_special (*cmd_args, elem*)

Handle \special command.

This command is used to embed HTML in LaTeX source.

handle_vspace (*cmd_args, elem*)

Handle \vspace command.

A display related command. It becomes a no-op.

2.5.3.2 innocnv_mintmod.mintmod_filter.elements

Convenience functions and classes for creating common elements.

class innocnv_mintmod.mintmod_filter.elements.**Exercise**

Wrapper/Factory class that inherits from `pf.Element` and will return `pf.Code` instances, with special classes and attributes, depending on the given mintmod class.

innocnv_mintmod.mintmod_filter.elements.**create_content_box** (*elem_content*,
elem_classes)

Create a content box.

Convenience function for creating content boxes that only differ by having different content and classes.

innocnv_mintmod.mintmod_filter.elements.**create_header** (*title_str*, *doc*, *level=0*,
parse_text=False)

Create a header element.

Because headers need to be referenced by later elements, references to the last found header is remembered.

innocnv_mintmod.mintmod_filter.elements.**create_image** (*filename*, *descr*, *elem*,
add_descr=True,
block=True)

Create an image element.

innocnv_mintmod.mintmod_filter.elements.**parse_ex_args** (*cmd_args*, **names*)

receive a list of argument names and a list of values and return a pandoc conformant argument array containing element's arguments. In other words: take a list of arguments and make them named arguments for easier referencing.

2.5.3.3 innocnv_mintmod.mintmod_filter.environments

Handle mintmod LaTeX environments.

Note: Provide a `handle_ENVNAME` function for handling ENVNAME environment. You need to [slugify](#) the environment name.

Example: `handle_mxcontent` method will receive the `\begin{MXContent}...\end{MXContent}` environment.

class innocnv_mintmod.mintmod_filter.environments.**Environments**

Handlers for environments are defined here.

Given the environment:

```
\begin{MXContent}{Foo title long}{Foo title}{STD}
  Foo content
\end{MXContent}
```

The handler method `handle_mxcontent` receives the following arguments:

- `elem_content`: 'Foo content'
- `cmd_args`: ['Foo title long', 'Foo title', 'STD']
- `elem`: `panflute.elements.RawBlock`

handle_html (*elem_content*, *env_args*, *elem*)

Handle `\html` environment.

handle_itemize (*elem_content*, *env_args*, *elem*)
 Handle itemize environments, that were not correctly recognized by pandoc. This e.g. happens if there are \MExerciseItems environments contained in the items.

handle_mcontent (*elem_content*, *env_args*, *elem*)
 Handle \MContent environment.

handle_mcoshzusatz (*elem_content*, *env_args*, *elem*)
 Handle \MCOSHZusatz environment.

handle_mexample (*elem_content*, *env_args*, *elem*)
 Handle \MExample command.

handle_mexercise (*elem_content*, *env_args*, *elem*)
 Handle \MExercise environment.

handle_mexercisecollection (*elem_content*, *env_args*, *elem*)
 Handle \MExerciseCollection environment.

handle_mexerciseitems (*elem_content*, *env_args*, *elem*)
 Handle \MExerciseitems environments by returning an ordered list containing the \items defined in the environment. This is needed on top of handle_itemize as there are also mexerciseitems environments outside itemize environments.

handle_mexercises (*elem_content*, *env_args*, *elem*)
 Handle \MExercises environment.

handle_mexperiment (*elem_content*, *env_args*, *elem*)
 Handle \MExperiment environment.

handle_mhint (*elem_content*, *env_args*, *elem*)
 Handle \MHint command.

handle_minfo (*elem_content*, *env_args*, *elem*)
 Handle \MInfo environment.

handle_mintro (*elem_content*, *env_args*, *elem*)
 Handle \MIntro environment.

handle_mquestiongroup (*elem_content*, *env_args*, *elem*)
 Handle \MQuestionGroup environments. That are used to group together exercises, in order to allow synchronous validation. Especially used in checkbox grids. In mintmod, a button is also rendered below the contained exercises in the group. This function just returns a div with a class, in order to leave the validation logic, to client scripts

handle_msectionstart (*elem_content*, *env_args*, *elem*)
 Handle \MSectionStart environment.

handle_mtest (*elem_content*, *env_args*, *elem*)
 Handle \MTest environment.

handle_mxcontent (*elem_content*, *env_args*, *elem*)
 Handle \MXContent environment.

handle_mxinfo (*elem_content*, *env_args*, *elem*)
 Handle \MXInfo environment.

2.5.3.4 innocnv_mintmod.mintmod_filter.filter_action

Pandoc filter that transforms mintmod commands.

class `innocnv_mintmod.mintmod_filter.filter_action.MintmodFilterAction` (*debug=False*)

The Pandoc filter is defined in this class.

filter (*elem, doc*)

Receive document elements.

This method receives document elements from Pandoc and delegates handling of simple substitutions, mintmod commands and .ments.

Parameters

- **elem** (`panflute.base.Element`) – Element to handle
- **doc** (`panflute.elements.Doc`) – Document

2.5.3.5 innocnv_mintmod.mintmod_filter.math

Handle mintmod math commands.

`innocnv_mintmod.mintmod_filter.math.handle_math` (*elem*)

Handle mintmod text substitutions and some commands with irregular arguments.

2.5.4 innocnv_mintmod.runner

Runner module

class `innocnv_mintmod.runner.InnocnvRunner` (*source, output_dir_base, language_code, ignore_exercises=False, remove_exercises=False, split_sections=False, input_format='latex+raw_tex', output_format='markdown', debug=False*)

innocnv (mintmod) runner that spawns a panzer instance.

run ()

Setup paths and options and run the panzer command.

Return type `str`

Returns output filename

2.5.5 innocnv_mintmod.utils

Utility module

`innocnv_mintmod.utils.block_wrap` (*elem, orig_elem*)

Wraps an element in a block if necessary.

If the original element was block panflute expects the return value to be also block. In many places we need to detect this and wrap an inline.

Parameters

- **elem** (`panflute.base.Element`) – Element to be wrapped
- **orig_elem** (`panflute.base.Element`) – Original element

Return type `panflute.base.Element`

Returns *elem* or *elem* wrapped in `panflute.elements.Plain`

`innoconv_mintmod.utils.destringify(string)`

Takes a string and transforms it into list of Str and Space objects.

This function breaks down strings with whitespace. It could be done by calling `parse_fragment()` but doesn't have the overhead involved.

Example

```
>>> destringify('foo bar baz')
[Str(foo), Space, Str(bar), Space, Str(baz)]
```

Parameters `string` (*str*) – String to transform

Return type `list`

Returns list of `panflute.Str` and `panflute.Space`

`innoconv_mintmod.utils.extract_identifier(content)`

Extract identifier from content and remove annotation element.

`\MLabel` commands that occur within environments are parsed in a child process (`innoconv_mintmod.mintmod_filter.commands.handle_mlabel()`). The id attribute can't be set directly as they can't access the whole doc tree. As a workaround they create a fake element and add the identifier. This function extracts the identifier and removes the annotation element.

Parameters `content` (*list*) – List of elements

Return type (*list*, *str*)

Returns updated content list and identifier (might be None)

`innoconv_mintmod.utils.get_panzer_bin()`

Get path of panzer binary.

`innoconv_mintmod.utils.get_remembered_element(doc)`

Retrieve remembered element from the document and forget it.

To remember elements use `remember_element()`.

Parameters `doc` (`panflute.elements.Doc`) – Document where the element is stored

Return type `panflute.base.Element`

Returns The remembered element or *None*

`innoconv_mintmod.utils.log(msg_string, level='INFO')`

Log messages when running as a panzer filter.

Parameters

- `msg_string` (*str*) – Message that is logged
- `level` (*str*) – Log level (INFO, WARNING, ERROR OR CRITICAL)

`innoconv_mintmod.utils.parse_cmd(text)`

Parse a LaTeX command using regular expressions.

Parses a command like: `\foo{bar}{baz}`

Parameters `text` (*str*) – String to parse

Return type (*str*, *list*)

Returns command name and list of command arguments

`innocnv_mintmod.utils.parse_fragment` (*parse_string*, *as_doc=False*,
from_format='latex+raw_tex')

Parse a source fragment using panzer.

Parameters

- **parse_string** (*str*) – Source fragment
- **as_doc** (*bool*) – Return elements as `panflute.elements.Doc`
- **from_format** (*str*) – Source format

Return type list of `panflute.base.Element` or `panflute.elements.Doc`

Returns parsed elements

Raises

- **OSError** – if panzer executable is not found
- **RuntimeError** – if panzer recursion depth is exceeded
- **RuntimeError** – if panzer output could not be parsed

`innocnv_mintmod.utils.parse_nested_args` (*to_parse*)

Parse LaTeX command arguments that can have nested commands. Returns arguments and rest string.

Parses strings like: `{bar}{baz{}}rest` into `['bar', 'baz{}'], 'rest']`.

Parameters **to_parse** (*str*) – String to parse

Return type (*list*, *str*)

Returns parsed arguments and rest string

`innocnv_mintmod.utils.remember_element` (*doc*, *elem*)

Remember an element in the document for later.

To retrieve remembered elements use `get_remembered_element()`.

Parameters

- **doc** (`panflute.elements.Doc`) – Document where to store the memory
- **elem** (`panflute.base.Element`) – Element to remember

`innocnv_mintmod.utils.remove_empty_paragraphs` (*doc*)

Remove empty paragraphs from document.

Parameters **doc** (`panflute.elements.Doc`) – Document

`innocnv_mintmod.utils.to_inline` (*elem*, *classes=[]*, *attributes={}*)

Convert any given pandoc element to inline element(s). Some information may be lost.

CHAPTER 3

Indices and tables

- `genindex`
- `modindex`
- `search`

i

- `innocnv_mintmod`, [1](#)
- `innocnv_mintmod.constants`, [8](#)
- `innocnv_mintmod.errors`, [9](#)
- `innocnv_mintmod.mintmod_filter`, [9](#)
- `innocnv_mintmod.mintmod_filter.commands`,
[9](#)
- `innocnv_mintmod.mintmod_filter.elements`,
[14](#)
- `innocnv_mintmod.mintmod_filter.environments`,
[14](#)
- `innocnv_mintmod.mintmod_filter.filter_action`,
[15](#)
- `innocnv_mintmod.mintmod_filter.math`,
[16](#)
- `innocnv_mintmod.runner`, [16](#)
- `innocnv_mintmod.utils`, [16](#)

B

block_wrap() (in module innoconv_mintmod.utils), 16

C

Commands (class in innoconv_mintmod.mintmod_filter.commands), 9

COMMANDS_IRREGULAR (in module innoconv_mintmod.constants), 8

create_content_box() (in module innoconv_mintmod.mintmod_filter.elements), 14

create_header() (in module innoconv_mintmod.mintmod_filter.elements), 14

create_image() (in module innoconv_mintmod.mintmod_filter.elements), 14

D

DEFAULT_LANGUAGE_CODE (in module innoconv_mintmod.constants), 8

DEFAULT_OUTPUT_DIR_BASE (in module innoconv_mintmod.constants), 8

DEFAULT_OUTPUT_FORMAT (in module innoconv_mintmod.constants), 8

destringify() (in module innoconv_mintmod.utils), 16

E

ELEMENT_CLASSES (in module innoconv_mintmod.constants), 8

ENCODING (in module innoconv_mintmod.constants), 8

Environments (class in innoconv_mintmod.mintmod_filter.environments), 14

Exercise (class in innoconv_mintmod.mintmod_filter.elements), 14

extract_identifer() (in module innoconv_mintmod.utils), 17

F

filter() (innoconv_mintmod.mintmod_filter.filter_action.MintmodFilterAction method), 16

G

get_panzer_bin() (in module innoconv_mintmod.utils), 17

get_remembered_element() (in module innoconv_mintmod.utils), 17

H

handle_bigskip() (innoconv_mintmod.mintmod_filter.commands.Commands method), 9

handle_clearpage() (innoconv_mintmod.mintmod_filter.commands.Commands method), 9

handle_glqq() (innoconv_mintmod.mintmod_filter.commands.Commands method), 9

handle_grqq() (innoconv_mintmod.mintmod_filter.commands.Commands method), 9

handle_highlight() (innoconv_mintmod.mintmod_filter.commands.Commands method), 9

handle_hspace() (innoconv_mintmod.mintmod_filter.commands.Commands method), 9

handle_html() (innoconv_mintmod.mintmod_filter.environments.Environments method), 14

handle_itemize() (innoconv_mintmod.mintmod_filter.environments.Environments method), 14

handle_jhtmlhinweiseingabefunktionen() (innoconv_mintmod.mintmod_filter.commands.Commands method), 9

handle_jhtmlhinweiseingabefunktionenexp() (innoconv_mintmod.mintmod_filter.commands.Commands method), 9

handle_math()	(in module innoconv_mintmod.mintmod_filter.math), 16	conv_mintmod.mintmod_filter.commands.Commands method), 10
handle_mblank()	(innoconv_mintmod.mintmod_filter.commands.Commands method), 10	handle_mglobalconftag() (innoconv_mintmod.mintmod_filter.commands.Commands method), 10
handle_mcontent()	(innoconv_mintmod.mintmod_filter.environments.Environments method), 15	handle_mglobaldatatag() (innoconv_mintmod.mintmod_filter.commands.Commands method), 10
handle_mcontenttable()	(innoconv_mintmod.mintmod_filter.commands.Commands method), 10	handle_mglobalfavotag() (innoconv_mintmod.mintmod_filter.commands.Commands method), 10
handle_mcopyrightcollection()	(innoconv_mintmod.mintmod_filter.commands.Commands method), 10	handle_mgloballocationtag() (innoconv_mintmod.mintmod_filter.commands.Commands method), 10
handle_mcoshzusatz()	(innoconv_mintmod.mintmod_filter.environments.Environments method), 15	handle_mgloballogintag() (innoconv_mintmod.mintmod_filter.commands.Commands method), 10
handle_mdeclaresiteuxid()	(innoconv_mintmod.mintmod_filter.commands.Commands method), 10	handle_mgloballogouttag() (innoconv_mintmod.mintmod_filter.commands.Commands method), 10
handle_medskip()	(innoconv_mintmod.mintmod_filter.commands.Commands method), 10	handle_mglobalsearchtag() (innoconv_mintmod.mintmod_filter.commands.Commands method), 10
handle_mentry()	(innoconv_mintmod.mintmod_filter.commands.Commands method), 10	handle_mglobalstart() (innoconv_mintmod.mintmod_filter.commands.Commands method), 10
handle_mequationitem()	(innoconv_mintmod.mintmod_filter.commands.Commands method), 10	handle_mglobalstesttag() (innoconv_mintmod.mintmod_filter.commands.Commands method), 10
handle_mexample()	(innoconv_mintmod.mintmod_filter.environments.Environments method), 15	handle_mgraphics() (innoconv_mintmod.mintmod_filter.commands.Commands method), 10
handle_mexercise()	(innoconv_mintmod.mintmod_filter.environments.Environments method), 15	handle_mgraphicssolo() (innoconv_mintmod.mintmod_filter.commands.Commands method), 11
handle_mexercisecollection()	(innoconv_mintmod.mintmod_filter.environments.Environments method), 15	handle_mgroupbutton() (innoconv_mintmod.mintmod_filter.commands.Commands method), 11
handle_mexerciseitems()	(innoconv_mintmod.mintmod_filter.environments.Environments method), 15	handle_mhint() (innoconv_mintmod.mintmod_filter.environments.Environments method), 15
handle_mexercises()	(innoconv_mintmod.mintmod_filter.environments.Environments method), 15	handle_mindex() (innoconv_mintmod.mintmod_filter.commands.Commands method), 11
handle_mexperiment()	(innoconv_mintmod.mintmod_filter.environments.Environments method), 15	handle_minfo() (innoconv_mintmod.mintmod_filter.environments.Environments method), 15
handle_mextlink()	(innoconv_mintmod.mintmod_filter.commands.Commands method), 10	handle_minputhint() (innoconv_mintmod.mintmod_filter.commands.Commands method), 11
handle_mformelzoomhint()	(innoconv_mintmod.mintmod_filter.commands.Commands method), 10	handle_mintro() (innoconv_mintmod.mintmod_filter.environments.Environments method), 15
handle_mglobalchaptertag()	(innoconv_mintmod.mintmod_filter.commands.Commands method), 10	handle_mlabel() (innoconv_mintmod.mintmod_filter.commands.Commands method), 11
		handle_mlcheckbox() (innoconv_mintmod.mintmod_filter.commands.Commands method), 11

method), 11	conv_mintmod.mintmod_filter.commands.Commands
handle_mlfunctionquestion()	(inno- handle_mlmethod), 12
conv_mintmod.mintmod_filter.commands.Commands	handle_msref() (innoconv_mintmod.mintmod_filter.commands.Commands
method), 11	method), 12
handle_mlintervalquestion()	(inno- handle_msubject() (inno-
conv_mintmod.mintmod_filter.commands.Commands	conv_mintmod.mintmod_filter.commands.Commands
method), 11	method), 12
handle_mlparsedquestion()	(inno- handle_msubsection() (inno-
conv_mintmod.mintmod_filter.commands.Commands	conv_mintmod.mintmod_filter.commands.Commands
method), 11	method), 12
handle_mlquestion()	(inno- handle_msubsubsection() (inno-
conv_mintmod.mintmod_filter.commands.Commands	conv_mintmod.mintmod_filter.commands.Commands
method), 11	method), 12
handle_mlsimplifyquestion()	(inno- handle_msubsubsectionx() (inno-
conv_mintmod.mintmod_filter.commands.Commands	conv_mintmod.mintmod_filter.commands.Commands
method), 11	method), 12
handle_mlspecialquestion()	(inno- handle_msubsubsubsectionx() (inno-
conv_mintmod.mintmod_filter.commands.Commands	conv_mintmod.mintmod_filter.commands.Commands
method), 11	method), 12
handle_mmodstartbox()	(inno- handle_mttest() (innoconv_mintmod.mintmod_filter.environments.Environment
conv_mintmod.mintmod_filter.commands.Commands	method), 15
method), 11	handle_mtikzauto() (inno-
handle_mnref() (innoconv_mintmod.mintmod_filter.commands.Commands	conv_mintmod.mintmod_filter.commands.Commands
method), 11	method), 12
handle_modsemph()	(inno- handle_mttitle() (innoconv_mintmod.mintmod_filter.commands.Commands
conv_mintmod.mintmod_filter.commands.Commands	method), 12
method), 11	handle_mugraphics() (inno-
handle_modtextbf()	(inno- conv_mintmod.mintmod_filter.commands.Commands
conv_mintmod.mintmod_filter.commands.Commands	method), 13
method), 12	handle_mugraphicssolo() (inno-
handle_mpragma()	(inno- conv_mintmod.mintmod_filter.commands.Commands
conv_mintmod.mintmod_filter.commands.Commands	method), 13
method), 12	handle_mvvideo() (inno-
handle_mprintindex()	(inno- conv_mintmod.mintmod_filter.commands.Commands
conv_mintmod.mintmod_filter.commands.Commands	method), 13
method), 12	handle_mwatermarksettings() (inno-
handle_mpullsite()	(inno- conv_mintmod.mintmod_filter.commands.Commands
conv_mintmod.mintmod_filter.commands.Commands	method), 13
method), 12	handle_mxcontent() (inno-
handle_mquestiongroup()	(inno- conv_mintmod.mintmod_filter.environments.Environments
conv_mintmod.mintmod_filter.environments.Environments	method), 15
method), 15	handle_mxinfo() (inno-
handle_mref() (innoconv_mintmod.mintmod_filter.commands.Commands	conv_mintmod.mintmod_filter.environments.Environments
method), 12	method), 15
handle_msection()	(inno- handle_myoutubevideo() (inno-
conv_mintmod.mintmod_filter.commands.Commands	conv_mintmod.mintmod_filter.commands.Commands
method), 12	method), 13
handle_msectionstart()	(inno- handle_mzahl() (innoconv_mintmod.mintmod_filter.commands.Commands
conv_mintmod.mintmod_filter.environments.Environments	method), 13
method), 15	handle_mzxyzhltrennzeichen() (inno-
handle_msetsectionid()	(inno- conv_mintmod.mintmod_filter.commands.Commands
conv_mintmod.mintmod_filter.commands.Commands	method), 13
method), 12	handle_newline() (inno-
handle_msetsubject()	(inno- conv_mintmod.mintmod_filter.commands.Commands

method), 13

handle_newpage() (innoconv_mintmod.mintmod_filter.commands.Commands method), 13

handle_noindent() (innoconv_mintmod.mintmod_filter.commands.Commands method), 13

handle_quad() (innoconv_mintmod.mintmod_filter.commands.Commands method), 13

handle_smallskip() (innoconv_mintmod.mintmod_filter.commands.Commands method), 13

handle_special() (innoconv_mintmod.mintmod_filter.commands.Commands method), 13

handle_vspace() (innoconv_mintmod.mintmod_filter.commands.Commands method), 13

OUTPUT_FORMAT_EXT_MAP (in module innoconv_mintmod.constants), 8

P

PANZER_SUPPORT_DIR (in module innoconv_mintmod.constants), 8

parse_cmd() (in module innoconv_mintmod.utils), 17

parse_cmds() (in module innoconv_mintmod.mintmod_filter.elements), 14

parse_fragment() (in module innoconv_mintmod.utils), 17

parse_nested_args() (in module innoconv_mintmod.utils), 18

ParseError, 9

R

I

INDEX_LABEL_PREFIX (in module innoconv_mintmod.constants), 8

innocnv_mintmod (module), 1

innocnv_mintmod.constants (module), 8

innocnv_mintmod.errors (module), 9

innocnv_mintmod.mintmod_filter (module), 9

innocnv_mintmod.mintmod_filter.commands (module), 9

innocnv_mintmod.mintmod_filter.elements (module), 14

innocnv_mintmod.mintmod_filter.environments (module), 14

innocnv_mintmod.mintmod_filter.filter_action (module), 15

innocnv_mintmod.mintmod_filter.math (module), 16

innocnv_mintmod.runner (module), 16

innocnv_mintmod.utils (module), 16

InnocnvRunner (class in innocnv_mintmod.runner), 16

REGEX_PATTERNS (in module innoconv_mintmod.constants), 8

remember_element() (in module innoconv_mintmod.utils), 18

remove_empty_paragraphs() (in module innoconv_mintmod.utils), 18

ROOT_DIR (in module innoconv_mintmod.constants), 8

run() (innocnv_mintmod.runner.InnocnvRunner method), 16

T

to_inline() (in module innoconv_mintmod.utils), 18

L

LANGUAGE_CODES (in module innoconv_mintmod.constants), 8

log() (in module innoconv_mintmod.utils), 17

M

MINTMOD_SUBJECTS (in module innoconv_mintmod.constants), 8

MintmodFilterAction (class in innoconv_mintmod.mintmod_filter.filter_action), 15

O

OUTPUT_FORMAT_CHOICES (in module innoconv_mintmod.constants), 8