# MEI2Volpiano

*Release 0.1.0*

**DDMAL**

**May 14, 2021**

# CONTENTS:

CHAPTER

# ONE

# CLASSES

Converts MEI files to volpiano strings.

Takes in one or more MEI files and outputs their volpiano representation. See README for flags and usage.

class `mei2volpiano.mei2volpiano.MEItoVolpiano`

Class: MEItoVolpiano

**[Main]**

get_mei_elements(file) -> list[MEI elements] sylb_volpiano_map(list[elements]) -> dict{string: string} get_syl_key(element, integer) -> string get_volpiano(char, char) -> char export_volpiano(dict{syllables: notes}) -> string convert_mei_volpiano(file) -> string

^ convert_mei_volpiano handles all methods in main.

**[Debugging]**

find_clefs(list[elements]) -> list[char] find_notes(list[elements]) -> list[char] find_syls(list[elements]) -> list[string] sylb_note_map(list[elements]) -> dict{string: string}

^ useful for MEI parsing and testing outputs.

**convert_mei_volpiano**(*filename*)

All-in-one method for converting MEI file to valid volpiano string.

**Parameters** **filename** (*file*) – Open MEI file you want the volpiano of.

**Returns** Valid volpiano string representation of the input.

**Return type** volpiano (string)

**export_volpiano**(*mapping_dictionary*)

Creates volpiano string with clef attached.

**Parameters**

- **mapping_dictionary** (*dict*) – Dictionary of syllables and their
- **volpiano notes.** (*corresponding*) –

**Returns** Final, valid volpiano with the clef attached in a single line.

**Return type** (string)

**find_clefs**(*elements: list*) → list

Finds all clefs in a given elements list

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

**Returns** char list of all clefs found, in order.

**Return type** clefs (list)

`find_notes`(*elements*)

Finds all notes in a given elements list

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

**Returns** char list of all notes found, in order.

**Return type** notes (list)

`find_syls`(*elements*)

Finds all syllables in a given elements list

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

**Returns** string list of all syllables found, in order.

**Return type** syls (list)

`get_mei_elements`(*filename: str*) → list

Returns a list of all elements in the MEI file.

**Parameters** `filename` (*string*) – An open MEI file.

**Returns** List of all elements found.

**Return type** elements (list)

`get_syl_key`(*element*, *bias*)

Finds the dictionary key of a syllable from their 'syl' and database identifier.

**Parameters**

- `element` (*element*) – A single element representing a syllable (syl)
- `bias` (*int*) – The database identifier.

**Returns** The dictionary key for the given syllable.

**Return type** key (string)

`get_volpiano`(*note*, *ocv*)

Finds the volpiano representation of a note given its value and octave.

**Parameters**

- `note` (*char*) – Note value taken from an element ('c', 'd', 'e' etc.)
- `ocv` (*char*) – Octave of a given note ('1', '2', '3', or '4')

**Returns**

Volpiano character corresponding to input note and octave

or

error (string): Error if octave is out of range or note not in octave.

**Return type** oct{x}[note] (char)

`sylb_note_map`(*elements*)

Creates a dictionary map of syllables and their notes (with octaves).

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

**Returns** Dictionary {identifier: notes} of syllables and their unique data base numbers as keys and notes (with octaves) as values.

**Return type** syl_dict (dict)

`sylb_volpiano_map`(*elements*)

Creates a dictionary of syllables and their volpiano values.

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

**Returns** Dictionary {identifier: volpiano notes} of syllables and their unique data base numbers as keys and volpiano notes with correct octaves as values.

**Return type** syl_note (dict)

## CHAPTER TWO

# COMMANDS

CLI program implementation of the MEI2Volpiano library

See README for details.

`mei2volpiano.driver.main()`

This is the command line application MEI2Volpiano

usage: mei2vol [-h] (-mei MEI [MEI . . . ] | -txt [TXT]) [-export] mei2vol: error: one of the arguments -mei -txt is required

# CHAPTER THREE

## INDICES AND TABLES

- genindex
- modindex
- search

# PYTHON MODULE INDEX

# INDEX