MEI2Volpiano

Release 0.1.0

DDMAL

May 20, 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[str, str] get_syl_key(element, integer) -> str get_volpiano(str, str) -> str export_volpiano(dict[str, str]) -> str convert_mei_volpiano(file) -> str

^ convert_mei_volpiano handles all methods in main.

[Western]

Wsylb_volpiano_map(list[elements]) -> dict[str, str] Wconvert_mei_volpiano(file) -> str

^ Wconvert_mei_volpiano calls methods in Main to give the volpiano string for MEI files written in Western notation.

[Debugging]

find_clefs(list[elements]) -> list[str] find_notes(list[elements]) -> list[str] find_syls(list[elements]) -> list[str] sylb_note_map(list[elements]) -> dict[str, str]

^ useful for MEI parsing and testing outputs.

Wconvert_mei_volpiano(*filename: str*) → str

All-in-one method for converting MEI in Western notation to volpiano.

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

Returns Valid volpiano string representation of the input.

Return type volpiano (str)

Wsylb_volpiano_map(*elements: list*) → dict

Western notation - 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)

convert_mei_volpiano(*filename: str*) → str

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 (str)

`export_volpiano`(*mapping_dictionary: dict*) → str

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 (str)

`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: list*) → list

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: list*) → list

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` (*str*) – An open MEI file.

Returns List of all elements found.

Return type elements (list)

`get_syl_key`(*element: object*, *bias: int*) → str

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 (str)

`get_volpiano`(*note: str*, *ocv: str*) → str

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

Parameters

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

Returns

Volpiano character corresponding to input note and octave

or

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

Return type oct{x}[note] (str)

`sylb_note_map`(*elements: list*) → dict

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: list*) → dict

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: driver.py [-h] (-N | -W) [–export] [{txt,mei}] mei [mei . . .]

positional arguments: {txt,mei} Choice indicating whether the inputs will be mei or txt files mei One or multiple MEI, or text file(s) with each relative MEI file/path to be converted per line

optional arguments: -h, –help show this help message and exit -N An MEI neume encoded music file representing neume notation -W An MEI western encoded music file representing western notation –export flag indicating output to be sent to a .txt file (name corresponding with input mei)

CHAPTER THREE

INDICES AND TABLES

- genindex
- modindex
- search

PYTHON MODULE INDEX

INDEX