fe3h-checklist

Release 0.1.1

Hermann Krumrey

Sep 09, 2019

CONTENTS

Contents:

CHAPTER

ONE

FE3H_CHECKLIST PACKAGE

1.1 Submodules

1.2 fe3h_checklist.calc module

fe3h_checklist.calc.**calculate_max_support_rank**(*char_one: str, char_two: str, support_levels_info: Dict[str, Dict[str, str]]*) → str

Calculates the maximal support level between two characters :param char_one: The first character :param char_two: The second character :param support_levels_info: The support levels info :return: The maximal rank

fe3h_checklist.calc.**calculate_remaining_support_levels**(*character: str, support_levels: Dict[str, Dict[str, str]], support_levels_info: Dict[str, Dict[str, str]]*) → Tuple[int, Dict[str, int]]

Calculates how many support levels are left for a character. This calculates both the total amount of support levels left for the character as well as the remaining support levels with individual characters. :param character: The character for which to calculate the remaining

> support levels
>
> **Parameters**
>
> - **support_levels** – The current support levels
> - **support_levels_info** – The support levels info
>
> **Returns** A tuple consisting of the total remaining support levels and a dictionary mapping individual support levels to characters

1.3 fe3h_checklist.formatting module

fe3h_checklist.formatting.**color_character_names**(*string: str*) → str

Colors character names in their respective house colors :param string: The string to format :return: The formatted string

1.4 fe3h_checklist.ods_interface module

`fe3h_checklist.ods_interface.`**`load_support_levels`**`()` → Dict[str, Dict[str, str]]

Loads the information about support levels from the ODS file and puts the data into a dictionary mapping the character's current support levels to each other :return: The dictionary mapping the support levels to characters.

Example: {"Byleth M": {"Edelgard": "A"}}

`fe3h_checklist.ods_interface.`**`load_support_levels_info`**`()` → Dict[str, Dict[str, str]]

Loads the information about support level info from the ODS file and puts the data into a dictionary mapping the character's support levels to each other :return: The dictionary mapping the support levels to characters.

Example: {"Byleth M": {"Edelgard": "CBAS"}}

`fe3h_checklist.ods_interface.`**`load_table`**`(`*ods_file: str*, *sheet_name: str*`)` → Dict[str, Dict[str, str]]

Loads a table from an ODS file. Assumptions about the ODS file:

- Character names at the top and left side of the table

Parameters

- **`ods_file`** – The ODS file from which to load the data
- **`sheet_name`** – The sheet name from which to load the data

Returns A dictionary mapping the characters to each other and containing the value mapped to both of them. Example: {"Byleth M": {"Edelgard": "CBAS"}}

`fe3h_checklist.ods_interface.`**`save_support_levels`**`(`*support_levels: Dict[str, Dict[str, str]]*`)`

Saves the support levels to file :param support_levels: The support levels to save :return: None

`fe3h_checklist.ods_interface.`**`write_table`**`(`*ods_file: str*, *sheet_name: str*, *data: Dict[str, Dict[str, str]]*, *plus_notation: bool = False*`)`

Writes the content of a dictionary mapping characters to each other to an ODS file. :param ods_file: The ODS file to write to :param sheet_name: The sheet name to write to :param data: The data to write :param plus_notation: Whether or not to use +-notation.

True: 1 -> C+ False: C1 -> CC

Returns None

1.5 fe3h_checklist.printer module

`fe3h_checklist.printer.`**`print_remaining_support_levels`**`(`*no_byleth: bool*, *multi_line: bool*`)`

Prints the remaining support levels for all characters and orders them in a descending order. :param no_byleth: Whether or not to include support levels with Byleth :param multi_line: Whether or not to use multiple lines per character :return: None

1.6 fe3h_checklist.setters module

`fe3h_checklist.setters.`**`set_support_level`**`(`*char_one: str*, *char_two: str*, *level: str*`)`

Sets the support level for two characters and saves it. :param char_one: The first character :param char_two: The second character :param level: The support level :return: None

1.7 fe3h_checklist.support_encoding module

fe3h_checklist.support_encoding.**decode_support_levels**(*encoded: str*, *for_sheet: bool = False*)

Decodes the support levels back into a human-readable format :param encoded: The encoded support level :param for_sheet: Changes the support levels into the format used in the

> data sheet (e.g. C1 -> CC)
>
> **Returns** The decoded support level

fe3h_checklist.support_encoding.**encode_support_levels**(*support_levels: str*) → str

Converts a support level string into something that can easily be indexed. The existance of a '+' rank like 'C+' is stored in the info sheet as "CC" and therefore uses up two spaces, making indexing hard. So we convert these '+' support levels like this:

> C+ -> 1 B+ -> 2 A+ -> 3
>
> **Parameters** **support_levels** – The support level string to encode
>
> **Returns** The encoded support level string

1.8 Module contents

fe3h_checklist.**local_dir = '/root/.config/fe3h-checklist'**

Directory containing local config files

fe3h_checklist.**support_levels_file = '/root/.config/fe3h-checklist/supportlevels.ods'**

File containing the support level info file

fe3h_checklist.**support_levels_info_file = '/root/.config/fe3h-checklist/supportinfo.ods'**

File containing the support level info file

CHAPTER

TWO

FE3H_CHECKLIST

CHAPTER

THREE

INDICES AND TABLES

- genindex
- modindex
- search

PYTHON MODULE INDEX

INDEX