MDMERGE

I developed `mdmerge` for my own usage. I was tired to update my main `README.md` file when it contains some codes (all or part of) from an existing file.

When the source file change, your `README.md` must be manually updated.

So I create my own **Node.js** application to update my `README.md` which using other files thanks to a custom instruction which is invisible in the rendered document (also in PDF document).

Currently, the application is not fully finished but the main feature are finished.

Getting Start

The Command Line Interface

The simpliest command line to merge your **Markdown** is `mdmerge <FILE>.md`. By default, the command line creates a new file nammed `<FILE>.merged.md`. I choose to do this to prevent breaks on the source file.

The command line have two **implicits** options :

- The first one will stands for the **input file** to merge.
- The second one will stands for the **output file** to put the result.

Examples with **implicits** options :

1. `mdmerge <INPUT_FILE>.md <OUTPUT_FILE>.md` : The merged document from `<INPUT_FILE>.md` will be place in the output file `<OUTPUT_FILE>.md`
2. `mdmerge <INPUT_FILE>.md <INPUT_FILE>.md` : Done like this, you will overwrite your source file will the result.

The second case `2.` can be simply with the following command : `mdmerge <FILE>.md -w`

Implicit Options

This application allows **implicits** options. **Implicits** options are the values which not begins with a hyphen char (`-`). They are determined with their position behind the **command** (`mdmerge`).

On the following example `mdmerge -c README.md README.final.md`, the first **implicit** option is `README.md` and the second is `README.final.md`.

Implicits options priority is lesser than **Explicits** ones.

Implicit Option `1`

The first **implicit** option stands for the file to merge. It's is equal to the option `-i`, `--in`.

Implicit Option `2`

The second **implicit** option stands for the output file. It's is equal to the option `-o`, `--out`.

Command Line Options

Please find below the detail of each options for the command `mdmerge`.

Be warned : **Explicit** options have the priority on the **implicit** ones.

Be warned : The **long options** have the **priority** on the **short ones**. For instance, if you use at the same time options `-i=file_1` and `--in=file_2`, this is the file `file_2` which will be processed.

Option `-i`, `--in` : Input file

The option `-i` (`--in`) allows you to specified the input file to merged.

- It requires a value.
- The input must be a **file** which exists.

- It equal to the implicit options `1`.

Option `-o`, `--out` : Output file

The option `-o` (`--out`) allows you to specified the target file in which the result will be placed.

- It requires a value.
- The output must be a **file**. If the file not exists, it will be created.
- It equal to the implicit options `2`.

This option will have no effect if option `-w` (`--write`) is used.

Option `-w`, `--write` : Overwrite

The option `-w` (`--write`) allows you to directly **overwrite** the input file with the result.

- It does not requires a value.

Option `-c`, `--clear` : Clear

The option `-c` (`--clear`) allows you to remove merged content between tags.

- It does not requires a value.

The Instruction

The detailed instruction hereafter must be placed at the begin of the line in a comment block in your Markdown file : `[](<instruction>)`

Once parsed by `mdmerge`, a closing instruction has been inserted. That will allows the application to update the included parts.

Please find below, the generic form of the instruction `<instruction>` :

The element separator is the colon char (`:`)

- `<instruction>` = `@import><pathToFile>[..:<inclusionInstruction>][:[<codeBlockLanguage>]]`
    - `<pathToFile>` : This is the relative path to the file used for this inclusion.
    - `:<inclusionInstruction>` : You can insert none to many of the inclusion instruction separated by `:`.
- `<inclusionInstruction` = `<line>[,<startOffset>][,<endOffset>]`
    - `<line>` : You can specified the line to toggle the inclusion.
    - `<startOffset>` : You can specified from which char you want to begin the inclusion on the line.
    - `<endOffset>` : You can specified when you stop the inclusion on the line
- `:<codeBlockLanguage>` : You can ask to `mdmerge` to insert your content in a Markdown code block in a specified language. You can omit `<codeBlockLanguage>` to let your favorite render to identify automatically the language (Example with **Highligh.js**)

The `<inclusionInstruction>` toggle the inclusion.

The generic rule is the **odds** elements will start the inclusion whereas the **evens** will stop the inclusion. When the total number of `<inclusionInstruction>` is **odd**, the last inclusion process will inclusion until the **EOF** is met.

Please find an example of instruction :

```
[](@import>bin/mdmerge.js:53:79:js)
```

The instruction can be read as "Please, include here the JavaScript code from file `mdmerge.js`, from line `53` to line `79` in a code block `js` ".

The result is the following

```
/**
 * Afficher un message dans le stream.
 *
 * @param message Message à afficher
 * @param level   Niveau de message. 0=OK,1=KO,2=WARN
 */
function log(message, level = 0, args = []){
    // 0 = Success
    // 1 = Error
    // 2 = Warn
    let levels = [
        {color: "Green", name: "SUCCESS", return: 0},
        {color: "Red", name: "ERROR", return: 1},
        {color: "Yellow", name: "WARNING", return: 0},
    ];

    console.log(
        "[ " +
        colors.fg[levels[level].color] +
        levels[level].name +
        colors.Reset +
        " ] : " +
        message
    );

    return levels[level].return;
}
```

Usage Cases

Please find below an exhaustiv list of usage cases and their development status

Instruction	Explanation	Dev. Status
@import>test/incl.md	Include the whole content of file `incl.md` in folder `test`	DONE
@import>test/incl.md:	Include the whole content of file `incl.md` in folder `test` in a code block	DONE
@import>test/incl.md:js	Include the whole content of file `incl.md` in folder `test` in a code block JavaScript `js`	DONE
@import>test/incl.md:53	Include the content of file `incl.md` from line `53` to the end of file `EOF`	DONE
@import>test/incl.md:53:	Include the content of file `incl.md` from line `53` to the end of file `EOF` in a code block	DONE
@import>test/incl.md:53:js	Include the content of file `incl.md` from line `53` to the end of file `EOF` in a code block JavaScript `js`	DONE
@import>test/incl.md::79	Include the content of file `incl.md` from the beginning to the line `79`	DONE
@import>test/incl.md::79:	Include the content of file `incl.md` from the beginning to the line `79` in a code block	DONE
@import>test/incl.md::79:js	Include the content of file `incl.md` from the beginning to the line `79` in a code block JavaScript `js`	DONE
@import>test/incl.md:53:79	Include the content of file `incl.md` from line `53` to the line `79`	DONE
@import>test/incl.md:53:79:	Include the content of file `incl.md` from line `53` to the line `79` in a code block	DONE
@import>test/incl.md:53:79:js	Include the content of file `incl.md` from line `53` to the line `79` in a code block JavaScript `js`	DONE
@import>test/incl.md:53,1	Include the content of file `incl.md` from line `53` , starting at position `1` to the end of file `EOF`	DONE
@import>test/incl.md:53,1:	Include the content of file `incl.md` from line `53` , starting at position `1` to the end of file `EOF` in a code block	DONE
@import>test/incl.md:53,1:js	Include the content of file `incl.md` from line `53` , starting at position `1` to the end of file `EOF` in a code block Javascript `js`	DONE
@import>test/incl.md:53,,1	Include the line number `53` from the file `incl.md` , starting at position `0` to position `1`	DONE
@import>test/incl.md:53,,1:	Include the line number `53` from the file `incl.md` , starting at position `0` to position `1` in a code block	DONE
@import>test/incl.md:53,,1:js	Include the line number `53` from the file `incl.md` , starting at position `0` to position `1` in a code block Javascript `js`	DONE

@import>test/incl.md:53,2,4	Include the line number `53` from the file `incl.md`, starting at position `02` to position `4`	DONE
@import>test/incl.md:53,2,4:	Include the line number `53` from the file `incl.md`, starting at position `02` to position `4` in a code block	DONE
@import>test/incl.md:53,2,4:js	Include the line number `53` from the file `incl.md`, starting at position `02` to position `4` in a code block Javascript `js`	DONE
@import>test/incl.md:53:79:526	Include the content of file `incl.md` from line `53` to the line `79`, and from line `526` to the end of line `EOF`	DONE
@import>test/incl.md:53:79:526:	Include the content of file `incl.md` from line `53` to the line `79`, and from line `526` to the end of line `EOF` in a code block	DONE
@import>test/incl.md:53:79:526:js	Include the content of file `incl.md` from line `53` to the line `79`, and from line `526` to the end of line `EOF` in a code block JavaScript `js`	DONE

Their is some cases not listed here which are functional and some other ideas which are not implemented yet