Markdown PDF

This extension converts Markdown files to pdf, html, png or jpeg files.

Japanese README

Table of Contents

Features

Supports the following features

- Syntax highlighting
- emoji
- markdown-it-checkbox
- markdown-it-container
- markdown-it-include
- PlantUML
    - markdown-it-plantuml
- mermaid
    - knsv/mermaid

Sample files

- pdf
- html
- png
- jpeg

markdown-it-container

INPUT

```
::: warning
*here be dragons*
:::
```

OUTPUT

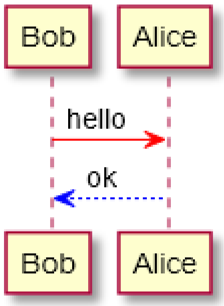
```
<div class="warning">
<p><em>here be dragons</em></p>
</div>
```

markdown-it-plantuml

INPUT

```
@startuml
Bob -[#red]> Alice : hello
Alice -[#0000FF]->Bob : ok
@enduml
```

OUTPUT



markdown-it-include

Include markdown fragment files: `:[alternate-text](relative-path-to-file.md)`.

```
├── [plugins]
│  └── README.md
├── CHANGELOG.md
└── README.md
```

INPUT

```
README Content

:[Plugins](./plugins/README.md)

:[Changelog](CHANGELOG.md)
```

OUTPUT

```
Content of README.md

Content of plugins/README.md

Content of CHANGELOG.md
```

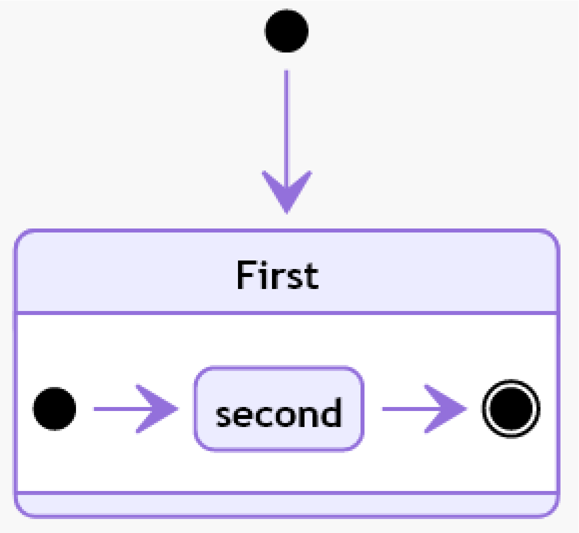
mermaid

INPUT

~~~
```mermaid
stateDiagram
    [*] --> First
    state First {
        [*] --> second
~~~

~~~
            second --> [*]
        }
    ```
~~~

OUTPUT



Install

Chromium download starts automatically when Markdown PDF is installed and Markdown file is first opened with Visual Studio Code.

However, it is time-consuming depending on the environment because of its large size (~ 170Mb Mac, ~ 282Mb Linux, ~ 280Mb Win).

During downloading, the message `Installing Chromium` is displayed in the status bar.

If you are behind a proxy, set the `http.proxy` option to settings.json and restart Visual Studio Code.

If the download is not successful or you want to avoid downloading every time you upgrade Markdown PDF, please specify the installed Chrome or 'Chromium' with markdown-pdf.executablePath option.

Usage

Command Palette

1. Open the Markdown file
2. Press `F1` or `Ctrl+Shift+P`
3. Type `export` and select below
   - `markdown-pdf: Export (settings.json)`
   - `markdown-pdf: Export (pdf)`
   - `markdown-pdf: Export (html)`
   - `markdown-pdf: Export (png)`
   - `markdown-pdf: Export (jpeg)`
   - `markdown-pdf: Export (all: pdf, html, png, jpeg)`

Menu

1. Open the Markdown file
2. Right click and select below
   - `markdown-pdf: Export (settings.json)`
   - `markdown-pdf: Export (pdf)`
   - `markdown-pdf: Export (html)`
   - `markdown-pdf: Export (png)`
   - `markdown-pdf: Export (jpeg)`
   - `markdown-pdf: Export (all: pdf, html, png, jpeg)`

Auto convert

1. Add `"markdown-pdf.convertOnSave": true` option to **settings.json**
2. Restart Visual Studio Code
3. Open the Markdown file
4. Auto convert on save

Extension Settings

Visual Studio Code User and Workspace Settings

1. Select **File > Preferences > UserSettings or Workspace Settings**
2. Find markdown-pdf settings in the **Default Settings**
3. Copy `markdown-pdf.*` settings
4. Paste to the **settings.json**, and change the value

Options

List

Category	Option name
Save options	markdown-pdf.type
	markdown-pdf.convertOnSave
	markdown-pdf.convertOnSaveExclude
	markdown-pdf.outputDirectory
	markdown-pdf.outputDirectoryRelativePathFile
Styles options	markdown-pdf.styles
	markdown-pdf.stylesRelativePathFile
	markdown-pdf.includeDefaultStyles
Syntax highlight options	markdown-pdf.highlight
	markdown-pdf.highlightStyle
Markdown options	markdown-pdf.breaks
Emoji options	markdown-pdf.emoji
Configuration options	markdown-pdf.executablePath

Category	Option name
Common Options	markdown-pdf.scale
PDF options	markdown-pdf.displayHeaderFooter
	markdown-pdf.headerTemplate
	markdown-pdf.footerTemplate
	markdown-pdf.printBackground
	markdown-pdf.orientation
	markdown-pdf.pageRanges
	markdown-pdf.format
	markdown-pdf.width
	markdown-pdf.height
	markdown-pdf.margin.top
	markdown-pdf.margin.bottom
	markdown-pdf.margin.right
	markdown-pdf.margin.left
PNG JPEG options	markdown-pdf.quality
	markdown-pdf.clip.x
	markdown-pdf.clip.y
	markdown-pdf.clip.width
	markdown-pdf.clip.height
	markdown-pdf.omitBackground
PlantUML options	markdown-pdf.plantumlOpenMarker
	markdown-pdf.plantumlCloseMarker
markdown-it-include options	markdown-pdf.markdown-it-include.enable

Save options

`markdown-pdf.type`

- Output format: pdf, html, png, jpeg
- Multiple output formats support
- Default: pdf

```
"markdown-pdf.type": [
  "pdf",
  "html",
  "png",
  "jpeg"
],
```

`markdown-pdf.convertOnSave`

- Enable Auto convert on save
- boolean. Default: false
- To apply the settings, you need to restart Visual Studio Code

`markdown-pdf.convertOnSaveExclude`

- Excluded file name of convertOnSave option

```
"markdown-pdf.convertOnSaveExclude": [
  "^work",
  "work.md$",
  "work|test",
  "[0-9][0-9][0-9][0-9]-work",
  "work\\test"  // All '\' need to be written as '\\' (Windows)
],
```

`markdown-pdf.outputDirectory`

- Output Directory
- All `\` need to be written as `\\` (Windows)

```
"markdown-pdf.outputDirectory": "C:\\work\\output",
```

- Relative path
  - If you open the `Markdown file`, it will be interpreted as a relative path from the file
  - If you open a `folder`, it will be interpreted as a relative path from the root folder
  - If you open the `workspace`, it will be interpreted as a relative path from the each root folder
    - See Multi-root Workspaces

```
"markdown-pdf.outputDirectory": "output",
```

- Relative path (home directory)
  - If path starts with `~`, it will be interpreted as a relative path from the home directory

```
"markdown-pdf.outputDirectory": "~/output",
```

- If you set a directory with a `relative path`, it will be created if the directory does not exist
- If you set a directory with an `absolute path`, an error occurs if the directory does not exist

`markdown-pdf.outputDirectoryRelativePathFile`

- If `markdown-pdf.outputDirectoryRelativePathFile` option is set to `true`, the relative path set with markdown-pdf.outputDirectory is interpreted as relative from the file
- It can be used to avoid relative paths from folders and workspaces
- boolean. Default: false

Styles options

`markdown-pdf.styles`

- A list of local paths to the stylesheets to use from the markdown-pdf
- If the file does not exist, it will be skipped
- All `\` need to be written as `\\` (Windows)

```
"markdown-pdf.styles": [
  "C:\\Users\\<USERNAME>\\Documents\\markdown-pdf.css",
  "/home/<USERNAME>/settings/markdown-pdf.css",
],
```

- Relative path
  - If you open the `Markdown file`, it will be interpreted as a relative path from the file
  - If you open a `folder`, it will be interpreted as a relative path from the root folder
  - If you open the `workspace`, it will be interpreted as a relative path from the each root folder
    - See Multi-root Workspaces

```
"markdown-pdf.styles": [
  "markdown-pdf.css",
],
```

- Relative path (home directory)
  - If path starts with `~`, it will be interpreted as a relative path from the home directory

```
"markdown-pdf.styles": [
  "~/.config/Code/User/markdown-pdf.css"
],
```

- Online CSS (https://xxx/xxx.css) is applied correctly for JPG and PNG, but problems occur with PDF #67

```
"markdown-pdf.styles": [
  "https://xxx/markdown-pdf.css"
],
```

`markdown-pdf.stylesRelativePathFile`

- If `markdown-pdf.stylesRelativePathFile` option is set to `true`, the relative path set with markdown-pdf.styles is interpreted as relative from the file
- It can be used to avoid relative paths from folders and workspaces
- boolean. Default: false

`markdown-pdf.includeDefaultStyles`

- Enable the inclusion of default Markdown styles (VSCode, markdown-pdf)
- boolean. Default: true

Syntax highlight options

`markdown-pdf.highlight`

- Enable Syntax highlighting
- boolean. Default: true

`markdown-pdf.highlightStyle`

- Set the style file name. for example: github.css, monokai.css ...
- file name list
- demo site : https://highlightjs.org/static/demo/

```
"markdown-pdf.highlightStyle": "github.css",
```

Markdown options

`markdown-pdf.breaks`

- Enable line breaks
- boolean. Default: false

Emoji options

`markdown-pdf.emoji`

- Enable emoji. EMOJI CHEAT SHEET

- boolean. Default: true

Configuration options

`markdown-pdf.executablePath`

- Path to a Chromium or Chrome executable to run instead of the bundled Chromium
- All `\` need to be written as `\\` (Windows)
- To apply the settings, you need to restart Visual Studio Code

```
"markdown-pdf.executablePath": "C:\\Program Files (x86)\\Google\\Chrome\\Application\\chrome.exe"
```

Common Options

`markdown-pdf.scale`

- Scale of the page rendering
- number. default: 1

```
"markdown-pdf.scale": 1
```

PDF options

- pdf only. puppeteer page.pdf options

`markdown-pdf.displayHeaderFooter`

- Enable display header and footer
- boolean. Default: true

`markdown-pdf.headerTemplate`

`markdown-pdf.footerTemplate`

- HTML template for the print header and footer
- `<span class='date'></span>` : formatted print date
- `<span class='title'></span>` : markdown file name
- `<span class='url'></span>` : markdown full path name
- `<span class='pageNumber'></span>` : current page number
- `<span class='totalPages'></span>` : total pages in the document

```
"markdown-pdf.headerTemplate": "<div style=\"font-size: 9px; margin-left: 1cm;\"> <span class='title'></span></div> <div style=\"font-size: 9px; margin-left: auto; margin-right: 1cm; \"> <span class='date'></span></div>",
"markdown-pdf.footerTemplate": "<div style=\"font-size: 9px; margin: 0 auto;\">
<span class='pageNumber'></span> / <span class='totalPages'></span></div>",
```

`markdown-pdf.printBackground`

- Print background graphics
- boolean. Default: true

`markdown-pdf.orientation`

- Paper orientation
- portrait or landscape
- Default: portrait

`markdown-pdf.pageRanges`

- Paper ranges to print, e.g., '1-5, 8, 11-13'
- Default: all pages

```
"markdown-pdf.pageRanges": "1,4-",
```

`markdown-pdf.format`

- Paper format
- Letter, Legal, Tabloid, Ledger, A0, A1, A2, A3, A4, A5, A6
- Default: A4

```
"markdown-pdf.format": "A4",
```

`markdown-pdf.width`

`markdown-pdf.height`

- Paper width / height, accepts values labeled with units(mm, cm, in, px)
- If it is set, it overrides the markdown-pdf.format option

```
"markdown-pdf.width": "10cm",
"markdown-pdf.height": "20cm",
```

`markdown-pdf.margin.top`

`markdown-pdf.margin.bottom`

`markdown-pdf.margin.right`

`markdown-pdf.margin.left`

- Paper margins.units(mm, cm, in, px)

```
"markdown-pdf.margin.top": "1.5cm",
"markdown-pdf.margin.bottom": "1cm",
"markdown-pdf.margin.right": "1cm",
"markdown-pdf.margin.left": "1cm",
```

PNG JPEG options

- png and jpeg only. puppeteer page.screenshot options

`markdown-pdf.quality`

- jpeg only. The quality of the image, between 0-100. Not applicable to png images

```
"markdown-pdf.quality": 100,
```

`markdown-pdf.clip.x`

`markdown-pdf.clip.y`

`markdown-pdf.clip.width`

`markdown-pdf.clip.height`

- An object which specifies clipping region of the page
- number

```
//  x-coordinate of top-left corner of clip area
"markdown-pdf.clip.x": 0,

// y-coordinate of top-left corner of clip area
"markdown-pdf.clip.y": 0,

// width of clipping area
"markdown-pdf.clip.width": 1000,

// height of clipping area
"markdown-pdf.clip.height": 1000,
```

`markdown-pdf.omitBackground`

- Hides default white background and allows capturing screenshots with transparency
- boolean. Default: false

PlantUML options

`markdown-pdf.plantumlOpenMarker`

- Oppening delimiter used for the plantuml parser.
- Default: @startuml

`markdown-pdf.plantumlCloseMarker`

- Closing delimiter used for the plantuml parser.
- Default: @enduml

markdown-it-include options

`markdown-pdf.markdown-it-include.enable`

- Enable markdown-it-include.
- boolean. Default: true

FAQ

How can I change emoji size ?

1. Add the following to your stylesheet which was specified in the markdown-pdf.styles

```
.emoji {
  height: 2em;
}
```

Auto guess encoding of files

Using `files.autoGuessEncoding` option of the Visual Studio Code is useful because it automatically guesses the character code. See files.autoGuessEncoding

```
"files.autoGuessEncoding": true,
```

Output directory

If you always want to output to the relative path directory from the Markdown file.

For example, to output to the "output" directory in the same directory as the Markdown file, set it as follows.

```
"markdown-pdf.outputDirectory" : "output",
"markdown-pdf.outputDirectoryRelativePathFile": true,
```

Page Break

Please use the following to insert a page break.

```
<div class="page"/>
```

Known Issues

`markdown-pdf.styles` option

- Online CSS (https://xxx/xxx.css) is applied correctly for JPG and PNG, but problems occur with PDF. #67

Release Notes

1.4.3 (2020/03/12)

- Fix: markdown-include regular expression
  - Fix: Unable to export to pdf from markdown #166
  - Fix: python code export err #178
- Fix: Add support for Ubuntu and Centos
  - Fix: Error: Failed to lanuch chrome! #97
  - Fix: I failed to launch chrome in WSL #160
  - Fix: Unable to export to pdf from markdown #166

License

MIT

Special thanks

- GoogleChrome/puppeteer
- markdown-it/markdown-it
- mcecot/markdown-it-checkbox
- leff/markdown-it-named-headers
- markdown-it/markdown-it-emoji
- HenrikJoreteg/emoji-images
- isagalaev/highlight.js
- cheeriojs/cheerio
- janl/mustache.js
- markdown-it/markdown-it-container
- gmunguia/markdown-it-plantuml
- camelaissani/markdown-it-include
- knsv/mermaid
- jonschlinkert/gray-matter

and

- cakebake/markdown-themeable-pdf