`cleanNote` - a Fancy Note Package

Yutian Chen

December 21, 2022

Contents

This package is an encapsulation of styling and environment definition for daily note-taking. As the name reveals, We aimed to make the output PDF more ‘modernized’ and ‘clean’.

`cleanNote` also provides some useful functions like **dark mode** and **hide solution** (for review use).

1 Theme and Options for `cleanNote`

```
% Argument Syntax
\usepackage[Dark?, HideSolution?, NoIndent?]{cleanNote}

% Examples
\usepackage{cleanNote}                     % Normal setup (light theme)
\usepackage[Dark]{cleanNote}               % Dark Theme
\usepackage[Dark, HideSolution]{cleanNote} % Hide solution, Dark theme
\usepackage[NoIndent]{cleanNote}    % No indent at beginning of paragraph
```

Currently, `cleanNote` supports three options:

- `Dark` - Passing this argument into `cleanNote` will make output PDF set to "dark mode"
- `HideSolution` - Passing this argument into `cleanNote` will make output PDF hide all solution (by setting color of solution text same as page background).
- `NoIndent` - Passing this argument into `cleanNote` will remove the indentation at beginning of paragraph.

2 Environments

Multiple environments are defined in `cleanNote`. This section will introduce and demonstrate each of them:

2.1 Definition

```
\begin{definition}[<Term>]?
    <Content>
\end{definition}
```

Definition (Terminology)**.** The technical or special terms used in a business, art, science, or special subject (source)

Note that the argument `<term>` is optional, meaning that you can create a definition without passing any argument.

Definition. This is a definition without the `<term>` argument.

2.2 Theorem

```
\begin{theorem}[<Theorem_name>]?
    <Content>
\end{theorem}
```

Theorem 2.1: Pythagorean Theorem

If c denotes the length of the hypotenuse and a and b denote the two lengths of the legs of a right triangle, then the Pythagorean theorem can be expressed as the Pythagorean equation:

$$a^2 + b^2 = c^2$$

Similar to environment `definition` , the `theorem` environment's argument can be omitted.

Theorem 2.2

This theorem is too trivial to receive a proper name :(

2.3 Proposition

```
\begin{proposition}[<Proposition_name>]?
    <Content>
\end{proposition}
```

Proposition. Proposition is important, but not so important as a theorem, so it does not have a fancy box.

Proposition (Some Name)**.** Maybe some proposition is more important than others that it also deserves a name

2.4 Proof

```
\begin{proof}
    <Content>
\end{proof}
```

The `proof` environment is similar to the `proof` environment provided in `amsmath` package. An empty square represents Q.E.D. will be shown on the right at the end of proof.

Proof. Here's a simple proof that the Q.E.D. square does exist at the end of proof.

□

2.5 Note

Something is so important that it is better to be highlighted from normal texts. If that's the case, then `note` environment is the one you need!

```
\begin{note}
    <Content>
\end{note}
```

> *Notes.* Something is just so special that it deserves a beautiful background.

2.6 Example

```
\begin{example}[<Question_name>]?
    <Content>
\end{example}
```

> **Example** (Example Question)**.** *Some ‘Example example’ is happening here haha.*

> **Example.** *Example of* `example` *environment without argument.*

2.7 Solution

Solution is just plain block with ‘End of solution’ marked at the end of block. However, when `HideSolution` argument is passed into the `cleanNote` package, all contents in the solution package will be hidden. (unless you explicitly set a color for the text!)

Solution. Example Solution...I really like the beautiful typeset of math equations in LaTeX.

$$\int_{\text{Born}}^{\text{Death}} \text{”My Heart is in the Work!”}\, \mathrm{d}t$$

→ *End of Solution*

2.8 Code Block

`codeblock` environment can be applied whenever you want to insert a chunk of code in your PDF. It’s syntax highlighting module (which relies on a python module called `pygmentize`) supports almost any language you can think of! (Sadly, no support for `C0` and `BC0` , though)

> *Notes.* Using code block feature with syntax highlighting requires a properly installed `pygmentize` package and also `-shell-escape` flag in compiler setting

```
\begin {codeblock} [<Language>]?
<Code_content>
\end {codeblock}
```

```
def main():
    print("Hello world")        # Python is the best language :)
```

```
function greeting() {
    console.log("Hello world"); // I like adding ; at the end of JS!
}
```

Maybe you are wondering what will happen if there's a suuuuuper long line in your code, so here's a demo:

```
const f = (UIHook: ReactHooks, mem: C0HeapAllocator, arg1: C0Value<Maybe<"value">>, arg2:
↪   C0Value<Maybe<"string">>) => { ... };
```

3 Integration with Visual Studio Code

3.1 `ltex` - Syntax Checking Plugin

If you are using `ltex` plugin as a syntax checker for your LaTeX project, you can insert following configurations into `./.vscode/settings.json` file.

These configurations will tell `ltex` not to check syntax for contents in `Codeblock` environment or `\code{...}` macro.

```
"ltex.latex.environments": {
    "lstlisting": "ignore",
    "verbatim": "ignore",
    "codeblock": "ignore",
},
"ltex.latex.commands": {
    "\\label{}": "ignore",
    "\\documentclass[]{}": "ignore",
    "\\cite{}": "dummy",
    "\\cite[]{}": "dummy",
    "\\code{}": "dummy"
}
```

3.2 Code Snippets

You can also import the `tex_snippet.code-snippets` into `./.vscode` directory to apply some simple snippets to your LaTeX project.

The available snippets are listed below:

1. `EQ*` - Equation block

```
\begin{equation*}
    <1>
\end{equation*}
```

2. AEQ* - Multiline Equation Block with Alignment

```
\begin{equation*}
    \begin{aligned}
        <1>
    \end{aligned}
\end{equation*}
```

3. EG - Example Environment Block

```
\begin{example}
    <1>
\end{example}
```

4. SOL - Solution Environment Block

```
\begin{solution}
    <1>
\end{solution}
```

5. CODE - Code Block Environment

```
\begin {codeblock}[<1>]
<2>
\end {codeblock}
```

6. LR[] - Quick Left Right Square Bracket

```
\left [ <1> \right ]
```

7. LR{} - Quick Left Right Bracket

```
\left\{ <1> \right\}
```

8. DEF - Definition Environment

```
\begin{definition}[<1>]
    <2>
\end{definition}
```

9. RA - Right arrow under Math Mode

```
\rightarrow
```

10. `THM` - Theorem Environment

```
\begin{theorem}[<1>]
    <2>
\end{theorem}
```

11. `VBAR` - Vertical bar within left-right brackets

```
\,\middle |\,
```

12. `TAB` - Quick framework for tabular environment

```
\begin{center}
\begin{tabular}{<1>}
    <2>
\end{tabular}
\end{center}
```

13. `FIGBLOCK` - Figure block (takes complete width on page)

```
\begin{figure}[ht]
   \begin{center}
   \centering\includegraphics[width=<1>]{assets/<2>}
   \caption{\color{\captionColor} <3>}
   \end{center}
\end{figure}
```

14. `FIGINLINE` - Figure inline (figure that floats on left / right, default right)

```
\begin{wrapfigure}{r}{<1>\textwidth}
    \centering\includegraphics[width=<2>\textwidth]{<3>}
\end{wrapfigure}
```

For the actual configuration files, see Section 4 Appendix.

3.3 Key Bindings

You can also import the key bindings file into `VSCode` setting (`keybindings.json`).

```
[
    {
        "key": "ctrl+shift+b",
        "command": "editor.action.insertSnippet",
        "when": "editorHasSelection",
        "args": {
            "snippet": "\\textbf{${TM_SELECTED_TEXT}}$0"
        }
    },
    {
        "key": "ctrl+shift+i",
        "command": "editor.action.insertSnippet",
        "when": "editorHasSelection",
        "args": {
            "snippet": "\\textit{${TM_SELECTED_TEXT}}$0"
        }
    }
]
```

After importing these configurations, when you select some text in editor and press `ctrl + shift + b`, the text will become **bold** automatically.

Similarly, pressing `ctrl + shift + i` will make text *italic*.

4 Appendix: Full Snippet Setup File

```
{
    "Clean Equation Block" : {
        "scope": "latex",
        "prefix": "EQ*",
        "body":[
            "\\begin{equation*}",
            "    $1",
            "\\end{equation*}"
        ]
    },
    "Aligned Equation Block": {
        "scope": "latex",
        "prefix": "AEQ*",
        "body": [
            "\\begin{equation*}",
            "    \\begin{aligned}",
            "        $1",
            "    \\end{aligned}",
            "\\end{equation*}"
        ]
    },
    "Example Block": {
        "scope": "latex",
        "prefix": "EG",
        "body":[
            "\\begin{example}",
            "    $1",
            "\\end{example}"
        ]
    },
    "Solution Block":{
        "scope": "latex",
        "prefix": "SOL",
        "body":[
            "\\begin{solution}",
            "    $1",
            "\\end{solution}"
        ]
    },
    "Code Block":{
        "scope": "latex",
        "prefix": "CODE",
        "body": [
            "\\begin{codeblock}[$1]",
            "$2",
            "\\end {codeblock}"
        ]
    },
    "Left Right []":{
        "scope": "latex",
        "prefix": "LR[]",
        "body":[
            "\\left[ $1 \\right]"
        ]
    },
    "Left Right {}":{
        "scope": "latex",
        "prefix": "LR{}",
        "body":[
            "\\left\\{ $1 \\right\\\\}"
        ]
    },
    "Definition Block": {
        "scope": "latex",
        "prefix": "DEF",
        "body":[
            "\\begin{definition}[$1]",
            "    $2",
            "\\end{definition}"
        ]
    },
    "Right Arrow":{
        "scope": "latex",
        "prefix": "RA",
        "body":[
            "\\rightarrow"
        ]
    },
    "Theorem Block": {
        "scope": "latex",
        "prefix": "THM",
        "body":[
            "\\begin{theorem}[$1]",
            "    $2",
            "\\end{theorem}"
        ]
    },
    "Vertical Bar": {
        "scope": "latex",
        "prefix": "VBAR",
        "body":[
            "\\,\\middle |\\,"
        ]
    },
    "Table": {
        "scope": "latex",
        "prefix": "TAB",
        "body":[
            "\\begin{center}",
            "\\begin{tabular}{$1}",
            "    $2",
            "\\end{tabular}",
            "\\end{center}"
        ],
    },
    "Block Figure": {
        "scope": "latex",
        "prefix": "FIGBLOCK",
        "body": [
            "\\begin{figure}[ht]",
            "   \\begin{center}",
            "   \\centering\\includegraphics[width=$1]{assets/$2}",
            "   \\caption{\\color{\\captionColor} $3}",
            "   \\end{center}",
            "\\end{figure}"
        ]
    },
    "Inline Figure": {
        "scope": "latex",
        "prefix": "FIGINLINE",
        "body": [
            "\\begin{wrapfigure}{r}{$1\\textwidth}",
            "    \\centering\\includegraphics[width=$2\\textwidth]{$3}",
            "\\end{wrapfigure}"
        ]
    }
}
```