

Bring Your Own Blocks

With ObsiTex, you can define custom parsing blocks within the Markdown parser. This allows you to incorporate custom elements into your Markdown files that go beyond the standard Markdown syntax.

1 Line Break

ObsiTex doesn't support line breaks, but they might be added by creating a custom parser, as follows:

```
from obsitex.parser.blocks import LaTeXBlock

class LineBreakBlock(LaTeXBlock):
    def __init__(self):
        super().__init__("\\newpage", in_latex=True)

    @staticmethod
    def detect_block(lines, index):
        if lines[index].startswith("—"):
            return LineBreakBlock(), index + 1

# Add the custom block to the parser
from obsitex import ObsidianParser


parser = ObsidianParser(custom_blocks=[LineBreakBlock])
```

And now a newline will be added to the LaTeX output whenever a line starting with --- is detected in the markdown file. For example:

In the PDF output, this will be rendered as a new page.

2 Warnings

It's also possible to include custom warnings, for example:

**This is a custom warning block.**

This is a very big and random multiline warning block. It can contain multiple lines of text. You can use formatting like **bold** or *italic*, as well as `code blocks`. And it will be rendered as a warning in the PDF output. This is the last line of the warning block.

This can be achieved with the following code:

```
from obsitex.parser.blocks import AbstractCallout
from obsitex.parser.formatting import format_text

class CustomWarningBlock(AbstractCallout):
    def formatted_text(self, **kwargs):
        content = "\n".join(format_text(self.content))

        return f"""
\\begin{{{bclogo}}}[logo=\\bcattention, couleurBarre=red,
nborder=true,
couleur=LightSalmon]{{{self.caption}}}
{content}
\\end{{{bclogo}}}
"""

    @staticmethod
    def detect_block(lines, index):
        return AbstractCallout.detect_block(lines, index,
            "warning", CustomWarningBlock)
```

For more information on how to use, check the `run_sample.py` file in the `samples/byob` folder.