Jupyter Book Plugins

Luuk Fröling and Freek Pols

September 23, 2025

Contents

0.1 PDF Support 1

0.1.1 PDF support - custom directives 2

0.2 Typst support 3

0.2.1 Typst support - raw typst 4

0.2.2 New typst element 5

0.1 PDF Support

0.1.1 PDF support - custom directives

Here is an example
With some text.

example Use this link to add support for the example directive when converting to pdf:

`https://github.com/TUD -JB -Templates/JB2_plugins/releases/download/example/example.mjs`

this is an experiment
with a description

experiment Use this link to add support for the experiment directive when converting to pdf:

`https://github.com/TUD -JB -Templates/JB2_plugins/releases/download/experiment/experiment.mjs`

this is an intermezzo
with a description

intermezzo Use this link to add support for intermezzo's when converting to pdf:

`https://github.com/TUD -JB -Templates/JB2_plugins/releases/download/intermezzo/intermezzo.mjs`

exercises

Exercise 1: 1 + 1 = ... ?

Solution to Exercise 1: 1 + 1 = ... ?
2

Use this link to add support for exercises when converting to pdf:

`https://github.com/TUD -JB -Templates/JB2_plugins/releases/download/exercise/pdf_exercise.mjs`

Custom element To generate a custom element, including css

- style must have been included in css file.

0.2 Typst support

0.2.1 Typst support - raw typst

use

```
const generateTypstNode = (val) => {
  let result = {
    type: ’raw’,
    lang: ’typst’,
    typst: val
  }
  return result
}
```

to replace any node with a typst node.

Here is an example
With some text.

0.2.2 New typst element

define element

Create new file, aside_style.typ in this example

```
// Custom typst element (https://typst.app/docs/reference/)

#let aside(title, body) = {

    // Title block
    block(
        // Outer block styled like a note block
        fill: rgb(244, 147, 147), // background - VS code automatically changes this to rgba s
        stroke: (left: 1pt + red),  // red line on the left
        width: 100%,

        // add padding top and bottom
        inset: (x: 0.8em, y: 0.4em),
        above: 0.5em,               //distance top to next element above
        below: 0em,                 //distance bottom to next element below

        strong(title)
    )
    block(
        // Inner block for the body of the aside
        fill: rgb(240, 240, 240),        // white background
        stroke: (left: 1pt + red),
        width: 100%,

        // add padding top and bottom
        inset: (x: 0.8em, y: 0.6em),
        above: 0em,
        below: 0.5em,

        body
    )
}
```

the content of the element follows the same style as a noteBlock.

Add new .typ file to template.yml

```
files:
  - template.typ
  - style.typ
  - aside_style.typ
  - LICENSE
  - Cover.PNG
```

add element via plugin

As we wanted to replace the aside element via a plugin, we inject the raw typst:

```
rootChildren[i] = generateTypstNode(
        ’#import "aside_style.typ": aside \n #aside("’ + title + ’", "’ + body.map(b => getTex
    );
```

Make sure to include the new file as well, before naming the element.