_extensions/custom-numbered-blo...	fix typo in _extension.yaml	3 months ago
doc	version 0.1.0	4 months ago
.gitignore	version 0.1.0	4 months ago
LICENSE	Create LICENSE	4 months ago
README.md	Update README.md	4 months ago
example.qmd	fix, and modify example	4 months ago

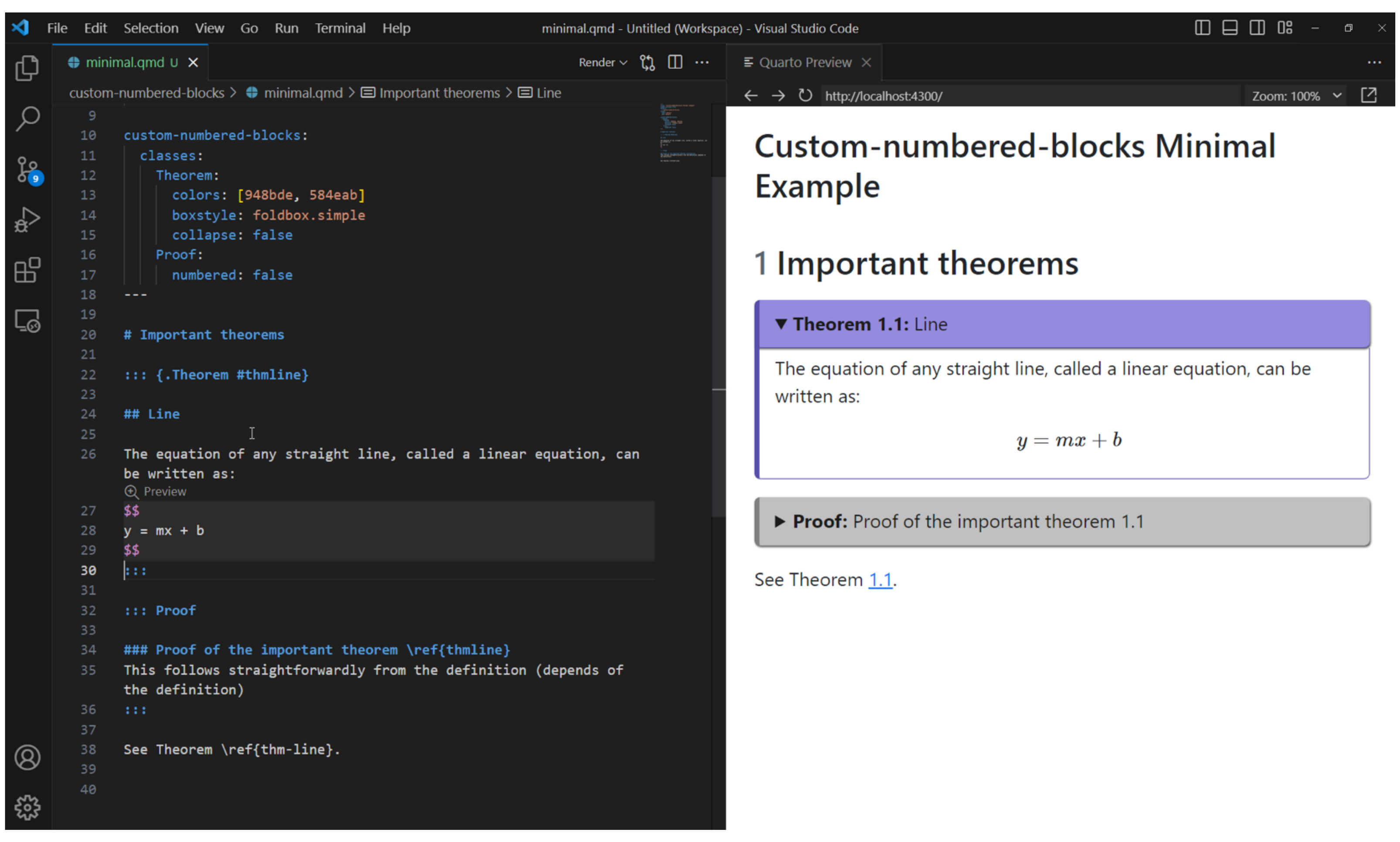
README.md

Custom Numbered Blocks Extension for Quarto

This extension provides user defined custom div classes (environments) that come with numbering, such as theorems, examples, exercises. Numbered blocks can be cross referenced.

- By default, the div's text block is enclosed in a collapsible box, similar to quarto callouts.
- Groups of classes can be defined that share style and numbering.
- Lists-of-classes can be extracted, such as a list of all theorems. It is also possible to generate a list for a group of classes.

The filter supports output formats pdf and html.



Status

Seems that Quarto 1.3 handles pdf books differently from Quarto 1.2. If chapters contain additional level 1 heading, this messes up numbering in Quarto 1.3 pdf books. I will likely fix that soon.

There may come changes to the yaml-UI for lists-of-classes, also soon ;-).

Installing

```
quarto add ute/custom-numbered-blocks
```

This will install the extension under the `_extensions` subdirectory. If you're using version control, you will want to check in this directory.

Using

Usage is illustrated more comprehensively in `example.qmd`.

Defining and using a user defined class

To specify a new class of numbered div blocks, `Example`, say, add yaml code:

```
custom-numbered-blocks:
  classes:
    Example: default
```

Use the class in a fenced dive. Title can be provided as a header immediately after div.

```
::: Example
### the best possible example, ever
here is some exemplary text
:::
```

Change default options for a class

The default style renders as a collapsible box with title button, similar to quarto callouts. It comes with a small close button bottom right. You can change the following options in yaml or individually in the div specification:

- `colors` : an array of two hex rgb color codes, for title button color and frame color. `colors: [a08080, 500000]` would give a pink title button and dark red frame.
- `collapse` : boolean, default `true`. Initial state of the collapsible box.
- `label` : the label to print before the number (string).
- `boxstyle` : set to `foldbox.simple` for a boxed environment without close button. There will quite likely come more options in a future version.
- `listin` : register for a list-of

Groups of classes with joint counter and joint default style

Jointly counted block classes are specified by yaml option `groups`. These groups can also have a common default style. For each class, option `group` indicates membership.

Example: we want to jointly count theorems, lemmas and propositions, and render boxes with initially open status, but propositions should be collapsed:

```
custom-numbered-blocks:
  groups:
    thmlike:
      collapse: false
  classes:
    Theorem:
      group: thmlike
    Proposition:
      group: thmlike
      collapse: true
    Lemma:
      group: thmlike
```

Lists-of ("listin" version)

To generate a list of all divs belonging to a class, `Example`, say, add key listin to the class and give the name of the list. The same can be done for groups of classes. This will produce a file `list-of-` name `.qmd` that contains headers and references to the respective blocks. The following code will generage files `list-of-allthingsmath.qmd` and `list-of-examples.qmd`:

```
custom-numbered-blocks
  groups:
    thmlike:
      collapse: false
      listin: [allthingsmath]
    Example:
      listin: [examples, allthingsmath]
```

Example

Here is the source code for a (not so) minimal example: example.qmd. And here's the rendered example.html and example.pdf

About

quarto extension for user defined, automatically numbered div blocks / latex environment

quarto quarto-extension quarto-filter

Readme
MIT license
Activity
11 stars
2 watching
0 forks
Report repository

Releases 2

v0.1.1: fixes Latest
on May 24

+ 1 release

Packages

No packages published

Languages

Lua 88.6% CSS 7.3% TeX 4.1%