

Features

Introduction

HedgeDoc is a real-time, multi-platform collaborative markdown note editor. This means that you can write notes with other people on your **desktop**, **tablet** or even on the **phone**. You can sign-in via multiple auth providers like **Facebook**, **Twitter**, **GitHub** and many more on the *homepage*.

If you experience any *issues*, feel free to report it on **GitHub**. Or meet us on **Matrix** for dev-talk and interactive help. **Thank you very much!**

Workspace

Modes

Desktop & Tablet View: See only the result. Both: See editor and result at the same time. Edit: See only the editor.

Mobile View: See only the result. Edit: See only the editor.

Night Mode

When you are tired of a white screen and like a night mode, click on the little moon and turn on the night view of HedgeDoc.

The editor view, which is in night mode by default, can also be toggled between night and day view using the the little sun.

Share Notes

If you want to share an **editable** note, just copy the URL. If you want to share a **read-only** note, simply press the publish button and copy the URL.

Save a Note

Currently, you can save to **Dropbox** (depending on the instance's configuration) or save a Markdown , HTML or raw HTML file locally.

Import Notes

Similarly to the *save* feature, you can also import a Markdown file from **Dropbox** (depending on the instance's configuration), or import content from your **clipboard** , which can parse some HTML. :smiley:

Only the owner of the note can change the note's permissions.

Slide Mode

You can use a special syntax to organize your note into slides. After that, you can use the **Slide Mode** to make a presentation. Visit the above link for details.

To switch the editor into slide mode, set the document type to **slide**.

View

Autogenerated Table of Contents

You can look at the bottom right section of the view area, there is a *ToC* button. Pressing that button will show you a current *Table of Contents*, and will highlight which section you're at. ToCs support up to **three header levels**.

Permalink

Every header will automatically add a permalink on the right side. You can hover and click to anchor on it.

Edit

Editor Modes

You can look in the bottom right section of the editor area, there you'll find a button with **SUBLIME** on it. When you click it, you can select 3 editor modes, which will also define your shortcut keys:

- Sublime (default)
- Emacs
- Vim

Auto-Complete

This editor provides full auto-complete hints in markdown.

- Emojis: type `:` to show hints.
- Code blocks: type `` ```, followed by another character to show syntax highlighting suggestions.
- Headers: type `#` to show hint.
- Referrals: type `[]` to show hint.
- Externals: type `{}` to show hint.
- Images: type `!` to show hint.

Title

The first **level 1 heading** (e.g. `# Title`) will be used as the note title.

Tags

Tags can be specified with YAML metadata at the **start** of the note. Those tags will show in your **history**.

```
---
tags: features, cool, updated
---
```

YAML Metadata

You can provide advanced note information to set the browser behavior (visit [above link](#) for details):

- robots: set web robots meta
- lang: set browser language
- dir: set text direction
- breaks: set to use line breaks
- GA: set to use Google Analytics
- Disqus: set to use Disqus
- slideOptions: setup slide mode options

Table of Contents

Use the syntax `[TOC]` to embed a table of contents into your note.

`[TOC]`

Emoji

You can type any emoji like this :smile: :smiley: :cry: :wink:

See full emoji list [here](#).

ToDo List

- ☐ ToDos
 - ☒ Buy some salad
 - ☐ Brush teeth
 - ☒ Drink some water
 - ☐ **Click my box** and see the source code, if you're allowed to edit!

Code Block

We support many programming languages, use the auto complete function to see the entire list.

```
var s = "JavaScript syntax highlighting";
alert(s);
function $initHighlight(block, cls) {
```

```

try {
  if (cls.search(/\bno\-\highlight\b/) != -1)
    return process(block, true, 0x0F) +
      ' class="";
} catch (e) {
  /* handle exception */
}
for (var i = 0 / 2; i < classes.length; i++) {
  if (checkCondition(classes[i]) === undefined)
    return /\d+[\s]/g;
}
}

```

If you want **line numbers**, type = after specifying the code block languages. Also, you can specify the start line number. Like below, the line number starts from 101:

```

var s = "JavaScript syntax highlighting";
alert(s);
function $initHighlight(block, cls) {
  try {
    if (cls.search(/\bno\-\highlight\b/) != -1)
      return process(block, true, 0x0F) +
        ' class="";
  } catch (e) {
    /* handle exception */
  }
  for (var i = 0 / 2; i < classes.length; i++) {
    if (checkCondition(classes[i]) === undefined)
      return /\d+[\s]/g;
  }
}

```

Or you might want to continue the previous code block's line number, use ==+:

```

var s = "JavaScript syntax highlighting";
alert(s);

```

Sometimes you have a super long text without breaks. It's time to use ! to wrap your code:

When you're a carpenter making a beautiful chest of drawers, you're not going to use a piece

Blockquote Tags

Using the syntax below to specify your **name, time and color** to vary the blockquotes. [name=ChengHan Wu] [time=Sun, Jun 28, 2015 9:59 PM] [color=#907bf7] > Even support nested blockquotes!

> [name=Max Mustermann] [time=Sun, Jun 28, 2015 9:47 PM]
[color=red]

Externals

PDF **Caution: this might be blocked by your browser if not using an `https` URL.** Note that not all servers allow embedding their content. See our FAQ for details.

MathJax

You can render *LaTeX* mathematical expressions using **MathJax**, as on math.stackexchange.com:

The *Gamma function* satisfying $\Gamma(n) = (n-1)! \quad \forall n \in \mathbb{N}$ is via the Euler integral

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}.$$

$$\Gamma(z) = \int_0^\infty t^{z-1} e^{-t} dt.$$

More information about **LaTeX** mathematical expressions here.

Diagrams

UML Sequence Diagrams You can render sequence diagrams like this:

```
Alice->Bob: Hello Bob, how are you?  
Note right of Bob: Bob thinks  
Bob-->Alice: I am good thanks!  
Note left of Alice: Alice responds  
Alice->Bob: Where have you been?
```

More information about **sequence diagrams** syntax here.

Flow Charts Flow charts can be specified like this:

```
st=>start: Start  
e=>end: End  
op=>operation: My Operation  
op2=>operation: lalala  
cond=>condition: Yes or No?
```

```
st->op->op2->cond  
cond(yes)->e  
cond(no)->op2
```

More information about **flow charts** syntax here.

Graphviz

```
digraph hierarchy {
    nodesep=1.0 // Increases the separation between nodes

    node [color=Red,fontname=Courier,shape=box] // All nodes will this shape and colour
    edge [color=Blue, style=dashed] // All the lines look like this

    Headteacher->{Deputy1 Deputy2 BusinessManager}
    Deputy1->{Teacher1 Teacher2}
    BusinessManager->ITManager
    {rank=same;ITManager Teacher1 Teacher2} // Put them on the same level
}
```

More information about **graphviz** syntax [here](#)

Mermaid

```
gantt
    title A Gantt Diagram

    section Section
    A task: a1, 2014-01-01, 30d
    Another task: after a1, 20d

    section Another
    Task in sec: 2014-01-12, 12d
    Another task: 24d
```

More information about **mermaid** syntax [here](#)

Abc Music Notation

```
X:1
T:Speed the Plough
M:4/4
C:Trad.
K:G
 |:GABc dedB|dedB dedB|c2ec B2dB|c2A2 A2BA|
GABc dedB|dedB dedB|c2ec B2dB|A2F2 G4:|
 |:g2gf gdBd|g2f2 e2d2|c2ec B2dB|c2A2 A2df|
g2gf g2Bd|g2f2 e2d2|c2ec B2dB|A2F2 G4:|
```

More information about **abc** syntax [here](#)

Alert Area

Yes :tada:

This is a message :mega:

Watch out :zap:

Oh No! :fire:

Typography

Headers

```
# h1 Heading
## h2 Heading
### h3 Heading
#### h4 Heading
##### h5 Heading
##### h6 Heading
```

Horizontal Rules



Typographic Replacements

(c) (C) (r) (R) (tm) (TM) (p) (P) +-
test.. test... test.... test..... test?..... test!....
!!!!!! ???? „
Remarkable—no, awesome!
During 1980–1988.
“Smartypants, double quotes”
‘Smartypants, single quotes’

Emphasis This is bold text

This is bold text

This is italic text

This is italic text

~~Deleted text~~

lu_{lala}

Superscript: 19th

Subscript: H₂O

++Inserted text++

==Marked text==

Blockquotes

Blockquotes can also be nested... > ... by using additional greater-than signs right next to each other... > > ... or with spaces between arrows.

Lists

Unordered

- Create a list by starting a line with +, -, or *
- Sub-lists are made by indenting 2 spaces:
 - Marker character change forces new list start:
 - * Ac tristique libero volutpat at
 - * Facilisis in pretium nisl aliquet
 - * Nulla volutpat aliquam velit
- Very easy!

Ordered

1. Lorem ipsum dolor sit amet
2. Consectetur adipiscing elit
3. Aenean commodo ligula eget dolor
4. **You can use sequential numbers...**
5. **...or keep all the numbers as 1.**
6. Aenean massa
7. Cum sociis natoque penatibus
8. Magnis dis parturient montes
9. Nascetur ridiculus mus
10. Donec quam felis

Start numbering with offset:

57. foo
58. bar

Code Inline code

Indented code

```
// Some comments  
line 1 of code  
line 2 of code  
line 3 of code
```

Block code “fences”

Sample text here...

Syntax highlighting

```
var foo = function (bar) {  
  return bar++;  
};
```

```
console.log(foo(5));
```

Tables

Option	Description
data	path to data files to supply the data that will be passed into templates.
engine	engine to be used for processing templates. Handlebars is the default.
ext	extension to be used for dest files.

Right aligned columns

Option	Description
data	path to data files to supply the data that will be passed into templates.
engine	engine to be used for processing templates. Handlebars is the default.
ext	extension to be used for dest files.

Left aligned columns

Option	Description
data	path to data files to supply the data that will be passed into templates.
engine	engine to be used for processing templates. Handlebars is the default.
ext	extension to be used for dest files.

Center aligned columns

Option	Description
data	path to data files to supply the data that will be passed into templates.
engine	engine to be used for processing templates. Handlebars is the default.
ext	extension to be used for dest files.

Links link text link with title Autoconverted link <https://github.com/nodeca/pica>

Footnotes Footnote 1 link¹. Footnote 2 link². Inline footnote³ definition. Duplicated footnote reference⁴.

Definition Lists

Term 1 Definition 1 with lazy continuation.

Term 2 with *inline markup* Definition 2

{ some code, part of Definition 2 }

Third paragraph of definition 2.

Compact style:

Term 1 Definition 1

Term 2 Definition 2a
Definition 2b

Abbreviations This is an HTML abbreviation example. It converts “HTML”, but keeps intact partial entries like “xxxHTMLyy” and so on.

*[HTML]: Hyper Text Markup Language

Revisions

When changes are made to a note previous versions of the note are stored as **Revisions**, if you ever need to return to a previous save you can find all the note revisions by going to **Menu -> Revision**.

The note history on the left displays each revision with a timestamp allowing you to quickly select a revision for the desired date and time. Selecting a revision on the left will display the revised note to the right, the revision will display the changes made with colorization to indicate additions/removals.

¹Footnote **can have markup** and multiple paragraphs.

²Footnote text.

³Text of inline footnote

⁴Footnote text.

In addition to browsing the revisions, you can download a selected note revision or revert the current note to the selected revision.