

remark

[ri-mahrk]

Go directly to [project site](#)

What is it and why should I be using it?

What is it?

A simple, in-browser, Markdown-driven slideshow tool targeted at people who know their way around HTML and CSS, featuring:

- Markdown formatting, with smart extensions
- Presenter mode, with cloned slideshow view
- Syntax highlighting, supporting a range of languages
- Slide scaling, thus similar appearance on all devices / resolutions *
- Touch support for smart phones and pads, i.e. swipe to navigate slides

* At least browsers try their best

What is it?

If your ideal slideshow creation workflow contains any of the following steps:

Why use it?

- Just write what's on your mind
- Do some basic styling
- Easily collaborate with others
- Share with and show to everyone

Then remark might be perfect for your next* slideshow!

* You probably want to convert existing slideshows as well

What is it?

As the slideshow is expressed using Markdown, you may:

- Focus on the content, expressing yourself in next to plain text not worrying what flashy graphics and disturbing effects to put where

Why use it?

As the slideshow is actually an HTML document, you may:

- Display it in any decent browser
- Style it using regular CSS, just like any other HTML content
- Use it offline!

As the slideshow is contained in a plain file, you may:

- Store it wherever you like; on your computer, hosted from your Dropbox, hosted on Github Pages alongside the stuff you're presenting...
- Easily collaborate with others, keeping track of changes using your favourite SCM tool, like Git or Mercurial

How does it work, then?

How does it work?

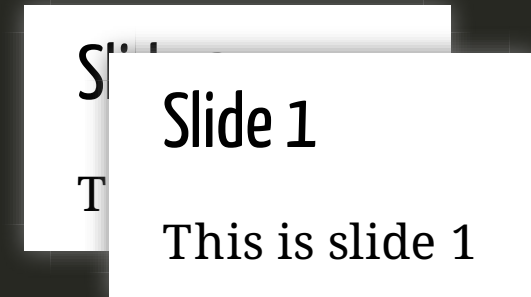
- Markdown

A Markdown-formatted chunk of text is transformed into individual slides by JavaScript running in the browser:

```
# Slide 1
This is slide 1

---

# Slide 2
This is slide 2
```



Regular Markdown rules apply with only a single exception:

- A line containing three dashes constitutes a new slide (not a horizontal rule, `<hr />`)

Have a look at the [Markdown website](#) if you're not familiar with Markdown formatting.

How does it work?

- Markdown

- Inside HTML

A simple HTML document is needed for hosting the styles, Markdown and the generated slides themselves:

```
<!DOCTYPE html>
<html>
  <head>
    <style type="text/css">
      /* Slideshow styles */
    </style>
  </head>
  <body>
    *   <textarea id="source">
        <!-- Slideshow Markdown -->
      </textarea>
    *   <script src="remark.js">
        </script>
        <script>
    *     var slideshow = remark.create();
        </script>
      </body>
</html>
```

You may download remark to have your slideshow not depend on any online resources, or reference the **latest version** online directly.

Of course, Markdown can only go so far.

Markdown extensions

To help out with slide layout and formatting, a few Markdown extensions have been included:

- Slide properties, for naming, styling and templating slides
- Content classes, for styling specific content
- Syntax highlighting, supporting a range of languages

Markdown extensions

- Slide properties

Initial lines containing key-value pairs are extracted as slide properties:

```
name: agenda  
class: middle, center
```

```
# Agenda
```

```
The name of this slide is {{ name }}.
```

Slide properties serve multiple purposes:

- Naming and styling slides using properties `name` and `class`
- Using slides as templates using properties `template` and `layout`
- Expansion of `{{ property }}` expressions to property values

See the [complete list](#) of slide properties.

Markdown extensions

- Slide properties

- Content classes

Any occurrences of one or more dotted CSS class names followed by square brackets are replaced with the contents of the brackets with the specified classes applied:

```
.footnote[.red.bold[*] Important footnote]
```

Resulting HTML extract:

```
<span class="footnote">  
  <span class="red bold">*</span> Important footnote  
</span>
```

Markdown extensions

- Slide properties

- Content classes

- Syntax Highlighting

Code blocks can be syntax highlighted by specifying a language from the set of **supported languages**.

Using **GFM** fenced code blocks you can easily specify highlighting language:

```
```javascript
function add(a, b)
 return a + b
end
```
```

```
```ruby
def add(a, b)
 a + b
end
```
```

A number of highlighting **styles** are available, including several well-known themes from different editors and IDEs.

Presenter mode

To help out with giving presentations, a presenter mode comprising the following features is provided:

- Display of slide notes for the current slide, to help you remember key points
- Display of upcoming slide, to let you know what's coming
- Cloning of slideshow for viewing on extended display

Presenter mode

- Inline notes

Just like three dashes separate slides, three question marks separate slide content from slide notes:

```
Slide 1 content
```

```
*???
```

```
Slide 1 notes
```

```
- - -
```

```
Slide 2 content
```

```
*???
```

```
Slide 2 notes
```

Slide notes are also treated as Markdown, and will be converted in the same manner slide content is.

Pressing P will toggle presenter mode.

Presenter mode

- Inline notes

- Cloned view

Presenter mode of course makes no sense to the audience.

Creating a cloned view of your slideshow lets you:

- Move the cloned view to the extended display visible to the audience
- Put the original slideshow in presenter mode
- Navigate as usual, and the cloned view will automatically keep up with the original

Pressing C will open a cloned view of the current slideshow in a new browser window.

It's time to get started!

Getting started

Getting up and running is done in only a few steps:

1. Visit the [project site](#)
2. Follow the steps in the Getting Started section

For more information on using remark, please check out the [wiki](#) pages.

That's all folks (for now)!

Slideshow created using **remark**.