

Slideous

A Lightweight HTML Based Presentation Tool

--- press F11 Key for Full Screen mode and then Page Down or Return Key to proceed ---

Stefan Gössner (stefan@goessner.net)

Content

- [Content](#)
- [Slideous is ...](#)
- [Navigating Slideous](#)
- [Mouse Navigation](#)
- [Incremental Display](#)
- [Incremental Display - Example](#)
- [Create a Slideous Show](#)
- [Customising Slideous](#)
- [Configuration variables](#)
- [Configuration example](#)
- [Predefined Variables](#)
- [Statusbar](#)
- [Miscellaneous](#)

Slideous is ...

- a web based presentation application.
- based on (X)HTML, CSS and Javascript
- easily customisable.
- lightweight.
- inspired by [S5](#) and [Html Slidy](#).

Navigating Slideous

Full screen

F11 key

Next Slide

Page down key, Return key

Previous Slide

Page up key

Next Item

Single mouse click, Right Arrow key, Space bar

Previous Item

Left Arrow key

First Slide

Home key

Last Slide

End key

Font size

Use +/- key to increase/decrease font size

Miscellaneous

C-key: navigates to Table of Content

M-key: Mouse navigation on/off

P-key: Print mode / Show all Slides

S-key: Statusbar off/on

Mouse Navigation

Slide navigation via mouse click can be enabled/disabled temporarily with the *M*-key and permanently by setting the `{$mousesensible}` variable. (See [Configuration variables](#)).

In order to allow selecting content during the presentation, please note some special aspects with Slideous' mouse navigation:

- If the cursor position when **pressing** the mouse button **differs** from its position when **releasing** the button, you might have selected some content and no navigation is performed.
- If the cursor position on button press and release is **unchanged** and **some content** was selected, the content is deselected and no navigation is performed.
- If the cursor position on button press and release is **unchanged** and **no content** was selected, a navigation to the next item/slide is performed.

Incremental Display

Slides can be defined to use incremental display in three different ways

- setting the class attribute *"incremental"* for the elements

blockquote, dd, dt, h1 - h6, li, p, pre, img, object, table, td, th, tr

- setting the class attribute *"incremental"* for any container element to affect its immediate child elements above.
- setting the `{ $\$autoincrement$ }` variable locally or globally to `true` affects the elements

dl, ol, ul

You can modify these element lists in the configuration section of *slideous.js*.

To see an example press Space Bar to proceed to next slide.

Incremental Display - Example

Slideous uses ...

- HTML
- CSS
- Javascript

The first two acronyms mean

HTML

Hypertext Markup Language

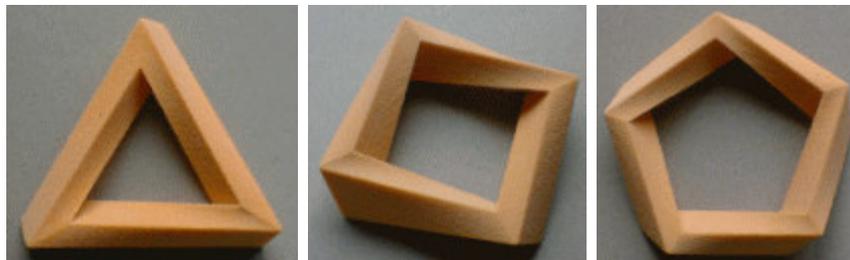
CSS

Cascading Style Sheets

The keystroke navigation follows the rules

Navigation Key	current slide is complete	current slide is incomplete
Tab, Right Arrow	next slide	next item
Page Down	next slide	show current slide completely

Incremental display with images behaves as expected



Penrose models from [Technion, Israel Institute of Technology](#), with permission.
Thanks to (c) [Gershon Elber](#) for the graphics.

See the handout for this example markup (*P-key*)

```

<p>Slideous uses ...</p>
<ul class="incremental">
  <li class="u">HTML</li>
  <li class="u">CSS</li>
  <li class="u">Javascript</li>
</ul>
<p class="incremental">The first two acronyms mean</p>
<dl class="incremental">
  <dt>HTML</dt><dd>Hypertext Markup Language</dd>
  <dt>CSS</dt><dd>Cascading Style Sheets</dd>
</dl>
<p class="incremental">The keystroke navigation follows the rules</p>
<div style="text-align:center;">
  <table class="incremental">
    <tr class="evn">
      <td><strong>Navigation Key</strong></td>
      <td><strong>current slide is complete</strong></td>
      <td><strong>current slide is incomplete</strong></td>
    </tr>
    <tr class="odd">
      <td class="lft">Tab, Right Arrow </td>
      <td class="lft">next slide </td><td class="lft">next item </td>
    </tr>
    <tr class="evn incremental">
      <td class="lft">Page Down </td>
      <td class="lft">next slide </td>
      <td class="lft">show current slide completely </td>
    </tr>
  </table>
</div>
<p class="incremental">Incremental display with images behaves as exp
<div class="incremental" style="text-align:center;">
  
  
  
</div>
<p class="incremental" style="text-align:center;font-size:75%;">
  Penrose models from [...]
</p>

```

Create a Slideous Show

- Create a single HTML or XHTML file using the template below.
- Reference the *slideous.css* and *slideous.js* files properly.
- Append an arbitrary number of `<div>`'s for each slide.
- Insert a `<h1>` heading and arbitrary HTML markup into every slide section. ...

Template

```
<html>
  <head>
    <title>A Slideous Show</title>
    <link rel="stylesheet" type="text/css" href="slideous.css" />
    <script src="slideous.js" type="text/javascript"> </script>
  </head>
  <body>
    <div id="statusbar">
      <!-- statusbar content -->
    </div>
    <div class="slide">
      <h1>First Slide</h1>
      <!-- slide content -->
    </div>
    <!-- more slides -->
  </body>
</html>
```

Customising Slideous

You can customise Slideous in several different ways.

1. **globally** by

- creating your individual HTML template file.
- modifying/extending the minimal CSS file *slideous.css*. Please note, that it includes two sections:
 - *@media screen, projection*
 - *@media print*
- modifying the configuration section of the script file *slideous.js*.

2. **locally** by

- adding meta elements to the template's *<head>* section.

Configuration variables

Whenever the string ***{\$name}*** appears in the rest of the document, it is substituted by the corresponding ***[value]*** of an existing local or global variable with that name.

- Global variables are defined as static javascript variables of the form `Slideous.varname.`
- Local variables are declared by a set of `<meta>` elements as *name/value* pairs.

```
<meta name="[name]" content="[value]" />
```

Local variables defined by `<meta>` elements are of higher priority and thus overwrite global script variables.

Configuration example

The following example illustrates the usage of configuration variables:

slideous.js

```
var Slideous = {  
  ...  
  company: "Virtual Inc.",  
  ...  
};
```

html file

```
...  
<title>Slide Show</title>  
<meta name="author" content="Jon Doe" />  
<meta name="date" content="2005-11-04" />  
...  
<div class="slide">  
  <h1>{$title}</h1>  
  <h3>{$author}, {$company}</h3>  
  <h4>{$date}</h4>  
</div>  
...
```

is resulting in ...

```
...  
<div class="slide">  
  <h1>Slide Show</h1>  
  <h3>Jon Doe, Virtual Inc.</h3>  
  <h4>2005-11-04</h4>  
</div>  
...
```

Predefined Variables

Slideous comes with a set of predefined variables, some of which you might use or change to another value within your presentation. Please note, that you are not restricted to them. In fact you can define arbitrary additional variables.

variable	type	meta	js	description
\$autoincrement	boolean	yes	yes	when true, use incremental display for certain elements, even when not set via <i>class</i> -attribute
\$fontsize	number	yes	yes	initial font-size (percent)
\$fontdelta	number	yes	yes	value for increment/decrement font-size (percent)
\$mousesensitive	boolean	yes	yes	enable forward navigation via mouse click
\$generator	string	no	no	"Slideous"
\$slidecount	number	no	no	number of slides
\$statusbar	boolean	yes	yes	visibility of statusbar
\$title	string	no	no	presentation title, taken from the documents <i><title></i> element
\$tocid	string	yes	yes	add automatically created table of contents to container element with this <i>id</i> -attribute. Use "" to suppress a ToC.
\$version	string	no	no	Version of <i>Slideous</i>

Statusbar



Slideous comes with a simple predefined status bar. It is visible in *single slide mode* only, always located at the bottom of the browser window.

The statusbar code reads:

```
<div id="statusbar">
  <span style="float:right;">
    <button id="contentbutton" title="content">C</button>
    <button id="homebutton" title="first slide">1</button>
    <button id="prevslidebutton" title="previous slide">&laquo;</button>
    <button id="previtembutton" title="previous item">&lsaquo;</button>
    <button id="nextitembutton" title="next item">&rsaquo;</button>
    <button id="nextslidebutton" title="next slide">&raquo;</button>
    <button id="endbutton" title="last slide">{$slidecount}</button>
    <button id="incfontbutton" title="content">A+</button>
    <button id="decfontbutton" title="first slide">A-</button>
  </span>
  <span id="eos">&frac{1}{2}</span>
  <span title="{ $location }, { $date }">{$title}, {$author}</span>
  <span style="position:absolute;left:45%;">
    <span id="slideidx"></span> of {$slidecount}
  </span>
</div>
```

Feel free to design your own statusbar.

- There are two elements with special meaning. Slideous identifies them by their *id*-attributes.
 1. The current slide index is automatically injected into an element with `id="slideidx"`, whenever the slide is changing. (see *2 of 7* in the above example statusbar).
 2. When the current slide is incomplete, i.e. there are still some invisible items available for incremental display, then an element with `(xml) id="eos"` is shown; otherwise hidden. (see $\frac{1}{2}$ in the above statusbar)).
- When using pretty images as navigation buttons, be aware to add the corresponding *id*-attributes to them.

Miscellaneous

Table of Contents

It is easy to let *Slideous* automatically create a ToC. Simply set the value of *\$tocid* to the *id*-attribute of an empty container element anywhere in your document. Slideous will insert a ToC item for each slide at startup. The ToC item's markup is defined by variable *Slideous.tocitem*. With that you can either create an unordered list of links in a ToC slide or build a drop down list in the status bar.

Print a Handout

- Add `<div class="handout">...</div>` sections for additional content visible only in the handout document.
- Eventually customise the *@media print* section in *slideous.css*.
- Change the View to Print View (*P*-key).
- Use the browsers *Print* or *Print Preview* command to send the document to the your printer. ([See this print result](#)).

Internal Links

Use the markup `...` for linking from one slide to another. Please note, that the browser's back button does not work as expected with internal slide links.

The Slideous Files

Get them [here](#).

Acknowledgment

Many thanks to [Eric Meyer](#) and [Dave Raggett](#) for their useful and inspiring work.

License

This work is licensed under [Creative Commons GNU LGPL License](#).

Contact

If you want to ask a question, give suggestions or simply have some comments please send them to [my weblog](#) or contact me directly under <mailto:stefan@goessner.net>.