$LectureDoc^2$ Tutorial

LectureDoc is an authoring system for creating lecture material; i. e., lecture slides, notes and exercises. LectureDoc enables you to write a single (HTML or) reStructuredText document that contains the slides, additional annotations and also exercises. Using LectureDoc's module system even more advanced use cases, such as integrated quizzes, are possible.

This tutorial is written in reStructuredText and can be used as a template for creating your own lecture slides. The *code* of this tutorial is available on GitHub: Delors/reStructuredTextToLectureDoc2/main/ld_base_example.rst

Prof. Dr. Michael Eichberg

DHBW
Duale Hochschule
Baden-Württemberg
Mannheim

Version: 1.0

Basics

A basic slide consists of a (section) header and some reStructuredText content.

Example

```
Basics
-----------

A basic slide consists of a (section) header
and some reStructuredText content.
```

Embedding Formulae

Embed math equations using reStructuredText's default directive (`.. math::`) and role (`:math:`...``).

Example

The following code:

```
Computation in :math:`GF(2)`:


   math

  \begin{matrix}
  1 + 1 & = 1 - 1 & = 0 \\
  1 + 0 & = 1 - 0 & = 1 \\
  0 + 1 & = 0 - 1 & = 1
  \end{matrix}
```

Will render like this:

Computation in $GF(2)$:

$$\begin{matrix} 1 + 1 & = 1 - 1 & = 0 \\ 1 + 0 & = 1 - 0 & = 1 \\ 0 + 1 & = 0 - 1 & = 1 \end{matrix}$$

A slide without an explicit title can be created by explicitly creating an empty title.

Example

```
\
--
```

Alternatively and also recommended, you can use the following class: `no-title` in combination with the `class` directive:

Note

You have to add a space after the backslash (\)!

Example

```
   class   no-title

I will only show up in an index...
------------------------------------
```

Animation

Basic *appear* animations can be created using the (CSS) class `incremental`[1]. You can also define a corresponding custom role (`.. role:: incremental`) to animate parts of a text.

Example

```
Animation
----------

Basic *appear* animations can be created using the (CSS) class
``incremental``. You can also define a corresponding custom role
(``.. role:: incremental``) :incremental:`to animate parts of a text.`

   admonition   Example
    :class: incremental

    ...
```

[1] Animation progress can be reset by pressing the `r` key.

Animation of Lists

In case of (un-)ordered lists (`ol` or `ul` in HTML) it is sufficient to associate the class `incremental` using the `class` directive with the list. It is also possible, to only specify the class attribute for the required list items.

Example

The following code:

```
   class   incremental


- this
- is
- a test
```

Will render incrementally like this:

- this
- is
- a test

Slide Dimensions

The slide dimensions can be controlled by specifying the corresponding meta information. If not specified, the dimension is set to 1920×1200 (default); i.e., a ratio of 16:10.

Example

In HTML documents add at the following meta tag:

```
meta name="slide-dimensions" content="1600x1200"
```

In reStructuredText documents add at the beginning:

```
   meta
    :slide-dimensions: 1600x1200
```

Associating a slide set with a unique id

Many functions in LectureDoc2 - e.g. persistence of the slide progress - require that a slide set is associated with a unique id. This id can be set using the meta directive.

Example

```
meta
 :id: lecturedoc2-tutorial
 :description: LectureDoc2 Tutorial
 :author: Michael Eichberg
 :license: Released under the terms of the `2-Clause BSD license`.
```

Adding Supplemental Information

Adding information that should not be on the slides, but provide additional information/explanations, can be added using the `supplemental` directive.

Example

```
supplemental

 **Formatting Slides**

 Formatting slides is done using classes and roles.
```

Alternatively, a container with the class `supplemental` can also be used:

Example

```
supplemental

 **Formatting Slides**
```

Formatting Slides

Creating heavily formatted slides is easily possible using rst directives and roles which are mapped to CSS classes.

1. Structuring Documents

Creating Sections

Creating a slide which marks the beginning of a new section can be done using the `new–section` class.

Example

```
   class   new-section

Structuring Documents
---------------------


   class   new-subsection

Creating Sections
-----------------
```

Slide Transitions

Slide transitions can be controlled using the `transition-...` classes:

- `transition-fade`
- `transition-move-left`
- `transition-move-to-top`
- `transition-scale`

Example

```
   class   transition-move-to-top

Slide Transitions
------------------
```

See the LectureDoc2 Cheat Sheet for a comprehensive list of predefined transitions.

Adding Code

Adding code can be done using reStructuredText's code directive.

Example

The following code:

```
   code    python

    for i in range 0 10
        print i
```

Will render like this:

```
for i in range 0 10
    print i
```

Links to External Resources

LectureDoc2 supports links to external resources:

- https://github.com/Delors/LectureDoc2
- LectureDoc2 Sourcecode

Example

```
LectureDoc2 supports links to external resources:

- https://github.com/Delors/LectureDoc2
- `LectureDoc2 Sourcecode <https://github.com/Delors/LectureDoc2>`_
```

Links to Internal Targets

LectureDoc2 supports links to external resources:

- The title of a slide can be used as a link target: Advanced Formatting
- An element which is explicitly marked as a target can be used as a link target:

  Link Target in Incremental Block

Example

Slide with explicit marked-up element:

```
Advanced Formatting
---------------------

   container   incremental

       _Link Target:

     See the LectureDoc2 Cheat Sheet.
```

References are defined as follows:

```
Links to internal targets:

- Link to slide: `Advanced Formatting`_
- Link to a marked-up element:

  `Link Target`_
```

Scientific Citations

Citations are fully supported in LectureDoc2.

A reference to a book: [Martin2017] (Details are found in the bibliography (see next slide)).

Example

```
A reference to a book: [Martin2017]_
```

Bibliography

- [Martin2017] Clean Architecture: A Craftsman's Guide to Software Structure and Design; Robert C. Martin, Addison-Wesley, 2017
- ...

Example

```
    [Martin2017] Clean Architecture: ...; Robert C. Martin, Addison-Wesley, 2017
```

Advanced Formatting

LectureDoc comes with a set of predefined (CSS) classes that can be used to format the slides. Some of these classes have explicit support by LectureDoc and will be rendered differently in the different situations (e.g., document view vs. slide view will render *stacked layouts* or *supplemental information* differently).

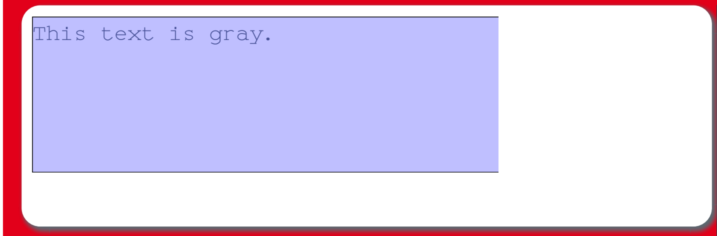
- dhbw-red
- minor
- ~~obsolete~~

See the LectureDoc2 Cheat Sheet for a comprehensive list of predefined CSS classes.

Stacked layouts

Stacked layouts enables updating parts of a slide by putting the content into layers and then showing the layers incrementally.

Example



This text is gray.

```
  stack   monospaced

    layer

   :dhbw-gray:`This text is gray.`

    layer   incremental overlay

      raw   html

      <svg width="600" height="200">
        <rect width="800" height="200"
          style="fill:rgb(0,0,255,0.25);
      ⇥ ⇥ ⇥ ⇥ ⇥ ⇥stroke-width:1;
      ⇥ ⇥ ⇥ ⇥ ⇥ ⇥stroke:rgb(0,0,0)" />
      </svg>
```

Integrated Exercises

Exercises can be integrated into the slide set.

Example

Exercise: 1+1

Compute: $\sqrt{2} = ?$

To unlock the solution go to the document view and enter the password.

```
 exercise   Exercise: 1+1

  Compute: :math:`\sqrt 2 = ?`.

     solution
      :pwd: sqrt

      Solution: :math:`1,4142135624`.
```

If you have multiple exercises, you can define a master password to unlock all solutions at once (press `m` to open the dialog).

```
.. meta::
    :exercises-master-password: 123456
```

Exercise: 1+1

Compute: $\sqrt{2} = ?$

2. Images

ld_base_example/tag_cloud.png

Image in the Background

Example

```
   class   padding-none no-title transition-scale

Image in the Background
------------------------

   rubric   Image in the Background

   stack   monospaced padding-none margin-none

       layer   padding-none margin-none

           image   ld_base_example/tag_cloud.png
            :width: 100%
            :align: center

       layer   overlay

          Content on the slide...
```