

# LectureDoc2 Cheat Sheet

## Information

Author: Michael Eichberg  
Version: August 2024

## SVGs

SVGs can easily be embedded and animated.

### Examples

## Vertical Titles

Add the class `vertical-title` to rotate the title and to change the layout of the slide to a column-based layout. To get back to a row based layout add a container with the class `width-100`.

### Examples

## Slide Tweaks

### Slide without Title

To hide the title of a slide assign the class `no-title`.

### Vertically Centered Content

Use the class `center-content`.

### Examples

## Sections and Subsections

Create a slide that marks the beginning of a new section or subsection by adding the class `new-section` or `new-subsection` to the slide.

## Exercises and Solutions

Associate the class `exercises` with a slide to indicate that the slide contains an exercise.

Use the directive `... exercise::` to add an exercise. To add a solution use the custom directive `... solution::` inside of an exercise block and specify the password (optional) using the attribute `:pwd::`.

### Example (Solution in supplemental information)

```
... exercise:: <title>
```

<Description of the exercise.>

```
... solution:: <title>  
:pwd: 1234
```

<Solution>

You can configure a master-password using meta information:

```
... meta::  
:master-password: <Master-password>
```

## Table of Contents

A navigable table of contents can be created using standard `rst` techniques.

### Example

## Table of Contents

---

- `Section 1 Title`\_
- `Subsection 1.1 Title`\_

## Footnotes

[#]\_ and . . . [#]\_ create footnotes.

Test\ [#]\_

---

... [#]\_ `test.org`\_

## Explicit Footers

A container with the class `footer-left`, `footer-right` or `block-footer`.

## References

Use standard rst references.

## Example

...  
Like described in [Eic24]\_ ...  
...

## References

---

... [Eic24]\_ LectureDoc2; 2024

## Copy to Clipboard

Add `copy-to-clipboard` to a code block to enable copying code to the clipboard.

## Example

... `code`:: java  
:class: copy-to-clipboard  
  
public static void main(...)

## Fade-out Content

Add the class `fade-out` to a container to whiten the content.

## Supplemental Information

Use the directive `supplemental` for respective information.

## Example

... `supplemental`::  
  
<Text>

## Text Alignment

Control text alignment: `text-align-[left|center|right]`

## Images

Adding a drop-shadow and rounded corners: `picture`.

## Tables

The layout can be adapted using: `compact`, `compact-cells`, `no-table-borders`, `no-inner-borders`,

no-column-borders, fake-header [-2nd]-row and fake-header [-2nd]-column.

## Animation

incremental (and wobble).

highlight-line-on-hover (always usable), highlight-on-hover (explicit column or row headers are not supported) or highlight-identical-cells

## Lists

list-with-explanations renders text paragraphs of list items less pronounced.

(As shown here.)

Use impressive to make the list more impressive:

Add negative-list to use "!" for bullet points.

Add positive-list to use "✓" for bullet points.

## Example

- Point 1

.. **class**:: negative-list list-with-explanations

- Point 2

Some on-slide explanation.

.. **class**:: positive-list

- Point 3

## Decorations

line-above draws a horizontal lines.

box-shadow adds a shadow.

rounded-corners the corners will be rounded.

## Example

```
.. container:: margin-top-1em
  line-above
  padding-top-1em
  box-shadow
```

Text

## Font Styling

**"rem" based relative sizes:** xxl, huge, large, small, footnotesize, scriptsize, tiny, x-tiny, xx-tiny

**"em" based relative sizes:** larger, smaller, far-smaller

**Font weight:** bold, light, thin

**Font family:** monospaced, serif

**Font style:** italic

## Slide Transitions

Available slide transitions: transition-move-left, transition-scale, transition-fade, transition-move-to-top

## Example

.. **class**:: transition-move-left

<Slide Title>

## -----

## Revealing Slide Content

All elements with the class `incremental` are revealed incrementally.

### Example

```
... class:: incremental  
- Item 1 - Part 1  
:incremental: Item 1 - Part 2  
- Item 2
```

## Column-based Layouts

Use two-columns and three-columns for respective layouts.

### Example

```
... container:: two-columns  
  ... container:: column no-separator  
    <Column 1>  
  ... container:: column  
    <Column 2>
```

Add `no-default-width` to the root container for content based column widths. Use class `no-separator` on the left column to remove the separator.

## Stacked Layouts

Stacked layouts are based on nested layers. Each layer - except of the first one - needs to have the class `incremental` and/or the class `overlay` for transparent layers. (Up to 10 layers are supported.) To turn off the numbering of opaque layers use `.no-number`.

## Images in Stacked Layouts

To avoid that a parent element of a floating element is collapsed add the class `clearfix` to the parent element; i. e., when a layer just contains a floating image.

### Example

```
... deck::  
  ... card:: clearfix  
    ... image:: <p1.svg>  
      :align: left  
    ... card:: overlay  
      ... image:: <p2.svg>  
    ... card:: warning  
      <Content>
```

## Semantic-based Text Markup

`peripheral`: for less important text. `obsolete`: for obsolete statements. `ger`: to markup German Words. `eng`: to markup English words.

## Box sizes

Use `width-100%` and `width-75%` to control the width of a container.

## Colors (roles)

### Font Colors

`red`, `gray`, `light-gray`, `blue`, `green`, `orange`, `black`, `shiny-green`, `shiny-red`, `dark-red`

### Background Colors

`red-background`, `dhw-gray-background`, `dhw-light-gray-background`, `white-background`, `blue-background`, `light-green-background`, `green-background`, `yellow-background`

## Example

```
:dhw-red: Red Text.
```

### Fine-grained Control (Try to avoid!)

```
margin-none, margin-0-5em, margin-1em, margin-top-1em, margin-top-2em, margin-bottom-1em,  
margin-bottom-2em, margin-right-1em, margin-left-1em, padding-none, padding-0-5em,  
padding-1em, padding-top-1em, padding-top-2em
```

### Hiding slides (⚠ rst2ld only)

Use `hide-slide` to exempt it from slide generation.

## Example

```
... class:: hide-slide
```

```
<Hidden Slide >
```

## Configuration

LectureDoc meta information:

`id` The unique identifier for the slide set. Required to store the current state of the presentation.

`slide-dimensions` the slides dimension (default: "1920x1200").

`first-slide` Determines the first slide that is shown (e.g., <Slide Number> or "last-viewed").

## Example

```
... meta::  
:id: <unique id>  
:slide-dimensions: 2560x1440  
:first-slide: last-viewed
```

## Cheat Sheets with LD<sup>2</sup>

A cheat-sheet is a slide with the class `cheat-sheet-8-columns`.

## Template

```
... class:: cheat-sheet-8-columns  
  
<Title>  
-----  
  
... container:: cheat-sheet-block  
  
... rubric:: <TOPIC>  
  
... rubric:: <SUB-TOPIC>
```

## Useful Role and Substitution Definitions

## Template

```
... role:: incremental
... role:: eng
... role:: ger
... role:: peripheral
... role:: obsolete
... role:: red
... role:: gray
... role:: light-gray
... role:: blue
... role:: green
... role:: orange
... role:: shiny-green
... role:: shiny-red
... role:: dark-red
... role:: black

... role:: raw-html(raw)
:format: html
```

## Links

[DocUtils \(rst reStructuredText\)](#)

[Example Slide Sets](#)

## Useful Role and Substitution Definitions

### Template

```
... role:: incremental
... role:: eng
... role:: ger
... role:: peripheral
... role:: obsolete
... role:: red
... role:: gray
... role:: light-gray
... role:: blue
... role:: green
... role:: orange
... role:: shiny-green
... role:: shiny-red
... role:: dark-red
... role:: black

... role:: raw-html(raw)
:format: html
```

## Useful Role and Substitution Definitions

### Template

```
... role:: incremental
... role:: eng
... role:: ger
... role:: peripheral
... role:: obsolete
... role:: red
... role:: gray
... role:: light-gray
... role:: blue
... role:: green
... role:: orange
... role:: shiny-green
... role:: shiny-red
... role:: dark-red
```

```
.. role:: black  
  
.. role:: raw-html(raw)  
    :format: html
```

## Useful Role and Substitution Definitions

### Template

```
.. role:: incremental  
.. role:: eng  
.. role:: ger  
.. role:: peripheral  
.. role:: obsolete  
.. role:: red  
.. role:: gray  
.. role:: light-gray  
.. role:: blue  
.. role:: green  
.. role:: orange  
.. role:: shiny-green  
.. role:: shiny-red  
.. role:: dark-red  
.. role:: black  
  
.. role:: raw-html(raw)  
    :format: html
```

## Useful Role and Substitution Definitions

### Template

```
.. role:: incremental  
.. role:: eng  
.. role:: ger  
.. role:: peripheral  
.. role:: obsolete  
.. role:: red  
.. role:: gray  
.. role:: light-gray  
.. role:: blue  
.. role:: green  
.. role:: orange  
.. role:: shiny-green  
.. role:: shiny-red  
.. role:: dark-red  
.. role:: black  
  
.. role:: raw-html(raw)  
    :format: html
```

## Useful Role and Substitution Definitions

### Template

```
.. role:: incremental  
.. role:: eng  
.. role:: ger  
.. role:: peripheral  
.. role:: obsolete  
.. role:: red  
.. role:: gray  
.. role:: light-gray  
.. role:: blue  
.. role:: green  
.. role:: orange  
.. role:: shiny-green  
.. role:: shiny-red
```

```
.. role:: dark-red
.. role:: black

.. role:: raw-html(raw)
:format: html
```

## Useful Role and Substitution Definitions

### Template

```
.. role:: incremental
.. role:: eng
.. role:: ger
.. role:: peripheral
.. role:: obsolete
.. role:: red
.. role:: gray
.. role:: light-gray
.. role:: blue
.. role:: green
.. role:: orange
.. role:: shiny-green
.. role:: shiny-red
.. role:: dark-red
.. role:: black

.. role:: raw-html(raw)
:format: html
```