

# Lecture Doc 2

LectureDoc 2 is an authoring system for lecture slides which makes use of modern HTML, CSS and JavaScript. Lecture Doc is also very well suited to create normal presentations; its main target is however the creation of lecture slides. LectureDoc makes it trivial to embed math and source code by relying on established projects (e.g. MathJax and HighlightJS).

# How To

A Lecture Doc document is basically a plain HTML5 document, which has a very simple structure. All functionality is enabled using CSS and JavaScript. The most basic presentation would be:

```
<!DOCTYPE html>
<html lang="en">

<head>
  <meta charset="UTF-8">
  <meta name="viewport"
    content="width=device-width, initial-scale=1.0, user-scalable=no">
  <link rel="stylesheet" href="ld/normalize.css" type="text/css">
  <link rel="stylesheet" href="ld/ld.css" type="text/css">
  <script src="ld/ld-lib.js"></script>
  <script src="ld/ld-help.js"></script>
  <script src="ld/ld-animation.js"></script>
  <script src="ld/ld-core.js"></script>
</head>

<body>
  <div class="ld-slide">
    <strong>Title of your presentation.</strong>
  </div>
  <div class="ld-slide">
    <h1>A Slide!</h1>
    <p>
      Lorem ipsum dolor sit amet consectetur adipisicing elit.
      Esse asperiores eos facilis quod, veritatis blanditiis
      aut delectus doloremque minima voluptate id ipsa sapiente.
    </p>
  </div>
  <div class="ld-slide">
    <div style="height: 100%;
      width: 100%;
      background-color: black;
      color: white ;">
      Final page.
    </div>
  </div>
  <div class="ld-slide">
    The following list is shown incrementally.
    <ul class="incremental">
      <li>This</li>
      <li>is it</li>
      <li>- a Test</li>
    </ul>
  </div>
</body>
</html>
```

```
        </ul>
    </div>
</body>

</html>
```

As seen in the above example, two stylesheets related to the rendering of the controls need to be imported and some JavaScript files which enable the base functionality. Support for math equations and syntax highlighting of code needs additional imports. See the advanced example for that.

The body of the HTML document should have only slides as child elements. The slides are div elements with the class **ld-slide**. When LectureDoc is initialized further **divs** will automatically be created for the control elements.

In general, we highly recommend to author your slides in reStructuredText and use **rst2ld** to create your slide set. **rst2ld** offers a wide range of features that makes using LectureDoc fun to use

# Controlling the initial appearance

- When you want to show the light table directly when opening the presentation add the following meta tag to the head of the document.

```
<meta name="ld-show-light-table" content="true">
```

- To start the presentation with a different slide than the first one, use the following meta tag.

```
<meta name="first-slide" content="5">
```

- if content is an int then the corresponding slide will be shown. (The first slide has the value 1).
- if content is *"last"* the last slide will be shown.
- if content is *"last-viewed"* the last viewed slide will be shown. Uses the browser's local storage for storing the slide number; may not work in all situations. Requires that the document has a unique id. The id can be set using: `<meta name="id" content="(unique id)">`.

# Slide Design

In general, no hard restrictions have to be followed regarding the design of your slides. However, the width, height, position, display and scale properties of slides (div.id-slide) are used by LectureDoc and must not be "styled" in custom style sheets.

# Non-Goals

## **Broad compatibility**

Lecture Doc does not strive for maximum compatibility with all (past) browsers. I.e., it is only regularly tested on the most modern versions of Chrome, Safari and Firefox as of 2023. In general, LectureDoc will not use features not fully supported by one of these browsers. Hence, in practice only mature features are going to be used. However, feel free to open a pull-request if something can be improved without introducing strong dependencies on specific browsers or adding compatibility layers with old browsers.