

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-sc
```

```
  <link rel="stylesheet" href="ld/normalize.css" type="text/css
```

```
  <link rel="stylesheet" href="ld/ld.css" type="text/css">
```

```
  <script src="ld/ld-help.js"></script>
```

```
  <script src="ld/ld-core.js"></script>
```

Advanced features

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. If less than

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.