

LectureDoc2

LectureDoc2 is an authoring system for lecture material which makes use of modern HTML, CSS and JavaScript. LectureDoc is also very well suited to create normal presentations; its main target is however the creation of lecture material.

LectureDoc makes it trivial to embed math and source code by relying on established projects (e.g. MathJax and HighlightJS).

Using LectureDoc

LectureDoc documents have to be served by a web server, because it makes use of modern JavaScript features which are not available when opening the document directly in a browser due to security reasons.

Starting a simple webserver, which is sufficient for most use cases, can be done using Python 3:

```
python3 -m http.server -d <PATH TO ROOT FOLDER>
```

Alternatively, the node.js **http-server** (from npm) can be used:

```
http-server -p 8888 -c-1 -g --cors="Access-Control-Allow-Origin:*" --no-dotfiles <PATH/TO/ROOT/FOLDER>
```

How To

A LectureDoc document is basically a plain HTML5 document with 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">
  <script src="../ld.js" type="module"></script>
  <link rel="stylesheet" href="../ld.css" />
</head>

<body>
  <template>
    <ld-topic>
      <strong>Title of your presentation.</strong>
    </ld-topic>
    <ld-topic>
      <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>
    </ld-topic>
    <ld-topic>
      <p>
        The following list is shown incrementally.
      </p>
      <ul class="incremental-list">
        <li>This</li>
        <li>is it</li>
        <li>- a Test</li>
      </ul>
    </ld-topic>
    <ld-topic>
      <div style="height: 100%;
        width: 100%;
        background-color: black;
        color: white ;">
        Final page.
      </div>
    </ld-topic>
  </template>
</body>

</html>
```

As seen in the above example, a stylesheet related to the rendering of the controls need to be imported along with 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 one template element with topics as child elements. The slides and the document view is created based on the ld-topic elements. When LectureDoc is initialized, the user-interface related elements will automatically be created.

In general, we highly recommend to author your slides in reStructuredText and use `rst2ld` to create

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 value 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. ld-slide`) are used by LectureDoc and must not be "styled" in custom style sheets.

Non-Goals

Broad compatibility

LectureDoc does not strive for maximum compatibility with all (past) browsers. I.e., as of 2024, it is only regularly tested on the most modern versions of Chrome, Safari and Firefox. 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.