

How to create presentations with emacs-reveal *

Jens Lechtenbörger

December 2021 (emacs-reveal 9.1.1 and later)

Presentation Hints

General

- This is a reveal.js presentation and an Open Educational Resource (OER)
 - Generated with emacs-reveal from Free/libre Org mode sources
 - * See [usage hints for emacs-reveal presentations](#)
 - * Emacs-reveal 8.0.0 supports reveal.js 4
 - Key bindings and navigation
 - * Press “?” to see key bindings of reveal.js
 - In general, “n” and “p” move to next and previous slide; mouse wheel works as well
 - Search with Ctrl-Shift-F
 - * Up/down (swiping, arrows) move within sections, left/right jump between sections (type “o” to see what is where)
 - * Type slide’s number followed by Enter to jump to that slide
 - * Browser history (buttons, Alt-CursorLeft, Alt-CursorRight)
 - * Zoom with Ctrl-Mouse or Alt-Mouse

Why?

- I created emacs-reveal as software bundle to produce Open Educational Resources (OER) for my own teaching
 - Described in (Lechtenbörger 2019a)
 - Personally, I prefer text over video when learning
 - * Skim reading with superior search, navigation, and hyperlinks; own speed
 - * Lots of students like audio explanations (and PDF), though
- Education should be free and open

*This PDF document is an inferior version of an OER HTML page; free/libre Org mode source repository.

- Recording of a talk “Open Educational Resources: What, why, and how?”
- Proper license attribution is a hassle
 - * Emacs-reveal simplifies that process (for me), see (Lechtenböcker 2019b)

Offline work

- Students often ask for download-able presentations
- Alternatives
 1. Clone repository, build presentations locally (see Usage)
 2. Download build artifacts from recent pipeline (if not expired)
 3. Generate PDF
 - Why, really?
 - * Why not download source files instead?
 - * Org mode, which is plain text
 - Change the URL by adding “?print-pdf” after “.html”, then print to PDF file (usually, Ctrl-p)
 - Alternatively, generate PDF via L^AT_EX from Org source file
 - * Replace `.html` (and whatever follows) in address bar of browser with `.pdf`
 - E.g., this howto as PDF

Audio

- Audio should start automatically here (differently from emacs-reveal’s default)
 - Enthusiast by Tours
 - * Licensed under Creative Commons Attribution 3.0 Unported (CC BY 3.0)
 - * Converted to free Ogg format with Audacity
 - See compatibility and known issues of the underlying audio plugin
 - * Firefox, which I recommend as browser in general (here in English and here in German), seems to work everywhere
 - Audio controls are shown at bottom left

(Speaker) Notes

- Slides contain additional notes as plain text if you see the folder icon at the top right (as on this slide)



Figure 1: Figure under CC0 1.0

- Either press “v” to see the “courseware view” or click on that icon or press “s” to see the “speaker notes view”
- You need to allow pop-ups
 - * If the pop-up window does not work, you may need to press “s” twice or close the pop-up window once

These are sample notes

- Lists can be used here
- You can time your presentation
 - Maybe look at one of my talks to see how to define timing

Introduction

What’s This?

- Emacs-reveal is free software to generate reveal.js presentations (slides with audio) from simple text files in Org mode



Figure 2: “Figure” under CC0 1.0; converted from Pixabay

- Benefits
 - * For your audience
 - Self-contained presentations embedding audio
 - Usable on lots of (including mobile and offline) devices with just a browser
 - * For you as producer
 - Separation of layout and contents (similarly to, e.g., \LaTeX)
 - Simple text format allows diff and merge for ease of collaboration

Prerequisites

- I suppose (and strongly recommend) that you use GNU/Linux ([help on getting started](#))
 - Actually, not much here is operating system specific
- `emacs-reveal` should really be used with the text editor GNU Emacs
 - (You could try other editors and build presentations within GitLab, thanks to GitLab's infrastructure)
 - * (In fact, you do not need an editor at all but could edit presentations using a Web browser on [GitLab.com](#), e.g., with the [Web IDE](#) (requires login))

Installation and Quickstart

- `emacs-reveal` builds upon Gnu Emacs with Org mode
 - `emacs-reveal` is available as free software on [GitLab](#)
- You also need Git
 - [Getting started](#)
 - * The [Pro Git](#) book is a great source in general
 - [Git introduction as OER](#) (created with `emacs-reveal`)
- And maybe more, see next slide

\LaTeX and other dependencies

- By default, `emacs-reveal` generates HTML presentations and PDF variants
 - PDF output requires a \LaTeX installation
 - * If missing, `elisp/publish.el` stops with an error, resulting in **broken** presentations
 - Add following to beginning of `elisp/publish.el` to generate only HTML


```
(setq oer-reveal-publish-org-publishing-functions
      '(oer-reveal-publish-to-reveal))
```

- This howto also contains a `DOT/Graphviz` example
 - Dependencies of `emacs-reveal` specified in two Docker files
 - * `docker/debian-emacs-tex/Dockerfile`
 - * `docker/emacs-reveal/Dockerfile`

Initial Consideration

- `Emacs-reveal` can manage bundled software
 - (Submodules for Lisp packages `Org` mode, `org-re-reveal`, `org-re-reveal-citeproc`, `org-re-reveal-ref`, `oer-reveal` as well as `reveal.js` with several plugins)
 - Default with customizable variable `emacs-reveal-managed-install-p` being `t`
 - Variable `oer-reveal-revealjs-version` specifies the target version of `reveal.js` for `emacs-reveal`
- Or, you manage those components yourself
 - Set `emacs-reveal-managed-install-p` to `nil`
- In any case, `emacs-reveal` changes values of other packages (`org-ref`, `oer-reveal`) without warning

Managed install of `emacs-reveal`

- Install `emacs-reveal` in a directory of your choice
 1. Choose directory, e.g., `~/emacs.d/elpa`, and clone software
 - `cd ~/emacs.d/elpa`
 - `git clone --recursive https://gitlab.com/oer/emacs-reveal.git`
* (Option `--recursive` downloads submodules)
 2. Add following lines to `~/emacs`
 - `(add-to-list 'load-path "~/emacs.d/elpa/emacs-reveal")`
 - `(require 'emacs-reveal)`
 3. Restart Emacs (installation of `org-ref` or `citeproc` is offered, if necessary)

Alternative installation

- You may prefer to manage submodules of `emacs-reveal` yourself
 1. Choose directory and clone (without option `--recursive`)
 - `cd ~/emacs.d/elpa`
 - `git clone https://gitlab.com/oer/emacs-reveal.git`
 2. Add following lines to `~/emacs`
 - `(add-to-list 'load-path "~/emacs.d/elpa/emacs-reveal")`
 - `(setq emacs-reveal-managed-install-p nil)`

- * Read doc string of `emacs-reveal-managed-install-p`
- `(require 'emacs-reveal)`
- 3. (Now, subdirectories under "`~/ .emacs.d/elpa/emacs-reveal`" remain empty)

Quickstart with `emacs-reveal`

- E.g., generate this howto
 1. Install `emacs-reveal` (see previous two slides for alternatives)
 2. Choose directory for howto, clone it
 - `git clone --recursive https://gitlab.com/oer/emacs-reveal-howto.git`
 - * Option `--recursive` gets an embedded repository for figures
 - `cd emacs-reveal-howto/`
 3. Generate the HTML presentation from Org source `howto.org`
 - `emacs --batch --load elisp/publish.el`
 - Publication code needs to be able to locate `emacs-reveal.el`
 - * Code in `elisp/publish.el` tries (a) `~/ .emacs.d/elpa/emacs-reveal` (suggested on earlier slide) and (b) sibling directory `emacs-reveal`

Usage

Alternatives

1. Create presentations locally on Command Line
2. Create presentations in GNU Emacs
3. Create presentations with Docker



Figure 3: “Docker logo” under Docker Brand Guidelines; from Docker

- Docker image `emacs-reveal`
 - Similarly to previous alternative; necessary software bundled
 - See [README](#) of `emacs-reveal`
 - [Introduction to Docker](#), built with `emacs-reveal`
- 4. Create and publish presentations on GitLab



Figure 4: “GitLab Logo” by GitLab under CC BY-NC-SA 4.0; from gitlab.com

- Based on [GitLab Continuous Integration](#) infrastructure and above Docker image

Build Presentations on Command Line

0. Install `emacs-reveal` and `howto`
1. Create Org file in directory `emacs-reveal-howto`
 - See contained source file for this presentation, `howto.org`
2. Build presentations for files ending in `.org`
 - (Except internal ones, see function `oer-reveal-publish-all`)
 - `emacs --batch --load elisp/publish.el`
 - Presentations are built in subdirectory `public/`
3. Open presentation in Firefox
 - E.g.: `firefox public/howto.html`
4. Optional: Copy `public/` to public web server

Building Presentations in Emacs

1. Generate HTML presentation for visited `.org` file using Org export functionality: Press `C-c C-e w b` (export with `oer-reveal`)
 - This generates HTML file in current directory and opens it in default browser
 - For this to work
 - (a) Settings of `emacs-reveal` should be in effect (`emacs-reveal.el` is loaded, e.g., with step (2) above)
 - (b) Necessary resources, in particular `reveal.js`, must be accessible in `.org` file's directory
 - I use `emacs --batch --load elisp/publish.el` once to populate `public/`, then create a symbolic link:
`ln -s public/reveal.js`
 - (c) For image grids, you may need: (`setq oer-reveal-export-dir "./"`)

Org-re-reveal and oer-reveal

- Emacs-reveal embeds the packages `org-re-reveal` and `oer-reveal`
 - Package `oer-reveal` is an Org mode export backend (extending `org-re-reveal`)
 - * Starting with `oer-reveal 1.4.0`, part of `emacs-reveal 4.1.0`
 - * With key binding mentioned on previous slide
 - You can export with `org-re-reveal` (`C-c C-e v v` and `C-c C-e v b`) or `oer-reveal` (`C-c C-e w w` and `C-c C-e w b`)
 - * With `oer-reveal`, additional `reveal.js` plugins are enabled by default
 - See customizable variables `oer-reveal-plugins` and `oer-reveal-plugin-config`

Build Presentations on GitLab

1. Fork `emacs-reveal-howto` on GitLab (fork [documentation](#))
 - `git clone <the URL of YOUR GitLab project>`
2. Create or update Org files in cloned directory
 - Push changes to your fork
3. GitLab infrastructure picks up changes and publishes presentations as [GitLab Pages](#)
 - Based on Continuous Integration (CI) infrastructure
 - Configured by file `.gitlab-ci.yml`
 - CI run takes some minutes
 - Go to Settings → Pages to see the Pages' address

Some Presentation Features

Text Slide

- A list
- With a sub-list whose items appear
 - This is *emphasized*
 - This is **bold**
 - This looks like `code`
 - This is **green**
 - Nothing special

Some Fragment Styles

- Forget
- Shrink
- Grow
- Very important

Fragments with Custom Order

- I'm first.
- Fourth.
- Third.
- Second.
- I'm also first.

Centered Text

Just some horizontally centered text. Created by assigning class `org-center` (for which `oer-reveal.css` specifies `text-align: center`).

Alternatively, Org's `center` blocks are exported by plain HTML export, see `org-html-center-block`.

On Sections

- This slide is part of section `Some Presentation Features`
 - We can link to slides, e.g., an earlier slide
 - * You can use the browser history to go back
 - Side note: Check source code to see two variants of link targets used on this slide
- This slide can also be perceived as its own subsection
 - The next slide is on a deeper level of nesting
- (This list item appears simultaneously with previous bullet point)

Another Slide

- This slide is on a deeper level of nesting
- This level of nesting is not shown in the table of contents in the slide's bottom
- By the way, the headings in the table of contents below are hyperlinks
 - And your browser remembers the history, back/forward buttons and shortcuts should work
 - Mousewheel and swiping work

Licensing

- Starting with `emacs-reveal 5.0.3` (and `oer-reveal 2.0.2`), presentations can show license information derived from `SPDX` headers of the `REUSE` project
 - See licensing slide at the end of this presentation
 - * Information on that slide is derived from header lines of `howto.org`

```
#+SPDX-FileCopyrightText: 2017-2020 Jens Lechtenbörger <https://lechten.gitlab.com>
#+SPDX-License-Identifier: CC-BY-SA-4.0
```
 - * Note that `SPDX` headers must be prefixed with `#+` to be recognized as Org mode keywords
 - License information is also embedded in machine-readable `RDFa` format
- Macros for OER figures with (human- and machine-readable) license information are discussed later

Two Columns: Pro/Con of emacs-reveal

Pro

- Free/libre open source software
- Device-independent presentations

Con

- Also mobile and offline
 - Generated from simple text format
 - * Easy to learn
 - * Collaboration with diff/merge/git
 - * Separation of layout and content
- No WYSIWYG
 - (Need to learn something new)

Hyperlinks

- Different types of hyperlinks exist
 - External ones
 - * Plain Org mode link
 - Or with emphasis that you should really check out Org mode before you continue
 - * Details of Docker are beyond the scope of this howto
 - Internal ones (within presentation)
 - * Maybe pointing to an earlier slide
 - * Or pointing to a later slide
 - * Or emphasizing that a mentioned concept like figures and audio is revisited later

URL Parameters

- See usage hints for emacs-reveal presentations, e.g.:
 - `./howto.html?default-navigation` switches to the default navigation mode of reveal.js
 - `./howto.html?hidelinks=32` hides hyperlinks that go beyond presentation topics
 - * (Note the link for navigation modes of reveal.js above)
 - Or both: `./howto.html?default-navigation&hidelinks=32`

Figures and Audio

- The following figures and their license metadata are maintained in a separate project

- Embedded here as Git submodule
- See [source file](#) for use of macros `reveallicense`, `revealing`, `revealgrid`
 - * Macros defined and documented in [config.org](#) of `oer-reveal`
- Presentation contains license information in machine-readable RDFa format ([Lechtenbörger 2019b](#))

Slide with Figure and Audio

- This figure is part of a different presentation **Warning!** Figure omitted as gif format **not** supported in \LaTeX : “Animation of Clock algorithm for page replacement”
(See [HTML presentation](#) instead.)
 - Notice: No license displayed for figure → License of document applies
- The song [Enthusiast by Tours](#) is licensed under [Creative Commons Attribution 3.0 Unported \(CC BY 3.0\)](#)

Figure with Caption and License

- Display image with meta-data specified in file
 - Simplify sharing of images with source and license
- Functionality and meta-data format are specific to `emacs-reveal`
 - See next slide for sample file



Figure 5: To share or not to share (“Figure” under [CC0 1.0](#); converted from [Pixabay](#))

Meta-Data File for Previous Image

```
;; Semicolon starts comment until end of line (Emacs Lisp).
;; Note that the line for dc:title below is just a comment. In that
;; case, "Image" is used as generic title; uncomment for real title.
;; CC0 does not require attribution of author/creator; uncomment if needed.

((filename . "./figures/3d-man/decision-1013751_1920.jpg") ; Note the path prefix
; (dc:title . "The title given by the author")
  (licenseurl . "https://creativecommons.org/publicdomain/zero/1.0/")
  (licensetext . "CC0 1.0")
; (cc:attributionName . "Jens Lechtenbörger")
; (cc:attributionURL . "https://lechten.gitlab.io/#me")
  (dc:source . "https://pixabay.com/en/decision-question-response-1013751/")
  (sourcetext . "Pixabay")
  (imgalt . "Balance tipping in favor of Yes")
  (imgadapted . "converted from") ; Adjust as needed
  (texwidth . 0.5) ; Width in percent of textwidth for LaTeX export
)
```

An Image Grid: Computers

Presentation contains image grid. L^AT_EX export not supported.

Creation of Previous Image Grid

- Single line in source file, using macro `revealgrid`

```
{{{revealgrid(42,"./figures/devices/computer.grid",60,4,3,"\"ga1 ga2 ga2 ga3\" \"ga1
```

- Arguments explained in config.org of `oer-reveal`
- With file `computer.grid` as follows

```
("./figures/devices/white-male-1834091_1920.meta"
"./figures/devices/commodore-160186_1280-CC0.meta"
"./figures/devices/laptop-154091_1280.meta"
"./figures/devices/router-157597_1280.meta"
"./figures/devices/car-49278_960_720.meta"
"./figures/devices/beauty-1260974_1920.meta"
"./figures/devices/vintage-tv-1116587_960_720.meta"
"./figures/devices/smartwatch-1874536_1280.meta"
"./figures/devices/Fairphone_2_reverse.meta")
```

Notes on figures

- If you used `emacs-reveal` previously and did not like that it exported all figures from a growing repository, note that as of `emacs-reveal 5.2.0`, only used figures are exported
- So far, `emacs-reveal` uses meta-data in an ad-hoc format (as shown on a previous slide)

- For all figures in this repository
- Please, contact me if you'd like to contribute with a different format, e.g., JSON-LD
 - * Maybe with an issue?

Appearing Items with Audio

(Audios produced with [MaryTTS](#), converted to Ogg format with [Audacity](#))

- One
- Two
- Three

Misc

Quiz Plugin

- Emacs-reveal embeds this quiz plugin
 - Demo of plugin's author
- In presentations, quizzes support active learning
 - In particular, retrieval practice

Sample Quiz

Klipse for Code Evaluation

- Org-re-reveal supports Klipse
 - Teach programming
 - * Code changes in upper part result in output changes in lower part
 - Browser-side code evaluation for various programming languages
 - * See [org-re-reveal-klipse-languages](#) for supported subset
 - clojure, html, javascript, js, php, python, ruby, scheme, sql
 - * To activate, either add option `reveal_klipsify_src:t` (as in header of this file) or set variable `org-re-reveal-klipsify-src` to `t`; be sure to disable scaling of `reveal.js`
 - * Correct indentation may require that you set `org-src-preserve-indentation` to `t` (see bottom of this file)
- Code on next two slides copied from [README](#) of Org-Reveal

HTML Src Block

```
<h1 class="whatever">hello, what's your name</h1>
```

Javascript Src Block

```
console.log("success");
var x='string using single quote';
x
```

PHP Src Block

```
$name = "Alice";
echo "Moin " . $name . "!";
```

Python Src Block

```
def factorial(n):
    if n < 2:
        return 1
    else:
        return n * factorial(n - 1)

print(factorial(10))
```

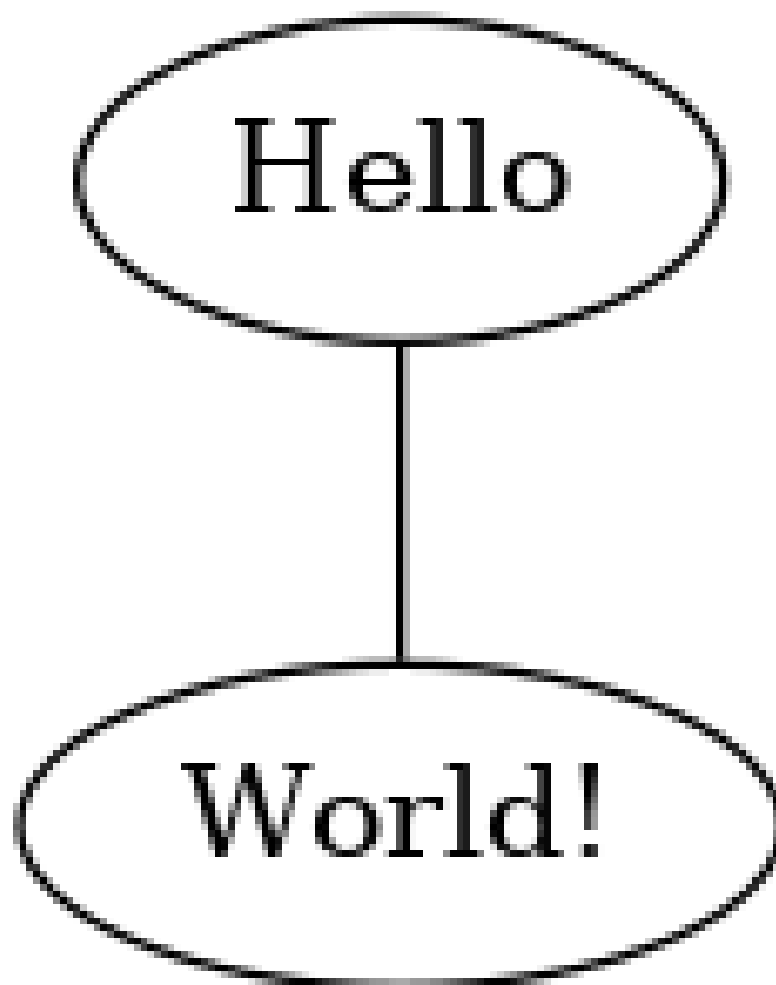
Figures with Babel

- Org export can execute embedded source code, with results injected into exported HTML presentation
 - For example, diagrams generated with dot of Graphviz
- With emacs-reveal
 - Activate necessary source languages in `oer-reveal-publish-babel-languages`
 - Maybe generate figures into separate directory
 - * Publish contents with `org-publish-project-alist`
- See subsequent slides for sample code

Hello World with Dot

```
graph {
    hello [label="Hello"];
    world [label="World!"];

    hello -- world;
}
```



Relevant Excerpt of Publication Code

- The following snippet of `elisp/publish.el` activates `dot` and publication of generated images

– Adapt based on your needs

* Note that necessary directories must exist (Babel does not create them)

```
(make-directory "img" t)
(setq oer-reveal-publish-babel-languages '((dot . t) (emacs-lisp . t))
      org-publish-project-alist
      (list (list "img"
                  :base-directory "img"
                  :base-extension "png"
                  :publishing-function 'org-publish-attachment
                  :publishing-directory "./public/img"))))
```

Need Additional Software in Publication Process?

- Maybe suggest as issue for Docker image of emacs-reveal
- Or install additional software in Docker container of your project with `before_script`

The End

Further Reading

- Quickstart for Org mode
- Presentations for a course on Operating Systems
 - My first use case for emacs-reveal
 - More features of Org mode (e.g., table of contents as agenda, keyword index) and reveal.js (e.g., notes, animated SVGs)

Go for it!



Figure 6: The road ahead . . . (“Figure” under CC0 1.0; converted from Pixabay)

<https://gitlab.com/oer/>

Bibliography

- Lechtenbörger, Jens. 2019a. “Emacs-reveal: A software bundle to create OER presentations.” *Journal of Open Source Education (JOSE)* 2 (18). <https://doi.org/10.21105/jose.00050>.
- . 2019b. “Simplifying license attribution for OER with emacs-reveal.” In *17. Fachtagung Bildungstechnologien (DELFI 2019)*, edited by Niels Pinkwart and Johannes Konert, 205–16. Bonn: Gesellschaft für Informatik e.V. https://doi.org/10.18420/delfi2019_280.

License Information

Except where otherwise noted, the work “How to create presentations with emacs-reveal”, © 2017-2022 Jens Lechtenbörger, is published under the Creative Commons license CC BY-SA 4.0.

No warranties are given. The license may not give you all of the permissions necessary for your intended use.

In particular, trademark rights are *not* licensed under this license. Thus, rights concerning third party logos (e.g., on the title slide) and other (trade-) marks (e.g., “Creative Commons” itself) remain with their respective holders.