OER GitHub Tutorial - Template

Sophie Matter

Create and publish OER with GitHub

We believe that open educational resources can be created using tools the open source community has been successfully using for many years now. This tutorial guides you through the creation of open educational resources (OER) with GitHub. Using our GitHub template, you can easily create and publish your own OER in just a few minutes.

What's in this tutorial:

After completing this tutorial, you will have an automatically generated OER with your own content and metadata, published on

> GitHub for free and ready to be put into our OER search index OERSI.

Quick Start

> If you want to get started really quickly, you can follow this quick start tutorial and create a simple OER within just a few minutes.

1. Create a GitHub account, if you don't have one yet

> As we will host our OER on GitHub, a free GitHub account is required.

If you do not have an account yet, go to GitHub's sign up page and create a free account: https://github.com/signup.

Afterwards, log into your account.

2. Create your project

> Create the place where your OER lies.

Go to the Markdown Documents Template and click on the green **Use this template** button, then on **Create a new repository**. This generated your own project (called `repository`). Assign the repository's owner to yourself (or to a group of your choice, if you are a member of one) and give it a short, but meaningful name that describes your OER. This name will also be the used for the URL to your OER. Optionally add a description. Make sure that the repository is set to **public**. Lastly, confirm by clicking on **Create repository from template**.

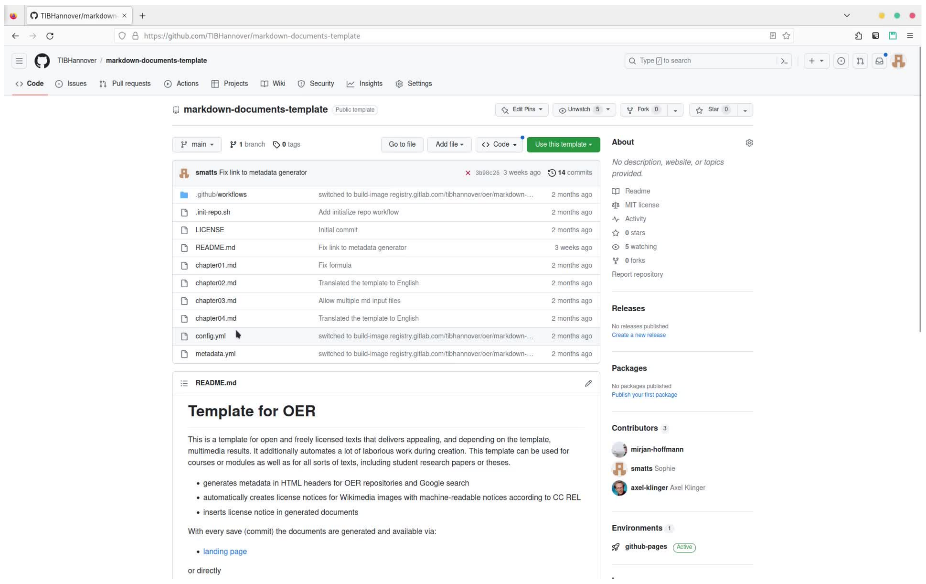
3. Fill the project with your content

> Now that you have your own project/repository, you can fill it with your own content.

Currently, you have the template's dummy data in your repository. You can edit a file by clicking on the file and then on the edit button. The files you should edit are:

- `chapter01.md`
- `chapter02.md`
- `chapter03.md`
- `chapter04.md`

You can start by edititing `chapter01.md`. Click on the file, the in the top right corner, click on the pen symbol ("Edit this file"). You can delete the file's content and start writing something of your own.



markdown-documents-template Public template

smatts Fix link to metadata generator ✕ 3b98c26 3 weeks ago 14 commits

.github/workflows	switched to build-image registry.gitlab.com/tibhannover/oer/markdown-…	2 months ago
.init-repo.sh	Add initialize repo workflow	2 months ago
LICENSE	Initial commit	2 months ago
README.md	Fix link to metadata generator	3 weeks ago
chapter01.md	Fix formula	2 months ago
chapter02.md	Translated the template to English	2 months ago
chapter03.md	Allow multiple md input files	2 months ago
chapter04.md	Translated the template to English	2 months ago
config.yml	switched to build-image registry.gitlab.com/tibhannover/oer/markdown-…	2 months ago
metadata.yml	switched to build-image registry.gitlab.com/tibhannover/oer/markdown-…	2 months ago

README.md

Template for OER

This is a template for open and freely licensed texts that delivers appealing, and depending on the template, multimedia results. It additionally automates a lot of laborious work during creation. This template can be used for courses or modules as well as for all sorts of texts, including student research papers or theses.

- generates metadata in HTML headers for OER repositories and Google search
- automatically creates license notices for Wikimedia images with machine-readable notices according to CC REL
- inserts license notice in generated documents

With every save (commit) the documents are generated and available via:

- landing page

or directly

About

No description, website, or topics provided.

Readme
MIT license
Activity
0 stars
5 watching
0 forks
Report repository

Releases

No releases published
Create a new release

Packages

No packages published
Publish your first package

Contributors 3

mirjan-hoffmann
smatts Sophie
axel-klinger Axel Klinger

Environments 1

github-pages Active

Figure 1: Video preview: "Create repository from template (videos/create-from-template.mp4)" (not available in PDF version). Click to view the video online.

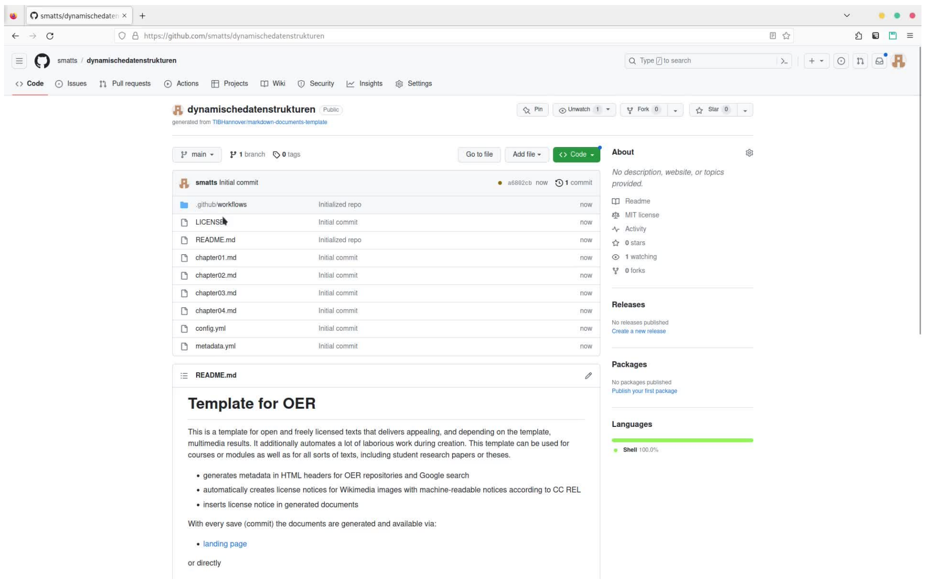
Now, in the top right corner, click on the green "Commit changes…" button. Write a short message on what you have changed ("Commit message"), if you want a longer description, and now click on the green "Commit changes" button.

You can also choose to create or delete files. If you do not want the four chapter files that were created by the template, you can click on the file, click on the three dots on the top right of the file, and then on "Delete file". To upload files, you can click on the "Add file" button in your repository.

4. Generate the OER

> The automatic generation will take your content and generate different output formats.

Lastly, enable the automatic generation of your OER. To do this, go to the project's `Settings -> Pages` and in `Build and Development` set the source to `GitHub Actions`. After this, you can head to the `Actions` tab and click on the newest workflow run. If the worklow already ran, you will find that it failed. This happened because the Pages were not enabled yet. In this case, click on re-run jobs. Otherwise, wait until the jobs have finished. The generated documents are now created.


smatts/dynamischedatenstrukturen

https://github.com/smatts/dynamischedatenstrukturen

smatts / dynamischedatenstrukturen

Code · Issues · Pull requests · Actions · Projects · Wiki · Security · Insights · Settings

dynamischedatenstrukturen Public

generated from TIBHannover/markdown-documents-template

main · 1 branch · 0 tags · Go to file · Add file · Code

smatts Initial commit · a6802cb now · 1 commit

.github/workflows	Initialized repo	now
LICENSE	Initial commit	now
README.md	Initialized repo	now
chapter01.md	Initial commit	now
chapter02.md	Initial commit	now
chapter03.md	Initial commit	now
chapter04.md	Initial commit	now
config.yml	Initial commit	now
metadata.yml	Initial commit	now

About

No description, website, or topics provided.

Readme · MIT license · Activity · 0 stars · 1 watching · 0 forks

Releases

No releases published · Create a new release

Packages

No packages published · Publish your first package

Languages

Shell 100.0%

README.md

Template for OER

This is a template for open and freely licensed texts that delivers appealing, and depending on the template, multimedia results. It additionally automates a lot of laborious work during creation. This template can be used for courses or modules as well as for all sorts of texts, including student research papers or theses.

- generates metadata in HTML headers for OER repositories and Google search
- automatically creates license notices for Wikimedia images with machine-readable notices according to CC REL
- inserts license notice in generated documents

With every save (commit) the documents are generated and available via:

- landing page

or directly

Figure 2: Video preview: "Pages (videos/pages.mp4)" (not available in PDF version). Click to view the video online.

5. Add your metadata

Go to our Metadata Generator and insert the metadata that describes your OER. In the top right corner, you have the option to switch the language between **German** ("DE") and **English** ("EN").

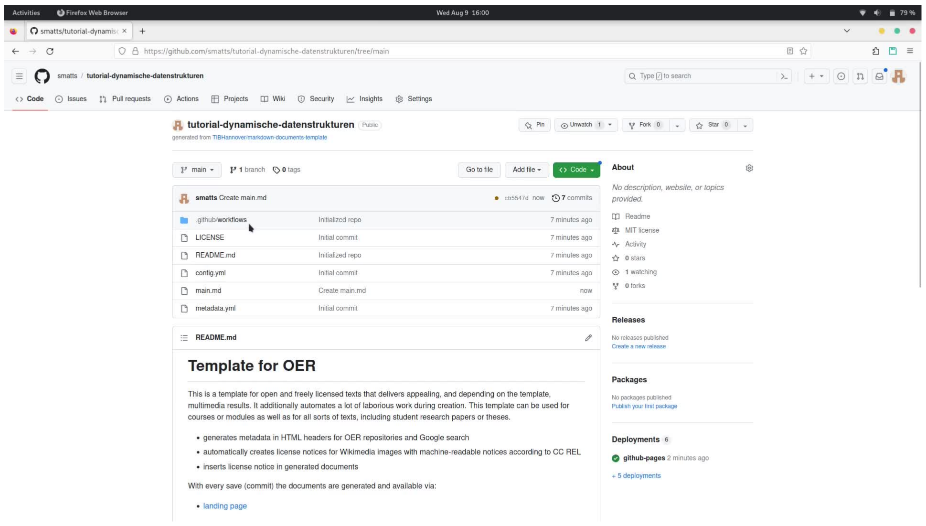
Once you are done, click on the "Generate" button in the bottom of the page. The metadata is now generated in a format our template understands. Now copy everything to your clipboard. For this, you can click on "Copy".

In your repository, click on the `metadata.yml` file and then on the pen symbol ("Edit this file") to edit the file. Delete the file's contents and paste the metadata from your clipboard. Now click on the green "Commit changes…" button and confirm with "Commit changes".

Now at the bottom of the page, you can click on `Generate`. This generates the metadata in the correct format. You can then copy the output to your clipboard either by using the `Copy` button, or by selecting the whole text (`Ctrl + A`) and copying it (`Ctrl + C`).

Done!

At the front page of your repository, inside the `README.md` content, there are several links you can use to view your generated documents. Click on the `landing page` link to view a page that lists metadata about your OER and



tutorial-dynamische-datenstrukturen Public

generated from TIBHannover/markdown-documents-template

smatts Create main.md · cb5547d now · 7 commits

.github/workflows	Initialized repo	7 minutes ago
LICENSE	Initial commit	7 minutes ago
README.md	Initialized repo	7 minutes ago
config.yml	Initial commit	7 minutes ago
main.md	Create main.md	now
metadata.yml	Initial commit	7 minutes ago

No description, website, or topics provided.

No releases published

No packages published

github-pages 2 minutes ago

README.md

Template for OER

This is a template for open and freely licensed texts that delivers appealing, and depending on the template, multimedia results. It additionally automates a lot of laborious work during creation. This template can be used for courses or modules as well as for all sorts of texts, including student research papers or theses.

- generates metadata in HTML headers for OER repositories and Google search
- automatically creates license notices for Wikimedia images with machine-readable notices according to CC REL
- inserts license notice in generated documents

With every save (commit) the documents are generated and available via:

- landing page

Figure 3: Video preview: "Add metadata (videos/metadata-placeholder.mp4)" (not available in PDF version). Click to view the video online.

supplies several links to different output formats (like a web version, a pdf version, . . .).

> To insert your OER into the OER search index oersi.org, head to the `About` section in the index of your repository, click on the settings symbol and add `open-educational-resources` to `Topics`. Your course will be indexed at night and appear on the next day.

Step by step tutorial

The step by step tutorial consists of the following steps, which you can click through either on the sidebar, using the arrows on the bottom or simply using the arrow keys on your keyboard.

- Requirements
- Create a project
- Generate Output
- Fill with your own content
- Add your metadata
- Configuration options
- Insert into OERSI

Requirements

> For this tutorial, **you will need a GitHub account**. It is also recommended to **get to know Markdown**, a markup language which

> can create formatted text using only plain-text, since Markdown is used to write and format your content.
>
> If you already have a GitHub account and know basic markdown syntax, you can skip this part and directly go do Create a project.

Create a GitHub account

If you don't have an account yet, go to GitHub and sign up. Confirm your e-mail address and log in.

Markdown

From Matt Cone's Markdown guide:

> Markdown is a lightweight markup language that you can use to add formatting elements to plaintext text documents. Created by John Gruber in 2004, Markdown is now one of the world's most popular markup languages.
>
> Using Markdown is different than using a WYSIWYG editor. In an application like Microsoft Word, you click buttons to format words and phrases, and the changes are visible immediately. Markdown isn't like that. When you create a Markdown-formatted file, you add Markdown syntax to the text to indicate which words and phrases should look different.
>
> For example, to denote a heading, you add a number sign before it (e.g., # Heading One). Or to make a phrase bold, you add two asterisks before and after it (e.g., **this text is bold**). It may take a while to get used to seeing Markdown syntax in your text, especially if you're accustomed to WYSIWYG applications.

If you want to learn about Markdown, what it is and what you can do with it, you can read this guide: Get Started.

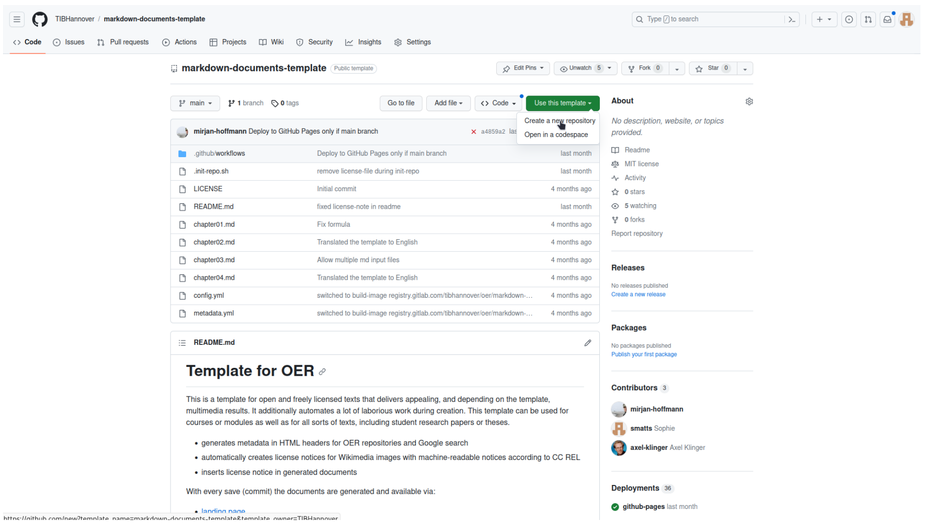
To see basic formatting options, check out the Cheat Sheet. More basic syntax can be found in the Basic Syntax guide.

Some of the basic Markdown syntax is shown in the reference part of this tutorial: Reference: Markdown.

Create a project

> The place where you will put your OER is called a `repository`. You can think of it as a project where all your files are located.

Once you are logged into GitHub, go to the TIB Hannover markdown documents template repository and click on "Use this template" `->` "Create a new repository".



mirjan-hoffmann Deploy to GitHub Pages only if main branch a4859a2

.github/workflows	Deploy to GitHub Pages only if main branch	last month
.init-repo.sh	remove license-file during init-repo	last month
LICENSE	Initial commit	4 months ago
README.md	fixed license-note in readme	last month
chapter01.md	Fix formula	4 months ago
chapter02.md	Translated the template to English	4 months ago
chapter03.md	Allow multiple md input files	4 months ago
chapter04.md	Translated the template to English	4 months ago
config.yml	switched to build-image registry.gitlab.com/tibhannover/oer/markdown-…	4 months ago
metadata.yml	switched to build-image registry.gitlab.com/tibhannover/oer/markdown-…	4 months ago

README.md

Template for OER

This is a template for open and freely licensed texts that delivers appealing, and depending on the template, multimedia results. It additionally automates a lot of laborious work during creation. This template can be used for courses or modules as well as for all sorts of texts, including student research papers or theses.

- generates metadata in HTML headers for OER repositories and Google search
- automatically creates license notices for Wikimedia images with machine-readable notices according to CC REL
- inserts license notice in generated documents

With every save (commit) the documents are generated and available via:

- landing page

https://github.com/new?template_name=markdown-documents-template&template_owner=TIBHannover

Figure 4: Create repository

Assign the repository to the correct owner and give it a short but meaningful name. The name will be the URL of the repository as well. You can add a description if you want. Now make sure that the visibility of the repository is set to **public**. This has two main reasons: firstly, so people can see your OER, and secondly to use the GitHub pages functionality which we will get to later. Lastly, confim by clicking on “Create repository from template”.

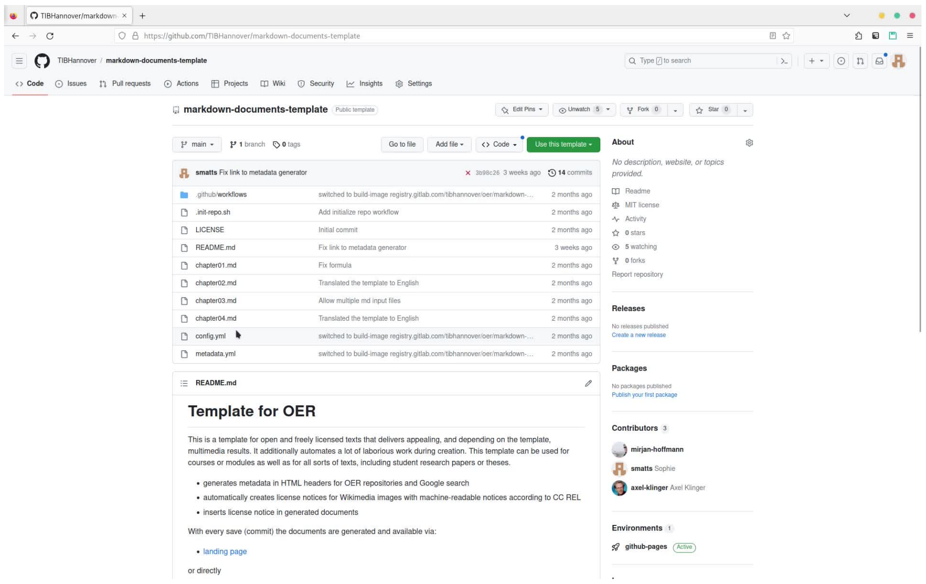
Now, a new repository is initialized for you containing the contents from the template.

Video tutorial

Generate output

> The automatic generation will take your content (everything inside the `Markdown` files, so those ending with `.md`) and generate different output formats. For example, these include a **web page** and a **pdf** version of your OER and are publicly accessible. This generation is done each time you change something in your repository, so your content will always be up to date.

In order to generate the different output formats for your OER, go to the project’s `Settings -> Pages` and in `Build and Development` set the source to `GitHub Actions`. After this, you can head to the `Actions` tab and click on the newest workflow run. If the worklow already ran, you will find that it failed. This happened because the Pages were not enabled yet. In this case, click on re-run jobs. Otherwise, wait until the jobs have finished. The pages and documents are


https://github.com/TIBHannover/markdown-documents-template

markdown-documents-template Public template

main · 1 branch · 0 tags

smatts Fix link to metadata generator · 3b98c26 3 weeks ago · 14 commits

.github/workflows	switched to build-image registry.gitlab.com/tibhannover/oer/markdown-…	2 months ago
.init-repo.sh	Add initialize repo workflow	2 months ago
LICENSE	Initial commit	2 months ago
README.md	Fix link to metadata generator	3 weeks ago
chapter01.md	Fix formula	2 months ago
chapter02.md	Translated the template to English	2 months ago
chapter03.md	Allow multiple md input files	2 months ago
chapter04.md	Translated the template to English	2 months ago
config.yml	switched to build-image registry.gitlab.com/tibhannover/oer/markdown-…	2 months ago
metadata.yml	switched to build-image registry.gitlab.com/tibhannover/oer/markdown-…	2 months ago

README.md

Template for OER

This is a template for open and freely licensed texts that delivers appealing, and depending on the template, multimedia results. It additionally automates a lot of laborious work during creation. This template can be used for courses or modules as well as for all sorts of texts, including student research papers or theses.

- generates metadata in HTML headers for OER repositories and Google search
- automatically creates license notices for Wikimedia images with machine-readable notices according to CC REL
- inserts license notice in generated documents

With every save (commit) the documents are generated and available via:

- landing page

or directly

About

No description, website, or topics provided.

Readme · MIT license · Activity · 0 stars · 5 watching · 0 forks · Report repository

Releases: No releases published. Create a new release

Packages: No packages published. Publish your first package

Contributors 3: mirjan-hoffmann; smatts Sophie; axel-klinger Axel Klinger

Environments 1: github-pages Active

Figure 5: Video preview: "Tutorial as video (videos/create-from-template.mp4)" (not available in PDF version). Click to view the video online.

now created. They can be accessed by the link that appeared under the `deploy` step.

Congratulations, you now have a complete course/document that you can make your own!

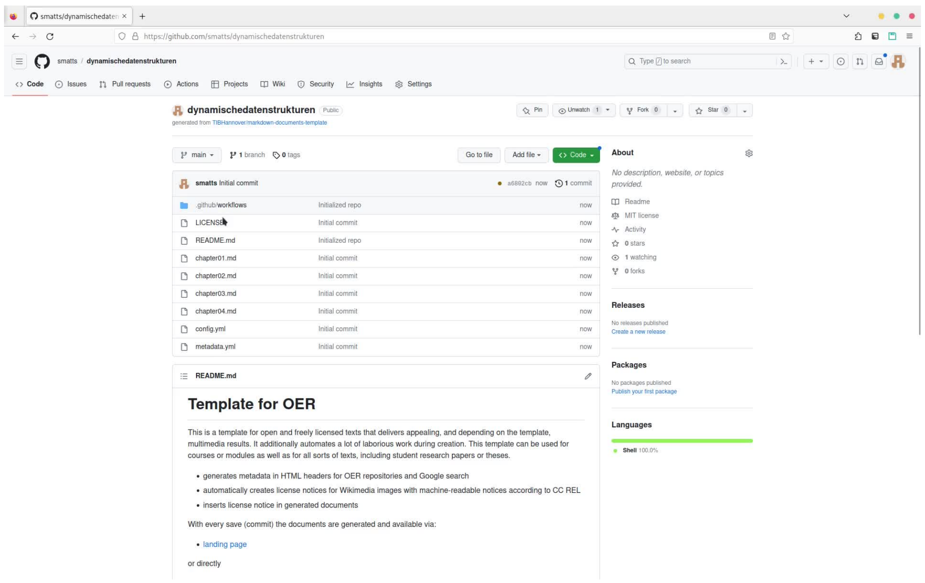
What are Actions?

> An `Action` is something that GitHub can execute with every commit (save) you make to the repository. This includes the generation of documents or pages, which we are using in this tutorial.
>
> A more detailed explanation of GitHub Actions can be found here: Understanding GitHub Actions.

What are Pages?

> A `Page` is a public website hosted by GitHub. We can use GitHub Actions to automatically generate our website and then host it on GitHub Pages for everyone to see. You can either host your site for free on `your-username.github.io` or use your own costum domain.
>
> More information on GitHub Pages can be found here: About GitHub Pages.

Video tutorial



dynamischedatenstrukturen Public
generated from TIBHannover/markdown-documents-template

1 branch 0 tags

smatts Initial commit a6802cb now 1 commit

.github/workflows	Initialized repo	now
LICENSE	Initial commit	now
README.md	Initialized repo	now
chapter01.md	Initial commit	now
chapter02.md	Initial commit	now
chapter03.md	Initial commit	now
chapter04.md	Initial commit	now
config.yml	Initial commit	now
metadata.yml	Initial commit	now

README.md

Template for OER

This is a template for open and freely licensed texts that delivers appealing, and depending on the template, multimedia results. It additionally automates a lot of laborious work during creation. This template can be used for courses or modules as well as for all sorts of texts, including student research papers or theses.

- generates metadata in HTML headers for OER repositories and Google search
- automatically creates license notices for Wikimedia images with machine-readable notices according to CC REL
- inserts license notice in generated documents

With every save (commit) the documents are generated and available via:

- landing page

or directly

About

No description, website, or topics provided.

Readme
MIT license
Activity
0 stars
1 watching
0 forks

Releases

No releases published
Create a new release

Packages

No packages published
Publish your first package

Languages

Shell 100.0%

Figure 6: Video preview: “Actions and Pages (videos/pages.mp4)” (not available in PDF version). Click to view the video online.

Fill with content

> Currently, there is still dummy data inside your repository. The next step is to replace this dummy data with your actual OER.

How and where to upload your own content

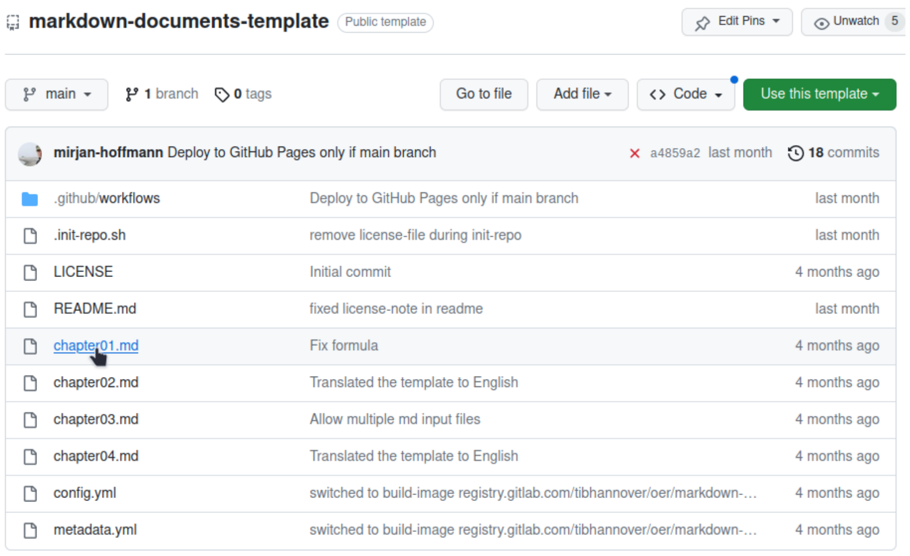
In the repository, you can find four `chapterXX.md` files. These are our dummy content files. You can either edit or delete them. You can of course create new files, too. If you create new files, make sure that they end with the correct file format ending `.md`, so `filename.md`.

To edit a file, click on the file name in the index of your repository.

Then, in the top right corner of the file, click on the pen button to edit the file.

Now, you can replace the content with your own. Add as much as you like. The file is a `Markdown` file, so you should write your content using this markup language. You can find more about Markdown in the Markdown section in the reference part of this tutorial.

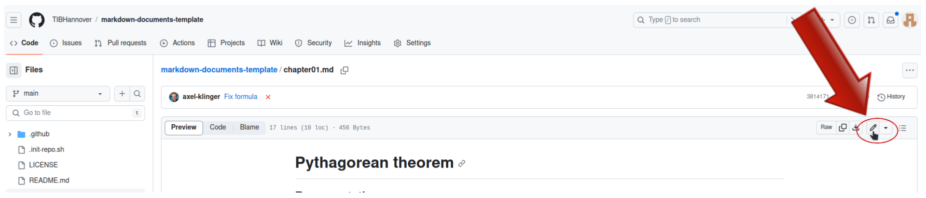
Once you are done, you can save your changes by clicking on the green `Commit changes...` button in the top right corner.



mirjan-hoffmann Deploy to GitHub Pages only if main branch ✕ a4859a2 last month 18 commits

.github/workflows	Deploy to GitHub Pages only if main branch	last month
.init-repo.sh	remove license-file during init-repo	last month
LICENSE	Initial commit	4 months ago
README.md	fixed license-note in readme	last month
chapter01.md	Fix formula	4 months ago
chapter02.md	Translated the template to English	4 months ago
chapter03.md	Allow multiple md input files	4 months ago
chapter04.md	Translated the template to English	4 months ago
config.yml	switched to build-image registry.gitlab.com/tibhannover/oer/markdown-…	4 months ago
metadata.yml	switched to build-image registry.gitlab.com/tibhannover/oer/markdown-…	4 months ago

Figure 7: Change content

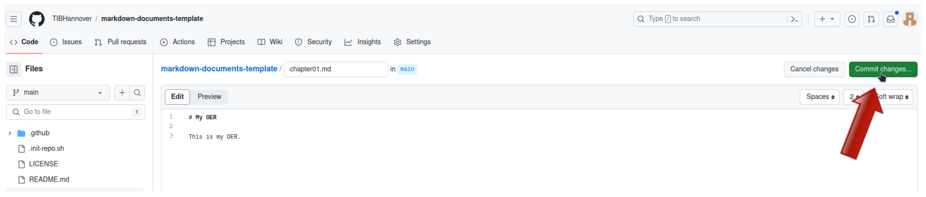


axel-klinger Fix formula ✕ 3814171

17 lines (10 loc) · 456 Bytes

Pythagorean theorem

Figure 8: Change content

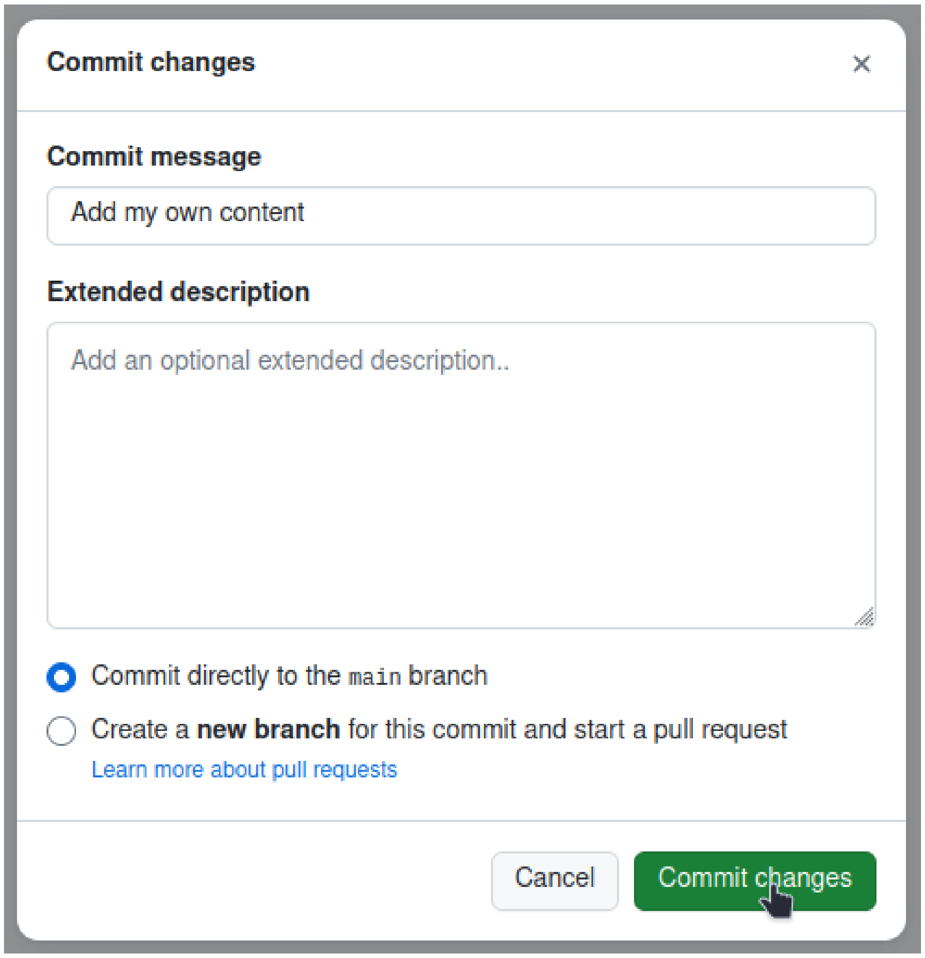


```
# My OER

This is my OER.
```

Figure 9: Change content

Add a meaningful but short commit message, which is a message describing the changes you have made, and optionally a longer description. Then confirm by clicking the green `Commit changes` button.



Add my own content

[✓ Commit directly to the `main` branch]
Create a **new branch** for this commit and start a pull request

Figure 10: Change content

Ensuring the correct order

By default, the automatic generator will look for all Markdown files (so those ending with `.md`) in the top-level of the repository except for the `README.md` file, order them alphabetically and create the different output formats by appending them in this order. If you want to upload multiple content files, you will either have to:

- adapt the names of the files so that they will be ordered correctly alphabetically, for example:

  - `01_Introduction.md`
  - `02_Some-Chapter.md`
  - `...`

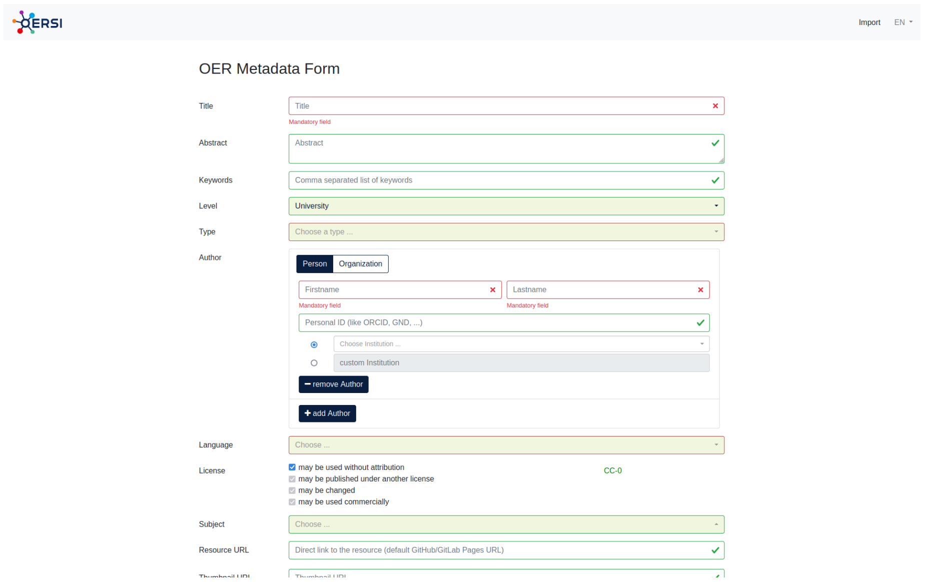
- or you will have to list every file in the correct order in `config.yml`.

You can learn more about that in the Configuration options section in the reference part of this tutorial.

Add your metadata

> Metadata is the data describing your OER. This includes information like the title, author, license and much more. If you don't supply metadata with your repository, it is unclear what your OER is about, who it is from, if and how your OER can be used and so on. This is why we have a `metadata.yml` file in our repository. This file lets us include the information about the OER directly in the repository. This is also necessary for inserting your OER into a search index like OERSI. This section shows how to replace the dummy metadata in the repository with the correct metadata describing your OER.

Perhaps the easiest way to generate your own metadata for your repository in the correct format is to use the OERSI metadata generator and fill out at least all required fields, but try to fill out as much as you can.

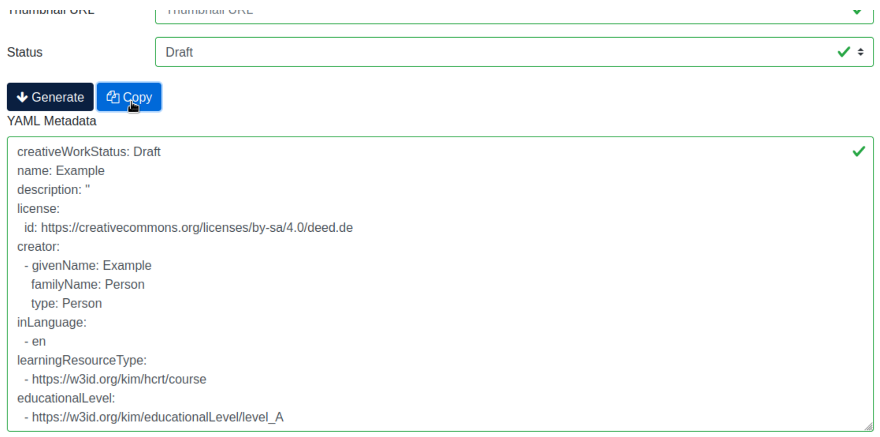


OER Metadata Form

Title	Title ✗ — Mandatory field
Abstract	Abstract ✓
Keywords	Comma separated list of keywords ✓
Level	University
Type	Choose a type ...
Author	[✓ Person] Organization; Firstname ✗ — Mandatory field; Lastname ✗ — Mandatory field; Personal ID (like ORCID, GND, ...) ✓; [✓ Choose Institution ...]; custom Institution; remove Author; add Author
Language	Choose ...
License	[✓ may be used without attribution] may be published under another license, may be changed, may be used commercially — CC-0
Subject	Choose ...
Resource URL	Direct link to the resource (default GitHub/GitLab Pages URL) ✓

Figure 11: Metadata generator

Now at the bottom of the page, you can click on `Generate`. This generates the metadata in the correct format. You can then copy the output to your clipboard either by using the `Copy` button, or by selecting the whole text (`Ctrl + A`) and copying it (`Ctrl + C`).



```
creativeWorkStatus: Draft
name: Example
description: ''
license:
  id: https://creativecommons.org/licenses/by-sa/4.0/deed.de
creator:
  - givenName: Example
    familyName: Person
    type: Person
inLanguage:
  - en
learningResourceType:
  - https://w3id.org/kim/hcrt/course
educationalLevel:
  - https://w3id.org/kim/educationalLevel/level_A
```

Figure 12: Copy output to clipboard

In your GitHub repository, edit the `metadata.yml` file. Now delete the whole file content and paste the output of the generator. To save, click on `Commit changes...` in the top right corner. Confirm by clicking on `Commit changes`.

Video: Update the metadata

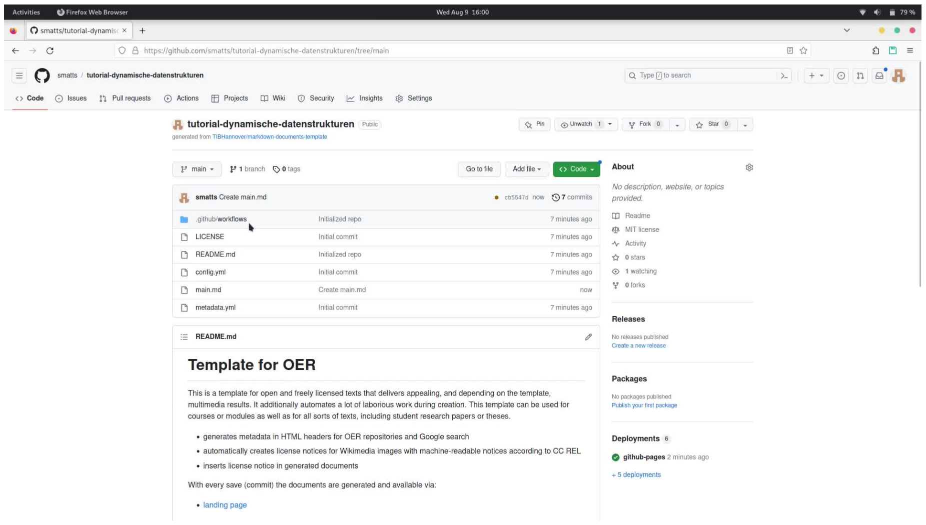
Insert your OER in OERSI

Let's assume that...

- you have put your whole completed course or document content into the repository, and it's either ordered alphabetically or you have defined the correct order in the `config.yml` file,
- you have enabled the `Pages` for `GitHub Actions` and there are no errors during Action execution, thus you have a published Page that represents the current version of your repository,
- you have entered full and correct metadata, ensuring the license is correct and you are not violating another work's license with this license,
- inside your `metadata.yml` file, you have set your creative work status to `Published` and the educational level to `University`,

then you are ready to put your OER into OERSI!

To put your course into the Open Ecucational Resources Search Index (OERSI), head to the `About` settings in the index of your repository. Then in `Topics`, add



tutorial-dynamische-datenstrukturen Public

generated from TIBHannover/markdown-documents-template

smatts Create main.md cb5547d now 7 commits

.github/workflows	Initialized repo	7 minutes ago
LICENSE	Initial commit	7 minutes ago
README.md	Initialized repo	7 minutes ago
config.yml	Initial commit	7 minutes ago
main.md	Create main.md	now
metadata.yml	Initial commit	7 minutes ago

README.md

Template for OER

This is a template for open and freely licensed texts that delivers appealing, and depending on the template, multimedia results. It additionally automates a lot of laborious work during creation. This template can be used for courses or modules as well as for all sorts of texts, including student research papers or theses.

- generates metadata in HTML headers for OER repositories and Google search
- automatically creates license notices for Wikimedia images with machine-readable notices according to CC REL
- inserts license notice in generated documents

With every save (commit) the documents are generated and available via:

- landing page

About

No description, website, or topics provided.

Readme
MIT license
Activity
0 stars
1 watching
0 forks

Releases

No releases published
Create a new release

Packages

No packages published
Publish your first package

Deployments 6

github-pages 2 minutes ago
+ 5 deployments

Figure 13: Video preview: "Update the metadata (videos/metadata-placeholder.mp4)" (not available in PDF version). Click to view the video online.

open-educational-resources.

The OERSI updates its index every night. So you will be able to find your OER the next day either through the search bar or by filtering using the filters on the left (e.g. search for your name in `Author` or setting the `Provider` to GitHub to only show OER from GitHub).

If you want to immediately see your changes in the OERSI, you can use the record updater.

To take the OER out of OERSI, simply set the status (`creativeWorkStatus`) in the `metadata.yml` to `Draft` or `Incomplete` (or alternatively, you could remove the topic `open-educational-resources`).

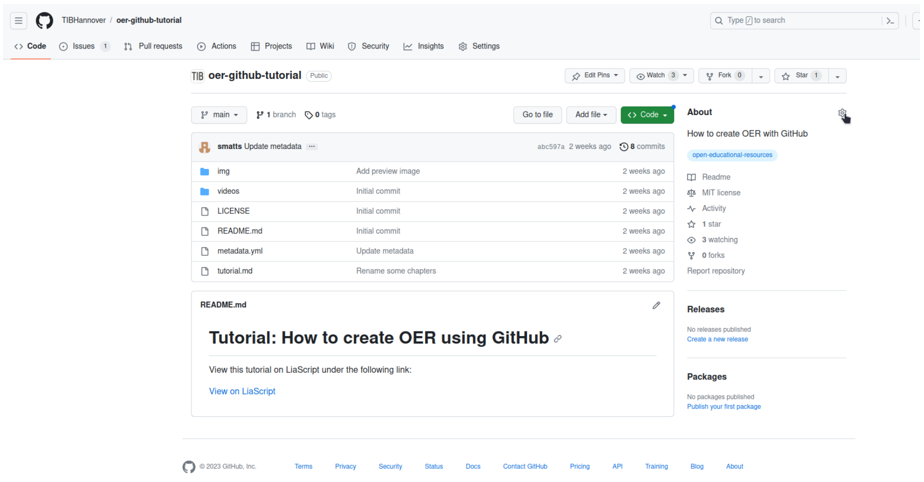
Done!

Congratulations, you did it!

If you want to learn more about what you can do with this template, you can continue with the reference part.

Reference

> You have your OER in different output formats, complete with your own content and metadata and it is even already indexed in oersi, but you want to learn more about what you can do with this template?



smatts Update metadata … abc597a 2 weeks ago 8 commits

img	Add preview image	2 weeks ago
videos	Initial commit	2 weeks ago
LICENSE	Initial commit	2 weeks ago
README.md	Initial commit	2 weeks ago
metadata.yml	Update metadata	2 weeks ago
tutorial.md	Rename some chapters	2 weeks ago

README.md

Tutorial: How to create OER using GitHub

View this tutorial on LiaScript under the following link:

View on LiaScript

About

How to create OER with GitHub

open-educational-resources

Readme
MIT license
Activity
1 star
3 watching
0 forks
Report repository

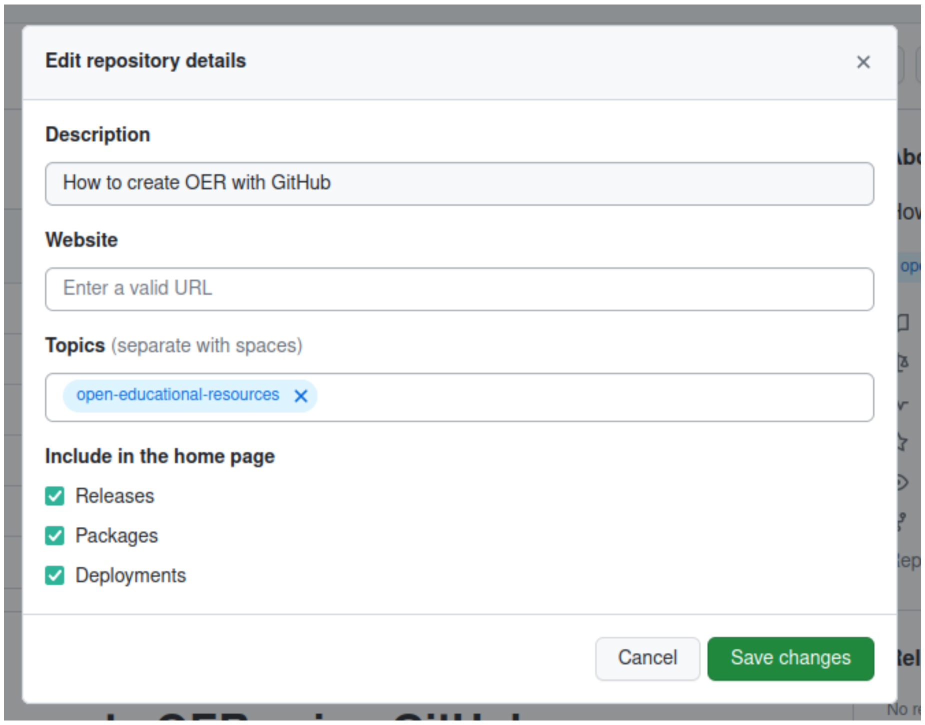
Releases

No releases published
Create a new release

Packages

No packages published
Publish your first package

Figure 14: About settings


Edit repository details

Description

How to create OER with GitHub

Website

Enter a valid URL

Topics (separate with spaces)

open-educational-resources

Include in the home page

- [✓] Releases
- [✓] Packages
- [✓] Deployments

Figure 15: Set the topic

In this part of the tutorial, we highlight further configuration options, Markdown basics, workflows in GitHub and much more.

- Configuration options
- Markdown
- Git
- Working offline
- Immediate update in OERSI
- Different formats

Configuration options

> In the repository, there is a file called `config.yml`. This gives you some configuration options concerning the automatic generation of your OER, like the order of your content.

In the top level of the repository, there is a file named `config.yml`. It includes configuration for these four things:

- `output`: state the output formats that you want to be automatically generated
- `generate_landingpage`: decide whether or not to create a landing page
- `content_files`: define the order of your content files
- `generate_reuse_note`: decide whether or not to generate a reuse note on the generated documents

Editing this file As you are editing this file, you have to consider the format of this file. This file is a `yaml` file. This file has a specific structure you have to follow, or else the automatic generator will not work.

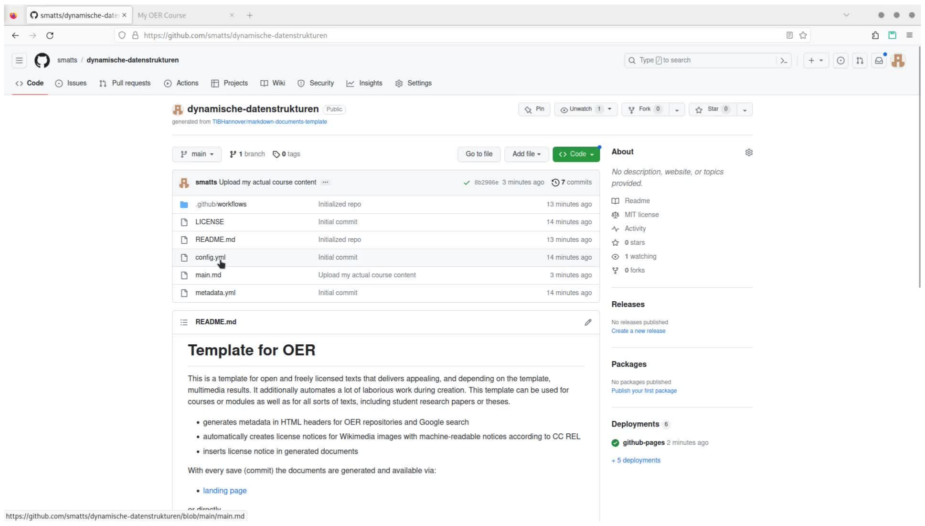
As you can see when opening the file, the structure looks like this:

```yaml
output:
  - format: asciidoc
  - format: epub
  - format: html
  - format: pdf
generate_landingpage: true
# content_files:  # uncomment this to set the order of the documents (default alphabetical)
#   - chapter01.md
#   - chapter02.md
#   - chapter03.md
#   - chapter04.md
generate_reuse_note: true
```

So we have to consider the identation and the correct symbols to use. If we want to use the `content_files` option, we uncomment this by deleting the `#` before each line and removing the residual spaces:

```
output:
  - format: asciidoc
  - format: epub
  - format: html
  - format: pdf
generate_landingpage: true
content_files:  # uncomment this to set the order of the documents (default alphabetical)
  - chapter01.md
  - chapter02.md
  - chapter03.md
  - chapter04.md
generate_reuse_note: true
```

The dashes (-) have to start with two spaces before the dash and then one space after the dash. Basically, just make sure the number of spaces is consistent throughout the whole file.



smatts Upload my actual course content ··· 8b2906e 3 minutes ago 7 commits

.github/workflows	Initialized repo	13 minutes ago
LICENSE	Initial commit	14 minutes ago
README.md	Initialized repo	13 minutes ago
config.yml	Initial commit	14 minutes ago
main.md	Upload my actual course content	3 minutes ago
metadata.yml	Initial commit	14 minutes ago

No description, website, or topics provided.

No releases published

No packages published

README.md

Template for OER

This is a template for open and freely licensed texts that delivers appealing, and depending on the template, multimedia results. It additionally automates a lot of laborious work during creation. This template can be used for courses or modules as well as for all sorts of texts, including student research papers or theses.

- generates metadata in HTML headers for OER repositories and Google search
- automatically creates license notices for Wikimedia images with machine-readable notices according to CC REL
- inserts license notice in generated documents

With every save (commit) the documents are generated and available via:

- landing page

or directly

Figure 16: Video preview: "Config file explained (videos/config.mp4)" (not available in PDF version). Click to view the video online.

Video tutorial

Markdown

For a good overview on what Markdown is and what you can do with it, you can check out the Markdown guide from Matt Cone:

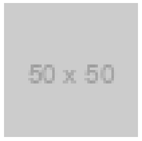
- What is Markdown and why should I use it?
- Basic Syntax

Headings

To create a heading, add number signs (`#`) in front of a word or phrase. The number of number signs you use should correspond to the heading level. For example, to create a heading level three (`<h3>`), use three number signs (e.g., `### My Header`) [1].

Heading level	How to write it	How it looks
Heading level 1	# Heading level 1	
Heading level 2	## Heading level 2	
Heading level 3	### Heading level 3	
Heading level 4	#### Heading level 4	
Heading level 5	##### Heading level 5	
Heading level 6	###### Heading level 6	

Markdown elements

Element	How to write it	How it looks
Bold	**Bold text**	**Bold text**
Italic	*Italicized text*	*Italic text*
Blockquote	> blockquote	To be seen at the top of this section
Ordered list	1. First item 2. Second item	1. First item 2. Second item
Unordered list	- First item - Second item	- First item - Second item
Code	‘code‘	`code`
Horizontal rule	---	—
Link	Link text	Markdown Guide
Image	Alt text	

[1] Matt Cone, markdownguide.org, Basic Syntax

Links to other sections

You can link to sections within your document using the link syntax shown in the table above. The links are generated from the heading:

- `# Heading` -> `#heading` (You can reference it like this: `[Heading](#heading)`)
- `# Heading 1` -> `#heading-1`
- `### Lower heading level!` -> `#lower-heading-level`

You can also set custom links like this:

```
## My heading {#custom-id}
```

Now the link to this section is `#custom-id`.

Git

> Here, you will only find a very brief overview of Git. If you want to learn more about Git, You can find an abundance of free tutorials, books and videos about Git, how it works and how to use it online. We highly recommend to check out the free online open textbook Pro Git.

What is Git and why should you use it?

Have you ever worked on different versions of a document, or even with different people? Then you likely know how hard it is to keep track of changes, and how easy to accidentally overwrite them. Git is a tool that helps with that. **It is a so-called *version control system.*** To learn more about version control, you can check out the Version Control section of the Pro Git book.

> By playing the video you agree to YouTube retrieving and storing information about you in the form of cookies.

How Git works

Git is used to keep track of changes of the files inside your repository. Once you add a file to git, it is tracked. Git now automatically detects whether the file was modified or not. You can then stage files and add them to a commit. All these commits form your history, you can see which changes were made when by whom and you can even revert files back to specific commits.

It is also possible to have multiple branches. You start out with one branch, usually called the **main** or **master** branch. However, at any given time from any branch, you can create a new branch. There, you can change files without anything on the main branch being affected by it - you are basically working in parallel, as if you copied your entire repository some other place, but much more efficient. Plus you can easily switch between branches and merge changes of one branch into another. This makes it easy to not only have one stable version on the main branch while changing things on other branches, but also to collaborate with others and not have each other's changes overwrite anything.

Working on different branches and eventually merging them can lead to merge conflicts. Merge conflicts arise when there are two changes to the same lines of a file. They are resolved by choosing which changes to keep or by bringing them together manually.

> By playing the video you agree to YouTube retrieving and storing information about you in the form of cookies.

Working offline

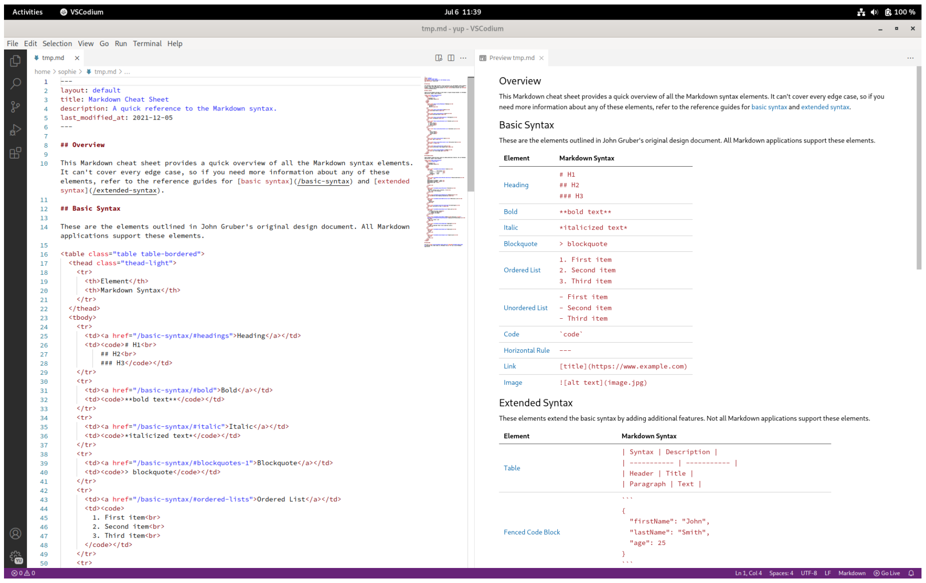
> In this tutorial, we work online the entire time. We create all files directly on GitHub. Another way of working is to work offline in a local folder on your computer and *then* uploading your files to a git repository. This, however, requires some knowledge about Git and a text editor.

Since we are working with `Git` (see Git section for more information), and want to publish our content on `GitHub` (see GitHub section), we have to find a way to bring our offline work online. For this, you should check out the Git section.

Editors

You know text editors like Word or LibreOffice. However, we want to use *plain text* that is readable for machines. This is why we are working with Markdown to format our *plain text*.

There are a lot of editors that help you with writing text using the Markdown syntax. They use syntax highlighting to better understand the structure of Markdown and can provide a live preview as you type.



Activities
VSCodium
Jul 6 11:39
100 %
tmp.md - yup - VSCodium
File Edit Selection View Go Run Terminal Help
tmp.md
home > sophie > tmp.md > ...
Preview tmp.md
Ln 1, Col 4 Spaces: 4 UTF-8 LF Markdown Go Live

```
---
layout: default
title: Markdown Cheat Sheet
description: A quick reference to the Markdown syntax.
last_modified_at: 2021-12-05
---

## Overview

This Markdown cheat sheet provides a quick overview of all the Markdown syntax elements.
It can't cover every edge case, so if you need more information about any of these
elements, refer to the reference guides for [basic syntax](/basic-syntax) and [extended
syntax](/extended-syntax).

## Basic Syntax

These are the elements outlined in John Gruber's original design document. All Markdown
applications support these elements.

<table class="table table-bordered">
  <thead class="thead-light">
    <tr>
      <th>Element</th>
      <th>Markdown Syntax</th>
    </tr>
  </thead>
  <tbody>
    <tr>
      <td><a href="/basic-syntax/#headings">Heading</a></td>
      <td><code># H1<br>
          ## H2<br>
          ### H3</code></td>
    </tr>
    <tr>
      <td><a href="/basic-syntax/#bold">Bold</a></td>
      <td><code>**bold text**</code></td>
    </tr>
    <tr>
      <td><a href="/basic-syntax/#italic">Italic</a></td>
      <td><code>*italicized text*</code></td>
    </tr>
    <tr>
      <td><a href="/basic-syntax/#blockquotes-1">Blockquote</a></td>
      <td><code>> blockquote</code></td>
    </tr>
    <tr>
      <td><a href="/basic-syntax/#ordered-lists">Ordered List</a></td>
      <td><code>
        1. First item<br>
        2. Second item<br>
        3. Third item<br>
      </code></td>
    </tr>
    <tr>
```

0 0

Overview

This Markdown cheat sheet provides a quick overview of all the Markdown syntax elements. It can't cover every edge case, so if you need more information about any of these elements, refer to the reference guides for basic syntax and extended syntax.

Basic Syntax

These are the elements outlined in John Gruber's original design document. All Markdown applications support these elements.

Element	Markdown Syntax
Heading	`# H1` `## H2` `### H3`
Bold	`**bold text**`
Italic	`*italicized text*`
Blockquote	`> blockquote`
Ordered List	`1. First item` `2. Second item` `3. Third item`
Unordered List	`- First item` `- Second item` `- Third item`
Code	`` `code` ``
Horizontal Rule	`---`
Link	`[title](https://www.example.com)`
Image	`![alt text](image.jpg)`

Extended Syntax

These elements extend the basic syntax by adding additional features. Not all Markdown applications support these elements.

Element	Markdown Syntax
Table	`| Syntax | Description |` `| ----------- | ----------- |` `| Header | Title |` `| Paragraph | Text |`
Fenced Code Block	```` ``` { "firstName": "John", "lastName": "Smith", "age": 25 } ``` ````

Figure 17: Editor

The editor you can see in the screenshot is called **VSCodium**, which is the open source version of a popular editor **VSCode** by Microsoft. In this editor, you can install plugins that can help you. For example, you can install a plugin that can render a preview of your LiaScript course in your browser without having to upload your changes. It also has a built-in Git functionality which helps you publish your changes.

Put your changes online

You *could* upload your locally edited files to GitHub, either by simply uploading this file or by editing the file you want to change and replacing that file's content with your local content.

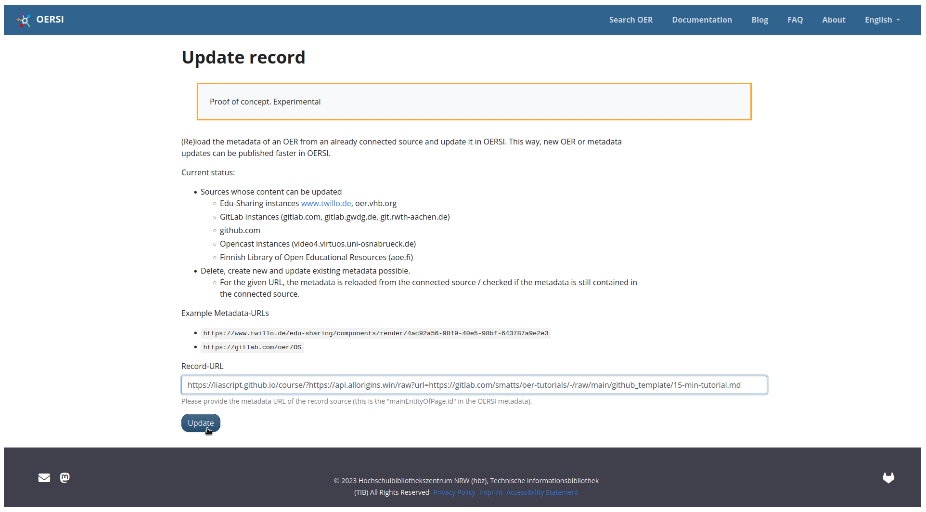
The better way, however, is to use Git for this - either by using your editor's built-in Git functionality or by using the Git program itself. Please check out the abundance of free online resources on how to do this.

> By playing the video you agree to YouTube retrieving and storing information about you in the form of cookies.

Immediate update in OERSI

> Your OER will be automatically inserted into the OERSI if you fulfill the requirements listed in the insert your OER into OERSI section. The resources are updated each night. If you want to see **immediate** changes, you can use the (experimental) record updater.

Head to the record updater page on oersi.org. Insert the link to your GitHub repository or to the generated landing page of your OER.



Update record

Proof of concept. Experimental

(Re)load the metadata of an OER from an already connected source and update it in OERSI. This way, new OER or metadata updates can be published faster in OERSI.

Current status:

- Sources whose content can be updated
  - Edu-Sharing instances www.twillo.de, oer.vhb.org
  - GitLab instances (gitlab.com, gitlab.gwdg.de, git.rwth-aachen.de)
  - github.com
  - Opencast instances (video4.virtuos.uni-osnabrueck.de)
  - Finnish Library of Open Educational Resources (aoe.fi)
- Delete, create new and update existing metadata possible.
  - For the given URL, the metadata is reloaded from the connected source / checked if the metadata is still contained in the connected source.

Example Metadata-URLs

- `https://www.twillo.de/edu-sharing/components/render/4ac92a56-9819-40e5-98bf-643787a9e2e3`
- `https://gitlab.com/oer/OS`

Record-URL

https://liascript.github.io/course/?https://api.allorigins.win/raw?url=https://gitlab.com/smatts/oer-tutorials/-/raw/main/github_template/15-min-tutorial.md

Please provide the metadata URL of the record source (this is the "mainEntityOfPage.id" in the OERSI metadata).

© 2023 Hochschulbibliothekszentrum NRW (hbz), Technische Informationsbibliothek (TIB) All Rights Reserved Privacy Policy Imprint Accessibility Statement

Figure 18: Record updater

Then, click on update. Your OER should now be updated in the OERSI.

Different formats

> After following this tutorial, we have different output formats like a HTML version, a PDF version and so on. We could, however, generate all kinds of different formats, for example a course format like this tutorial.

Below you find a short list of several possible formats your OER could use.

Markdown documents template

This is the template that is described in this tutorial. As you know by now, it takes your Markdown files, puts them together and generates different formats from them. These formats are linked to and can be downloaded from the landing page that is generated using GitHub Actions and GitHub Pages.

Pro:

- Easy setup with our template
- Automatic generation of different formats
- Can include interactive elements
- Always have the newest changes online
- GitHub automatically tracks changes made to your files
- Easy collaboration with others on GitHub
- Customizable with CSS

Con:

- Needs a GitHub account
- Customizing can be tricky

Markdown slides template

The Markdown slides template is very similar to the Markdown documents template from this tutorial. But instead of creating a single text document, the slides template creates several slides. For this, you create one Markdown file for one set of slides and the template generates the slides in both HTML and PDF format and shows a list of all generated slides together with preview images in a GitHub Page.

Pro: + Easy automatically generated slides in two formats + Easy setup with our template + Automatic up to date overview page + Always have the newest changes online + Automatically tracks changes made to your files + Easy collaboration with others

Con: - Slides are only partly customizable - Needs a GitHub account

LiaScript

LiaScript takes a markdown file and automatically generates a course format from it. The tutorial you are currently viewing is actually made with LiaScript! Or rather, the Markdown file this tutorial is written in is being interpreted by

LiaScript, which therewith generated this course. So all you really need is a Markdown file.

Pro: + Super easy to set up, just need one markdown file somewhere on the internet! + Runs everywhere + Large number of elements you can use like graphs, quizzes, … + No installation, everything happens live & online + Easy to click through the different sections + Automatic translation into many different languages + Interactive elements and extended Markdown can be used + Responsive website

Con: - Only online, no download - Depends on one single service - Only one Markdown file at a time

Static Site Generators

A static site generator generates a static site. Typically, it will be possible to also write your content using Markdown, but at the same time, you will be able to edit your layout and include more elements yourself which the static site generator then uses to create your web page. Of course, this requires some basic knowledge about HTML, CSS and the static site generator you are using. You should also be familiar with the command line or using GitHub Actions yourself.

Pro: + Creates a lightweight web page + Very customizable

Con: - Not for complete beginners - you need some knowledge about web development - Takes longer to get a first version running / more configuration and technical know-how necessary

JupyterBooks

Jupyter Book is a free and open source tool to create online books. You can create sections and subsections that you can click through. They are added to a table of contents, which can be viewed in a sidebar or accessed via a menu. It is also possible to include executable content. Moreover, you can download your book in both Markdown and PDF format. You can start out with a template supplied by the software itself.

Pro: + Creates online books with sections and table of contents to click through + Allows lots of configuration and structuring + Supplies download as Markdown and PDF + Can include executable content

Con: - Not for complete beginners (you will need to run commands from the command line or create a GitHub Action) - Configuration is done via config files, which can be tricky to learn if you are unfamiliar with coding

Troubleshooting

Something does not work? Maybe you find your issue right here.

I do not see my changes

You have added your content and your metadata but can not see your changes in your landing page and generated documents? Following these steps might help you.

1. Delete your cache and reload your page/document

Often, the old version of the page is still loaded in your browser's cache. If you reload the page or document by hitting `Ctrl + F5` together, you can reload your page while deleting the cache of that page. You can also open your page or document in a new private tab or window, as the browser usually does not use its cache there.

2. Check your file names and content

Check if your file name contains characters like **spaces**. These are not allowed and cause the document generation to break.

Also, ensure that there are no special characters like emojis or other unicode characters in your documents. Our document processor does not understand these characters, which also leads to breaking the document generation.

3. Check your media

Sometimes, images or videos can cause the document generation to break. Usually, this happens when you use an unsupported format. Stick to widespread formats like `png` and `jpg` to be sure.

FAQ

Can I upload non-text files like PDF files?

Yes, you can! Git however won't be able to track changes made to those files, it can only track *that* it was changed.

How can I change how images are displayed?

You can change image size, placement and more by using HTML-tags. HTML is, like Markdown, a markup language, basically converting plain text into formatted output.

Change image size:

```
<img src="path/to/image.png" alt="Image description" style="width:100px";
/>
```

The **src** contains the path to your image, the **alt** contains an image description, and the **style** contains the information about the image size. Here, we have set the **width** to **100 pixels**. You can also set the height (but note that setting a fixed width *and* height that does not match the original ratio of the image can skew it), and of course, change the size how you want.

Make text float around an image:

```
<img src="path/to/image.png" alt="Image description" style="float:
left; margin: 0 20px 20px 0;" />
```

Again, **src** contains the path to the image and **alt** its description. Now in the **style** tag, you find two directions: `float: left;` places your image on the left and allows other content like text to *float* around it. The tag `margin: 0 20px 20px 0;` is about spacing: you can set a margin around your image (or element in general), leaving that space blank. Here, the values for the margin are sorted top, right, bottom, left. So in this case, there is a margin of 20 pixels to the right and to the bottom. This prevents the text to directly "touch" your image, making it easier to read.

Why are the direct paths to my files different?

You may have encountered links containing `https://raw.githubusercontent.com/`. If you are wondering what this means and why GitHub changes your URL this way, the reason is the following:

When you go through your repository's files in GitHub, you see them embedded in the GitHub web page, which lets you do several things like edit, delete, show its history and so on. So when you have a link to this file, the link points to this web page and not to the file itself (which is called the *raw* file). If you want to access the file *only*, so the raw file, GitHub allows you to do that by changing the URL to `https://raw.githubusercontent.com/`.

How can I create a new folder?

You can not create an empty folder. This means you have to add files in GitHub *in* that new folder. Either create a new file and change the path to that new folder (i.e. instead of creating a new file `file.md`, write it with a new folder like `newfolder/file.md`, a new folder will be automatically created) or upload a non-empty folder.

Note on reuse