

BookStack

What is BookStack? How do I install it?

- [BookStack: A Clear, Structured Wiki for Teams](#) □
- [Install your own BookStack instance](#)
- [Backup + Migration](#)
- [Formeln & Diagramme](#)
- [art10m's BookStack Hacks](#)

BookStack: A Clear, Structured Wiki for Teams ☐☐

BookStack is an open-source platform for creating and organizing documentation—think *internal knowledge base*, *team wiki*, or *product docs portal*—with a structure that’s intentionally familiar: **Books → Chapters → Pages**. That simple model (paired with a clean editor and solid permissions) makes it especially appealing for teams that want documentation to be *easy to write*, *easy to navigate*, and *easy to govern*.

What BookStack is (and why people like it) ☐

BookStack is designed to reduce the friction that often comes with documentation tools. Instead of treating knowledge as an unstructured pile of pages, it encourages a **hierarchy** that matches how many teams already think:

1. **Books**
 - Top-level collections (e.g., *Engineering Handbook*, *IT Runbooks*, *Customer Support Playbook*).
2. **Chapters**
 - Subsections within a book (e.g., *Onboarding*, *Deployments*, *Incident Response*).
3. **Pages**
 - The actual documentation content (procedures, how-tos, references, policies).

This hierarchy helps readers quickly orient themselves, and it helps authors avoid “where do I put this?” paralysis. ☐☐

Core features that matter in practice ☐☐

1) Editing experience (built for real-world docs)

BookStack provides a modern, approachable authoring workflow:

1. **WYSIWYG editor** (commonly preferred for non-technical contributors)
 - Great for teams where everyone—from engineers to operations to support—needs to contribute.
2. **Markdown support**
 - Useful for technical teams who prefer writing in plain text with predictable formatting.
3. **Rich content tools**
 - Easy inclusion of images, links, tables, and code blocks.
4. **Page revision history**
 - Helps you see what changed and when, which is crucial for controlled documentation.

2) Organization and navigation

BookStack's structure doesn't just help authors—it improves consumption:

1. **Predictable browsing**
 - Readers can move from a *Book* down to *Chapters* and *Pages* without hunting.
2. **Search**
 - Fast searching becomes more valuable when paired with consistent structure and naming.
3. **Cross-linking**
 - Pages can reference other pages to build a connected knowledge graph without losing hierarchy.

3) Permissions and access control

Governance is where many wikis struggle. BookStack typically shines here:

1. **Role-based access**
 - Control who can view, create, edit, or delete content.
2. **Granular permissions**
 - Apply restrictions at different levels (e.g., specific books or content areas).
3. **Team-friendly collaboration**
 - Helps keep sensitive runbooks or HR policies restricted while leaving general knowledge open.

4) Authentication options (fits into existing identity stacks)

Depending on your setup and version, BookStack can integrate with common authentication approaches to reduce account sprawl and simplify onboarding/offboarding:

1. **Local authentication**
2. **LDAP/Active Directory**
3. **SSO-style integrations** (commonly via SAML/OAuth-like approaches in organizational environments)

(Exact availability can depend on how you deploy and configure it.)

5) Media and attachment handling ☐☐

Docs rarely live as pure text:

1. **Image management**
 - Useful for diagrams, screenshots, and annotated procedures.
 2. **File attachments**
 - Handy for templates, exports, and reference files—though many teams prefer linking to a source of truth (like Git) for certain assets.
-

Typical use cases (where BookStack fits best) ☐☐

BookStack is broadly useful, but it's especially strong when your team values clarity and structure.

1. **Internal team wiki**
 - Decision logs, meeting notes, standards, and best practices.
2. **IT & Ops runbooks**
 - Incident response steps, on-call procedures, system recovery guides.
3. **Engineering documentation**
 - Architecture overviews, onboarding guides, deployment instructions.
4. **Support & customer success playbooks**
 - Troubleshooting flows, known issues, escalation processes.
5. **Policy and compliance documentation**
 - Controlled edits, auditable changes, restricted sections.

Strengths and trade-offs ⚖️

Strengths

1. **Strong information architecture**
 - The Books/Chapters/Pages model makes messy knowledge more navigable.
2. **Low barrier to contribution**
 - Non-technical users often feel comfortable quickly.
3. **Practical permissions**
 - Good for teams that need structure *and* control.
4. **Self-host friendly**
 - Ideal for organizations that prefer keeping data on their own infrastructure.

Trade-offs to consider

1. **Hierarchy can be limiting for some knowledge styles**
 - If your team prefers a purely tag-driven, graph-like, or database-like knowledge system, you may feel constrained.
2. **Not a full doc-as-code pipeline**
 - While Markdown exists, BookStack isn't primarily designed to be a Git-native docs workflow in the way some static-site generators are.
3. **Customization**
 - You can brand and configure it, but extreme customization may require deeper technical effort and ongoing maintenance.

How teams keep BookStack content high-quality 📝

A tool helps, but process makes it stick. Common patterns that work well:

1. **Documentation templates**
 - For recurring page types (runbooks, how-tos, policies).
2. **Naming conventions**
 - Consistent titles improve search and scanning (e.g., "How to ...", "Runbook: ...", "Policy: ...").

3. **Ownership and review cadence**

- Assign a “page owner” or “book maintainer” and review quarterly or after major changes.

4. **Link to sources of truth**

- Reference tickets, diagrams, repos, or monitoring dashboards rather than duplicating volatile data.

5. **Use permissions to reduce accidental edits**

- Keep “official” procedures protected while allowing contributions in draft areas.
-

Deployment and operations overview

BookStack is commonly deployed in a web-app style environment with a database backend.

1. **Containerized deployment**

- Many teams run it via Docker for predictable setup and upgrades.

2. **Backups**

- Plan backups for both:
 1. The **database** (core content, users, settings)
 2. The **uploaded files** (images/attachments)

3. **Upgrades**

- Regular updates help with security patches and new features; test in staging if possible.

4. **Performance**

- For most teams, default performance is solid; larger orgs may tune caching and database resources.
-

Who should choose BookStack?

BookStack is a great choice if you want:

1. A **straightforward wiki** that’s easy to navigate
2. A **structured documentation hierarchy**
3. **Permissions** that can match real organizational needs
4. A **self-hostable** solution with a clean UI

If your top priority is a **Git-first** docs workflow with automated builds, PR reviews, and versioned docs tied tightly to code releases, you might instead prefer a doc-as-code toolchain—though many

teams still use BookStack effectively alongside those systems (e.g., BookStack for runbooks and onboarding, Git for developer reference docs).

If you tell me your context, I can tailor it ☐☐

If you share a bit about your situation, I can adapt the article into a recommendation or a deployment plan:

1. **Team size** and who will author docs (engineers only vs. cross-functional)
2. Whether you need **SSO/LDAP**
3. Whether this is **internal-only** or partially public
4. Your preferred hosting approach (Docker, VM, Kubernetes, etc.)
5. Your documentation style (runbooks, policies, product docs, onboarding, etc.)

Install your own BookStack instance

This is the way this instance of BookStack is installed.

This guide starts **from the point** where you already have a **Linux server** and **Docker** is installed. The “steps before that” will follow later — it’s really not hard (especially with AI help).

Overview: What’s being set up here?

In the end you’ll have:

- **BookStack** (wiki/docs system)
- **MariaDB** as the database
- **Caddy** as a reverse proxy with **automatic HTTPS** (Let’s Encrypt) ☐

Background: What is Caddy? ☐☐

“**Caddy** is a modern web server (similar to *Nginx/Apache*), written in *Go*, that embraces “*security-by-default*” and, above all, makes **HTTPS** extremely convenient.

Core idea: “A web server that just works” ☐

Key features ☐

1. Automatic HTTPS (TLS)

- Certificates are obtained and renewed automatically (typically via *Let's Encrypt*).

2. Simple configuration

- Via an easy-to-read **Caddyfile**.

3. Reverse proxy & load balancing

- Ideal for forwarding requests to services (e.g., Docker containers).

4. Good defaults

- Many sensible security standards are enabled by default.

5. Modular extensibility

- If you have special requirements, Caddy can be extended.

Typical use cases ☐☐

- Hosting websites (static/dynamic)
- Reverse proxy in front of an app (e.g., `/api` → backend)
- TLS termination
- Local dev setups with HTTPS

Step 1: Prepare directory & files ☐☐

Create the directory:

- `/opt/bookstack`

Create **two files** inside it:

- `docker-compose.yml`
- `Caddyfile`

“📌 **Important:** You'll need to adjust a few values in a moment — you can clearly see *where* in the YAML.

Step 2: Configure Docker Compose (`docker-compose.yml`) ☐☐

Paste the following content (and adjust it where necessary):

services:

 mariadb:

 image: lscr.io/linuxserver/mariadb:latest

 container_name: bookstack-mariadb

 environment:

- PUID=1000
- PGID=1000
- TZ=Europe/Berlin
- MYSQL_ROOT_PASSWORD=PW_OF_MYSQL_ROOT
- MYSQL_DATABASE=bookstack
- MYSQL_USER=bookstack
- MYSQL_PASSWORD=PW_OF_MYSQL_DB

 volumes:

- ./mariadb:/config

 restart: unless-stopped

 bookstack:

 image: lscr.io/linuxserver/bookstack:latest

 container_name: bookstack

 depends_on:

- mariadb

 environment:

- PUID=1000
- PGID=1000
- TZ=Europe/Berlin

- APP_URL=https://wiki.fabula.vision
- # Where will BookStack be accessible? (Your URL)
- APP_KEY=base64:...
- # To generate it, run: docker run -it --rm --entrypoint /bin/bash

lscr.io/linuxserver/bookstack:latest appkey

- APP_THEME=custom

#'custom' makes it possible to use hacks; more on that here:

<https://www.bookstackapp.com/hacks/applying/>

- DB_HOST=mariadb
- DB_PORT=3306
- DB_DATABASE=bookstack
- DB_USERNAME=bookstack
- DB_PASSWORD=PW_OF_MYSQL_DB

```
# (As above!)

- APP_DEFAULT_DARK_MODE=true
# (Personal preference)

volumes:
  - ./bookstack:/config
restart: unless-stopped

caddy:
  image: caddy:latest
  container_name: caddy
  depends_on:
    - bookstack
  ports:
    - "80:80"
    - "443:443"
  volumes:
    - ./Caddyfile:/etc/caddy/Caddyfile:ro
    - ./caddy/data:/data
    - ./caddy/config:/config
  restart: unless-stopped
```

Step 3: Configure Caddy (Caddyfile)

Paste the following and replace the domain:

```
wiki.fabula.vision {
  # Your URL, of course...
  encode zstd gzip

  # Reverse proxy to BookStack (container is named "bookstack", internal port 80)
  reverse_proxy bookstack:80

  # sensible headers (optional)
```

```
header {
  # HSTS (only set this if you're sure HTTPS should remain enabled permanently)
  Strict-Transport-Security "max-age=31536000; includeSubDomains; preload"
  X-Content-Type-Options "nosniff"
  X-Frame-Options "SAMEORIGIN"
  Referrer-Policy "strict-origin-when-cross-origin"
}
```

Step 4: Start the containers ►

1. Change into the directory:

- `cd /opt/bookstack/`

2. Start the stack:

- `docker compose up -d`

Updating later works like this:

- `docker compose pull`
- `docker compose up -d`

3. Done ☐

Now quickly open the URL and change the default credentials!

- Default login: `admin@admin.com`
- Default password: `password`

Step 5: Cron job for BookStack ☐

Create a cron job:

1. Open crontab:

- `crontab -e`

2. Add the following line (cron every minute):

- `* * * * * docker exec -i bookstack php /app/www/artisan schedule:run >/dev/null 2>&1`

Limitations & modules/hacks ☐☐

There are three limitations that have stood out so far:

1. Email sending

- (e.g., “Forgot password”) doesn’t work for me, and my experiments haven’t produced a solution yet.
- I don’t strictly need it — but for example for comments (notifications) you can’t avoid it.
- More info:

<https://www.bookstackapp.com/docs/admin/email-webhooks/#email-configuration>

2. Extending via “hacks”

- BookStack can be extended via so-called “hacks.”

EDIT: I created [my own hacks](#) that are better!

- Two recommendations:

1. MathJax/TeX (math formulas):

- <https://www.bookstackapp.com/hacks/mathjax-tex/>

2. Mermaid Viewer (diagrams):

- <https://www.bookstackapp.com/hacks/mermaid-viewer/>

Important note about export ☐

- Mermaid (and formulas as well) **does not export cleanly** for me (e.g., PDF): code blocks appear instead of rendered content.

How do you install hacks as modules?

- We already did the first step: In the YAML it says

```
APP_THEME=custom
```

This loads modules from the following path:

- `/opt/bookstack/bookstack/www/themes/custom/modules`

- Then via SSH:

1. Switch into the container:

- `docker exec -it bookstack /bin/bash`

2. Go to the web directory:

- `cd /app/www/`

3. Run the command that’s listed in the respective hack (example *Mermaid*):

- On the hack page in the “Install as Module” section

<https://www.bookstackapp.com/hacks/mermaid-viewer/>

3. Shift+Enter — line break (CommonMark)

- Due to the decision to use the CommonMark standard, a simple line break is not rendered by default.
- Instead you need “two spaces at end of line + Enter” — that drove me crazy because many Markdown editors are used to *Shift+Enter*.
- My fix consists of two parts:

1. In the BookStack settings under

<https://example.com/settings/customization>

under “Custom HTML Head Content”, enter the following code:

```
<script>
  window.addEventListener('editor-markdown::setup', event => {
    event.detail.markdownIt.set({breaks: true});
  });
</script>
```

2. Create a `functions.php` under:

- `/opt/bookstack/bookstack/www/themes/custom/`

With the following content:

```
<?php

use BookStack\Theming\ThemeEvents;
use BookStack\Facades\Theme;

Theme::listen(ThemeEvents::COMMONMARK_ENVIRONMENT_CONFIGURE, function
($environment) {
  $environment->mergeConfig([
    'renderer' => [
      'soft_break' => "<br>",
    ]
  ]);

  return $environment;
});
```

Additional QoL fixes (under “Customization”)

Configure here:

- <https://yoururl.com/settings/customization>

Just paste each one under “Custom HTML Head Content”:

1. **Blockquotes: no thick bottom border / clean spacing**

- Cause: CSS (among other things, `<p>` with `margin-bottom`)

- Fix (same spacing top/bottom + no scrollbars):

```
<style>
  /* 1) Last & first direct child element in the blockquote: margin */
  .content-wrap blockquote > :last-child {
    margin-bottom: .3em;
  }

  .content-wrap blockquote > :first-child {
    margin-top: .3em;
  }

  /* 2) Blockquote: no scrollbars */
  .content-wrap blockquote {
    overflow: visible;    /* default "no scrolling" */
    overflow-x: visible;
    overflow-y: visible;
  }
</style>
```

2. Code blocks: soft wraps (no horizontal scrolling needed)

```
<script>
window.addEventListener('library-cm6::pre-init', event => {
  const detail = event.detail;
  const config = detail.editorViewConfig;
  const EditorView = detail.libEditorView;

  if (detail.usage === 'content-code-block') {
    config.extensions.push(EditorView.lineWrapping);
  }
});
</script>
```

3. Dark mode: lighter text (headings 70%, text 90%)

```
<style>
html.dark-mode .page-content h1, html.dark-mode .page-content h2, html.dark-mode
.page-content h3, html.dark-mode .page-content h4, html.dark-mode .page-content h5,
html.dark-mode .page-content h6 {
  color: hsl(0 0% 70%);
}

```

```
html.dark-mode body {  
  color: hsl(0 0% 90%);  
}  
</style>
```

4. **Text size: main content 110%, headings 90%**

```
<style>  
.page-content {  
  zoom:1.1;  
}  
  
.page-content h1, .page-content h2, .page-content h3, .page-content h4, .page-  
content h5, .page-content h6 {  
  zoom:.9;  
}  
</style>
```

Backup + Migration

Not tested yet!

“ I installed BookStack via Docker (as in [the guide](#)). Could you please explain how to create backups (full content backup) so that I can migrate the website if needed, e.g., to another server/domain? [Here](#) is BookStack's official guide.

Full BookStack backup (Docker) – so that a move to *another server/domain* works

For a **complete, restorable backup**, you essentially need **two components**:

1. **Database** (all pages, books, users, permissions, settings ...)
2. **Files/config** (uploads, attachments, `.env`/APP_KEY, possibly themes/modules)

Since you have the setup like in your guide (`/opt/bookstack` with `docker-compose.yml`, volumes `./mariadb` and `./bookstack`), this is pleasantly simple.

What exactly needs to be backed up?

1) Database (MariaDB)

- The content is in the DB container, and in your setup it's persisted under:

- `./mariadb` (as volume `./mariadb:/config`)
- **Cleanest approach:** *SQL dump* (portable, tolerant across versions)

2) Files (BookStack container volume)

- Persistent data is located under:
 - `./bookstack` (as volume `./bookstack:/config`)
- In it, especially important:
 - `.env` (including **APP_KEY!**)
 - uploads/attachments
 - possibly `www/themes/custom/...` (your hacks/modules)

3) Compose/Caddy (helpful for migration)

- `docker-compose.yml`
- `Caddyfile`
- Caddy data optional:
 - `./caddy/data` & `./caddy/config` (certificates/state).
Not strictly required, but sometimes saves re-issuing/rate limits.

☐☐ Create a backup (recommended approach)

“ Assumption: project is located in `/opt/bookstack` and the containers are named as in your YAML:
`bookstack-mariadb` and `bookstack`.

Step 1: Create a backup folder

```
cd /opt/bookstack
mkdir -p backups
```

Step 2: Dump the database as SQL ☐☐

```
docker exec -i bookstack-mariadb \  
mysql_dump -u bookstack -pbookstack \  
--single-transaction --routines --triggers \  
bookstack > backups/bookstack-db-$(date +%F).sql
```

Important: Replace `-pbookstack` with your real password (from `MYSQL_PASSWORD`).

“ Alternative (safer because the password won't end up in shell history):
Omit `-p...`, then `mysql_dump` will prompt interactively for the password:

```
docker exec -it bookstack-mariadb mysql_dump -u bookstack -p --single-  
transaction --routines --triggers bookstack > backups/bookstack-db-$(date  
+%F).sql
```

Step 3: Archive files/volumes ☐☐

```
tar -czvf backups/bookstack-files-$(date +%F).tar.gz \  
docker-compose.yml Caddyfile \  
bookstack mariadb caddy
```

If you **don't** want to back up **Caddy**, you can do it leaner:

```
tar -czvf backups/bookstack-files-$(date +%F).tar.gz \  
docker-compose.yml Caddyfile \  
bookstack mariadb
```

Step 4: (Optional) Integrity check ☐

```
ls -lh backups/  
head -n 5 backups/bookstack-db-*.sql  
tar -tzf backups/bookstack-files-*.tar.gz | head
```

☐☐ Restore (on a new server) – incl. domain/URL migration

Step 0: Prerequisites

- Docker + Docker Compose installed
 - DNS of the new domain points to the new server (for HTTPS via Caddy)
-

Step 1: Copy the backup over

Pack on the **old** server or copy directly:

```
scp -r /opt/bookstack/backups user@new-server:/opt/bookstack/
```

Step 2: Restore files (Compose/volumes)



On the **new** server:

```
cd /opt/bookstack  
tar -xvzf backups/bookstack-files-YYYY-MM-DD.tar.gz
```

(Adjust the date, of course.)

Step 3: Start containers (DB should already be running) ☐☐

```
cd /opt/bookstack  
docker compose up -d mariadb
```

Wait a few seconds until MariaDB is ready.

Step 4: Import the database ☐☐

```
cat backups/bookstack-db-YYYY-MM-DD.sql | \  
docker exec -i bookstack-mariadb mysql -u bookstack -pbookstack bookstack
```

Again: adjust the password or use interactive mode.

Step 5: Start BookStack

```
docker compose up -d
```

☐☐ Domain/URL changed? (very important!)

If you move to a **different URL/domain**, you must:

1) Adjust `APP_URL` in the `.env`

In your volume, the `.env` is typically located at:

- `./bookstack/www/.env` (common with linuxserver/bookstack)
- or `./bookstack/.env` (depending on image/setup)

Find it like this:

```
find /opt/bookstack/bookstack -maxdepth 3 -name ".env" -print
```

Then set `APP_URL=` <https://new-domain.tld>.

“ **Pay attention:** The `APP_KEY` **must remain the old one** (otherwise encrypted data will break, e.g., MFA secrets).

2) BookStack command: “Update System URL”

So that content/links in the DB are rewritten cleanly:

```
docker exec -it bookstack php /app/www/artisan bookstack:update-url https://old-domain.tld https://new-domain.tld
```

(If the path in the container is different, this often also works: `php artisan ...` from `/app/www/`.)

☐☐ Practical notes (so it’s truly “migration-safe”)

- Always store **backups offsite** (another system/storage).
- **Database dump + file archive** is the most portable.
- If you restore to a **newer BookStack version**:
 1. Restore DB
 2. Bring containers up
 3. Then run migrations (if needed):

```
docker exec -it bookstack php /app/www/artisan migrate --force
```

If you want: I’ll make you a “one-command backup script” ☐☐

Just tell me briefly:

- Is your `.env` located at `bookstack/www/.env`?
- What are your containers actually named (`docker ps` output)?
- Do you want to back up Caddy certificates or preferably not?

Then I’ll give you a small script including rotation (e.g., “keep the last 7 backups”) and optional upload to S3/Hetzner Storage Box.

Formeln & Diagramme

□ Eine bunte Sammlung *vieler* Formeln (Inline **und** Block)

Unten findest du **ganz viele unterschiedliche Formeln** – von *sehr einfach* bis *ziemlich komplex*, teils **inline** (mit `$...$`) und teils als **Block** (mit `$$...$$`). Viel Spaß beim Kopieren & Testen □

1) Kurze Inline-Formeln (sehr gemischt)

- Lineare Terme: $3x-7$, $-2a+5b$, $m \cdot x+c$
 - Potenzen & Wurzeln: x^2 , a^{n+1} , $\sqrt{2}$, $\sqrt{x^2+y^2}$
 - Brüche: $\frac{1}{2}$, $\frac{x+1}{x-1}$, $\frac{a^2-b^2}{a-b}$
 - Beträge & Intervalle: $|x|$, $|a-b|$, $x \in [0,1]$
 - Summen/Produkte: $\sum_{k=1}^n k$, $\sum_{k=0}^{\infty} r^k$, $\prod_{i=1}^n i$
 - Fakultät & Binomial: $n!$, $\binom{n}{k}$
 - Logarithmen: $\ln(x)$, $\log_{10}(1000)$, $\log_a(x)$
 - Exponentialfunktionen: e^x , 2^{3x-1}
 - Trigonometrie: $\sin(x)$, $\cos^2(\theta)$, $\tan(\alpha+\beta)$
 - Hyperbelfunktionen: $\sinh(x)$, $\cosh^2(x)-\sinh^2(x)=1$
 - Grenzwerte: $\lim_{x \rightarrow 0} \frac{\sin x}{x}$
 - Ableitungen: $f'(x)$, $\frac{d}{dx} \left(x^3 \right)$, $\frac{\partial f}{\partial x}$
 - Integrale: $\int_0^1 x^2 dx$, $\int e^x dx$
 - Komplexe Zahlen: $i^2=-1$, $z=a+bi$, $|z|=\sqrt{a^2+b^2}$
 - Vektoren (symbolisch): $|v|$, $\langle u, v \rangle$
 - Wahrscheinlichkeiten: $P(A)$, $P(A \mid B)$, $E[X]$, $\mathrm{Var}(X)$
 - Informatik/Logik: $p \wedge q$, $p \rightarrow q$, $x \in S$
-

2) Klassische Block-Formeln (Basics bis Standard)

$$a^2+b^2=c^2$$

$$(x+y)^2=x^2+2xy+y^2$$

$$(a-b)(a+b)=a^2-b^2$$

$$\frac{d}{dx}(x^n)=n x^{n-1}$$

$$\int x^n dx = \frac{x^{n+1}}{n+1} + C \quad \text{für } n \neq -1$$

$$\int_0^{\infty} e^{-ax} dx = \frac{1}{a} \quad \text{für } a > 0$$

$$\sum_{k=1}^n k = \frac{n(n+1)}{2}$$

$$\sum_{k=1}^n k^2 = \frac{n(n+1)(2n+1)}{6}$$

$$\sum_{k=0}^n \binom{n}{k} = 2^n$$

3) Algebra, Polynome & Gleichungen (auch etwas „knackiger“)

$$ax^2+bx+c=0 \quad \Leftrightarrow \quad x=\frac{-b\pm\sqrt{b^2-4ac}}{2a}$$

$$x^3-6x^2+11x-6=(x-1)(x-2)(x-3)$$

$$\gcd(a,b)\cdot\mathrm{lcm}(a,b)=|ab|$$

$$\frac{1}{1-x}=\sum_{k=0}^{\infty} x^k \quad \text{für } |x|<1$$

$$\ln(1+x)=\sum_{k=1}^{\infty}(-1)^{k+1}\frac{x^k}{k} \quad \text{für } |x|<1$$

4) Trigonometrie & Analysis

$$\sin(\alpha+\beta)=\sin\alpha\cos\beta+\cos\alpha\sin\beta$$

$$\cos(\alpha+\beta)=\cos\alpha\cos\beta-\sin\alpha\sin\beta$$

$$\sin^2(x)+\cos^2(x)=1$$

$$\lim_{x \rightarrow 0} \frac{1 - \cos x}{x^2} = \frac{1}{2}$$

$$\frac{d}{dx} \big(\sin x\big) = \cos x$$

$$\frac{d}{dx} \big(\ln x\big) = \frac{1}{x}$$

$$\int_0^{2\pi} \sin(nx) dx = 0 \quad \text{für } n \in \mathbb{Z} \setminus \{0\}$$

5) Komplexe Zahlen & Euler

$$e^{i\theta} = \cos\theta + i\sin\theta$$

$$e^{i\pi} + 1 = 0$$

$$z = re^{i\theta} \quad \rightarrow \quad \overline{z} = re^{-i\theta}$$

$$|z_1 z_2| = |z_1| \cdot |z_2|$$

6) Lineare Algebra (ohne Matrix-Umgebung)

(Da du „ohne Matrix-Umgebung“ erwähnt hast, nutze ich eher symbolische Schreibweisen.)

$$\|x\|_2 = \sqrt{\langle x, x \rangle}$$

$$\langle x, y \rangle = \sum_{k=1}^n x_k y_k$$

$$\mathrm{proj}_u(v) = \frac{\langle v, u \rangle}{\langle u, u \rangle} u$$

$$\det(A) \neq 0 \quad \Leftrightarrow \quad A^{-1} \text{ existiert}$$

7) Differentialgleichungen & Dynamik

$$\frac{dy}{dx} = ky$$

$$y(x) = Ce^{kx}$$

$$m \frac{d^2x}{dt^2} + c \frac{dx}{dt} + kx = 0$$

$$\frac{\partial u}{\partial t} = D \frac{\partial^2 u}{\partial x^2}$$

8) Wahrscheinlichkeit & Statistik □□

\$\$

$$P(A \cup B) = P(A) + P(B) - P(A \cap B)$$

\$\$

\$\$

$$P(A \mid B) = \frac{P(A \cap B)}{P(B)}$$

\$\$

\$\$

$$P(A \mid B) = \frac{P(B \mid A)P(A)}{P(B)}$$

\$\$

\$\$

$$E[X] = \sum_x x \cdot P(X=x)$$

\$\$

\$\$

$$\mathrm{Var}(X) = E[X^2] - E[X]^2$$

\$\$

\$\$

$$\mathrm{Cov}(X, Y) = E[XY] - E[X]E[Y]$$

\$\$

\$\$

$$\rho_{X,Y} = \frac{\mathrm{Cov}(X, Y)}{\sqrt{\mathrm{Var}(X)\mathrm{Var}(Y)}}$$

\$\$

\$\$

$$\hat{p} = \frac{k}{n}$$

\$\$

9) Optimierung

\$\$

$$\min_{x \in \mathbb{R}^n} ; f(x)$$

\$\$

\$\$

$$\nabla f(x^*) = 0$$

\$\$

\$\$

$$x_{t+1} = x_t - \eta \nabla f(x_t)$$

\$\$

\$\$

$$\arg \min_x \left(\frac{1}{2} \|Ax - b\|_2^2 + \lambda \|x\|_1 \right)$$

\$\$

10) Diskrete Mathematik & Logik



\$\$

$$p \rightarrow q \equiv \neg p \vee q$$

\$\$

\$\$

$$p \leftrightarrow q \equiv (p \rightarrow q) \wedge (q \rightarrow p)$$

\$\$

\$\$

$$|A \times B| = |A| \cdot |B|$$

\$\$

\$\$

$$\sum_{k=0}^n \binom{n}{k} x^k y^{n-k} = (x+y)^n$$

\$\$

11) „Komplexere“ kombinierte Ausdrücke

\$\$

$$\int_0^1 x^\alpha (1-x)^\beta$$

$$\text{right), } dx = \frac{\Gamma(\alpha+1)\Gamma(\beta+1)}{\Gamma(\alpha+\beta+2)}$$

\$\$

\$\$
\frac{d}{dx}\left(\frac{\ln(1+x)}{x}\right)=\frac{\frac{x}{1+x}-\ln(1+x)}{x^2}
\$\$

\$\$
\sum_{n=1}^{\infty}\frac{1}{n^2}=\frac{\pi^2}{6}
\$\$

\$\$
\forall \epsilon>0 \exists \delta>0: |x-a|<\delta \rightarrow |f(x)-f(a)|<\epsilon
\$\$

Wenn du willst, kann ich dir das auch **thematisch sortiert** (z.B. nur Analysis, nur Statistik, nur Diskrete Mathe) oder als „**Stress-Test**“ mit besonders vielen verschachtelten Brüchen, Summen, Indizes, Fällen usw. bauen.

art10m's BookStack Hacks

<head>

```
<!-- =====  
BOOKSTACK CUSTOM HEAD CONFIGURATION  
  
This file contains custom styles, scripts, and integrations for BookStack:  
- Typography and layout adjustments (zoom, colors, spacing)  
- CodeMirror 6 line wrapping for code blocks  
- Markdown-it configuration (soft line breaks)  
- MathJax integration for LaTeX math rendering  
- Mermaid diagram rendering with theme synchronization  
- Light/Dark mode toggle button (works for both guests and logged-in users)  
===== -->  
  
<!-- =====  
CONSOLIDATED STYLES  
===== -->  
  
<style>  
  /* -----  
  DETAILS ELEMENT SPACING  
  Ensures proper bottom margin for the last child inside <details> elements  
  ----- */  
  .page-content details > *:last-child {  
    margin-bottom: .2em;  
  }  
  
  /* -----  
  BLOCKQUOTE ADJUSTMENTS  
  - Adds consistent vertical spacing for first/last children  
  - Disables overflow scrolling to prevent unwanted scrollbars  
  ----- */  
  .content-wrap blockquote > :last-child {  
    margin-bottom: .3em;
```

```
}

.content-wrap blockquote > :first-child {
  margin-top: .3em;
}

.content-wrap blockquote {
  overflow: visible;
  overflow-x: visible;
  overflow-y: visible;
}

/* -----
PAGE CONTENT ZOOM AND TYPOGRAPHY
- Applies 1.15x zoom to page content for better readability
- Compensates heading size with 0.9x zoom to maintain visual hierarchy
----- */
.page-content {
  zoom: 1.15;
}

.page-content h1,
.page-content h2,
.page-content h3,
.page-content h4,
.page-content h5,
.page-content h6 {
  zoom: .9;
}

/* -----
COLOR SCHEME: LIGHT MODE
- Dark text on light background for optimal contrast
- Slightly muted headings for visual hierarchy
----- */
html .page-content {
  color: hsl(0 0% 10%);
}
```

```
html .page-content h1,
html .page-content h2,
html .page-content h3,
html .page-content h4,
html .page-content h5,
html .page-content h6 {
  color: hsl(0 0% 30%);
}

/* -----
   COLOR SCHEME: DARK MODE
   - Light text on dark background
   - Adjusted heading colors for dark theme
   - Muted gutter colors for CodeMirror editor
   ----- */
html.dark-mode .page-content {
  color: hsl(0 0% 90%);
}

html.dark-mode .page-content h1,
html.dark-mode .page-content h2,
html.dark-mode .page-content h3,
html.dark-mode .page-content h4,
html.dark-mode .page-content h5,
html.dark-mode .page-content h6 {
  color: hsl(0 0% 70%);
}

html.dark-mode .co .cm-gutters {
  color: hsl(0 0% 33%);
}

/* -----
   CODEMIRROR EDITOR STYLING
   Removes border-radius for a cleaner, squared appearance
   ----- */
.page-content .cm-editor {
  border-radius: 0;
}
```

```
/* -----  
  MERMAID DIAGRAM CONTAINER  
  - Dashed border for visual identification during development/editing  
  - Centered layout with horizontal scroll for large diagrams  
  ----- */  
.mermaid-container {  
  border: 1px dashed #4238ff !important;  
}  
  
.mermaid {  
  margin: 1em auto;  
  overflow-x: auto;  
  text-align: center;  
}  
  
.mermaid svg {  
  max-width: 100%;  
  height: auto;  
  display: inline-block;  
}  
  
/* -----  
  THEME TOGGLE BUTTON (FIXED POSITION)  
  - Positioned at bottom-left corner for easy access  
  - Circular button with hover/active states  
  - Semi-transparent by default, full opacity on hover  
  ----- */  
.theme-toggle-fixed {  
  position: fixed;  
  bottom: 20px;  
  left: 20px;  
  z-index: 9999;  
  background: var(--color-primary, #206ea7);  
  border: none;  
  border-radius: 50%;  
  width: 44px;  
  height: 44px;  
  cursor: pointer;
```

```

    box-shadow: 0 2px 8px rgba(0, 0, 0, 0.25);
    display: flex;
    align-items: center;
    justify-content: center;
    transition: transform 0.2s ease, box-shadow 0.2s ease, opacity 0.2s ease;
    opacity: 0.7;
}

.theme-toggle-fixed:hover {
    transform: scale(1.1);
    box-shadow: 0 4px 12px rgba(0, 0, 0, 0.35);
    opacity: 1;
}

.theme-toggle-fixed:active {
    transform: scale(0.95);
}

.theme-toggle-fixed svg {
    width: 22px;
    height: 22px;
    fill: #ffffff;
}
</style>

<!-- =====
      MATHJAX CONFIGURATION
      Enables LaTeX math rendering with $ for inline and $$ for display math
      ===== -->
<script>
    window.MathJax = {
      tex: {
        inlineMath: [['$', '$']],
        displayMath: [['$$', '$$']]
      }
    };
</script>
<script id="MathJax-script" async src="https://cdn.jsdelivr.net/npm/mathjax@4/tex-mml-
chtml.js"></script>

```

```

<!-- =====
BOOKSTACK EVENT LISTENERS AND THEME TOGGLE
- CodeMirror 6 line wrapping configuration
- Markdown-it soft line break configuration
- Light/Dark mode toggle for guests and logged-in users
===== -->
<script>
(function () {
  'use strict';

  /* =====
  SHARED THEME STORAGE KEY
  Used by both the toggle button and Mermaid for consistent theme state
  ===== */
  var THEME_STORAGE_KEY = 'bookstack-guest-dark-mode';

  // Expose storage key globally for the Mermaid module to access
  window.BOOKSTACK_THEME_STORAGE_KEY = THEME_STORAGE_KEY;

  /* =====
  CODEMIRROR 6: LINE WRAPPING FOR CODE BLOCKS
  Listens for the CM6 pre-init event and enables line wrapping
  for content code blocks to prevent horizontal scrolling
  ===== */
  window.addEventListener('library-cm6::pre-init', function (event) {
    var detail = event.detail;
    var config = detail.editorViewConfig;
    var EditorView = detail.libEditorView;

    // Only apply line wrapping to content code blocks (not the main editor)
    if (detail.usage === 'content-code-block') {
      config.extensions.push(EditorView.lineWrapping);
    }
  });

  /* =====
  MARKDOWN-IT: SOFT LINE BREAKS
  Configures the Markdown editor to convert single newlines to <br> tags

```

```

    (GFM-style line breaks)
    ===== */
window.addEventListener('editor-markdown::setup', function (event) {
    event.detail.markdownIt.set({breaks: true});
});

/* =====
    LIGHT/DARK MODE TOGGLE BUTTON

    Features:
    - Works for both guests (localStorage) and logged-in users (server-side)
    - Applies saved preference immediately to prevent flash of wrong theme
    - Detects system color scheme preference as fallback for guests
    ===== */

/**
 * Checks if the guest has dark mode enabled based on localStorage
 * Falls back to system preference if no stored value exists
 * @returns {boolean} True if dark mode should be enabled
 */
function isGuestDarkModeEnabled() {
    var stored = localStorage.getItem(THEME_STORAGE_KEY);
    if (stored !== null) {
        return stored === 'true';
    }
    // Fallback: Check system color scheme preference
    return window.matchMedia && window.matchMedia('(prefers-color-scheme: dark)').matches;
}

/**
 * Determines if the current user is logged in
 * Checks for CSRF token AND user menu elements (both required)
 * @returns {boolean} True if user is logged in
 */
function isUserLoggedIn() {
    var csrfMeta = document.querySelector('meta[name="csrf-token"]');
    var hasToken = csrfMeta && csrfMeta.getAttribute('content');
    var hasUserMenu = document.querySelector('.dropdown-container [data-
shortcut="favourites_view"]');

```

```

    || document.querySelector('[href*="/logout"]');
    return !(hasToken && hasUserMenu);
}

/**
 * Applies the theme by toggling the 'dark-mode' class on <html>
 * @param {boolean} isDark - Whether to enable dark mode
 */
function applyTheme(isDark) {
    if (isDark) {
        document.documentElement.classList.add('dark-mode');
    } else {
        document.documentElement.classList.remove('dark-mode');
    }
}

// -----
// IMMEDIATE THEME APPLICATION (before DOMContentLoaded)
// Prevents flash of wrong theme by applying saved preference early
// -----
var currentlyDark = document.documentElement.classList.contains('dark-mode');
var guestPref = localStorage.getItem(THEME_STORAGE_KEY);

if (guestPref !== null) {
    if (!currentlyDark && guestPref === 'true') {
        document.documentElement.classList.add('dark-mode');
    } else if (guestPref === 'false' && currentlyDark) {
        document.documentElement.classList.remove('dark-mode');
    }
}

// -----
// CREATE TOGGLE BUTTON (after DOM is ready)
// -----
document.addEventListener('DOMContentLoaded', function () {
    var isDarkMode = document.documentElement.classList.contains('dark-mode');
    var loggedIn = isUserLoggedIn();

    // SVG icons for sun (light mode) and moon (dark mode)

```

```

var sunIcon = '<svg xmlns="http://www.w3.org/2000/svg" viewBox="0 0 24 24"><path d="M20
15.31 23.31 12 20 8.69V4h-4.69L12 .69 8.69 4H4v4.69L.69 12 4 15.31V20h4.69L12 23.31 15.31
20H20z" data-bbox="12 18 20 28"/>M12 18c-3.31 0-6-2.69-6-6s2.69-6 6-6 6 2.69 6 6-2.69 6-6 6"/></svg>';
var moonIcon = '<svg xmlns="http://www.w3.org/2000/svg" viewBox="0 0 24 24"><path d="M12
3a9 9 0 1 0 9 9c0-.46-.04-.92-.1-1.36a5.389 5.389 0 0 1-4.4 2.26 5.403 5.403 0 0 1-3.14-9.8c-
.44-.06-.9-.1-1.36-.1z" data-bbox="12 18 20 28"/>"/></svg>';

// Create the toggle button element
var button = document.createElement('button');
button.className = 'theme-toggle-fixed';
button.type = 'button';
button.title = isDarkMode ? 'Activate Light Mode' : 'Activate Dark Mode';
button.innerHTML = isDarkMode ? sunIcon : moonIcon;

// Handle click: server-side for logged-in users, client-side for guests
button.addEventListener('click', function () {
  if (loggedIn) {
    // Logged-in users: Submit form to BookStack's preference endpoint
    var csrfToken = '';
    var csrfMeta = document.querySelector('meta[name="csrf-token"]');
    if (csrfMeta) {
      csrfToken = csrfMeta.getAttribute('content') || '';
    }

    var form = document.createElement('form');
    form.method = 'POST';
    form.action = '/preferences/toggle-dark-mode';
    form.innerHTML =
      '<input type="hidden" name="_token" value="' + csrfToken + '">' +
      '<input type="hidden" name="_method" value="PATCH">' +
      '<input type="hidden" name="_return" value="' + window.location.href + '">';
    document.body.appendChild(form);
    form.submit();
  } else {
    // Guests: Toggle theme client-side and save to localStorage
    var nowDark = document.documentElement.classList.contains('dark-mode');
    var newMode = !nowDark;

    applyTheme(newMode);
  }
});

```

```

        localStorage.setItem(THEME_STORAGE_KEY, newMode.toString());

        // Update button appearance
        button.innerHTML = newMode ? sunIcon : moonIcon;
        button.title = newMode ? 'Activate Light Mode' : 'Activate Dark Mode';
    }
});

    document.body.appendChild(button);
});
})();
</script>

<!-- =====
MERMAID DIAGRAM RENDERING (ES MODULE)

Features:
- Automatic detection of mermaid code blocks (multiple selector formats)
- Theme synchronization with BookStack's dark/light mode
- Uses shared localStorage key with theme toggle button
- Theme changes apply on next page load (no live re-rendering)
===== -->
<script type="module">
    import mermaid from "https://cdn.jsdelivr.net/npm/mermaid@11/dist/mermaid.esm.min.mjs";

    // Use the shared storage key from the main script
    const STORAGE_KEY = window.BOOKSTACK_THEME_STORAGE_KEY || "bookstack-guest-dark-mode";

    /* =====
    THEME DETECTION UTILITIES
    ===== */

    /**
     * Checks if BookStack is currently in dark mode
     * @returns {boolean} True if 'dark-mode' class is present on <html>
     */
    function isBookStackDarkMode() {
        return document.documentElement.classList.contains("dark-mode");
    }

```

```

/**
 * Retrieves the stored theme preference from localStorage
 * @returns {boolean|null} true = dark, false = light, null = not set
 */
function getStoredThemeIsDark() {
  const stored = localStorage.getItem(STORAGE_KEY);
  if (stored === null) return null;
  return stored === "true";
}

/**
 * Saves the theme preference to localStorage
 * @param {boolean} isDark - Whether dark mode is enabled
 */
function storeThemeIsDark(isDark) {
  localStorage.setItem(STORAGE_KEY, isDark ? "true" : "false");
}

/**
 * Determines the Mermaid theme based on current BookStack mode
 * Uses actual DOM state (dark-mode class) for accurate theme selection
 * @returns {string} Mermaid theme name ("dark" or "default")
 */
function getMermaidTheme() {
  return isBookStackDarkMode() ? "dark" : "default";
}

/* =====
   MERMAID INITIALIZATION
   Disabled auto-start to allow manual control over rendering
   ===== */
mermaid.initialize({
  startOnLoad: false,
  securityLevel: "strict",
  theme: getMermaidTheme()
});

/* =====

```

DOM UTILITIES FOR MERMAID CODE BLOCKS

===== */

```
/**
 * Finds all elements containing Mermaid diagram source code
 * Supports multiple code block formats used by different editors
 * @returns {HTMLElement[]} Array of source elements
 */
function findMermaidSources() {
  const selectors = [
    "pre.mermaid",
    "pre > code.language-mermaid",
    "pre > code.lang-mermaid",
    "code.language-mermaid"
  ];

  const nodes = Array.from(document.querySelectorAll(selectors.join(", ")));

  // Normalize to parent <pre> element when applicable
  return nodes.map(n =>
    (n.tagName.toLowerCase() === "code" && n.parentElement?.tagName.toLowerCase() === "pre")
      ? n.parentElement
      : n
    );
}

/**
 * Extracts the text content (Mermaid code) from a source element
 * @param {HTMLElement} node - The source element
 * @returns {string} The trimmed Mermaid code
 */
function extractCodeFromNode(node) {
  const codeEl = node.tagName.toLowerCase() === "pre"
    ? (node.querySelector("code") || node)
    : node;

  return (codeEl.textContent || "").trim();
}
```

```

/**
 * Replaces the original code block with a Mermaid container div
 * Initially hidden to prevent flash of unstyled content
 * @param {HTMLElement} originalNode - The original code block element
 * @param {string} code - The Mermaid diagram code
 * @returns {HTMLElement} The new container element
 */
function replaceWithContainer(originalNode, code) {
  const container = document.createElement("div");
  container.className = "mermaid";
  container.textContent = code;
  container.style.visibility = "hidden";

  originalNode.replaceWith(container);
  return container;
}

/* =====
PAGE READY DETECTION
===== */

/**
 * Waits for the page to be fully loaded and rendered
 * Uses multiple techniques to ensure DOM is stable:
 * 1. Wait for 'load' event (or skip if already complete)
 * 2. Wait for two animation frames (ensures layout is calculated)
 * 3. Additional timeout for any async CSS/font loading
 */
async function waitForPageReady() {
  // Wait for window load event
  await new Promise(resolve => {
    if (document.readyState === "complete") return resolve();
    window.addEventListener("load", () => resolve(), {once: true});
  });

  // Wait for layout stabilization (two animation frames)
  await new Promise(r => requestAnimationFrame(() => requestAnimationFrame(r)));

  // Additional buffer for async resources

```

```

    await new Promise(r => setTimeout(r, 150));
  }

  /* =====
  MAIN RENDERING FUNCTION
  ===== */

  /**
   * Main entry point: finds, transforms, and renders all Mermaid diagrams
   * Theme is determined once at render time and applied consistently
   * Theme changes will take effect on next page load
   */
  async function renderAllMermaid() {
    const sources = findMermaidSources();

    // Even without diagrams, ensure theme state is stored for consistency
    if (!sources.length) {
      await waitForPageReady();
      storeThemeIsDark(isBookStackDarkMode());
      return;
    }

    // Transform code blocks into Mermaid containers
    const containers = [];
    for (const src of sources) {
      const code = extractCodeFromNode(src);
      if (!code) continue;

      const container = replaceWithContainer(src, code);
      // Store original code for potential future use
      container.setAttribute("data-original-code", code);
      containers.push(container);
    }

    await waitForPageReady();

    // Render all diagrams with the current theme
    for (const el of containers) {
      try {

```

```
        await mermaid.run({nodes: [el]});
        el.style.visibility = "visible";
    } catch (e) {
        el.style.visibility = "visible";
        console.error("Mermaid render failed for element:", el, e);
    }
}

// Store current theme state for next page load
storeThemeIsDark(isBookStackDarkMode());
}

// Start rendering process
renderAllMermaid();
</script>
```

functions.php

```
/opt/bookstack/bookstack/www/themes/custom/functions.php
```

```
<?php

// Import the ThemeEvents class which contains event constants for the theming system
use BookStack\Theming\ThemeEvents;
// Import the Theme facade to register event listeners
use BookStack\Facades\Theme;

/**
 * Register a listener for the CommonMark environment configuration event.
 * This hook is triggered when BookStack sets up the Markdown parser.
 *
 * CommonMark is the Markdown parsing library used by BookStack.
 */
Theme::listen(ThemeEvents::COMMONMARK_ENVIRONMENT_CONFIGURE, function ($environment) {

    // Merge custom configuration into the CommonMark environment
    $environment->mergeConfig([
        'renderer' => [
```

```
// Convert soft line breaks (single newlines) into <br> HTML tags
// By default, CommonMark ignores single newlines in Markdown.
// This setting makes single line breaks visible in the rendered output,
// which is useful for preserving line formatting in user content.
'soft_break' => "<br>",
]
]);

// Return the modified environment so other listeners can further customize it
return $environment;
});
```

oEmbeds in `<head>`

```
<style>
  .auto-embed {
    position: relative;
    width: 100%;
    max-width: 100%;
    margin: 1rem 0;
    overflow: hidden;
    background: #000;
  }

  .auto-embed.video {
    aspect-ratio: 16/9;
  }

  .auto-embed.audio {
    aspect-ratio: 16/5;
  }

  .auto-embed.code {
    aspect-ratio: 4/3;
    min-height: 400px;
  }
```

```

.auto-embed.square {
  aspect-ratio: 1/1;
}

.auto-embed iframe {
  position: absolute;
  inset: 0;
  width: 100%;
  height: 100%;
  border: 0;
}
</style>

<script>
document.addEventListener('DOMContentLoaded', () => {
  const providers = [
    // YouTube
    {
      regex: /youtu(?:be\.com\/watch?v=|\.be\/)([w-]+)/i,
      embed: id => `https://www.youtube-nocookie.com/embed/${id}`, type: 'video'
    },
    {
      regex: /vimeo\.com\/(\d+)/i,
      embed: id => `https://player.vimeo.com/video/${id}`, type: 'video'
    },
    {
      regex: /dailymotion\.com\/video\/([w-]+)/i,
      embed: id => `https://www.dailymotion.com/embed/video/${id}`, type: 'video'
    },
    {
      regex: /twitch\.tv\/videos\/(\d+)/i,
      embed: id => `https://player.twitch.tv/?video=${id}&parent=${location.hostname}`,
type: 'video'
    },
    {
      regex: /twitch\.tv\/([w-]+)$/i,
      embed: id => `https://player.twitch.tv/?channel=${id}&parent=${location.hostname}`,
type: 'video'
    },
  ],

```

```

{
  regex: /loom\.com\/share\/([\w-]+)/i,
  embed: id => `https://www.loom.com/embed/${id}`, type: 'video'
},
{
  regex: /streamable\.com\/([\w-]+)/i,
  embed: id => `https://streamable.com/e/${id}`, type: 'video'
},

// 🎧 Audio
{
  regex: /open\.spotify\.com\/(track|album|playlist|episode)\/([\w-]+)/i,
  embed: (t, id) => `https://open.spotify.com/embed/${t}/${id}`, type: 'audio'
},
{
  regex: /soundcloud\.com\/([\w-]+\/[\w-]+)/i,
  embed: path =>
`https://w.soundcloud.com/player/?url=https://soundcloud.com/${path}&color=%23ff5500&auto_play
=false&hide_related=true&show_comments=false&show_user=true&show_reposts=false&show_teaser=fal
se`,
  type: 'audio'
},

// 📄 Code - CodePen

// 📄 CodePen v2.0 Editor-Format (DIREKTES IFRAME!)
{
  regex: /codepen\.io\/editor\/([\w-]+)\/pen\/([\w-]+)/i,
  embed: (user, penId) => `https://codepen.io/editor/${user}/embed/${penId}?default-
tab=html%2Cresult&theme-id=dark&editable=true`,
  type: 'code'
},

// 📄 Klassisches CodePen-Format
{
  regex: /codepen\.io\/([\w-]+)\/(?:pen|full|details)\/([\w-]+)/i,
  embed: (user, penId) => `https://codepen.io/${user}/embed/${penId}?default-
tab=html%2Cresult&theme-id=dark&editable=true`,
  type: 'code'
}

```

```

},

// Andere Code-Plattformen
{
  regex: /codesandbox\.io\/s\/([\w-]+)/i,
  embed: id => `https://codesandbox.io/embed/${id}`, type: 'code'
},
{
  regex: /stackblitz\.com\/edit\/([\w-]+)/i,
  embed: id => `https://stackblitz.com/edit/${id}?embed=1`, type: 'code'
},
{
  regex: /jsfiddle\.net\/([\w-]+\.[\w-]+)/i,
  embed: path => `https://jsfiddle.net/${path}/embedded/result,js,html,css/`, type:
'code'
},
{
  regex: /gist\.github\.com\/([\w-]+)\.([\w-]+)/i,
  embed: (u, id) => `data:text/html,<script
src="https://gist.github.com/${u}/${id}.js"></script>`,
  type: 'code'
},

// 🗺️ Maps & Design
{
  regex: /figma\.com\/(file|design)\.([\w-]+)/i,
  embed: (t, id) =>
`https://www.figma.com/embed?embed_host=bookstack&url=https://www.figma.com/${t}/${id}`,
  type: 'code'
},
{
  regex: /google\.com\/maps\/embed\?pb=([^\s]+)/i,
  embed: pb => `https://www.google.com/maps/embed?pb=${pb}`, type: 'video'
},
];

document.querySelectorAll('p').forEach(p => {
  const text = p.textContent.trim();
  if (!/^https?:\/\/\S+$/i.test(text)) return;

```

```
for (const {regex, embed, type} of providers) {
  const match = text.match(regex);
  if (!match) continue;

  const src = embed(...match.slice(1));
  p.outerHTML = `

## And this must be added to docker-compose.yml



Under environment:



```
- ALLOWED_IFRAME_SOURCES=https://*.codepen.io https://codepen.io https://*.jsfiddle.net
https://jsfiddle.net https://*.codesandbox.io https://codesandbox.io https://open.spotify.com
https://gist.github.com https://*.youtube.com https://youtube.com https://www.youtube-
nocookie.com https://*.vimeo.com https://player.vimeo.com https://*.dailymotion.com
https://*.twitch.tv https://player.twitch.tv https://clips.twitch.tv https://*.tiktok.com
https://*.loom.com https://*.wistia.com https://fast.wistia.net https://streamable.com
https://*.vidyard.com https://rumble.com https://*.soundcloud.com https://w.soundcloud.com
https://embed.music.apple.com https://*.deezer.com https://widget.deezer.com
https://*.bandcamp.com https://*.mixcloud.com https://*.audiomack.com https://anchor.fm
https://*.transistor.fm https://*.stackblitz.com https://*.replit.com https://*.glitch.com
https://jsbin.com https://*.observablehq.com https://carbon.now.sh https://ascinema.org
https://platform.twitter.com https://*.instagram.com https://*.reddit.com
https://*.linkedin.com https://assets.pinterest.com https://*.figma.com https://*.canva.com
https://docs.google.com https://*.pitch.com https://prezi.com https://*.slideshare.net
https://speakerdeck.com https://maps.google.com https://*.openstreetmap.org
https://*.notion.so https://*.notion.site https://*.airtable.com https://*.typeform.com
https://form.typeform.com https://calendly.com https://*.miro.com https://excalidraw.com
https://*.diagrams.net https://whimsical.com https://*.lottiefiles.com https://lottie.host
https://*.sketchfab.com
```


```

<https://www.youtube.com/watch?v=UclrVWafRAI>