

Portfolio 13. Grundlegende Überarbeitung eigener Webpräsenz: mjmrozek.de, mjmrozek.com, mjmrozek.eth

Tags: [#portfolio](#)

Startdatum: 17.11.2023

Autor: Matthäus J. Mrozek

Kontext:

- Mitte 2021 Beginn Selbständigkeit als Freiberufler im IT-Bereich und nicht viel Ahnung von Webdesign usw.
- Webpräsenz sollte mit minimalem Aufwand realisiert werden und hauptsächlich meine Person, Services und Portfolio portraituren
 - WordPress als CMS und Hosting usw. auch via wordpress.com
 - Domain und E-Mail auch via wordpress.com
- Zwischenzeitlich Wechsel zu IONOS' unmanaged WP ...
- gewisse Angst (begründet?), dass deutsche oder englische Schwesterseite nicht erreichbar, durch Fehler in Plugin oder Kompatibilitätsproblemen nach Aktualisierung o.ä.
- Zunehmend viel Aufwand für Pflege zweier WordPress-Seiten, was leicht auch eine multilinguale, statische Seite

Ziele:

- bessere Skalierbarkeit der Serverleistung / schlankere, statische Webseite
- einfacheres, günstigeres Backupsystem
- Möglichkeit der Migration
- höhere Geschwindigkeit
 - Google Pagespeed Wert 100
- Erhöhung Sicherheit
- Senkung Pflegeaufwand
- zwecks Stabilität gesamter Webpräsenz/ Erreichbarkeit von Portfolio & Blog auch bei Serverausfall o.ä.

Plan:

- Umzug von IONOS zu LINODE
 - Instanz Nanode? (1GB Shared CPU, Backupsystem, einfache Migration, vielfältige und einfache Skalierbarkeit,
- Wechsel von Marianne-Theme zum Block-Theme „Twenty Twenty-Four“, da DAS Standard-Theme von Entwicklern des CMS WP selbst
- Einrichtung LiteSpeed-Cache für schnellere Ladezeiten
- Einrichtung von QuickCloud als CDN-Lösung für global bessere

Verfügbarkeit

- Routing der Domain über CloudFlare
- Einrichtung von 2FA, Änderung Adresse [xxx.de/wp-admin](#) in [xxx.de/das-kenne-nur-ich](#), Einrichtung Sperre Datenbankänderungen
- Einrichtung einfacher Version von Blog und Portfolio auf IPFS / Ethereum
- Hugo Webseite auf Englisch lokal
- Verbindung von GitHub mit Netlify und Deployment mit eigener Domain
- Hugo Webseite multilingual
- Oracle Cloud Compute einrichten mit Ubuntu, Remote Desktop und dort VSC, da Probleme mit Silicon-Builds bei Fleek
- Ersetzen von Netlify durch Fleek und nur noch mjmrozek.eth und IPFS-Hosting / Pinning
- Auch dezentraler E-Mail-Dienst
 - [Ethermail.io](#)?
- Feintuning der Webseite
 - Captions (kleiner, kursiv, näher am Bild)
 - Kommentare
 - Contact-Formular
 - Eigenes Favicon
 - Bilder Link Full Page
 - Profilfoto
 - Bilder optimieren

Umsetzung / Protokoll:

- 4 Hauptschritte: [Create a Decentralized Website | ENS Support](#)
- im Zentrum meiner Online-Identität bzw. web3-Identität überhaupt soll der Nickname „mjmrozek“ stehen
- Einrichtung von Metamask und Kopplung an Google Chrome
- Kauf von ETH im Wert von ca. 100 EUR via Kreditkarte
- Erwerb von web3-username „mjmrozek“ via ENS (<https://app.ens.domains/>) für 5 Jahre für ca. 50 EUR.
 - Eintragen von 5 Records (weitere werden später eingetragen)
 - url: <https://mjmrozek.com>
 - name (nickname): mjmrozek
 - Twitter: mjmrozek
 - location: Germany
 - avatar: mein übliches Profilbild
 - Suche nach Anleitung fürs Erstellen einer dezentralen Webseite
 - <https://support.ens.domains/en/articles/7890637-create-a-decentralized-website>
 - Bauen einer Webseite (einfachere Version meiner WordPress-Seite)
 - Prüfung von web3-website-builder Heaven's Tool

<http://heavens.tools/>

- „create new project“
 - website and landing page
 - start from scratch, da mir die Vorlagen nicht gefallen
 - own style / block
- Prüfung von HUGO
 - fleek.co
 - Aufrufen in Chrome, da dort MetaMask-Plugin
 - Autorisieren via MetaMask
 - Add new site
 - Connect with GitHub
 - Authorize repo „mjmrozek/hugo“
 - Hosting Service: IPFS
 - Framework: Hugo
 - Probleme bez. extended version und von Fleek angebotenen Hugo-v0.85 und außerdem Apples Silicon Image ...
- Installation von HUGO auf meinem Mac
 - Installation von Homebrew <https://www.youtube.com/watch?v=4r4yG7NJgZc>
 - <https://brew.sh>
 - Kopieren von `/bin/bash -c "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/HEAD/install.sh)"`
 - Einfügen per Terminal
 - Hinzufügen von Homebrew zum Path
 - Überprüfung via Befehl „help brew“
 - Installation von Hugo <https://www.youtube.com/watch?v=cev4gGE41e8>
 - Terminal
 - „install hugo“
 - „hugo version“
 - Installation von Go via Terminal „brew install go“
 - „hugo new site mj.eth“
 - Wechsel zu VSC
 - neues Terminal und „hugo server“
 - https-Adresse nach Safari: Page Not Found, da kein Theme
 - Suche nach passendem Theme via gohugo.io
 - Beautiful Hugo
 - Suche nach Installationshinweisen auf GitHub
 - in VSC „git clone https://github.com/halogenica/beautifulhugo.git themes/beautifulhugo“
 - Eintrag in Config-File „hugo.toml“ Name des neuen Themas
 - „theme = beautifulhugo“
 - Prüfung neuen Themen via lokalem Server

- Terminal „hugo server“ zwecks neuer http
 - Einfügen in Safari
 - neues Theme aktiv
- da ich immer zuerst auf Deutsch schreibe usw. kümmere ich mich zunächst um mjmrozek.de und setze auf LINODE eine WordPress-Seite komplett neu auf und kopiere alle Inhalte dorthin
- Installation Hugo
- Installation Git
- Wechsel in meinen Entwicklungsordner
 - „cd ~/dev“
- Erstellen neuer Hugo-Seite
 - „hugo new site mjmrozek.eth-pm“
- innerhalb von dev/mjmrozek.eth-pm Erstellen des Themes PaperMod
 - „git clone https://github.com/adityatelange/hugo-PaperMod themes/PaperMod --depth=1“
- Update des Themes
 - „cd themes/PaperMod“
 - „git pull“
- VSC
 - öffnen von relevantem Ordner (dev/mjmrozek.eth-pm)
 - source control
 - publish to GitHub
 - alle Ordner auswählen
 - hugo.toml (config file)
 - neuer Eintrag „theme: „PaperMod““
 - source control
 - Changes auswählen per +
 - Commit
 - unten Synchronize Changes
- Fleek als IPFS-Hosting bzw. Pinning
 - mit MetaMask verbinden
 - add new Site
 - Configure the Fleek app on GitHub
 - Zugriff nur auf eine private Repo „mjmrozek.eth-pm“
 - Auswahl Hosting-Plattform
 - IPFS Hosting, da ich ENS nutzen will
 - Auswahl Framework
 - Hugo
 - Auswahl Build
 - aktuellste
 - Probleme beim Deployment, da config-Datei nicht aufzufinden

- benenne in VCL in Root hugo.toml um in config.toml und commit und sync
- Fleek synchronisiert mit GitHub und startet von selbst zweiten Deploymentversuch erfolgreich
- vergessen Seite überhaupt zu bauen, also in VCL Terminal „hugo“
- Leider funktioniert es nicht: nach Fleeks Deployment erhalte ich eine https-Adresse zwecks Test meiner Seite, aber es wird nur die Startseite angezeigt; im Protokoll des Builds sehe ich den Fehler, der auf fehlendes Submodule verweist
 - deshalb im Arbeitsordner „touch .gitmodules“

—
nochmal Hugo

- u.U. Installation von Homebrew
- Terminal
 - brew install hugo
 - Wechsel in Ordner „dev“
 - „hugo new site mjm-hugo-v02“
- Suche nach geeignetem Theme
 - themes.gohugo.io
 - <https://themes.gohugo.io/themes/hugo-papermod/>
 - <https://github.com/adityatelange/hugo-PaperMod/wiki/Installation>
 - Methode 1: git clone https://github.com/adityatelange/hugo-PaperMod themes/PaperMod --depth=1
- Terminal
 - in vorhin neu erstelltem Ordner Einfügen von Code der Methode 1, um Theme „papermod“ als neues Theme zu installieren
 - Wechsel zu VSC per „code“
- VSC
 - umbenennen von „hugo.toml“ in „config.toml“
 - config.toml und darin
 - title als ‚Matthäus J. Mrozek‘
 - in Zeile 4 neues Theme definieren
 - theme = ‚PaperMod‘
 - Terminal
 - hugo server (für Vorschau)
 - Ordner „Content“
 - alle Inhalte, wie Posts, Seiten
 - Ordner „Layouts“
 - für Theme-Overrides
 - Ordner „Themes“ besitzt eigenes Layout-Ordner
 - Ordner „Static“ ist für Bilder
 - neues Post

- Terminal
 - hugo new posts/hub.md (hub = name des neuen Posts)
- Ordner contents/posts
 - hub.md
 - title ändern in richtigen Titel, der angezeigt wird
 - draft auf false, falls vor, dass veröffentlicht
 - Text des Posts selbst als Markdown unter +++
 - dieser Post soll immer oben sein, deshalb „weight = 1“

```
content > posts > ↴ hub.md
1  +++
2  title = 'Personal or professional website'
3  date = 2023-11-24T16:19:30+01:00
4  draft = false
5  weight = 1
6  +++
```

- Bilder, z.B. als Feature Image
 - Bild ablegen in Ordner „static“ oder darin in Unterordner „img“
 - unter weight oder draft
- config.toml
 - Menü

```
[[menu.main]]
```

```
name = "Blog"
```

```
url = "/"
```

```
weight = 1
```

```
[[menu.main]]
```

```
name = "Portfolio"
```

```
url = "/portfolio/"
```

```
weight = 2
```

```
[[menu.main]]
```

```
name = "Business"
```

```
url = "/business/"
```

```
weight = 3
```

- weight bestimmt die Reihenfolge
- da ich doch mit der Zeit sehr viele Posts haben werde, füge ich auch noch „Archives“ und „Tags“ ein.

```

6    [[menu.main]]
7    name = "Blog"
8    url = "/"
9    weight = 1
10
11   [[menu.main]]
12   name = "Portfolio"
13   url = "/portfolio/"
14   weight = 4
15
16   [[menu.main]]
17   name = "Business"
18   url = "/business/"
19   weight = 5
20
21   [[menu.main]]
22   name = "About"
23   url = "/about/"
24   weight = 6
25
26   [[menu.main]]
27   name = "Legal"
28   url = "/legal/"
29   weight = 7
30
31   [[menu.main]]
32   name = "Archives"
33   url = "/archives/"
34   weight = 2
35
36   [[menu.main]]
37   name = "Tags"
38   url = "/tags/"
39   weight = 3

```

- dazu erstelle ich im Content-Ordner „archives.md“ mit folgendem Eintrag:


```

content > ↴ archives.md
1    +++
2    title = "Archives"
3    layout = "archives"
4    url = "/archives/"
5    summary = "Archives"
6    +++
7

```

- das Layout für diese spezielle Seite wird übernommen aus dem Layouts-Ordner

```

themes > PaperMod > layouts > _default > ↵ archives.html > ...
1  [{{- define "main" }}]
2
3  <header class="page-header">
4    <h1>
5      {{ .Title }}
6      {{- if (.Param "ShowRssButtonInSectionTermList") }}
7      {{- $rss := (.OutputFormats.Get "rss") }}
8      {{- if (eq .Kind `page`) }}
9        {{- $rss = (.Parent.OutputFormats.Get "rss") }}
10     {{- end }}
11     {{- with $rss }}
12       <a href="{{.RelPermalink }}" title="RSS" aria-label="RSS">
13         <svg xmlns="http://www.w3.org/2000/svg" viewBox="0 0 24 24" fill="none" stroke="currentColor" stroke-width="2"
14           stroke-linecap="round" stroke-linejoin="round" height="23">
15           <path d="M4 11a9 9 0 0 1 9 9" />
16           <path d="M4 4a16 16 0 0 1 16 16" />
17           <circle cx="5" cy="19" r="1" />
18         </svg>
19       </a>
20     {{- end }}
21   {{- end }}
22 /{{- end }}

```

- Änderungen am Layouts-Ordner würden durch ein Update verlorengehen. Deshalb lokale Kopie mit exakt selber Ordnerstruktur

```
✓ MJM-HUGO-V02
  > archetypes
  > assets
  > content
  > data
  > i18n
  ✓ layouts / _default
    <> archives.html
  ✓ static
  ✓ themes / PaperMod
    > .github
    > assets
    > i18n
    > images
  ✓ layouts
    ✓ _default
      > _markup
      <> archives.html
```

- unter Layouts (wo nichts sein sollte) „_default“ und darin die Datei „archives.html“ erstellen und Inhalt aus der Archives-Datei des Themes hineinkopieren. Änderungen an dieser Archives-Datei würden angewendet werden, es würde also das Standard-archives überschrieben. In meinem Fall der Befehl für eine Linie unter dem Titel

```

layouts > _default > < archives.html > < header.page-header > < h1 > < hr
1  {{- define "main" }}
2
3  <header class="page-header">
4    <h1>
5      {{ .Title }}
6    <hr>
7    {{- if (.Param "ShowRssButtonInSectionTermList") }}
8    {{- $rss := (.OutputFormats.Get "rss") }}
9    {{- if (eq .Kind `page`) }}
10     {{- $rss = (.Parent.OutputFormats.Get "rss") }}
11    {{- end }}
12    {{- with $rss }}
13    <a href="{{.RelPermalink }}" title="RSS" aria-label="RSS">
14      <svg xmlns="http://www.w3.org/2000/svg" viewBox="0 0 24 24" fill="none" stroke="currentColor" stroke-widt
15        stroke-linecap="round" stroke-linejoin="round" height="23">
16        <path d="M4 11a9 9 0 0 1 9 9" />
17        <path d="M4 4a16 16 0 0 1 16 16" />
18        <circle cx="5" cy="19" r="1" />
19      </svg>
20    </a>
21    {{- end }}

```

- den Menüpunkt und die Seite „Categories“ ersetze ich lieber durch „Tags“
 - in den Posts werden Tags so eingetragen:

```

content > posts > 📄 hub.md
1  +++
2  title = 'Personal or professional website?'
3  date = 2023-05-20T16:19:30+01:00
4  draft = false
5  weight = 1
6  tags = ["clarification"]
7  +++
8

```

- für Breadcrombs und Lesezeit in der config.toml:

```
⚙ config.toml
1  baseURL = 'https://example.org/'
2  languageCode = 'en-us'
3  title = 'Matthäus J. Mrozek'
4  theme = 'PaperMod'
5
6  [params]
7    showBreadcrumbs = true
8    showReadingTime = true
9
```

- für einen Titel auf der Startseite „Blogging & IT Solutions“ und meine SocialMediaKonten folgendes:

```

❏ config.toml
1  baseURL = 'https://example.org/'
2  languageCode = 'en-us'
3  title = 'Matthäus J. Mrozek'
4  theme = 'PaperMod'
5
6  [params]
7      showBreadcrumbs = true
8      showReadingTime = true
9      showShareButtons = true
10     showPostNavLinks = true
11
12     [params.homeInfoParams]
13     Title = "Blogging & IT Solutions"
14     Content = """"This website is a hub, but not in the way of
15     amalgamating various areas of life, or fields, within itself. For
16     clients to swiftly access my portfolio, which predominates
17
18
19     [[params.socialIcons]]
20     name = "email"
21     url = "mailto:mjmrozek@gmail.com"
22
23     [[params.socialIcons]]
24     name = "twitter"
25     url = "https://twitter.com/mjmrozek"
26
27     [[params.socialIcons]]
28     name = "facebook"
29     url = "https://www.facebook.com/matthaeusjmrozek"
30
31     [[params.socialIcons]]
32     name = "instagram"
33     url = "https://www.instagram.com/mjmrozek/"
34

```

- Seiten, nicht Posts, für die Bereiche „Portfolio“, „About“ usw.
- per Befehl „hugo“ die Webseite endlich bauen
 - Ordner „public“ wird automatisch erstellt und darin alle Dateien fürs Veröffentlichen
 - Ordner kann gelöscht und erneut per „hugo“ erstellt werden
- im Arbeitsordner
 - git init
 - erstellt leeres Repo
 - touch .gitmodule

Versuche nochmals nach Anleitung von Davide <https://malwarewerewolf.com/posts/deploy-a-decentralized-hugo-static-website-with-ipfs/> mit Fleek und einem IPFS-Pinning. Da Netlify einwandfrei meine Seite deployed hat, behalte ich mir diese Lösung als Backup bei. Deshalb arbeite ich ab jetzt mit einem Duplikat weiter

- erstelle lokal neuen Ordner
 - mkdir mjm-hugo-v02-duplicate
- clone v02 darein
 - git clone https://github.com/deinusername/mjm-hugo-v02.git
mjm-hugo-v02-duplicate
- wechsle in den neuen Ordner
 - cd mjm-hugo-v02-duplicate
- lösche die `.git`-Verzeichnisse, um die bestehende Versionskontrolle zu entfernen
 - rm -rf .git
- Initialisierung des neuen Repos

- GitHub Repo mjmrozek-hugo-v02-duplicate
 - Actions
 - Neuer eigener Workflow

NEIN, erst Zugriff auf Linux und dazu Einrichtung von Oracle Cloud, dann Remote Desktop, dort VSC, dann Hugo dort bauen, da Fleek Probleme mit Silicon-Builds hat

- Bei oracle.com registrieren
 - Servicelinks
 - Gepinnt
 - Instanzen Compute
 - Ubuntu 22.04
 - Ampere 6 GB RAM
 - SSH herunterladen
 - Abrechnung & Kostenmanagement
 - Budgets
 - Festlegen auf Warnung, wenn 50 % erreicht von 5 USD / mtl.
 - Compute / Instanzen / Instanzdetails / Konsolenverbindung
 -

Bilder sollen einen dünnen schwarzen Rahmen haben

- Dazu neuer Ordner unter Root
 - Assets / css / extended
 - Darin „borders.css“
 - Darin .(site root)

```
assets > css > extended > # borders.css > ...
1  img {
2      border: 1px solid black; /* Stellt einen dünnen schwarzen Rahme
3  }
4  |
```

- Neuer Shortcode
 - Imageg.html
 -
- Bilder jetzt einzufügen mit (am Beispiel des Logos für meine Firma)
 - {{< image src="/img/mmits-it-logo.png" alt="Alt-Text für das Bild" class="no-border" >}}

```
rm -rf /users/matthaeusjohann/dev/mjm-hugo-v02
```

```
git clone https://github.com/mjmrozek/mjm-hugo-v02.git
```

Funktioniert!!!

Rahmen um Bilder mit Ausnahme für manche Bilder:

In Astes / css / extended / custom_css1.css

```
/* Rahmen für alle Bilder */
img {
    border: 1px solid black;
}
```

```
/* Ausnahme für Bilder ohne Rahmen */
.no-border {
    border: none;
```

}

```
/* custom_css1.css */
img {
  border: 1px solid black;
}

img.no-border {
  border: none;
}
```

-

In layouts / shortcakes / image.html

```

```

-

Wenn Bild ohne Rahmen dann so einfügen:

```
{{< image src="/img/mmits-it-logo.png" alt="Alt-Text für das Bild" class="no-
border" >}}
```

Mit Rahmen:

```
![Alt-Text für das Bild](/img/about-earphones-1.jpg)
*Fig. 5: My Tin Hifi T2 in-ear headphones*
```

Vielsprachigkeit


```

content/
├─ english/
|   ├─ about.md
|   ├─ business.md
|   ├─ legal.md
|   ├─ portfolio/
|   ├─ posts/
|   ├─ archives.md # Englische Version der Archives-Seite
|   └─ search.md   # Englische Version der Suchseite
└─ deutsch/
    ├─ about.md
    ├─ business.md
    ├─ legal.md
    ├─ portfolio/
    ├─ posts/
    ├─ archives.md # Deutsche Version der Archives-Seite
    └─ search.md   # Deutsche Version der Suchseite

```

Obere Struktur usw. erst in einem späteren Schritt.

Ich beginne mit der config.toml

```

baseURL = 'https://mjmrozek.com/'
title = 'Matthäus J. Mrozek'
theme = 'PaperMod'

```

```

# Definiere Standard-Sprache
defaultContentLanguage = "en" # Standardmäßig Englisch
defaultContentLanguageInSubdir = true # Platzierung der Standard-Sprache in
einem Unterordner

```

```

[languages]
[languages.en]
  languageName = "English"
  contentDir = "content/english"
  weight = 1
  languageCode = "en-us"
  title = "Matthäus J. Mrozek - English"

```

```

[languages.de]
  languageName = "Deutsch"

```

```
contentDir = "content/deutsch"  
weight = 2  
languageCode = "de-de"  
title = "Matthäus J. Mrozek - Deutsch"
```

```
[params]  
showPostNavLinks = true  
showShareButtons = true  
showBreadCrumbs = true  
mainSections = ["posts"]
```

```
[params.homeInfoParams]  
Title = "Blogging & IT Solutions"  
Content = ""  
This website is a hub, but not in the way a switch star-connects  
network nodes; it's more like a person in an immutable center,  
seemingly chaotically  
amalgamating various areas of life, or fields, within itself. [Read more...](/  
posts/hub/)  
""
```

```
# Social Icons bleiben gleich, da sie sprachunabhängig sind  
[[params.socialIcons]]  
# ... (deine vorhandenen Social-Icons)
```

```
# Menüeinträge für Englisch  
[[menu.main]]  
name = "Blog"  
url = "/posts/"  
weight = 1
```

```
[[menu.main]]  
name = "Portfolio"  
url = "/portfolio/"  
weight = 2
```

```
[[menu.main]]  
name = "Business"  
url = "/business/"  
weight = 3
```

```
[[menu.main]]  
name = "About"  
url = "/about/"  
weight = 4
```

```
[[menu.main]]
```

name = "Legal"
url = "/legal/"
weight = 5

[[menu.main]]
name = "Archives"
url = "/archives/"
weight = 6

[[menu.main]]
name = "Tags"
url = "/tags/"
weight = 7

Menüeinträge für Deutsch
[[languages.de.menu.main]]
name = "Blog"
url = "/de/posts/"
weight = 1

[[languages.de.menu.main]]
name = "Portfolio"
url = "/de/portfolio/"
weight = 2

[[languages.de.menu.main]]
name = "Geschäft"
url = "/de/business/"
weight = 3

[[languages.de.menu.main]]
name = "Über"
url = "/de/about/"
weight = 4

[[languages.de.menu.main]]
name = "Rechtliches"
url = "/de/legal/"
weight = 5

[[languages.de.menu.main]]
name = "Archive"
url = "/de/archives/"
weight = 6

[[languages.de.menu.main]]
name = "Tags"

url = "/de/tags/"
weight = 7

Ich: Die Seitentitel müssen nicht mit -English / - Deutsch enden, deshalb lösche ich das und die Reihenfolge der Menüeinträge ist auch nicht richtig, aber leicht per weight zu beheben

Jetzt erst content / deutsch bzw English:

Erstelle unter content 2 Ordner „deutsch“ und „english“

- Posts, About, ... alles in Ordner „English“ verschieben
- Englisches habe ich ja schon, also kopiere ich es und füge die ganzen Dateien und Ordner in „deutsch ein“:

21.12.2023

- Feintuning
 - Viele Bilder sollen beim Draufklicken in voller Größe erscheinen
 - Dazu Shortcode

```
layouts > shortcodes > <> lightbox.html > ...  
1 <!-- layouts/shortcodes/lightbox.html -->  
2 <a href="{{ .Get "src" }}" target="_blank">  
3 | 
```

- Einfügen per: `{{< lightbox src="/img/p11.1.jpg" alt="Alt-Text">}}`
 - Also ersetze ich `![Alt-Text](/img/p13.1.jpeg)` dadurch
- Eintrag von „summary“ ins Front Matter, dadurch wird auf Vorschau-Elementen meine Zusammenfassung gewählt und nicht automatisch der erste Text
- Entfernen von „:“ nach Überschriften und Verringerung des Abstandes zum Text darunter
- Probleme / Unstimmigkeiten mit der Datumsanzeige
 - In deutschen Posts stimmen Tag und Jahr aber es wird immer Januar angezeigt
 - Jetzt fällt mir auch auf, dass deutsche Posts auch englische Titel haben, also in der Adresszeile; das kommt von der Bezeichnung der md-Dateien; deutsche und englische müssen aber exakt gleich lauten, da sonst ein Wechsel zur jeweils anderen Sprache für einen Besucher nicht geht; als Kompromiss benenne ich alle Posts entsprechend des Datums und so ist die Adresse für den Post „Berufliche oder private ...“ `...de/posts/2022-12-22/`

- Für die andere Darstellung des Datums (11. Januar 2023 / January 11, 2023) ist ein Code zuständig in layouts / partials / post_meta.html; ich ersetze ihn durch eine Modifikation (rot = falsch; grün = richtig:

```
{{- if not .Date.IsZero -}}
  {{- $dateFormat := "January 2, 2006" -}} {{/* Standardformat */}}
  {{- if eq .Site.Language.Lang "de" -}}
    {{- $dateFormat = "2. Januar 2006" -}} {{/* Deutsches Format */}}
  {{- end -}}
  {{- $scratch.Add "meta" (slice (printf "<span title='%s'>%s</span>" (.Date)
(.Date | time.Format (default $dateFormat site.Params.dateFormat)))) }}
{{- end }}
```

```
{{- if not .Date.IsZero -}}
  {{- $dateFormat := .Site.Params.dateFormat | default "January 2, 2006" -}}
  {{- if eq .Site.Language.Lang "de" -}}
    {{- $dateFormat = .Site.Params.dateFormatDe | default "2. Januar 2006" -}}
  {{/* Deutsches Format */}}
  {{- end -}}
  <span title="{{ .Date | time.Format $dateFormat }}">{{ .Date | time.Format
$dateFormat }}</span>
{{- end -}}
```