

Hugo (HTML)

Bernd Plumhoff

2025-04-21

“I am always doing things I can't do, that's how I got to do them.”

Pablo Picasso

1 Abstract

Ich betreibe keinen Webshop und benötige keine Datenbank oder automatische Aktualisierung von Webseiten. Deshalb habe ich meine Blogs `bernd-plumhoff.de` und `sulprobil.de` mit Hugo erstellt. Hugo ist sehr effizient. Die Webseiten werden schnell übersetzt. Die generierten Seiten werden schnell geladen und sind sowohl auf PCs wie auch auf Smartphones gut zu lesen. Eine englischsprachige Einführung zu Hugo finden Sie hier: [Hugo Quick Start](#).

2 Voraussetzungen

1. Sie brauchen eine eigene Website mit Provider.
2. Sie müssen Hugo installieren, siehe [Hugo Quick Start](#).
3. Ich habe auch das Komprimierprogramm `7zip` installiert, um meine generierten Webseiten einfach zu packen und auf meine Website zu übertragen. Irgendwann mache ich das auch wieder mit `FileZilla`.
4. Normalerweise wird auch empfohlen, die Versionsführung `Git` zu installieren. Ich habe darauf zunächst verzichtet, weil ich meine Versionen in der Cloud führe.
5. Man sollte ein gutes Grundwissen von `HTML` oder `Markdown` Dateien haben und `Batchfiles` ändern und ausführen können.
6. Als Thema wählte ich das einfache `Ghostwriter` mit der `config.toml` Datei:

```
baseurl = "https://www.berndplumhoff.de/"
title = "Bernd Plumhoff"
theme = "ghostwriter"
```

[Params]

```
mainSections = ["post"]
intro = true
headline = "Bernd Plumhoff Blog"
copyright = "(C) (P) 2025 by Bernd Plumhoff"
defaultContentLanguage = 'de'
description = "Mein englischsprachiger Blog: https://www.sulprobil.de"
email = "mail@berndplumhoff.de"
opengraph = true
pygmentsCodeFences = true
pygmentsStyle = "solarized-light"
dateFormat = "Mon, Jan 2, 2006"
```

[Permalinks]

```
post = "[:filename/]"
```

[markup]

[markup.highlight]

```
anchorLineNos = false
codeFences = true
guessSyntax = false
hl_Lines = ''
lineAnchors = ''
lineNoStart = 1
lineNos = false
lineNumbersInTable = true
noClasses = true
style = 'vs'
tabWidth = 2
```

3 Markdown Dateien

Mich interessiert HTML nicht besonders. Ich arbeite lieber auf der Meta-Ebene Markdown.

Da kann ich meinen Fließtext einfach eingeben, um *kursive* Texte muss ich links und rechts ein einfaches Sternchen * setzen, **fett gesetzte** Texte benötigen links und rechts zwei Sternchen **.

Geschützten Text, den ich so sehen will wie ich ihn eingabe, muss ich mit ‘ ‘ ‘ beginnen und beenden.

3.1 Grafiken

Grafiken oder Images speichere ich als .png oder .jpg Dateien im Unterverzeichnis static. Mit dem Befehl

```
! [Grafik1] (/grafik1.png)
```

füge ich sie in eine Webseite ein.

3.2 Tabellen

Tabellen erzeuge ich einfach mit dem Markdown Table Generator:

Beispiel:

Quellcode:

```
[//]: # ( https://www.tablesgenerator.com/markdown_tables )
---
|          | Values | Percent |
|-----:|-----:|-----:|
|          |      11 |      1.80 |
|          |      45 |      7.36 |
|          |     555 |     90.83 |
| **Sum** | **611** | **99.99** |
---
```

Ergebnis:

	Values	Percent
	11	1.80
	45	7.36
	555	90.83
Sum	611	99.99

Natürlich kann man die Markdown-Tabelle auch direkt von Hand eingeben. Die Tabelle beginnt und endet jeweils mit ---, und mit dem Doppelpunkt : in der Unterstreichung der Titelzeile kann man festlegen, ob die Spalte links- oder rechtsbündig ausgerichtet werden soll.

3.3 Interner Link

Einen internen Link gebe ich beispielsweise mit

```
[sbRandTriang]({{< ref "/post/excel_vba_eine_sammlung_de.md" >}} "sbRandTriang")  
ein.
```

3.4 Externer Link

Einen externen Link definiere ich ganz am Ende meiner Markdown-Datei beispielsweise mit

```
[Hugo Quick Start]: https://gohugo.io/getting-started/quick-start/
```

Den oder die Verweis(e) darauf gebe ich dann beispielsweise mit

```
(externer Link!) [Hugo Quick Start]
```

ein. Den etwas unschönen Hinweis "(externer Link!)" füge ich lediglich aus rechtlichen Überlegungen ein, weil ich mit Markdown kein Target `_blank` erzeugen kann.

3.5 Syntax Hervorhebung (Highlighting)

Syntax hebe ich mit `Chroma` hervor.

In der oben genannten Datei `config.toml` im Website-Verzeichnis habe ich:

```
[markup]  
  [markup.highlight]  
    anchorLineNos = false  
    codeFences = true  
    guessSyntax = false  
    hl_Lines = ''  
    lineAnchors = ''  
    lineNoStart = 1  
    lineNos = false
```

```
lineNumbersInTable = true
noClasses = true
style = 'vs'
tabWidth = 2
```

Und in meinen Webseiten nutze ich dann z. B.

Start String	Für
```VB.net	VBA Code
```Batchfile	Batch Datei
```markdown	Markdown Datei
```Perl	Perl Code
```SQL	SQL Code

### 3.6 Gliederungspunkte

Wenn Sie Gliederungspunkte in Ihren Text einfügen wollen wie z. B.

- Erster Gliederungspunkt
- Zweiter Gliederungspunkt
- Dritter Gliederungspunkt

dann geben Sie einfach ein:

- Erster Gliederungspunkt
- Zweiter Gliederungspunkt
- Dritter Gliederungspunkt

## 4 Einfaches Editieren und Testen der Webseiten

Mit Hugo ist es sehr einfach, die eigenen Webseiten schnell in HTML zu übersetzen und zu testen.

Der Befehl `hugo -D` übersetzt eine Website vollständig. Die HTML-Dateien werden im Unterverzeichnis `public` erzeugt.

Mit `hugo serve` werden die Webseiten on-the-fly übersetzt und auf der lokalen Seite `http://localhost:1313/` gezeigt. Hugo wartet auf mögliche Änderungen und zeigt diese sofort an.

Mit dem unten gezeigten Batchfile `run_hugo.bat` lösche ich zunächst vollständig das Verzeichnis `public` und übersetze dann die gesamte Website

neu und komprimiere diese Übersetzung im Unterverzeichnis `public` in die Zip-Datei `public.zip`. Ich habe dafür die Zip-Anwendung `7zip` installiert.

Zusätzlich beende ich ggf. einen bereits wartenden Hugo-Prozess, starte die lokale Testseite `http://localhost:1313/` und lasse dann Hugo auf Änderungen warten, damit ich meine neuen oder geänderten Webseiten testen kann. Dann ggf. ändere sie, sie werden dann sofort verändert angezeigt und wieder getestet, bis ich mit dem Ergebnis zufrieden bin.

```
REM @ECHO OFF
TITLE Running hugo now and showing local test site
REM Please note that you need to add C:\Programme\Hugo\bin
REM (your Hugo installation folder) to your %PATH%.
:loop
RMDIR /S /Q public
timeout /t 1 /nobreak > nul
if exist public goto loop
taskkill /f /im hugo.exe
hugo -D || PAUSE && EXIT /B
CD public && "C:\Programme\7-Zip\7z.exe" a -tzip "public.zip" "."
CD ..
start "" http://localhost:1313/
REM hugo serve
hugo server --disableFastRender
```