

Webseitenerstellung mit Hubzilla

Eine Webseite zu erstellen ist unter Hubzilla sehr einfach. Voraussetzung für die Erstellung und Anzeige von Webseiten ist lediglich die Installation des Addons. Mit dem Aufruf des Addon wird eine Liste der bereits existierenden Webseiten angezeigt und eine Schaltfläche "Create" unterstützt die Erstellung neuer Webseiten. Bei der Erstellung einer neuen Webseite wird der

Inhaltstyp

gefragt. Zur Auswahl stehen

- BBcode
- HTML
- Markdown
- Text
- Comanche Layout

Der einfachste Inhaltstyp ist **Text**. Jeglicher eingegebene Text wird ohne weitere Gestaltungsmöglichkeiten angezeigt. Sehr viel mehr Komfort in der Gestaltung wird bei **BBcode** und **Markdown** geboten. **BBcode** hat im Editor auch Steuerelemente, die sich interaktiv nutzen lassen, um Fettschrift oder kursive Anzeigen zu erzielen. Beide, sowohl **BBcode** und **Markdown**, sind Auszeichnungssprachen, die mächtige

Gestaltungsmöglichkeiten durch Anwendung bestimmter Codes bieten. Beispielsweise ist Text in Kursivschrift bei BBcode mit `[i]kursiv[/i]` und bei Markdown mit `*kursiv*` zu erzielen. Intern wird bei beiden der passende Html Code erzeugt, der in vielen Anwendungsfällen aufwändiger zu formulieren wäre. Einfach ist dabei noch `kursiv`.

Das kann mit dem Inhaltstyp **HTML** vereinbart werden. Zwar bietet Html zweifellos die mächtigeren Möglichkeiten der Gestaltung, ist aber zugleich die kompliziertere und unübersichtlichere Eingabemöglichkeit von Texten und aus diesem Grund oft weniger zu empfehlen. **BBcode** und **Markdown** werden zu Html übersetzt und führen in den meisten Fällen zu guten Darstellungen.

Edit Webpage

Cancel Delete

Page content type

Markdown

Page layout

default

You can create your own with the layouts tool

A 1st webpage

Summary (optional)

a1p


using md

for saying blah blah ``blah`` and blah.

Permissions offered are Public, OnlyMe, Advanced .> customSelection.




A 1st webpage

 **Hilmar R (u-zapalot)**
 ✓ 13 days ago

using md

for saying blah blah
blah
 and blah.

Permissions offered are Public, OnlyMe, Advanced .-> custumSelection.



Ein sehr spezieller Inhaltstyp ist der des **Comanche Layout**. Gelegentlich erweist sich der als mehr nützlich, ist aber zunächst für einfache Webseiten nicht erforderlich.

Ergänzend sei angemerkt, eine Übersicht mit Beispielen zu [BBcode](#) und [Markdown](#) umfaßt jeweils wenige Seiten. Dokumentationen zu [Html](#) umfassen dagegen viele hunderte von Seiten und sind eben mal nicht adhoc zu verinnerlichen. Das Thema Html ist auch noch nicht abgeschlossen, sondern ist im

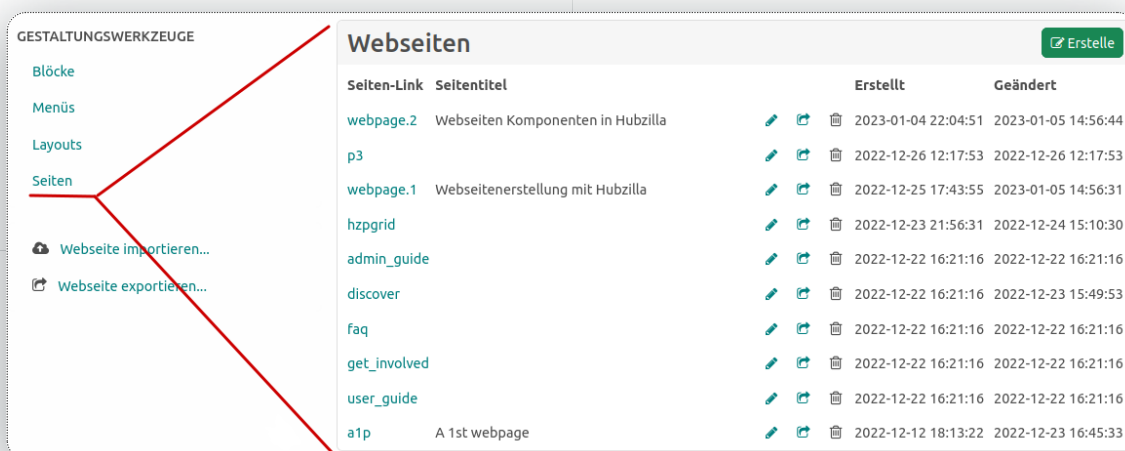
Zusammenhang mit CSS/CSS3 zu betrachten und CSS wiederum ist auch auf vielen hundert Seiten dokumentiert.

Diese Seite hier ist mit Markdown geschrieben und konnte allein deshalb viel flotter erstellt werden, als es mit Html auch nur ansatzweise möglich gewesen wäre.

Page content type:	Markdown
Page layout:	Default
Title:	Webseitenerstellung
Summary:	
Name:	webpage.1

Webseiten Komponenten in Hubzilla

Mit Aufruf der Webseiten App wird das Steuerungspanel für Webseiten eröffnet und angezeigt:



Im linken Seitenpanel sind Navigationseinträge zu den Komponenten der Webseiten angeboten:

- **Blocks (Blöcke)**

Obwohl Blöcke Teile der Webseiten enthalten, sind sie nicht in den Pages (Seiten) unmittelbar angesiedelt, sondern können (nur) in Layouts (Gestaltungen) angesprochen werden.
| *Blöcke können nur in Layouts verwendet werden.*

- **Menüs**

Hier lassen sich Navigationsmenüs definieren, die ähnlich den Blöcken in Layouts angeordnet werden können.
| *Menüs werden im Layouts platziert.*

- **Layouts (Gestaltungen)**

hier wird die grundlegende Gestaltung der Seite beschrieben: Beispielsweise die Regelung, dass die Navigationsmenüs im linken Seitenbereich platziert werden sollen (so wie im Bild oben bei den Gestaltungswerkzeugen).

| *Ein Layout wird einer oder mehreren Seiten zugeordnet, und zwar beim*

Gestaltungswerkzeug der Seiten.

- **Pages (Seiten)**

Die eigentlichen Webseiten sind hier aufgelistet. Eine einfache Webseite, die keine besonderen Anforderungen an Gestaltungen hat, wie beispielsweise die erste Webseite, benötigt keine Blöcke, keine Menüs und kein spezielles Layout. Aber wenn ein Layout gewünscht ist, ist es hier der Seite zuzuordnen.

| *Die Verknüpfung zwischen einer Seite und einem Layout ist bei der Seite herzustellen.*

Zwei weitere Menüeinträge befinden sich im unteren Teil der Gestaltungswerkzeuge:

- Import website ...
- Export website ...

Mittels einer Zip-Datei können die Komponenten der Webseiten exportiert und (auch woanders) importiert werden.

Navigationsmenü

Diese Seite ist die zweite mit Beispielen. Um die Navigation zwischen den beiden (und weitere) Seiten zu erleichtern, ist ein Navigationsmenü hilfreich. Die Arbeitsschritte sind nun:

1. Ein Menü erstellen
2. Das Menü um Menüeinträge ergänzen
3. Das Menü im Layout zuordnen

Bei den Menüeinträgen handelt es sich um Navigationslinks, die jeweils auf eine bestimmte Seite verweisen. Die bisherigen zwei Seiten sind benannt als *webpage.1* und *webpage.2*. Das Menü soll die Überschrift *Navi Beispiele* als Titel erhalten. Das Menü benötigt im Schritt 1 noch einen eindeutigen Namen, z.B. *webpagemenu*

GESTALTUNGSWERKZEUGE

- Blöcke
- Menüs**
- Layouts
- Seiten

Menüs Erstelle

Menu Name	Menütitel	Erstellt	Geändert
bookmarks	Hilmar R (u-zapalot)s Lesezeichen	2021-09-06 20:57:14	2023-01-05 10:34:26
nav	Navi	2022-12-23 13:34:52	2022-12-23 13:37:28
webpagemenu	Navi Beispiele	2023-01-04 22:08:09	2023-01-05 14:01:30

Schritt 1 beschreibt das Menü::

GESTALTUNGSWERKZEUGE

- Blöcke
- Menüs**
- Layouts
- Seiten

Menü bearbeiten Bearbeite Menü Inhalte

Name des Menüs *

Muss eindeutig sein, ist aber nur für Dich sichtbar

Menu title

Menü Titel wie er von anderen gesehen wird

Allow bookmarks im Menü können gespeicherte Lesezeichen abgelegt werden Ja Nein

Absenden und Fortfahren >

Schritt 2 umfaßt die Menüeinträge:

GESTALTUNGSWERKZEUGE

- Blöcke
- Menüs**
- Layouts
- Seiten

Menü: Navi Beispiele Menüelement hinzufügen

Name des Links	Ziel des Links
Beispielseite 1: Erstellung	page/hi-u/webpage.1
Beispielseite 2: Komponenten	page/hi-u/webpage.2

NAVI BEISPIELE

- Beispielseite 1: Erstellung
- Beispielseite 2: Komponenten

Die Details des zweiten Menüeintrages im Menü:

GESTALTUNGSWERKZEUGE

- Blöcke
- Menüs**
- Layouts
- Seiten

Bearbeite Menü-Bestandteil

Link Text *

Ziel des Links oder Untermenüs *

Enter URL of the link or select a menu name to create a submenu

Reihenfolge in der Liste

Größere Nummern werden weiter unten in der Auflistung einsortiert

Magic-Auth verwenden, falls verfügbar Ja Nein

Öffne Link in neuem Fenster Ja Nein

Absenden

Der **Link-Text** ist die verbale Beschreibung der Menüzeile, so wie sie schon links sichtbar ist. Das Ziel der Navigation, die zweite Webseite, ist so wie im Beispiel

dargestellt anzugeben: `page/kanal/name.der.webseite` Eine nicht unwichtige Eigenschaft *Magic Auth verwenden falls verfügbar* ist idealerweise meistens eingeschaltet. Diese Eigenschaft bewirkt, dass ergänzend zum Navigationsziel in Form einer URL die Identität des Seitenbesuchers ergänzt wird. Dadurch ist es möglich, für die Seite zu regeln, ob der Zugriff gestattet ist oder

nicht. Die Einstellung ist nicht die, wer Zugriff auf die Seite hat (die wird in der Eigenschaft der Seite festgelegt), sondern die Ergänzung der Identität des Seitenbesuchers in der URL. Ist die Eigenschaft nicht eingeschaltet, kann keine Identitätsprüfung stattfinden und der Seitenbesucher wird als Anonym behandelt, was zur Konsequenz hat, dass der Zugriff dann oft verweigert wird.

Menüzuordnung im Layout

Das Layout Gestaltungswerkzeug zeigt eine Übersicht der vorhandenen Layouts, die bearbeitet oder neu hinzugefügt werden können:

Layout-Name	Layout-Beschreibung	Erstellt	Geändert
webpage.lo		2023-01-05 13:56:01	2023-01-05 14:09:49
behubz2		2022-12-29 19:32:29	2023-01-05 13:51:14
l3		2022-12-26 10:59:32	2022-12-28 20:43:53
l2		2022-12-24 13:16:05	2022-12-24 14:10:04
hzipgridm		2022-12-23 20:49:22	2022-12-23 20:49:22
hubzilla_layout		2022-12-22 15:21:14	2022-12-22 15:21:14
l1		2022-12-17 16:48:50	2023-01-05 11:34:46

In dem Beispiel soll das Layout als **webpage.lo** benannt sein.

Schritt 3 Layout Beschreibung

Ich habe mir mal erlaubt, in der Abbildung rumzupinseln: Die Schaltfläche sollte nicht mit *Bearbeiten*, sondern mit *Speichern* beschriftet sein. Die Schaltfläche *Löschen* ist an der Stelle unsinnig, weil sie das Layout (vielleicht versehentlich) löschen läßt. Aber nun zum Inhalt des Layouts, das mit der *Comanche Page Description Language (PDL)* formuliert und festgelegt wird.

Die Struktur jeder Anzeige mit Hubzilla ist durch ein mehr oder weniger komplexes Gerüst an HTML Anweisungen geregelt. In diesem Gerüst sind oft sogenannte Regionen (Regions) zu finden, die, wie der Name schon andeutet, bestimmte Bereiche (oder Flächen) der Anzeige markieren. Fast immer ist der Hauptteil, simpel als *Content* (Inhalt) bezeichnet, anzutreffen und der wird meistens im mittleren Bereich des Bildschirms

dargestellt. Manchmal nur links, manchmal auf beiden Seiten sichtbar ist die Region, die mit *aside* benannt ist. In der Layout Beschreibung sind die beiden Regionen mit `[region=...]...[/region]` abgegrenzt. Das weiter oben mühsam erarbeitete Menü ordnen wir im *aside* Bereich ein und die Seite wird über eine Variable namens *\$content* repräsentiert und im mittigen Hauptteil platziert.

Das sieht nicht so kompliziert aus und ist es vermutlich für viele auch nicht, wenn erst eine gewisse Gewöhnung an diese Logik akzeptiert wird, Eigentlich könne das Beispiel hier beendet sein, aber da ist ja noch die merkwürdige *template* Anweisung, die besagt, dass nicht das Standard Template (namens *default*) verwendet werden soll, sondern eines mit dem Namen *chocklet*, und das soll darüber hinaus mit der Stylistik *bannertwo* wirken. Hier wird es noch einmal richtig toll kompliziert.

Zunächst der Hinweis, ein Template ist allgemein eine Vorlage und hier eine Design-Vorlage. Templates in Hubzilla bestehen aus HTML, CSS und Comanche Anweisungen. Der Begriff *Comanche* ist weiter oben schon einmal erwähnt worden (das Layout hier wird ja mit Comanche Anweisungen zur Seitenbeschreibung (PDL) beschrieben). Das komplizierte (und immens mächtige) an Comanche ist nun, das sich Comanche in zwei Teile gliedert, und zwar

1. der PDL (das ist das, was wir hier tun)

und

2. dem exponentiell mächtigeren Formalismus der Template Anweisungen. Die PDL Anweisungen (gem.1) sind formell mit `[...]` verwendet, wogegen die Comanche Template Anweisungen (gem.2) formell mit `{{...}}` festgelegt sind.

In den Layout Beschreibungen der Webseiten läßt sich nur der PDL Teil von Comanche anwenden. Die Comanche Template Anweisung - mit einem simplen `{{$content}}` im Hubzilla Code sicherlich irgendwo zu finden - tritt hier nur noch statisch und indirekt durch die Markierung *\$content* in der PDL in Erscheinung. Lange Rede, kurzer Sinn: wenn es gelänge, Comanche PDL mit Comanche Template Language (ich bringe hier mal das Kürzel CTL ins Spiel) für die Webpages auch gefahrlos für den Benutzer zugänglich zu haben, die Hubzilla Webpages App wäre vermutlich eines der mächtigsten CMS Systeme der Welt.

Die Template Anweisung von weiter oben ist noch nicht ausreichend erklärt. Tatsächlich wird dadurch eine CSS Datei `...chocklet_bannertwo.css...` im HTML Strom der Anzeige festgelegt, die bewirkt, das ein mehr geschmeidiges zweiteiliges Spaltengerüst erzeugt, wo der Menübereich in *aside* und der mittige Hauptteil *content* mehr Raum haben als beim ähnlich konstruierten *Default* Template, das bei den Webseiten auch eine zweiseitige Aufteilung bietet.

Zusammenhänge zusammen gebracht:

Der Inhalt einer Webseite wird im Formalismus der Wahl (BBcode, Markdown, Html) in einer Webseite editiert und gespeichert. Zur Webseite gehört ein Layout; da kann der Standardwert *default* zugeordnet werden oder ein eigenes Layout, das mit dem PDL Formalismus zu beschreiben ist und die Aufteilung der Regionen einer Webseite berücksichtigt (Beispiel Menü in *aside* und Content als *\$content* in *content*). Menüs

sind Navigations-Links zu den Webseiten und unterstützen eine leichtere Navigation zwischen den Webseiten.

Im Layout kann ein Template deklariert werden, wie das im Beispiel genannte *chocklet*, das unterschiedliche Stylistiken unterstützt. Im Beispiel wurde *bannertwo* zugeordnet.


```
[template=bannertwo]choklet[/template]
[region=aside]
[menu]webpagemenu[/menu]
[/region]
[region=content]
$content
[/region]
```

Hinter dem Template *Chocklet* steckt erstens eine Datei <view/php/chocklet.php> mit dem Skelett einer vollwertige HTML Seite, inklusive Header, Script sowie Style Zuordnungen und zweitens eine CSS Datei unter view/css/chocklet_bannertwo.css und genau die ist von mir um ein wenig Stylistik ergänzt worden:

```
.page-body {
    columns:2;
}
p {
    text-align: justify;
}
h2, h3, h5 {
    column-span: all;
}
img {
    column-span: all;
    display: block;
    margin: 0.6rem;
    width: 86%;
    border: 1px #444 dotted;
    border-radius: 0.6rem;
    box-shadow: rgba(100, 106, 112, 0.6)
                0px 7px 28px 2px;
}
```

Ein Experiment

In der Untersuchung der Anatomie der Comanche template Anweisung noch ein interessantes Testergebnis: Die Datei

<view/php/choklet.php>

die das Html Skelett darstellt, ist kopiert worden als

<view/php/hubgate.php>

und die css Dateien

<view/css/choklet.css>

view/css/choklet_bannertwo.css

Die Webseite (Klasse *page-body*) wird in zwei Spalten eingeteilt. Paragraphen (*p*) werden im Blocksatz angezeigt. Die Überschriften-ebenen h2, h3 und h5 (in Markdown mit ##, ###, ##### angewendet) führen zu einer Spaltenüberspannung der zwei Spalten; was bewirkt, dass das bisherige Zweispalten-raster ausbalanciert abgeschlossen und die Spalten-überschrift die Spalten überspannt. Danach gehts automatisch wieder mit dem Zweispalten-raster weiter. Diese Eigenschaft ist auch für Bilder definiert, die in etwas verminderter Größe (86%) angezeigt werden, einen abgerundeten Rahmen und eine dezente Schattierung erhalten. Also, wenige Css3 Anweisungen können optisch schon viel bieten.

wurden entsprechend kopiert nach

<view/css/hubgate.css>

view/css/hubgate_banner2.css

Im Gestaltungswerkzeug *Layout* der Webseiten ist sodann die Template Anweisung innerhalb *layout.lo* geändert worden auf

```
[template=bannerc2]hubgate[/template]
```

Das hat funktioniert und sagt uns, zu einem beliebig benennbarem Html

Grundgerüst im Ordner `view/php/` lassen
sich beliebige Stilkombinationen anlegen
in `view/css/` die namentlich auf das Html
Grundgerüst in `view/php/` verweisen und

in geeigneter Kombination mit der
Template Anweisung im
Gestaltungswerkzeug *Layout* der
Webseiten zugeordnet werden können.