

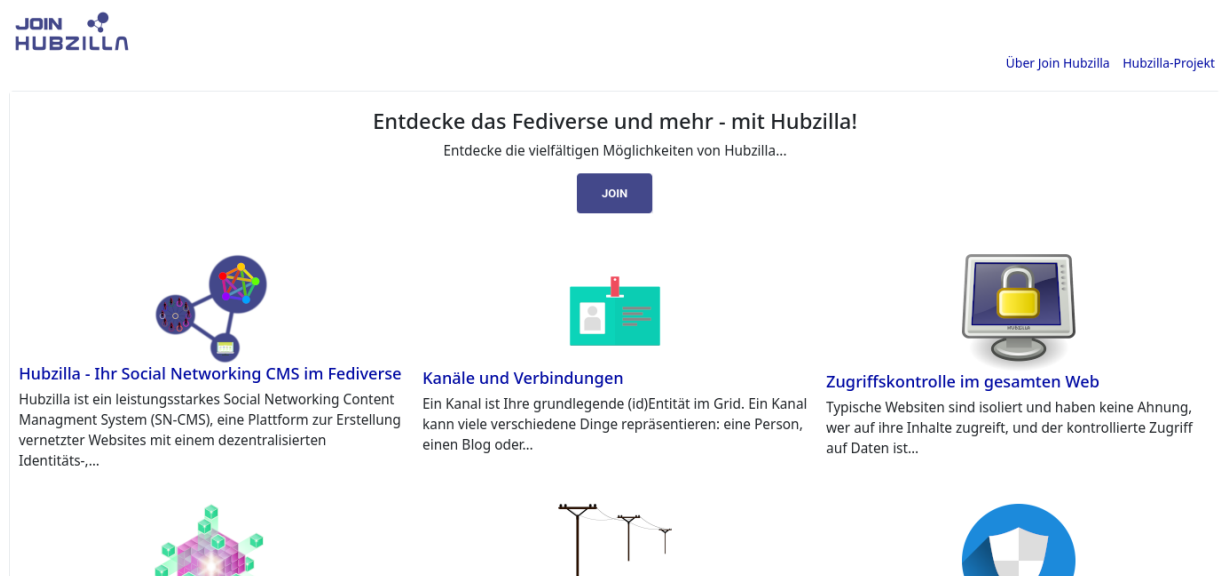
# Webseiten mit Hubzilla gestalten

Hubzilla bietet neben der Netzwerk- (Grid-) und Fediverse-Funktionalität, also der Nutzung als Zugangs- und Verwaltungssoftware für ein Soziales Netzwerk, auch die Werkzeuge eines Content Management Systems. Es kann also Web-Inhalte verwalten und darbieten.

Eine zentrale Funktion dabei ist die App "Webseiten", mit welcher man professionelle Webseiten erstellen und anbieten kann. Beispiele für Hubzilla-Webseiten sind z.B. die Vereins-Webseite des Vereins "Emmaus Köln", die von [Chris](#) erstellt wurde: [Webseite Emmaus Köln](#)



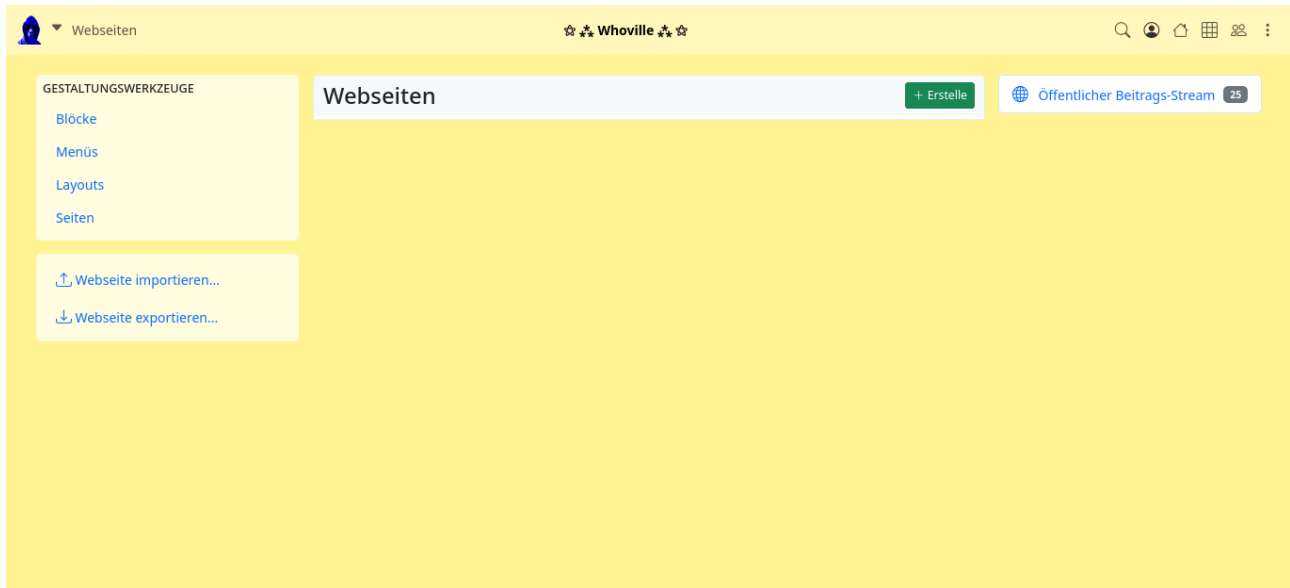
oder die Webseite [Join Hubzilla](#) von [PepeCyB](#).



# Webseiten-App

Es gibt verschiedene Möglichkeiten, Webseiten mit Hubzilla anzubieten. Die Grundlegende Methode ist die, dafür die Hubzilla-App „Webseiten“ zu nutzen. Diese muss zunächst installiert und über das Menü zugänglich gemacht werden.

Ruft man die App schließlich auf, werden in der Seitenleiste die verschiedenen Gestaltungswerkzeuge angezeigt:



- Blöcke
- Menüs
- Layouts
- Seiten

## Gestaltungswerkzeuge

### Blöcke

Blöcke sind wiederverwendbare Inhaltscontainer. Sie können mittels des Layouts an verschiedenen Stellen der eigentlichen Webseite erscheinen und sind damit ein hervorragendes Werkzeug, um bestimmte Inhalte variabel einzusetzen.

### Menüs

Die Menüs der Webseiten-App sind einfache Link-Listen, die sich mit einem Editor erstellen lassen und die, ähnlich wie Blöcke, variabel einsetzbar sind.

## Layouts

Layouts sind die Beschreibung des Aufbaus einer Hubzilla-Webseite. Sie sind Dreh- und Angelpunkt dafür, wie die Webseite letztlich dargestellt wird und welche Elemente wo erscheinen.

## Webseiten

Webseiten sind einerseits der Rahmen für Inhalte, aber auch quasi der "Klebstoff", der die einzelnen Elemente einer Webseite zusammenhält. Neben dem eigentlichen Inhalt einer Webseite, wird hier auch das zu verwendende Layout festgelegt und mit diesem können Blöcke und Menüs zur Webseite hinzugefügt werden.

## Inhaltstypen

Erstellt man einen Block oder eine Webseite, kann man zwischen verschiedenen Inhaltstypen wählen, also mit welcher Sprache der Inhalt gestaltet oder erzeugt wird. Es stehen dafür

- BBcode
- HTML
- Markdown
- Text (Plaintext)
- PHP
- Comanche-Layout

zur Auswahl.

Man kann Blöcke und Webseiten sehr gut mit BBcode gestalten, sofern die Gestaltungsfähigkeiten von BBcode dafür ausreichen oder wenn man hubzilla-spezifischen BBcode (z.B. für den Zugriff auf den Observer oder andere spezielle Informationen) benötigt.

HTML ist natürlich das absolute Optimum für die Webseitengestaltung und ermöglicht alles, was man mit HTML so machen kann. Es sollte, sofern nichts Spezielles benötigt wird, die erste Wahl für Blöcke und Webseiten sein.

Reiner Text (Plaintext) ist natürlich auch möglich, aufgrund der eingeschränkten Gestaltungsmöglichkeiten aber nur dann wirklich angesagt, wenn es auch wirklich um reine Textinhalte geht.

PHP ermöglicht die Webseiten- und Blockgestaltung mit der serverseitigen Skriptsprache PHP. Wer mit PHP umgehen kann, hat hier noch wesentlich weitreichendere Möglichkeiten als mit reinem HTML.

Hinweis: Erstellt man Blöcke oder Webseiten des Inhaltstyps "PHP", dann müssen im Quelltext der öffnende `<?php` und der schließende `?>` PHP-Tag weggelassen werden!

Commanche...

Bei der Webseitenerstellung sollte man isch zuerst mit dem

## Layout

beschäftigen, denn dieses bestimmt, wie die Webseite letztendlich aussieht.

Layouts werden mit der Comanche Seitenbeschreibungssprache, einer BBcode-ähnliche Auszeichnungssprache definiert. Mit Comanche legt man fest, welche Inhalte in den verschiedenen Bereichen der Seite erscheinen sollen. Die verschiedenen Bereiche haben Namen, und diese Namen können sich je nach gewählter Layoutvorlage ändern.

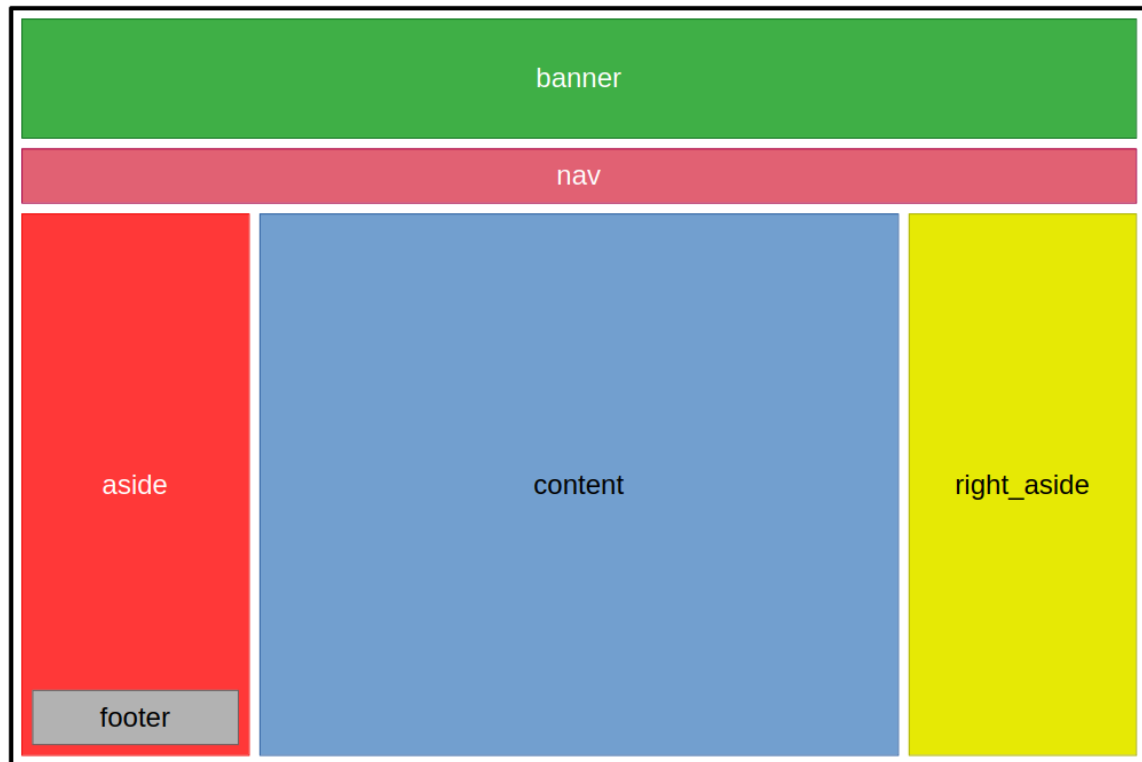
Aktuell gibt es fünf verschiedene vordefinierte Vorlagen: default, full, choklet, redable und zen, wobei choklet in sechs verschiedenen "Flavours" daher kommt (default, bannertwo, three, edgestwo, edgesthree und full).

Um eine Layout-Vorlage auszuwählen, verwendet man die Tags

`[template] [/template]`. Im Fall von choklet wird der "Geschmack" (Flavour) über das öffnende Tag festgelegt `[template=<flavour>]choklet[/template]`. L#sst man für choklet die Auswahl des Flavour weg, wird "default" verwendet.

## Bereiche

Es gibt verschiedene Bereiche (Regionen) einer Webseite:



In der Region "content" sollte in der Regel die Variable `$content` eingetragen werden. Damit erscheint der Inhalt einer Webseite, die man mit dem Gestaltungswerkzeug "Webseiten" erstellt hat, im Inhaltsbereich.

```
[region=content]
$content
[/region]
```

Die Region `[htmlhead] [/htmlhead]` kann dazu verwendet werden, andere css- und js-Definitionen zu verwenden. Derzeit werden als Standard jquery (js), bootstrap (css/js) und foundation (css/js) verwendet.

Diese Region wird nicht dargestellt und dient den Zwecken, welche der gleichnamige Bereich in HTML-Dateien ebenfalls erfüllt.

```
<head>
  ...
</head>
```

## Vorlagen

Es gibt insgesamt fünf fertige Vorlagen, wobei die Vorlage "choklet" in sechs verschiedenen Stilen definiert ist:

1. **default** verwendet folgende Bereiche: nav, aside (mit fester Breite), content und footer
2. **full** verwendet folgende Bereiche: nav, content, footer
3. **redable** zum Lesen längerer Texte hat keinen nav Bereich und verwendet folgende Bereiche: aside, content und right\_aside
4. **zen** ist eine völlig leere Vorlage und verwendet lediglich den content Bereich.
5. **choklet** ist eine floating Layoutvorlage in verschiedenen Flavours:
  - 5.1. **default** verwendet folgende Bereiche: nav, aside, content und footer
  - 5.2. **bannertwo** verwendet folgende Bereiche: banner, nav, aside, content und footer
  - 5.3. **three** verwendet folgende Bereiche: nav, aside, content, right\_aside und footer
  - 5.4. **edgestwo** verwendet folgende Bereiche: (fixed size) nav, aside und content
  - 5.5. **edgesthree** verwendet folgende Bereiche: (fixed size) nav, aside, content und right\_aside
  - 5.6. **full** verwendet folgende Bereiche: (fixed size) nav, (darunter!) banner, aside, content und right\_aside

Je nachdem was man nun für eine Webseite gestalten möchte, muss man das passende Layout auswählen. Verwendet man ein Layout, welches die Region `nav` beinhaltet und legt keinen geänderten Inhalt für den `nav`-Bereich fest, wird dort die Navigationsleiste des Hubzilla-Kanals angezeigt. Die Webseite bleibt also als Hubzilla-Webseite erkennbar und erlaubt über diesen Bereich, der ja auch das App-Menü beinhaltet, Zugriff auf andere Anwendungen und Ansichten des Kanals, soweit dies erlaubt ist.

Für eine eigene Navigationsleiste, muss man ein solche als Block erstellen und im Layout dann in die Region `nav` einfügen.

Völlig frei in der Gestaltung des Layouts ist man mit der Vorlage "zen", die ausschließlich den Bereich `content` anzeigt. Damit muss man dann das Layout aber auch selbst mit den Gestaltungsmöglichkeiten, die einem HTML selbst und CSS speziell bieten, selbst erstellen.

## Blöcke

Blöcke sind die optimalen Objekte für Inhalte, die man ggf. wiederkehrend anbieten möchte. Erstellt man einen solchen Block und ändert den Inhalt, dann wird die Änderung auch an jeder Stelle sichtbar, wo er eingesetzt wurde.

Angenommen, wir möchten auf verschiedenen Seiten in der Seitenleiste den Besucher begrüßen, sofern er bei einem Hub angemeldet ist, dann könnte man für diesen Zweck einen Block "begruessung" unter Verwendung von BBcode (um den Namen zu ermitteln) erstellen.

```
Hallo, herzlich willkommen [observer.name]!
```

`[observer.name]` zeigt den Kanalnamen des Besuchers an.

Wenn wir nun diese Begrüßung auf jeder Webseite in der Seitenleiste anzeigen lassen möchten, fügen wir den Block zu dem Layout, welches für die Webseiten genutzt wird, in der Region "aside" hinzu:

```
[/region]
[block]begruessung[/block]
[region=aside]
```

Für ein eigenes Menü im Stil der Hubzilla-Navigationsleiste kann man z.B. einen Block mit einer Bootstrap5-Navbar erstellen und dann in die Region "nav" im Layout einfügen.

Blöcke kann man übrigens auch bei den Hubzilla-Ansichten verwenden. Sie lassen sich mit dem PDL-Editor als ITEM in das jeweilige Ansichts-Layout einfügen. Ebenso lassen sich auch Widgets des Hubzilla-Systems in das Layout für eigene Webseiten einbauen, z.B. ließe sich mit

```
[aside]
[widget=profile] [/widget]
[/aside]
```

im Layout für die eigenen Webseiten, die Profilkarte des eigenen Kanals anzeigen lassen.

## Webseiten

Webseiten dienen der Darstellung der eigentlichen Inhalte. Sie werden unter Nutzung des ausgewählten Layouts angezeigt.

An der Stelle im Layout, wo die Variable \$content auftaucht, wird der Inhalt des Webseiten-Objekts gerendert.

## Inhalte erstellen

Hubzilla verfügt aktuell noch nicht über WYSIWYG-Editoren für HTML-Code und BBcode in der App "Webseiten". Das mag ein wenig lästig sein, weil man externe Editoren nutzen muss. Der Hubzilla-Workshop bietet aber zwei einfache Online-Editoren an, die man für das Erstellen der Inhalte nutzen und dann den Quellcode, der sich bei diesen Editoren anzeigen lässt, in den jeweiligen Editor der Webseiten-App kopieren kann:

[Webeditor](#) (für HTML-Inhalte)

[SCEd](#) (für BBcode-Inhalte)

Generell sollte jedem, der eine Webseite mit Hubzilla erstellen möchte, klar sein, dass er gewisse grundlegende Kenntnisse in Hubzilla-BBcode, HTML und CSS haben sollte.

Darüber hinaus ist es von Vorteil, wenn man sich auch mit Bootstrap5 auskennt.

Ressourcen um sich mit diesen Themen vertraut zu machen sind z.B.:

- [bbcode.org](http://bbcode.org)
- [SelfHTML](#)
- [W3S Bootstrap5 Tutorial](#)

# Andere Wege für Webseiten

Eine weitere Möglichkeit, Webseiten mit Hubzilla anzubieten, ist das Hosten von statischen Webseiten in der Cloud.

Es gibt eine Vielzahl von Static Website Generatoren, also Systemen und Programmen, mit denen man statische Webseiten, z.B. komplette Blogs erstellen kann.

Beispiele dafür sind [Hugo](#), [Jekyll](#), [Publii](#), [mdBook](#)...

Mit diesen Systemen kann man mit einfachen Mitteln eine schön gestaltete Webseite oder ein ganzes Webseiten-System erstellen lassen, welches man dann in einen Ordner der Hubzilla-Cloud hochlädt und mit der URL zu einer HTML-Datei (in der Regel index.html) in diesem Verzeichnis aufrufen kann. Hubzilla in Verbindung mit der Cloud fungiert dann praktisch als Webserver für die Webseiten-Dateien.

Publii ist dabei das System, welches am nächsten an einem üblichen Webseiten-CS, wie z.B. WordPress dran ist.

Wenn man ein solches System nutzt, bleibt natürlich die Webseiten-Funktionalität von Hubzilla außen vor, trotzdem fungiert Hubzilla mit seiner Cloud als eine Art CMS.

Hier ein paar Demos:

[Webseite mit Hugo](#)

[Webseite mit mdBook](#)

[Webseite mit Publii](#)

