

Wir erstellen eine Demo-Webseite

Um die Informationen aus dem Skriptum „Webseiten mit Hubzilla“ einmal in der Praxis umzusetzen, werden wir nun eine Webseite erstellen.

Die Seite soll zeigen, wie man eine Webseite erstellt, die nicht auf den ersten (und auch nicht einmal auf den zweiten) Blick erkennen lässt, dass sie durch Hubzilla angeboten wird. Soll bedeuten: Sie erscheint wie eine eigenständige Webseite, zeigt nicht die Hubzilla-Navigationsleiste und auch keine hubzilla-spezifischen Widgets. Damit am Ende trotzdem klar wird, dass die Webseite mit Hubzilla verwirklicht ist, soll aber ein expliziter Hinweis darauf erscheinen.

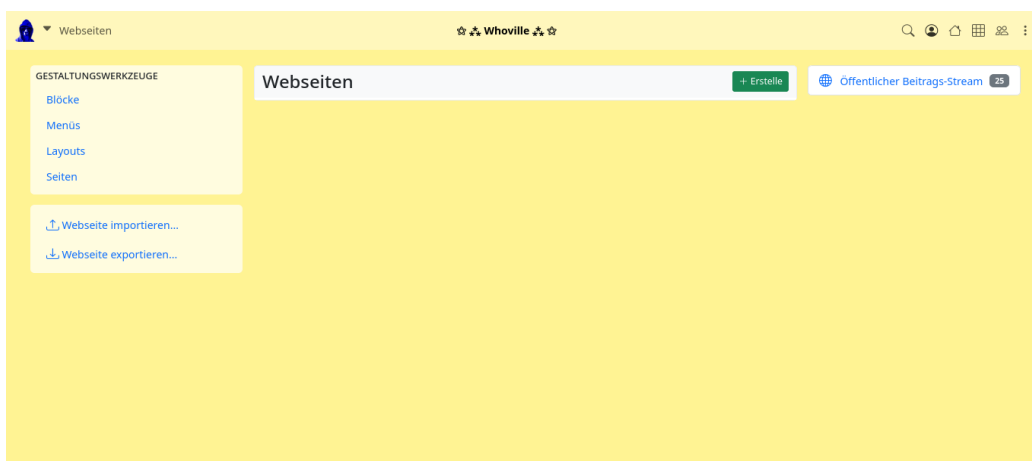
Weil wir ja „der Hubzilla-Workshop sind“, entwerfen wir eine einfache Seite in Art einer Landingpage, die ein Titelbild zeigt, Informationen, was der Hubzilla-Workshop ist. Außerdem sollen beispielhaft Verweise zu den bereits durchgeführten ersten drei Workshops angezeigt werden, und das Workshop-Team kurz vorgestellt. Die Seite soll eine Navigationsleiste bieten, die lediglich einen Link zum Workshop-Kanal und ein Hamburgermenü-Symbol anzeigt. Der Klick auf das Menü-Symbol soll dann ein Off-Canvas Menü einblenden, welches zu anderen Bereichen des Workshop-Kanals führt.

Im Footer schließlich soll der Hinweis auf Hubzilla als Unterbau und ein Copyright-Vermerk zu sehen sein.

Die einzelnen Bereiche werden wir mit ein paar „Schmankerln“ würzen, die uns z.B. Bootstrap5 (von Hause aus unter Hubzilla verfügbar) bietet.

Die Webseite bauen wir zusammen in Pepes Workshop-Demo-Kanals „Blue Nomad“. Wer den Webseitenbau im eigenen Kanal nachvollziehen möchte, muss die URLs z.B. zu Grafiken in den HTML-Quelltexten entsprechend für die eigene Kanaladresse anpassen.

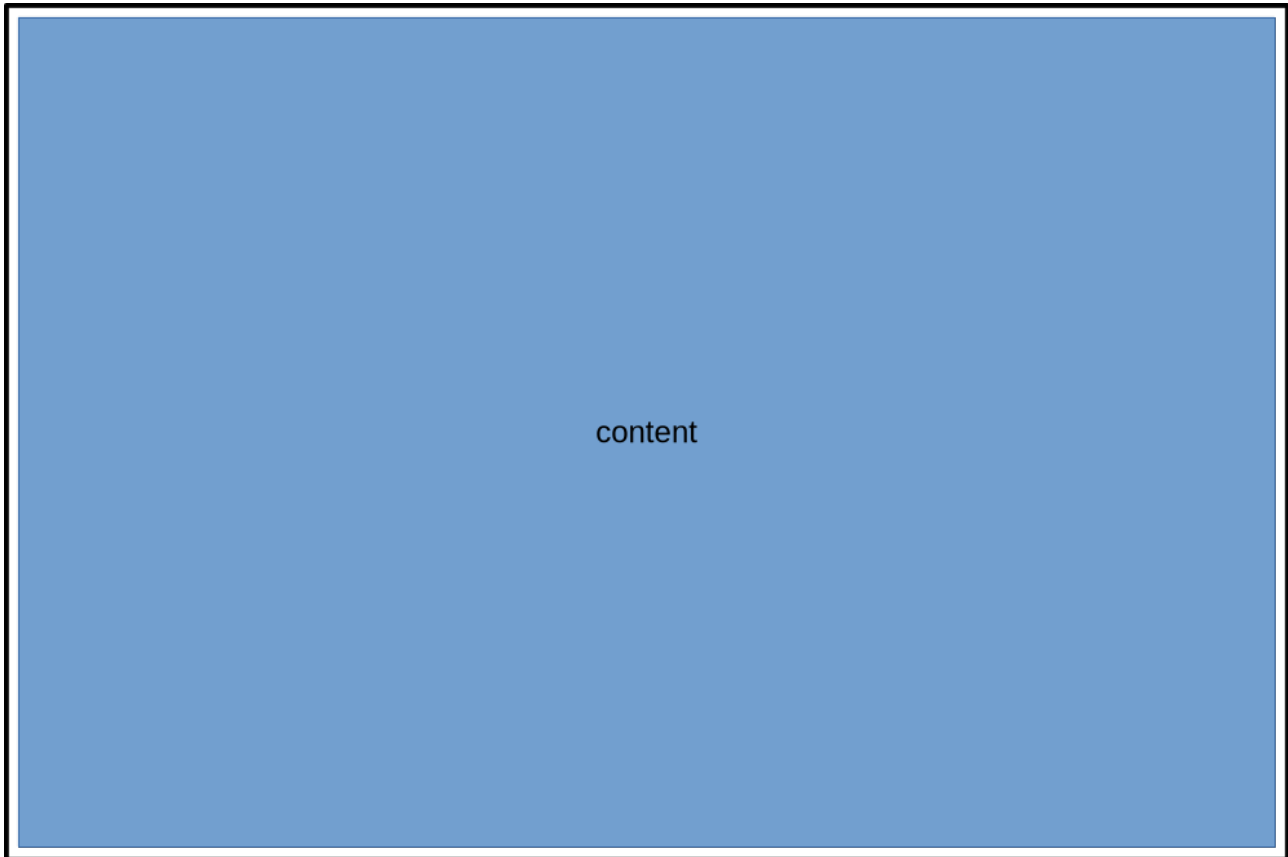
Als erstes rufen wir im Kanal aus dem App-Menü die App „Webseiten“ auf und sehen (sofern wir noch nie etwas mit Webseiten gemacht haben) die leere Liste der existierenden Webseiten des Kanals.



Ein Layout – Auswahl der passenden Vorlage

Da wir völlig frei gestalten möchten, benötigen wir quasi ein „leeres Blatt Papier“ für unsere Webseite. Es soll also lediglich den eigentlichen Inhaltsbereich geben, keine Hubzilla-Navigationsleiste keine Seitenleisten und keinen expliziten Footer.

Dafür bietet sich die Vorlage „Zen“ an, die genau das bietet.



Wir wählen aus den Gestaltungswerkzeugen „Layouts“ aus und erstellen ein neues Layout mit dem Namen "blanko".

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

Jetzt haben wir unsere Leinwand und wir können auch schon mit der Arbeit an der Webseite beginnen, wenn wir sie monolithisch als eine einzige Datei erstellen möchten (wie eine solche Webseite aus einzelnen Blöcken zusammengebaut wird, sehen wir im später).

Dateien zur Verfügung stellen

Wir benötigen für unsere Webseite einige Grafiken, die wir durch unseren Kanal zur Verfügung stellen möchten. Diese legen wir in der Cloud des Kanals in einem eigenen Ordner "website" ab, den wir in der Cloud erstellen.

Der erste Teil unserer Webseite

Nun fangen wir an, unsere Webseite zu bauen. Sie soll zunächst einen Titel und darunter ein Titelbild anzeigen. Darunter soll in Fließtext (stellenweise mit einer unsortierten Liste) der Hubzilla-Workshop vorgestellt werden.

Unter dem Text soll dann auch noch, als „erstes Schmankerl“ das Video „Einstieg in Hubzilla“ aus dem Workshop-Kanal inklusive Vorschaubild eingebettet werden.

Unter diesem Video sollen dann Informationen zu den ersten drei durchgeführten Workshops angezeigt werden. Damit es nicht einfach eine langweilige Aufzählung ist, wollen wir hier ein Accordion für die Darstellung nutzen, welches wir mit Bootstrap5 verwirklichen.

Wir müssen also jetzt erst einmal das Titelbild in das neue Verzeichnis in der Cloud hochladen, damit es uns zur Verfügung steht. Das Bild hat den Dateinamen "title.png" und ist nach dem Hochladen unter der URL

<https://hub.hubzilla.hu/cloud/bluenomad/website/title.png> erreichbar.

Jetzt öffnen wir das Gestaltungswerkzeug „Seiten“ und erstellen eine neue Webseite.

Die erste Webseite, die in der App „Webseiten“ erstellt wird, bekommt als Vorgabe den Namen "home", der bereits im entsprechenden Feld eingetragen ist. Diesen Namen können wir natürlich nach unserem Geschmack ändern, belassen es hier aber bei "home". Über diesen Namen wird die Webseite, sobald sie erstellt wurde, erreichbar sein. Sie hat die URL <https://hub.hubzilla.hu/page/bluenomad/home>.

Als Inhaltstyp wählen wir HTML und für das zu verwendende Layout wählen wir das gerade erstellte mit dem Namen „blanko“ aus.

In das große Eingabefeld kommt nun der HTML-Quelltext für die Webseite:

```
<!-- Teil 01 -->
<div class="row">
  <div class="col"></div>
  <div class="col">
    <h1>Willkommen beim Hubzilla-Workshop </h1>
    <p></p>
    <p>Für alle, die Hubzilla kennenlernen, souverän nutzen und bisher unbekannte Funktionen entdecken möchten.</p>
    <p>Egal ob Einsteiger, Fortgeschrittene oder Entwickler - gemeinsam wollen wir dazu beitragen, dass Hubzilla einem breiten Publikum bekannt wird, leichter anzuwenden ist und noch mehr Freude bei der Nutzung bringt.</p>
    <p>Hubzilla ist besonders im Fediverse: Es verbindet verschiedene dezentrale Plattformen miteinander, ermöglicht sichere Direktnachrichten, verschlüsselte Kommunikation und die Verwaltung eigener Kanäle oder Instanzen.</p>
    <p>Zu den besonderen Funktionen gehören unter anderem:</p>
    <p><ul>
      <li>Verwalten mehrerer Kanäle unter einem Account</li>
      <li>Zugriffs- und Berechtigungskontrolle für Inhalte</li>
      <li>Kalender- und Dateifreigaben</li>
    </ul>
  </div>
</div>
```

```

    <li>Content-Streams abonnieren und interagieren</li>
    <li>Integration in andere Fediverse-Plattformen (Mastodon, Friendica, PeerTube etc.)</li>
</ul></p>
<p>um nur einiges zu nennen</p>
<p>So können Anwender, Vereine, Firmen oder Entwickler flexibel und unabhängig arbeiten - und trotzdem mit anderen Nutzern im Netzwerk interagieren.</p>
<p><video
    controls
    width="100%"
    height="auto"
    preload="metadata"
    poster="https://hub.hubzilla.hu/cloud/hz-workshop/tn/tn-einstieg.png"
    style="max-width:100%; display:block;"
    >
    <source src="https://hub.hubzilla.hu/cloud/hz-workshop/Videos/ws2-einstieg_720p.mp4"
type="video/mp4">
    Dein Browser unterstützt das Video-Tag nicht.
</video></p>
<br><br>
<!-- Accordion für die bisherigen Workshops -->
<h2>Workshops</h2>
<div id="accordion">

    <div class="card">
        <div class="card-header">
            <a class="btn" data-bs-toggle="collapse" href="#collapseOne">
                <strong>Workshop #01: Mi., 17.09.2025 ★ 19:30</strong>
            </a>
        </div>
        <div id="collapseOne" class="collapse" data-bs-parent="#accordion">
            <div class="card-body">
                <b>Filter...</b>
                <ul>
                    <li>Anhand praktischer Beispiele verschiedene Einstellungen & deren Auswirkungen</li>
                    <li>Filter - Welche gibt es? Wozu dienen sie? Wie kann ich sie nutzen?</li>
                    <li>Tipps für Account-Nutzer wie auch für Instanzbetreiber</li>
                </ul>
                <a href="https://hub.hubzilla.hu/cards/hz-workshop/hzws01" target="_blank"><strong>...:
LINK :...</strong></a>
            </div>
        </div>
    </div>

    <div class="card">
        <div class="card-header">
            <a class="collapsed btn" data-bs-toggle="collapse" href="#collapseTwo">
                <strong>Workshop #02: Mi., 29.10.2025 ★ 20:30</strong>
            </a>
        </div>
        <div id="collapseTwo" class="collapse" data-bs-parent="#accordion">
            <div class="card-body">
                <b>Einstieg in Hubzilla</b>
                <ul>
                    <li>Server-Auswahl</li>
                    <li>Unterschiede Account/Kanal</li>
                    <li>verschiedene Registrierungstypen</li>
                    <li>Account anlegen</li>
                    <li>Kanal anlegen</li>
                    <li>Profil vervollständigen</li>
                    <li>Apps installieren</li>
                    <li>Die fünf wichtigsten Apps</li>
                    <li>Verbindungen eingehen</li>
                    <li>Anzeige-Einstellungen - Themes</li>
                    <li>Einstellungen</li>
                    <li>Wir erstellen eine "Features-App"</li>
                    <li>Privacy Gruppen und Kontaktrollen</li>
                </ul>
                <a href="https://hub.hubzilla.hu/cards/hz-workshop/hzws02" target="_blank"><strong>...:
LINK :...</strong></a>
            </div>
        </div>
    </div>


    <div class="card">
        <div class="card-header">
            <a class="collapsed btn" data-bs-toggle="collapse" href="#collapseThree">
                <strong>Workshop #03: Mi., 26.11.2025 ★ 19:30</strong>

```

```
</a>
</div>
<div id="collapseThree" class="collapse" data-bs-parent="#accordion">
  <div class="card-body">
    <b>Hubzilla - Posten und alles was dranhängt</b>
    <ul>
      <li><strong>Postings</strong>: Editor, Textgestaltung, Anh&auml;nge, eingebettete
Medien</li>
      <li><strong>Bilder</strong>: Bilder einf&uuml;gen und das "AltText-Drama"</li>
      <li><strong>Konversation</strong>: Direktnachrichten, Konversationscontainer,
Erw&auml;hnungen und wozu sie dienen, Berechtigungseinstellungen</li>
      <li><strong>Mehr</strong>: Editieren und L&ouml;schen, Wall-to-Wall Postings, Posten
in einen Foren-Kanal</li>
      <li><strong>Fortgeschritten</strong>: Unterschiede zu anderen Diensten, der Weg eines
Postings, der Weg eines Kommentars, Antwortkontrolle</li>
    </ul>
    <a href="https://hub.hubzilla.hu/cards/hz-workshop/hzws03" target="_blank"><strong>...:
LINK :...</strong></a>
  </div>
</div>
</div>
</div>
<div class="col"></div>
</div>
```

Nach dem Absenden des Inhalts können wir uns unsere Webseite erst einmal anschauen:

Willkommen beim Hubzilla-Workshop



Für alle, die Hubzilla kennenlernen, souverän nutzen und bisher unbekannte Funktionen entdecken möchten.

Egal ob Einsteiger, Fortgeschrittene oder Entwickler – gemeinsam wollen wir dazu beitragen, dass Hubzilla einem breiten Publikum bekannt wird, leichter anzuwenden ist und noch mehr Freude bei der Nutzung bringt.


Hubzilla ist besonders im Fediverse: Es verbindet verschiedene dezentrale Plattformen miteinander, ermöglicht sichere Direktnachrichten, verschlüsselte Kommunikation und die Verwaltung eigener Kanäle oder Instanzen.

Zu den besonderen Funktionen gehören unter anderem:

- Verwalten mehrerer Kanäle unter einem Account
- Zugriffs- und Berechtigungskontrolle für Inhalte
- Kalender- und Dateifreigaben
- Content-Streams abonnieren und interagieren
- Integration in andere Fediverse-Plattformen (Mastodon, Friendica, PeerTube etc.)

um nur einiges zu nennen

So können Anwender, Vereine, Firmen oder Entwickler flexibel und unabhängig arbeiten – und trotzdem mit anderen Nutzern im Netzwerk interagieren.



Workshops

Workshop #01: Mi., 17.09.2025 ★ 19:30
Filter... <ul style="list-style-type: none">• Anhand praktischer Beispiele verschiedene Einstellungen & deren Auswirkungen• Filter - Welche gibt es? Wozu dienen sie? Wie kann ich sie nutzen?• Tipps für Account-Nutzer wie auch für Instanzbetreiber ...: LINK ...
Workshop #02: Mi., 29.10.2025 ★ 20:30
Workshop #03: Mi., 26.11.2025 ★ 19:30

Das sieht doch schon ganz gut aus.

Zweiter Teil unserer Webseite

Ok, dann gehen wir mal an die Navigationsleiste und das einblendbare Off-Canvas-Menü heran.

Wir öffnen die Seite "home" erneut zum Bearbeiten (Klick auf das Bleistift-Symbol) und fügen ganz oben, über dem Code für den ersten Teil nun den Quelltext für Navigationsleiste und das Menü ein:

```
<!-- Teil 02 -->

<!-- Navbar -->

<nav class="navbar navbar-expand-sm bg-dark navbar-dark">
  <div class="container-fluid">
    <!-- Klickbares Offcanvas-Trigger-Element →
    <button class="btn btn-outline-light me-2" type="button" data-bs-toggle="offcanvas" data-bs-target="#offcanvasNavbar" aria-controls="offcanvasNavbar">
      
    </button>

    <!-- Statischer Navbar-Brand -->

    <a class="navbar-brand" href="https://hub.hubzilla.hu/channel/hz-workshop">
      <span class="navbar-text"><b>Hubzilla-Workshop</b></span>
    </a>
  </div>
</nav>

<!-- Offcanvas Sidebar -->

<div class="offcanvas offcanvas-start" tabindex="-1" id="offcanvasNavbar" aria-labelledby="offcanvasNavbarLabel">
  <div class="offcanvas-header">
    <h3 class="offcanvas-title" id="offcanvasNavbarLabel">Navigation</h3>
    <button type="button" class="btn-close text-reset" data-bs-dismiss="offcanvas" aria-label="Schließen"></button>
  </div>
  <div class="offcanvas-body">
    <ul class="navbar-nav justify-content-end flex-grow-1 pe-3">
      <li class="nav-item mb-2">
        <a class="nav-link active" aria-current="page" href="https://hub.hubzilla.hu/wiki/hz-workshop/Verzeichnis/Home">Verzeichnis</a>
      </li>
      <li class="nav-item mb-2">
        <a class="nav-link" href="https://hub.hubzilla.hu/wiki/hz-workshop/Videos/Home">Videos</a>
      </li>
      <li class="nav-item mb-2">
        <a class="nav-link" href="https://hub.hubzilla.hu/cards/hz-workshop">Workshops</a>
      </li>
      <li class="nav-item mb-2">
        <a class="nav-link" href="https://hub.hubzilla.hu/cal/hz-workshop">Kalender</a>
      </li>
    </ul>
  </div>
</div>
```

Damit das Symbol für das Hamburger-Menü angezeigt wird, muss die Grafik auch erst wieder in den Cloud-Ordner "website" hochgeladen werden: list.svg

Speichern... und anschauen!



und

- Navigation
- Verzeichnis
- Videos
- Workshops
- Kalender



Auch das hat gut funktioniert. Unsere Webseite hat nun eine Navigationsleiste und ein einblendbares Seitenmenü.

Dritter Teil unserer Webseite

Abschließend soll nun das Workshop-Team vorgestellt werden. Die fünf Protagonisten sollen als Bootstrap5-Karten angezeigt werden. Als Grafik dafür verwenden wir einfach die Profilkarte, die wir als Screenshots aus ihren Hubzilla-Kanälen erstellen. Darunter soll in Fettschrift der Kanalname angezeigt werden und darunter jeweils das Webbie des Kanals,

das mit einem Link zu jeweiligen Kanalseite belegt ist. Wir müssen nun also die Screenshots anfertigen und wieder in unser Cloud-Verzeichnis für die Webseite hochladen.

Als optisches "Schmankerl" sollen die Karten, sobald man in den Anzeigebereich hineinscrollt seitlich in den Inhaltsbereich gleiten. Dafür brauchen wir ein wenig Code, den wir, ebenso wie CSS-Styles, auch einfach in den Seiten-Quelltext einbauen können.

```
<!-- Teil 03 -->
<!-- Style für die eingleitenden Cards -->
<style>
.slide-card {
  transform: translateX(40px);
  opacity: 0;
  transition: transform 600ms cubic-bezier(.22,.9,.39,1), opacity 600ms ease;
}

.slide-card.in-view {
  transform: translateX(0);
  opacity: 1;
}

.slide-card.delay-0 { transition-delay: 0ms; }
.slide-card.delay-1 { transition-delay: 80ms; }
.slide-card.delay-2 { transition-delay: 160ms; }
.slide-card.delay-3 { transition-delay: 240ms; }
.slide-card.delay-4 { transition-delay: 320ms; }

.hero {
  padding: 6rem 1rem;
  background: #f8f9fa;
  text-align: center;
}

.cards-section { padding: 4rem 1rem; }
</style>

<!-- hereingleitende Cards -->
<main>
<p></p>

<section class="cards-section">
  <div class="container">
    <h2 class="mb-4">Das Team</h2>

    <div id="cards-row" class="row g-4">
      <div class="col-12 col-md-6 col-lg-4">
        <div class="card slide-card delay-0">
          
          <div class="card-body">
            <h5 class="card-title"><b>Doris</b></h5>
            <p class="card-text"><a href="https://www.top-netz.eu/channel/doris" target="_blank">doris@top-netz.eu</a></p>
          </div>
        </div>
      </div>
    </div>
  </div>
</section>
</main>
```

```

</div>
</div>

<div class="col-12 col-md-6 col-lg-4">
  <div class="card slide-card delay-1">
    
    <div class="card-body">
      <h5 class="card-title"><b>Chris</b></h5>
      <p class="card-text"><a href="https://im.allmendenetz.de/channel/chris"
target="_blank">chris@im.allmendenetz.de</a></p>
    </div>
  </div>
</div>

<div class="col-12 col-md-6 col-lg-4">
  <div class="card slide-card delay-2">
    
    <div class="card-body">
      <h5 class="card-title"><b>Jupiter Rowland</b></h5>
      <p class="card-text"><a href="https://hub.netzgemeinde.eu/channel/jupiter_rowland"
target="_blank">jupiter_rowland@hub.netzgemeinde.eu</a></p>
    </div>
  </div>
</div>

<div class="col-12 col-md-6 col-lg-4">
  <div class="card slide-card delay-3">
    
    <div class="card-body">
      <h5 class="card-title"><b>Steffen</b></h5>
      <p class="card-text"><a href="https://www.top-netz.eu/channel/steffen" target="_blank">steffen@www.top-
netz.eu</a></p>
    </div>
  </div>
</div>

<div class="col-12 col-md-6 col-lg-4">
  <div class="card slide-card delay-4">
    
    <div class="card-body">
      <h5 class="card-title"><b>PepeCyB</b></h5>
      <p class="card-text"><a href="https://hub.hubzilla.hu/channel/pepecyb"
target="_blank">pepecyb@hub.hubzilla.hu</a></p>
    </div>
  </div>
</div>

</div>
</div>
</section>
</main>

```

```

<script>
document.addEventListener('DOMContentLoaded', function () {
  const cards = document.querySelectorAll('.slide-card');

  // Observer startet Animation, wenn Teil des Elements sichtbar ist (threshold > 0)
  const enterObserver = new IntersectionObserver((entries) => {
    entries.forEach(entry => {
      if (entry.isIntersecting) {
        entry.target.classList.add('in-view');
      }
    });
  }, { root: null, rootMargin: '0px', threshold: 0.15 });

  // Observer entfernt die Klasse, wenn das Element komplett aus dem Viewport ist
  const exitObserver = new IntersectionObserver((entries) => {
    entries.forEach(entry => {
      if (!entry.isIntersecting && entry.boundingClientRect.top > 0 && entry.intersectionRatio === 0) {
        // Element ist unterhalb des Viewports (nach oben gescrollt) — hier entfernen wir Klasse
        entry.target.classList.remove('in-view');
      } else if (!entry.isIntersecting && entry.boundingClientRect.top < 0 && entry.intersectionRatio === 0) {
        // Element ist oberhalb des Viewports (nach unten gescrollt) — ebenfalls zurücksetzen
        entry.target.classList.remove('in-view');
      }
    });
  }, { root: null, rootMargin: '0px', threshold: 0 });

  cards.forEach(card => {
    enterObserver.observe(card);
    exitObserver.observe(card);
  });
});
</script>

```

Diesen Quelltext fügen wir nun in die zum Bearbeiten wieder geöffnete Seite "home" am Ende ein.

Workshop #02: Mi., 29.10.2025 ★ 20:30

Workshop #03: Mi., 26.11.2025 ★ 19:30

Das Team



Doris
doris@www.top-netz.eu

Neues entdecken und für Probleme Lösungen finden.

Doris
doris@top-netz.eu



Chris
chris@im.allmendenetz.de

Fediverse Aktivist - souverän und mündig digital vernetzt - Seit 2017 im Fediverse - dies ist mein öffentlicher Kanal in dem es hauptsächlich um das Fediverse an sich geht

Chris
chris@im.allmendenetz.de



Jupiter Rowland
jupiter_rowland@hub.netzgemeinde.eu

Duothematischer Kanal. Primäres Thema sind virtuelle Welten/OpenSim, sekundäres Thema ist das Fediverse jenseits von Mastodon. Auf diesem Kanal geht es NICHT ums Real Life!

Jupiter Rowland
jupiter_rowland@hub.netzgemeinde.eu



Auch das hat wieder funktioniert und die Karten gleiten auch schön beim Hereinscrollen seitlich ein.

Der Webseite sieht man jetzt aber wirklich nicht mehr an, dass sie mit Hubzilla realisiert wurde. Das ändern wir nun...

Vierter Teil unserer Webseite

Jetzt folgt noch ein Footer, der auf Hubzilla als Unterbau verweist und eine Copyright-Notiz und einen Lizenzhinweis anzeigt. Die Seite soll als CC-BY-NC-SA 4.0 lizenziert werden. Die Grafik für diese Lizenz laden wir auch noch in den Cloud-Ordner.

```
<!-- Teil 04 -->
<!-- Footer -->
<div class="mt-4 p-5 bg-dark text-white">
  <p><div style="text-align: center;">Powered by<br>
  by<br>
  <a href="https://hubzilla.org" target="_blank">Hubzilla</a> - The Social Network CMS</div></p>
  <p><div style="text-align: center;">&copy; 2026 Hubzilla-Workshop</div></p>
  <p><div style="text-align: center;"><a
href="https://creativecommons.org/licenses/by-nc-sa/4.0/deed.de" target="_blank"></a></div></p>
</div>
```

Diesen Quelltext packen wir nun ganz unten in unseren Webseiten-Quelltext.



Und fertig ist unsere Webseite. Zu besichtigen ist sie unter

<https://hub.hubzilla.hu/page/bluenomad/home>

Webseiten mit Blöcken gestalten

Die Webseite, die wir nun erstellt haben, ist monolithisch aufgebaut. Sie besteht aus einem großen "Klotz" HTML-Code inklusive CSS und etwas Javascript.

Auf diese Weise kann man einzelne Webseiten sehr gut erstellen. Möchte man aber Unterseiten hinzufügen, die sich von Stil und Inhalt her einheitlich präsentieren, dann kann man das besser mit Blöcken verwirklichen. Der veränderliche Inhalt der Seite wird dann in den Seiten verpackt, das Layout oder die Layouts hingegen dienen dann dazu, die immer wiederkehrenden Elemente mit Blöcken fest einzubauen. Das bedeutet, solche wiederkehrenden Elemente packt man in Blöcke, die man mit der Webseiten-App erstellt.

Bei unserem Beispiel würde es sich z.B. anbieten, die Navigationsleiste und das einblendbare Off-Canvas-Menü als Block einzusetzen, damit man, wenn man weitere Seiten im passenden Stil erstellen möchte, immer diese zentralen Navigationselemente zur Verfügung hat. Das gilt auch für den Footer, der z.B. auch auf allen Seiten identisch erscheinen soll.

Was wir dafür benötigen

Um das zu realisieren, erstellen wir uns einfach ein weiteres Layout. Dabei belassen wir es bei der Vorlage "Zen", damit die Webseite nicht "nach Hubzilla-Kanal" aussieht.

Und wir benötigen einen Block, der für die Darstellung der Navigationsleiste und dem Seitenmenü zuständig ist.

Der Navigationsblock

Wir können hier einfach den bisherigen Quelltext recyceln und erstellen einen Block mit dem Namen "navigation" vom Inhaltstyp "HTML", in den wir den Inhalt des oben stehenden Teils zwei (Navigationsleiste und Seiten-Menü) einfügen.

Blöcke Erstelle

Art des Seiteninhalts

Titel (optional)

Summary (optional)

Content

```
<!-- Teil 02 -->

<!-- Navbar -->
<nav class="navbar navbar-expand-sm bg-dark navbar-dark">
  <div class="container-fluid">
    <!-- Klickbares Offcanvas-Trigger-Element -->
    <button class="btn btn-outline-light me-2" type="button" data-bs-toggle="offcanvas" data-bs-target="#offcanvasNavbar" aria-controls="offcanvasNavbar">
        
<p><div style="text-align: center;">Powered by<br>  
by<br>  
<a href="https://hubzilla.org" target="_blank">Hubzilla</a> - The Social Network CMS</div></p>  
<p><div style="text-align: center;">&copy; 2026 Hubzilla-Workshop</div></p>  
<p><div style="text-align: center;"><a href="https://creativecommons.org/licenses/by-nc-sa/4.0/  
deed.de" target="_blank"></a></div></p>  
</div>
```

Absenden

Das Layout für die Block-Variante

Nun müssen wir nur noch ein Layout für diese Variante erstellen. Als Vorlage kommt wieder "Zen" zum Einsatz. Wir setzen dann im Inhaltsbereich [region=content] zunächst den Block "navigation" ein, darunter den Seiteninhalt (mit der Variable \$content) und darunter dann den Block "footer".

Das Layout benennen wir mit "ausbloecken".


Layouts

Erstelle Hilfe


Layout-Beschreibung (optional)

Summary (optional)

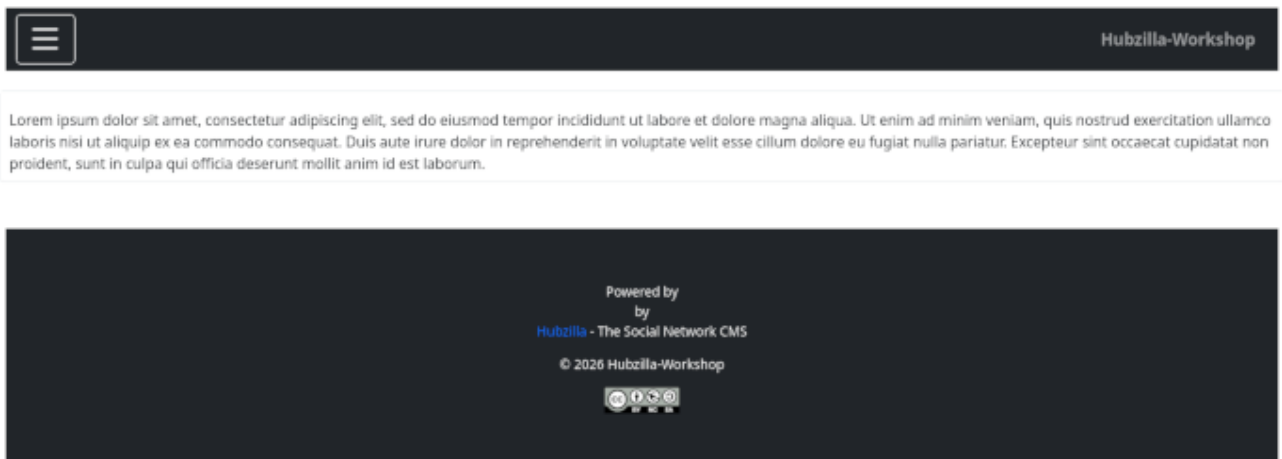
ausbloecken

[template]zen[/template] 

[region=content]
[block]navigation[/block]
\$content
[block]footer[/block]
[/region]

 Absenden

Wenn wir uns aus der Layout-Liste ein solches Layout per Klick auf den Namen anzeigen lassen, wird es uns mit einem Lorem-Ipsum-Text als Platzhalter für \$content angezeigt:



Der Inhalt für die Block-Webseite

Nun erstellen wir eine weitere Seite, welche den bisherigen Inhalt, aber ohne die Navigationsleiste, ohne das Sidebar-Menü und ohne den Footer darstellt und lassen es mit dem gerade frisch angelegten Layout "ausblöcken" darstellen. Diese Seite nennen wir schlicht "home2".

Webseiten

+ Erstelle

Art des Seiteninhalts
HTML



Seiten-Layout
ausblöcken

Mit dem Gestaltungswerkzeug kannst Du Deine eigenen Layouts erstellen

Titel (optional)

Summary (optional)
home2

```
<style>
.slide-card {
  transform: translateX(40px);
  opacity: 0;
  transition: transform 600ms cubic-bezier(.22,.9,.39,1), opacity 600ms ease;
}
.slide-card.in-view {
  transform: translateX(0);
  opacity: 1;
}
```

  Absenden

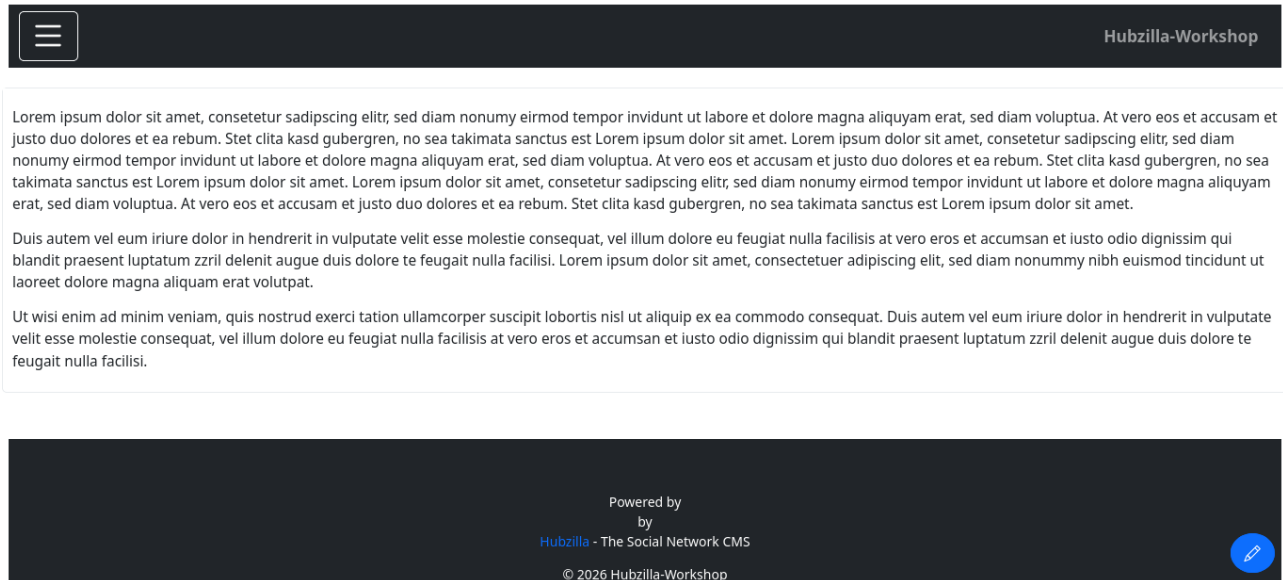
Das Ergebnis unterscheidet sich bei Aufruf nicht von der monolithischen Version "home":

<https://hub.hubzilla.hu/page/bluenomad/home2>

Aber wir haben jetzt den Vorteil, dass wir, wenn wir Seiten mit anderen Inhalten anbieten möchten, dieses Layout nutzen können, um den Inhalt mit der Navigationsleiste, dem Seitenmenü und dem Footer "garnieren" können.

Eine weitere "sinnfreie" Seite mit dem Block-Layout

Erstellen wir also eine weitere Seite, z.B. "home0", geben, damit wir etwas haben, als Inhalt drei Lorem-Ipsum-Absätze ein und nutzen wieder das Layout "ausblöcken", dann wird der "sinnfreie" Inhalt genauso dargestellt, wie bei unserer Hauptseite aus Blöcken:



Blöcke mit dem PDL-Editor nutzen

Eine weitere Möglichkeit, Blöcke zu nutzen, ist die Verwendung in den eigentlichen Hubzilla-Ansichten, also auf Seiten wie die Kanalanzeige, die Streamansicht, die Profilansicht etc.

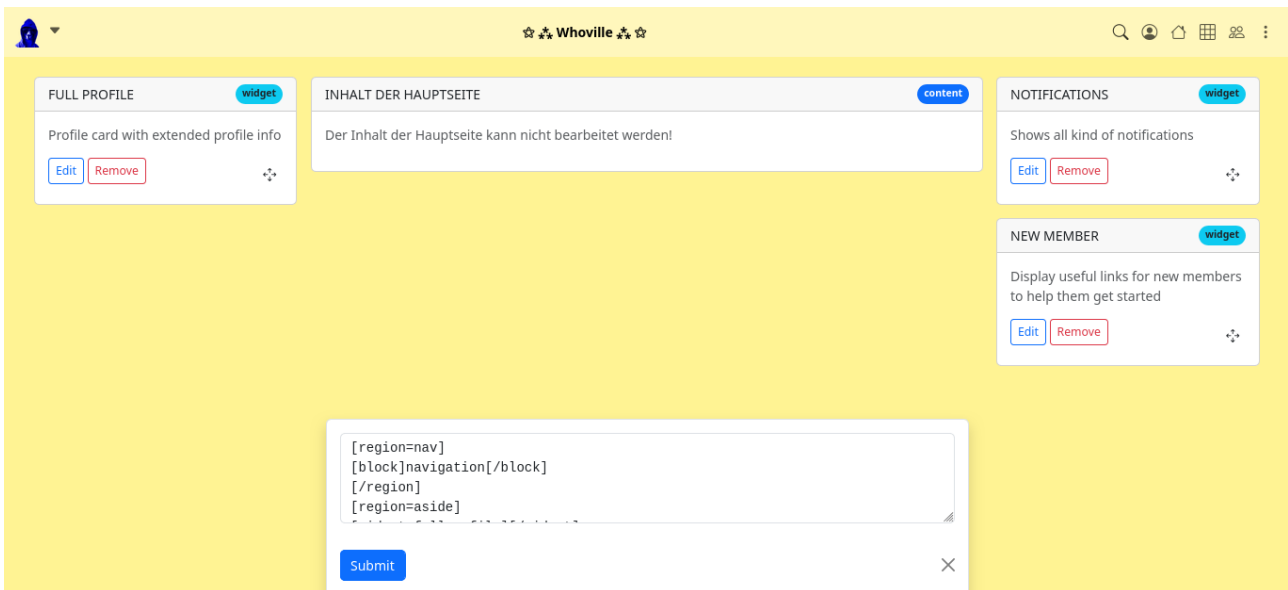
Wählt man im PDL-Editor das entsprechende Modul, dann findet man unter Items auch die selbst erstellten Blöcke und kann sie in das Layout des Moduls einfügen.

Damit kann man dann bestimmte Ansichten seines Kanals mit Elementen bestücken, die man z.B. so auch auf einer seiner Webseiten verwendet.

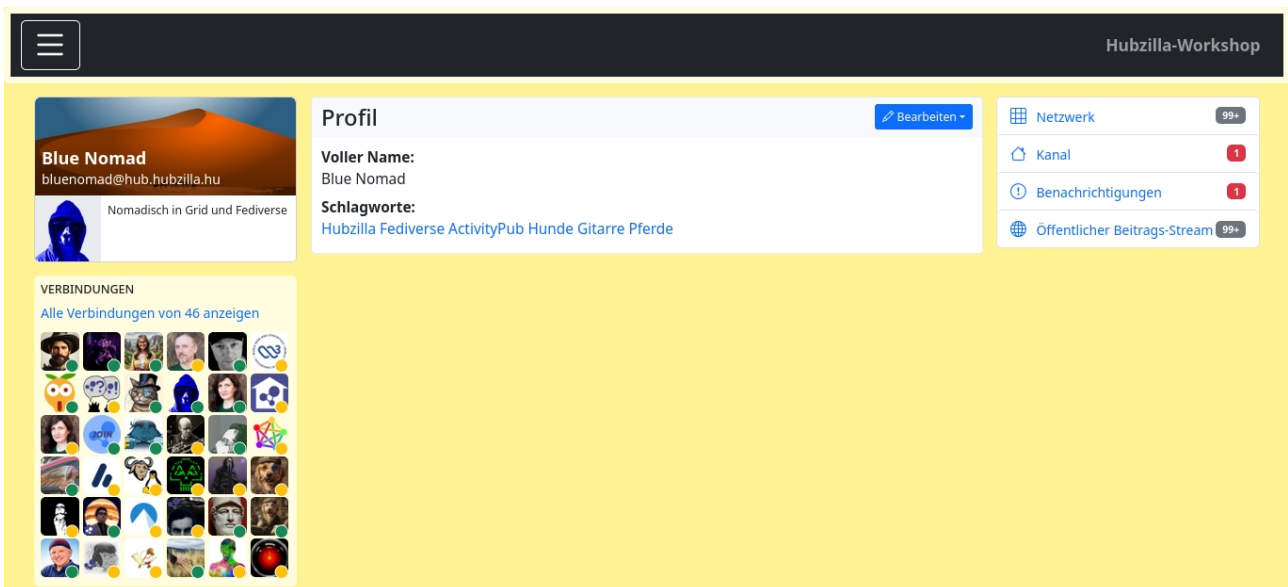
Das geht übrigens auch mit der Navigationsleiste. Man kann für ein Modul die hubzilla-eigene Navigationsleiste durch eine eigene ersetzen. Das geht zwar nicht, indem man das Item zum Navigationsbereich schiebt (dieser steht im PDL-Editor nämlich gar nicht zur Verfügung), aber man kann das Layout des Moduls auch mit dem Menüpunkt "Source" im Quelltext bearbeiten. Fügt man dort `[region=nav][region]` ein und dort zwischen den Tags den Code zum Einbetten eines Blocks, wird die Navigationsleiste durch diesen Block ersetzt.

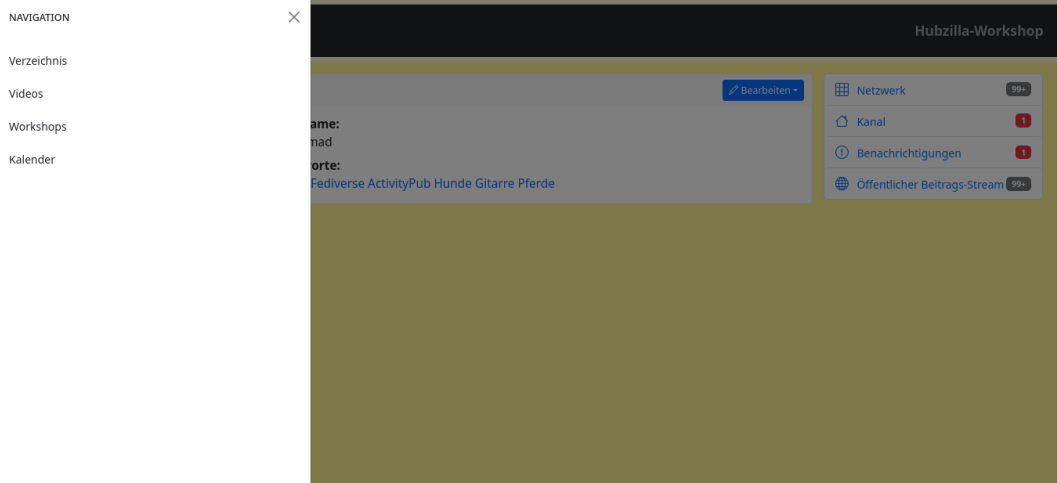
Ergänzen wir beispielsweise nun im Modul "profile", das die Profilansicht gestaltet, den Code

```
[region=nav]
[block]navigation[/block]
[/region]
```



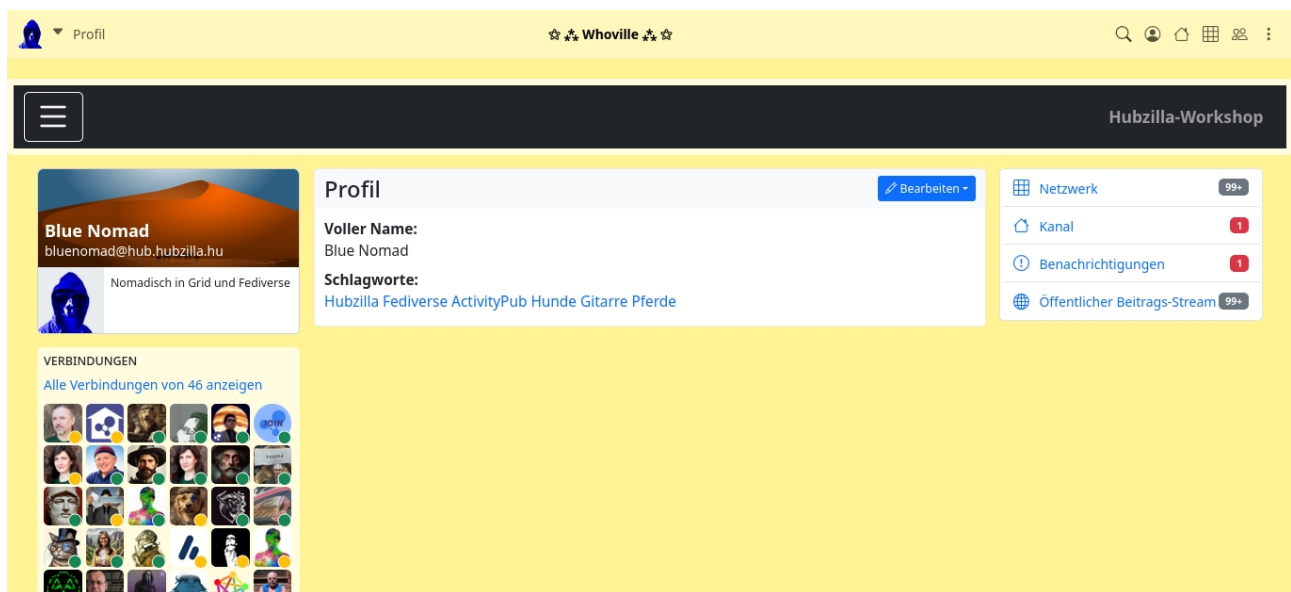
wird bei Aufruf der Profilseite nun nicht mehr die Hubzilla-Navigationsleiste angezeigt, sondern unsere eigene.





Gimmick: Das einblendbare Off-Canvas Seitenmenü steht auch zur Verfügung, weil es Bestandteil des Blocks ist. 😊

Hinweis: Fügt man in der Region "nav" zusätzlich noch die Variable \$nav ein, so wird nicht nur die eigene Navigationsleiste, sondern zusätzlich auch noch die Hubzilla-Navigationsleiste angezeigt.



Fazit

Die App "Webseiten" macht es möglich, mit Hubzilla anspruchsvolle Webseiten und sogar komplette Webpräsenzen zu erstellen und anzubieten.

Mit Blöcken ist es außerdem möglich, seinen kompletten Kanal mithilfe des PDL-Editors zu ergänzen, anzupassen und mit einem einheitlichen Erscheinungsbild zu versehen, was es z.B. möglich macht, dass sich der gesamte Kanal an das Design der eigenen Webseite anpasst.

Einen hab' ich noch!

Dank des Engagements von [@Chris](#) und dem Hubzilla-Entwickler [@Harald Eilertsen](#) gibt es nun auch ein Addon, welches für das Bearbeiten von Blöcken und Webseiten einen WYSIWYG-Webeditor zur Verfügung stellt. Ist das Addon "[Summernote Web Editor](#)" auf dem eigenen Hub installiert und aktiviert, kann man in seinem Kanal die App "Summernote Web Editor" installieren und damit Webseiten und Blöcke mit einem grafischen Webseiteneditor bearbeiten.