



Praktische **Anleitung** für Elementor **Pro**



V 3.0.0 1/2025

Harry Martin

Praktische Anleitung für Elementor Pro

(Elementor Pro Skriptum)

Lizenz

Veröffentlicht unter CC BY-NC-ND 4.0

<https://creativecommons.org/licenses/by-nc-nd/4.0/deed.de>



Kurzform: Weitergabe unter Namensnennung nicht kommerziell erlaubt, wenn keine Änderungen vorgenommen wurde.

Trademark

Elementor ist eine Marke und ein Unternehmen.

Diese Veröffentlichung ist davon unabhängig und in keiner Weise autorisiert.

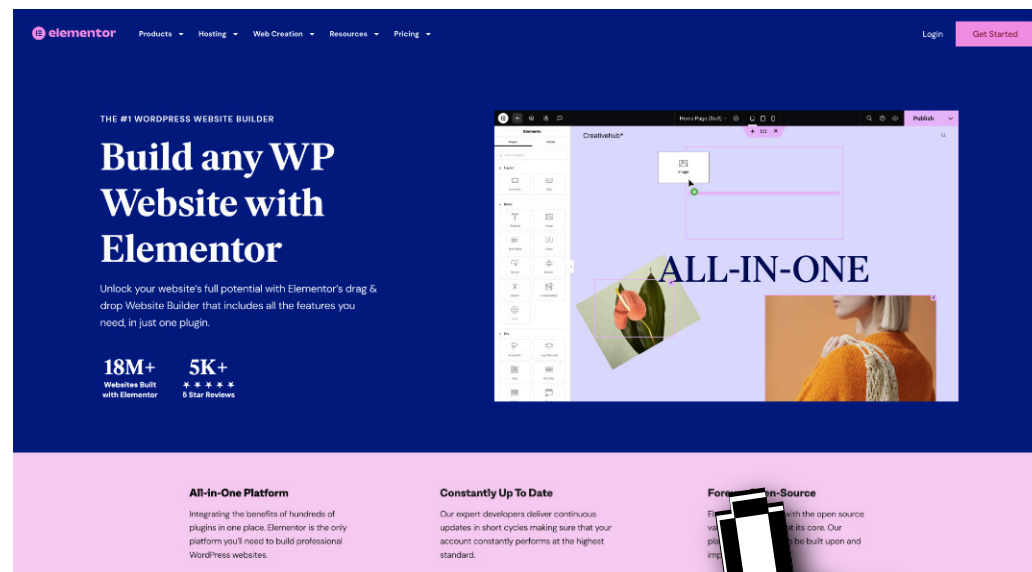
Skriptum

Der Begriff Skriptum ist ein in Österreich gebräuchlicher Terminus für eine Arbeitsunterlage einer Vorlesung. Ein Skriptum ist noch kein Buch, aber schon mehr als eine pure Transskription. Daher auch das gut ausdrückbare Format A4 in pdf mit großer Typographie auch für E-Reader.

Haftungsausschluss

Trotz sorgfältiger inhaltlicher Kontrolle übernimmt der Autor keine Haftung für die Inhalte und die verwendeten externen Links. Irrtümer, Fehler und Änderungen vorbehalten - sollten aus der Anwendung der in diesem Dokument beschriebenen Techniken und Ideen Schäden entstehen kann der Autor in keiner Weise dafür haftbar gemacht werden. Dieses Dokument soll lediglich Hilfe zur Selbsthilfe geben.

Version: 3.0.0 05012025



Du hast noch keine Elementor Pro-Lizenz? [Klick hier!](#)

Anzeige

Impressum

Ing. Harald Martin
Schwalbachgasse 6
3323 Hößgang
Österreich

Kontakt

werkform.at
service@werkform.at

1. Einleitung	4	5.3. Grid Container	58	10.4. CSS in den Webseite Einstellungen	100
1.1. Formalitäten	4	5.4. Der Container als Designelement	59	10.5. Stil anwenden - Stil zurücksetzen	101
1.2. Der Einstieg	5				
1.3. Elementor Oberfläche	7	6. Widgets	62	11. Formulare	103
1.4. Theme für Elementor	12	6.1. Erweiterte Funktionen	62	11.1. Das Widget Formular	103
		6.2. Layout und Grundlegende Widgets	69	11.2. Besondere Feldtypen	107
2. Einstellungen	14	6.3. Pro	71	11.3. Gestaltung der Formularelemente	110
2.1. Programmeinstellungen	14	6.4. Generell	73	11.4. Google reCaptcha V3 verwenden	111
2.2. Website-Einstellungen	17	6.5. Link in Bio	74	11.5. Mehrstufige Formulare	113
2.3. Benutzer-Einstellungen	25	6.6. Website	75	11.6. Formulardaten Speichern	114
2.4. Element Manager	27	6.7. Single	76	11.7. Probleme beim E-Mail-Versand	115
2.5. Seite Einstellungen	28	6.8. WordPress	76		
				12. Dynamik mit Elementor	116
3. Elementor und WordPress	30	7. Responsive Design	77	12.1. Benutzerdefinierte Felder	118
3.1. Import und Export mit Site Kit	30	7.1. Schriften responsive einstellen	79	12.2. Advanced Custom Fields - ACF	119
3.2. Custom Code	31	7.2. Container responsive gestalten	80	12.3. Eigene Post Typen	122
3.3. Schriften Lokal einbinden	32	7.3. Zusätzliche Breakpoints	81		
3.4. Eigene Icons verwenden	34				
3.5. Integration von Diensten	35	8. Templates	82		
3.6. Elementor Werkzeuge	35	8.1. WordPress Templates	82		
3.7. Weitere Elementor Plugins & Services	37	8.2. Die Elementor Bibliothek	83		
		8.3. Eigene Elementor Templates erstellen	86		
4. Theme Builder	39	8.4. Globale Widgets	87		
4.1. Kopfzeile - Header	41	8.5. Eigene Container als dynamische Templates	88		
4.2. Fußzeile - Footer	44	8.6. Popups	89		
4.3. Single	45	8.7. Floating Elements	90		
4.4. Single Post	46				
4.5. Single Page	47	9. Schriften	91		
4.6. Die 404-Seite	48	9.1. Typographie	92		
4.7. Archive	48	9.2. Mehrspaltiger Text	96		
4.8. Loop	50	9.3. Automatische Silbentrennung	97		
4.9. Suchergebnisse	51				
		10. CSS in Elementor	98		
5. Container	52	10.1. CSS über die ID im Widget einfügen	98		
5.1. Flexbox Container	53	10.2. CSS über Klassen im Widget anwenden	99		
5.2. Einstellungen für Elemente	57	10.3. CSS für Seiten definieren	100		

"

Als Trainer und Freelancer ist es meine Aufgabe meinen Kunden die passenden Werkzeuge in die Hand zu geben, damit sie ihre Vorstellungen umsetzen können.



Harry Martin
werkform

werk form

Erstgespräch buchen





Anzeige

1. Einleitung

1.1. Formalitäten

Für die praktische Arbeit mit diesem Skriptum wird vorausgesetzt, dass eine aktuelle WordPress-Installation besteht und die Erweiterungen **Elementor** und **Elementor Pro** installiert und aktiviert sind. Da laufend Updates erfolgen – und es unmöglich ist immer die selben Versionen und Subversionen zu verwenden – wird versucht, alle Schritte und Erklärungen so zu beschreiben, dass diese auch in leicht veralteten (und hoffentlich auch neueren Versionen) funktionieren. Auch die Inhalte dieses Skriptums werden laufend aktualisiert.

Gestartet wurde dieses Projekt im Sommer 2021 unter den Versionen WordPress 5.8 und Elementor 3.3.1 sowie Elementor Pro 3.3.5. Der Autor ist Freelancer, die Veröffentlichung erfolgt im Eigenverlag und ist von keinem Unternehmen gefördert oder autorisiert. Diese Version 3.0 dieses Dokuments wurde im Oktober 2024 unter WordPress 6.6 und Elementor 3.24 begonnen.

Das Skriptum wendet sich an **Publisher** mit **WordPress Grundkenntnissen** die zügig konsistente Webdesigns umsetzen möchten. Alle Beispiele und Screenshots wurden am Mac erstellt und umgesetzt. Als Standardbrowser wird Google Chrome verwendet, zur Kontrolle wird auch Safari und Firefox eingesetzt.

WordPress, Elementor, Google, Chrome und Mac sind eingetragene **Warenzeichen**. Alle Inhalte, Screenshots und Grafiken sind – so ferne sie nicht anderes gekennzeichnet sind – vom Autor angefertigt und unterliegen dem Copyright (siehe Seite 2), dürfen im Rahmen der CC BY-NC-ND 4.0 verwendet werden.

Oft werden **Pfadangaben** verwendet, zur einfacheren Unterscheidung werden Hinweise zu Angaben in WordPress **blau**, Pfade innerhalb von Elementor **rosa** hinterlegt.

Dashboard meint das WordPress Dashboard (Backend) wie man es nach dem Login vorfindet. **Panel** meint die Elementor-Umgebung, wie man sie beim Bearbeiten einer Seite oder eines Templates vor sich hat. Diese Pfade können auch unvollständig sein oder Auslassungen ... beinhalten. Es kann vorkommen,

Beispiele für Pfadangaben:

Dashboard > Elementor > Einstellungen > *Integration* > *reCaptcha V3*

Im WordPress Dashboard ist der Haupt-Menüpunkt Elementor und der Unter-Menüpunkt Einstellungen zu suchen. Die kursive Darstellung von Integration bedeutet, dass sich dieser Punkt innerhalb der Seite (ein eigener Tab) befindet, reCaptcha V3 ist innerhalb dieses Tabs zu finden.

Panel > + (Element hinzufügen) > Pro > Loop Grid

Bei aktivem Elementor (=Panel) ist im Bereich Pro das Widget Loop Grid gemeint.

Panel > ~~Webseite-Einstellungen~~ > Globale Farben

Bei aktivem Elementor ist das Icon (Schieberegler) Elementor Einstellungen gemeint, darüber geht es zu den Globale Farben. In diesem Beispiel wurde der Zwischentitel Design System weggelassen.

Panel > Seite Einstellungen > Stil > Body-Stil > Hintergrundtyp

Unter Seite Einstellungen ist das Zahnrad in der Mitte gemeint, danach der Tab Stil mit dem Untertab Body-Stil und dort der Hintergrundtyp.

dass die Pfade nicht mehr stimmen weil Elementor Namen ändert oder Funktionen verschiebt. Oft kommt es bei Elementor zu **Fehlern bei der Anzeige von übersetzten Begriffen auf Deutsch**. Beispiel: *Hintergrundtyp* wird als *Background Type* angezeigt. Diese Phänomene kommen und gehen und sind nicht immer nachvollziehbar – aber sie helfen, die engl. Originalbegriffe zu lernen.

1.2. Der Einstieg

Wozu benötigt man Elementor? Was kann man damit tun, was man nicht auch direkt in WordPress tun kann? Um die Funktionsweise dieses Werkzeuges zu verstehen ist es hilfreich sich verschiedene Ebenen vorzustellen. Diese sind notwendig um eine dynamische (von einem Content Management System / **CMS** unterstützte) Webseite zu betreiben.

WordPress-Websites benötigen ein **Theme**, das die Aufgabe der Gestaltung der Inhalte übernimmt. Dort werden Gestaltungselemente, Templates und viele weitere Funktionen zusammengefasst. Dazu kommen noch die Plugins, die sich – je nach Funktion – an verschiedenen Stellen in das CMS einhaken und diverse Funktionen übernehmen, die nicht im Kern von WordPress vorhanden sind oder vorhandene Funktionen ersetzen.

PageBuilder dienen dazu, die ursprünglich sehr rudimentären Gestaltungsmöglichkeiten zu erweitern. WordPress hat mit der Einführung von Gutenberg selber den Weg vom TextEditor (TinyMCE) hin zu einem Blocksystem angetreten. Mit der Version 5.8 wurde das Full Page Editing (jetzt *Website Editor*) eingeführt. Damit soll die Blockfunktionalität auf die gesamte Website ausgedehnt werden.

Elementor beschreitet diesen Weg schon ein wenig länger, der **Theme Builder** ist genau dazu da und ersetzt die wesentlichsten Funktionen eines Themes. Dazu kommen die Gestaltungselemente innerhalb einer Seite/eines Beitrages, also die klassische *Page Builder* Funktion.

Weitere Funktionen die im Elementor Pro enthalten sind (der *Popup Builder*, das *Formularsystem*, die *Templates* etc.) können eine ganze Reihe von Plugins ersetzen. Ein *Site Builder* ist Elementor, weil durch die Angabe von Bedingun-



WordPress (blau) benötigt ein Theme und oft Plugins (grün), das Theme verwendet Templates (violett). Zur Gestaltung verwendet man einen PageBuilder (rot) wie Elementor, Gutenberg, ...

Praktisch wirken diese Dinge miteinander und bauen aufeinander auf.

Elementor ist ein *Page Builder*, **Elementor Pro** ist (besonders in Verbindung mit dem Theme Hello Elementor) mehr: er deckt Anforderungen an das Theme, einige Plugins und des Page Builders ab.



Elementor ist in der freien Version ein **PageBuilder**.

In der **Pro Version** ist er:

- Page Builder
- Theme Builder
- Site Builder
- Popup Builder
- Form Builder

gen die verschiedenen Templates und sogar einzelne Elemente angezeigt werden können - oder eben nicht. **Elementor ist ein Builder.**

Elementor ist also die erste Ebene, die direkt unter der sichtbaren Webseite liegt. Man kann Elementor als Pendant zu einem Layout-Programm für Druckwerke verstehen: er hilft dabei Texte und Bilder auf einer Seite anzuordnen. Die Texte werden in der WordPress-Datenbank gespeichert. Die Bilder liegen als Files am Webserver und Elementor ist in seiner Funktion als Theme-Builder dafür zuständig, diese Texte und Bilder – gemeinsam mit anderen Dingen wie Grafiken, Videos etc. – zu dynamischen Inhalten zu kombinieren. Und das Ganze noch dazu jeweils abhängig vom Ausgabegerät, angereichert mit einer ordentlichen Portion möglicher Interaktivität durch den Benutzer wie Formulare, Navigationen etc.

Diese Komplexität existiert immer. Die Aufgabe des Webdesigners besteht darin diese Schwierigkeiten vor dem Benutzer zu verstecken, und ihm eine einfache, sofort verständliche Antwort auf die implizierte Fragestellung – also die Zielsetzung des Projektes – zu geben. Einfacher formuliert: zu gestalten.

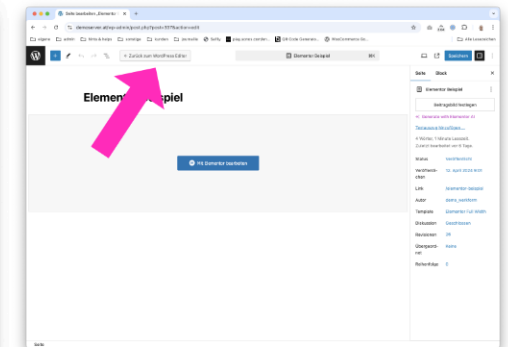
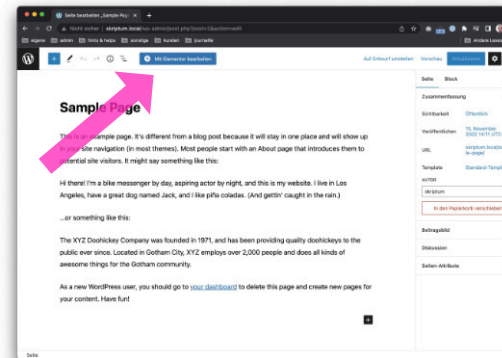
Auch Elementor selber ist eine Webanwendung. Es stehen also nur jene beschränkten Mittel von HTML, CSS und Skripten zur Verfügung die auch für komplette Webseiten verwendet werden. Das erklärt auch, warum man Elementor nicht einfach wie ein Programm am Endgerät starten kann. Elementor benötigt immer ein Werkstück (also eine Seite, einen Beitrag oder ein Template) um zu arbeiten. **Ohne Blatt kein Layout!**

Hinweis: Wenn man auf einer mit Elementor bearbeiteten Seite die Funktion *Zurück zum WordPress Editor* auswählt zerfällt die Gestaltung. Der Inhalt bleibt jedoch erhalten.

Dashboard > Seiten > Alle Seiten > *Beispiel-Seite* > *Bearbeiten*

Um damit Elementor zu arbeiten muss die Bearbeitung mit diesem Editor gestartet werden. Jede WordPress-Installation hat eine Beispielseite, in der Kopfzeile findet man den Button:

Mit Elementor Bearbeiten



WordPress öffnet im Dashboard den Gutenberg Editor. In der Kopfzeile kann man die Bearbeitung mit Elementor starten. Dadurch wird Elementor als bevorzugter Editor mit dieser Seite verknüpft.

Sobald eine Seite mit Elementor bearbeitet wurde, kann sie auch wieder entkoppelt werden. Dazu ist der Button:

← Zurück zum WordPress Editor

1.3. Elementor Oberfläche

Die Elementor-Oberfläche wurde vor einiger Zeit überarbeitet, die in rot und blau gehaltenen Bedienteile wurden durch einen schwarzen **Editor-Top-Bar** ersetzt. Dieser ist entweder bereits aktiviert (zB. innerhalb der Elementor-Hosting Umgebung) oder sollte aktiviert werden:

Dashboard > Elementor > Einstellungen > *Eigenschaften* > *Editor Top Bar*

Die Einstellung ist auf **Aktiv** zu setzen, irgendwann wird das die Grundeinstellung werden und die bunten Panele werden verschwinden. Der **Top Bar** ist 3geteilt, im ersten Block links sind

- Elementor-Menü (Logo)
- Element hinzufügen (+)
- Website-Einstellungen
- Struktur (ehem. Navigator)
- Notes

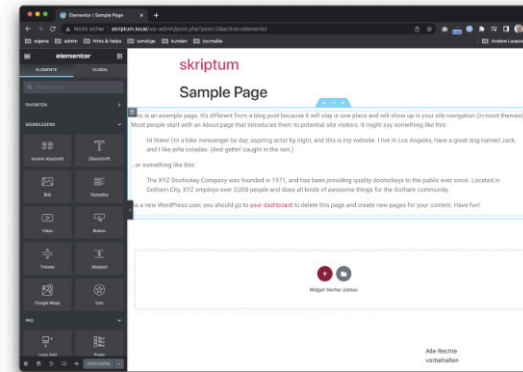
in der Mitte des Balkens sind

- Recent Menü
- Seite Einstellungen (Zahnrad)
- Responsive Modi (Icons)

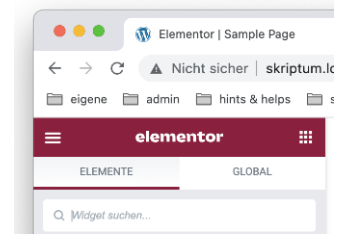
im rechten Block sind die Funktionen für

- Finder (Lupe)
- Hilfe
- What's New
- Vorschau
- Veröffentlichen
- Optionen speichern (Chevron ganz rechts).

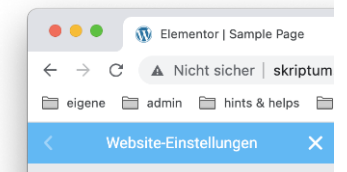
Die Verteilung der Funktionen ist nicht ganz schlüssig, mit Änderungen in der Position oder Umbenennungen muss man rechnen, die grundsätzliche Funktionalität wird aber sicher bleiben.



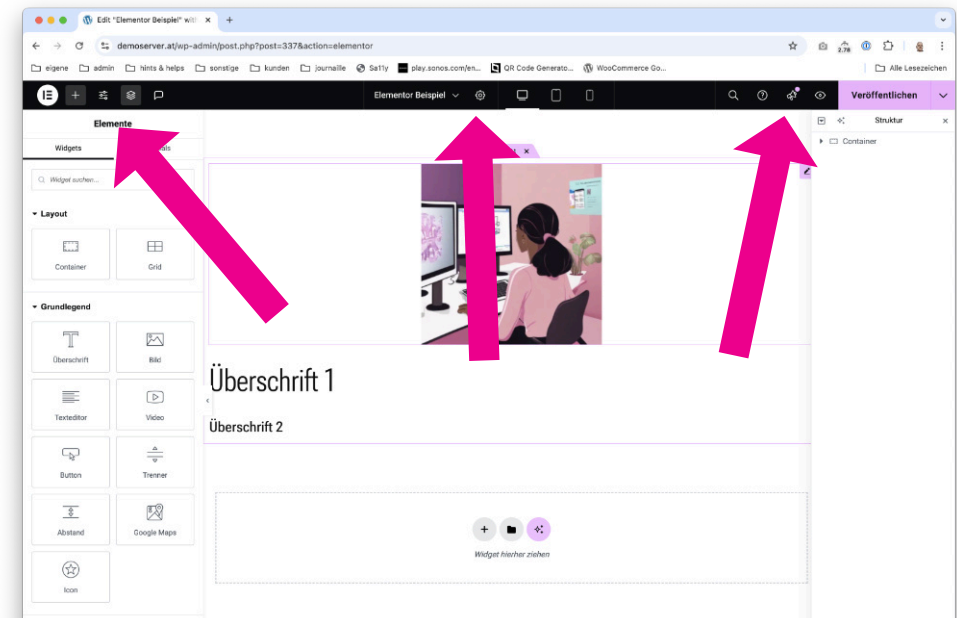
Die alte Elementor-Oberfläche war mit einem linksseitigem Panel organisiert. Wer Elementor Hosting verwendet oder die Grundeinstellung ändert bekommt das bessere Interface: den **Editor Top Bar**. Die Farben wurden gestrichen, nun ist das Pa-nel in schwarz/weiß kontrastreich und klar.



ALT: Das rote Panel zeigt den normalen Bearbeitungsmodus.an.



ALT: Ist das Panel blau befindet man sich bei den Einstellungen.



Der Linke Panelbereich / Top Bar

Das Elementor Menü

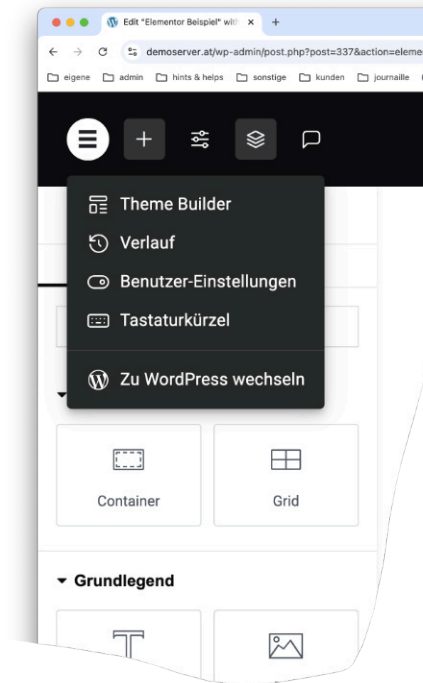
Theme Builder. Ist eine Übersichtsseite über alle Templates die man benötigt, um alle Bestandteile einer Website zu gestalten. (Header, Footer, Einzelseitenvorlagen, etc.) Näheres siehe Kapitel 4.

Verlauf. Neben der im Umgang mit allen Programmen üblichen Rückgängig-Funktion (Befehl-z oder Strg-z) speichert Elementor auch die Bearbeitungsschritte (solange der Editor geöffnet ist). Man kann damit mehrere Schritte zurück kehren (*Aktionen*) und von diesem Punkt weiter arbeiten. Ausserdem kann man zu früheren **gespeicherten Versionen** zurückkehren (*Revisionen*).

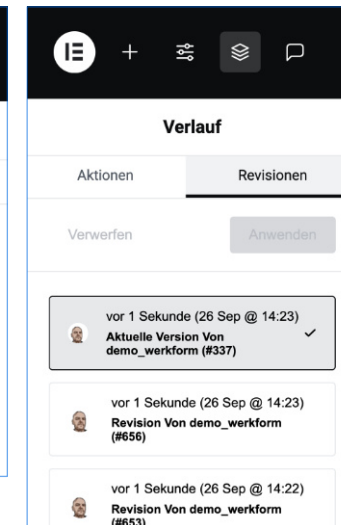
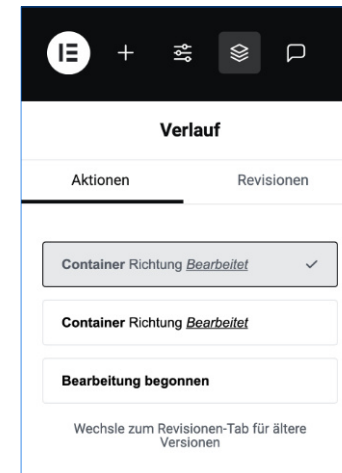
Benutzer-Einstellungen. Hier kann man einige persönliche Einstellungen treffen (Dark Mode etc.). Wichtig ist die Option *Show Hidden Elements* womit man ausgeblendete Elemente auch im Editor aus/einblenden kann. Weiters ist *Navigation > Gehe zu* hier zu konfigurieren. Damit legt man fest, wo man landet sobald man Elementor verläßt und zurück zum WordPress Dashboard kommt. Die anderen Optionen sind ± selbst erklärend, mit Sprachengewirr ist hier zu rechnen.

Tastaturkürzel. Zeigt eine Übersicht über alle Befehle an, die man direkt über die Tastatur erreichen kann.

Zu WordPress wechseln. Damit kehrt man zurück *zum WordPress Dashboard*. Leider ist die hier verwendete Sprache ungenau: wir arbeiten *immer in WordPress*, auch wenn wir Elementor verwenden. Die Idee, dass Elementor etwas anderes als WordPress ist mag dem Marketing geschuldet sein, ist aber verwirrend und irreführend.



Das Elementor Menü führt zu grundsätzlichen Funktionen und - ganz wichtig - auch wieder zurück zum WordPress Dashboard.



Über den Verlauf kann man gesetzte Schritte rückgängig machen.

Element hinzufügen +

Bei aktivierter Funktion öffnet sich am linken Rand ein seitenhohes Panel mit Scrollbar, indem alle Widgets die verfügbar sind angezeigt werden. Welche das sind hängt von der Lizenz, den Einstellungen und Erweiterungen ab.

Neben dem aktivierten Tab **Widgets** gibt es einen weiteren namens **Globals**. Dort findet man alle Widgets, die man selbst als Globale Widgets gespeichert hat. (Siehe Kapitel 8.4.)

Darunter ist eine **Suchmaske** die den schnellen Zugriff auf das gesuchte Widget ermöglicht. Das funktioniert übrigens auch mehrsprachig: die Suche nach *Beitrag* bringt die selben Ergebnisse wie die Suche nach *Post*.

Website Einstellungen

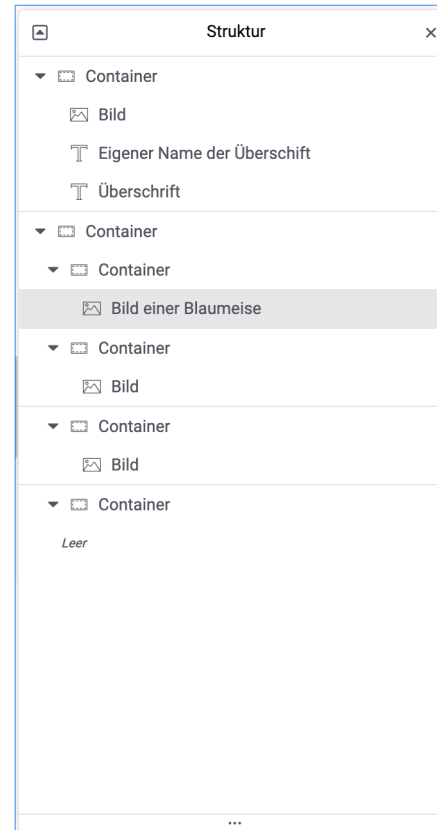
Öffnet im linken Panel die Tab-Gruppen zu jenen Einstellungen, die das gesamte Projekt betreffen. Mehr dazu siehe Kapitel 2.2.

Struktur (früher Navigator)

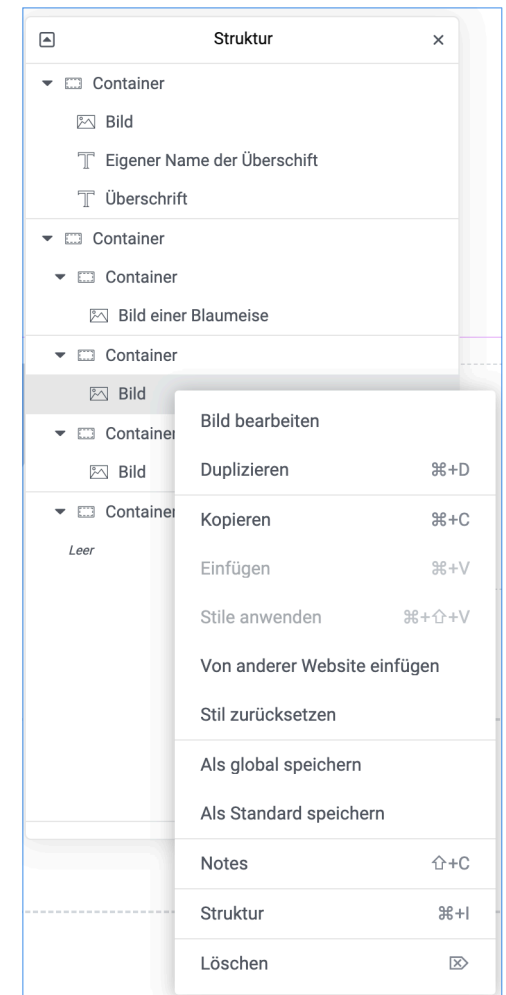
Der Schalter blendet ein Panel ein/aus, das entweder als frei verschiebbares Fenster über der Elementor-Arbeitsfläche liegt oder aber am rechten Rand fixiert werden kann.

Dieses sehr mächtige Werkzeug zeigt alle im aktuellen Dokument verwendeten Widgets an, so verschachtelt und in jener Reihenfolge, wie sie eben platziert wurden.

Jedes Element kann hier umbenannt (Doppelklick auf den Namen) werden und auch verschoben. Durch markieren mehrerer Zeilen können diese auch gemeinsam verschoben oder gelöscht werden. Auch über die Strukturpalette kann das Kontext-Menü jedem Widget aufgerufen werden (rechts-Klick bzw. `ctl`-Klick).



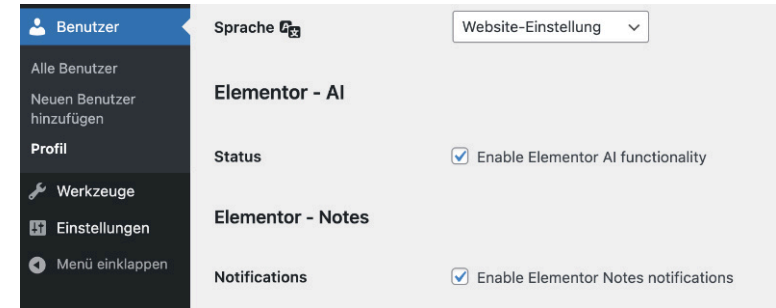
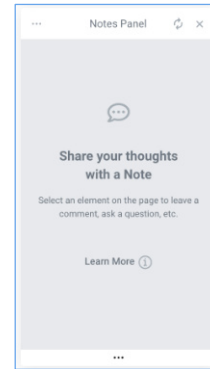
Das Struktur-Modalfenster dient der Übersicht und dem schnellen Zugang zu den einzelnen Elementen des Dokuments. Verschieben, Löschen, Umbenennen und Zugang zum Context-Menü jedes Elements sind die Kernfunktionen.



Notes

Die Notes-Funktion ist ebenfalls in einem Floating Panel organisiert. Gedacht ist diese Funktion, um Mitteilungen direkt im Dokument zu speichern, was besonders bei Projekten mit mehreren EditorInnen praktisch sein kann. Diese Funktion ist für bestimmte Benutzer (wie übrigens auch die Elementor-AI-Funktion) abschaltbar:

Dashboard > Benutzer > Profil > *Elementor AI* bzw. *Elementor Notes*



Notes: wer sie nicht benötigt kann User von der Nutzung ausschliessen.

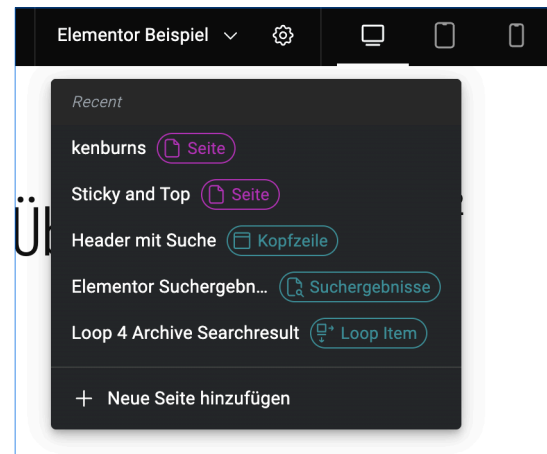
Der Mittlere Bereich im Top Bar

Recent. Neben dem Namen des aktuellen Dokuments führt ein kleines Chevron zu den kürzlich bearbeiteten Dokumenten in Elementor. Das ist praktisch, da man hier schnell zwischen dem aktuellen Dokument und zB. den Templates wechseln kann. **+ Neue Seite hinzufügen** unter der Auflistung erstellt eine neue Seite, die Elementor als Editor zugewiesen hat. Damit das funktioniert speichert Elementor diese neue Seite unter dem Namen *Elementor Seite #1677*, wobei die Nummer am Schluß die WordPress-ID ist. Auch bei neu angelegten Templates macht das Elementor so: also immer gleich umbenennen, wenn man auch morgen noch wissen will was auf dieser Seite ist!

Seite Einstellungen / Zahnrad. Darüber gelangt man zu den Einstellungen, die das aktuelle Dokument betreffen. Das gilt für Seiten genauso wie für Templates und Popups. Näheres siehe Kapitel 2.4.

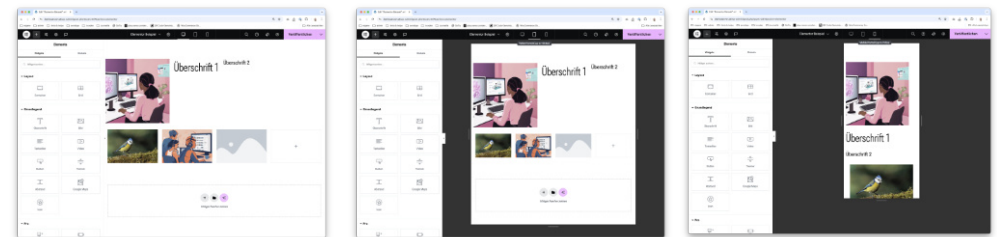
Responsive Einstellungen: Desktop | Tablet Portrait | Mobile Portrait

Die 3 Icons zeigen die voreingestellten, aktivierten Responsive Modi. Der aktuelle Modus (vmtl. Desktop) wird durch Unterstrich angezeigt. Durch einfaches anklicken der Icons kann man jederzeit zwischen den Modi wechseln und so das Design auf sein Verhalten auf anderen Bildschirmgrößen überprüfen. Näheres siehe Kapitel 7.



Das Recent Menü zeigt aktuelle Elementor-Dokumente an. Praktisch: nicht nur der Name, sondern auch die Art des Dokumentes wird angezeigt.

Die 3 Responsive-Icons sind der schnellste Weg um das Dokument schon im Editor in einer anderen Auflösung zu bearbeiten.



Die Ansicht in einem anderen Bildschirmmodus ist immer nur einen Klick entfernt.

Der Rechte Bereich im Top Bar

Finder. Mit nur einem Tastenklick bietet der Elementor-Finder eine Auswahl von zu bearbeitenden Dokumenten, Generelle Links und auch Einstellungen. Damit kann man zwischen den Dokumenten wechseln ohne Elementor zu verlassen und neu aufzurufen - was aufgrund der doch manchmal längeren Ladezeiten recht lästig werden kann.

Hilfe. Öffnet einen neuen Browser-Tab mit der Startseite der Elementor Online-Hilfe.

What's New. Öffnet eine Panel mit den neuesten Nachrichten aus dem Hause Elementor.

Vorschau der Änderungen. Öffnet einen neuen Browser-Tab mit einer Vorschau des aktuellen Dokumentes.

Veröffentlichen. Veröffentlicht oder aktualisiert das aktuelle Dokument.

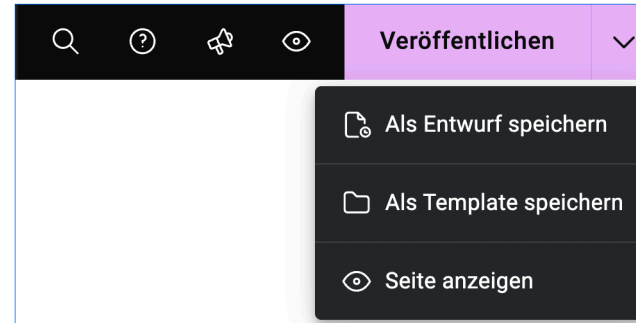
Im Submenü zu finden sind:

Als Entwurf speichern. Speichert das Dokument im Status *Entwurf (Draft)*.

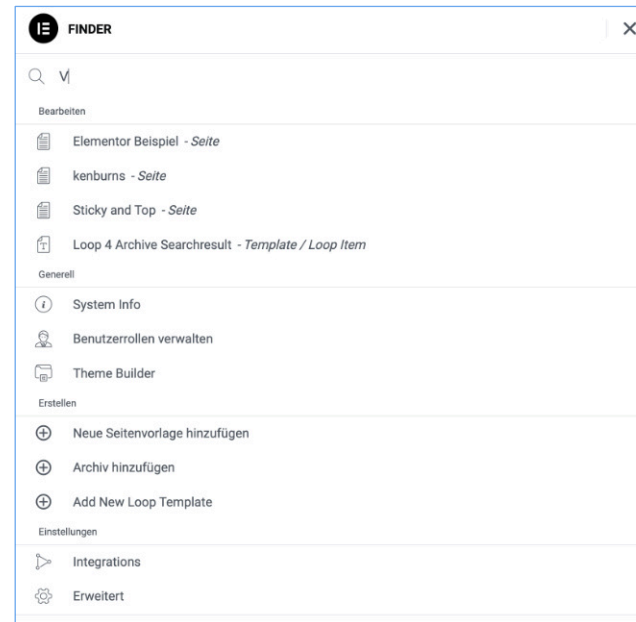
Als Template speichern. Speichert das Dokument als Template. Damit ist es im Dashboard zu finden:

Dashboard > Templates > Gespeicherte Templates

Seite anzeigen. Zeigt die Seite in einem neuen Browser-Tab an.



Im Top Bar rechts sind die verschiedenen Speicher- und Vorschaufunktionen zu finden.



Der Finder ist die Abkürzung wenn man innerhalb von Elementor woanders hingelangen will. Man kann zwischen Dokumenten wechseln oder zu den Einstellungen gelangen.

1.4. Theme für Elementor

Die kurze Anweisung ist:

Dashboard > Design > Themes > *Theme hinzufügen* > *Hello Elementor*

Hello Elementor ist der Name des Themes, das von Elementor selber publiziert wird. Ursprünglich war es ein Barebone-Theme. Dh. es hatte überhaupt keine Design-Funktionen. Später wurden rudimentäre Header- und Footer-Optionen eingebaut. Diese benötigt man bei Arbeit mit der Pro-Version von Elementor nicht. Ausschalten lassen sich diese Teile unter

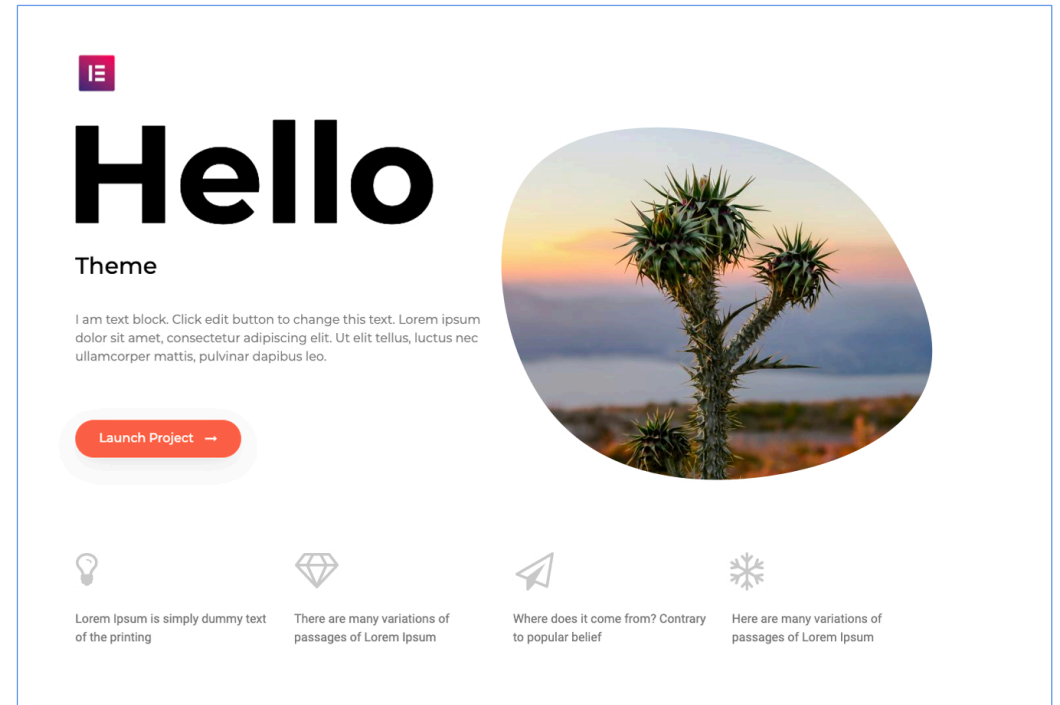
Dashboard > Elementor > Einstellungen > *Eigenschaften* | *Header and Footer*

Was ist aber ein **Theme** überhaupt und warum muss man darüber (kurz) nachdenken? Ganz einfach erklärt ist es so: WordPress sorgt für das WAS, das darüber gelegte Theme sorgt für das WIE die Inhalte in der Webseite angezeigt werden. Ein Theme ist also die Ebene, die über der technischen Ebene des CMS (Content Management System) liegt und alle gestalterischen Dinge wie Farben, Schriften, Layout, Menügestaltung, etc. bestimmt.

Daher kann man auch das Theme einer Webseite wechseln ohne die Inhalte der Seite zu verändern – theoretisch zumindest. Denn die Themes wurden mit den Jahren zu tw. riesigen Containern die verschiedenste JavaScript Bibliotheken, Fonts und jede Menge Plugins mitgebracht haben.

PageBuilder waren anfänglich einfach Tools, welche auf verschiedene Weise versucht haben die engen Design- und Usability-Grenzen von WordPress zu erweitern. Die Entwicklung von Elementor startete 2016 genau so: als Page-Builder, der möglichst universell einsetzbar mit allen Themes arbeiten kann. In der Zwischenzeit ist die Entwicklung an einem Punkt angekommen, wo das Konzept der Themes immer mehr zu hinterfragen ist.

1. Die **Einstellungen im Elementor** übernehmen in vielen Bereichen die Funktionen die bei vielen Themes im Customizer oder in eigenen Einstellungs-Panels zu finden sind.



Das Hello-Elementor Theme ist die einfachste und beste Lösung für eine Elementor-Website.

2. Der **Theme-Builder** von Elementor Pro übernimmt die Aufteilung in Seitenbereiche (Header, Footer, ...) und sorgt mit seinem Management der Templates für konsistente Websites.

Kann man auch andere Themes verwenden? Natürlich. Genau dazu wurde Elementor entwickelt, um als PageBuilder universell im WordPress-Universum eingesetzt zu werden. Das beginnt damit, dass einzelne im Elementor gestaltete Abschnitte über Shortcodes oder in (WordPress)Widget-Bereichen in Seiten eingebaut werden. Auch gibt es immer mehr Profi-Themes die von anderen PageBuildern auf Elementor wechseln und auch eigene Widgets im Paket mit anbieten. Die Frage ist immer: brauche ich das wirklich? Muss es verschiedene Dinge doppelt und dreifach geben oder reicht nicht auch eine einfache, zentrale Logik, über die ich meine unterschiedlichen Designebenen verwalte?

Je komplizierter und individueller die Aufgabe ist desto seltener wird man ein komplexes Theme verwenden, egal ob ein sogenanntes Universal-Theme oder eine spezielles, zB. auf eine Branche spezialisiertes Theme.

Die Empfehlung lautet: alle Funktionen, die nicht Teil von Elementor Pro sind, auf Ebene der Plugins ergänzen. Und auch hier möglichst kleine, spezialisierte Lösungen suchen. Also zB. ein Event-Plugin statt eines Event-Themes. Oder Eigene Post-Types und eigene Felder über ACF statt ein Theme zu nehmen, das auch nichts anderes macht (die Technik aber hinter einer anderen Oberfläche versteckt).




Borlabs Cookie

WordPress Cookie Lösung

Die beste Lösung für Deine Elementor Website!

Anzeige




elementor

Hosting und Pro-Lizenz in einem ab € 9,99 p.m.*

- 10 Gig Storage
- 14 Tage Backup
- Enterprise Cloudflare (CDN)
- Web Application Firewall
- ...

* Aktionspreis, regulär € 14,99

Anzeige



Domaintechnik®

- 550+ Top-Level-Domains
- Webhosting
- E-Mail

Hosting aus Österreich

Anzeige

2. Einstellungen

Es gibt drei verschiedene Ebenen, in denen Einstellungen vorgenommen werden können:

1. Die über das Dashboard zugänglichen Einstellungen, wenn man Elementor als Programm definiert sozusagen die **Programmeinstellungen**.
2. Die über das Panel zugänglichen **Website-Einstellungen**, unterteilt in Design System, Theme Stil und Einstellungen.
3. Einstellungen, die (über das Panel zugänglich) die einzelnen Seiten betreffen, also **Seiteneinstellungen**.

2.1. Programmeinstellungen

Dashboard > Elementor > Einstellungen > *Generell*

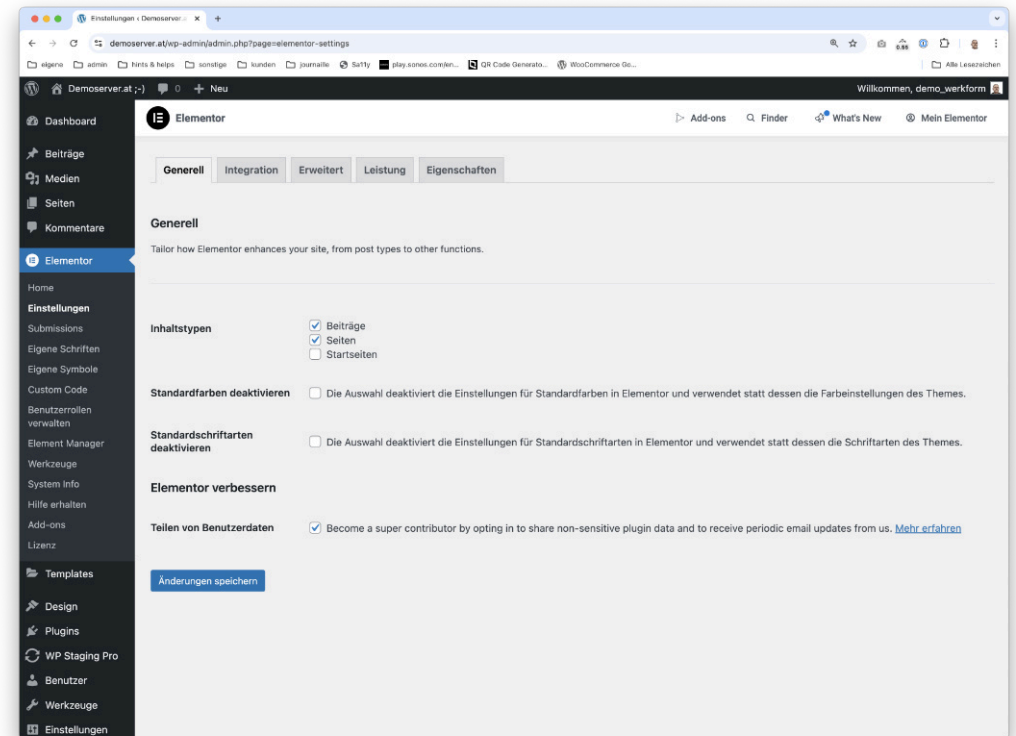
Hier ist auszuwählen, bei welchen Inhaltstypen (Seiten, Beiträge, ...) Elementor angewendet werden soll. Die hier angeführten Startseiten meinen die *Landingpages*, die als Feature unter den *Eigenschaften* > *Laufender Experimente* an/abgewählt werden können.

Darunter sind die beiden wichtigen Checkboxes für **Standardfarben** und **Standardschriftarten deaktivieren**. Diese sind **anzukreuzen**, damit nicht die Voreinstellungen des Themes, sondern jene über das Design System greifen, also zB jene der Typographie (Kapitel 2.2.2).

Unter **Teilen von Benutzerdaten** kann man das Teilen von nicht sensiblen Daten im Hintergrund erlauben.

Dashboard > Elementor > Einstellungen > *Integration*

Services verschiedener Drittanbieter benutzen API-Keys um die Nutzung auf eigenen Webseiten zu ermöglichen. Diese werden hier eingetragen. Die wichtigsten sind wohl *Google Maps Embed API* und *reCaptcha*.



Die generellen Einstellungen sind über das Dashboard zu finden. Man kann Elementor als Page-Builder für verschiedenen Inhaltstypen (Posttypes) ermöglichen oder verhindern.

Willst du deinen Elementor-Coach kennen lernen?

Erstgespräch kostenlos!

Termin hier!

Harry Martin

Unter dem Reiter **Erweitert** kann man die **Methode zum Laden des Editors** verändern, eine Funktion die man nur im Falle von Serverproblemen zur Diagnose benutzen wird.

Wenn man **Ungefilterten Dateien Upload** zulässt verbietet Elementor nicht mehr verschiedene Dateitypen wie .json und .svg auf den Server zu kopieren. Diese Formate benötigt man für Icons oder den Import/Export von Templates. Normalerweise ist das zu aktivieren. Die Warnung über das bestehende Sicherheitsrisiko darunter ist durchaus ernst zu nehmen.

Die Auswahl in **Google Fonts** legt fest, ob eine direkte Verbindung zum Google Font-Server hergestellt werden soll. Diese ist im Regelfall zu deaktivieren, die Einbindung der Schriften erfolgt lokal. (Siehe auch Kapitel 3.3.)

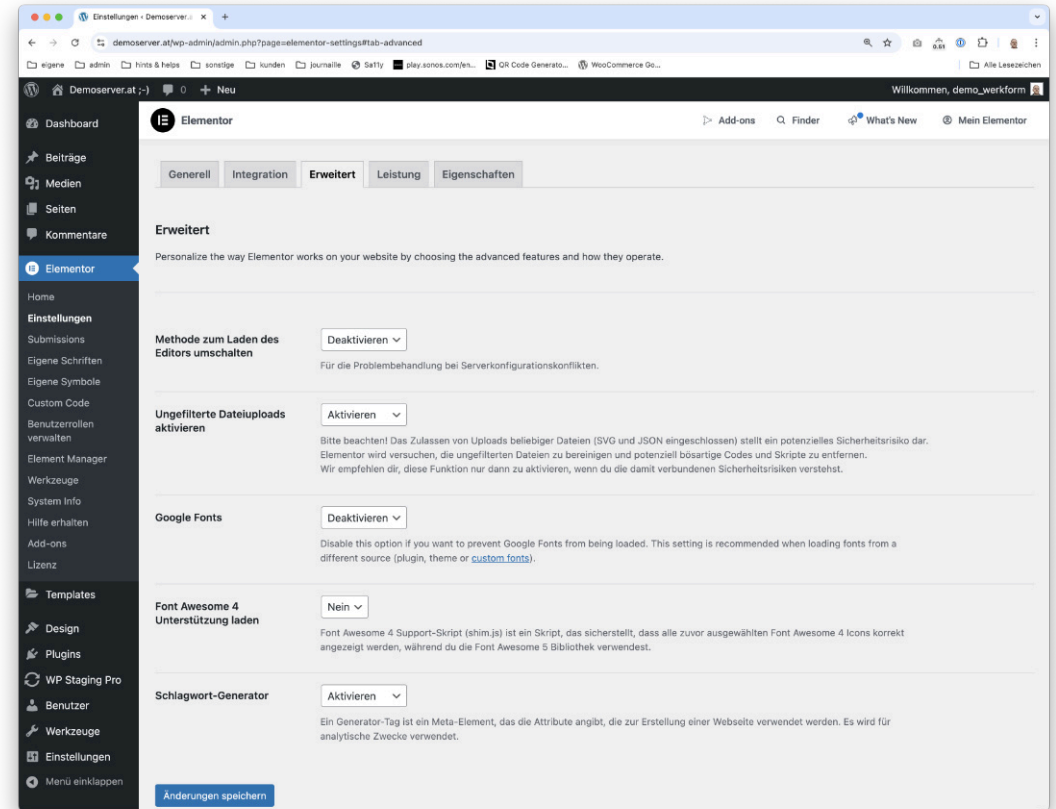
Die Option **Font Awesome 4 Unterstützung** laden sorgt dafür, dass auch ältere (eben in Version 4 von Font Awesome verwendete) Icons richtig geladen und angezeigt werden können. Welche das sind ist am einfachsten über die Font Awesome Homepage zu ermitteln. Bei neuen Projekten nicht aktivieren, das ist veraltet.

Der **Schlagwort-Generator** fügt auf den Seiten im Header einen Tag ein, sollte man deaktivieren.

Unter dem Tab **Leistung** werden nun die Optionen zusammengefasst, die Einfluß auf die Ladezeit der Website haben.

CSS-Ausgabemethode. Legt fest, ob die von Elementor generierten CSS in einer eigenen Datei oder inline platziert werden sollen. Die Voreinstellung (*Externe Datei*) ist gut für eine Produktivseite, während der Entwicklung kann *In-line platziert* die bessere Wahl sein, besonders, wenn man händisch CSS ändert oder Probleme mit dem Browser-Caching erkennt.

Optimized Image Loading. Soll dazu führen, dass das größte Bild einer Website (LCP: Largest Contentful Paint) priorisiert geladen wird Bilder die ausserhalb des sichtbaren Bereichs liegen langsamer geladen werden. Gute Einstellung, aktivieren!



```
<meta name="generator" content="Elementor 3.24.4; features: e_font_icon_svg, additional_custom_breakpoints; settings: css_print_method-external, google_font-disabled, font_display-swap">
```

Das ist ein Beispiel für den Generator-Tag, der von Elementor eingefügt wird. Das Abschalten wird empfohlen, zu einfach sollte man den Bots die Analyse der Website auch nicht machen.

Optimized Gutenberg Loading. Versucht den von WordPress eingestellten aber vmtl. nicht notwendigen Block-Code nicht zu laden. Ok.

Hintergrundbilder nachladen. Soll dazu führen, dass alle Hintergrundbilder mit Ausnahme des ersten erst nach dem anderen Content geladen wird. Das kann gut sein, nicht aber bei Designs, die als Hintergrundbilder (zB. in Loops) die Beitragsbilder verwenden.

Dashboard > Elementor > Einstellungen > Eigenschaften

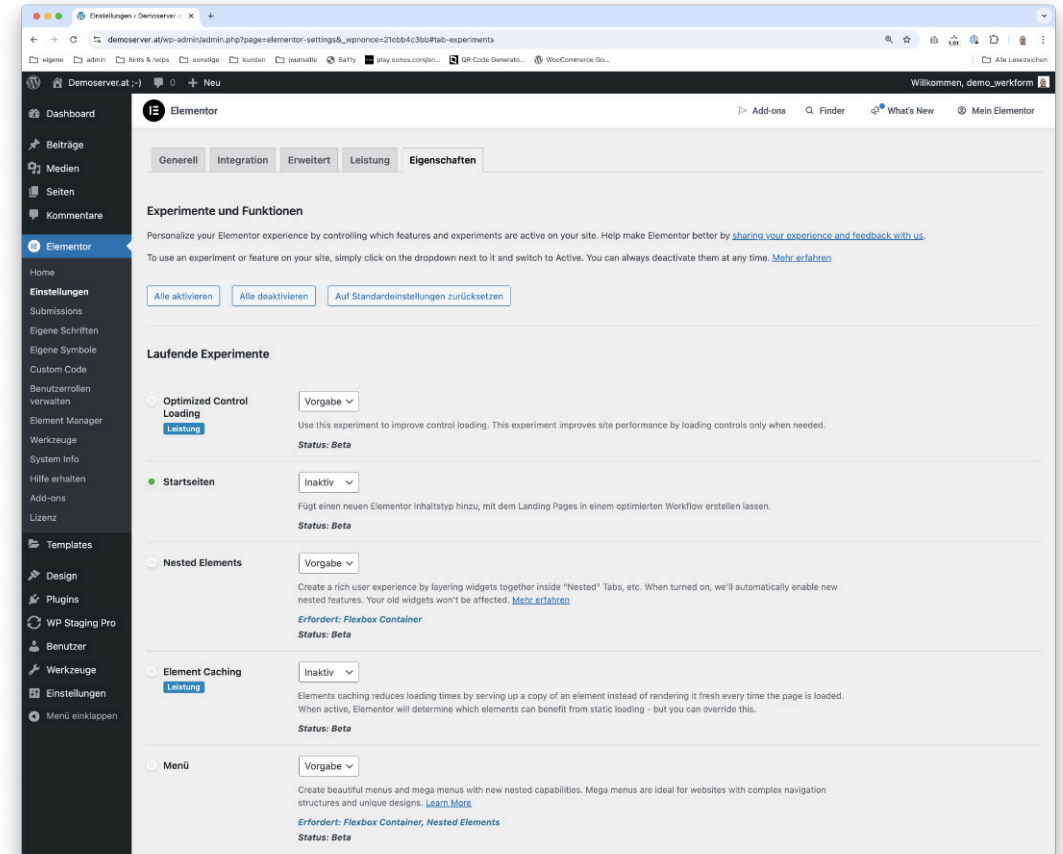
Der **Eigenschaften**-Tab unterteilt zwischen **Laufende Experimente** (oben) und **Stabile Merkmale** (unten). Dazu muss man wissen, das Elementor neue Funktionen nicht einfach mit Updates ausrollt sondern statt dessen jedes neue Feature in so eine einschaltbare Eigenschaft packt, wo sie den klassischen Entwicklungszyklus (Alpha, Beta,...) durchlaufen muss, bevor sie schließlich irgendwann im Elementor default ankommt.

Das ist gut bezüglich der Rückwärts-Kompatibilität, immerhin existieren ja bereits Millionen von Elementor unterstützten Websites. Es kann mitunter auch Jahre dauern, bis Features den Weg durchlaufen. Es ist immer eine gute Idee sich die entsprechenden Release-Notes bei neuen Versionen anzuschauen um zu verstehen, ob ein neues Feature relevant für das eigene Projekt ist.

Entsprechend der hier angewandten Logik kann sich die Liste der Einträge jederzeit ändern (und tut das auch häufig).

Unter **Experimente** sind alle die Optionen zusammengefasst die noch nicht ganz ausgegoren sind aber trotzdem schon im breiten Einsatz angewandt werden können. Bei jeder Option steht ein kurzer erklärender Text der mehr oder weniger deutlich macht, worum es sich handelt. Sehr oft hinkt dieser Text bei der deutschen Übersetzung hinter her, der aktuelle Status (meist beta) wird ebenfalls angeführt.

In diesem Skriptum (V3) wird das Stabile Merkmal **Container** verwendet und auch die Nutzung des **Loops** beschrieben. Auszuschalten sind hingegen die Funktionen *Startseiten* und *Hello Theme Header and Footer*. Weiters wird empfohlen alle Funktionen zu deaktivieren, die nicht verwendet werden (zB:



Die Eigenschaften sind der Platz für das aktivieren und deaktivieren neuer Funktionen.

Additional Custom Breakpoints, Build with AI etc.). Ausserdem ist die Option *Element Caching* während der Entwicklung zu deaktivieren, da es sonst vermutlich zu Problemen bei der Darstellung gerade geänderter Templates kommt. *Editor Top Bar* hingegen wird empfohlen (und wird in diesem Dokument verwendet).

2.2. Website-Einstellungen

Um die Einstellungen innerhalb von Elementor zu erreichen muss ein Dokument (Beitrag, Seite oder Template) geöffnet werden, dann kann man links oben über das Hamburger-Menü das Einstellungs-Menü öffnen. Der oberste Eintrag führt zu den Website-Einstellungen, diese wiederum sind unterteilt in drei Bereiche: *Design-System*, *Theme-Stil* und *Einstellungen*.

Design System

Die *Design System* genannten Einstellungen bieten die Basis des gesamten Designs, hier werden *Globale Farben* und *Globale Schriftart* eingestellt. Dabei handelt es sich immer um Grundeinstellungen, die im Bedarfsfall überschrieben werden können.

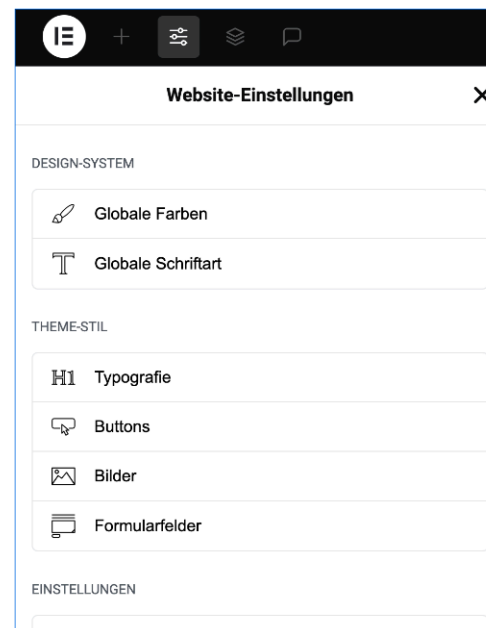
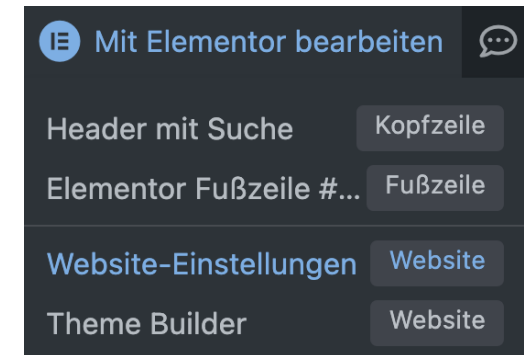
Elementor bereitet vier System-Stile vor, die sowohl Farben als auch Schriftdefinitionen beinhalten:

- Primär
- Sekundäre
- Text
- Accent

Diese 4 Stile können definiert und umbenannt aber nicht gelöscht werden. Zusätzlich können beliebig viele Farben und Font-Stile angelegt werden.



Der Zugang zu den Website Einstellungen gelingt über das Schieberegler-Icon im Top Bar oder über den Link im WordPress-Adminbar.



Die Website-Einstellungen sind in 3 Bereiche **Design-System**, **Theme-Stil** und **Einstellungen** gegliedert.

Globale Farben

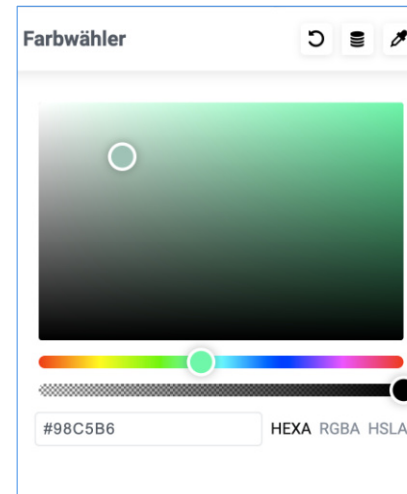
Um eine Farbe zu verändert klickt man auf eine Farbbox und kann so den **Farbwähler** bedienen, der überall zum Einsatz kommt. In der Kopfzeile des Farbwählers sind 2 Funktionen untergebracht: Löschen und der Color Sampler. Die Löschfunktion entfernt die Farbwerte komplett. Die Pipette (Color Sampler) hilft um Farben aus der Umgebung zu entnehmen. Und zwar nicht so wie man es mit diesem Icon gewohnt ist (also indem die Farbe des gewählten Pixels angezeigt wird) sondern indem aus dem Objekt (zB. dem gewählten Bild) eine kleine 4farbige Farbpalette via Algorithmus zusammengestellt wird. Man kann damit aus einem Bild die passenden Farben extrahieren um damit dann Schriften, Hintergründe etc. einzufärben.

Das darunter liegende Farbfeld beginnt links oben bei *Weiß* und endet rechts unten in *Schwarz*. Über den Schieberegler wird die Farbe rechts oben eingestellt. Ein Klick in das Farbfeld zeigt im darunter liegenden Textfeld den Farbwert in HEX an, also zB #98C5B6. Ein zweiter Schieberegler über dem Textfeld dient der Einstellung der Sättigung. Voreingestellt ist immer volle Sättigung, Ein Beispiel mit einem Deckungsgrad von etwa 2/3 bringt dann den Farbwert #98C5B6**99**. Das vierte Zeichenpaar 99 steht hier für den **Alphawert**. Ein Klick neben dem Farbwähler schließt diesen, der zuletzt ausgewählte Wert wird angezeigt, die Farbe im Kästchen daneben angewendet. Die Codierung wird als HEXA bezeichnet.

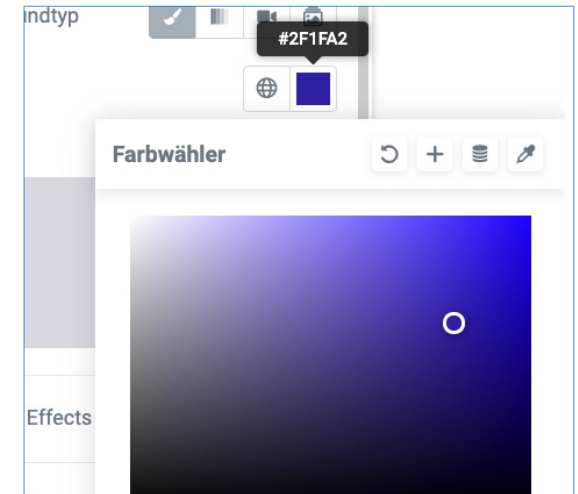
Wer die Schreibweise nicht mag kann alternativ zur HEX-Codierung die Farben auch als RGB(A) und HSL(A) eingeben. Die Schreibweise lautet für den obigen Wert dann *rgba(151, 196, 181, 0.6)* bzw. *hsla(160, 28%, 68%, 0.6)*.

Das Icon für *Dynamische Werte* bedeutet, dass man hier nicht nur fixe Werte vergeben kann sondern auch andere Quellen wie zB. Datenbankfelder nutzen kann, um den Farbwert festzulegen.

Wenn man den Farbwähler bei der Arbeit mit einem Widget anwendet taucht eine weitere Funktionen auf: Wenn eine neue Farbe definiert wird kann man diese über das **+**-Icon zu den Globalen Farben hinzufügen.



Der Farbwähler im Elementor erlaubt die numerische Eingabe oder die Auswahl über Schieberegler. Das Farbmodell ist wählbar.



Innerhalb der Anwendung der Farbwahl in einem Widget kann über das Globus-Icon eine Globale Farbe ausgewählt werden, zusätzlich gibt es ein dynamisches Farbwahlfeld und den Color-Picker.

Website-Heroes
Die WP Business Community

Weiterbildung, Support und Community für WordPress.

Website Heroes lernen laufend dazu und helfen sich gegenseitig. Live via Zoom oder online im Forum.

60+ Aufzeichnungen **111+** 5★-Bewertungen **3.000+** Beiträge im Forum

www.websiteheroes.co

Globale Schriftart

Analog zu den Farben werden im nächsten Schritt die Grundeinstellungen für die Schriften durchgeführt. Ausserdem ist hier die Fallback-Schriftart einzustellen. Also jene Schriftart anzugeben die verwendet werden soll, wenn aus irgend einem Grund die gewählte Schrift (noch) nicht zur Verfügung steht. Die Voreinstellung Sans-serif ist zu meist die richtige Wahl.

Eingestellt werden kann für jede Schriftart die *Schriftfamilie*, die *Größe*, *Schriftschnitt*, *Umwandlung*, *Stil*, *Auszeichnung*, *Zeilenhöhe*, *Zeichenabstand* und *Wortabstand*.

Webseiten funktionieren immer am besten, wenn so wenig Einstellungen wie möglich und so viele wie nötig durchgeführt werden.

Nähere Informationen über die Schriften in Elementor gibt es im Kapitel 3.3 (Lokale Einbindung) und Kapitel 9 (Schriften) dieses Skriptums.

Theme Stil

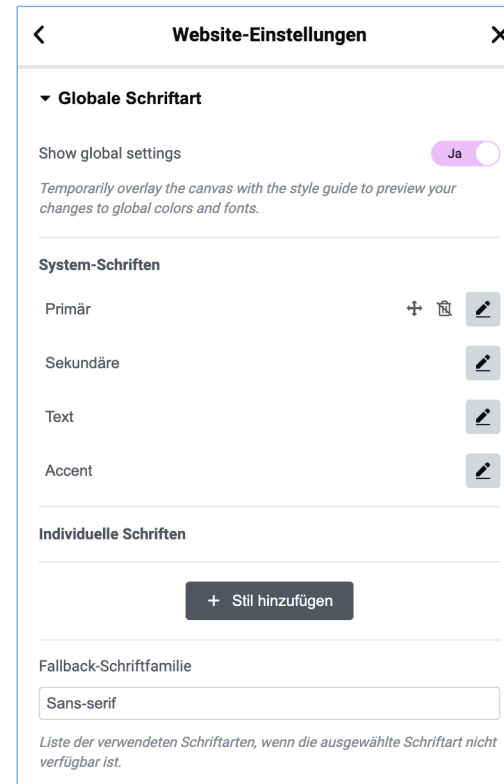
In diesem mittleren der 3 Einstellungsbereiche werden Dinge konfiguriert, die einerseits auf eine globale Einstellung zurückgreifen können, andererseits aber auch die Grundlage für die Gestaltung eines Elements durch ein Widget sein können.

Typographie

Hier ist nun die Stelle, wo man im Elementor die Voreinstellungen für die gesamte Typographie der Webseite vornehmen kann. Und zwar für

- Body
- Link
- Überschriften H1 bis H6

Eingestellt werden kann jeweils die Farbe und die Typographie, für den **Body**



Globale Schriftart ist der Bereich, wo man eigene Schriften komplett festlegen kann - unabhängig vom verwendeten Einsatzzweck und HTML-Tag. Man kann beliebig viele Individuelle Schriften erstellen.

Globale Schriftart vs. Typographie

Wenn Schriften als Globale Schriftart definiert werden kann man sie überall einsetzen, wo man eben typographische Angaben machen kann. Es werden technisch gesehen Klassen erstellt. Die Einstellungen zur Typographie hingegen wirken auf die HTML-Tags der Website. Diese wirken also auch dann, wenn keine andere Einstellung vorgenommen wurde.

(also den gesamten sichtbaren Bereich) zusätzlich ein Absatzabstand (auch responsive) und für **Link** auch die Pseudoklasse *Hover*. Bei Farben und Schriften kann man hier nun auf die in den *Globalen Einstellungen* vorgenommenen Werte zurückgreifen. Ein Beispiel für eine sinnvolle Konfiguration für Links wäre also zB. folgendes:

Definition einer Globalen Farbe mit einem Namen *Linkgrün*

In den Webseiten-Einstellungen > Typographie und **Links** diese Farbe bei *Normal* auswählen

Unter *Links* > *Hover* bei Typographie unter Auszeichnung *Unterstreichen* auswählen

Damit werden: alle verlinkten Texte grün, wenn ich mit dem Cursor darüber fahre wird der Link unterstrichen. Diese Definition enthält absichtlich keine weitere Schriften-Vorgabe. Dh. die Gestaltung der Links funktioniert dann unabhängig von der im jeweiligen Widget angewandten Typographie.

Für die Gestaltung von Überschriften kann man es ähnlich durchführen:

In den Globalen Einstellungen werden die Schriften für Primary & Secondary definiert

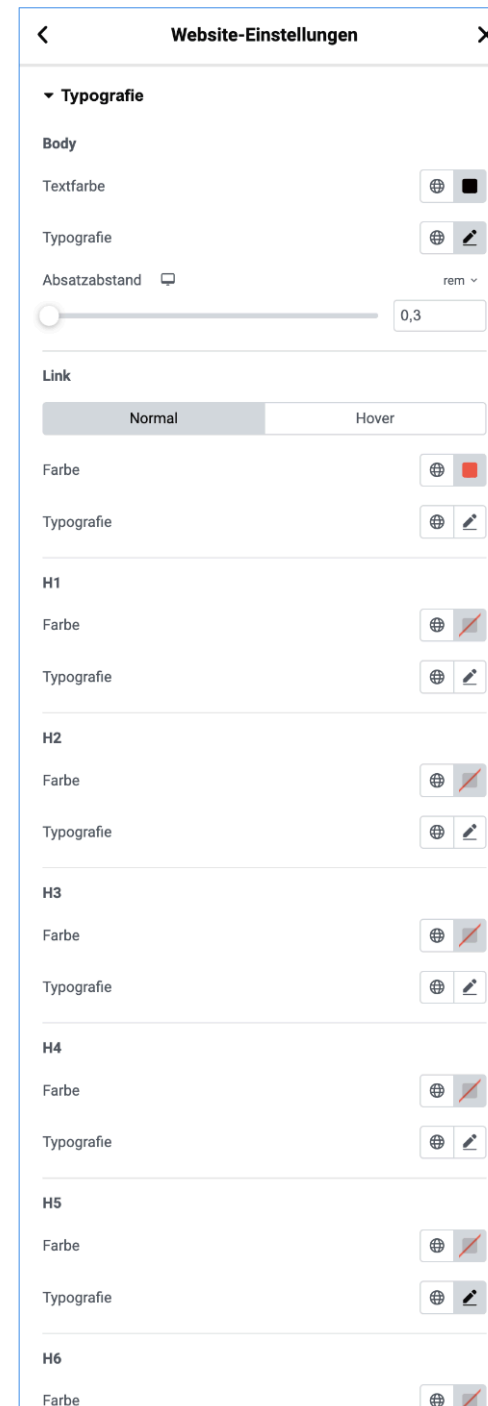
Bei der Überschriften H1 wählt man die Globale Farbe Primary aus

Unter H1 Typographie wählt man das Globale Setting Primary aus

Zusätzlich definiert man dort die Schriftgröße mit 2 REM

Bei H2 bis H6 geht man analog dazu vor, natürlich kann man statt Primary auch andere - selbst definierte globale Einstellungen - auswählen. Die Schriftgrößen werden sinnvoller Weise immer kleiner gewählt je höher die Ordnungszahl ist.

Tipp: Es ist eine gute Idee, die Farben *Schwarz*, *Weiß* und *Transparent* als Globale Farben zu definieren. Praktisch ist es auch, wenn kleine Überschriften (H5, H6) die selbe Größe wie der Text haben. Und: Profis verwenden die Einheiten **REM** und **EM** um ein konsistentes Schriftbild mit passenden Abständen zu erhalten.



Die **Typographie-Einstellungen** wirken auf alle Elemente der Website. An der Darstellung der Icons sieht man übrigens nicht nur die Farbe sondern auch ob ein Wert eingetragen wurde oder nicht. Das Globus-Icon erlaubt eine Auswahl aus den Globalen Farben oder Schriften.

Buttons

Die Voreinstellungen wirken sich neben dem Widget **Button** auch auf andere Widgets aus: auch bei **Formular**, **Anmeldung** und **Call To Action** wirken diese Grundeinstellungen. Einstellen lassen sich hier die globalen Farben/Schriften oder auch eigene Angaben. Dazu kommen Rahmen, Schatten und Abstände (diese auch responsive) sowie die Pseudoklasse `:hover`.

Die Einstellungen für **Schatten** und **Rahmen** tauchen auch bei vielen Widgets auf. Voreinstellungen für Icons gibt es hier nicht. Die Gestaltung von Icons im Button liegen beim Widget Button selber und können nicht in den Website-Einstellungen getroffen werden.

Bilder

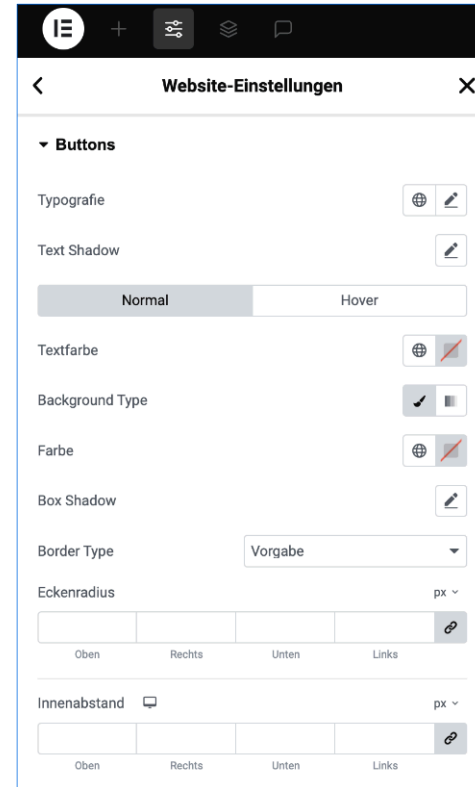
Für Bilder können ebenfalls verschiedene **Rahmenoptionen**, **Schatten** und **Ab-rundungen** voreingestellt werden. Auch gibt es `:hover`. Zusätzlich kann die **Deckkraft** verändert werden und es können **CSS-Filter** (zB. für Farbwerte und Sättigung) angewandt werden.

Erfahrungsgemäß kommen diese Grundeinstellung nur sehr selten zur Anwendung, anders als die Buttons zuvor oder die

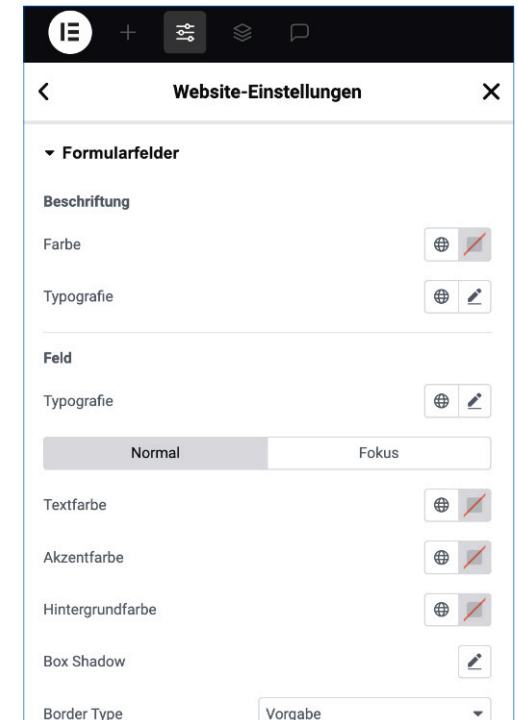
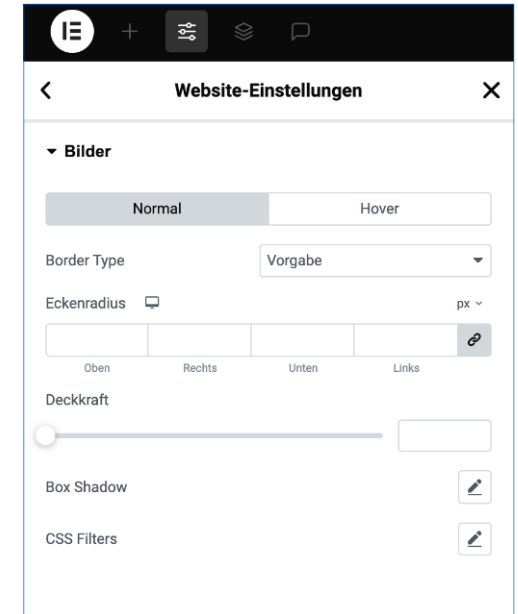
Formularfelder

Bei Feldern, die für Benutzereingaben gedacht sind, muss man besonders Bedacht auf die Endgeräte nehmen. Oft wird man größere Schriften wählen; deutliche Abstände und Rahmen erleichtern die Eingabe. Daher gibt es hier die Pseudoklasse **Fokus**, also die Einstellungsmöglichkeiten für die Gestaltung eines Formularfeldes im Augenblick der aktiven Benutzung.

Die Nutzung dieser Voreinstellung empfiehlt sich besonders, wenn man mehr als ein Formular in der Webseite plant. Wer nur ein Kontaktformular verwendet kann diese Einstellungen genauso gut innerhalb des Formular-Widgets vornehmen.



Die Grundeinstellungen für Buttons, Bilder und Formularfelder sind sinnvoll dann vorzunehmen, wenn man mehr als ein Widget damit konfiguriert. Wer nur ein Formular verwendet kann dieses auch gleich dort gestalten, sobald es mind. 2 sind sollte man die Einstellungen hier verwenden.



Einstellungen

Im dritten Einstellungspunkt werden die Einstellungen zusammengefasst die keinen anderen Platz gefunden haben.

Website-Informationen

Wer bisher über den WordPress Customizer sein Logo und das Favicon definiert hat kann jetzt auf diesen Weg verzichten. Hier können genau 4 wichtige Dinge eingestellt werden:

- Titel der Website (Site Name)
- Untertitel (Site Description)
- Logo (Site Logo)
- Website Icon (Site Favicon)

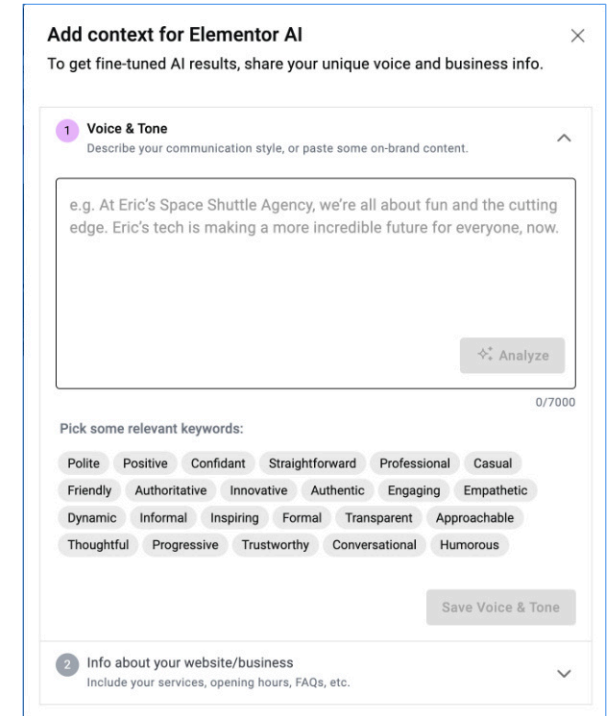
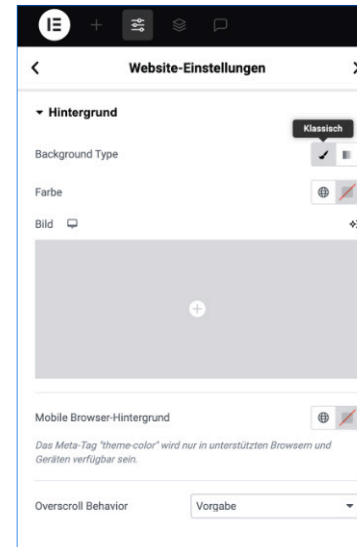
Titel und Untertitel können natürlich auch wie bisher über die WordPress eigenen Einstellungen geändert werden. Das Website Icon sollte quadratisch sein, Elementor empfiehlt 512 x 512 Pixel. Daraus werden dann die verschiedenen Icons erstellt die in der Browserhistory, in Bookmarks, Feeds etc. auftauchen.

AI Context

Die Idee dieser Definition ist, dass man eine Kontext-Umgebung der Website definiert, auf die dann die Elementor AI zurückgreift, wenn sie Bilder oder Texte erstellt. Ausserdem kann man hier Öffnungszeiten und ähnliche Informationen angeben, die dann ggf. für generierte Inhalte verwendet werden.

Hintergrund

Voreinstellungen für den Hintergrund sind hier für die gesamte Webseite möglich. Neben Farben und Hintergrundbildern kann man auch lineare oder radiale Verläufe einstellen, die auch gedreht werden können. Gestalterische Vorsicht ist hier besonders gefragt.



Während Website-Informationen und Hintergrund ein Bearbeitungspanel links öffnen erscheint beim AI Context ein Modal-Fenster.

Diese Website Hintergrund-Einstellungen lassen Farben und Bild zu, während die Hintergrund-Einstellungen für Container auch Videos und Slideshows erlauben.

Layout

Sehr grundsätzliche Voreinstellungen mit besonders großer Auswirkung werden im Punkt Layout getroffen. Die **Inhaltsbreite** legt fest, wie breit der Inhalt sein soll, sobald das Browserfenster breiter als der hier festgelegte Wert ist. Immer dann, wenn bei der Breite *Boxed* eingestellt wird (zB. bei einem Container) kommt dieser Wert zum Tragen. Das Gegenstück dazu heißt *Volle Breite*.

Der **Container-Innenabstand** hat die Voreinstellung 10 px, also den halben Widget-Abstand. Damit entstehen - so fern diese Werte nicht verändert werden, gleichmäßige Designs, weil alle Abstände zueinander ausgewogen sind. Hier sind auch Angaben in relativen Einheiten möglich,

Der Wert beim Punkt **Lücken** (Standard-Abstand zwischen den Widgets) hat extreme Auswirkungen auf die Gestaltung. Die Vorgabe von 20 px ist gut gewählt, Änderungen sollte man nur machen wenn man genau weiß warum. Die Werte können für Zeilen- und Spaltenabstand gleich oder getrennt eingegeben werden.

Der **Seitentitel-Selektor** wird verwendet, falls der Titel ausgeblendet werden soll.

Die Option **Streckt den Abschnitt auf die passende Größe** bleibt Spezialisten vorbehalten, beide Werte einfach belassen ist die Empfehlung.

Praktischer ist der Wert der bei **Standard-Seitenlayout** eingetragen wird. Theme verwendet wie zu erwarten die Voreinstellung des Themes. Für die Arbeit mit Elementor sind die beiden anderen Werte spannend: *Elementor Canvas* bedeutet, dass Elementor eine leere Leinwand (also keine Header/Footer) anwendet. *Elementor Volle Breite* wiederum blendet (je nach gewählten Bedingungen) Header und Footer ein. Letzteres ist die wohl häufigste Anwendung und daher empfohlen.

Der darunter liegende Tab **Breakpoints** ist der Platz wo die Grenzen für das Arbeiten in und mit den Responsive Modi beginnt. Der Ausgangsmodus im Webdesign heißt *Desktop*, dieser existiert immer, er benötigt keine weiteren Definitionen. **Alles, was nichts anderes ist, ist Desktop.** Dazu kommen 2 Modi

Website-Einstellungen

▼ **Layout-Einstellungen**

Inhaltsbreite px

Setzt die Standard-Breite des Inhaltsbereichs (Standard: 1140px)

Container-Innenabstand px

Oben Rechts Unten Links

Sets the default space inside the container (Default is 10px)

Lücken px

Spalte Zeile

Definiert die Vorgabe für den Abstand zwischen Widgets (Standard-Vorgabe: 20px)

Seitentitel Selektor

Elementor kann den Seitentitel ausblenden. Das funktioniert bei allen Themes, die den „h1.entry-title“-Selektor verwenden. Sollte dein Theme einen anderen Selektor verwenden, kannst du ihn hier eintragen.

Streckt den Abschnitt auf die passende Größe

Gib den Selektor für das Eltern-Element ein, welches die seitlich ausgedehnten Abschnitte ausfüllen sollen (z.B. #primary / .wrapper / main usw.). Freilassen, um die ganze Breite der Seite auszufüllen.

Standard-Seitenlayout

Die Standardseitenvorlage, wie sie in Elementor Panel > Hamburger Menu > Website Einstellungen definiert ist.

Änderungen werden erst in der Vorschau angezeigt, nachdem die Seite neu geladen wurde.

► **Breakpoints**

Die Grundeinstellungen zum Layout in der Einheit px (Pixel) durchzuführen ergibt auch Sinn, wenn man sonst REM verwendet. Pixel ist die einzige absolute Einheit, die man im Web hat.

Daher lassen sich bei den Breakpoints auch keine anderen Einheiten angeben. Ob man wirklich zusätzliche Breakpoints aktiviert hängt sehr stark von der Zielgruppe des Projekts ab. Praktisch sehr selten.

▼ **Breakpoints**

Aktive Breakpoints

Mobile Portrait Tablet Portrait +

Die Optionen Mobile und Tablet können nicht gelöscht werden.

Mobile Portrait

Breakpoint (px)

Tablet Portrait

Breakpoint (px)

die auch immer da sind, auch wenn man keine besonderen Einstellungen trifft: *Tablet Portrait* und *Mobil Portrait*. Immer wenn man den Breakpoint unterschreitet wirken die Einstellungen, die zu den jeweiligen Modi getroffen wurde. Die Voreinstellungen von 767 px (Mobil) und 1024 px (Tablet) passen wohl fast immer.

Dazu kommen optionale Modi für 4 weitere Breakpoints. Werden diese hier aktiviert kann immer dann, wenn responsive Anpassungen möglich sind, diese auch für die zusätzlichen Modi vornehmen. Diese sog. Additional Breakpoints sind:

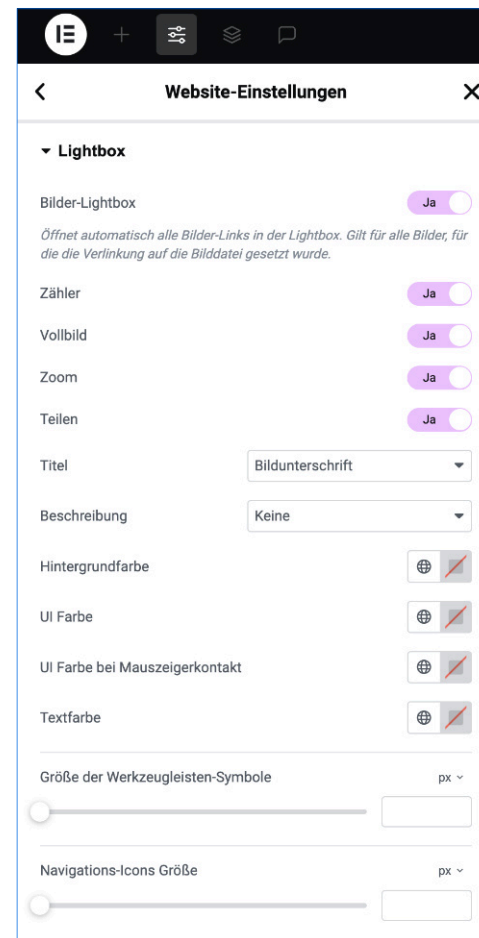
- Mobile Landscape (880 px)
- Tablet Landscape (1200 px)
- Laptop (1366 px)
- Breitbild (2400 px)

Die Werte sind veränderbar und zielen wie schon die Namen vermuten lassen auf genauere Designanpassungen auf verschiedene Gerätetypen ab. Ob sich der Mehraufwand in der Gestaltung wirklich lohnt muss natürlich genau abgewogen werden.

Lightbox

Wenn ein Link bei einem Bild *auf die Bilddatei* gesetzt wird verwendet Elementor ein Lightbox-Skript. Diese Funktion kann in dieser Voreinstellung grundsätzlich ausgeschaltet werden. Bei aktiver Lightbox lassen sich einige Optionen einstellen: *Zähler*, *Vollbild*, *Zoom* und *Teilen* können einzeln deaktiviert werden.

Weiters kann man wählen, welche Medieninformation als *Titel* und welche als *Beschreibung* angezeigt werden soll. Ausserdem lassen sich die Farben für den *Hintergrund*, die *Textfarbe* und die *Bedienungselemente* einstellen, ebenso die Symbolgröße der Werkzeugleiste sowie die Icon-Größe der Navigationselemente.



Die grundsätzlich vorhandene Lightbox-Funktionalität kann generell abgeschaltet werden. Wer sie verwendet kann ihr Aussehen hier genau konfigurieren.

Seitenübergänge

Sobald ein User von einer Seite auf eine andere innerhalb der Website wechselt können die sog. **Page-Transitions** aktiviert werden. Diese gelten für die gesamte Website, man kann (im Augenblick) diese Übergänge nur generell ein- oder ausschalten.

Die Idee dieser Funktion ist es, dass Look&Feel einer Webseite zu verbessern indem zwischen den Seitenaufrufen zum Beispiel ein animierter Übergang erfolgt oder ein Hintergrund angezeigt wird. Mittels **Preloader** kann so auch die gefühlte Ladezeit verkürzt werden.

Individuelles CSS

Es gibt mehrere Möglichkeiten eigenes CSS in Elementor unterzubringen. Hier ist jene für CSS-Teile, die wirklich immer verfügbar sein sollen. Mehr über CSS in Kapitel 10.

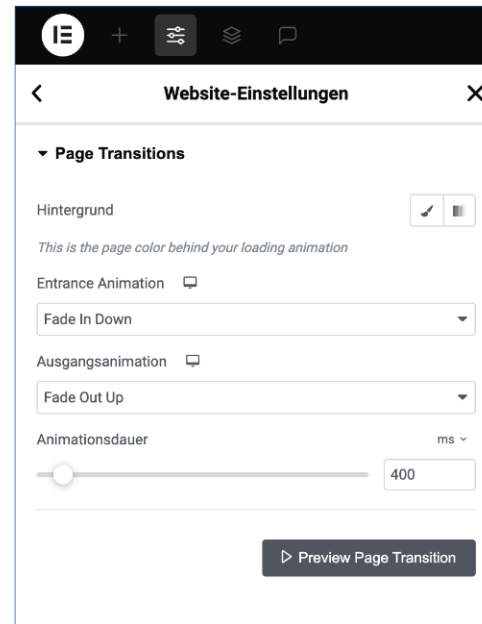
Zusätzliche Einstellungen

Diese Option ist lediglich ein Link zu den in Kapitel 2.1. beschriebene Optionen.

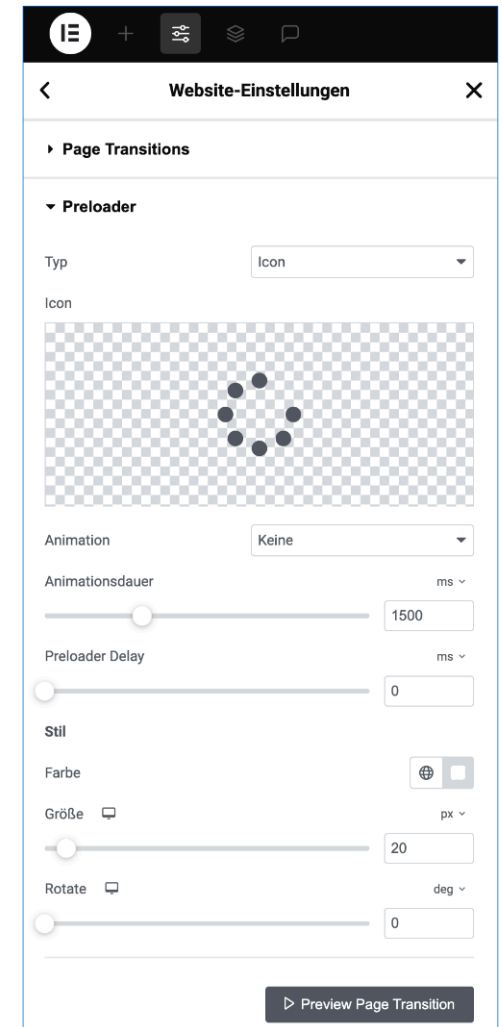
2.3. Benutzer-Einstellungen

Die Benutzereinstellungen sind über das Elementor-Icon im Panel zu erreichen und bieten nur wenige, aber doch nützliche Optionen. Diese wurden in 4 Bereiche gegliedert (wie so oft hinken die DE-Übersetzungen hinterher).

Panel > (Elementor Icon) > Benutzer-Einstellungen



Seitenübergänge dienen dazu, die subjektive Ladezeit zu verkürzen. Sie können nur für alle Übergänge einer Website konfiguriert werden.



Panel

Oberfläche-Theme legt fest, ob die Elementor-Oberfläche hell, dunkel oder automatisch vom System bestimmt werden soll.

Breite macht genau das, die Panel-Breite. Diese kann auch jederzeit über den kleinen Anfasser in der Mitte des Panels verschoben werden. Ausserdem kann das Panel in der vertikalen Mitte eingeklappt werden.

Canvas

Bearbeitungs-Icons (Quick Edit Options). Damit werden bei Mauszeigerkontakt die Funktionen Duplizieren und Löschen angeboten.

Lightbox im Editor aktivieren - auch selbsterklärend, aber gut zu wissen wo man das ausschalten kann. Manchmal sind Lightboxes während der Arbeit ausgesprochen lästig.

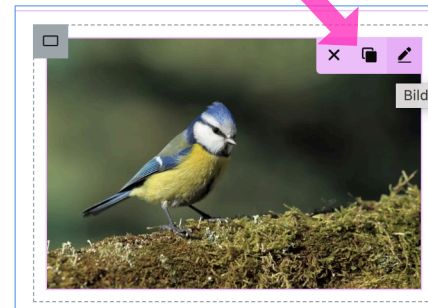
Verborgene Elemente - legt fest, ob im Editor die im Frontend ausgeblendeten Elemente angezeigt werden sollen. Besonders hilfreich, wenn man das Design für kleinere Bildschirme überprüfen will und dies durch graue (weil ausgeblendete) Elemente erschwert wird.

Design System

Show Global Settings. Dies ist eine generierte Seite, die alle definierten Globalen Farben und Schriften einblendet. Sehr hilfreich beim Eingeben und Kontrollieren der Grundeinstellungen eines Projektes.

Navigation

Gehe zu - diese Funktion erlaubt es auszuwählen, wohin man gelangt sobald Elementor beendet wird. Zur Auswahl stehen *Dieser Beitrag*, *Alle Beiträge* oder *WP-Dashboard*.



Die beiden Icons für Löschen und Duplizieren erscheinen erst wenn man über dem Bearbeiten-Icon ist. Kann hilfreich sein.

”

Als Lehrender beherrscht Harry die Kunst, sein immenses Fachwissen leicht verständlich, kurzweilig und vor allem einprägsam zu vermitteln. Darüber hinaus brilliert er als Consulter mit trefflichen Lösungen, selbst bei komplizierten und nichtalltäglichen Problemstellungen.



Goran Miletić
Journalist und Content Creator, Wien

werk form

Erstgespräch buchen



Anzeige

Benutzerrollen verwalten

Elementor erlaubt es festzulegen, welche Benutzerrolle welche Elementor-spezifischen Rechte erhält:

Dashboard > Elementor > Benutzerrollen verwalten

Damit ist es zB. möglich, Redakteuren den Zugriff auf die Gestaltung mit Elementor zu erlauben, den Autoren diese aber vorzuenthalten. Das funktioniert auch mit selbst definierten Benutzerrollen.

2.4. Element Manager

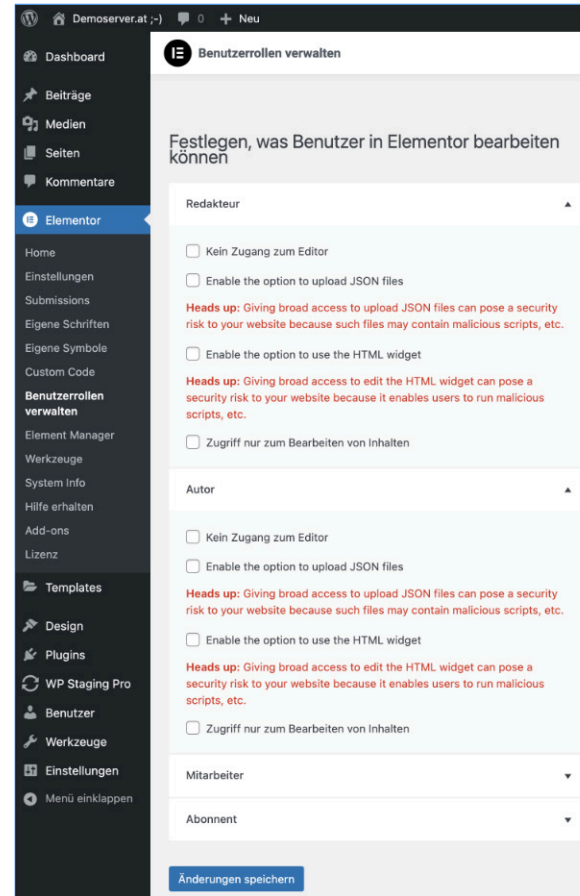
Es gibt viele Widgets, wirklich viele. Alleine die Pro-Version bietet 86 weitere Widgets mehr als die kostenlose Version. Wirklich niemand benötigt alle. Und irgendwann hat auch die Entwicklungsabteilung bei Elementor begriffen, dass es besser sein kann weniger zu haben.

Dashboard > Elementor > Element Manager

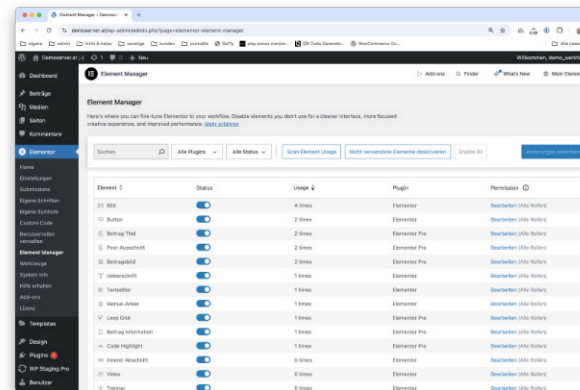
Bringt eine Einstellungsseite, die im ersten Schritt eine sehr wertvolle Funktion bringt: *Scan Element Usage*. Diese Funktion durchsucht das gesamte Projekt und listet neben jedem Widget auf, wie oft es verwendet wurde. Damit ist es sehr einfach zu erkennen, was man nie verwendet!

Im nächsten Schritt kann man einfach jene Widgets deaktivieren, die man nicht benötigt. Sie stehen dann im Backend nicht mehr zur Verfügung - und natürlich auch nicht im Frontend.

Falls man jedoch ein verwendetes Widget deaktiviert, kann das zum Zerfall des Designs führen. Es ist daher ratsam vorerst jene Widgets zu deaktivieren, die man sicher nicht verwenden wird und erst nach Ende der Designarbeit alle anderen, die nicht verwendet wurden. Das spart Ressourcen und ist ein sehr einfaches Mittel, um eine Elementor-Site zu beschleunigen.



Wer darf mit Elementor was machen? Über die Benutzerrolle können die Möglichkeiten limitiert und damit die Website optisch und technisch abgesichert werden.



Der Element Manager hilft beim Sparen: nur notwendige Widgets sollten auf einer Produktivseite aktiviert sein.

2.5. Seite Einstellungen

Das Zahnrad-Symbol in der Mitte des schwarzen Top-Bars führt zu den Seiten-Einstellungen - auch dann, wenn es um Beiträge, Templates und Popups geht, nicht nur bei Seiten. Dementsprechend können sich die einstellbaren Optionen je nach Typ des Dokuments voneinander unterscheiden.

So haben die Einstellungen bei **Templates** eine *Vorschau-Option* um Beispiel-Daten für das Dokument einzubinden. Beim **Popup** gibt es Optionen zur *Größe* und *Position*. An dieser Stelle werden nur die Einstellungen für Seiten/Beiträge besprochen.

Die Einstellungen sind in die 3 Tabs *Einstellungen* / *Stil* / *Erweitert* gegliedert.

Einstellungen

Hier findet man Optionen, die auch über das Elementor-Dashboard bzw. über die bearbeiten-Funktion innerhalb von WordPress eingestellt werden können.

Titel. Der Titel des Dokuments, grundsätzlich unabhängig davon ob er angezeigt oder auch (zB. durch ein SEO-Plugin) nach aussen überschrieben wird.

Status. Neben *Entwurf* und *Veröffentlicht* sind hier auch *Ausstehende Überprüfung* und *Privat* einstellbar. Der notwendige Rest für diese Optionen ist über das WP-Dashboard einzustellen.

Textauszug. Das Textfeld für einen manuellen Auszug.

Beitragsbild. Das *Featured Image*, das auch bei Seiten eingestellt werden sollte.

Sortierung. Im Deutschen meist *Menü-Reihenfolge* genanntes Ordnungsfeld.

Kommentare erlauben. Erlaubt die Kommentarfunktion zum einzelnen Beitrag einzustellen.

The screenshot shows the 'Seite Einstellungen' (Page Settings) interface in Elementor. The top navigation bar includes icons for home, add, settings, and chat. The main title is 'Seite Einstellungen'. Below the title are three tabs: 'Einstellungen' (selected), 'Stil', and 'Erweitert'. The 'Allgemeine Einstellungen' (General Settings) section is expanded, showing the following options:

- Titel:** A text input field containing 'Elementor Beispiel'.
- Status:** A dropdown menu set to 'Veröffentlicht'.
- Textauszug:** A large text area for a manual excerpt.
- Beitragsbild:** A placeholder for a featured image with a plus sign.
- Sortierung:** A numeric input field set to '0'.
- Kommentare erlauben:** A toggle switch set to 'Nein'.
- Titel ausblenden:** A toggle switch set to 'Nein'.
- Seitenlayout:** A dropdown menu set to 'Elementor Full Width'.

At the bottom, there is a note: 'Dieses Template enthält den Header, Inhalt in voller Breite und den Footer'.

Einstellungen zur Seite sind teilweise auch über das WP Dashboard zugänglich. Die hier als Seitenlayout bezeichnete Option meint das Template, das WP verwendet.

Titel ausblenden. Blendet den Titel innerhalb einer mit Elementor erstellten Seite aus, auch wenn das verwendete Template diesen anzeigen würde.

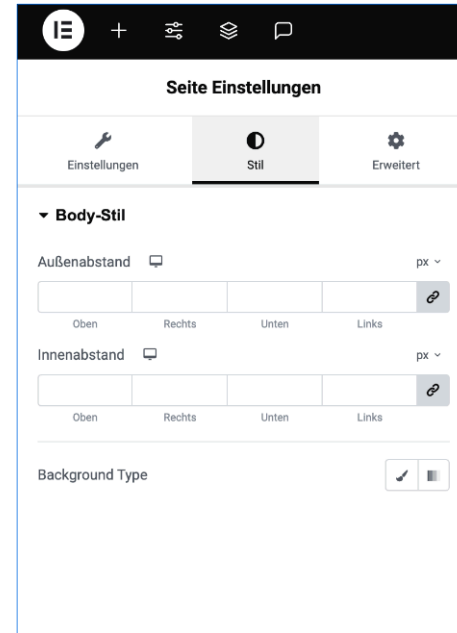
Seitenlayout. Hier wird das Template eingestellt, das die Basis der Darstellung der Seite ist. Mit Elementor werden 2 Templates installiert: *Elementor Canvas* und *Elementor Volle Breite*. *Canvas* bedeutet leere Leinwand, es wird keine Header oder Footer verwendet. *Volle Breite* benutzt die Einstellungen, die bei den Bedingungen für die Theme-Builder-Templates gesetzt wurden um Header, Footer etc. anzuzeigen. Im Regelfall ist das die richtige Einstellung.

Stil

Im Stil für eine Seite können nur Aussen- und Innenabstände und ein Hintergrundtyp eingestellt werden.

Erweitert

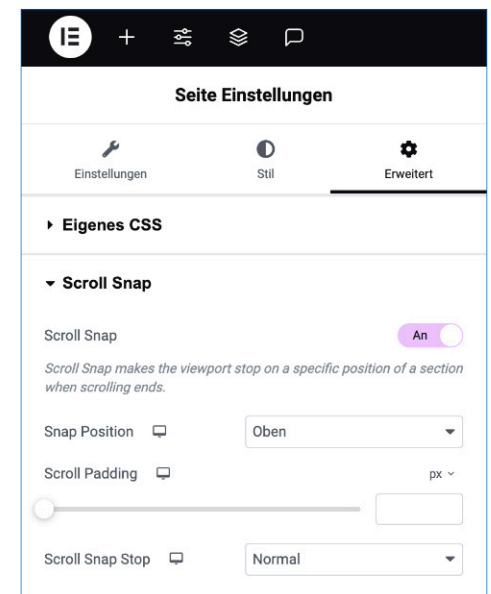
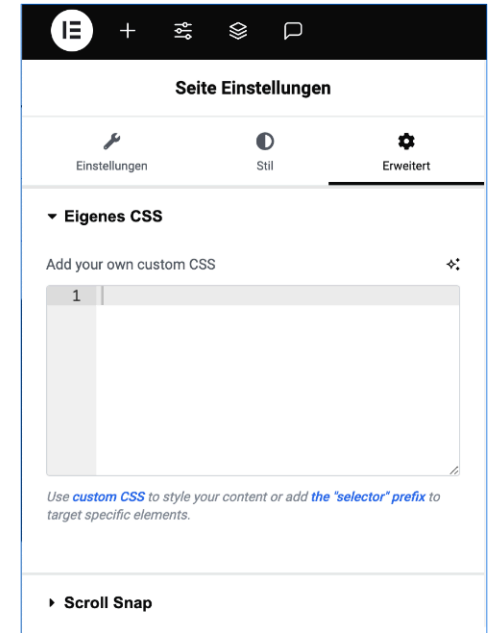
Hier ist der Platz um Seitenspezifisches **CSS** unterzubringen. Ausserdem sind hier die Einstellungen zu den **Scroll-Snap**-Optionen möglich.



Außen- und Innenabstände können hier Seitenweise definiert werden, ebenso der Hintergrund als Farbe, Bild oder Verlauf.

Falls CSS auf der Seite benötigt wird ist hier der beste Platz dafür: technisch funktioniert es natürlich auch, wenn man das CSS irgendwo bei einem Widget einbindet - doch findet man das dann jemals wieder?

Die Scroll Snap Einstellungen sollen das Scrollverhalten zugunsten der Ergonomie verbessern: Container rasten an bestimmten Positionen ein.



3. Elementor und WordPress

Als **Plugin** ist Elementor gut in WordPress eingebettet. An verschiedenen Stellen treffen hier die Notwendigkeiten des Site-Building mit den Funktionen, die ein CMS können muss, aufeinander. In diesem Kapitel geht es um die Schnittstellen zwischen den Bereichen WordPress, Elementor und dem Rest der Welt.

3.1. Import und Export mit Site Kit

Um verschiedene Bestandteile einer Webseite in einer anderen nutzen zu können bedarf es **Import- und Export**funktionen. Neben dem Arbeiten mit Templates funktioniert das auch über das sog. **Site Kit**. Damit ist ein gemeinsames Archiv von Seiten, Beiträgen, Templates und auch Inhalten und den damit verbundenen Einstellungen gemeint.

Über die **Elementor-Bibliothek** werden viele solcher Kits angeboten. Zu finden sind diese unter

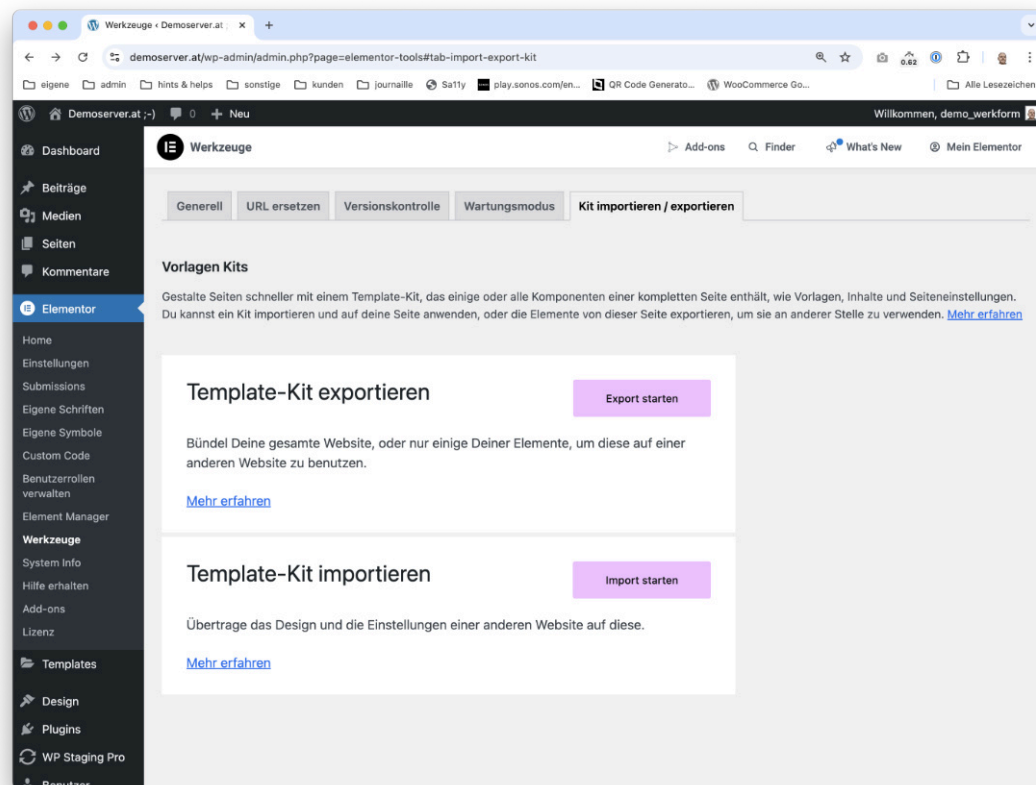
Dashboard > Templates > Kit Bibliothek

Weiters gibt es auch ein großes Angebot an kostenlosen und kostenpflichtigen Kits im Netz zu finden. Für Webdesigner wichtiger ist es aber die eigenen Arbeiten und Einstellungen von einem Projekt zum nächsten zu transportieren. Oder auch eine eigene Sammlung von Templates zu erstellen, die dann bei Bedarf in Projekten angewandt werden können.

Mit den Site-Kits können auch Inhalte von einer Website zur nächsten kopiert werden. Auch kann man das *zuletzt importierte* Site-Kit (und damit auch die Inhalte) wieder entfernen.

Aufgerufen werden diese Funktionen über

Dashboard > Elementor > Werkzeuge > Kit importieren/exportieren > Export starten oder Import starten



Der Prozess von Kit Im- oder Export ist nicht besonders anspruchsvoll. Aufpassen muss man nur, dass man nicht zu viele Daten importiert - nur der letzte Import kann rückgängig gemacht werden.

Danach wird ausgewählt welche (voreingestellt sind alle) Komponenten der Webseite exportiert werden sollen. Die Vorlagen umfassen alle Templates des Theme Builders, zusätzlich Popups und Global Widgets. Der Inhalt umfasst sowohl mit Elementor als auch im WordPress erstellte Seiten und Beiträge, auch Custom Post Types. Mit den Website-Einstellungen werden jene Werte exportiert, die man über

Panel > Website-Einstellungen

getroffen hat. Zusätzlich lassen sich ein Name für das Kit und eine Beschreibung dafür angeben. Nach erfolgreichem Download findet man eine Datei namens *elementor-kit.zip* im Download Verzeichnis. Diese enthält verschiedene .json und .xml-Dateien.

Der Import funktioniert genau gegengleich, man kann auch hier auswählen welche Teile des Kits importiert werden sollen. Diese Funktionen sind nicht geeignet um ein Backup anzulegen oder wieder einzuspielen, dazu sind andere Plugins notwendig!

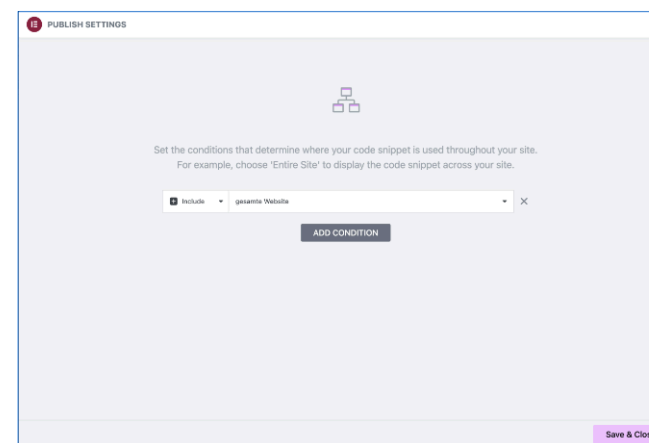
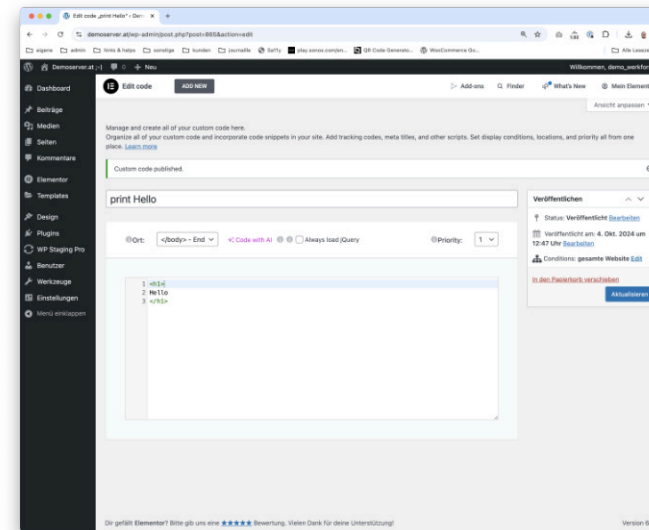
3.2. Custom Code

Wer eigenen Code auf einer Webseite einbauen will muss kein Child-Theme mehr bemühen oder die functions.php-Datei bearbeiten. Unter

Dashboard > Elementor > Custom Code

Kann man eigenen Code (zB. den von Google Analytics, Facebook oder anderen Plattformen und natürlich auch selber geschriebenen Code) zur Webseite hinzufügen. Dazu erstellt man einen neuen Custom Code, vergibt einen sinnvollen Namen und kann den Code in einem Textfenster eingeben.

Zu wählen ist, ob dieser Code im Header der Webseite, am Beginn oder am Ende des Bodys eingefügt werden soll. Zusätzlich kann man die einzelnen Snippets noch mit Ordnungszahlen von 1-10 ausstatten und dadurch die Reihenfolge der verschiedenen Codes bestimmen. Beim Speichern wird man gefragt, wo der Code eingebunden werden soll.



Eigener (HTML, Script) Code kann innerhalb des Elementor-Tools erstellt und bearbeitet werden. Angezeigt wird der Code über die bei Elementor üblichen Bedingungen-Einstellungen.

Diese Anzeigebedingungen kennen wir auch vom Theme Builder. Nicht eingegeben werden kann PHP-Code. Dafür benötigt man ein Plugin wie zB. Code Snippets.

3.3. Schriften Lokal einbinden

Von Haus aus bindet Elementor die Schriften von Google Fonts ein. In Europa (und auch sonst) ist es eine gute Idee, diese automatische Einbindung zu verhindern:

Dashboard > Elementor > Einstellungen > Erweitert > Google Fonts

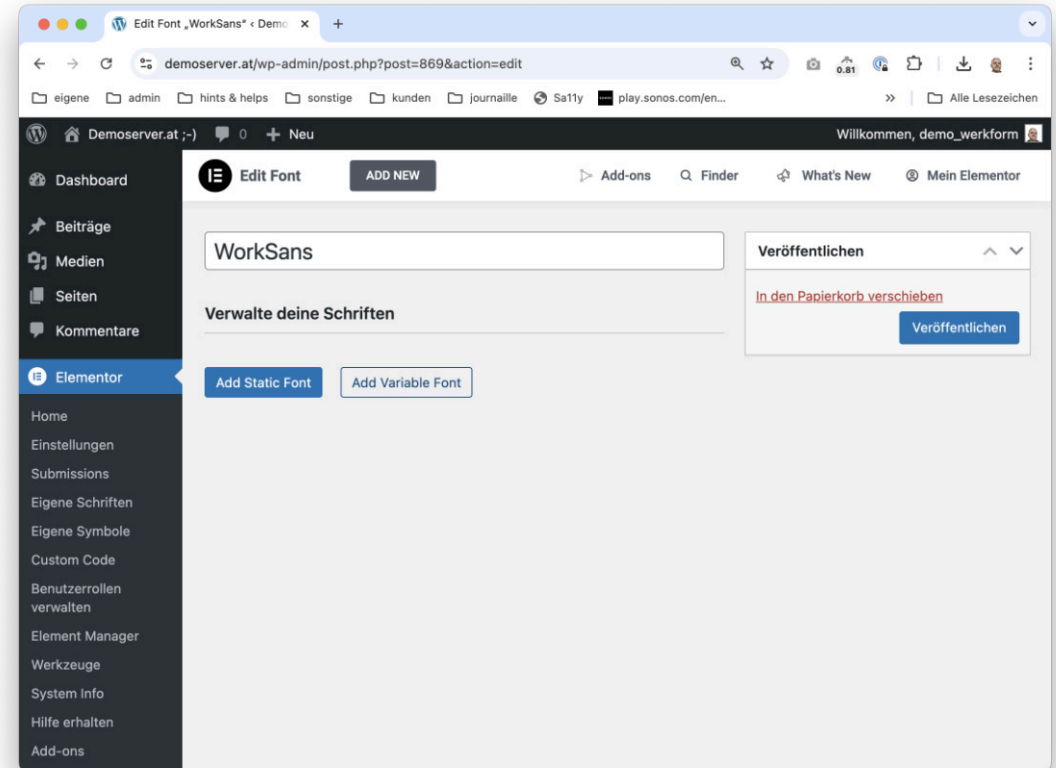
Das bedeutet aber nicht, dass man auf die Vielfalt der Google-Fonts verzichten muss. Google erlaubt die lokale Einbindung, der Download ist direkt über die Google Font-Seite oder auch über spezielle Service-Seiten möglich. Wenn man die entsprechenden Font-Dateien zur Verfügung hat gibt es unter

Dashboard > Elementor > Eigene Schriften

den Einstellungsbereich dafür. Elementor erlaubt 5 verschiedene technische Formate. Je nachdem, wie sicher man gehen will um auch ältere und exotischere Browser zu unterstützen, sind diese Files einzubinden. Die Formate sind:

- woff2:** Web Open Font Format 2 – die kleinsten und modernsten Font Files
- woff:** Web Open Font Format – W3C Standard seit 2012
- svg:** Scaleable Vector Graphics, große Files, besonders für iPhone und iPad
- ttf:** True Type Fonts – Schrift Format aus den 1980ern
- eot:** Embedded Open Type – uraltes Microsoft Format

Im Regelfall wird es ausreichen, nur das Format **woff2** zu verwenden und sonst auf den Fallback Font (siehe Einstellungen) zu vertrauen. Wer ältere Schriftformate benötigt wird vermutlich auch viele CSS-Features nicht nutzen können, die mit der Arbeit mit Elementor erstellt werden. Seit der Version 3.24 unterstützt Elementor auch **Variable Fonts**. Wobei man sich hier nicht zu viel



Vor dem Upload nach dem Anlegen einer neuer Schrift muss man wählen, ob es sich um *Statische Schriften* oder einen *Variablen Font* handelt. Während mit dem Upload eines statischen Fonts alles erledigt ist, muss man beim Variablen Font wissen, welche Achse unterstützt wird und dessen Grenzwerte auch angeben.

versprechen sollte: unterstützt werden nur die Achsen *Width* und *Weight*, also die Schriftstärke und die Buchstabenbreite.

Häufiger ist (noch) der Einsatz statischer Schriften. Man legt eine neue Schriftart an und vergibt den richtigen Namen. Danach kann man die beschriebenen Files uploaden. Die verschiedenen **Schnitte** werden in Elementor als *Schriftart Variante* bezeichnet. Für jeden Schriftschnitt (Variante) ist anzugeben, um welche Schriftstärke und welchen Stil es sich handelt.

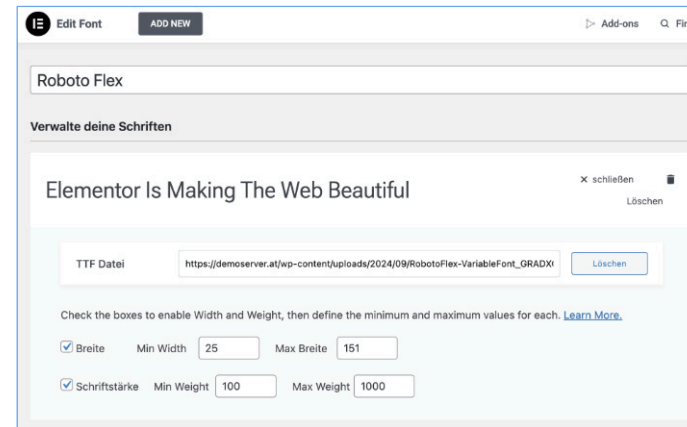
Mit dem Schriftschnitt legt man fest, wie **fett** die Buchstaben sein sollen, von 100 (ganz dünn) bis 900 (ganz dick), oder aber auch Normal oder Bold. Was ist jetzt aber der Unterschied zwischen Normal und 400? Oder Bold und 700?

Einfach zusammengefasst kann man sagen:

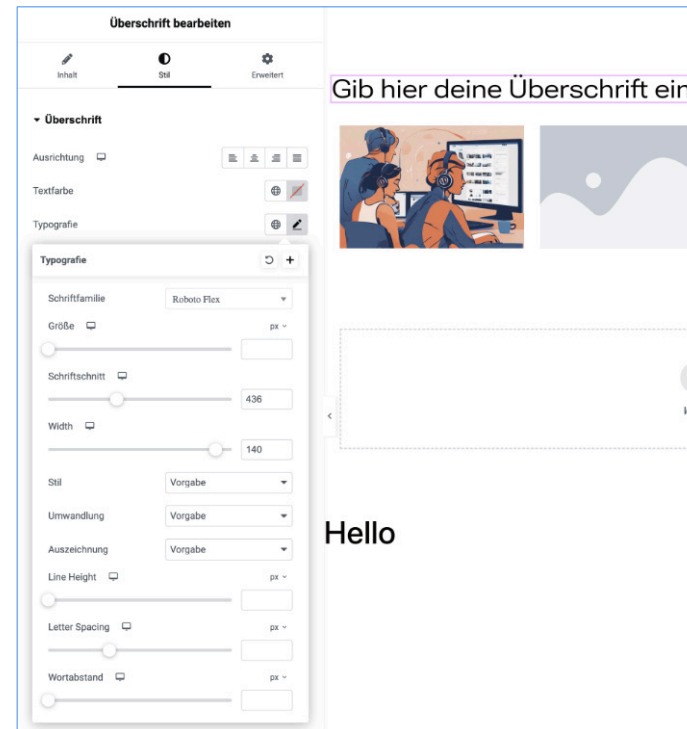
1. Wenn nichts eingestellt wurde wird die Schrift *normal* dargestellt, das entspricht dem Wert 400.
2. Die Einstellung **bold** (fett) entspricht dem Wert **700**.
3. Schriftschnitte mit numerischen Werten (eigentlich 1-1000, in der Praxis aber die 9 Stufen der hunderter-Schritte) bieten genauere Abstufungen. Natürlich müssen diese Schriftschnitte auch geladen sein damit diese dargestellt werden können.

Nach erfolgreicher Installation stehen die Schriften in den diversen Popups unter dem Punkt **Eigene Schriften** bereit. Damit können sie sowohl bei den *Globalen Schriften*, als auch beim *Theme Stil* oder bei jeder Anwendung eines *Widgets* ausgewählt werden.

Wird ein Schriftschnitt ausgewählt der nicht geladen wurde (zB. 600) wird der *nächstbeste geladene Schnitt* (etwa bold, 700) verwendet. Obwohl die modernen Schriftformate nur mehr weniger Filegröße und Ladezeit benötigen ist es trotzdem sinnvoll, in diesem Bereich sparsam zu sein.



Die Google-Schrift **Roboto Flex** ist eine Variable mit 13 Achsen voll ausgestattet Schrift, 2 davon, die Breite und die Schriftstärke, unterstützt Elementor. Die Grenzwerte müssen händisch eingetragen werden.



Die Anwendung einer Variablen Schrift als Beispiel: der Schriftschnitt wurde mit 436, die Breite mit 140 angegeben. Variable Fonts können solche Dinge darstellen, statische Fonts sind hier limitiert.

3.4. Eigene Icons verwenden

Das SVG-Format eignet sich für manche Grafiken besonders gut, zB. Logos. Das liegt daran, dass SVGs nicht nur als Files im herkömmlichen Sinn - sondern auch als Textdateien die interpretiert werden - betrachtet werden können. Daher muss man auch (wo es Textdateien gibt kann es auch immer Unsicherheiten geben) den Upload dieses Dateitypes explizit erlauben:

Dashboard > Elementor > Einstellungen > *Erweitert* > *Ungefilterte Dateiuploads aktivieren* > *Aktivieren*

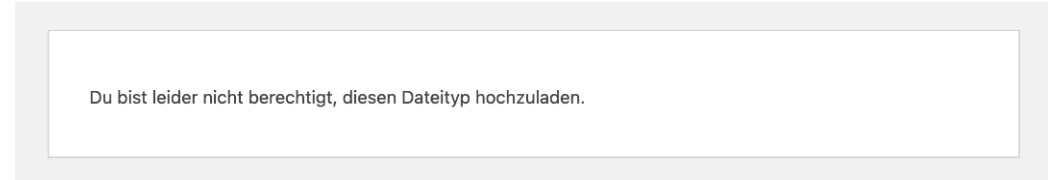
Der Erklärungstext unter der Funktion weist nochmals auf das damit verbundene Sicherheitsrisiko hin. Wenn der Upload erlaubt ist kann unter

Dashboard > Elementor > *Eigene Symbole*

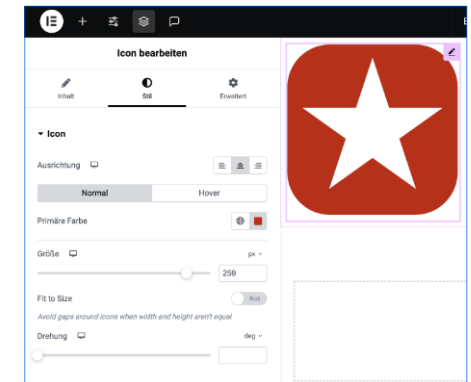
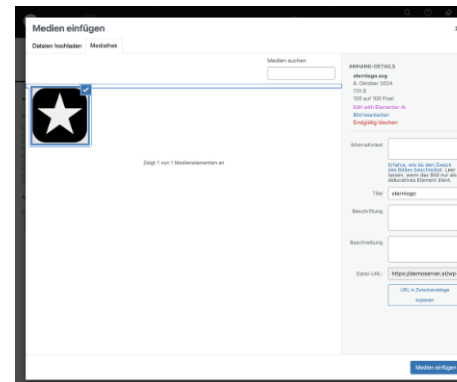
der Upload einer eigenen Icon-Bibliothek im zip-Format der Anbieter *Fontello*, *IcoMoon* oder *Fontastic* erfolgen. Einiges ist kostenpflichtig, es gibt jedoch auch viele freie Icons zur Auswahl. Leider werden die bereit gestellten ZIP-Archive nicht immer von Elementor erkannt. Manchmal bleibt nur der Weg über das Entpacken des Archivs und der manuelle Upload des gewünschten SVG-Files in die Mediathek. Grundsätzlich scheint es für die Anbieter oder Elementor schwer zu sein komplette Bibliotheken zu benutzen, dem Autor ist es nicht gelungen hier ein brauchbares Ergebnis zu erreichen.

Eine eigene Icon-Bibliothek zu nutzen bleibt dem Anwender leider verwehrt, ausser man benutzt einen der 3 Dienste und bekommt diesen auch zum Laufen. Es bleibt immer der klassische Weg über die Mediathek, über die man eigene SVG-Icons nutzen kann.

SVG-Dateien können u.a. in den Widgets Icon, Icon Box und Icon-Liste eingesetzt werden. Die großen Vorteile dieses Formates liegen in der kleinen Dateigröße, der verlustfreien Skalierbarkeit und der über die Einstellungen in den unterschiedlichen Widgets möglichen Farbgebung sowie Animation.



Wer versucht direkt in WordPress ein SVG-Icon hochzuladen wird vermutlich scheitern. Auch dann, wenn im Elementor der Upload erlaubt wurde. Aber: wenn man über Elementor den Upload startet (zB indem man das Widget Icon einfügt und dort den Upload über SVG hochladen aktiviert) klappt es dann.



Aber: nur wenn der Upload klappt ist noch nichts gewonnen. SVG-Icons neigen zu ein wenig Eigenwilligkeit. Damit sie funktionieren sollten sie schwarz sein, dann lassen sie sich vmtl. über Elementor umfärben, skalieren etc.

Je nachdem, wo man Icons erstellt (Adobe Illustrator, Affinity Designer, ...) gibt es beim SVG-Export unterschiedliche Optionen. Man muss herausfinden welche funktionieren. Ein Fingerzeig ist die Vorschau: wenn schon im WordPress die Icon-Ansicht nicht funktioniert wird es im Frontend auch nicht funktionieren. Ausserdem sollte ein Icon in der Erstellung keine Farbe haben, wenn es später umgefärbt werden soll (dazu muss man es schwarz zeichnen).

3.5. Integration von Diensten

Elementor ermöglicht als PageBuilder die Integration von diversen Diensten. In Kombination mit den unter 3.2. beschriebenen Möglichkeiten eigenen Code einzubinden kann man so die Anzahl der notwendigen Plugins reduzieren. Im Normalfall muss man einen Lizenzschlüssel oder ein Schlüsselpaar angeben, damit die Einbindung funktioniert. Manche der Dienste sind kostenlos, kontingiert (zB. Google reCaptcha, Google Maps) oder auch kostenpflichtig (Adobe Font Kit).

Dashboard > Elementor > Einstellungen > Integration

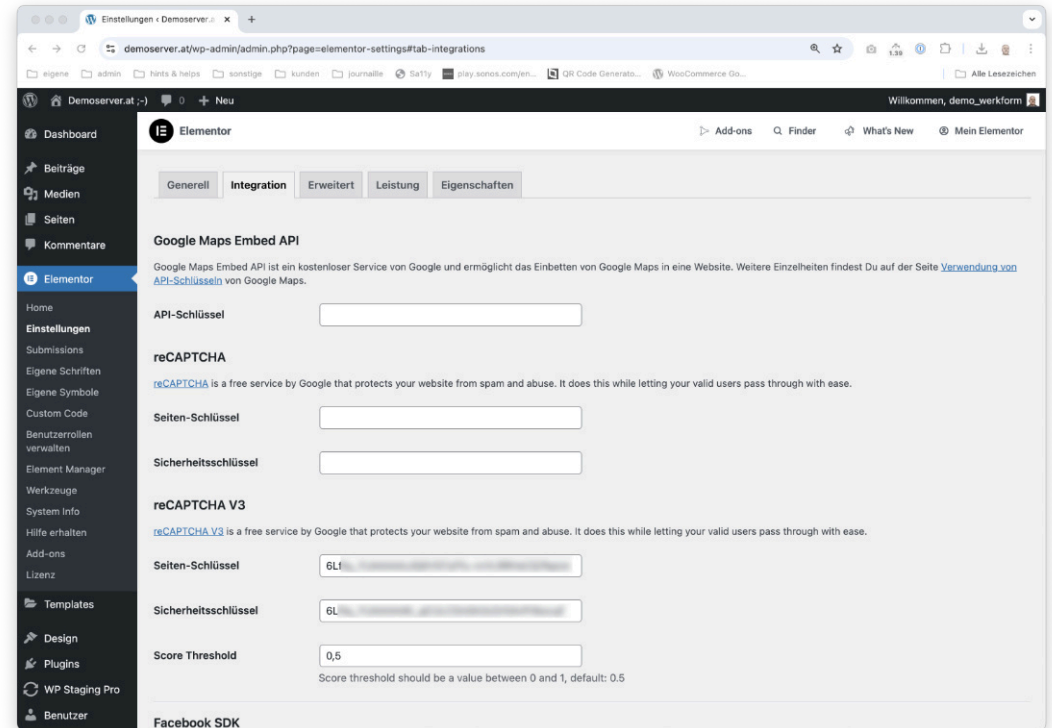
zeigt eine Liste der zur Verfügung stehenden Dienste:

- Google Maps Embed API
- reCAPTCHA
- reCAPTCHA V3
- Facebook SDK
- MailChimp
- ActiveCampaign
- ConvertKit
- Drip
- GetResponse
- MailerLite
- Adobe Fonts (TypeKit)
- Font Awesome Pro
- Stripe

3.6. Elementor Werkzeuge

Neben den in Kapitel 3.1. Bereits beschriebenen Site Kit bietet Elementor eine Reihe weiterer praktischer und auch notwendiger Werkzeuge, unter:

Dashboard > Elementor > Werkzeuge > ...



Drittlösungen werden in Elementor im Tab Integration konfiguriert. Am häufigsten werden wohl Google Dienste integriert, aber auch diverse Schnittstellen zu Marketingdiensten findet man hier sowie zum Bezahlndienstleister Stripe.

Der erste Tab *Generell* hat gleich als erstes das wichtigste aller Werkzeuge im Angebot: **CSS & Daten neu generieren**. Es kommt beim intensiven Arbeiten schon öfter vor, dass die Seite plötzlich zerfällt. Die häufigste Ursache ist ein im Cache (Browser oder Server) defektes CSS-File. Ein einfacher Klick auf diesen Button sollte - verbunden mit einem Reload des Browsers - das Problem lösen.

Ebenfalls in diesem Tab zu finden sind die Funktionen zum **Template synchronisieren**, welches die Elementor Library händisch aktualisiert (normalerweise erfolgt das automatisch täglich).

Ein weiteres - besonders wenn man eigene Plugins entwickelt oder mit Skripten hantiert - manchmal notwendiges Feature ist der **Abgesicherter Modus**, der hier de/aktiviert werden kann. Damit wird nur der Editor, nicht aber das Theme oder Plugins in den Editor geladen.

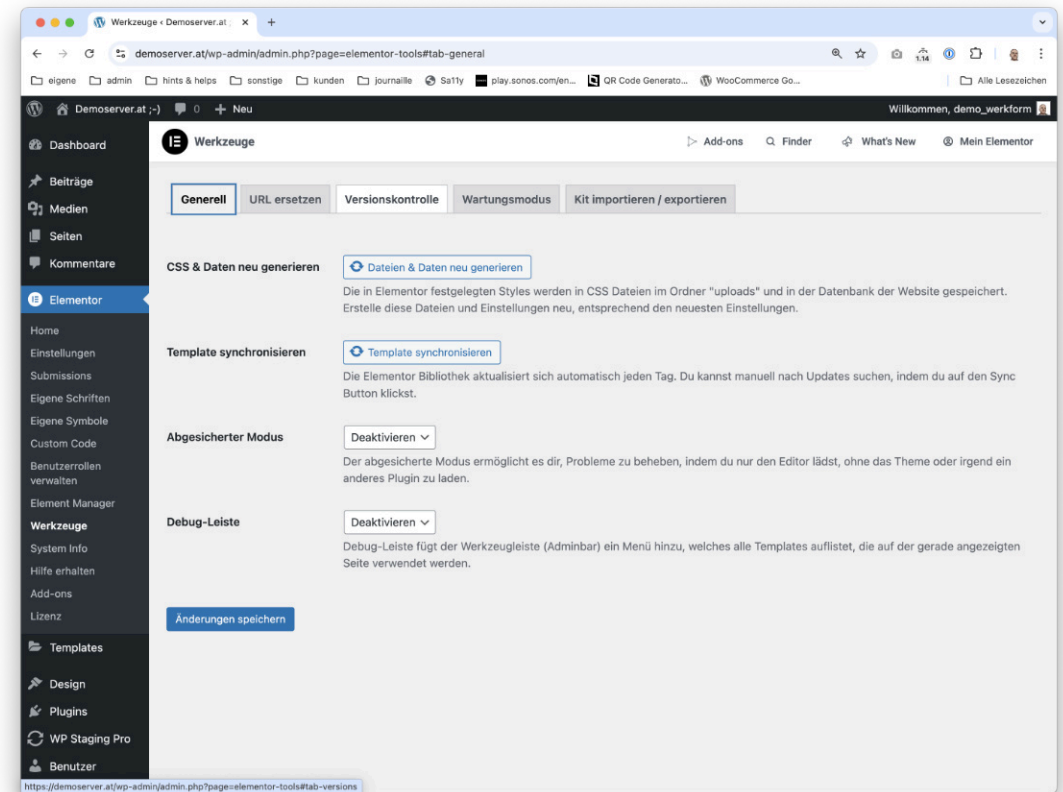
Weiters kann man eine **Debug-Leiste** anzeigen lassen.

Der Zweite Tab heisst **Url ersetzen** und bietet auch nur genau diese eine Funktion an. Diese benötigt man, wenn man eine Webseite von http auf https umstellt oder auch wenn eine Seite (händisch) übersiedelt wurde. Manche WordPress-Migrationstools erwischen nicht alle URL-Vorkommen, dann ist diese Funktion einzusetzen.

Der Dritte Werkzeug-Tab **Versionskontrolle** bietet Spielraum für besonders mutige (Beta-Tester) oder (update) faule Webmaster. Immer wenn man nach einem Update Probleme bekommt weil Plugins nicht mehr miteinander können kann hier ein temporärer Lösungsweg liegen.

Man kann die Elementor und Elementor Pro getrennt von einander in der installierten Version erneuern oder aber auch Versionsschritte zurück setzen. Ausserdem kann man hier dem *Betatester*-Programm beitreten - die Warnung, das nicht auf Produktivseiten zu machen, kann nur eindrücklich wiederholt werden!

Im Tab **Wartungsmodus** schließlich ist die Funktion verborgen eine Webseite in *Coming Soon* oder eben den *Wartungsmodus* zu versetzen. Der Unterschied



Elementor Werkzeuge sind eine Sammlung nützlicher und auch notwendiger Tools.

besteht darin, dass ersterer einen HTTP-200 Status zurückschickt, also der Server ganz normal antwortet, während der Wartungsmodus einen Status 503 zurückgibt. Dadurch weiß z.B. eine Suchmaschine, dass dieser Inhalt nicht relevant ist, weil die eigentliche Seite im Augenblick nicht erreichbar ist. Der Wartungsmodus sollte nicht zu lange aktiviert werden.

Egal wofür man sich entscheidet, die Vorgangsweise ist die selbe: zuerst ist ein entsprechendes Seiten-Template zu erstellen, und zwar unter

[Dashboard](#) > [Templates](#) > [Gespeicherte Templates](#) > [Seite](#)

Dieses Template ist dann auf der Wartungsmodus-Seite auszuwählen.

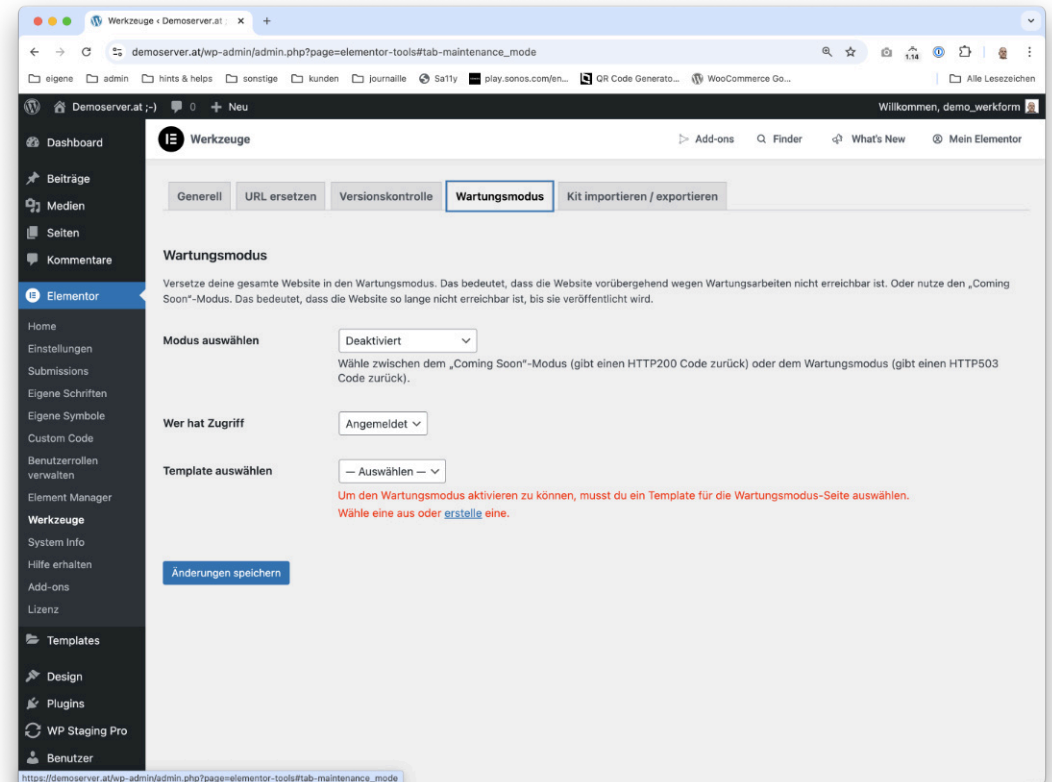
Die Werkzeuge **Kit importieren / exportieren** werden in Kapitel 3.1. beschrieben.

3.7. Weitere Elementor Plugins & Services

Elementor versteht sich nicht mehr nur als Plugin-Hersteller sondern entwickelt sich zu einem Allround-Anbieter weiter. So bietet Elementor auch ein eigenes Hosting-Service weltweit (auf Google Cloud Basis) an. Dazu kommen noch verschiedene damit verbundene Leistungen wie Migrationsservice, Updateservice, Formularintegration etc. Diese werden im Regelfall einmalig verrechnet.

Dazu kommen noch Dienstleistungen, die mit dem Account verbunden sind und nicht mit einer Site und andere Plugins, die zusätzlich zu Elementor und Elementor Pro entwickelt werden.

WordPress Hosting. Die verschiedenen Hosting Pakete bestehen aus dem physischen Webhost und einer Elementor Installation. Die Pakete werden von *Lite* bis *Ultimate* bezeichnet und beinhalten verschiedene Versionen von Elementor (Pro) und sind für einfache Websites bis WooCommerce-Shops geeignet. Verbunden damit ist immer eine Support-Möglichkeit (die Erfahrung lehrt Geduld) und eine WordPress-Umgebung, die nicht immer alle Plugins zulässt.



Der Wartungsmodus hilft schon bei der Erstellung einer Website, um diese vorerst vor Besuchern abzuschirmen (Coming Soon).

Elementor AI. Innerhalb von Elementor kann man auf die AI bei Texten, Bildern und bei Code-Eingabe zugreifen. Ausserdem gibt es Unterstützung beim Design, was Elementor AI *Copilot* nennt. Die AI basiert auf einem Credit-System, wobei Texte „billig (1)“ und Bilder „teuer (33)“ sind. Allerdings ist es für den Anwendern nicht gerade einfach zu verstehen wieviel man verbraucht hat und was ein Klick auf so ein Service kostet, denn oft reicht ja der erste Klick nicht und wie ist das mit den Variationen, die man nicht bestellt hat?

Die Erfahrungen mit der Qualität kann jeder selber machen indem das kostenlose Einstiegsangebot nutzt. Die Erfahrungen des Autors sind gemischt, praktisch angewendet werden die Illustrationen bei der Bildgenerierung und Übersetzungen, hin und wieder die Textgenerierung bei wenig bedeutenden Marketingtexten. Bisher nicht wirklich erfolgreiche Versuche waren bei fotorealistischen Bildern. Der generierten CSS-Codes war mäßig brauchbar. Nicht verwendet wurden die Optionen zum Design, schon gar nicht die mit Angabe einer „Inspirationsquelle“.

Preise starten € 48,- pro Jahr und sind an den User gebunden, dh. auf beliebig vielen Websites einsetzbar!

Image Optimizer. Ein kostenpflichtiger Bildoptimierer auf Creditbasis. Macht das, was andere (auch kostenlose) Plugins auch machen. Kann aber direkt aus Elementor aufgerufen werden. Wer's braucht ...

Site Mailer. Ersetzt ein SMTP-Plugin für den E-Mail-Versand. Verrechnet pro Domain und Anzahl der versandten E-Mails. Ganz neu (Herbst 2024), vom Autor nicht getestet.

4. Theme Builder

Der Theme-Builder ist ein entscheidendes Argument, warum man die Pro-Version von Elementor verwenden sollte. Damit kann man rasch konsistente Websites erstellen die auch langfristig wartungsarm und trotzdem flexibel bleiben.

Elementor unterscheidet (inkl. WooCommerce) 10 verschiedene Template-Arten. Das klingt komplizierter als es ist. Auf WooCommerce wird in diesem Skriptum nicht näher eingegangen.

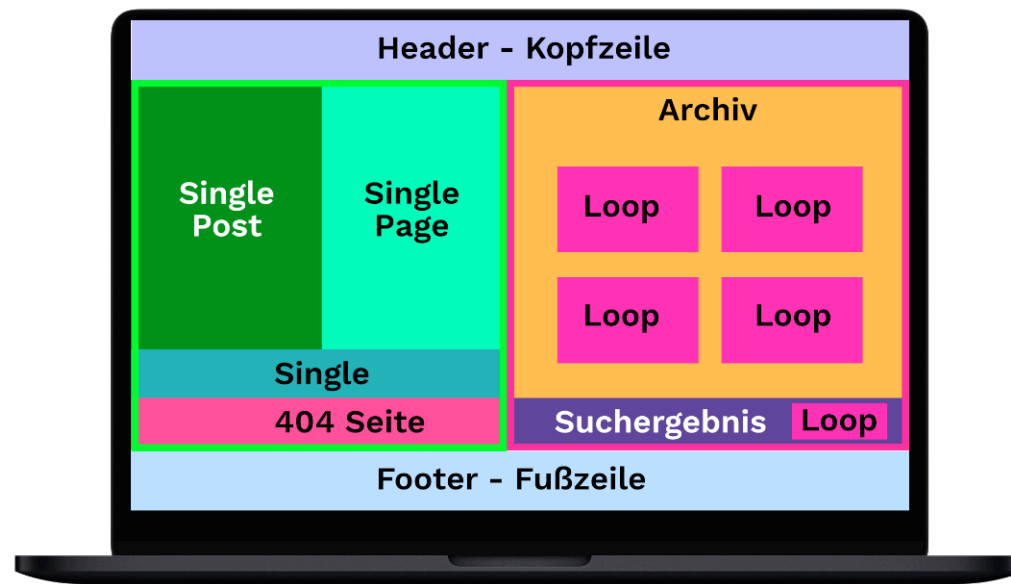
Alle diese Templates sind optional zu verstehen. Während **Kopfzeile** (Header) und **Fußzeile** (Footer) im Regelfall einmalig gestaltet (=statische Templates) und dann nach eingestellter Bedingung (zB: gesamte Website) angezeigt werden, werden die Templates für **einzelne Beiträge** (Single Post) oder **Produktliste** (Product Archive) unterschiedliche Inhalte anzeigen. So gesehen sind es *dynamische* Templates, also Templates, mit denen man unterschiedliche Dinge anzeigen lassen kann.

Ob man nun einen einzelnen Beitrag, eine Seite oder auch ein Produkt anzeigt mag inhaltlich sehr unterschiedlich sein, die Logik dahinter ist jedoch immer gleich. Genauso ist eine **Archivseite** immer eine Auflistung von mehreren Singles, ob das nun Produkte, Posts oder Suchergebnisse sind ist nur eine kleiner Unterschied.

Die einzelnen Posts auf den Archivseiten nennt man **Loop**. Seit Version 3.8. sind auch Loop-Templates im Elementor angekommen.

Die **404-Seite** (also die Anzeige für den HTTP-Status für nicht gefundene Inhalte) ist nichts anderes als eine statische Seite. Spannend ist nur wann sie angezeigt wird, also zu welcher Bedingung sie ausgespielt wird.

Mit der Kombination der verschiedenen Theme-Builder-Templates kann man alle Bestandteile einer Webseite gestalten. Daher benötigt man auch kein Theme mehr das eigenes Design anbietet – das übernimmt der Theme-Builder. Die Gestaltung selber unterscheidet sich nicht von der sonstigen Arbeit mit Elementor. Je nach Template werden unterschiedliche Widgets zur Anwen-



Der Theme-Builder ist das Werkzeug-Set, das Templates für die ganze Website organisiert. Im Unterschied zu anderen Templates werden jene des Theme-Builders über Bedingungen gesteuert.

dung kommen. Beispielsweise wird das Widget *Site-Logo* wohl im Header eingesetzt während das Beitrags-Bild in einer Single-Umgebung zu Hause ist.

Elementor bietet 2 Wege an im Theme-Builder zu arbeiten: den klassischen über

Dashboard > Templates > Theme Builder

Und auch den *New Theme Builder* der einerseits über einen Promotion-Link im klassischen Theme-Builder, andererseits auch über einen Menüeintrag im Elementor selber

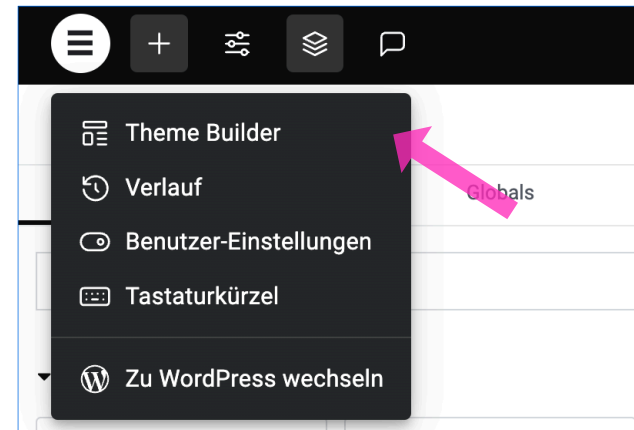
Panel > Theme Builder

zu erreichen ist. Das Wesentliche an den Theme-Builder-Templates sind aber nicht die Gestaltungsmöglichkeiten sondern die Bedingungen, die man zu diesen Templates setzen kann.

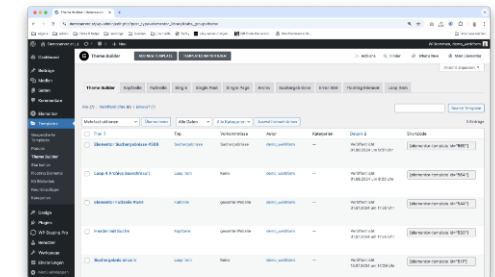
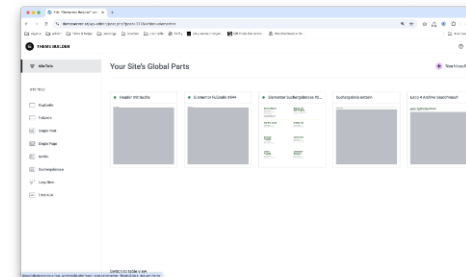
Beispielsweise legt man einen Header an, indem die Funktion *Add New Template* auswählt. Damit öffnet sich ein Modalfenster mit Auswahl aus der Elementor-Bibliothek, vorausgewählt sind Header-Blöcke. Je nach Anwendung sucht man hier eine Vorlage aus und passt diese an oder man verzichtet auf eine Vorlage und geht „from scratch“, also von einem weißen „Blatt Papier“ aus.

Beim Speichern fragt Elementor nach den Bedingungen (Conditions), zu welchen das Template angezeigt werden soll. Im Dialogfenster kann man nun auf „Bedingung hinzufügen“ klicken und bekommt als Vorschlag *Beifügen, Gesamte Webseite* angeboten. Wer das bestätigt bekommt das Template (in dem Beispiel einen Header) immer angezeigt (so ferne das WordPress Template *Elementor Gesamte Breite* gewählt wurde).

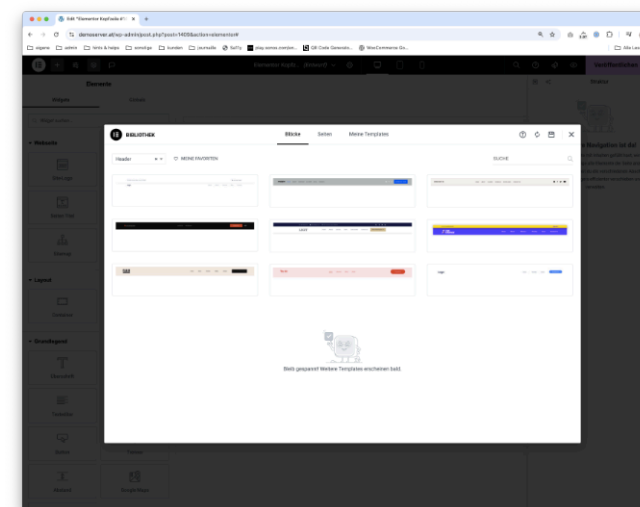
Man kann die Bedingungen auch nachträglich bearbeiten. Dazu kann man rechts neben dem Veröffentlichen-Button auf den Pfeil klicken um das Popup zu erhalten das u.a. zu den Bedingungen führt. Alternativ kann man im neuen Theme-Builder unter jedem Template den Link zu den Conditions finden.



Über das Elementor Icon links oben geht's zum Theme Builder.



Ob man die Elementor eigene Oberfläche (links) bevorzugt oder die klassische WordPress-Tabelle (rechts) ist reine Geschmacksache: der Theme-Builder ist eine Sammlung von Templates.



Die Elementor-Bibliothek bietet eine kleine Auswahl an möglichen Designs, doch der Start mit einem leeren Container ist oft die bessere Wahl.

Während bei Kopf- und Fußzeilen die Bedingungen normalerweise recht simpel auszuwählen sind kann es bei dynamischen Inhalten schon komplexer werden. Man kann zB. unterschiedliche Archivseiten und Single-Pages bei unterschiedlichen Post-Typen verwenden. Also ein ganz eigenes Layout für Produktbeschreibungen verwenden und ein anderes für Beiträge und wieder ein anderes für statische Seiten.

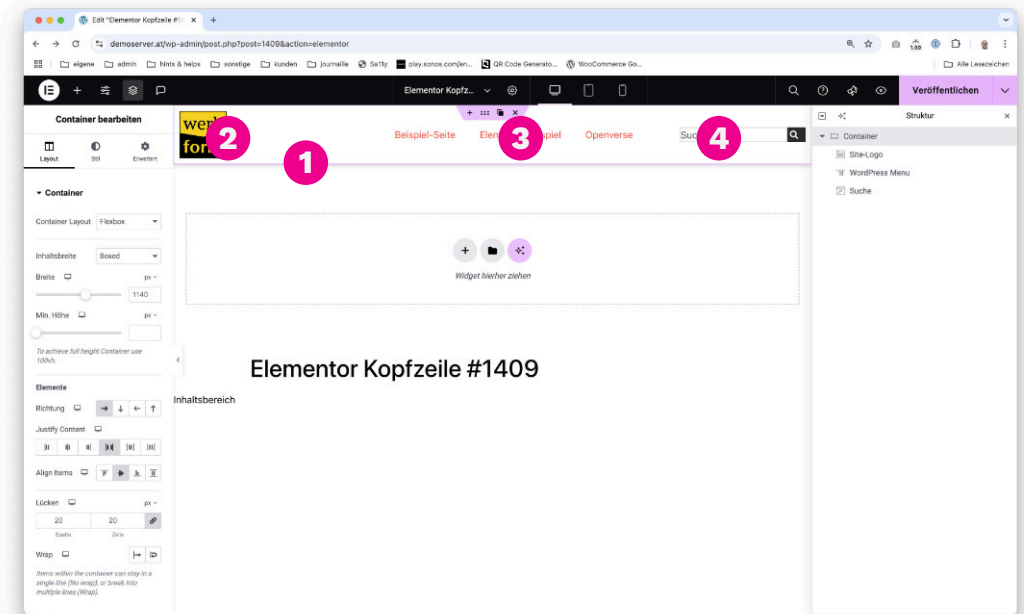
4.1. Kopfzeile - Header

Als wichtiges Gestaltungselement verdient die Kopfzeile einer Webseite die größte Aufmerksamkeit. Webuser erwarten im Header Logo und ggf. ein Menu, als Gestalter will man bereits hier alle wesentlichen Elemente des Corporate Designs wie Farben und Schriften unterbringen.

Ein Header besteht aus einem oder mehreren Containern, kann über die volle Breite gehen oder nur die Inhaltsbreite einnehmen und muss auch in den verschiedenen responsive Modi funktionieren. Im Beispiel nebenan wurde eine Kopfzeile erstellt die Logo, Menu und Suchfunktion beinhaltet.

Der schrittweise Aufbau dieses Header erfolgt so:

1. Einen Container in das Feld der Arbeitsfläche ziehen oder auf das **+** klicken und die Struktur mit einem Feld und horizontaler Ausrichtung wählen.
2. Tab *Layout*: der Container hat die **Inhaltsbreite Boxed**, die **Richtung** (Direction) läuft *von links nach rechts*. Die **Ausrichtung** der Inhalte (Justify Content) wurde mit *Abstand dazwischen* (Space Between) gewählt um die Widgets im Container zu verteilen. Die **Vertikale Ausrichtung** (Alignment) wurde mit *zentriert* (center) gewählt, damit die 3 Widgets an einer Mittelachse angeordnet werden. Unter **Zusätzliche Optionen** (unter *Container* im Panel) bei **HTML-Tag** > *header* auswählen.
3. Im *Stil*-Tab wurde unter **Rahmen** die Funktion *Box-Schatten* aktiviert ohne einen Rahmen zu setzen. Dadurch entsteht ein leichter Schatten zwischen Kopfzeile und Inhalt der Seite.



Der Container (1) sorgt für die Breite, Ausrichtung und Verteilung der Header-Elemente. Ausserdem ist er zuständig für den leichten Schlagschatten. Das Logo (2) wird mit dem Widget Site-Logo eingesetzt (das ist mit der Startseite verlinkt) Die Menüpunkte werden über das WordPress-Menü-Widget (3) eingesetzt, wobei das Menü selber über das WordPress-Dashboard verwaltet wird. Das Widget Suche (4) integriert die Interne Suche, die über alle Inhalte der Website ausgeführt wird.

4. Im Tab *Erweitert* bei CSS-ID einen Namen eintragen (zB. *top*). Damit kann man dieses Element als Sprungmarke nutzen und mit einem Link im Footer der Seite ansteuern.
5. Die Widgets *Site-Logo*, *Navigations-Menu* und *Suchformular* wurden in den Container gezogen und individuell angepasst.

Damit ist die Kopfzeile grundsätzlich gestaltet. Im nächsten Schritt werden nun die Anpassungen für das responsive Design - primär für die Mobil-Ansicht - durchgeführt. Dazu wechselt man über den entsprechenden Schalter in die Ansicht für Mobilgeräte. Im Beispiel hier sollen Logo und Menü (als Hamburger-Menü) untereinander angeordnet werden, die Suchfunktion soll am Handy nicht angezeigt werden.

Um nun die Widgets im Header nicht mehr nebeneinander sondern untereinander anzuordnen muss nur der Container aktiviert werden und dann die Richtung von *Horizontal (Row)* auf *vertikal (Column)* geändert werden. An dem Icon für mobile Ansicht neben der Richtungseinstellung erkennt man, dass es sich hier um die Einstellung **nur** für den Modus *Mobile* handelt.

Da zuvor bei der Ausrichtung (*Alignment*) *zentriert (Center)* gewählt wurde sind die Elemente auch hier, nur eben um die Vertikalachse, zentriert.

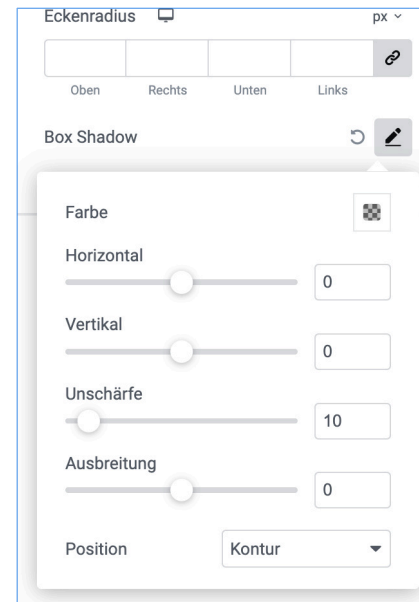
Elementor bietet die Möglichkeit jedes Element abhängig vom Ansichtsmodus auszublenden. In diesem Fall wurde das Widget **Suchformular** markiert und danach unter

... > **Erweitert** > **Responsive** > **Auf Mobile ausblenden**

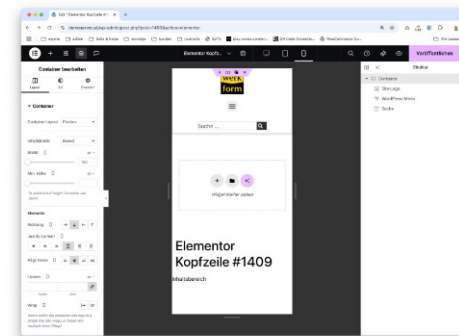
aktiviert. Das Widget **Navigations-Menu** sorgt dafür, dass in der mobilen Ansicht automatisch auf Mobiles-Aufklapp-Menü gewechselt wird (=Voreinstellung). Für ein besseres Verhalten am Handy wird unter

... > **Layout** > **Mobiles Ausklapp-Menü** > **Volle Breite** > **Ja**

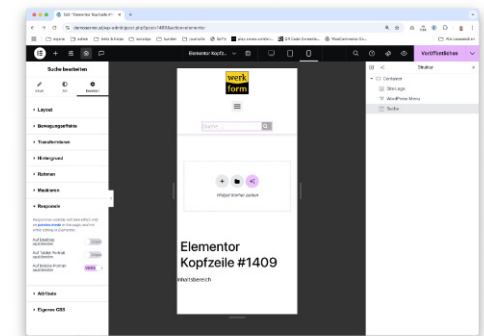
aktiviert und dann bei **Ausrichtung (Text-Align)** *Mitte* eingestellt.



In den Rahmen-Einstellungen (Erweitert) findet man den Schalter für den Schlagschatten (Box Shadow). Bei aktiviertem Schatten lassen sich Parameter und Schattenfarbe genau an das Wunschdesign anpassen.



Nach Wechsel auf die Mobil-Ansicht kann man die Richtung des Containers ändern, um die Widgets untereinander anzuordnen.



Das Suchfeld wird in der Mobil-Ansicht ausgeblendet, daher wird es grau schraffiert dargestellt (Erweitert > Responsive).

Zur genaueren Gestaltung bietet der Stil-Tab die beiden Bereiche *Aufklappen* und *Umschaltknopf*.

Die Anpassungen für weitere Schirmgrößen werden analog zu der beschriebenen Weise durchgeführt. Zurück in der Desktop-Version kann man kontrollieren, ob es zu keinen Verschiebungen oder anderen Änderungen gekommen ist.

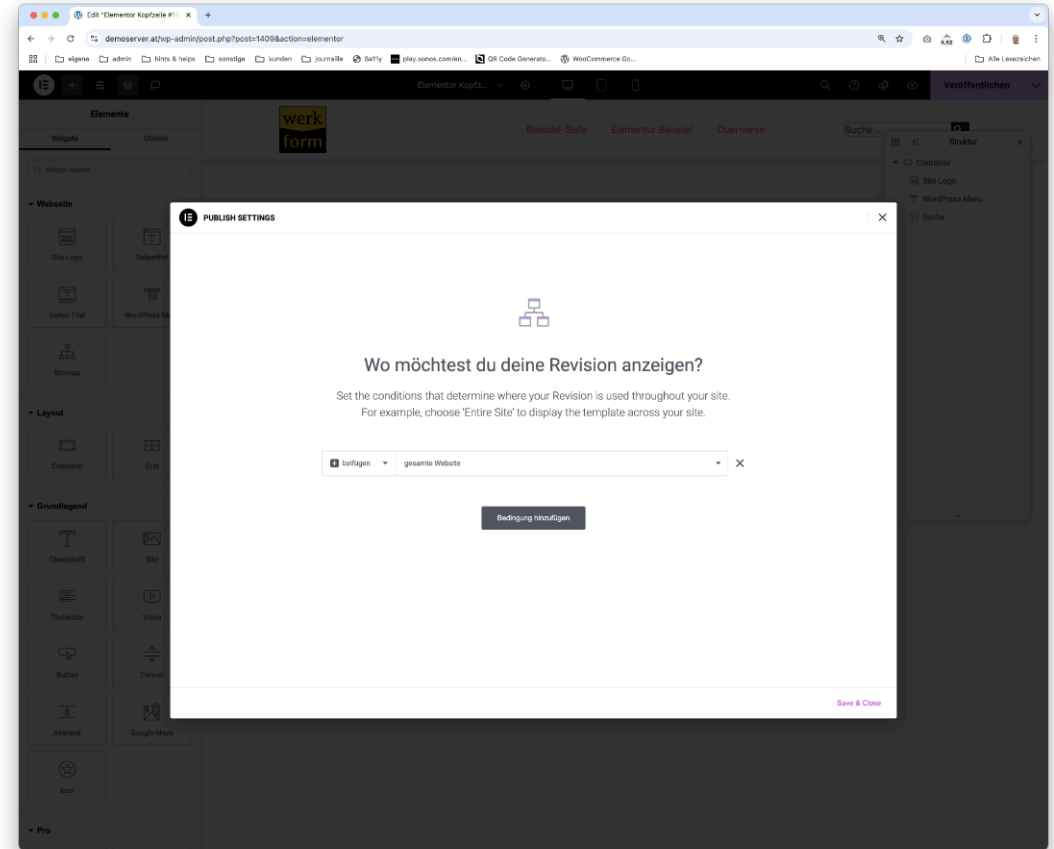
Wer will, dass die Kopfzeile am oberen Bildschirmrand stehen bleibt, wählt bei aktivem Container

... Container bearbeiten > Erweitert > Bewegungseffekte > Sticky > oben & Sticky Ein > Desktop/Tablet

Beim Speichern wird man nun im Dialog gefragt, wo (also genau genommen zu *welchen Bedingungen*) diese Kopfzeile angezeigt werden soll.

Vorausgewählt ist *gesamte Website*. Wer das nun bestätigt bekommt diese Kopfzeile **immer** angezeigt - so ferne die Seite die Einstellung *Elementor Volle Breite* hat (was eben genau bedeutet, dass bei dieser Einstellung die Theme-Template ausgespielt werden sollen). Über diese Bedingungen ist es möglich verschiedene Kopfzeilen für verschiedene Bereiche der Webseite zu erstellen, als zB. unterschiedliche Header für statische Seiten, Blog oder Shop.

Der aktuelle Trend geht zu immer dezenteren Kopfzeilen die nur mehr aus Logo und Navigations-Icon bestehen. Header sollten nie überladen sein und den User möglichst gut unterstützen, eine generelle Antwort welche Elemente er beinhalten soll, und wie eine Kopfzeile aussehen soll, kann es nicht geben.



Beim Speichern (oder über das Submenü *Optionen Speichern* neben dem Veröffentlichen-Button) gelangt man in den Bedingungen-Dialog. Trotz des in Elementor (leider) üblichen Sprachen-Wirrwars funktioniert das ganz intuitiv.



Dir gefällt dieses Skriptum?

Kaffeespende
hier!

jetzt spenden

Spende Skriptum 3.0



Anzeige

4.2. Fußzeile - Footer

Die Umsetzung von Fußzeilen erfolgt ganz genau dem Schema von Kopfzeilen. Footer zeigen oft eigene Menüs oder Linklisten (Impressum, Datenschutzlinks etc.). Ausserdem werden oft Copyright-Hinweise ausgespielt und ausserdem gibt es manchmal Links nach oben auf der Seite.

Dynamisches Copyright

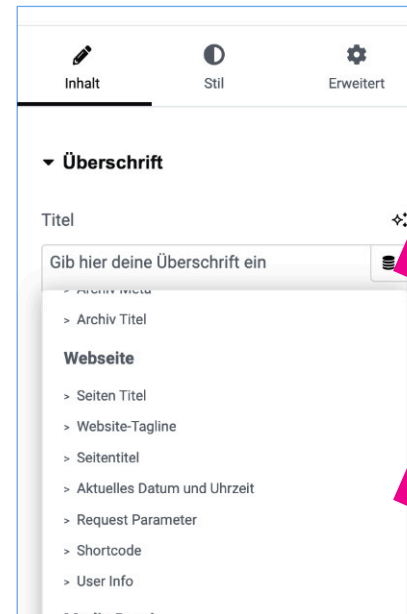
Elementor unterstützt **dynamische Inhalte** an sehr vielen Stellen in allen Widgets. Die Funktionsweise ist am Beispiel eines dynamischen Copyright-Hinweises einfach zu erklären. Das Ziel ist es, eine Zeile wie

© 2025 skriptum. Alle Rechte vorbehalten.

zu erhalten. Dabei soll die Jahreszahl immer aktuell bleiben, dh. an dieser Stelle soll das **aktuelle Jahr** angezeigt werden. Die Schritte zur Lösung sind folgende:

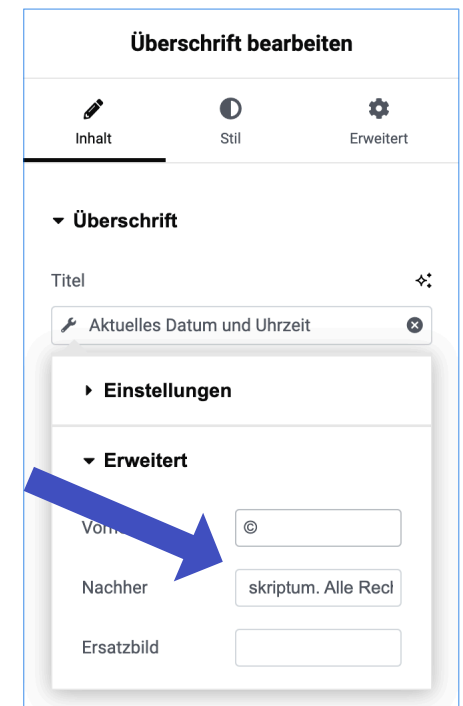
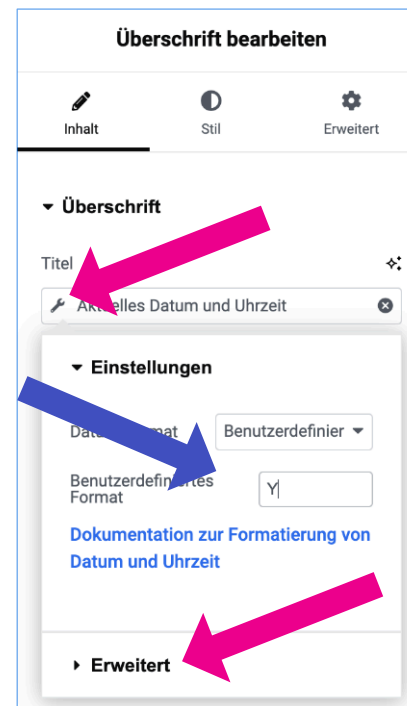
1. Ein geeignetes Widget wählen, beispielsweise *Titel*, dort den HTML-Tag auf *H6* setzen.
2. Anstatt einen Text einzugeben klickt man auf das kleine **Icon rechts oben** in der Textbox. Im aufklappenden Menü wird ... > Webseite > Aktuelles Datum und Uhrzeit gewählt.
3. Auf den nun erscheinenden kleinen Schraubenschlüssel neben dem Titelplatzhalter klicken. Datumsformat > Benutzerdefiniert auswählen.
4. Im Feld Benutzerdefiniertes Format **Y** eintragen (steht für Jahr, 4-stellig)
5. In den darunter befindlichen Tab *Erweitert* wechseln, dort den Text für *vorher* (©) und *nachher* (skriptum. Alle Rechte vorbehalten.) eingeben. Leerzeichen sind notwendig.

Nun wird aus den beiden statischen Textstücken und dem aktuellen Jahr die



Statt eine Überschrift einzugeben klickt man auf das Icon für dynamische Inhalte. Danach wählt man Aktuelles Datum und Uhrzeit aus.

Das Werkzeugsymbol neben der Auswahl führt zu weiteren Eingabeoptionen.



fertige Textzeile generiert. Die Formatierung kann wie gewohnt über die Tabs *Stil* und *Erweitert* erfolgen.

Mehr über dynamische Inhalte und wie man diese mit Hilfe von eigenen Feldern und Post-Typen einsetzt in Kapitel 12.

Nach oben - der Toplink

Icons mit einem Pfeil nach oben findet man häufig in Fußzeilen. Technisch gesehen sind das nichts anderes als einfache Links innerhalb der bestehenden Seite, wobei das Ziel der Links eben der Seitenanfang ist.

Man benötigt zuerst also das Ziel, welches angesprungen werden soll. Dieses kann man entweder mit dem Widget *Menü-Anker* umsetzen - oder man vergibt dem ersten Element auf der Seite eine CSS ID. Die Eingabemöglichkeit findet man unter

...Container / Widget > **Erweitert** > CSS ID

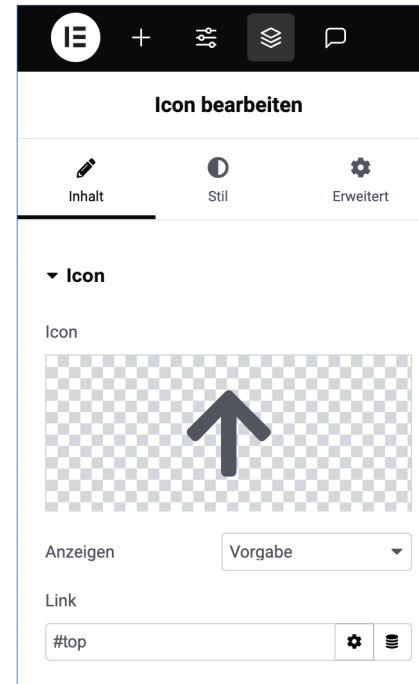
Leerzeichen sind Tabu, sinnvoll sind kleingeschriebene, eindeutige Begriffe, zB. **top**. Danach benötigt man ein Icon oder einen Text in der Fußzeile mit dem entsprechenden Link. In diesem Beispiel wurde das Widget *Icon* verwendet. Der einzugebende Link heisst `#top`, die `#` symbolisiert, dass die aktuelle Seite verwendet werden soll um die ID mit dem Namen „top“ anzuspringen.

4.3. Single

Die **Tab-Ansicht** des Theme Builder bietet einen eigenen Tab *Single* an über den man zwar ein Template erstellen kann, das dann aber nicht im **New Theme Builder** auftaucht. Vielmehr legt man hier ein Seiten-Template an wie man es zB. für eine Maintenance-Mode-Seite verwenden kann. Die Templates sind dann auch unter

Templates > **Gespeicherte Templates**

zu finden. Das ist zugegeben verwirrend.



Der Toplink ist ein häufiger Inhalt im Footer. Einfach ein Icon-Widget mit dem Link `#top` als Ziel. Diese ID muss entweder über das Widget Menü-Anker oder über die CSS-ID in einem Widget des Headers angelegt werden.

4.4. Single Post

Dieses Template wird verwendet wenn einzelne Beiträge (=Posts, auch eigene Post-Typen) angezeigt werden. Der wesentliche Unterschied zu Pages liegt in der Art, wie die Daten dargestellt werden. Technisch ist eigentlich zwischen Post (Beitrag) & Page (Seite) kein großer Unterschied. Beiträge können Kategorien haben, Seiten nicht. Beiträge haben Kommentare, Seiten im Regelfall nicht. Bei Beiträgen wird ein Beitragsbild verwendet, bei Seiten eher nicht. Beiträge werden meist mit Datum dargestellt, Seiten eher nicht.

Ein klassisches Post hat neben den eigentlichen Inhalten also ein paar Features, die über eigene Widgets eingebaut werden können. In der Widget-Rubrik Single sind folgende Widgets gelistet:

- Beitrag Titel
- Post Ausschnitt (= Textauszug)
- Beitrag Inhalt
- Beitragsbild
- Table of Contents
- Autor Box
- Beitrag Kommentare
- Beitrag Navigation
- Beitrag Information
- Progress Tracker

Diese Widgets sind dynamisch: ihre Inhalte sind unterschiedlich, je nachdem welchen Beitrag sie zeigen.

Für die Arbeit an den Inhalten der Beiträge, die mit diesem Templates dargestellt werden, wird der WordPress interne Blockeditor verwendet. Autoren und Redakteure dürfen (müssen) nicht mit Elementor arbeiten.

Das gesamte Templates besteht aus Widgets, die dynamisch befüllt werden. Man gestaltet also einen Rahmen ohne Inhalte, eine Vorlage ohne Texte, wenn man ein Single Post Template erstellt.

Entscheidend sind wieder die Bedingungen, zu denen dieses Template ausgespielt werden soll. Im einfachsten Fall wird es für alle Beiträge angewandt.

Ein Sonnenblumen-Beitrag

Alles was im großen Bereich steht, wird über das Widget Beitrag Inhalt im Template eingebunden.

Tippe /, um einen Block auszuwählen

Beitrag Block

Ein Sonnenblumen-Beitrag

Generate with Elementor AI

Das ist jener Text, der auch als Post-Ausschnitt bezeichnet wird.

Textauszug bearbeiten

15 Wörter, 1 Minute Lesezeit.
Zuletzt bearbeitet vor 2 Minuten.

Status Entwurf

Veröffentlichen Sofort

Link /ein-sonnenblumen-beitrag

Autor demo_werkform

Template Standard-Template

Diskussion Offen

Oben gehalten

Kategorien

Schlagwörter

▼ Single

Beitrag Titel

Post-Ausschnitt

Beitrag Inhalt

Beitragsbild

Table of Contents

Autor Box

Beitrag Kommentare

Beitrag Navigation

Beitrag Information

Progress Tracker

Die ersten 4 Single-Widgets dienen dazu, als Platzhalter die Inhalte des Beitrags im Template unterzubringen.

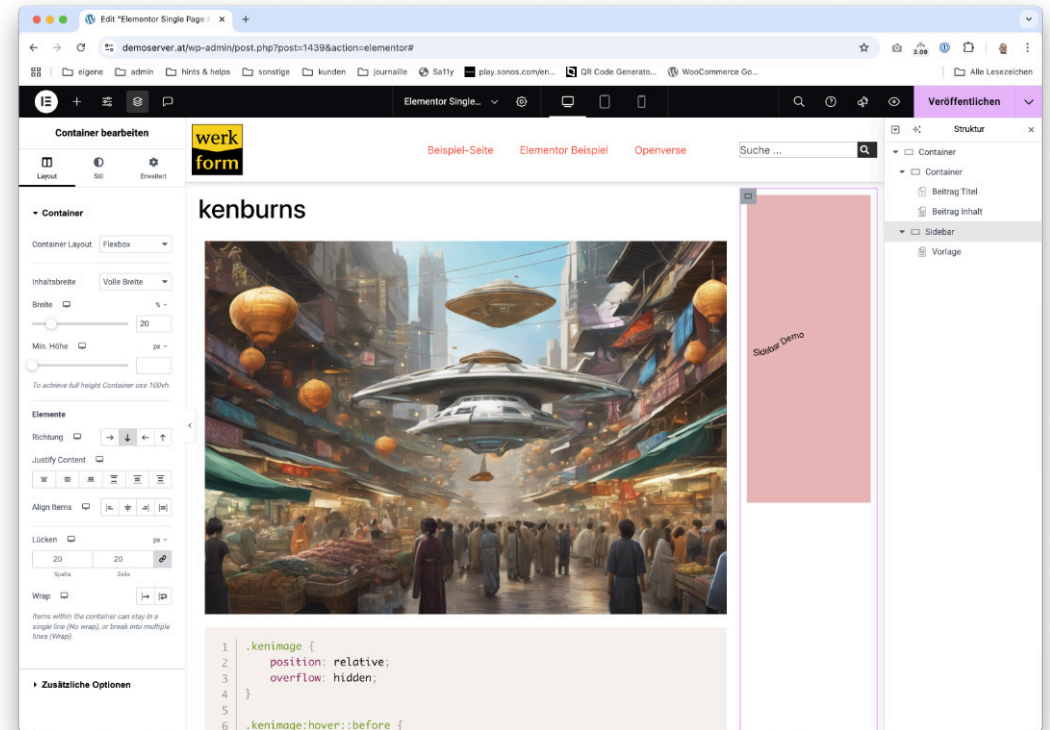
Die anderen Widgets zeigen Meta-Informationen (Autor Box, Beitrag Information), die Kommentare (Beitrag Kommentare), Navigieren zum vorigen/nächsten Beitrag (Beitrag Navigation) oder sind Widgets, die innerhalb des Beitrages unterstützen sollen (Table of Contents, Progress Tracker).

Möglich ist aber auch, ein Template nur für bestimmte Kategorien oder einen bestimmten Autor zu gestalten. Man kann auch nach bestimmten Post-Typen eine Bedingung festlegen.

Über die *Seiten-Einstellungen* kann man über *Vorschau-Einstellungen* einen Demoinhalt auswählen.

4.5. Single Page

Analog zum davor besprochenen *Single Post* wird ein *Single Page* Template hergestellt. Da Seiten per Definition weniger komplex sind wird man auch weniger Widgets benötigen. Häufig werden bei Seiten auf Autor, Kommentare, Featured Images (Beitragsbild) etc. verzichtet. Viel mehr handelt es sich um eher einfache Vorgaben die sich mehr um Layoutvorgaben etc. kümmern. Wenn man davon ausgeht, dass mittels Kopf- und Fußzeile das grundlegende Design bereits steht sind auch die gemeinsamen Merkmale, die die klassischen Seiten (Home, Kontakt, Impressum, DSGVO, etc.) eher wenige. Praktisch sind diese Templates, wenn man ein traditionelles WordPress-Design mit Sidebars anlegen will. Das Beispiel zeigt einen Sidebar der über das Widget Vorlage mit ausgewähltem Container-Template realisiert wurde. Das mag auf den ersten Blick ein wenig kompliziert wirken, führt aber dazu, dass jeder abgrenzbare Teil einer Website nur ein Mal erstellt werden muss. Da man die Anzeigebedingungen für Templates sehr genau einschränken kann wird es immer wieder Anwendungen geben, wo man auch für Seiten Templates benötigen kann.



Bei diesem Beispiel wurde ein Single Page Template mit einer 2spaltigen Container-Konstruktion erstellt. Links wurden mit Beitrag Titel und Beitrag Inhalt die Seiteninhalte eingebaut, rechts wurde mit dem Widget Vorlage ein Container-Template (rosa) eingebunden. Diese Seitenkonstruktion taucht immer dann auf, wenn als Seiten-Template Standard gewählt wird.



Willst du deinen Elementor-Coach kennen lernen?

Erstgespräch kostenlos!

Termin hier!



Harry Martin

4.6. Die 404-Seite

Ein Sonderfall für eine einzelne Seite ist die Statusseite 404. Diese Seite wird vom Server ausgespielt, wenn die verlangte Seite innerhalb eines Webhostes nicht vorhanden ist. Dieser HTTP-Status hat die Nummer 404 - daher der Name.

Wie die Gestaltung gewählt wird (mit Suchfunktion, Navigationsmenü, mehr oder weniger passenden Bild) ist eine Geschmacksfrage. Die einzige Notwendigkeit besteht im Bedingungsdialog, der beim Speichern aufgerufen wird: hier muß **404 Seite** ausgewählt werden.

4.7. Archive

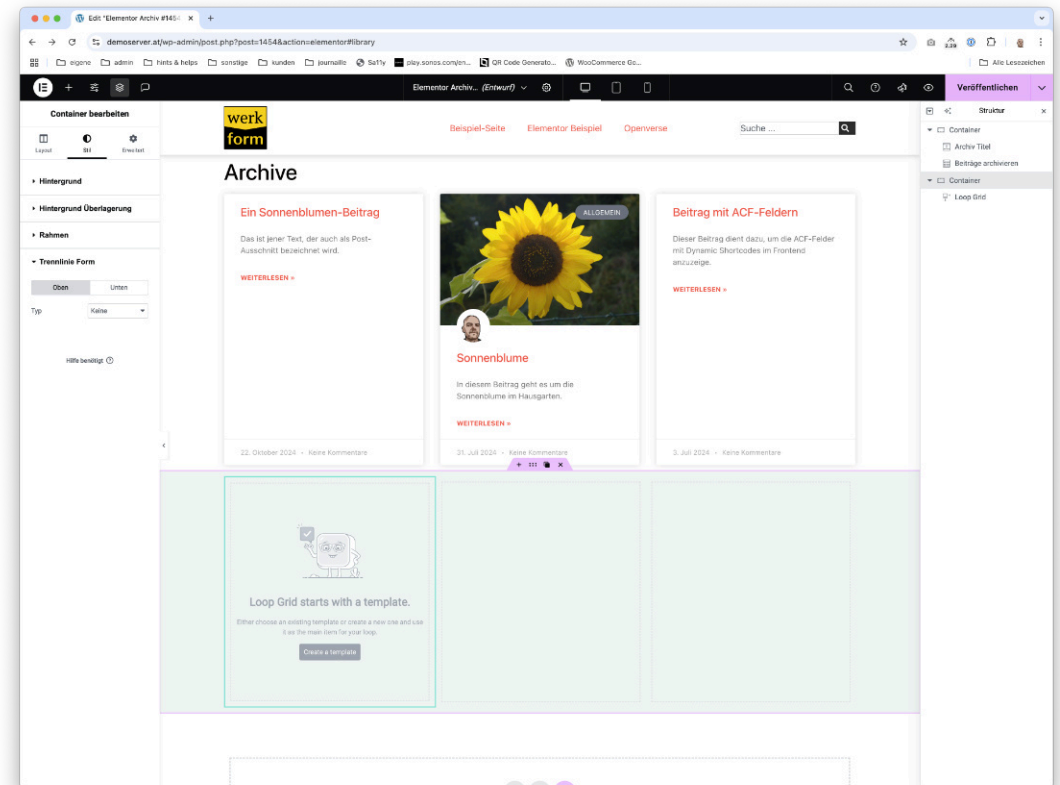
Immer dann, wenn man eine **Liste** von Beiträgen, Produkten, Suchergebnissen, Feeds oder auch eigenen Post-Typen (Events, Jobs, Immobilien, ...) anzeigen will benötigt man ein Archiv. Diese existieren immer, auch dann, wenn man ihnen keine Bedeutung oder Gestaltung zugesteht. Typische Archivseiten sind Blogseiten mit Titel, Beitragsbildern und Textauszügen die dann mit einem *weiter lesen* Link auf die Single Post-Ansicht des Beitrages verweisen.

Archive können als Widgets vorliegen (zB. Loop-Grid, Beiträge), welches man innerhalb eines Containers einbindet (zB. die 3 neuesten Beiträge eines Blogs werden auf der sonst statischen Homepage einer Firma angezeigt). Oder aber man hat eine eigene Seite, die Archive anzeigen (Blogseiten, Produktübersichten, etc.). Auch auf dieser Seite wird dann wieder ein passendes Widget (Loop-Grid, Beiträge (Posts), Beiträge archivieren (Archive Posts), Portfolio) eingesetzt welches die Beiträge oder eigenen Post-Typen auflistet. Das Widget *Loop-Grid* verwendet den Template-Type **Loop** für die Darstellung der einzelnen Beiträge.

Die wesentlichen Unterschiede zwischen den 5 Widgets sind:

Loop-Grid: benötigt Loop-Templates, völlig freie Gestaltung und freie Abfrage. Ergänzend dazu gibt es das Widget **Taxonomy Filter**, das als Kategoriefilter innerhalb des Grids verwendet werden kann.

Loop Carousel: Funktioniert analog dem Loop-Grid Widget, nur dass die



Oben ein Container mit dem Widget Beiträge archivieren. Dieses sorgt dafür, dass automatisch zB. nur die Beiträge der aktuell gewählten Kategorie angezeigt werden. Es gibt 3 Darstellungsmöglichkeiten (Klassisch, Karten, Full Content).

Darunter ein Loop Grid: hier muss der Einzelne Beitrag erst über ein Loop-Template gestaltet werden. Was angezeigt werden soll wird im Tab Abfrage eingestellt (zb. Related für die aktuelle Kategorie).

einzelnen Beiträge nicht in einem Gitter sondern eben in einem Karussell dargestellt werden.

Beiträge: kann überall eingesetzt werden, hat die Möglichkeit die Abfrage zu bestimmen.

Beiträge Archivieren: nur für den Einsatz in Archiv-Templates, keine Abfrageänderung.

Portfolio: kann überall eingesetzt werden, hat Abfrage und Filterbar (veraltet, kann durch Loop-Grid ersetzt werden).

Um ein Archiv zu erstellen wählt man

Dashboard > Elementor > Theme Builder > Neu hinzufügen > Template Typ: Archiv

oder

Panel > Theme Builder > Archiv +

vergibt einen sinnvollen Namen, erstellt einen Container und zieht das Widget Loop-Grid hinein. 3 Bereiche müssen konfiguriert werden: das Layout, die Abfrage und die Paginierung.

Unter **Layout** wird festgelegt, welches Loop-Template zum Einsatz kommen soll. Dieses kann ausgewählt oder aber auch direkt hier angelegt werden. Wichtig ist dem Template einen Namen zu geben, damit man es leichter auswählen kann. Erst nachdem das Loop-Template erstellt wurde kann man die Spalten und die Anzahl der Einträge pro Seite wählen. Hier kann auch eine Masonry-Ansicht aktiviert werden.

Unter **Abfrage** wird die Quelle gewählt, also welche Inhalte in welcher Reihenfolge angezeigt werden sollen. Man kann also nicht nur Beiträge sondern beliebige Inhalte auf diese Art anzeigen lassen.

Die **Paginierung** bietet die Einstellungen an die wirken, wenn mehr Beiträge vorhanden sind als in der Anzahl pro Seite eingetragen wurden. Man kann

The screenshot shows the 'Layout' tab of the Loop Grid configuration panel. It features three tabs: 'Inhalt', 'Stil', and 'Erweitert'. Under the 'Layout' section, there are two dropdown menus: 'Choose template type' set to 'Beiträge' and 'Choose a template' set to 'Loop 4 Archive Searchresult (Template)'. An 'Edit template' button is located below these menus. At the bottom, there are three rows of settings: 'Spalten' with a value of 3, 'Items Per Page' with a value of 6, and 'Masonry' and 'Equal height' both set to 'Aus' (disabled). An 'Apply an alternate template' toggle is also present and disabled.

The screenshot shows the 'Abfrage' (Query) tab of the Loop Grid configuration panel. It includes a 'Quelle' (Source) dropdown with a 'beifügen' (add) button. A dropdown menu is open, showing options: 'Beiträge' (checked), 'Seiten', 'Startseiten', 'Manuelle Auswahl', 'Aktuelle Suchanfrage', and 'Related'. Below this, there are three rows of settings: 'Datum' set to 'Alle', 'Sortieren nach' set to 'Datum', and 'Sortierung' set to 'absteigend'. An 'Ignore Sticky Posts' toggle is set to 'Ja'. A note at the bottom states 'Sticky-posts ordering is visible on frontend only'. An 'Abfrage ID' field is at the bottom.

Im **Layout**-Tab wählt man ein Loop-Template (oder erstellt eines) - erst dann (!!!) erscheinen die Optionen, um die Anzahl der Spalten und die abgefragten Beiträge pro Seite auswählen zu können.

Masonry erzeugt einen Raster mit gleicher Breite und unterschiedlicher Höhe der einzelnen Loop-Bereiche, während Equal height (gleiche Höhe) dafür sorgt, dass in jeder Zeile die einzelnen Loop-Beiträge gleich hoch werden.

Unter **Abfrage** wird ausgewählt was überhaupt abgefragt werden soll und wie die einzelnen Dinge sortiert werden sollen.

Die **Paginierung** erlaubt Seitennummern in Variationen, Load on Click und Infinite Scroll.

The screenshot shows the 'Paginierung' (Pagination) tab of the Loop Grid configuration panel. It features a 'Paginierung' dropdown menu with options: 'Keine', 'Nummern' (checked), 'Voriger/Nächster', 'Nummern + Voriger/Nächster', 'Load on Click', and 'Infinite Scroll'. Below this, there are three rows of settings: 'Seitenlimit' with a bidirectional arrow icon, 'abkürzen' (truncated), and 'Ausrichtung' (alignment) with three icons. A 'Load Type' dropdown is set to 'Page Reload'. An 'Individual Pagination' toggle is set to 'Aus'. A note at the bottom states 'For multiple Loop Widgets on the same page, toggle this on to control the pagination for each individually. Note: It affects the page's URL structure.'

wählen ob man *Blättern* will mit Seitennummern, Vor-Zurück-Links oder einen *Mehr Laden*-Button will (Load on Click) oder *Infinite Scroll*, also Laden durch Scrollen aktivieren will.

Im *Stil*-Tab des Widgets Loop-Grid gibt es nur eine Einstellung: Layout. Hier wird der Abstand zwischen den Spalten und den Zeilen eingestellt.

4.8. Loop

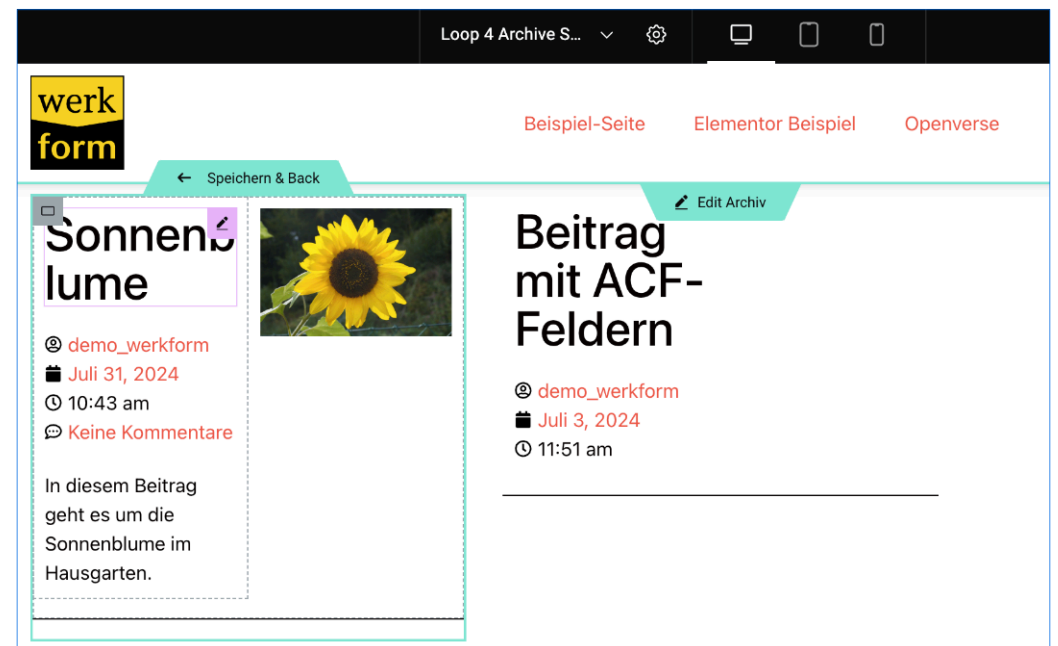
Die in Version 3.8. eingeführten Loop-Templates ermöglichen die freie Gestaltung der einzelnen Items. Diese Templates können innerhalb des Loop-Grids erstellt und bearbeitet werden oder aber über den Theme-Builder - letzteres erlaubt eine sinnvolle Namensgebung, was immer empfohlen wird!

Dafür eignen sich die Widgets die auch für das Single Post Template verwendet werden sehr gut. Zusätzlich darf natürlich auf den Link zum vollständigen Beitrag nicht vergessen werden. Dieser wird über das Dynamische Attribut auf Post-URL eingestellt. Das kann man überall dort machen, wo der Link aktiv sein soll. Also zB. beim angezeigten Beitragstitel oder dem Beitragsbild.

Man kann aber auch das *ganze Loop-Template* mit diesem Link ausstatten. Hier ist aber wichtig zu beachten, dass dann kein Element des Templates ebenfalls einen Link haben darf, dies würde das HTML und damit das Layout zerstören. Eingestellt wird dies in den *Zusätzlichen Optionen* des äussersten Containers des Loop-Templates. Hier wird der **HTML-Tag a (link)** eingestellt und dann der Link über das dynamische Attribut auf Post-URL gestellt.



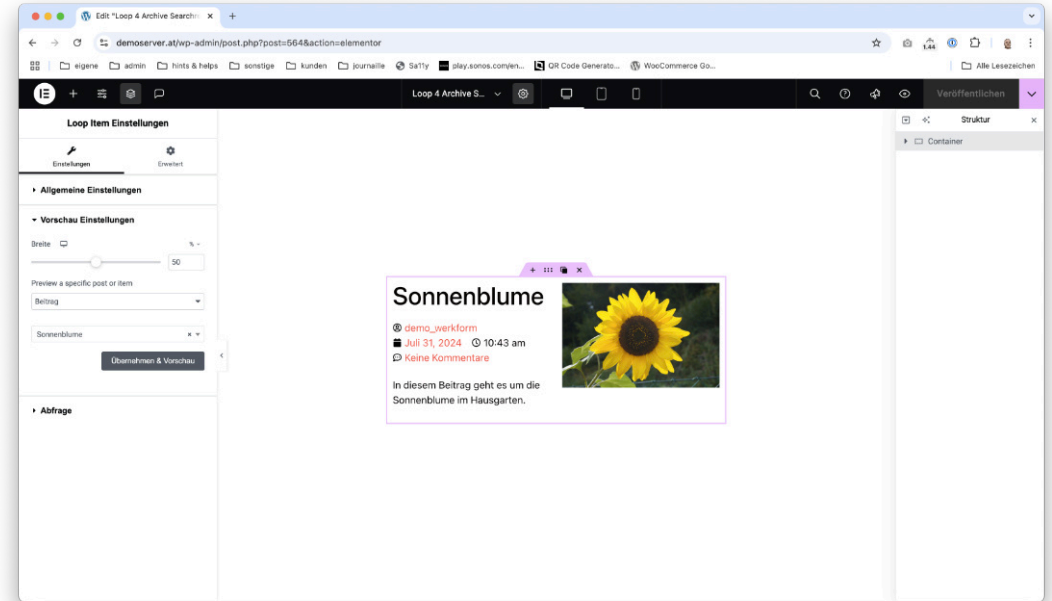
Man kann innerhalb des Grids das Loop-Template bearbeiten (siehe türkisen Reiter) und danach wieder zurück in das Loop-Grid wechseln.



4.9. Suchergebnisse

Ein eigener Template-Bereich sind die Suchergebnisse. Der einzige Unterschied zum oben Besprochenen ist, dass man bei den Bedingungen eben Suchergebnisse auswählt. Bei der Gestaltung kann man das Suchformular einbinden, darin wird der verwendete Suchbegriff automatisch mit den Suchergebnissen angezeigt.

Übrigens kann man das selbe Loop-Template natürlich für Loop Grid und für Suchergebnisse verwenden, so ferne das Design dafür geeignet ist.



Wenn man das Loop-Template über den Theme-Builder bearbeitet empfiehlt es sich eine geeignete Auswahl in den Vorschau-Einstellungen zu treffen: nur wenn man anhand eines typischen Beispiels sieht, was man gestaltet, werden die Einstellungen gelingen.

“

Die Zusammenarbeit mit Harry war durchwegs angenehm und sehr professionell. Für mein Projekt zur Erstellung einer neuen Homepage hat er das Grundgerüst sorgfältig entwickelt und dabei mit kreativen Lösungsansätzen überzeugt. In den regelmäßigen Face-to-Face-Video-Calls konnten wir alle Ideen ausführlich diskutieren und gemeinsam weiterentwickeln. Mit den Ergebnissen bin ich äußerst zufrieden und empfehle Harry uneingeschränkt weiter!

Max W.
Kellerfabrik, Wien
www.kellerfabrik.at



werk form

Erstgespräch buchen



Anzeige

5. Container

Die in Gutenberg vorhandenen Gruppen, Reihen, Stapel und Spalten entsprechen technisch und logisch den Containern in Elementor.

Widgets werden per Drag & Drop auf die zu bearbeitende Seite (Beitrag, Template) gezogen. Dabei ist wichtig zu verstehen, dass Widgets **immer** in Containern (früher Abschnitten und Spalten) organisiert werden. Sehr hilfreich ist die *Strukturpalette* (früher Navigator), ein kleines Popup-Tool welches bei der Organisation von Widgets und Containern helfen kann.

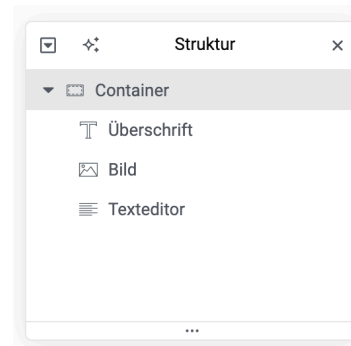
Arbeiten mit Containern

Flexbox oder Grid? Neben der Voreinstellung *Flexbox* kann man in der ersten Option **Container Layout** auch *Grid* auswählen. Dahinter verbergen sich zwei unterschiedliche Techniken, die auch kombiniert zum Einsatz kommen können. Bei Flexbox kann man eine Dimension bestimmen (Spalte oder Zeile), während man bei Grid beides definieren kann. Grob gesagt eignet sich Flexbox dann besser, wenn die Inhalte bekannt sind. Grid ist eher dann erste Wahl, wenn man layoutbasiert arbeitet und die Inhalte (automatisch) einfließen.

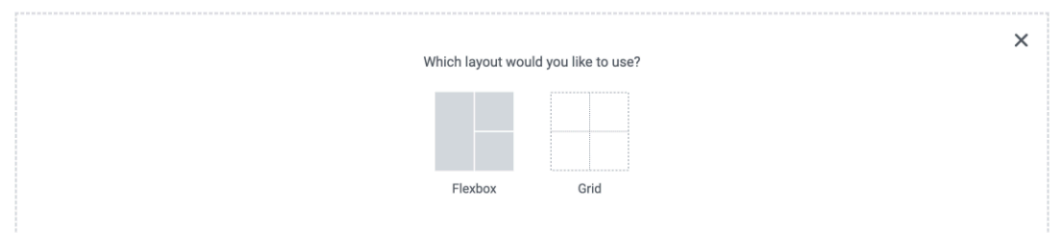
Die Idee der *Flexbox* entstand aus dem Bedarf des responsive Designs. Es geht darum, in welcher Weise Dinge (also Widgets, Elemente) in einem bestimmten Bereich (dem Container) angeordnet werden und welche Abstände sie zu- und voneinander haben und wie sie zueinander angeordnet werden (Achsen).

Container können ineinander verschachtelt werden, die gestalterischen Möglichkeiten sind also größer als beim veralteten Konzept des Inneren Abschnittes. Es gibt keine Limitierung, wie viele Container ineinander stecken können. Trotzdem ist es – wie immer – ratsam so wenige Container wie möglich und so viele wie notwendig zu verwenden.

Beim Anlegen einer Struktur werden die Möglichkeiten sichtbar: man kann einfache Boxen aussuchen oder auch aus komplexere Strukturen vor-auswählen. Je nachdem, welche Auswahl man getroffen hat, bekommt man eine ver-



Die Strukturpalette hilft bei der Organisation, die Elemente lassen sich umbenennen, verschieben, löschen und einiges mehr (rechts-Klick bzw. ctl-Klick). Das Fenster kann frei schweben oder rechts angeheftet werden.



Flexbox Container sind die erste Wahl, wenn man die Inhalte kennt, also ein Inhaltsbasiertes Layout gestalten möchte. Grid kommt zum Einsatz, wenn man viele Inhalte gleich darstellen will, die man ev. noch nicht kennt.

schachtelte Containerlandschaft angelegt. Jeder dieser Container kann einzeln verändert und gestaltet werden.

5.1. Flexbox Container

Folgende Einstellungen können für den Container selber getroffen werden:

Inhaltsbreite: *Boxed* oder *Volle Breite*. Hier wird festgelegt, ob der Inhalt des Containers die in den Einstellungen (siehe Kapitel 2.2 Layout) festgelegte Inhaltsbreite wirksam sein soll oder die Volle Breite des Bildschirms.

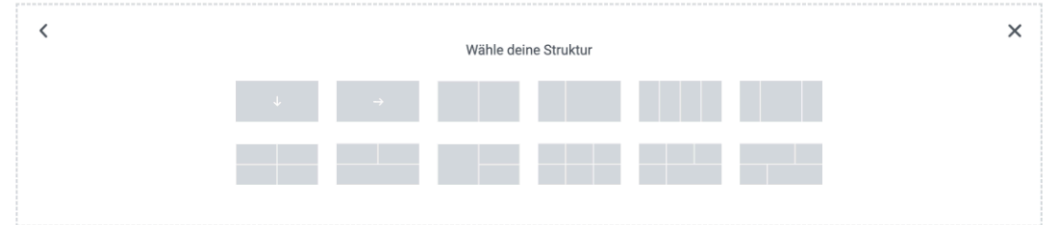
Breite (Width): Responsive und in den Einheiten PX, %, EM, REM, VW und Custom (selbstdefiniert) kann die horizontale Ausdehnung des Containers angegeben werden. Wird keine Einstellung getroffen wird die in den Website-Einstellungen getroffene Inhaltsbreite verwendet.

Min. Höhe (Min Height): in PX, EM, REM, VH und Custom (Bleistift-Icon) kann die Mindesthöhe angegeben werden. Wer die volle Bildschirmhöhe will muss 100 VH angeben. Diese Höheneinstellung wird dann wirksam, wenn die Elemente innerhalb des Containers niedriger als die hier eingestellte Mindesthöhe ist. Sonst wird der Container so hoch, wie es sein Inhalt notwendig macht.

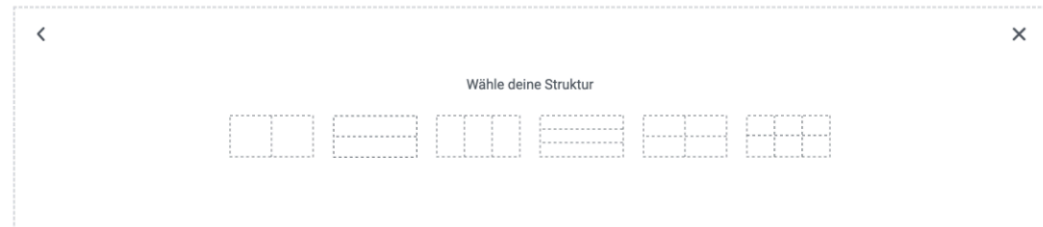
Diese Einstellungen betreffen die Anordnung der Elemente im Container:

Richtung (Direction): gibt an, wie die darin befindlichen Elemente angeordnet werden. *Vorgabe, Reihe* (von links nach rechts), *Spalte* (von oben nach unten), *Umgekehrte Reihe* (von rechts nach links) und *Umgekehrte Spalte* (von unten nach oben) stehen zur Auswahl.

Verteilung (Justify Content): Analog zur Ausrichtung kann hier *Flex Start*, *Center* und *Flex End* verteilt werden. Je nachdem ob man eine Reihe (horizontale Richtung) oder Spalte (vertikale Richtung) gewählt hat bedeutet das eben links oder oben, horizontal oder vertikal zentriert und rechts oder unten. Dazu kommen noch die 3 Verteilungsvarianten:



Die Vorauswahl bei einem Flexbox-Container erlaubt schon recht komplexe Anordnungen, alle diese Layouts können jederzeit verändert werden.



Grid-Strukturen sind einfacher, können dafür besser große Inhaltsmengen aufnehmen und diese dann auch gleichförmig darstellen.

Zwischenraum (Space Between), Abstand im Umkreis (Space Around) und Gleichmässiger Abstand (Space Evenly). Bei diesen 3 Varianten wird der freie Platz zwischen den Elementen wie benannt verteilt.

Ausrichtung (Align Items): gibt an wie die Elemente innerhalb des Containers ausgerichtet werden. Man kann keine Angabe machen oder auswählen: *Flex Start* (oben oder links), *Center* (je ob Reihe oder Spalte horizontal oder vertikal zentriert), *Flex End* (rechts oder unten) oder *Stretch* (ausgerichtet über die volle Breite oder Höhe).

Lücken (Spacing): hier wird der Widgetabstand der Elemente zueinander eingestellt. beim Abstand 0 stossen diese direkt aneinander. Diese Abstände können für die Spalten und Reihen auch getrennt eingegeben werden.

Umbruch (Wrap): hier kann man festlegen, ob die einzelnen Widgets hintereinander ähnlich wie Text umbrechen soll oder ob kein Umbruch erfolgen soll.

In den *Zusätzlichen Optionen* kann weiters definiert werden:

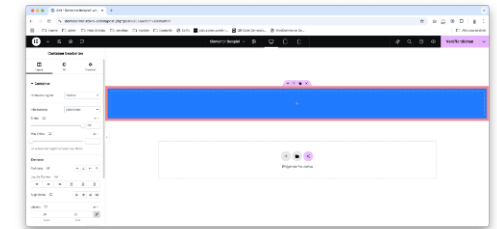
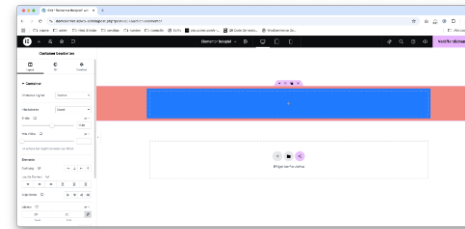
Overflow: soll im Falle eines überlaufenden Inhalt das Elemente gestreckt werden (default) oder der überlaufende Inhalt versteckt (hidden)?

HTML-Tag: ein Tag kann gewählt werden, voreingestellt ist *div*. Wichtig: wenn man den ganzen Container als Link setzen will (also den Tag *a* wählt) darf innerhalb des Containers kein weiterer Link gesetzt werden, da dieser die HTML-Struktur und damit auch das Layout zerstören würde.

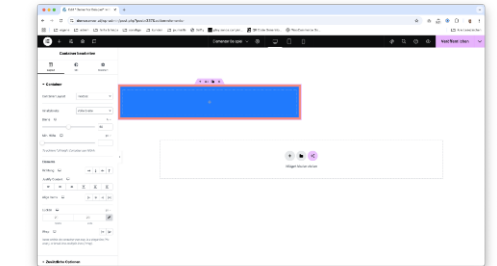
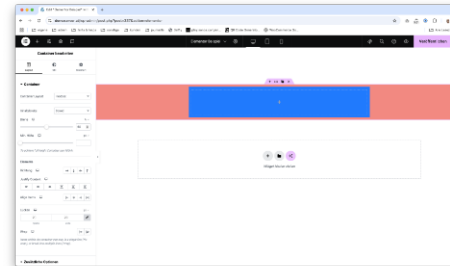
Diese Einstellungen im Detail:

Richtung

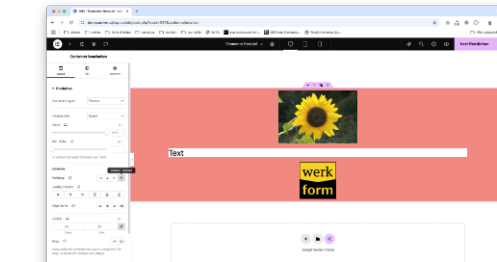
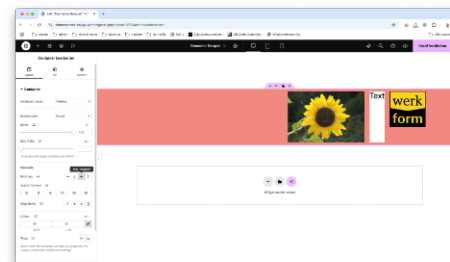
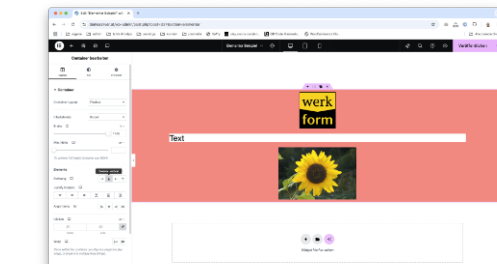
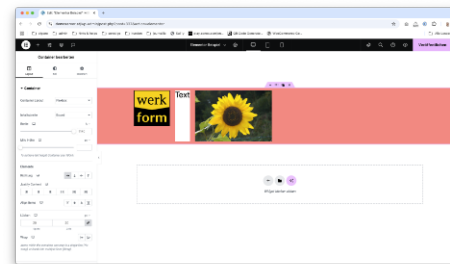
Die erste und wichtigste Einstellung ist die Direction (Richtung), in der die Inhalte eines Containers angeordnet werden können. Die Richtung kann für verschiedene Responsive Modi unterschiedlich eingestellt werden. Man kann da-



Der blaue Bereich zeigt die Breite: links Boxed (=Inhaltsbreite), rechts Volle Breite.



Die Breite (das Beispiel zeigt 50%) wirkt sich bei Boxed (links) anders aus als bei 50% der Volle Breite.



Die vier möglichen Richtungen eines Containers. Bei den vertikalen Richtungen wird der Container höher, horizontal ist er so hoch wie der höchste Inhalt.

mit für den Desktop von Links und am Mobilgerät von Oben einstellen und damit sehr einfach eine passende Anpassung an den Gerätetyp erreichen.

Verteilung (Justify Content)

Nun kann man die einzelnen Elemente innerhalb eines Containers verteilen. Das Beispiel zeigt die 6 Möglichkeiten der Verteilung innerhalb einer Reihe, wobei keine Ausrichtung (Alignment) ausgewählt wurde, die Elemente also an der Oberkante ausgerichtet werden (=default).

Eine Verteilung kann man auch einstellen, wenn man als Richtung *vertikal* gewählt hat. Diese wird aber nur sichtbar, wenn die Höhe des Containers höher als die Summe dessen Inhalte ist (Einstellung *Minimale Höhe*). Das ist übrigens auch bei der Breite so, nur da sorgt normalerweise die Voreinstellung dafür, dass die Inhaltsbreite größer als der darin platzierte Inhalt ist.

Der Unterschied zwischen *Between*, *Around* und *Evenly* ist nicht immer leicht zu sehen aber leicht erklärt:

In einem ersten Schritt addiert der Browser den Platz den die einzelnen Elemente benötigen. Im zweiten Schritt zieht der Browser diese Breite von der zur Verfügung stehenden Fensterbreite ab. Ist die Positiv - also die Elemente in Summe schmaler als die zur Verfügung stehende Breite - bleibt ein gewisser Rest. Und dieser kann auf 3 Arten verteilt werden:

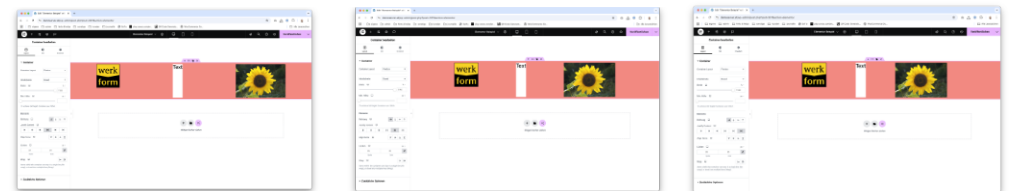
Abstand dazwischen (Space Between) bedeutet, dass der Abstand zwischen den Elemente, nicht aber links und rechts vom ersten und letzten Element, eingehalten wird.

Abstand herum (Space Around) heisst, das jedes Element gleich viel Abstand zugewiesen bekommt und dieser um das Element herum fließt. Das führt dazu, das links vom ersten und rechts vom letzten Element eine Abstandsbreite bleibt während zwischen den Elementen immer 2 Abstandsbreiten wirken.

Abstand gleich (Space Evenly) wiederum bedeutet, dass alle Abstände, auch



Start, Center und End am Beispiel eines von Links nach Rechts gerichteten Containers: die Justify-Einstellung (Verteilung) bestimmt, wo die Elemente im Container stehen.



Zwischenraum (links) bedeutet, dass das erste Element ganz am Anfang und das letzte ganz zum Ende gesetzt wird, der Raum dazwischen wird aufgeteilt. Abstand im Umkreis (Mitte) bedeutet, dass um alle Elemente ein Abstand gehalten wird, während bei Gleichmässiger Abstand (rechts) die Abstände zwischen den Elementen und dem Rand genau gleich aufgeteilt werden.

links und rechts am Rand, genau gleich eingehalten werden.
Wenn eine Containerausrichtung senkrecht ist gilt das selbe einfach um 90° gedreht.

Ausrichtung (Align Items)

Der Dritte Parameter ist die Ausrichtung im Container. Diese kann am Start, in der Mitte und vom Ende her die Elemente anordnen. Ausserdem können - so ferne der Container groß genug ist, die Elemente auch maximal gedehnt (Stretch) werden, ein Beispiel dafür ist das Titel-Widget mit Hintergrundfarbe rechts.

Es gibt also 4 mögliche Richtungen, darin kann man 4 verschiedene Ausrichtungen vornehmen (inkl. dehnen/stretch) und ausserdem kann man 6 unterschiedliche Verteilungen vornehmen: $4 \times 4 \times 6 = 96$ Möglichkeiten.

Jetzt wird klar, warum dieses Modell Flexbox genannt wurde.

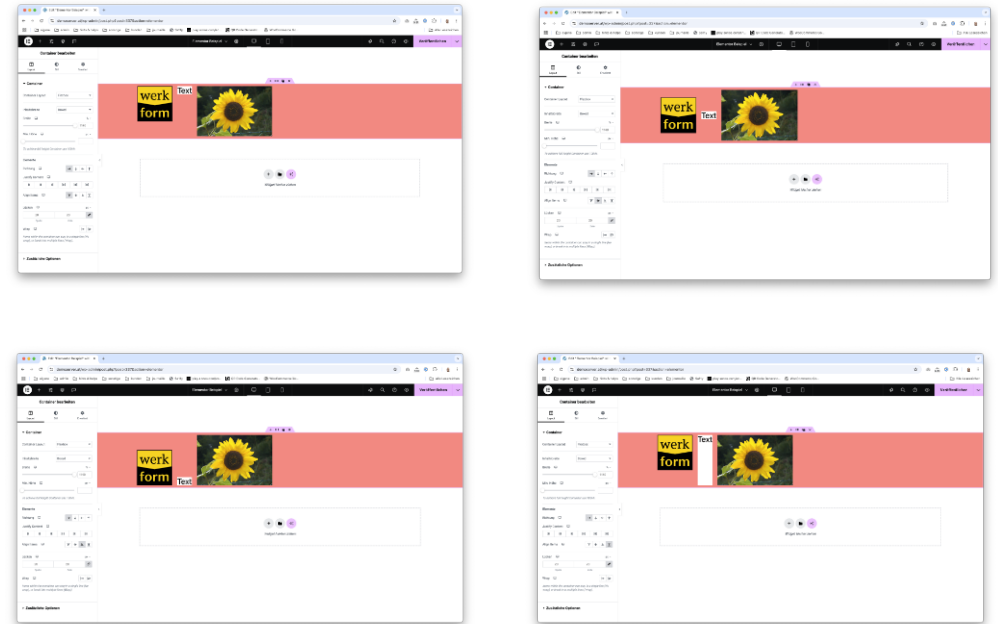
Lücken (Abstände)

Wirklich brauchbar wird das Ganze aber erst, wenn man den Abstand der Widgets zu einander bestimmen kann. Dazu wurde der Parameter *Lücken* (Spacing) eingeführt (früher *Abstand*) den man in jedem einzelnen Container (responsive!) ändern kann. Dieses Spacing kann für Spalten und Zeilen auch getrennt definiert werden.

Wie bei allen Widgets kann auch beim Container der Aussenabstand (Margin) und der Innenabstand unter

... > Erweitert > Layout > Aussenabstand/Innenabstand

eingegeben werden. Mögliche Einheiten sind %, EM, REM, VW und wiederum Custom. Für jeden Container einzeln kann man den Abstand der Elemente zu einander einstellen.



Die Ausrichtung (Align Items) im Container ist mit Beginn (oben links), Zentriert (oben rechts) und Ende (unten links) (also je nach Richtung des Containers Oben, Mitte, Unten oder eben anders gerichtet) noch einfach zu verstehen. Die vierte Option Dehnen (Stretch, unten rechts) hingegen funktioniert nur dann, wenn das Element auch dehnbare ist: Im Beispiel ist das der weisse Bereich um den Text: die ersten 3 Optionen setzen den Text, die vierte sorgt dafür, dass der Text auf die volle Containerhöhe gedehnt wird.



Dir gefällt dieses Skriptum?

Kaffeespende hier!

jetzt spenden

Spende Skriptum 3.0



Anzeige

Wichtig ist auch die Einstellung Umbruch / Wrap die dafür verantwortlich ist, dass mehrere nacheinander angeordnete Elemente wie Text in Zeilen umbrechen oder aber nebeneinander stehen bleiben soll und ggf. ein Scrollbar erscheint.

5.2 Einstellungen für Elemente

Innerhalb eines Containers kann jedes Element eigene Einstellungen haben die sich auf die Darstellung im Container auswirken. Damit werden die zuvor für den Container getroffenen Einstellungen überschrieben. Diese sind unter

... > Erweitert > Layout > Align Self | Order | Size

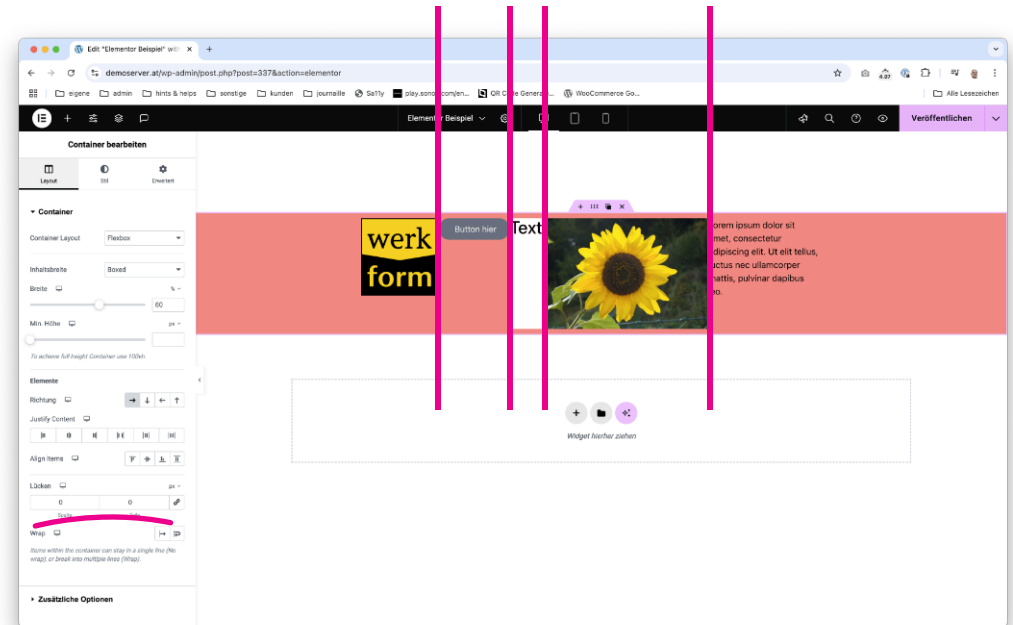
zu finden.

Align Self erlaubt es einem einzelnen Element anders als die anderen Elemente innerhalb eines Containers ausgerichtet zu werden. Also zB *Center* oder *End* während die anderen Elemente *Start* als Ausrichtung haben.

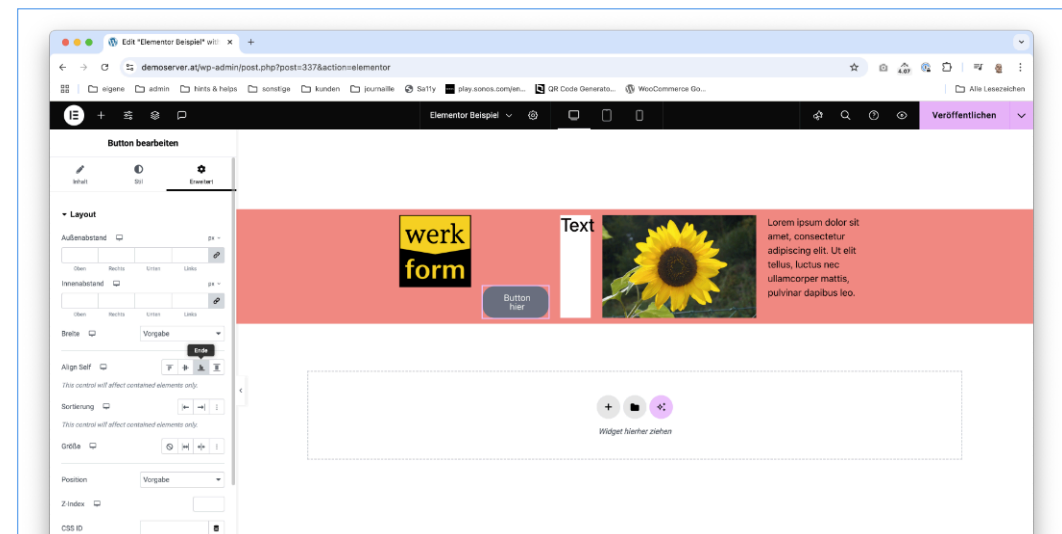
Order ermöglicht die Position des einzelnen Elements innerhalb des Containers zu ändern nur über CSS ohne HTML-Änderung.

Size erlaubt Einstellungen die bei komplexeren Container-Konstruktionen von Nöten sind, die Flexbox ermöglicht.

Da man Container beliebig ineinander platzieren kann sind durch diese vielen Einstellungsmöglichkeiten wohl alle sinnvollen Layouts darstellbar.



Die Lücken (früher Abstände) regeln die Abstände zwischen den Widgets. Der Abstand zum Containerrand hingegen wird über die Padding (Innenabstand)-Einstellungen definiert. Das Beispiel hier zeigt einen Spaltenabstand von 0.



Um ein einzelnes Element innerhalb eines Containers aus der Ordnung zu lösen gibt es die Align-Self Option. Das Beispiel zeigt, dass der Button unten ausgerichtet wurde, während der Container eine obere Ausrichtung verwendet.

5.3. Grid Container

Auch nachdem man schon Elemente platziert hat kann die Art des Containers geändert werden (Container-Layout). Die Auswirkungen sieht man sofort: in einem Grid-Container sind alle Felder gleich groß, und zwar so groß, wie das größte einzelne Feld im ganzen Grid.

Eine Eigenart des Grids ist es, dass es immer von vorne weg aufgefüllt wird, man kann sich also nicht wie bei einer Flexbox-Struktur aussuchen, welches Feld man nutzt. Ausserdem sorgen Grids dafür, dass - unabhängig von der eingestellten Anzahl an Spalten und Zeilen - weitere Felder dem Grid hinzugefügt werden, sollten mehr Inhalte als eingestellte Felder vorliegen. **Grid Outline** ist ein Schalter der im Editor die einzelnen Felder sichtbar macht. Das ist eine Arbeitshilfe, keine Designeinstellung.

Bei Grids taucht eine neue Einheit für **Spalten** und **Zeilen** auf: fr. fr steht für *fraction*, also für Anteil. Wenn man den Wert 3 bei Spalten einträgt bedeutet das, dass man eine Einteilung 1fr 1fr 1fr erzeugt. Jede Spalte ist damit gleich breit.

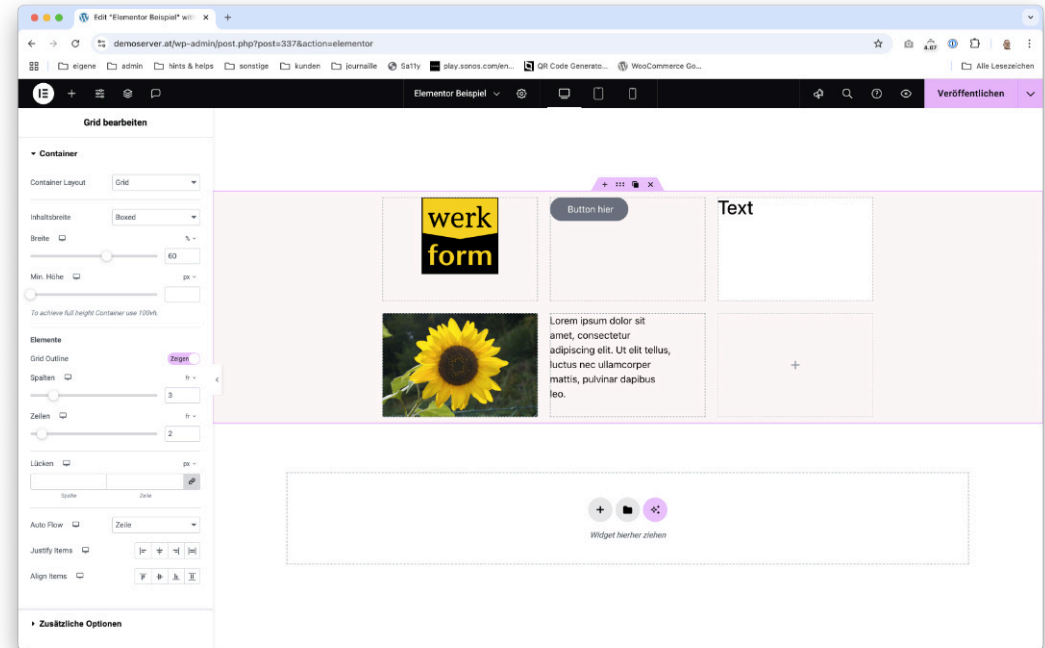
Wenn man in die *Custom Units* wechselt (über das kleine Popup wo fr steht) kann man auch eigene Werte eingeben. So erzeugt eine Eingabe von 1fr 2fr 1fr 3 Spalten, wobei die mittlere doppelt so breit wie die beiden äusseren Spalten ist.

Analog dazu kann man das auch mit den Zeilen machen.

Lücken verhält sich wie bei den Flexbox-Containern, man kann unterschiedliche Abstände für Zeilen und Spalten definieren

Auto Flow bietet eine Auswahl, ob zusätzliche Inhalte automatisch als neue Zeilen oder neue Spalten angelegt werden sollen.

Justify Items und **Align Items** verhalten sich wie erwartet: man kann die Inhalte der Felder jeweils links, mittig, rechts und oben, zentriert, unten positionieren bzw. auch wieder dehnen.



Bei einem Grid-Container werden automatisch gleich große Felder angelegt, so groß wie der größte Inhalt eines einzelnen Feldes. Grids werden immer von vorne nach hinten befüllt, leere Felder sind nicht möglich. Gibt es mehr Inhalte als Felder werden automatisch weitere Felder eingefügt.



Wenn man die Einheiten bearbeitet kann man auch unterschiedliche Spaltenbreiten bzw. Zeilenhöhen umsetzen.

5.4. Der Container als Designelement

Beide Containerarten sind also Layoutelemente die helfen eine Seitenstruktur zu erstellen. Allerdings sind Container auch als Designelemente nutzbar, und zwar lassen sich folgende Dinge im Stil-Tab einstellen:

- Hintergrund
- Hintergrund Überlagerung
- Rahmen
- Trennlinie Form

Hintergrund

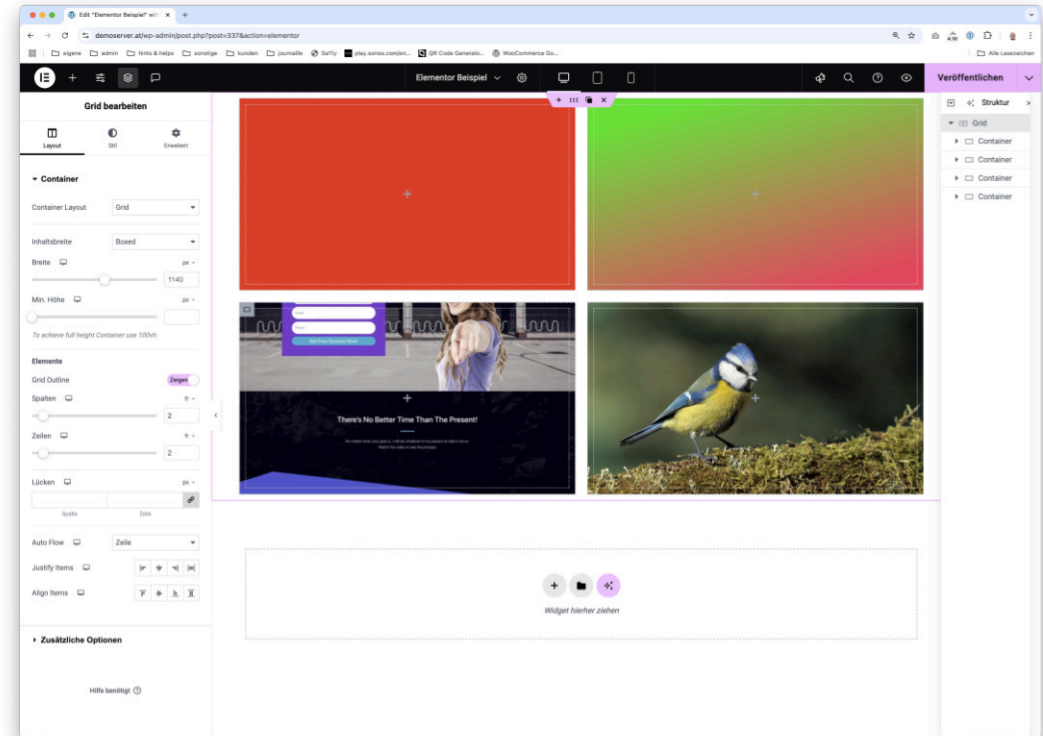
Man kann aus 4 verschiedenen *Normal* Background Types (Hintergrundarten) wählen und zusätzlich 2 Arten bei *Hover* einstellen.

Klassisch. Hier kann man eine *Farbe* und/oder ein *Hintergrundbild* einstellen. Farbe ist wohl einfach zu verstehen, wenn man ein Hintergrundbild aus der Mediathek auswählt ergeben sich verschiedene weitere Möglichkeiten: *Position* ermöglicht die Ausrichtung des Hintergrundbildes innerhalb der Containerfläche. *Anhang* erlaubt das fixieren des Bildes an der Position, sobald weitergescrollt wird.

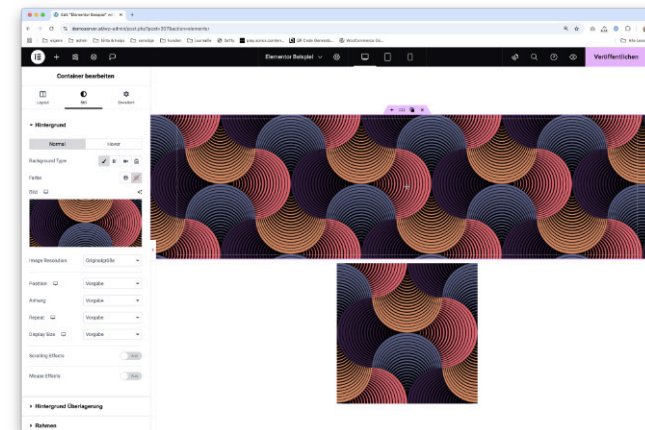
Repeat (Wiederholung) regelt ob ein Bild (sobald es kleiner als der Container ist) erneut dargestellt werden soll. *Display Size* (Darstellungsgröße) ermöglicht unter anderem, die vollständige Anzeige des Bildes innerhalb des Containers zu erzwingen (Beinhalten, Contain) oder aber die komplette Abdeckung der Containerfläche zu erreichen (Ausfüllen, Cover).

Farbverlauf. Erlaubt die Definition 2er Farben und erzeugt einen Übergang dazwischen. Start und Endfarbe können eine *Position* haben (zB: „starte den Verlauf nach 10 % der ersten Farbe“), eine Art (Linear, Radial) und im Fall eines Linearverlaufes einen *Winkel* (damit der Verlauf zB. von links oben nach rechts unten geht) und im Fall eines Radialverlaufes eine Position.

Video. Erlaubt das Einstellen eines Videohintergrundes für den Container. Neben der Quelle des Videos kann man einige sinnvolle Optionen einstellen:



Hier wurde ein 2 x 2 Grid-Container verwendet, in jedem Feld ist ein Flexbox-Container mit unterschiedlichem Hintergrund untergebracht: Farbe (links oben), Verlauf (rechts oben), Video (links unten) und Slideshow (rechts unten).



Ein Bild als Hintergrund mit automatischer Wiederholung. Darunter das einzelne Bild.

Start- und Endzeit, einmaliges Abspielen, privater Modus und auch ein Fall-back-Bild.

Slideshow. Damit kann man mehrere Bilder hintereinander als Hintergrund eines Containers anzeigen lassen, verschiedene notwendige Optionen (*Endlosschleife*, *Dauer* der Anzeige eines Bildes, *Übergangsanimation* zwischen den Bildern, *Dauer* des Übergangs, *Hintergrundgröße* und *Hintergrundposition* können definiert werden. Ausserdem besteht hier die Möglichkeit *Lazyload* zu aktivieren sowie einen *Ken-Burns-Effekt* einzuschalten (der sonst in den Elementor-Animationen fehlt).

Hintergrund Überlagerung

Diese Einstellungen erlauben es den zuvor in Hintergrund eingestellten Dinge mit Farben oder einem Verlauf zu überlagern. Ein Beispiel für eine sinnvolle Anwendung wäre ein Video oder eine Slideshow die durch abdecken einer halb durchscheinenden Farbe „entschärft“ werden sollen.

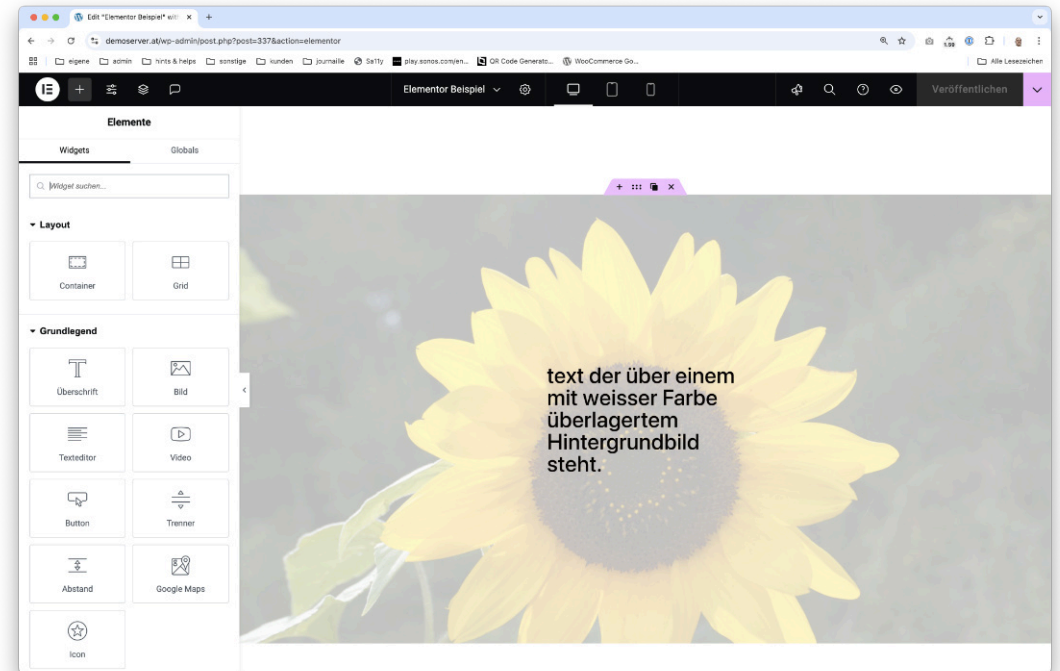
Auch hier gibt es *Hover*-Einstellungen die man sogar mit CSS-Filtern wie *Verwischen*, *Sättigung* etc. ausstatten kann.

Rahmen

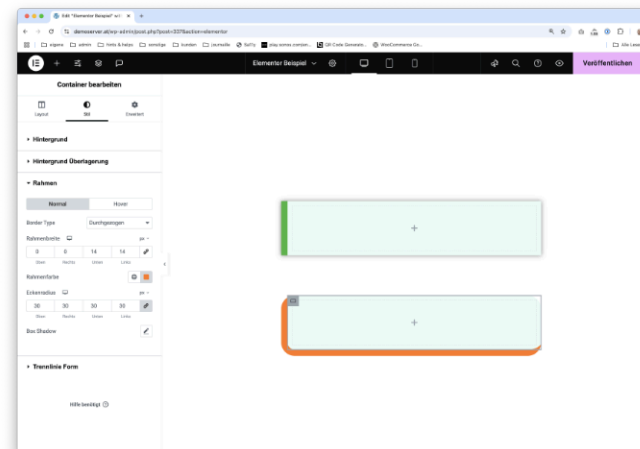
Wie jedes andere Widget auch können auch Container Rahmeneinstellungen bekommen. (Grundeinstellung ist kein Rahmen). Ein Rahmen benötigt immer 3 Definitionen: die *Art* des Rahmens (meistens *solid*/*durchgezogen*), die *Rahmenbreite* und die *Rahmenfarbe*.

Rahmen können für alle 4 Seiten eines Containers gleich oder unterschiedlich definiert werden, wobei es in Elementor ohne CSS nicht möglich ist einem Container unterschiedliche Rahmenfarben für unterschiedliche Seiten zuzuweisen.

Weiters kann ein *Eckradius* definiert werden, wobei man auch hier unterschiedliche Abrundungen der Ecken für jeder einzelne Ecke eingeben kann.



Das Bild einer Sonnenblume wurde in diesem Container als Hintergrundbild ausgewählt, dazu wurde die Option: *Display Size: Ausfüllen* gesetzt. Danach wurde bei *Hintergrund Überlagerung* die Farbe *Weiss* gesetzt mit einer *Deckkraft* von 73%. Daher ist der Text, der in der Mitte des Containers steht, halbwegs gut lesbar.



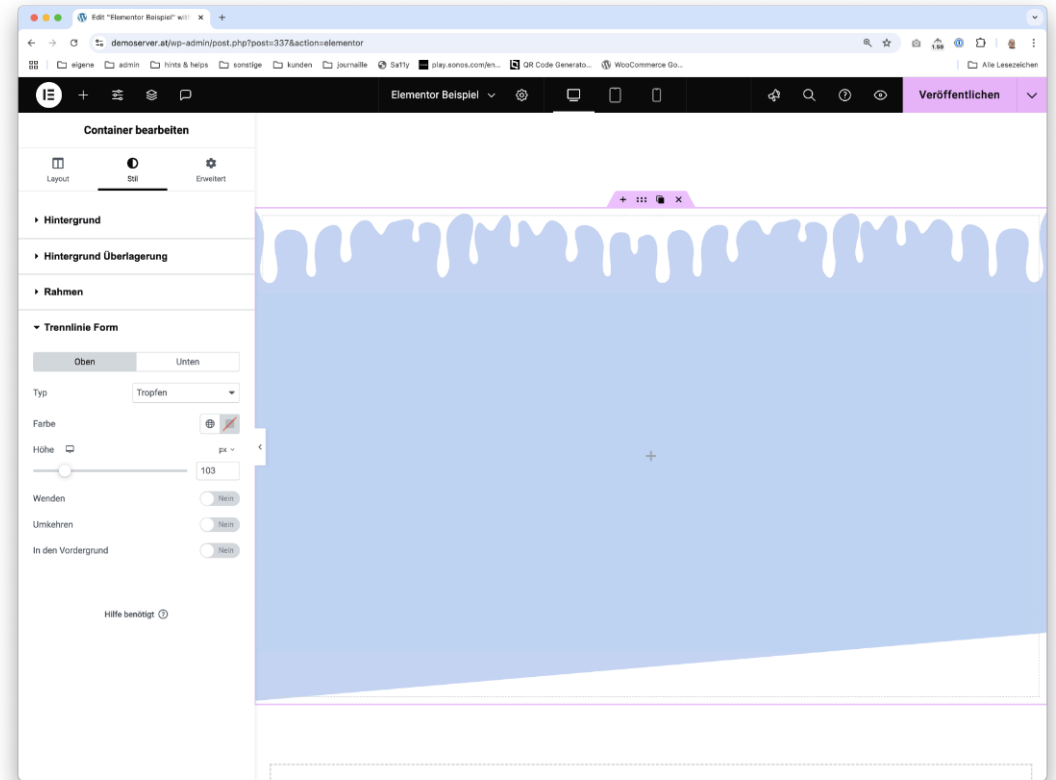
2 Rahmenbeispiele:
oben ein Rahmen nur links mit aktiviertem *Box Shadow*.
Darunter ein Rahmen: *Durchgezogen*, *Orange*, 14 px *Breite* nur links und unten mit einem *Eckradius* von 30 px.

Box Shadow erlaubt das Erzeugen eines Schlagschattens, und zwar auch dann, wenn keine andere Rahmenangabe gesetzt wurde.

Rahmen haben ebenfalls eine *Hover*-Einstellung, hier kann auch noch eine Übergangsdauer eingestellt werden ab wann die sichtbar sein sollen.

Trennlinie Form

Dieses Feature kann oben und/oder unten an einem Container eine SVG-Grafik als Trenner einsetzen. Man kann aus einer Liste von 18 Grafiken auswählen. Diese können verschiedene Farben und Formen annehmen, gespiegelt und in den Vordergrund gesetzt werden.



Mit dem Feature Trennlinie Form lassen sich Tropfeneffekte (oben) oder auch einfach eine Neigung (unten) realisieren. Diese Lösungen benötigen nur wenig Speicher und Rechenzeit.

”

Besonders schätze ich Harrys Fähigkeit, Kundenanforderungen intuitiv zu erfassen, sein breitgefächertes Fachwissen äußerst verständlich zu vermitteln, flexibel Lösungsvorschläge anzubieten und neue Impulse zu setzen. Die Zusammenarbeit ist absolut verlässlich und unkompliziert.



Anja H.
EUROfusion
Garching bei München

werk form

Erstgespräch buchen



Anzeige

6. Widgets

Die Umsetzung des Webdesigns funktioniert im Wesentlichen durch Drag & Drop der Widgets in die erstellten Container. Die Widgets erstellen Elemente, die funktional mit den Blöcken des Gutenberg-Editors von WordPress vergleichbar sind.

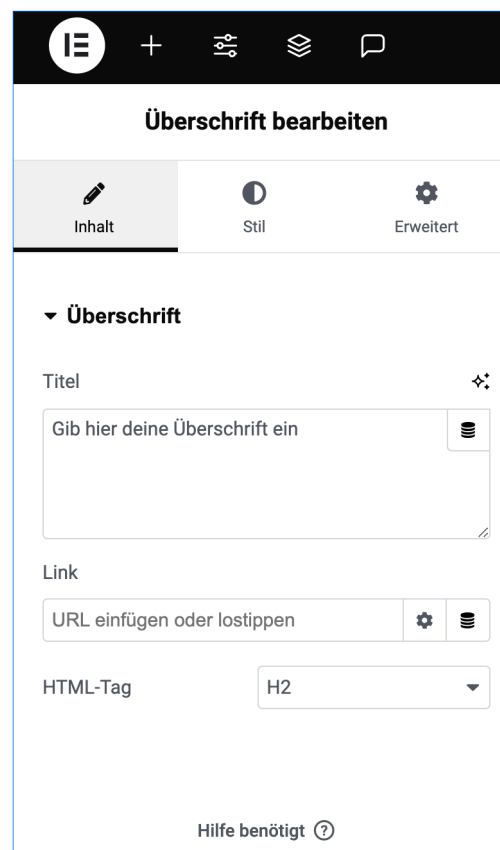
Viele Drittanbieter überbieten einander in der Anzahl der Widgets, die sie in ihre Plugins packen. Die Frage ist immer was man benötigt, und ob eine Designänderung über ein eigenes Widget oder über Stil-Optionen erledigt werden kann. Ein Vergleich ist schwierig. Die praktische Erfahrung sagt wie so oft: weniger ist mehr, einfacher ist besser. In der täglichen Arbeit kommt man oft mit recht wenigen Widgets aus.

Auf der Elementor-Homepage gibt es Beschreibungen und Videos zu jedem einzelnen Widget. In diesem Skriptum wird ein Überblick gegeben.

Für jedes Widget kann man in den 3 Tabs *Inhalt/Stil/Erweitert* Einstellungen treffen. Diese sind abhängig vom jeweiligen Widget unterschiedlich. Die Optionen unter **Erweitert** sind immer die selben. Sie behandeln den Widget-Container, dh. eine HTML-Konstruktion die um das eigentliche Widget angelegt wird. Daher sind hier auch die so wesentlichen Dinge wie Margin (Aussenabstand), Padding (Innenabstand) und Rahmen (Border) zu finden. Ausserdem sind hier die diversen CSS-Optionen (von Bewegungseffekten, Positionierung bis hin zum eigenen CSS) und damit auch die Responsive-Sichtbarkeits-Optionen für das Widget einzustellen.

6.1. Erweiterte Funktionen

Jedes Widget bietet im Bereich *Erweitert* Optionen an, die sich auf Aussehen und Verhalten auswirken. Für die sinnvolle Anwendung sind Grundlagen-Kenntnisse unbedingt notwendig. Diese Optionen sind unabhängig vom Widget selber und gelten daher für alle Widgets gleich, egal ob Bild, Text oder kompliziertere Struktur.



Jedes Widget hat 3 Tabs für die Konfiguration: Inhalt, Stil und Erweitert. Während die ersten beiden abhängig von der Art des Widgets sind bleiben die Möglichkeiten unter Erweitert bei allen Widgets gleich.

Je nach den Bestandteilen eines Widgets können die Tabs Inhalt und Stil zusätzliche Abschnitte für die unterschiedlichen Teile des Widgets haben.

Layout

Im ersten Tab sind die wichtigen Eingabefelder Außenabstand (Margin) und Innenabstand (Padding). Diese können verknüpft sein (= der selbe Wert auf allen 4 Seiten) oder einzeln verändert werden. Die möglichen Einheiten sind px, %, em, rem und vw. Zusätzlich lässt sich (Bleistift-Icon) auch ein eigener Wert eintragen, auch CSS-Berechnungen wie zB: `calc(3vh - 20px)`.

Darunter die Einstellung für die Breite. Diese kann oft bei dem Wert *Vorgabe* belassen werden, hier kann man aber auch auf *Volle Breite* (100%) oder *Inline* (auto) wechseln. Ausserdem kann hier auch eine *Individuelle Breite* angegeben werden.

Der nächste Abschnitt wurde in Kapitel 5. beschrieben: *Align Self*, *Sortierung* und *Größe* haben Auswirkungen den Container betreffend.

Im letzten Abschnitt des Tabs findet man Position, Z-Index, CSS ID und CSS-Klassen.

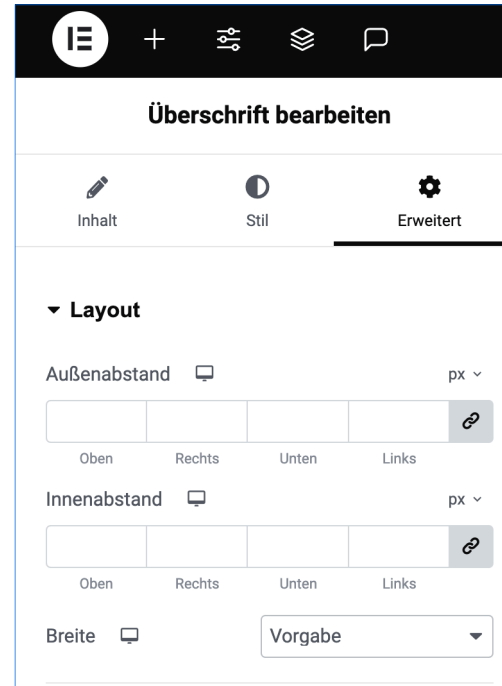
Position

Die Position kennt 2 grundsätzlich verschiedene Möglichkeiten (neben *Vorgabe*, was keine Positionierung bedeutet):

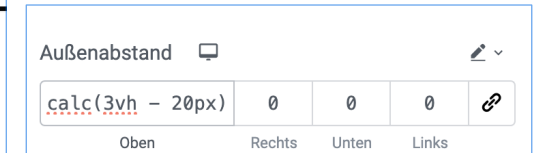
Absolut positioniert gemäß den Angaben (default ist immer links oben) ein Element in der darüber befindlichen Layout-Struktur (dem Container in dem das Widget steckt). Dh. je nachdem wo dieser Container ist wird dann das Element absolut zu dessen Nullpunkt angezeigt.

Nachdem jeder Container immer ein Rechteck ist kann man ausgehend von der *Anordnung* (Horizontal/Vertikal) und dem *Versatz* (in PX, %, VH, VW) jedes Widget pixelgenau platzieren. Vorsicht ist geboten bei Pixelangaben und responsive Ansichten. Und generell bei Positionierungen.

Fixiert macht ähnliches, nur dass das Element als Ursprungspunkt das Browserfenster selber, den Viewport nimmt. Das führt dazu, dass die Ladereihenfolge keine Auswirkung auf die Anzeige hat. Anwendung dafür sind oft Social Media Icons die in einer Browserecke immer sichtbar sind. Bei Arbeiten mit

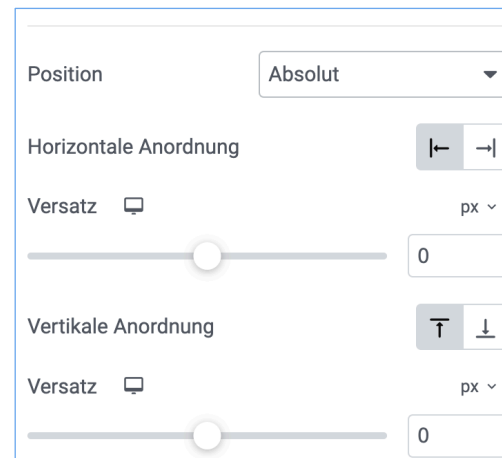


Im Bereich Layout können Margin, Padding und Width (Außenabstand, Innenabstand und Breite) definiert werden. Die Werte können gleich (verknüpft) oder auch getrennt für die 4 Seiten eingegeben werden. Neben den übliche Einheiten kann auch (Bleistift-Icon) ein eigener, in CSS gültiger Wert eingegeben werden.



Hier wurde die CSS Calc-Funktion verwendet. Sie erlaubt es relative Werte (VH) mit absoluten (PX) zu kombinieren.

Eigene Werteangaben sind sehr oft möglich.



Sobald eine Positionierung gewählt wurde werden die Optionen für den Ursprungspunkt sowie den möglichen Versatz zu diesem Punkt angezeigt. Die Optionen sind für beide Arten gleich, diese unterscheiden sich nur vom Ausgangspunkt.

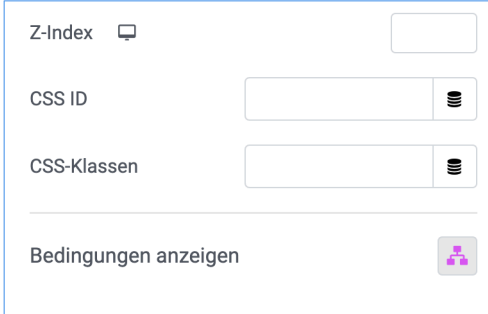
Positionierung ist besonders auch auf die vertikale Lage eines Elements, also den Z-Index zu achten.

Z-Index. Dieser Index legt die vertikale Position eines Elements auf der Seite fest. Grundsätzlich werden die Elemente in der Ladereihenfolge angeordnet: das zuvor geladene Element liegt unter dem später geladenen. Durch Vergabe eines Wertes kann diese Reihenfolge geändert werden. Ein höherer Wert bedeutet, dass das Element über allen Elementen mit einem niedrigeren Wert liegt. Bei zwei gleichen Werten liegt das später geladene über dem vorigen. Der Z-Index kann für einzelne Widgets aber auch für Layout Elemente vergeben werden. Wird ein Container durch Vergabe eines Z-Index-Werts definiert werden alle darin befindlichen Elemente ebenfalls diesem Indexwert zugeordnet. Dieser Index ist Responsive Mode fähig.

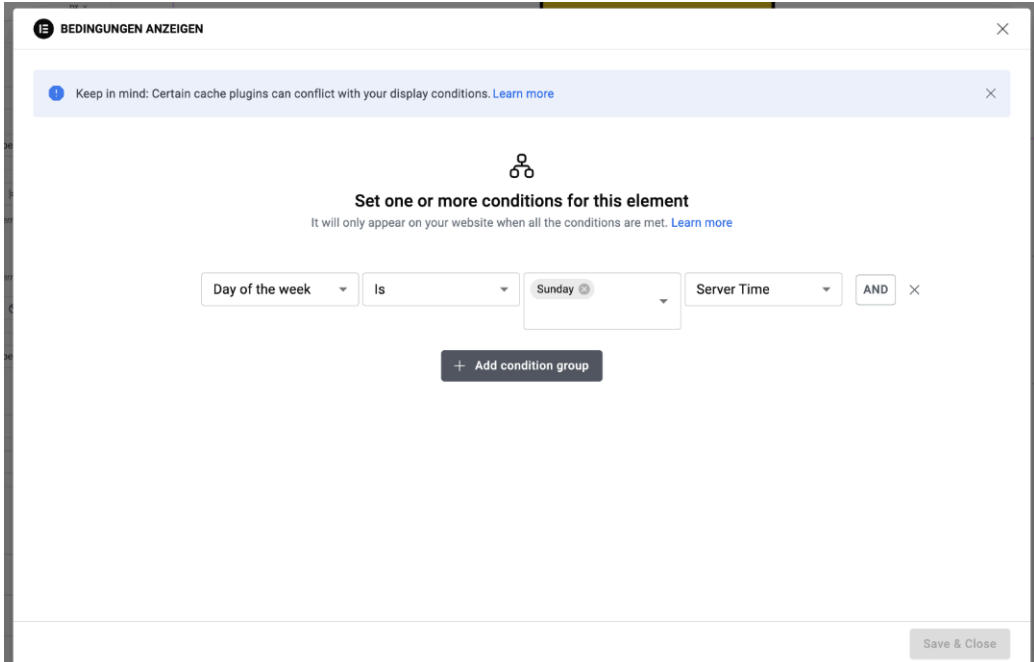
CSS-ID. Durch die manuelle Vergabe einer CSS-ID kann das Element zB. als Textanker verwendet werden (Link nach *#VergebenerName*) oder aber das Element kann damit mittels Scripts oder auch mit eigenem CSS angesprochen werden.

CSS-Klassen die angewendet werden sollen werden auch hier eingetragen. Diese können an verschiedenen Stellen definiert werden: siehe auch im Kapitel 10 für Individuelles CSS. Mehrere Klassen werden durch Leerzeichen getrennt angegeben.

Bedingungen anzeigen. Dieser recht neue Dialog erlaubt es, jedes einzelne Widget nur dann anzuzeigen, wenn die im Dialogfeld eingestellten Bedingungen zutreffen. Das ist besser als nichts, wer solche Funktionen benötigt sollte sich aber das (auch kostenlose) Plugin *Dynamic Visibility for Elementor* näher anschauen. Mit diesem sind komplexere Optionen und Verknüpfungen möglich, ausserdem kann damit Dinge nicht nur Anzeigen, sondern auch Ausblenden.



Kleine Eingaben mit großer Wirkung: Z-Index, CSS ID und CSS-Klassen können sehr wichtig sein. Der Schalter darunter öffnet den Dialog für die Bedingungen.



Bedingungen für die Anzeige eines Widgets können sehr unterschiedliche Trigger haben. Hier ein Beispiel für Einblendungen, die nur Sonntags erfolgen sollen.

Bewegungseffekte

Um über CSS verschiedene Bewegungen anzuwenden kann man Einstellungen in 4 Bereichen vornehmen.

- Scrolling Effects
- Mouse Effects
- Sticky
- Eingangsanimationen

Scrolling Effects werden sichtbar, wenn der Inhalt bewegt wird. Beim *Vertical Scroll* wird der Inhalt (zusätzlich zur Bewegung des Fensterinhaltes) vertikal bewegt. *Horizontal Scroll* bewegt seitlich. Mit der Option *Transparenz* kann man den Alpha-Kanal manipulieren. *Unschärfe* sorgt genau für ebendiese. Mittels *Rotate* kann man Dinge um einen zu definierenden Ankerpunkt drehen und mit *Scale* ermöglicht man eine Skalierung.

Da man die Effekte kombinieren kann ergeben sich zB. aus Unschärfe und Transparenz einfache, aber sehr stimmige Effekte. Man kann die Nutzung dieser Effekte auf *Responsive Modi* beschränken. Ausserdem kann man wählen, ob die *Effekte relativ* zum Fensterausschnitt (Viewport) oder über die Gesamte Seite errechnet werden sollen.

Mouse-Effects - damit werden die Inhalte relativ zur Position des Cursors bewegt (*Mouse Track*) oder 3-dimensional verzerrt (*3D Tilt*). Die deutsche Beschriftung *Geschwindigkeit* ist ein wenig verwirrend, sie definiert nicht die Geschwindigkeit der Bewegung sondern die Ausdehnung der Animation (groß = weiter).

Sticky legt fest, ob ein Inhalt (also auch ein Container) oben oder unten an der Seite beim Scrollen stehen bleibt. Einstellbar sind auch ein Versatz zum Rand (*Auslassen*) und natürlich, ob das Verhalten bei unterschiedlichen *Responsive-Modi* auftreten sollen. Die Option *Offset Effekte* bedeutet, dass bevor ein Element feststeckt eine gewisse Scrollstrecke bereits zurückgelegt sein muss.

Die Schalter für *In der Spalte bleiben* sorgt dafür, dass das gewählte Element nicht zwangsläufig am Fuß (Kopf) der Seite kleben bleibt sondern in der um-

▼ **Bewegungseffekte**

Scrolling Effects An

Vertical Scroll

Horizontal Scroll

Transparenz

Unschärfe

Rotate

Scale

Apply Effects On

× Desktop × Tablet Portrait × Mobile Portrait

Effects Relative To

Mouse Effects An

Mouse Track

3D Tilt

Sticky

Sticky Ein

× Desktop × Tablet Portrait × Mobile Portrait

Sticky Offset

Offset Effekte

Anchor Offset

Using the Anchor offset may require you to adjust the offset of other sticky elements. [Learn more](#)

In der Spalte bleiben

Eingangsanimation

Fade In

Animationsdauer

Verzögerung (ms)

Die 4 verschiedenen Arten von Bewegungseffekten können beliebig miteinander kombiniert werden - nicht immer zum Vorteil der Seite!

Scrolling Effects sind ein beliebtes Mittel um eine individuelles Feedback abhängig von der Fensterposition zu erreichen.

Mouse Effects sind mit Vorsicht einzusetzen, da sie oft eine ungewohntes Feedback beim User erzeugen.

Sticky Effekte sind beliebt und ganze Container (zb Header) oder auch Social-Media-Icons im Sichtbereich des Users zu fixieren.

Eingangsanimationen helfen die gefühlte Ladezeit zu verkürzen.

gebenden Layoutstruktur (Container). Wenn also der Container den Viewport verlässt so verschwindet damit auch der sticky content.

Eingangsanimation legen fest, ob ein Element beim Erscheinen in den Viewport über eine Standard-CSS-Animation verändert werden soll. Die möglichen Animationen sind zahlreich, eingestellt werden kann eine Geschwindigkeit (langsam, normal, schnell) und eine Eingangsverzögerung in Millisekunden.

Transformieren

In diesem Tab sind die Möglichkeiten versteckt wenn man Elemente (besonders natürlich Bilder) Transformieren will. Dies ist grundsätzlich möglich oder aber erst bei Mauskontakt, also Hover. Es stehen 6 Transformationen bereit:

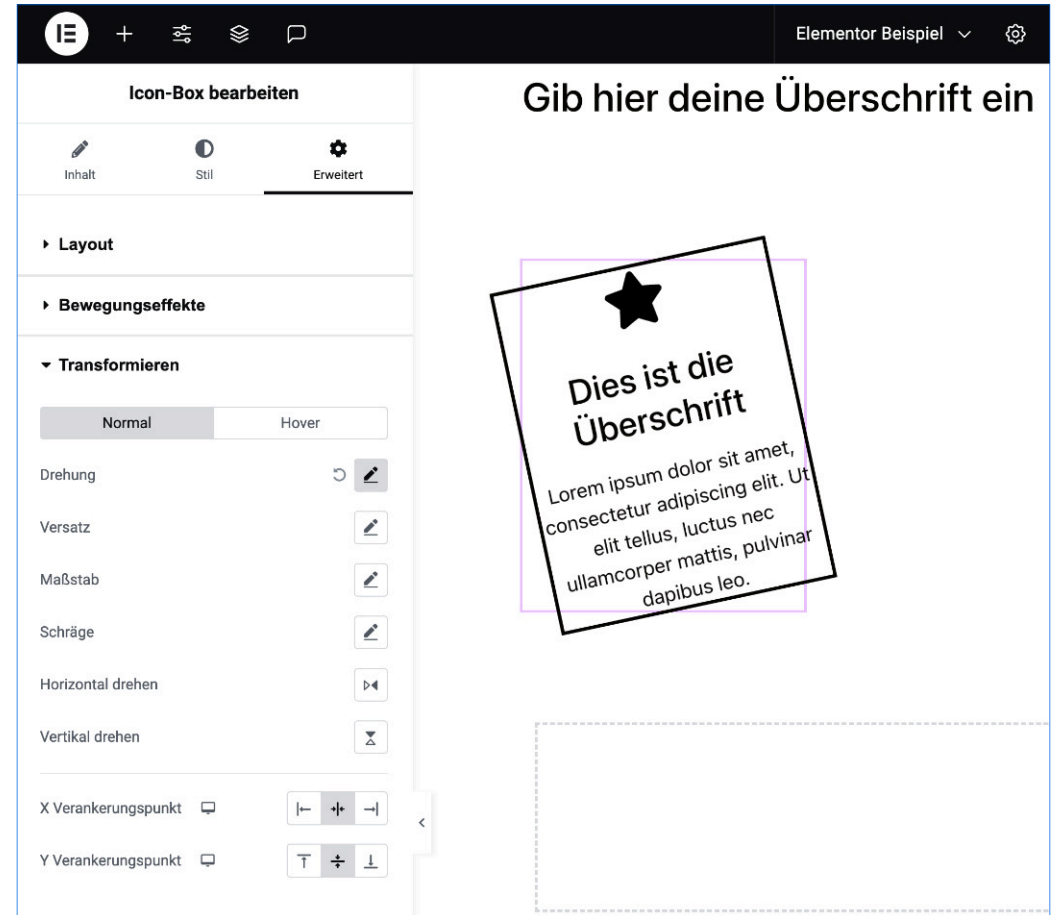
- Drehung (rotate)
- Versatz
- Maßstab (scale)
- Schräge (skew)
- Horizontal drehen (=eigentlich ist ein Flip gemeint)
- Vertikal drehen (auch hier flippen).

Für alle kann Hover mit einer Übergangsdauer eingestellt werden (in ms).

Drehen. Auf den ersten Blick ist drehen einfach (0 - 360°), darunter ist aber der Schalter 3D-Drehen der es ermöglicht die Drehung über die X- und/oder Y-Achse zu führen und auch eine Perspektive einzustellen. Mit Bedacht eingesetzt kann das spannend sein.

Versatz. Der Versatz ist eine gute Möglichkeit Dinge bei Mauskontakt horizontal und/oder vertikal zu bewegen. Angaben sind in % oder in PX möglich. Wie immer beziehen sich Angaben in % auf das nächst äussere Element.

Maßstab. Die etwas unglückliche Übersetzung des Wortes Skalieren. Es kann gewählt werden ob die Proportionen beibehalten werden sollen (default) oder ob eine unterschiedliche Skalierung erfolgen soll. Die Werte können positiv (=



Transformationen sorgen dafür, dass das Widget ausgehend vom einzustellenden Verankerungspunkt verändert wird. der rosa Rahmen ist technisch gesehen das Widget, die schwarz umrandete Iconbox ist dessen Darstellung.

Dieses Werkzeug eignet sich hervorragend um falsch definierte Bilder (Hoch/Querformat) richtig auszurichten ohne die Bilddatei verändern zu müssen.

vergrößern) oder aber auch negativ (= verkleinern) sein.

Schräge. Erlaubt eine Verzerrung entlang der X (=Neigung) oder der Y-Achse (=Schräg).

Horizontal / Vertikal drehen. Meint eigentlich Spiegeln (flip). Der Punkt um den (herum) gespiegelt/gedreht wird kann als X / Y Ankerpunkt links/Mitte/rechts gesetzt werden.

Hintergrund

Zusätzlich zu dem ev. im Layout-Tab oder im Widget vorhandenen Einstellungsmöglichkeiten kann man in diesem Bereich Hintergründe gestalten. Das können Farben, Verläufe und Bilder sein. Weiters kann auch ein Hover-Status definiert werden, wodurch die Inhalte bei Mauskontakt ein anderes Aussehen annehmen können.

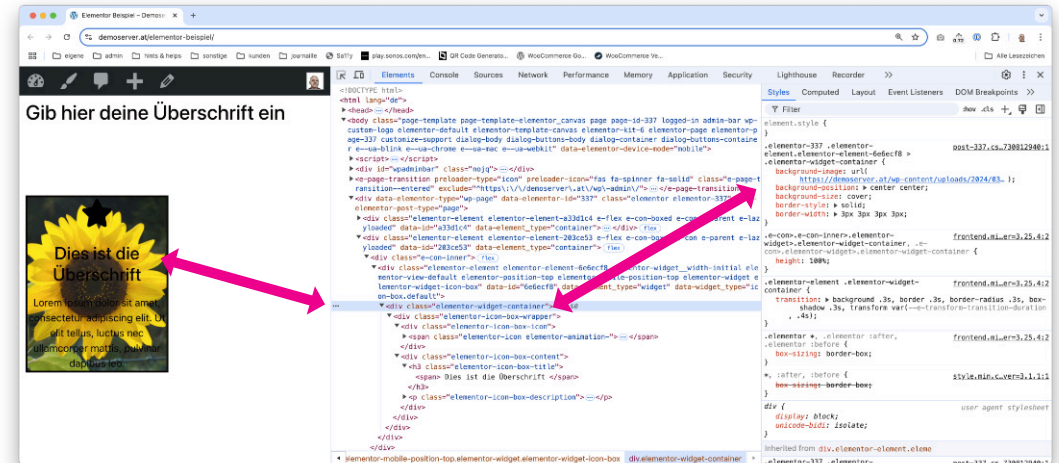
Es es eher selten, dass an dieser Stelle ein Hintergrund gesetzt wird. Aber bei zB. Inline-Elementen oder auch Überschriften kann man es manchmal gut brauchen.

Rahmen

Um einen Rahmen um ein Element zu zeichnen müssen 3 Parameter angegeben werden. Der erste ist der *Rahmentyp*, der Durchgezogen, Doppelt, Gepunktet, Gestrichelt oder Groove (Pseudo 3D) gestaltet sein kann. Der zweite Parameter ist die *Rahmenbreite*, die einheitlich sein kann oder aber auch unterschiedliche Stärken aufweisen darf. Als dritter notwendiger Parameter muss eine *Farbe* angegeben werden.

Rahmen können für alle 4 Seiten gleich oder getrennt gesetzt werden. Daher eignen sie sich gut auch für Trennlinien die zB. Nur unter einem Element erscheinen sollen und genau dessen Breite haben müssen.

Zusätzlich kann für Rahmen ein Hover eingestellt werden. Weiters kann hier auch ein (gleicher oder unterschiedlicher) Eckradius angegeben werden. Die



Elementor packt seine Widgets in HTML-Div-Tags. Das mag auf den ersten Blick verschwenderisch wirken, ermöglicht aber auch sehr einfach all diese Manipulationen wie auch Transformationen, Hintergrundeneinstellungen, Rahmen, Aussenabstände etc.

Optionen für *Box-Schatten* erlauben schließlich eine Schattierung des Elements.

Maskieren

Diese Technik wird nicht sehr häufig eingesetzt. Angeboten werden fertige Formen wie Kreis oder Dreieck oder auch die Möglichkeit, ein eigenes SVG-File als Maske zu verwenden. Es gibt verschiedene Einstellungsmöglichkeiten um diese Maske genau über ein Bild zu legen. Durch Maskierung kann man Bilder in eine andere Form als das Rechteck bringen.

Responsive

Um Elemente innerhalb bestimmter Gerätegruppen auszublenden kann im Tab *Responsive* der jeweilige Gerätetyp eingestellt werden. Es ist also sehr einfach bestimmte Elemente nur oder eben nicht im Mobil-Modus anzuzeigen etc. Innerhalb von Elementor werden verborgende Elemente grau schraffiert als ausgeblendet gekennzeichnet. Man kann auch unter

Panel > Logo (links oben) > Benutzer Einstellungen > Verborgene Elemente anzeigen

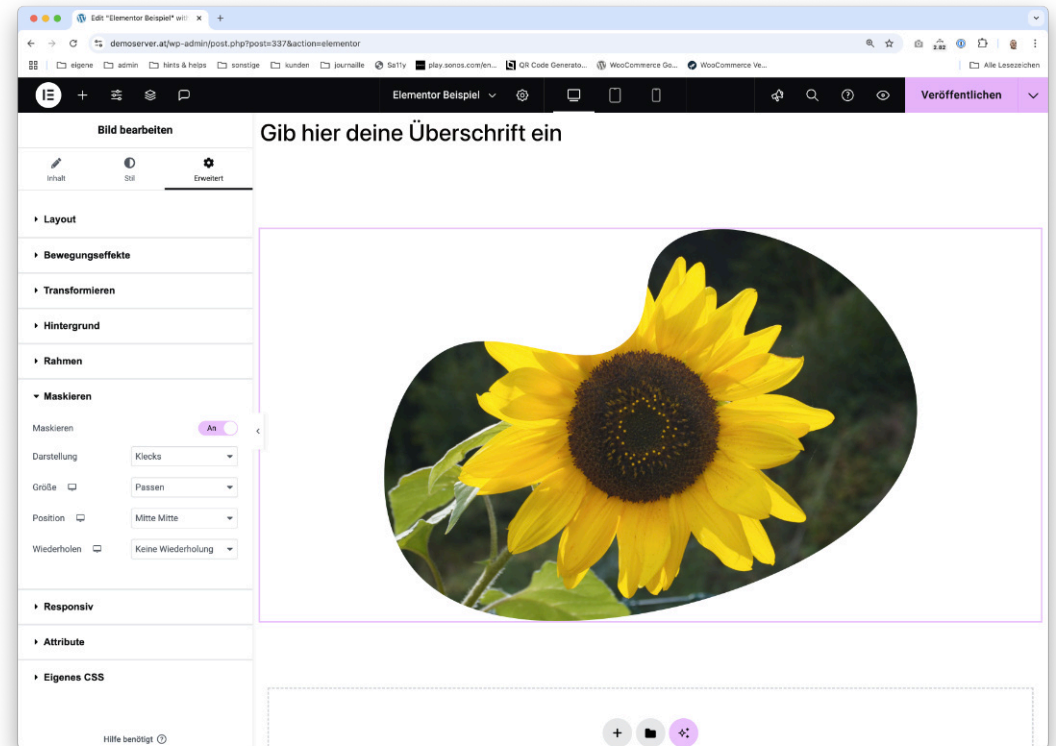
diese Ansicht deaktivieren, damit werden verborgene Elemente auch innerhalb der Elementor-Ansicht ausgeblendet.

Attribute

Wer Attribute eingeben muss macht das hier. Anwendungen sind zB. Werte die für Google Analytics für das Scroll-Tracking eingefügt werden müssen.

Eigenes CSS

Eigener CSS-Code der nur für dieses eine Element benötigt wird kann hier



Maskierungen werden selten, und wenn dann am ehesten bei Bildern eingesetzt. Die (SVG)-Masken decken Teile des Bildes ab, so entstehen frei definierbare Formen.

eingetragen werden. Das Keyword **selector** hilft und bedeutet, „nimm dieses Element“. Mehr über CSS in Kapitel 10.

Die Widgets sind im Panel gruppiert in *Layout*, *Grundlegend*, *Pro*, *Generell*, *Website* und *Single*. Bevorzugte Widgets kann man in der Gruppe *Favoriten* zusammenfassen.

6.2. Layout und Grundlegende Widgets

Container, Grid

Zur Anwendung siehe Kapitel 5.

Die 9 Basis-Widgets stehen auch in der kostenlosen Elementor-Version zur Verfügung (wie auch die unter Generell zusammengefassten Widgets) (Reihenfolge im Panel):

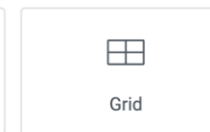
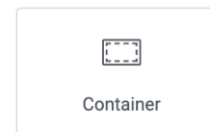
Überschrift

Überschriften (Titel) können eingegeben oder auch dynamisch eingebunden werden. Sie können verlinkt werden und der HTML-Tag kann neben den H1...H6-Tags auch auf div, span und p gesetzt werden. Ausrichtungen Links/Mitte/Rechts und Ausgerichtet (=Blocksatz, wirkt nur bei mehrzeiligen Überschriften) sind möglich. Formatiert werden Überschriften im Stil-Tab wo Textfarbe, Typographie und Textschatten eingestellt werden können. Die Anwendung von Globalen Farben und Schriften ist hier möglich und sinnvoll.

Bild

Das Bild-Widget läßt Ausrichtungen, Bildunterschrift und Link einstellen. Das Bild selber wird aus der Mediathek gewählt. WordPress speichert jedes Bild in unterschiedlichen Größen ab, welche angewandt wird ist unter Bildgröße einzustellen. Die richtige Wahl reduziert Ladezeit! Im Stil-Tab lassen sich Breite

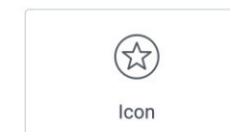
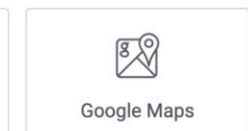
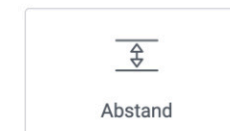
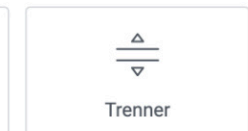
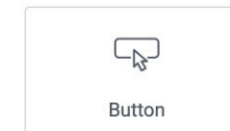
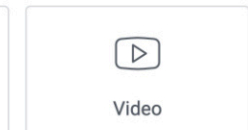
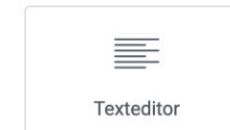
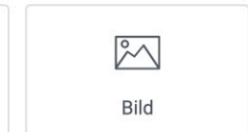
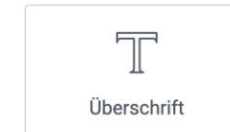
▼ Layout



Ein Container (und Grid) wird automatisch angelegt wenn man ein Widget in den strichliert umrahmten Bereich zieht. Man kann Container aber auch einfach wie jedes andere Widget platzieren.

Die 9 grundlegenden Widgets sind auch die wichtigsten - weil sie am häufigsten verwendet werden.

▼ Grundlegend



und Höchstbreite (in %, PX und VW) wählen. Weiters kann ein Hover festgelegt werden, die Deckkraft (Alpha) kann verändert werden und die klassischen CSS-Filter angewandt werden. Ausserdem gibt es die üblichen Rahmen, Eckradius und Schattenoptionen.

Text-Editor

Das Widget verwendet den bekannten Inline-Editor und dient als universeller Container aller möglicher Inhalte. Der Wechsel zwischen Visuell und Text erlaubt den Blick auf den SRC-Code des Inhaltes. Mehrspaltiger Text kann (für die verschiedenen Responsive Modi) eingestellt werden. Die Ausrichtung ist im Stil-Tab untergebracht. Im Text-Editor können auch Shortcodes untergebracht werden. Die Formatierungen erzeugen HTML-Formatierungen.

Video

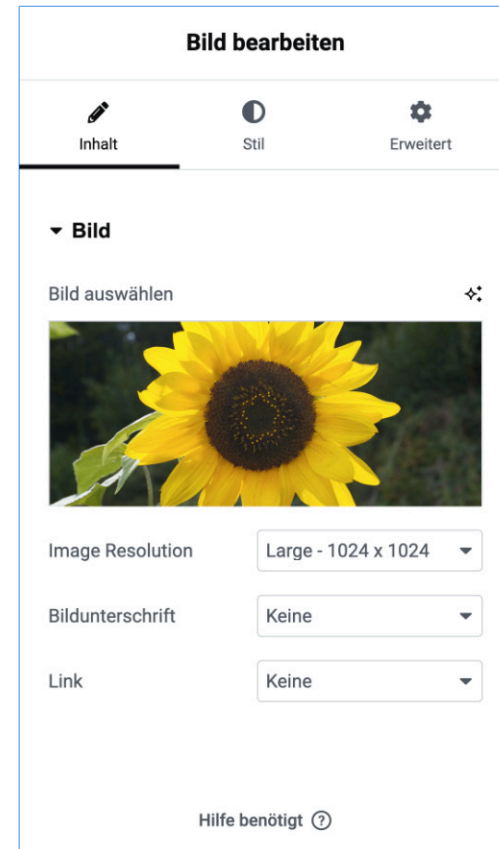
Mit dem Video-Widget lassen sich Inhalte von YouTube, Vimeo, Dailymotion, Videopress oder auch selbst gehostete Videos einbinden. Mittels Bildüberlagerung kann man das angezeigte Standbild selber festlegen. Bei aktivierter Bildüberlagerung kann man das Video auch im Lightbox-Fenster anzeigen lassen. Die angezeigten Optionen sind abhängig vom jeweiligen Video-Hoster und variieren deutlich.

Button

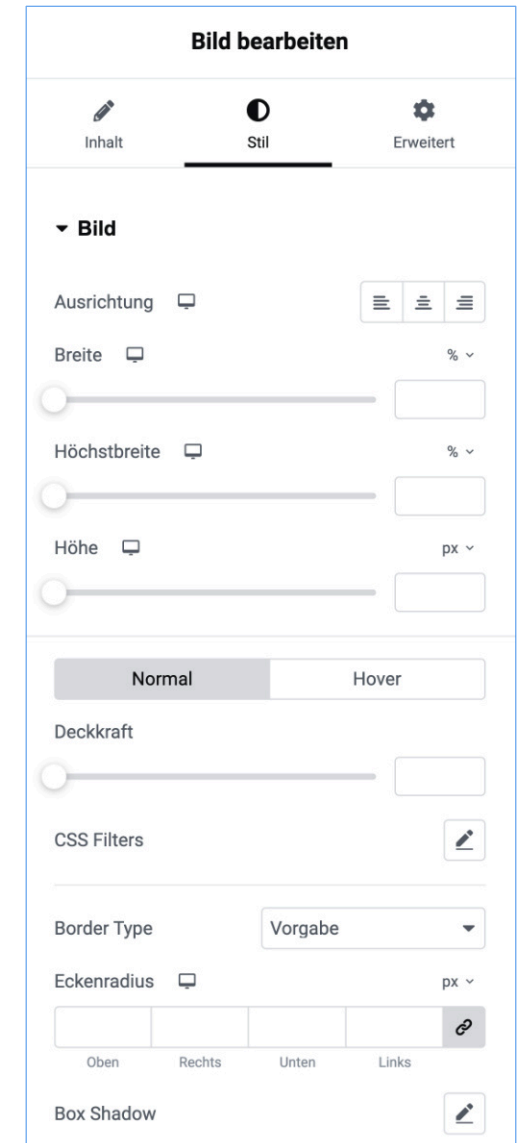
Für auffällige Links ist das Button Widget geeignet. Die in den Webseiten-Einstellungen vorgenommenen Designoptionen können hier überschrieben werden.

Trenner

Der historische HR-Tag hat hier seine Entsprechung im Elementor gefunden. Kann hier mit Icons verfeinert werden.



Jedes Widget bietet je nach Anzahl und Art der Inhalte unterschiedliche, tw. sehr vielfältige Einstellungs- und Gestaltungsmöglichkeiten. Diese können rein formal oder auch technisch relevant wirken. Für jedes Widget gibt es auf der Elementor-Website eine detaillierte Beschreibung.



Abstand

Das Abstand-Widget macht genau wie es heisst: Abstand. Sinnvoll ist der Einsatz auch dann, wenn Container mit Farben oder Hintergrund ausgestattet wurden aber diese aufgrund fehlenden Inhaltes nicht angezeigt werden.

Google Maps

Spannend an der Umsetzung ist hier, dass im Ort-Feld eine Adresse eingegeben werden kann und der Rest wird im Hintergrund erledigt. Erweiterte Funktionen wie eigene Pins etc. sind nicht möglich, aber eine rasche Karteneinbindung ist mit wenigen Klicks erreicht. Achtung: DSGVO-konforme Nutzung beachten!

Icon

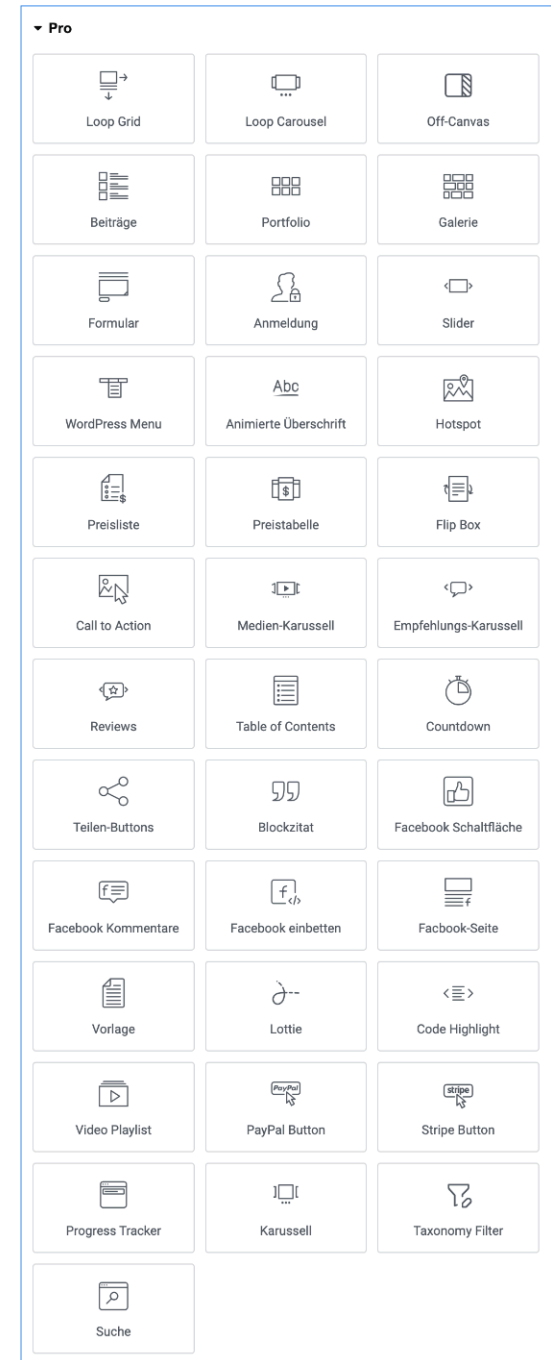
Icons können verlinkt - und wenn die Font-Awesome-Icons, eigene Icon-Bibliotheken oder andere funktionierende SVG-Dateien verwendet werden, auch gefärbt und gedreht werden (auch als Hover). Ein stark unterschätztes Gestaltungswerkzeug.

6.3. Pro

Im Augenblick werden 37 Widgets unter Pro eingeordnet. An dieser Stelle werden nur wenige davon näher besprochen. Pro heisst, dass diese nur bei einer gültigen Pro-Lizenz eingebunden werden können. Das bedeutet, dass diese Inhalte aus der Webseite verschwinden, sollte die Pro-Erweiterung nicht mehr installiert ist.

In diesem Skriptum werden nicht alle Widgets beschrieben.

Der Überblick über die Pro-Widgets fällt langsam schwer. In dieser Rubrik wurden sehr unterschiedliche, mitunter auch sehr komplexe Widgets, die mehr können als nur eindimensional Dinge anzuordnen.



Loop Grid / Beiträge / Portfolio / Taxonomy Filter

Diese Widgets dienen dazu Archive darzustellen. *Loop Grid* ist notwendig für die Arbeit mit dem Loop-Template (siehe Kapitel 4 Theme Builder). *Beiträge* ermöglicht die Archivanzeige mit Klassischem-, Karten- oder Full Content-Design. *Portfolio* wiederum bietet weniger Gestaltungsmöglichkeiten als das Beiträge-Widget, kann aber dafür Filterleisten darstellen mit denen man zB. nach Kategorien oder anderen Taxonomien auswählen kann. Taxonomy Filter ergänzt das Loop Grid.

Formular

Mit diesem Widget wird das Formularsystem bedient das im Elementor Pro vorhanden ist. Für sehr viele Formularanwendungen reicht das aus, ein zusätzliches Formular-Plugin ist nicht mehr notwendig. Ausführlich besprochen wird es in Kapitel 11 Formulare.

Navigationsmenü

Um die über das WordPress Dashboard Menüs auf der Webseite darzustellen benötigt man dieses Widget. Neben der Menüwahl kann man die grundsätzliche Ausrichtung (horizontal, vertikal) wählen. Weiters sind hier auch die Möglichkeiten für das Responsive-Verhalten untergebracht. Die Zeiger-Funktion hilft bei der ergonomischen Seitengestaltung durch Anzeige der aktiven Seite.

Table of Contents

Wer Seiten mit viel Inhalt hat wird dieses Widget verwenden. Es erstellt aus der angegebenen Struktur (H1 ... H6, konfigurierbar) ein Inhaltsverzeichnis das automatisch aktuell gehalten wird. Das mühsame erstellen von Textankern entfällt. Man kann die gewünschten Überschriftenebenen einstellen die verwendet werden sollen. Weiters kann man einen Container definieren dessen Inhalt auf die Überschriften gescannt wird.



werkform trainings

Einzelstunden

Die **persönliche** Einzelstunde ist (und bleibt) das erfolgreichste Angebot der werkform Trainings: deine ganz persönliche Anforderung (Elementor, WordPress oder auch Freelancer- oder Agentur-business) werden besprochen.

WordPress Basics

In diesem Kurs lernst du die **Grundlagen** für die Arbeit mit dem erfolgreichsten CMS der Welt. Gutenberg ist die Basis. Geeignet für einzelne TeilnehmerInnen oder auch für Kleingruppen.

WordPress Aufbau

Erweitere dein **Know-How**: welche Konzepte, welche Plugins um was zu erreichen? Dieser Kurs hilft dir bei der Wegentscheidung, welche Technik für welche Anforderung die richtige ist. Praktische Beispiele und Lösungen für genau dein Projekt.

Elementor Einstieg

Die Oberfläche, die Arbeitsweise, die **Grundeinstellungen**. Seitenaufbau, Widgetüberblick, einige Webdesignbasics. Mit diesem Kurs schaffst du den ersten Schritt in professionelles Webdesign ohne Coding. Freie Elementor Version ausreichend.

Elementor Aufstieg

Vom **Theme Builder** über weitere Templates zu Archiven und Beiträgen (auch CPT). Elementor Formularsystem verstehen und erweitern. Nach diesem Kurs **kannst du alles** für eine komplexe Pro-Website. Inkl. Plugin-Empfehlungen, Backups und den Einstieg in Elementor AI.

Bei allen Schulungen können Trainings-Webhosts und **Lizenzen** der verwendeten Software für die Dauer der Schulungen bereit gestellt werden (ohne Zusatzkosten!).

Vorlage

Zeigt Templates an, die nicht Teil des Theme-Builders sein müssen. Auf diese Weise lässt sich zB. die traditionelle Sidebar-Funktionalität umsetzen.

Off-Canvas

Dieses recht neue Widget ist eine Ausnahme: hier gibt es keinen Tab Erweitert. Das liegt am Einsatzzweck: ähnlich wie ein Popup erzeugt es eine Struktur, die - mit anderen Widgets ausgestattet - nur bei bestimmten Aktionen des Users angezeigt wird. Folge dessen gibt es neben dem Inhaltsbereich auch eine Hintergrundüberlagerung. Ein Beispiel wäre ein erweiterte Bildanzeige nach einem Button-Klick.

6.4. Generell

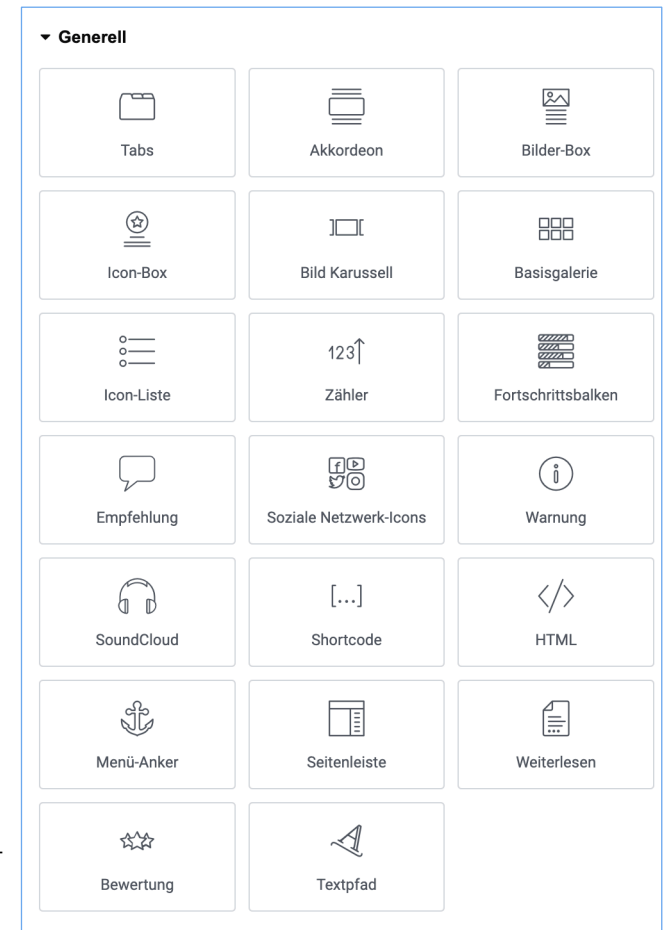
Im Augenblick werden 20 Widgets als Generell eingeordnet, ganz logisch scheinen die Einordnungen nicht. Viele Dinge braucht man häufig, diese Widgets sind auch in der freien Elementor-Version verfügbar, was wohl auch ein Grund für diese Zusammenstellung ist.

Icon Liste

Für alle Listeninhalte geeignet (auch ohne Icons einsetzbar). Listen können vertikal und horizontal ausgegeben werden. Elementor erlaubt das einfache Umsortieren der Listeneinträge per Drag&Drop.

Tabs/Akkordeon

Die klassischen Layout-Tools, beide sind mit Vorsicht und Bedacht einzusetzen. Beim Akkordeon läßt sich das FAQ-Schema aktivieren (SEO-relevant). Die Widgets werden inhaltlich gleich bedient, die einzelnen Inhalte lassen sich



Klassische Designelemente sind in den Generell-Widgets zu finden. Diese Widgets sind auch in der freien Version von Elementor zu finden.

einfach umsortieren, Tabs sind vertikal und horizontal möglich.

Soziale Netzwerk-Icons

Ist eine Icon-Liste die auf Social Media Links getrimmt wurde. Durch die Auswahl eines Social Media Icons aus der Font-Awesome Bibliothek kennt dieses Widget auch die jeweiligen Originalfarben des Dienstes die verwendet werden können.

Shortcode

Immer dann, wenn Dritthersteller Plugins eingesetzt werden sollen kann man die notwendigen Shortcodes im Text, oder aber auch mit diesem Widget einbinden. Ein Vorteil des Widgets besteht in der Vorschaufunktion, ein anderer darin, dass über die erweiterten Funktionen Gestaltungen wie Margin, Border, etc. auch auf das Element angewandt werden können.

HTML

Händisch HTML-Codes einzubauen ist mit dieses Widget möglich. Elementor versucht den Code auf Gültigkeit zu prüfen und stellt das auch im Eingabefenster dar.

Textpfad

Wer Texte im Kreis oder auf einem beliebigen (SVG) Pfad stellen möchte kann das mit diesem Widget tun - Text bleibt Text!

6.5. Link in Bio

Die mit der Verbreitung der Block-Themes immer beliebter werdenden Pat-

terns haben schon länger in der Elementor-Bibliothek seine Entsprechung. Ein weitere Baustein aus der Werkzeugkiste zur schnellen Umsetzung sind diese 7 Widgets. Sie sind Kombinationen aus Bild, Listeninfos, Text und Links. So wirklich überzeugen diese Widgets jedoch nicht.

6.6. Website

Die 5 Website-Widgets dienen dazu, WordPress-eigene Funktionen im Elementor anzuzeigen bzw. eine Auflistung (Sitemap) der Seiten und/oder Beiträge anzuzeigen. Häufigstes Einsatzgebiet sind Templates des Theme-Builders.

Site-Logo

Das Logo einer Webseite kann man (je nach Theme) über den Customizer oder über die Webseiten-Einstellungen im Elementor festlegen. Neben den üblichen Optionen die bei Bildern üblich sind ist das Site-Logo von Haus aus mit der Startseite verlinkt.

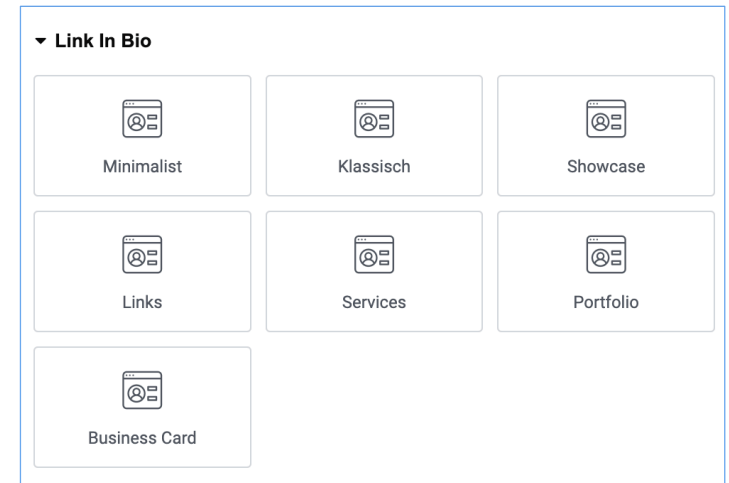
Seitentitel / Seiten Titel

Diese beiden im deutschen beinahe gleich benannten Widgets haben komplett unterschiedliche Funktionen. Mit Seitentitel ist eigentlich der Website-Titel gemeint, also jener Name den man in den Webseiten-Einstellungen vergeben hat. Seiten Titel hingegen ist der Titel der einzelnen Seite die man gerade anzeigt.

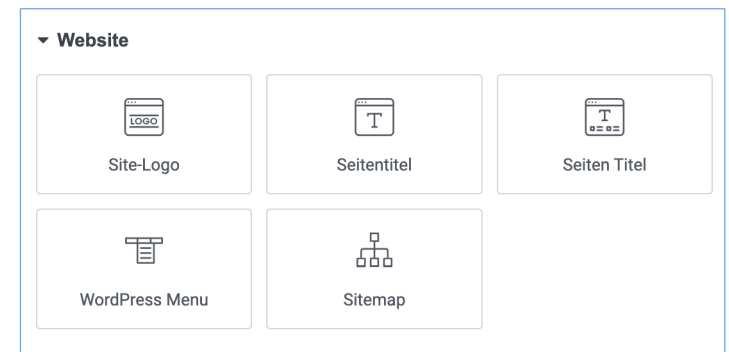
WordPress Menü

Mit diesem Widget lassen sich die im Dashboard > Design > Menu angelegten Menüs anzeigen und gestalten.

Die 7 neuen Widgets in der neuen Kategorie Link in Bio sind kombinierte Designelemente. Eigentlich sind es Designpatterns die Lösungen anbieten, die man auch „händisch“ umsetzen könnte.



Die Website-Widgets sind bis auf die Sitemap einfach technisch notwendig und erwartbar. Das Sitemap-Widget kann durchaus hilfreich sein für Inhaltsverzeichnisse o.ä. Anwendungen.



Sitemap

Zur Erstellung einer Übersicht über die aktuelle Webseite ist dieses Widget gedacht. Die Aufzählungen der Seiten und Beiträge kann nach eigenen Taxonomien erfolgen, eigene Post Types sind möglich. Das Ausschließen einzelner Posts ist möglich.

6.7. Single

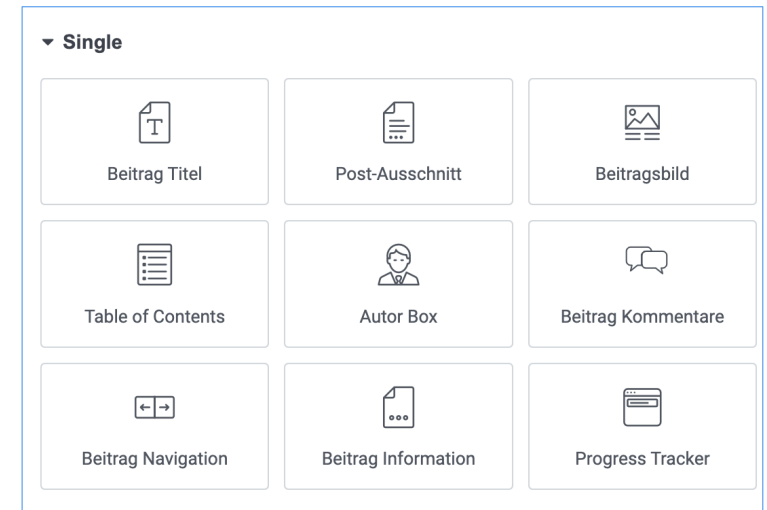
Die unter Single eingereihten Widgets dienen im Wesentlichen dazu Templates für Blogposts (oder eigene Post-Typen) zu gestalten - Also Single Post Templates. Ausserdem kommen sie bei beim Loop-Template zum Einsatz. Die Widgets dafür sind:

- Beitrag Titel
- Post-Ausschnitt
- Beitragsbild
- Table of Contents
- Autor Box
- Beitrag Kommentare
- Beitrag Navigation
- Beitrag Information
- Progress Tracker

Beispiele für deren Anwendung in Kapitel 4. dieses Skriptums.

6.8. WordPress

Das WordPress-eigene Widget-System kann auch über den Elementor angesprochen werden. Hier wird nicht näher darauf eingegangen. Über die Einstellungen können diese Widgets im Elementor ausgeblendet werden.



Zur Gestaltung von Single Post Templates dienen diese Widgets.



Dir gefällt dieses Skriptum?

Kaffeesspende hier!

jetzt spenden

Spende Skriptum 3.0



Anzeige

7. Responsive Design

Es lohnt sich, schon in der Entwurfsphase eines Projektes über die unterschiedlichen Nutzererfahrungen (UX) der hoffentlich zahlreichen Benutzer der Webseite nachzudenken. Die erste Frage lautet also: wie groß (oder breit) ist der Bildschirm des Besuchers? Und die Antwort drauf lautet: wir wissen es nicht! Was wir aber wissen ist, dass unterschiedliche Geräte unterschiedliche Anforderungen an unser Design stellen. Gemeinhin eingeteilt in Desktop, Tablet und Mobile.

Da wir die Bildschirmbreite nicht kennen (und diese ggf. schon recht breit werden kann) können wir eine Breite angeben, in welcher der Inhalt der Seite untergebracht werden soll. Wenn wir diese **Inhaltsbreite** überschreiten wollen (z.B. im Header, Footer oder für Hero-Bilder) müssen wir das gesondert einstellen. Diese Breite wird verwendet wenn die Einstellung als *boxed* angegeben wird.

Wenn jetzt ein Gerät (oder das Browserfenster) schmaler als die Inhaltsbreite ist möchte man eine andere Darstellung anzeigen, zB. um von einer horizontalen zu einer vertikalen Ausrichtung zu wechseln. Dieser Punkt wo man zwischen diesen Ansichten wechselt ist der sog. **Tablet-Breakpoint**.

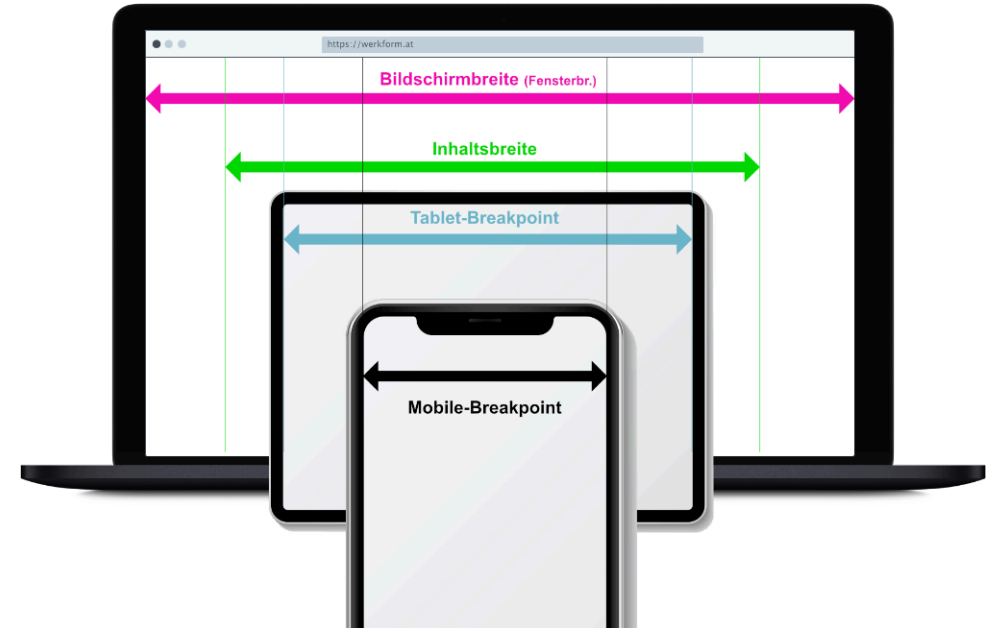
Wenn der User nun ein Gerät hat, das noch schmaler ist – ein Handy – wird eine gedrungene Gestaltung sinnvoll sein. Der **Mobile-Breakpoint**.

Elementor bietet die Aktivierung weiterer Breakpoints an. Dabei gilt es zu berücksichtigen, ob der Mehraufwand an Gestaltung für zB. eine 4. oder 5. Version eines Footers wirklich notwendig ist.

Die Voreinstellungen im Elementor für diese 3 Breiten sind:

- Inhaltsbreite: 1140 px
- Tablet-Portrait-Breakpoint: 1024 px
- Mobile-Portrait-Breakpoint: 767 px

Wer diese Einstellung ändern möchte kann das unter



Die Breiten des Bildschirms sind nicht nur abhängig von den Geräten, sondern auch von deren Ausrichtung. Daher heißen die Modi jetzt Desktop, Tablet Portrait und Mobile Portrait.

Panel > Website Einstellungen > (Einstellungen) Layout

Die Maßeinheit ist hier Pixel – das geht auch nicht anders, diese Definitionen können nur mit absoluten Werten funktionieren. Für die Inhaltsbreite lassen sich auch andere Einheiten (% , VW, REM, EM und Custom) einstellen, diese relativen Angaben beziehen sich dann auf die Breite des Viewports (=Browserfensters).

Wenn man an einem (responsive) Design arbeitet sollte man versuchen, nicht in absoluten sondern in relativen Angaben zu denken und zu gestalten. Zumindest dort, wo es angebracht ist.

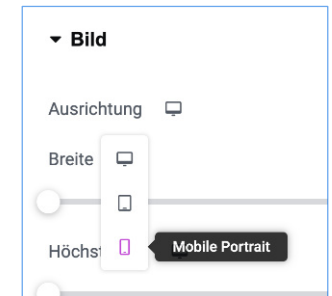
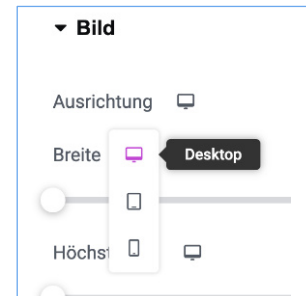
Es gibt eine Voreinstellung Lücken (getrennt für Reihen und Spalten) die auf 20 px gesetzt ist. Das ist grundsätzlich ein sehr brauchbarer Wert, schliesslich will man vermeiden, dass ein Fließtext bis zum Bildrand läuft oder eine Icon-box direkt an eine andere stößt. Eine Abstand in Pixel kann jedoch auch problematisch werden. Ein klassisches Problem ist hier ein Padding (also der Abstand des Inhaltes zum Rand des Elements) das dazu führt, dass der Inhalt am Handy kaum mehr Platz hat, weil seitlich die Abstände absolut gesetzt wurden.

Immer dann, wenn man nicht genau weiß, wieviel Platz man eigentlich hat, sind also *relative* Angaben sinnvoll. Die bekannteste Einheit für eine relative Angabe ist natürlich % [Prozent], wobei sich die Angabe der Zahl immer auf den zur Verfügung stehenden Platz, also das nächst äussere Element bezieht. Oft sind auch Angaben in VW oder VH sinnvoll (1-100 beziehen sich immer auf 1/100tel des Viewports).

Das Elementor Interface bietet nun dort, wo es Elementor für angebracht hält, die Möglichkeit an unterschiedliche Einstellungen für die verschiedenen Modi Desktop – Tablet – Mobil vorzunehmen. Der Wechsel zwischen den Modi kann entweder durch kleine Popups oder in der Mitte im Top-Bar über Auswahl der entsprechenden Modi-Icons erfolgen.



Wenn man den Bildschirmmodus wechseln werden die Anzeige und die Eingaben auf den ausgewechselten Modus gesetzt. Sind zusätzliche Breakpoints aktiviert tauchen diese als Icons in der Leiste und bei den Popups zu den Einstellungen auf.



Der Popup-Umschalter um den Modus zu wechseln taucht an sehr vielen Stellen bei Elementor auf. Wenn er nicht da ist bedeutet das, dass man die gewünschte Einstellung nicht über das Interface ändern kann sondern auf CSS ausweichen muß.

7.1. Schriften responsive einstellen

Eine der wichtigsten Anpassungen die man im Responsive Design vornehmen kann sind die Schriften. Am mobilen Gerät ist einfach viel weniger Platz auf der einen Seite, der User ist aber näher davor auf der anderen Seite. Es ist also eine gute Idee, die Schriftgrößen (aber auch Farben, Schriftarten etc.) **grundsätzlich** responsive einzustellen. Generell geschieht dies über die Einstellungen (Kapitel 2).

Schritt 1:

Panel > Website Einstellungen > (Design System) Globale Schriftart

Unterschiedlich einstellbar sind bei der Globalen Schriftart die Schriftgrößen für die drei Modi.

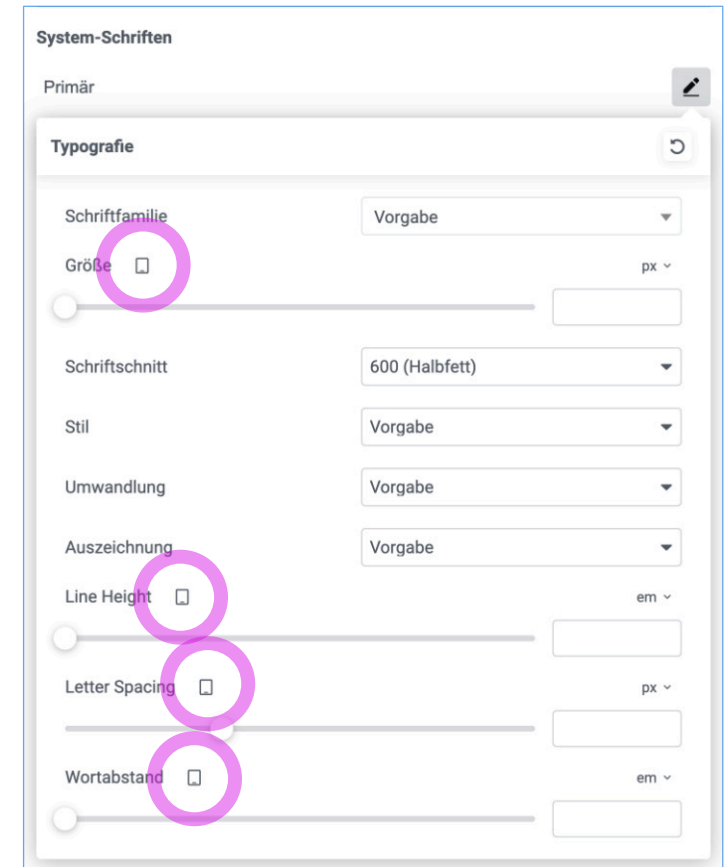
Schritt 2:

Panel > Website Einstellungen > (Theme Style) Typographie

Unter dem gewählten Menüpunkt findet man auch die (responsive) Einstellungen für die Überschriften H1 bis H6. Während Angaben wie zB. Schriftschnitt oder Stil nicht unterschiedlich einstellbar sind, kann man

- Größe
- Zeilenhöhe
- Zeichenabstand
- Wortabstand

für die Responsive-Modi anpassen. Aufpassen bei den Einheiten: Elementor hat in den Voreinstellungen manchmal Pixel, manchmal aber auch EM ausgewählt - dabei sollte man überall nach Möglichkeit mit REM arbeiten.



Für die Globalen Schriften und die Typographie-Einstellungen lassen sich 4 Parameter responsive definieren.

7.2. Container responsive gestalten

Als wichtigstes Gestaltungselement ist jeder einzelne Container an 7 Stellen für die jeweiligen Ansichtsmodi anpassbar.

- Breite
- Min. Höhe
- Richtung
- Verteilung
- Ausrichtung
- Lücken
- Umbruch

Die Breite kann helfen Elemente untereinander anzuordnen auch ohne die Richtung zu ändern: Elemente die 100 % Breit sind sorgen fürs umbrechen - natürlich nur, wenn der Umbruch auch erlaubt ist.

Einfacher ist es oft, einfach die Richtung zu ändern. Dann ist oft mit einem Klick ein gutes Verhalten auf schmalen Schirmen erreicht.

Wer mit Abschnitten und Spalten gearbeitet hat kann sich vmtl. an die Möglichkeit erinnern, die Reihenfolge der Spalten für einen bestimmten Modus umzukehren. Das gibt es nicht mehr, dafür kann man aber jedes Element an eine andere Position innerhalb des Containers setzen - und das eben auch responsive. Von den 7 responsive Optionen die unter

... > Erweitert > Layout

jedes Elements zu finden sind haben 3 Auswirkungen auf den Container bzw. die Positionierung im Container:

- Align Self
- Order
- Size

Align Self erlaubt das ändern der Ausrichtung des einzelnen Elements im Unterschied zur Ausrichtung aller im Container selbst eingestellten Ausrichtung. Unter *Order* kann das einzelne Element (zB. auch ein

Container lassen sich sehr gut an den jeweiligen Bildschirmmodus anpassen.

eingebetteter Container) an eine andere Stelle sortiert werden kann. *Size* ist die Stelle, wo die sog. *Shorthand CSS property* eingestellt werden kann. Also (für Auskenner) *flex-grow*, *flex-shrink*, ...

7.3. Zusätzliche Breakpoints

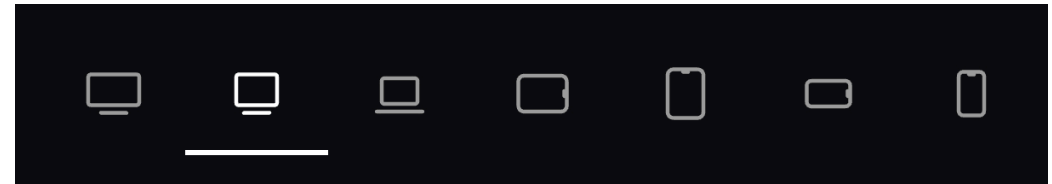
Wem die 3 Modi für Desktop/Tablet/Mobil nicht genug sind kann mit weiteren Standardbreiten experimentieren. Es stehen unter

Panel > Website Einstellungen > Layout-Einstellungen > Breakpoints

bis zu 6 frei zu konfigurierende Umbruch-Breiten + Desktop zur Verfügung:

- Mobil Portrait (default, nicht löschar) (767 px)
- Mobil Landscape (880 px)
- Tablet Portrait (default, nicht löschar) (1024 px)
- Tablet Landscape (1200 px)
- Laptop (1366 px)
- Desktop (default, nicht löschar, keine Breite)
- Breitbild (2400 px)

In der Klammer jeweils die Angaben zu den default-Werten, die aber nach individuellen Wünschen angepasst werden können. Ob sich diese Idee in der Praxis auch durchsetzt wird man erst sehen, auch jetzt ist es so, dass viele den jetzt mittleren Bereich Tablet nur zum Fixen von Problemen verwenden und ihn sonst wenig berücksichtigen. Das hat natürlich auch stark mit der Nutzung der Website zu tun: wer kaum Tablet-User hat benötigt diese Anpassungen kaum.



Breakpoints

Aktive Breakpoints

Mobile Portrait x Mobile Landscape x Tablet Portrait x Tablet Landscape x Laptop x Breitbild

Die Optionen Mobile und Tablet können nicht gelöscht werden.

Mobile Portrait
Breakpoint (px) 767

Mobile Landscape
Breakpoint (px) 880

Tablet Portrait
Breakpoint (px) 1024

Tablet Landscape
Breakpoint (px) 1200

Laptop
Breakpoint (px) 1366

Breitbild
Breakpoint (px) 2400

Die Einstellungen für Breitbild Breakpoints gelten ab dem ausgewählten Wert und aufwärts.

Wenn man alle möglichen Breakpoints aktiviert kann man - und muss man - sehr viele Anpassungen vornehmen, damit das Design auch wirklich zum Endgerät passt.

8. Templates

Der Begriff *Template* wird mit verschiedenen Bedeutungen im Webdesign mit Elementor verwendet. In diesem Kapitel wird beschrieben welche Elementor Templates es gibt und wodurch sich diese unterscheiden.

Was ist ein Template?

Die wörtliche Übersetzung lautet *Vorlage*, doch dieser Begriff bringt uns leider nicht wirklich weiter. Einerseits kann man damit fertige Layouts mit Musterdaten meinen, auf der anderen Seite sind Templates Schablonen, durch die Inhalte dynamisch in eine Form gebracht werden. Es gibt also *statische* und *dynamische* Templates. Und ausserdem gibt es Templates die ganze Seiten umfassen, und dann wiederum welche, die nur Sektoren beschreiben. Und dann gibt es Templates noch auf der Ebene von WordPress bzw. dem gewählten Theme und dann noch die Elementor Templates selber.

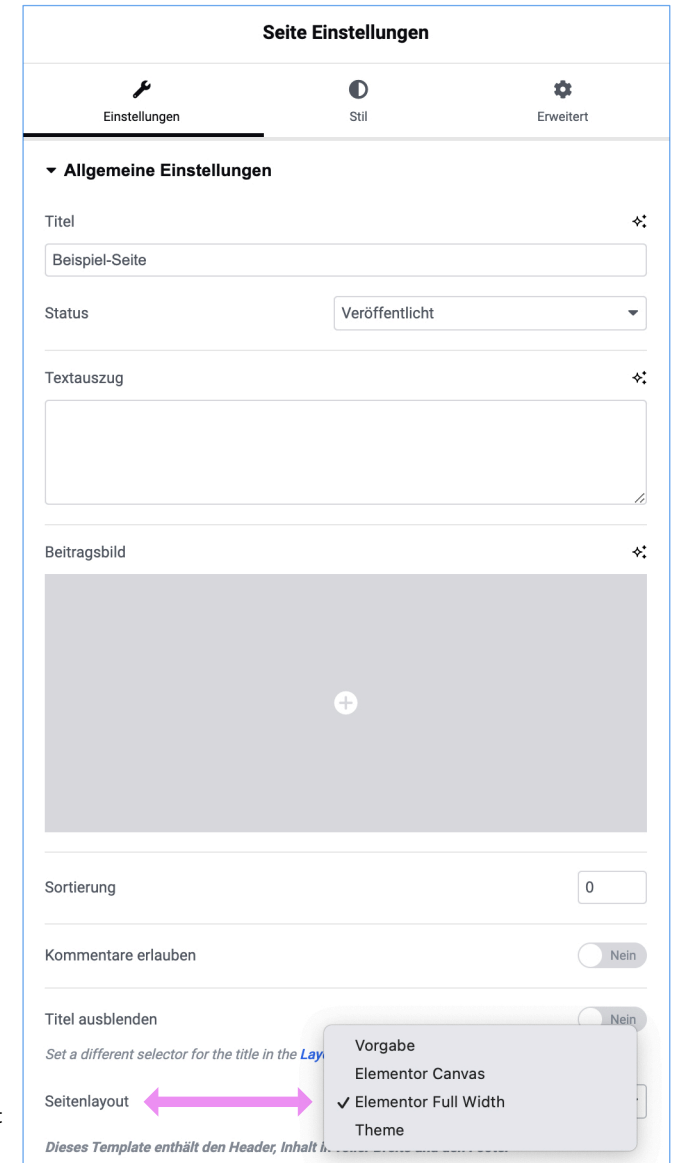
Es ist also genug da, um verwirrt zu werden. Beginn wir auf der ersten Ebene, in WordPress. Das Wesen eines CMS ist es, dass Form und Inhalt voneinander getrennt sind. Erst wenn man eine Webseite aufruft wird aus den Daten (die aus der Datenbank gelesen werden) und der Vorlage, wie diese Daten angezeigt werden sollen – ergänzt mit binären Daten (also zB. Bildern) – eine Seite errechnet. Diese wird an den Browser übermittelt der diese Daten dann aufzeichnet – also die Seite rendert.

8.1. WordPress Templates

Auf der untersten Ebene gibt es *Seitenlayouts*, die auch die Bezeichnung Templates tragen. Diese stammen vom Theme (oder von Elementor) und geben das Grundgerüst einer Seite vor.

Einstellen kann man das für jede Seite/jeden Beitrag getrennt entweder über die Seiteneinstellungen im Elementor (Zahnrad oben Mitte)

Panel > Seite Einstellungen > Allgemeine Einstellungen > Seitenlayout



Über *Seite Einstellungen* findet man im Elementor die als *Seitenlayout* bezeichneten Auswahl, wo das Template gewählt werden kann.

oder wenn man im WordPress Dashboard ist und eine Seite bearbeitet rechts unten unter

Seite > Templates

Damit legte man fest welches Seitenlayout=Template der aktuelle Beitrag oder die gerade bearbeitete Seite als Vorlage verwenden soll.

Elementor kennt zwei solcher Templates:

- Elementor Canvas
- Elementor Volle Breite (engl. Elementor Full Width)

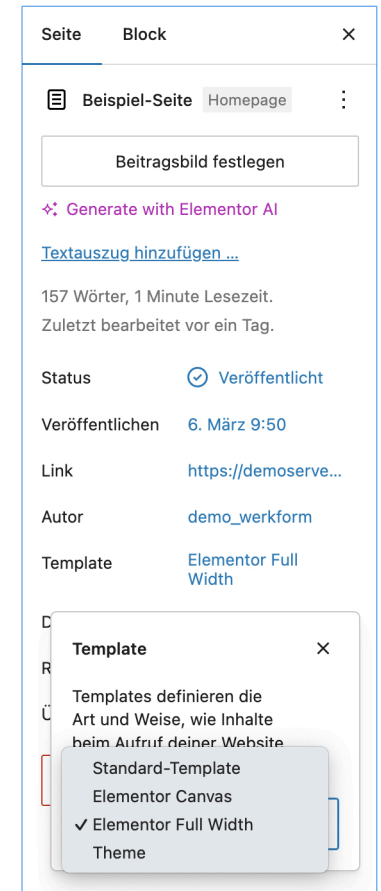
Wer *Elementor Canvas* wählt bekommt genau das, also eine leere Leinwand. *Elementor Volle Breite* hingegen ist eine Vorlage, welche die im Theme Builder (Kapitel 4) angelegten Seitenbestandteile wie Header und Footer anwendet. Dieses Level der Templates ist nicht mit einfachen Mitteln zu verändern – muss es auch nicht, denn ausser den beiden Möglichkeiten benötigt man im Regelfall auch nicht mehr. Die weiteren Optionen *Vorgabe* und *Theme* bedeuten, dass keine besondere Einstellung getroffen wird oder eben das Theme dafür sorgen soll, ein Template auszusuchen.

8.2. Die Elementor Bibliothek

Wer Google mit dem Suchbegriff Template bemüht, wird vermutlich eine große Zahl von bezahlten und darunter eine ganze Reihe von Suchmaschinen-optimierten Einträgen erhalten. Im Amateurnfeld werden darunter einzelne oder zusammenhängende Musterseiten verstanden, die je nach Qualität und Jahreszahl, mehr oder weniger aktuelles Webdesign mit Dummy-Daten und Bildern anbieten.

Diese kann man dann im Webdesign-Tool nach Wahl öffnen und verändern. Auch für Elementor gibt es natürlich eine Reihe solcher Angebote – nicht zuletzt von Elementor selber. Auch werden diese Templates als Kits angeboten.

Wenn man eine Seite nicht über Elementor sondern direkt im WordPress Dashboard bearbeitet findet man ebenfalls diese Einstellung. Hier heisst sie einfach Template. Wo man diese Einstellung vornimmt ist egal.



Dh. man bekommt nicht nur einzelne Seiten sondern die Bestandteile für ganze Websites, die man nur mehr mit eigenen Inhalt füllen muss.

Elementor nennt sein Angebot **Elementor Library**. (<https://elementor.com/library/>). Hier werden zusammenhängende Site Kits angeboten oder auch einzelne Bestandteile. Der Nutzung ist oft kostenlos, in vielen Fällen aber auch funktional an eine Pro-Lizenz gebunden. Das Angebot umfasst bereits mehr als 100 *Full Website Designs Ready To Go*. Jedes Kit kann online genau schon vor der Installation geprüft werden, seit Elementor 3.8 kann man jeweils das letzte installierte Kit auch wieder rückstandsfrei entfernen.

Elementor versucht die Paarung Website-Hosting und Website-Kite in der Vermarktung besonders zu betonen. Eine Antwort auf die weit verbreiteten Web-Baukästen - nur eben mit voller Elementor-Funktionalität.

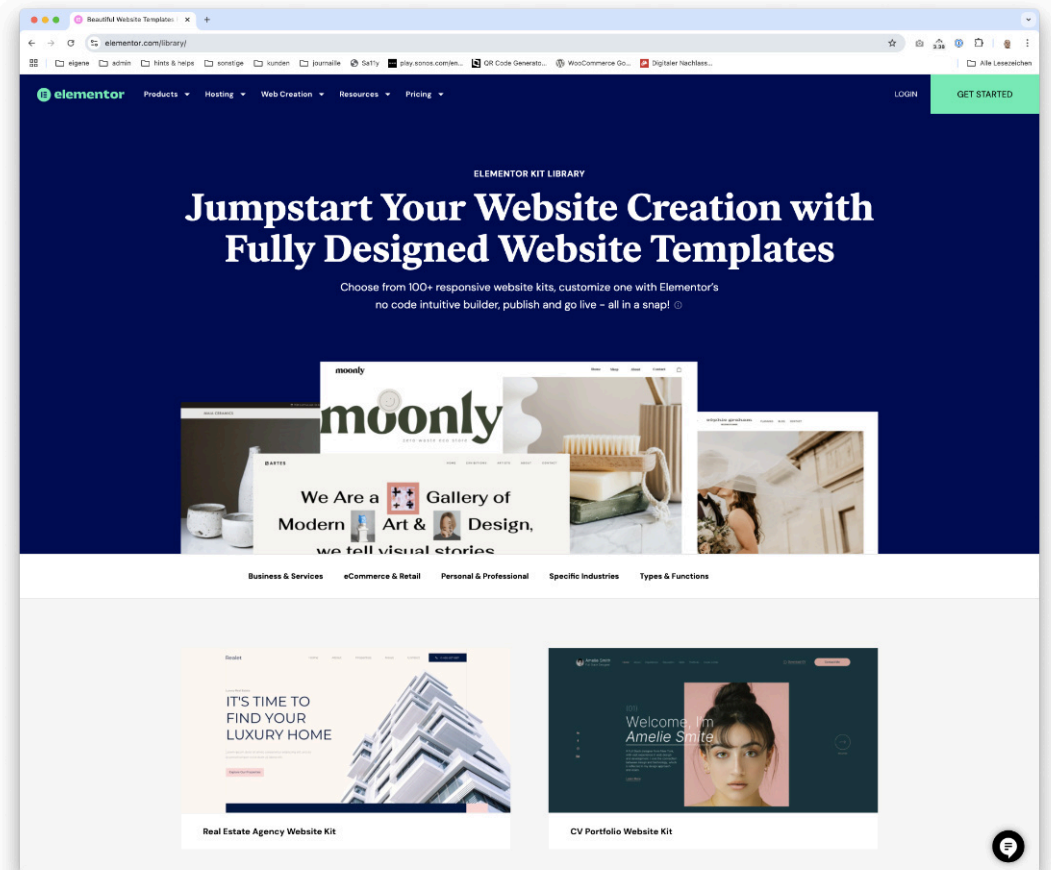
Wenn es schnell gehen muss oder so wirklich gar keine kreative Ader vorhanden ist mag die Nutzung fertiger Designs in Ordnung sein, wirklich passend wird es nur zufällig sein können. Es ist in etwa wie der Unterschied zwischen Maßanzug und Stangenware, nur dass bei der Stangenware auch nur ein oder zwei Größen vorhanden sind - und der Stil oft eher uninspiriert ist.

Trotzdem kann es sehr spannend sein von diesen Templates zu lernen. Wer mit Elementor arbeitet wird sicher schnell erkennen, dass nicht die ganzen Seiten sondern die einzelnen Blocks die spannendere Alternative sind.

Block Templates

Die Anwendung von Templates ist denkbar einfach: immer wenn man einen Container erstellt und wo ein Widget platzieren kann kann man auch ein Template einbinden. Bei einer leeren Seite ein Seitentemplate, wenn man nur eine Sektion gestalten möchte eben ein Block Template. Die Nutzung ist rein statisch:

1. mit dem Klick auf den Ordner
2. der Auswahl des Templates und
3. der Klick auf „Einbinden“



Die Elementor Library richtet sich an Nutzer, die die Funktionalität von Website Baukästen suchen. Sehr viele dieser Designs leben von der Bildqualität. Ob es wirklich zum eigenen Inhalt passt muss sorgfältig abgewogen werden.

werden die notwendigen Inhalt in die Datenbank bzw. in die Mediathek kopiert. Von da an wurde aus der Vorlage ein veränderbarer Inhalt, der wie gewohnt bearbeitet werden kann.

In der Elementor Library sind Blöcke zu verschiedenen Anwendungen zu finden, kategorisiert nach:

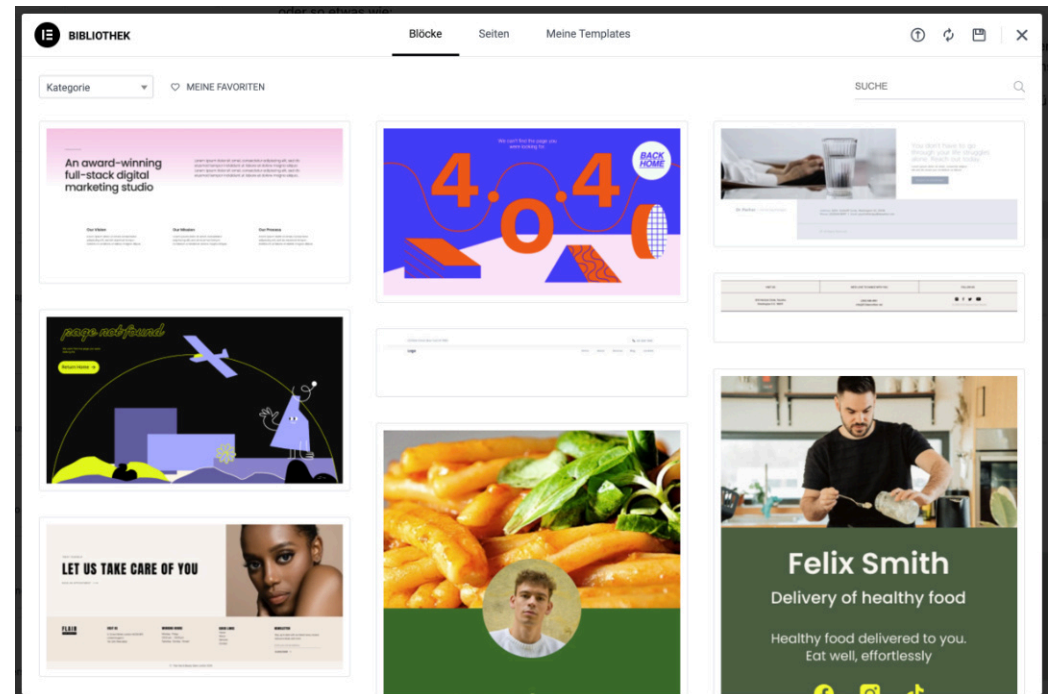
- | | |
|---------------|-----------------|
| About | Portfolio |
| Archive | Pricing |
| Benefits | Product Archive |
| Call 2 Action | Quote |
| Clients | Service Details |
| Contact | Services |
| Data | Single Page |
| FAQ | Single Post |
| Features | Single Product |
| Footer | Stats |
| Gallery | Subscribe |
| Header | Team |
| Hero | Testimonials |
| Link In Bio | 404 Page |

Wer seine Webseite strategisch gut geplant hat und die Einstellungen sorgfältig vorgenommen hat wird schnell aus einem Bibliotheksblock einen Bestandteil seiner Seite bauen können.

Manchmal kann es jedoch einfacher sein einen Block nachzubauen, als die mit dem Block mitgelieferten Farben und typographischen Details anzupassen. Technisch sauberer wird es wohl, wenn man mit möglichst wenigen Stilanpassungen vom leeren Container aus arbeitet.



Im Bearbeitungsmodus einer Seite kann man durch klicken auf das Ordner-Icon die Bibliothek mit den Blöcken aufrufen.



Nach Kategorie geordnet (oder eher aufgelistet) wählt man einen Block, der dann inkl. Bilder in die eigene Website hinein kopiert wird.

8.3. Eigene Elementor Templates erstellen

Es ist einfach eigene Blöcke oder Seiten für die weitere Verwendung in anderen Seiten und Beiträgen (oder auch Custom Posts) zu speichern. Nach dem Speichern steht das erstellte Template auch für den *Export/Import* zur Verfügung.

Eine Seite speichert man als Template, indem man über das kleine Pfeilchen rechts vom Speichern-Button das Popup öffnet und dort *Als Template Speichern* wählt. Man gibt einen sinnvollen Namen und schon ist es erledigt. Um einen einzelnen Abschnitt als *Block-Template* zu speichern muss man nur mittels Kontext-Menü die entsprechende Option wählen.

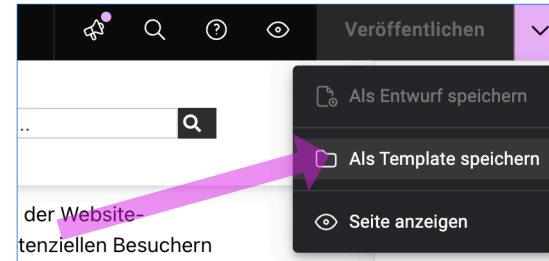
Alternativ dazu kann man auch den Weg über das Dashboard gehen:

Dashboard > Templates > Gespeicherte Templates > Neu hinzufügen

Nun muss man noch die Art des zu erstellenden Templates wählen:

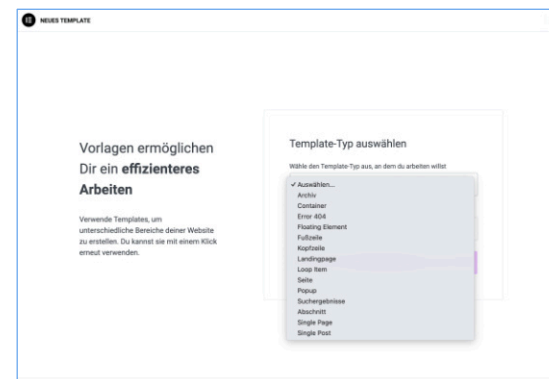
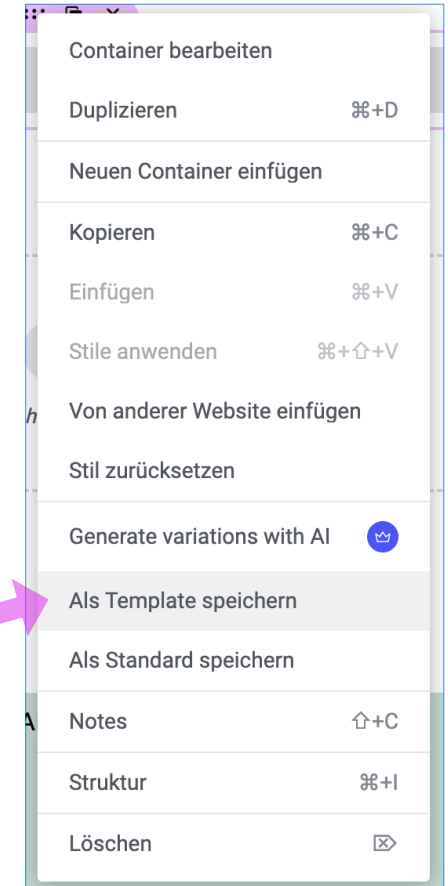
- Archiv
- Container
- Error 404
- Floating Element
- Fußzeile
- Kopfzeile
- Landingpage
- Loop Item
- Seite
- Popup
- Suchergebnisse
- Abschnitt
- Single Post
- Single Page

Jene Typen die zum Theme-Builder zählen haben mit *Bedingungen* (Conditions) eine eigene Logik.



Im Submenu rechts oben kann jedes Dokument auch als Template gespeichert werden. (oben)

Will man einen Container als Template speichern ist diese Option über das Context-Menü (rechts-Klick) direkt auf das Element selbst oder in der Strukturpalette zu erreichen. (rechts)



Auch der Weg über das WordPress Dashboard ist möglich, um ein neues Template anzulegen. Hier wird vorerst nicht unterschieden, ob es sich um ein Theme-Builder oder um ein herkömmliches Template handelt.

8.4. Globale Widgets

Eine besondere Form des Templates ist das Global Widget. Jedes Widget kann über Rechtsklick als *Global* gespeichert werden. Dieses taucht dann innerhalb der Templates wieder auf, angewandt werden kann es in der Widget-Übersicht wenn man dort auf *Global* wechselt.

Die Besonderheit ist, dass man ein solches Widget einmal erstellt und mehrfach einsetzen kann. Erfolgt eine Änderung dieses Widgets werden diese Änderungen auf alle Anwendungen durchgereicht.

Wenn man so ein Widget im Elementor anklickt muss man sich entscheiden, ob man es lösen will (also aus dem Globalen Widget wieder ein normales Widget macht) oder ob man die Änderung gleich für alle Instanzen durchführen will.

Häufige Anwendungen sind Call2Actions oder auch Kontaktinformationen, die oft geändert werden aber überall gleich sein sollten.

”

Harry unterstützt mich per Fernsupport bei meinen Website-Projekten. Die Distanz zwischen Österreich und Norddeutschland spielt dabei keine Rolle. Er arbeitet so anschaulich, als säße er direkt neben mir. Durch seine professionelle und zugleich humorvolle Art macht er jeden fachlichen Austausch zu einer kurzweiligen Trainingseinheit in Elementor Pro. Ich empfehle werkform.at allen, die ihre Wordpress-Website professionell voranbringen wollen.

Jan Deißner
Kommunikationsberater, Hamburg
www.jandeissner.de

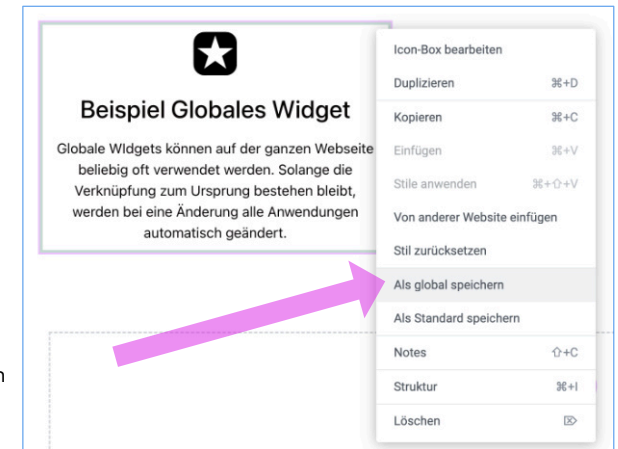


werk form

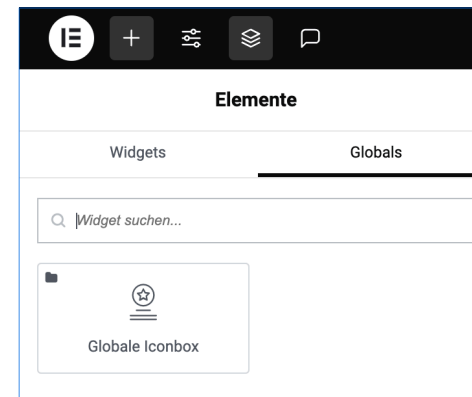
Erstgespräch buchen



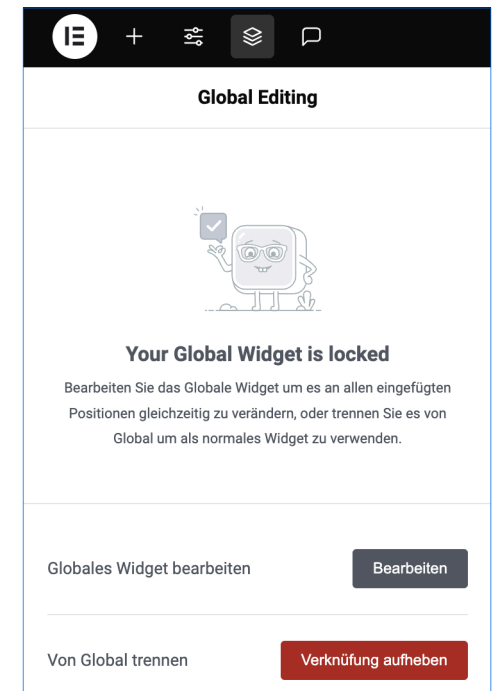
Anzeige



Jedes Widget lässt sich global speichern
- nicht jedoch aber Container.



Über den Tab Globals können die global gespeicherten Widgets in Seiten, Beiträgen und Templates integriert werden. Will man sie bearbeiten kann man wählen ob man alle Anwendungen des Widgets bearbeitet will oder man für das einzelne Widget die Verknüpfung aufhebt, wodurch es dann kein Globals Widget mehr ist.



8.5. Eigene Container als dynamische Templates

Es gibt neben der Möglichkeit eines Theme-Builder-Templates und des Globalen Widgets auch über Container (oder auch die früheren Abschnitte) eine dritte Variation, wie man *mehrfach genutzte Blöcke* erstellen und verwalten kann. Sollte eine Seite zu groß und ein Widget zu klein für den gewünschten Inhalt sein kann man einen Container als Template anlegen. (siehe auch Kapitel 8.3. Eigene Elementor Templates erstellen).

Dashboard > Templates > Gespeicherte Templates > *Container*

Container-Templates können die Lösung sein. Wenn man also zB. ein Iconlisten- und eine Google-Maps-Widget für einen Kontaktseitenblock zusammen in einem Container unter bringt kann diesen auf 2 Arten an jeder beliebigen Stelle unterbringen: entweder man verwendet das Widget Shortcode um dort über die ID des Templates dieses einzubinden:

```
[elementor-template id="115"]
```

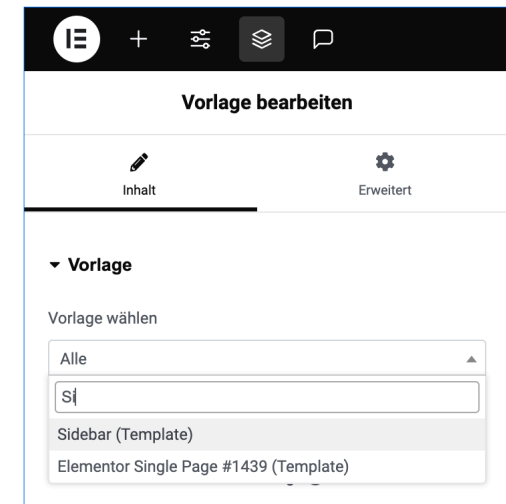
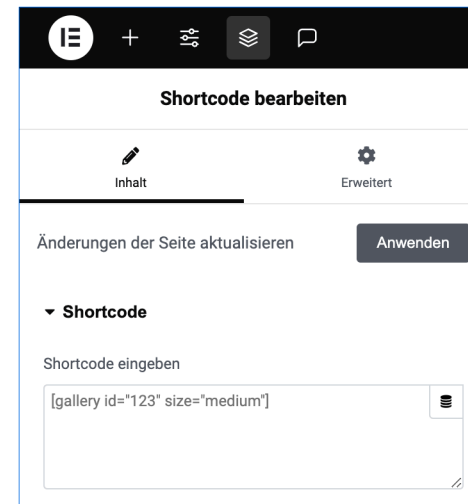
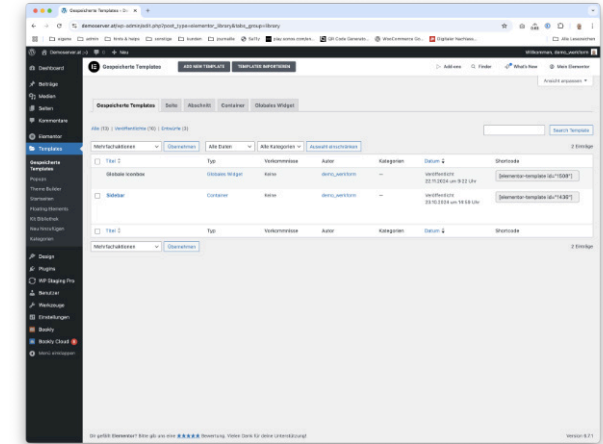
Die zweite Möglichkeit ist das Widget *Vorlage*, das genau dazu da ist, beliebige Templates an beliebigen Stellen der Website anzuzeigen.

Die so eingebundenen Container sind dann über das Dashboard oder über den Finder im Panel aufzurufen und zu bearbeiten.

Damit kann man jeden einzelnen Bereich einer Website durch zentral verwaltete Templates steuern. Es ist also nicht notwendig, dass irgendein Inhalt mehr als 1 x pro Website erstellt wird.

In der Template-Übersicht wird jedes Template auch mit dem Shortcode aufgelistet. Diesen kann man über das Widget Shortcode und auch den Block Shortcode auf Gutenberg-Seiten einbinden.

Ausserdem gibt es das Widget Vorlage, mit dem man aus den vorhandenen Templates wählen kann.



8.6. Popups

Dieser Template-Typ hat 2 Besonderheiten: 1. muss man - im Unterschied zu Seiten und Beiträgen - die Größe und den Hintergrund festlegen. 2. muss man festlegen, wann das Popup sichtbar sein soll.

Die Verwendungsmöglichkeiten sind vielfältig: natürlich werden Popups besonders von Marketing-Abteilungen und Werbetreibenden geschätzt um - mehr oder weniger passend zum Besucher der Website und der Seite - Produktbotschaften, Newsletter-Anmeldungen etc. in der Vordergrund zu rücken.

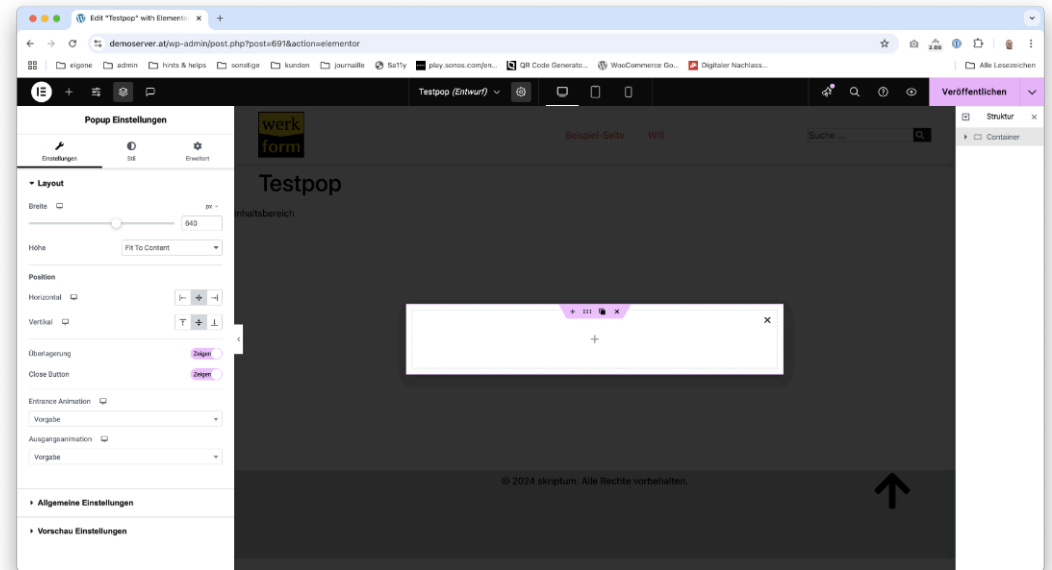
Da man Popups aber nicht nur „automatisch“ sondern auch aktionsbedingt (im Normalfall durch Klick) anzeigen kann wurden und werden sie gerne auch für Menüs verwendet, die bei Bedarf eingeblendet werden können. Diese Anwendungsform war sehr erfolgreich, daher hat Elementor das recht neue Widget *Off Canvas* spendiert, mit dem die selbe Funktionalität erreicht werden kann. Angelegt und verwaltet werden Popups über

[Dashboard](#) > [Templates](#) > [Popups ...](#)

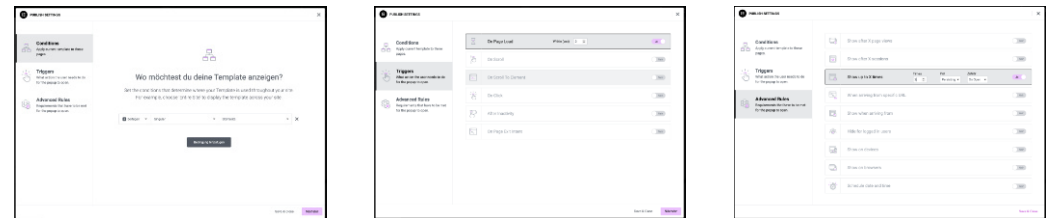
Weil es ja für Popups kein Seitentemplate gibt muss man sich die Dimensionen, Ein- und Ausgangsanimationen, Hintergrund, Rahmen, Überlagerung und Positionierung überlegen.

Nach der Gestaltung gilt es zu definieren, wann ein Popup erscheinen soll. Dies kann automatisch mit den Bedingungen und Triggern geschehen, oder man beschränkt sich darauf ein Popup nur über eine Aktion wie einen Klick einzublenden (in einem Linkfeld über *Dynamische Attribute* > *Actions* *Popup*).

Damit kann man ein Regelset erstellen wie zum Beispiel:
zeige das Popup auf der Startseite, 2 Sekunden nach dem Seitenaufruf, aber nur maximal 2 Mal und nicht öfter.



Dort wo Seiten eingestellt werden (Zahnrad oben Mitte) gelangt man auch zu den Popup-Einstellungen. Hier werden Breite und Höhe, Position, Überlagerung und Animationen angegeben.



Die Einstellungen für automatische Popups sind 3geteilt: zuerst die *Conditions* (Bedingungen), wie man sie auch aus dem Theme-Builder kennt. Danach den/die *Trigger*, die zum Einblenden führen und schliesslich zusätzliche Regeln (*Advanced Conditions*).

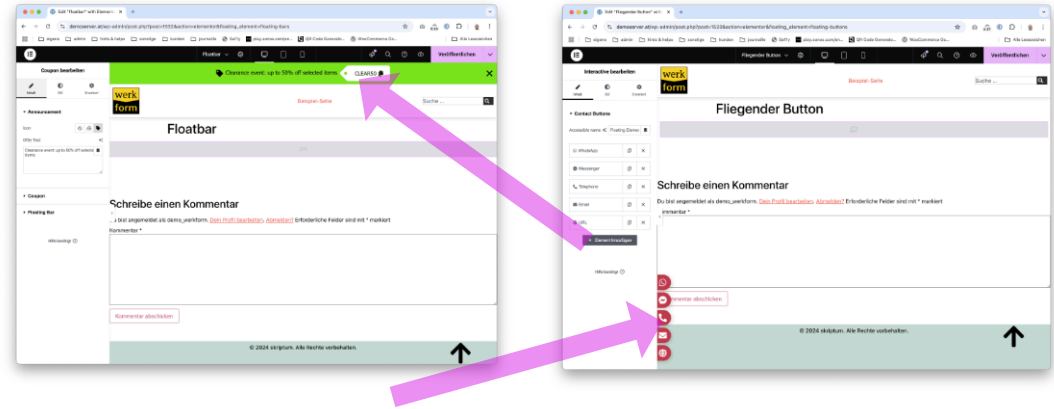
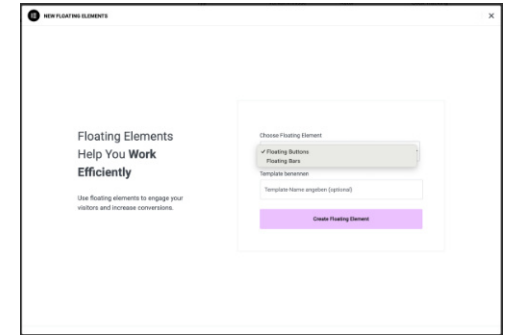
8.7. Floating Elements

Dieser neue Template-Typ hat im Augenblick 2 Anwendungsmöglichkeiten: **Floating Bars** (Coupon) und **Floating Buttons** (Interactives). Damit kann man Coupon-Codes anzeigen die - wahlweise am oberen oder unteren Fensterrand - über dem eigentlichen Inhalt der Seite eingeblendet werden können. Das zweite Einsatzgebiet sind Kontaktbuttons, die man an allen 4 Fensterseiten jeweils links, zentriert und rechts anzeigen lassen kann.

Die Inhalte dieser Templates sind entsprechend der vorgesehenen Anwendungen einstellbar. Die Einstellmöglichkeiten im Stil dieser beiden Elementarten sind erwartbar logisch wie immer in Elementor. Die gewünschte Position stellt man im Erweitert-Tab ein.

Auch bei den Bedingungen zur Anzeige gibt es keine Überraschungen: wie bei den Templates, Popups etc. kann man für jedes einzelne Floating Element genau definieren, wo es auftauchen soll.

Floating Elements sind ein Template-Typ, der über den eigentlichen Inhalten Couponbars oder Kontaktbuttons ermöglicht.



Interactive bearbeiten

Inhalt
Stil
Erweitert

▼ **Layout**

Horizontale Position ← | → | ⇄

Versatz px ▾

Vertikale Position ↑ | ↓ | ⇅

Versatz px ▾

▼ **Responsiv**

Coupon bearbeiten

Inhalt
Stil
Erweitert

▼ **Layout**

Vertikale Position ↑ | ↓

Sticky

► **Responsiv**

► **CSS**

► **Eigenes CSS**

► **Attribute**

Website-Heroes
Die WP Business Community

Weiterbildung, Support und Community für WordPress.

Website Heroes lernen laufend dazu und helfen sich gegenseitig. Live via Zoom oder online im Forum.

60+
Aufzeichnungen

111+
5★-Bewertungen

3.000+
Beiträge im Forum

www.websiteheroes.co

Anzeige

Neben den logischen Optionen bei Inhalt und Stil findet man unter Erweitert die Positionierungs-Einstellungen.

werkform Pro Skriptum

90

900 V 3.0

9. Schriften

Wer je in einem Layout-Programm oder einem Zeichenprogramm gearbeitet hat ist verwöhnt: Einstellungsmöglichkeit bis hinunter in Bruchteile von Millimetern, Zeilenhöhen, Spationierungen... eine Jahrhunderte alte Schriftsetzer Tradition hat seinen Weg in die Computer gefunden. Und dann kam das Web – und alles war weg! Nur mühsam mit der Weiterentwicklung von CSS kamen und kommen langsam die Tugenden einer guten Typographie zurück in den Alltag des Screendesigns.

Leseverhalten und technische Voraussetzungen (Handys!) beeinflussen das Nutzerverhalten stark. Es gibt eine große Anzahl an guten Schriften, die auch im Web das Lesen erleichtern. Ausserdem ist eine gute Typographie ein Kernbestandteil eines funktionierenden Corporate Designs.

Ins Detail: was kann man zB. innerhalb des Widgets *Überschrift* alles gestalten? Welche Änderung führt wo hin?

Der HTML-Tag

Der gewählte HTML-Tag wirkt sich dann auf die Gestaltung aus, wenn keine anderen Einstellungen getroffen wurden. Ja nach Theme und Grundeinstellungen wird der Text formatiert. Es kommt dann zu einer Änderung des Aussehens, wenn man von zB. H2 auf H4 wechselt. So ferne man diese über die *Einstellungen > Typographie* (Kapitel 2.2) definiert, aber keinen Globalen Schriftstil zugeordnet hat.

Wurde aber eine Globale Schrift ausgewählt ist die Darstellung unabhängig vom gewählten HTML-Tag.

Der Tab Stil

Textfarbe muß nicht erklärt werden. Wie immer bei Farbwählern kann man hier aus den *Globalen Farben* wählen.

Der gewählte HTML-Tag kann, muss aber keine Auswirkung auf das Erscheinungsbild der Überschrift haben.

The screenshot shows the 'Überschrift bearbeiten' (Header Edit) interface. At the top, there are three tabs: 'Inhalt' (Content), 'Stil' (Style), and 'Erweitert' (Advanced). The 'Inhalt' tab is selected. Below the tabs, there is a section for 'Überschrift' (Header) with a dropdown arrow. Underneath, there are three input fields: 'Titel' (Title) with the placeholder text 'Beispielüberschrift', 'Link' (Link) with the placeholder text 'URL einfügen oder lostippen', and 'HTML-Tag' (HTML Tag) with a dropdown menu currently set to 'H2'. A pink arrow points from the text below to the 'HTML-Tag' dropdown.

The screenshot shows the 'Überschrift bearbeiten' (Header Edit) interface with the 'Stil' (Style) tab selected. The 'Inhalt' tab is now greyed out. The 'Überschrift' section is expanded, showing several styling options: 'Ausrichtung' (Alignment) with four icons, 'Textfarbe' (Text Color) with a color picker, 'Typografie' (Typography) with a font icon, 'Textumrandung' (Text Shadow) with a shadow icon, and 'Text Shadow' with a shadow icon. At the bottom, there is a 'Überblendmodus' (Blend Mode) dropdown set to 'Normal'.

Die Einstellungsmöglichkeiten zur Schrift sind in sehr vielen Widgets zu finden und entsprechen ± immer jenen der Überschrift.

Typographie zeigt ein Popup das es in sich hat. Siehe unten (9.1.).

Textumrandung zeichnet eine Outline um die Buchstaben, die in Farbe und Stärke eingestellt werden kann.

Der **Textschatten** erlaubt es eben genau so einen zu definieren. Eingestellt werden kann Farbe und Unschärfe und die vertikale sowie horizontale Ausbreitung des Schattens.

Überblendmodus, hier lassen sich die CSS Eigenschaften `mix-blend-mode` einstellen um verschiedenen Effekte mit zB: Hintergrundbildern oder Hintergrundfarben zu erzielen.

Diese Stil-Einstellungen können beliebig kombiniert werden. Dementsprechend vorsichtig sollte man diese Einstellungen treffen um das Wichtigste an Text - die Lesbarkeit - zu fördern und nicht zu stören.

9.1. Typographie

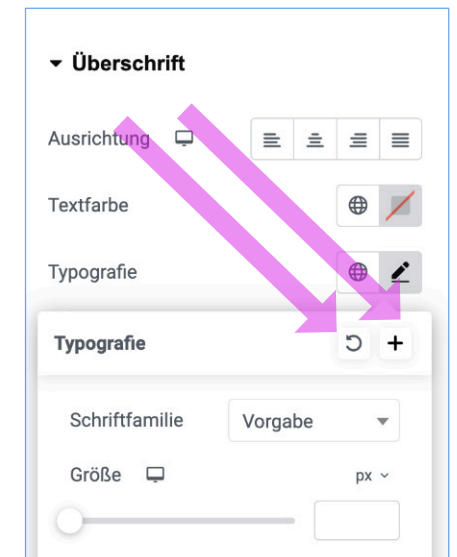
Ganz oben die zwei Icons erlauben das zurücksetzen der getroffenen Einstellungen bzw. [+] bieten eine Abkürzung zu den Grundeinstellungen, um dort eine Globale Schriftart festzulegen. Die einzelnen Punkte:

- Schriftfamilie
- Größe
- Schriftschnitt
- Stil
- Umwandlung
- Auszeichnung
- Zeilenhöhe
- Zeichenabstand
- Wortabstand

werden nun im Detail besprochen. Responsive Einstellungen können für die



Textumrandung und Textschatten: bei großen Überschriften kann damit mit sehr wenig Aufwand ein spannendes Design geschaffen werden.



Wer sich zu weit von den Grundeinstellungen entfernt hat kann die Typographie zurücksetzen. Oder man kann auch getroffenen Einstellungen in einen Globalen Schriftstil übernehmen.

Größe, die Zeilenhöhe. Laufweite und den Wortabstand vorgenommen werden, nicht aber für die restlichen Angaben. Hier muß man sich ggf. mit CSS weiterhelfen.

Schriftfamilie

Die Auswahlmöglichkeiten hier werden unterteilt in *Vorgabe*, *Eigene Schriften* (falls vorhanden) *System* und *Google*. Wenn *Adobe Fonts* aktiviert wurden (kostenpflichtig) dann stehen diese auch bereit. Je nachdem, welche Schriften man verwendet kann es ev. notwendig sein, dies auch in der Datenschutzerklärung der Webseite anzuführen bzw. die Einwilligung (Consent) des Users einzuholen, bevor die Schrift geladen wird. Der Hintergrund ist der, dass Schriften ggf. von einem anderen Server geladen werden. Es wird dringend empfohlen, Schriften lokal einzubinden. (Kapitel 3.3). Dazu besorgt man sich die Schriftdateien (besonders *woff2* ist wichtig!) und legt die Schrift unter

Dashboard > Elementor > Benutzerdefinierte Schriftarten

an. Elementor erlaubt jetzt auch *Variable Fonts*, wobei hier im Augenblick nur 2 Achsen unterstützt werden. Mit Verbesserungen in diesem Bereich ist laufend zu rechnen.

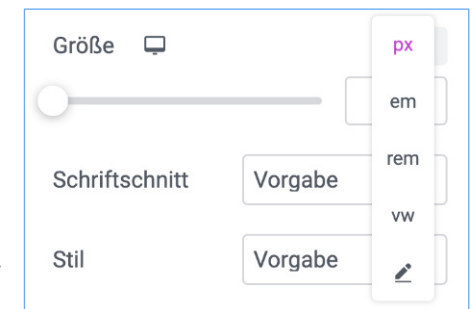
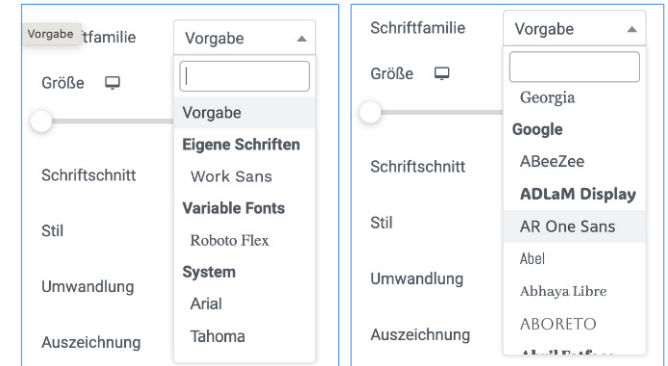
Größe

Zuerst die Einheit zu wählen, in der man die Schriftgröße einstellen will. Es stehen

- PX (Pixel)
- EM (Ems)
- REM (Root Ems)
- VW (Viewport With)
- Custom (= freie Eingabe)

zur Auswahl. Während die Angabe in **px** absolut ist beziehen sich **em/rem** auf die Größe des Elternelements und **vw** auf die Breite des Viewports in 1/100

Die Auswahl unter Schriftfamilie ist abhängig von den installierten Fonts. Neben den immer vorhandenen **System**schriften werden **Eigene Schriften** (=statische Fonts), **Variable Fonts** und **Google Fonts** unterstützt (wovon hier ausdrücklich abgeraten wird - besser lokal speichern!).



Die Größenangaben können in allen sinnvollen Einheiten eingegeben werden.

Einheiten. Alles bis auf px ist relativ. Je nach Einheit erlaubt Elementor nur ganzzahlige Werte (px) oder limitiert auch die maximale Größe (10 bei vw). Diese Einschränkungen wurden sinnvoll gewählt.

Mittels Schieberegler oder im Eingabefeld daneben wird die Wunschgröße ausgewählt. Für die responsiv Modi können unterschiedliche Größen und Einheiten gewählt werden.

Schriftschnitt

Mit dem Schriftschnitt legt man fest wie „fett“ die Buchstaben sein sollen, von 100 (ganz dünn) bis 900 (ganz dick), oder aber auch *Normal* oder *Bold*. Was ist jetzt aber der Unterschied zwischen Normal und 400? Oder Bold und 700? Einfach zusammengefasst kann man sagen:

Wenn nichts eingestellt wurde wird die Schrift normal dargestellt, das entspricht dem Wert 400. Die Einstellung bold (fett) entspricht dem Wert 700. Schriftschnitte mit numerischen Werten (eigentlich 1-1000, in der Praxis aber die 9 Stufen der hunderter-Schritte) bieten genauere Abstufungen. Natürlich müssen diese Schriftschnitte auch geladen sein damit diese dargestellt werden können.

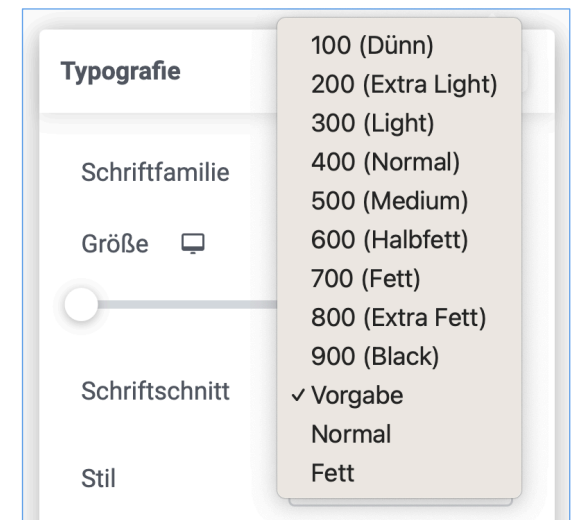
Wird ein Schriftschnitt ausgewählt der nicht geladen wurde ersetzt das System diese Schrift durch die ähnlichste geladene, also wenn zB. der Schnitt 600 ausgewählt wurde aber nur 700 geladen wurde wird auch 700 dargestellt.

Bei der Wahl der Schriftschnitte ist darauf zu achten, dass jeder einzelne Schnitt ein File lädt. Bei vielen Schriften und Schnitten erfolgen also eine große Zahl an Serveraufrufen, das bremst, auch wenn die Schriften selber oft nur sehr kleine Files benötigen.

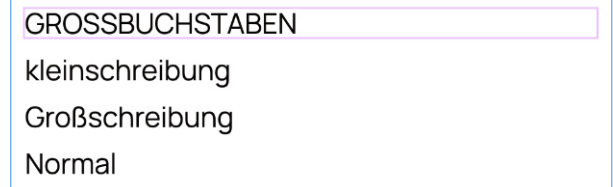
Umwandlung

Bei Umwandlung werden die CSS-Möglichkeiten von `text-transform` angewandt. VERSALIEN, klein- und Großschreibung sind selbsterklärend. *Normal* bedeutet,

Der Schriftschnitt kann in 100er-Schritten ausgewählt werden. Normal entspricht dem Wert 400, Fett 700. Vorgabe bedeutet, dass an dieser Stelle keine Definition erfolgt.



Die Umwandlungsmöglichkeiten sind selbsterklärend. **Normal** bedeutet Normalschrift während **Vorgabe** bedeutet, dass keine Umwandlung vorgenommen wird.



dass `text-transform: none`; gesetzt wird (im Unterschied zu *Vorgabe*, wo kein Wert angegeben wird).

Auszeichnung

Wer eine Auszeichnung benötigt hat hier die Möglichkeit zwischen *Unter-*, *Über-* und *Durchstreichen* zu wählen. *Keine* bedeutet hier wieder, das *none* gesetzt wird.

Stil

Unter dem Punkt Stil kann man die CSS-Angaben für *font-style* vornehmen. Auch hier wieder: *Vorgabe* setzt keinen Wert während *Normal* den Wert *none* setzt.

Der Unterschied zwischen Kursiv (`font-style: italic;`) und Schräg gestellt (`font-style: oblique`) ist jener, dass im Falle einer geladenen passenden Schrift bei Kursiv ein eigener Font verwendet wird. Schräg gestellt macht genau was man vermutet: es verzerrt die Schrift (normalerweise um 14°). Dieses *oblique* ist übrigens auch der Fallback zu kursiv. Ausserdem bietet CSS die Möglichkeit hier einen anderen Winkel als 14° einzustellen.

Zeilenhöhe

Bei der Wahl Zeilenhöhe kommt es für eine gute Lesbarkeit auf die Schriftgröße an. Die echte **Zeichenhöhe** kann bei unterschiedlichen Schriften bei der selben numerischen Angabe verschieden hoch sein. Man kann keine allgemeinen Angaben machen, wie das Verhältnis zwischen Textgröße und Zeilenabstand zu wählen ist. Das Beispiel zeigt bei einer gewählten Schriftgröße von 1 rem die Zeilenabstände von 1 rem, 1,5 rem und 2 rem.

Wo wirkt nun die Zeilenhöhe auf den Text? Vereinfacht gesprochen wird die angegebene Zeilenhöhe halbiert und über der Oberlänge und unter der Unterlänge der Buchstaben addiert. Daher rutscht bei einer größeren Zeilenhöhe auch der Inhalt innerhalb des Blocks nach unten, wenn die Zeilenhöhe niedriger als die Schriftgröße ist auch hinauf.

Linien sind als Unter- und Überstrich sowie als Durchstreichen möglich.

Unterstreichen

Überstrich

~~Durchstreichen~~

Während bei der oberen Schrift zwei verschiedene Schnitte für normal und kursiv zuständig sind, wird beim unteren die Schrift schräg gestellt. Kursive sind besser.

normal

kursiv

normal

schräg gestellt

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut elit tellus, luctus nec ullamcorper mattis, pulvinar dapibus leo. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut elit tellus, luctus nec ullamcorper mattis, pulvinar dapibus leo.

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut elit tellus, luctus nec ullamcorper mattis, pulvinar dapibus leo. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut elit tellus, luctus nec ullamcorper mattis, pulvinar dapibus leo.

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut elit tellus, luctus nec ullamcorper mattis, pulvinar dapibus leo. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut elit tellus, luctus nec ullamcorper mattis, pulvinar dapibus leo.

Bei einer 1 rem großen Schrift die Zeilenhöhen 1 rem, 1,5 rem und 2 rem. Die Grundlinie der ersten Zeile rutscht die halbe Zeilenhöhe nach unten (siehe roter Strich bzw. Abstand dazu).

Zeichenabstand

Mit dem Zeichenabstand kann man den Raum zwischen den Buchstaben einstellen. Dies wird mittels der CSS-Eigenschaft letter-spacing umgesetzt. Elementor erlaubt hier Werte von -5 Pixel bis + 10 Pixel (em, rem), die Voreinstellung ist ± 0 .

Wortabstand

Wer die Einstellungen zum Wortabstand verändert sorgt damit dafür, dass nach jedem Wort (inkl. eines ggf. anschließenden Satzzeichens) der Abstand nach dem Wort verändert wird. Dh. der vom Schriftdesigner ausgewählte Abstand (=Leerzeichen) wird durch den eingegebenen Wert verändert. 0 oder kein Wert bedeutet einen unveränderten Wortabstand.

9.2. Mehrspaltiger Text

Wenn man das Widget *Texteditor* verwendet kann man auch die *Spaltenanzahl* einstellen, auf die der Text innerhalb des Blocks aufgeteilt werden soll. Anzahl und Abstand der Spalten kann responsiv extra eingestellt werden. Ausdrücklich gewarnt wird hier vor der Kombination *schmale Spalten* und *Blocksatz*.

Im Unterschied zu Layout-Programmen wie Indesign, XPress oder Affinity Publisher beherrscht das Web (CSS) die Details eines qualitativen Satzsetzes nicht. Daher *besser keinen Blocksatz* (also Ausgerichteten Text, align: justify;) verwenden. Ein harmonischeres - und damit besser lesbares - Schriftbild entsteht bei ausreichend breiten Spalten (also genug Anschlägen) in Kombination mit Flattersatz (=linksbündiger Text) und einer aktivierten Silbentrennung.

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut elit tellus, luctus nec ullamcorper mattis, pulvinar dapibus leo. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut elit tellus, luctus nec ullamcorper mattis, pulvinar dapibus leo.

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut elit tellus, luctus nec ullamcorper mattis, pulvinar dapibus leo. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut elit tellus, luctus nec ullamcorper mattis, pulvinar dapibus leo.

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut elit tellus, luctus nec ullamcorper mattis, pulvinar dapibus leo. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut elit tellus, luctus nec ullamcorper mattis, pulvinar dapibus leo.

Der **Zeichenabstand** links 0, in der Mitte 0,05 rem und rechts 0,1 rem.

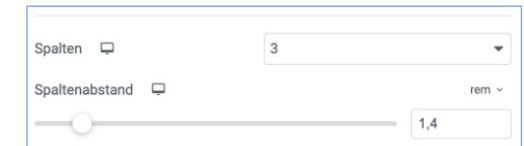
Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut elit tellus, luctus nec ullamcorper mattis, pulvinar dapibus leo. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut elit tellus, luctus nec ullamcorper mattis, pulvinar dapibus leo.

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut elit tellus, luctus nec ullamcorper mattis, pulvinar dapibus leo. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut elit tellus, luctus nec ullamcorper mattis, pulvinar dapibus leo.

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut elit tellus, luctus nec ullamcorper mattis, pulvinar dapibus leo. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut elit tellus, luctus nec ullamcorper mattis, pulvinar dapibus leo.

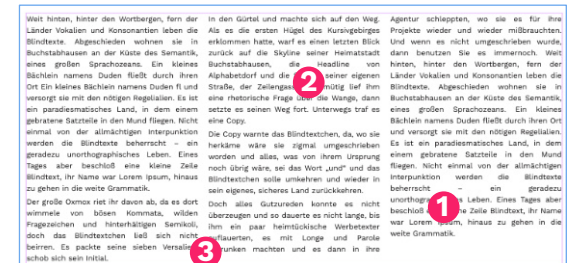
Der **Wortabstand** links 0, in der Mitte 5 px und rechts 10 px.

Texte können einfach auch mehrspaltig angezeigt werden. Da diese Einstellungen responsive-sensibel sind kann man das für schmale Bildschirme leicht verhindern.



Nur auf den ersten Blick scheint ein mehrspaltiger Blocksatz aufgeräumt und ordentlich. Im Detail sind schnell einige Fehler zu erkennen:

1. Der eingestellte Spaltenabstand ist zu gering, wenn die Lücken zwischen den Worten Breiter als die Spalten sind.
2. Da nur der Wortabstand erhöht wird entstehen große Löcher im Schriftbild.
3. Die Grundlinien der Spalten sind nicht gleich, der Text „springt“.



9.3. Automatische Silbentrennung

Da (besonders) die deutsche Sprache sehr lange Wörter zulässt ist sowohl für den mehrspaltigen Text als auch für schmale Spalten (responsive Design!) Silbentrennung eine gute Sache. Seit CSS3 gibt es das, ob und wie das unterstützt wird liegt am Browser. Die gültige CSS-Anweisung lautet:

```
hyphens: auto;
```

was von Chrome und Firefox verstanden wird. Damit auch Safari damit umgehen kann und auch ein paar andere Browser ist

```
selector {
  hyphens: auto;
  -ms-hyphens: auto;
  -o-hyphens: auto;
  -webkit-hyphens: auto;
}
```

ein möglicher CSS-Code. Das Beispiel hier verwendet *selector*, also jenen speziellen Begriff den man bei einem Widget unter

... Erweitert > Eigenes CSS

eingeben kann, damit das aktuelle Widget diesen zusätzlichen CSS-Code interpretiert.

Wer sich den Weg über CSS sparen will kann auch das sogenannte *Weiche Trennzeichen* (Soft Hyphen, Bedingter Trennstrich) einsetzen. Dieses hat den Zeichen-Code

```
&shy; bzw. &#173;
```

Ob und wo man dieses Zeichen auf der Tastatur findet ist vom Betriebssystem abhängig.

Für automatische Silbentrennung muß CSS angewandt werden.

▼ **Eigenes CSS**

Add your own custom CSS ✕

```

1 selector {
2   hyphens: auto;
3   -ms-hyphens: auto;
4   -o-hyphens: auto;
5   -webkit-hyphens: auto;
6   }
7

```

Use *custom CSS* to style your content or add the "selector" prefix to target specific elements.

<p>Weit hinten, hinter den Wortbergen, fern der Länder Vokalien und Konsonantien leben die Blindtexte. Abgeschieden wohnen sie in Buchstaben an der Küste des Semantik, eines großen Sprachozeans. Ein kleines Bächlein namens Duden fließt durch ihren Ort. Ein kleines Bächlein namens beherrscht – ein geradezu unorthographisches Leben. Eines Tages aber beschloß eine kleine Zeile Donaudampfschiffahrtakapitänsmütze Blindtext, ihr Name war Lorem Ipsum, hinaus zu gehen in die weite Grammatik.</p> <p>Der große Oxmox riet ihr davon ab, da es dort wimmelte von bösen Kommata, wilden Fragezeichen und hinterhältigen</p>	<p>Semikoli, doch das Blindtextchen ließ sich nicht beirren. Es packte seine sieben Versalien, schob sich sein Initial.</p> <p>In den Gürtel und machte sich auf den Weg. Als es die ersten Hügel des Kursivgebirges erklimmen hatte, warf es einen letzten Blick zurück auf die Skyline seiner Heimatstadt Buchstaben, die Headline von Alphabetdorf und die Subline seiner eigenen Straße, der Zeilengasse. Wehmütig lief ihm eine rhetorische Frage über die Wangen, dann setzte es seinen Weg fort. Unterwegs traf es eine Copy.</p> <p>Die Copy warnte das Blindtextchen, da, wo sie</p>	<p>herkäme wäre sie zimal umgeschrieben worden und alles, was von ihrem Ursprung noch übrig wäre, sei das Wort "und" und das Blindtextchen solle umkehren und wieder in sein eigenes, sicheres Land zurückkehren.</p> <p>Doch alles Gutzureden konnte es nicht überzeugen und so dauerte es nicht lange, bis ihm ein paar heimtückische Werbetexter auflauerten, es mit Longe und Parole betrunken machten und es dann in ihre Agentur schleppten, wo sie für ihre Projekte wieder und wieder mißbrauchten. Und wenn es nicht umgeschrieben wurde, dann benutzen Sie es immernoch. Weit hinten, hinter den Wortbergen, fern der Länder</p>	<p>Vokalien und Konsonantien leben die Blindtexte. Abgeschieden wohnen sie in Buchstaben an der Küste des Semantik, eines großen Sprachozeans. Ein kleines Bächlein namens Duden fließt durch ihren Ort und versorgt sie mit den nötigen Regellalien. Es ist ein paradisiematisches Land, in dem ein gebratene Satzstelle in den Mund fliegen. Nicht einmal von der allmächtigen Interpunktion werden die Blindtexte beherrscht – ein geradezu unorthographisches Leben. Eines Tages aber beschloß eine kleine Zeile Blindtext, ihr Name war Lorem Ipsum, hinaus zu gehen in die weite Grammatik.</p>
--	---	--	---

<p>Weit hinten, hinter den Wortbergen, fern der Länder Vokalien und Konsonantien leben die Blindtexte. Abgeschieden wohnen sie in Buchstaben an der Küste des Semantik, eines großen Sprachozeans. Ein kleines Bächlein namens Duden fließt durch ihren Ort. Ein kleines Bächlein namens beherrscht – ein geradezu unorthographisches Leben. Eines Tages aber beschloß eine kleine Zeile Donaudampfschiffahrtakapitänsmütze Blindtext, ihr Name war Lorem Ipsum, hinaus zu gehen in die weite Grammatik.</p> <p>Der große Oxmox riet ihr davon ab, da es dort wimmelte von bösen Kommata, wilden Fragezeichen und hinterhältigen Semikoli,</p>	<p>doch das Blindtextchen ließ sich nicht beirren. Es packte seine sieben Versalien, schob sich sein Initial.</p> <p>In den Gürtel und machte sich auf den Weg. Als es die ersten Hügel des Kursivgebirges erklimmen hatte, warf es einen letzten Blick zurück auf die Skyline seiner Heimatstadt Buchstaben, die Headline von Alphabetdorf und die Subline seiner eigenen Straße, der Zeilengasse. Wehmütig lief ihm eine rhetorische Frage über die Wangen, dann setzte es seinen Weg fort. Unterwegs traf es eine Copy.</p> <p>Die Copy warnte das Blindtextchen, da, wo sie herkäme wäre</p>	<p>sie zimal umgeschrieben worden und alles, was von ihrem Ursprung noch übrig wäre, sei das Wort "und" und das Blindtextchen solle umkehren und wieder in sein eigenes, sicheres Land zurückkehren.</p> <p>Doch alles Gutzureden konnte es nicht überzeugen und so dauerte es nicht lange, bis ihm ein paar heimtückische Werbetexter auflauerten, es mit Longe und Parole betrunken machten und es dann in ihre Agentur schleppten, wo sie es für ihre Projekte wieder und wieder mißbrauchten. Und wenn es nicht umgeschrieben wurde, dann benutzen Sie es immernoch. Weit hinten, hinter den</p>	<p>Vokalien und Konsonantien leben die Blindtexte. Abgeschieden wohnen sie in Buchstaben an der Küste des Semantik, eines großen Sprachozeans. Ein kleines Bächlein namens Duden fließt durch ihren Ort und versorgt sie mit den nötigen Regellalien. Es ist ein paradisiematisches Land, in dem ein gebratene Satzstelle in den Mund fliegen. Nicht einmal von der allmächtigen Interpunktion werden die Blindtexte beherrscht – ein geradezu unorthographisches Leben. Eines Tages aber beschloß eine kleine Zeile Blindtext, ihr Name war Lorem Ipsum, hinaus zu gehen in die weite Grammatik.</p>
--	--	--	---

Bei schmalen Spalten laufen lange Worte über die nächste Spalte, mit automatischer Silbentrennung vermeidet man das Problem.

10. CSS in Elementor

Am Anfang des Web war alles einfacher: wer die Inhalte auf seiner Webseite mit CSS gestalten wollte hat diese entweder in eine externe Datei geschrieben und diese im HTML-Dokument eingebunden. Oder man hat mit `<style>` einen HTML-Tag verwendet, oder man hat Stil-Attribute innerhalb eines HTML-Tags gesetzt. Wer mit WordPress arbeitet hat dann die CSS-Definitionen im Theme-Ordner in der Datei `style.css` gesetzt. Oder im *Customizer* oder in einem *Settings-Bereich* des jeweiligen Themes CSS definiert.

All das funktioniert noch immer. Aber mit Elementor geht es auch deutlich eleganter und auch ein wenig einfacher. Wer CSS lernen will findet Unmengen an Tutorials, zum Nachschlagen ist <https://www.w3schools.com/css/> immer ein guter Tipp. Was man mit CSS alles machen kann ist nicht Teil dieses Kapitels.

10.1. CSS über die ID im Widget einfügen

Elementor legt um jedes Widget immer einen DIV-Tag namens *elementor-widget-container*. Daher kann man auch bei jedem Widget die gleichen Einstellungen für Abstände, Rahmen etc. treffen. Diesen Widget-Container kann man nutzen. Im Beispiel hier wurde

```
selector {
    background-color: lightgreen;
}
```

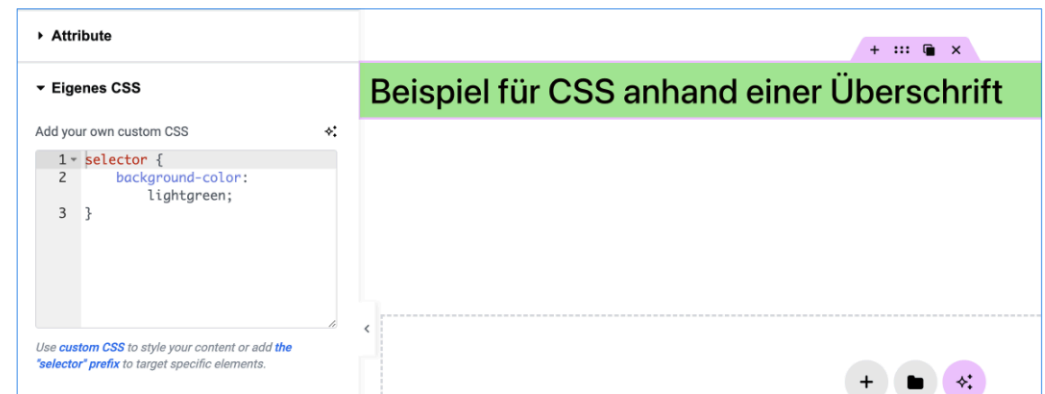
Im Tab *Eigenes CSS* unter *Erweitert* eingetragen, damit bekommt die Überschrift eine Hintergrundfarbe.

Dabei ist die Bezeichnung **selector** ein „magischer Begriff“. Elementor hat diesen Begriff eingefügt damit man genau dieses eine Widget ansprechen kann. Man muss also keine ID händisch vergeben, man verwendet die ID die ohnehin vergeben wurde.

Man kann aber auch eine eigene ID vergeben und dann diese ansprechen, dazu muss man unter

CSS Ebene	Auswirkung
Widget ID	nur dieses Widget
Widget CLASS	alle Anwendungen der Klasse
Post CSS	diese Seite oder dieser Beitrag
Theme Style	alle Seiten und Beiträge die Elementor verwenden
Child Theme	alle Inhalte der Website
Theme	alle Inhalte der Website

Die Tabelle listet die Möglichkeiten auf die man nutzen kann, wenn die anderen Einstellmöglichkeiten (also der Tab *Stil* eines Widgets) nicht ausreichen.



Im Tab *Erweitert* findet man unten *Eigenes CSS*. Hier kann man beispielsweise Code für die Hintergrundfarbe hinterlegen.

...(Widget) Erweitert > Layout > CSS ID

eine ID eintragen. Danach wird im

...(Widget) Erweitert > Eigenes CSS

Der CSS Code eingetragen, in diesem Beispiel:

```
#meineID {  
    background-color: pink;  
}
```

Wichtig ist natürlich das Zeichen # am Anfang. Das Resultat ist hier ein rosa Hintergrund. Da eine einmalige ID verwendet wird kann der CSS-Code an beliebiger Stelle untergebracht werden, also auch in der *Seite Einstellungen* oder auch innerhalb der *Website Einstellungen*.

Wichtig: Wenn eine Gestaltung (zB. eine Farbe für eine Überschrift) in den Webseiten-Einstellungen getroffen wurden kann es sein, dass diese Einstellung durchsticht, da diese direkt am HTML-Tag des Elements (zB. am H3-Tag) ansetzt und damit näher ist. Eigenes CSS sollte nicht eingesetzt werden, wenn die Grund- oder Widget-Einstellungen ausreichen um ein Element zu gestalten.

10.2. CSS über Klassen im Widget anwenden

Wenn man keine ID sondern eine Klasse anlegen will funktioniert das analog zur ID-Eingabe. Der Unterschied in der Anwendung liegt darin, dass eine Klasse (class) auf der Seite öfter als 1 x angewendet werden kann. Man kann also bei einem Widget einen Style definieren und durch eintragen der entsprechenden Klasse auch auf anderen Widgets den Stil anwenden.

Vorsicht! Man verliert schnell den Überblick, wo man den eigenen CSS-Code hingeschrieben hat. Unter

...(Widget) Erweitert > Layout > CSS-Klassen

Im Tab *Erweitert* unter Layout findet man das Feld CSS ID, die hier eingetragen werden kann. Unter *Eigenes CSS* kann diese angesprochen werden.

Position:
Z-Index:
CSS ID:
CSS-Klassen:

Beispiel für CSS anhand einer Überschrift

```
▼ Eigenes CSS  
Add your own custom CSS  
1 #meineID {  
2   background-color: pink;  
3 }
```

Im Tab *Erweitert* unter Layout findet man auch das Feld CSS-Klassen, die hier - durch Leerzeichen getrennt - angegeben werden können.

Position:
Z-Index:
CSS ID:
CSS-Klassen:

Beispiel für CSS anhand einer Überschrift

```
▼ Eigenes CSS  
Add your own custom CSS  
1 .skyblue {  
2   color: darkblue;  
3   background-color: lightblue;  
4 }
```

werden eine oder mehrere (durch Leerzeichen getrennte) Klassen eingetragen, Diese Klassen können dann im Bereich *Eigenes CSS* des Widgets oder aber auch (besser!) im Bereich *Eigenes CSS* der Seite oder im entsprechenden Bereich der *Webseiten-Einstellungen* definiert werden:

Panel > Website Einstellungen > Individuelles CSS > Eigenes CSS

10.3. CSS für Seiten definieren

Wer oben mittig beim Bearbeiten einer Seite auf das Zahnrad-Icon klickt kommt zu den Seiten-Einstellungen. Dort gibt es ebenfalls einen Tab Erweitert der nur die eine Option anbietet: Eigenes CSS.

Panel > Seite Einstellungen > Erweitert > Eigenes CSS

Natürlich können hier auch Klassen definiert werden oder auch CSS für bestimmte IDs hinterlegt werden. Klassen und IDs die vergeben werden betreffen immer das Element inkl. umgebenden Tag!

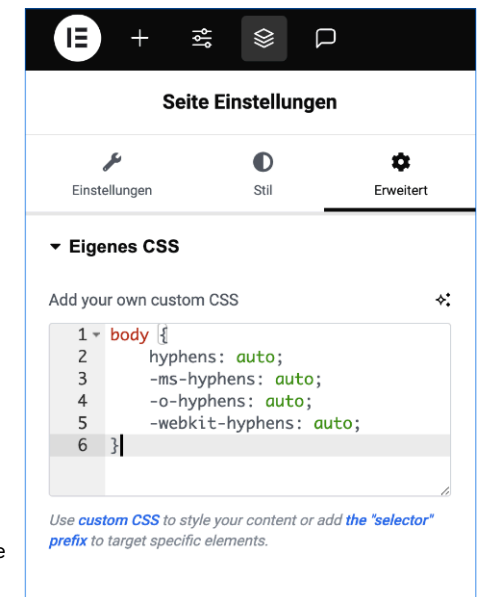
10.4. CSS in den Webseite Einstellungen

Wie im Kapitel 2 beschrieben gibt es auch in den Webseiten Einstellungen einen Bereich für CSS:

Panel > Webseite Einstellungen > Einstellungen > Eigenes CSS

Eine sinnvolle Anwendung hier ist es zB. Die CSS-Option zur automatischen Wortabteilung zu aktivieren. Der Code hierfür kann

```
body {  
  hyphens: auto;  
  -ms-hyphens: auto;
```



Die Seite Einstellungen sind der richtige Platz, um die CSS-Optionen zur Silbentrennung zu aktivieren.

```
-o-hyphens: auto;
-webkit-hyphens: auto;
}
```

so aussehen. Ausserdem kann man CSS auch über das Dashboard hinzufügen, und zwar über

Dashboard > Elementor > Custom Code

10.5. Stil anwenden - Stil zurücksetzen

Das über den Rechtsklick oder CTRL-Klick zugängliche **Kontext-Menü** hält zwei praktische Funktionen beim Arbeiten mit Stilzuweisungen bereit.

...(Widget) Kontextmenü > Stil zurücksetzen

löscht die Stilzuweisungen wie Schriften, Farben, Ausrichtung etc. und versetzt das Widget zurück in den Zustand wie es aus dem Panel gezogen wurde.

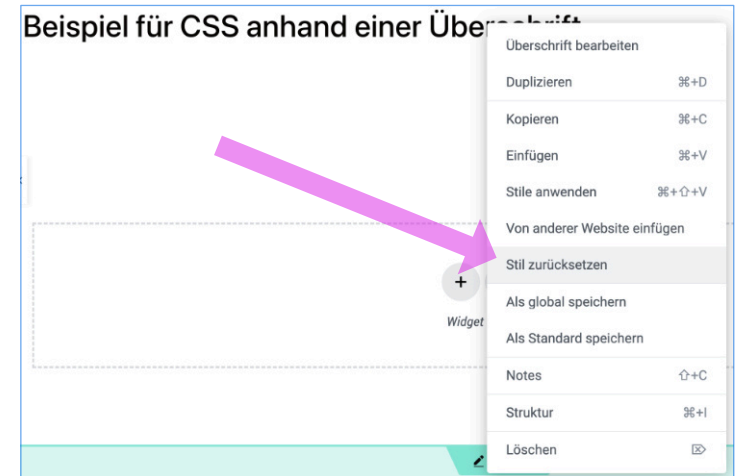
Das Kontextmenü funktioniert übrigens nicht nur innerhalb der Arbeitsfläche sondern läßt sich auch über den Navigator aktivieren.

Eine der wirklich praktischen Funktionen im Elementor ist die Möglichkeit, den Stil eines Widgets oder eines Layout-Elements wie einer Spalte oder einem Abschnitt von einer auf eine andere Anwendung dieses Elementes zu übertragen. Dazu muss man

...Widget (Kontextmenü) > Kopieren

Und danach beim Zielobjekt auf welches man die Stilzuweisung übertragen möchte

...Widget (Kontextmenü) > Stil anwenden



Die Zurücksetzen-Option ist der schnellste Weg zum Ausgang zurückzukehren, wenn man sich bei der Gestaltung verlaufen hat.

wählen. Man kopiert also das ganze Objekt in die Zwischenablage, eingefügt wird aber nur der Stil, der auf dieses Objekt angewandt wurde. Das funktioniert auch bei Layout-Widgets wie Containern sehr zuverlässig.

werkform **qualified web solutions**

Website Complete Von der ersten **Konzeption** über das **Webdesign** bis hin zum **Contentplan**, inkl. aller notwendigen Leistungen wie **E-Mail** und **Backup**: werkform bietet dir die **Komplettlösung** für deine WordPress-Website, vom ersten Strich über die Umsetzung bis hin zum laufenden Betrieb inkl. Wartung und Updates.

Umsetzungen Du hast ein Konzept für Inhalte, und/oder Design und suchst einen **kompetenten Partner** für die **Umsetzung**? werkform füllt die Lücken in deinem Projekt!

Upgrades Du hast eine **alte Website** und suchst den besten Weg um mit wenig Aufwand wieder eine moderne, aktuelle und sichere Website zu bekommen? **werkform weiß wie!**

Templates 2 go Eigentlich ist alles fertig - nur umgesetzt ist es noch nicht - dir fehlt der richtige Zugang, wie aus deinen Ideen am schnellsten eine konsistente, sichere und funktionierende Website gemacht wird? werkform baut mit **deinen Teilen** die fertige Website.

Hosting werkform bietet verschiedene Hosting-Lösungen. Je nach Projekt verwenden wir eigene Server oder greifen auf den genau passenden Dienstleister zurück. **Der Host muss zum Projekt passen!**

E-Mail Websites sind super - aber Web ohne E-Mail ist wie Brot ohne Butter. werkform bietet dir die **passende E-Mail-Lösung** für deine Anforderungen.

Domains werkform kümmert sich um deine Domains. Automatische Verlängerung, zuverlässige DNS. Unsere Partner sind absolute Spezialisten - und bieten dir das beste Preis/Leistungsverhältnis.

werk form

Kostenlos: Erstgespräch buchen

11. Formulare

Ein Formular dient dazu Daten, die der User im Browser eingibt, an eine E-Mailadresse oder eine Software zu schicken. Diese Daten sind immer ein Paar aus dem *Namen* der Daten und dem *Wert* (also den Daten selber). Damit ein Formular abgeschickt wird, muss entweder ein Button gedrückt werden oder irgendwie anders eine Aktion gesetzt werden (zB. die Return-Taste gedrückt werden).

11.1. Das Widget Formular

Um ein Formular auf einer Seite, in einem Beitrag oder auch in einem Template zu positionieren gibt es das Widget, das genau so heisst: *Formular*.

Wenn man das Widget wie gewohnt im Elementor per Drag&Drop in die Arbeitsfläche zieht wird ein Formular eingefügt das bereits die Elemente

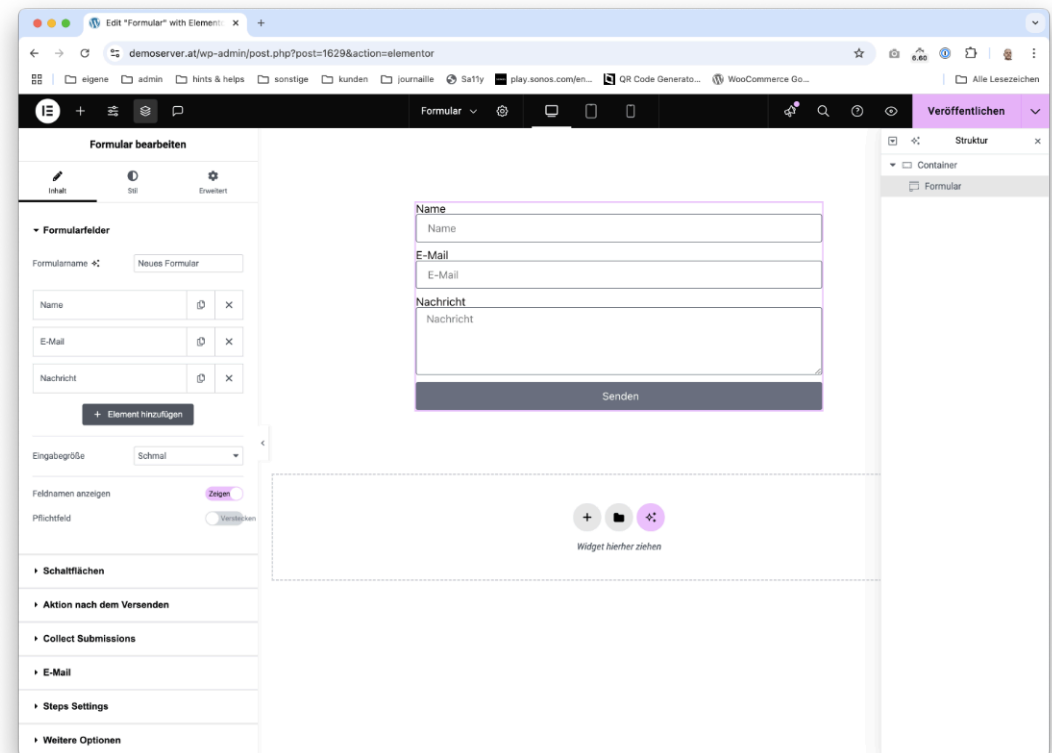
- Name
- E-Mail
- Nachricht
- Senden-Button

enthält. Die Eingabegröße ist mit *schmal* auf die zweite von 5 Feldgrößen vor-eingestellt. Die Option *Feldnamen anzeigen* ist gesetzt, die für *Pflichtfelder* nicht. Diese werden durch einen roten Stern * gekennzeichnet.

Formularfelder

Im ersten Tab **Formularfelder** wird zuerst ein sinnvoller Formularname vergeben. Dieser Name dient zur Identifikation des Formulars auch bei den daraus generierten E-Mails.

Jedes einzelne Formularfeld hat einen Bereich für *Inhalt* und einen für *Erweitert*.



Wer ein Formular im Dokument platziert bekommt gleich eine Grundausstattung mit Name, E-Mail und Nachrichtenfeld.

Festgelegt wird im **Inhalt**:

- Typ (= Art des Formfeldes)
- Feldnamen
- Platzhalter
- Pflichtfeld
- Spaltenbreite (responsive)

Der Bereich **Erweitert** bietet Platz für

- Standardwert
- ID
- Shortcode

Dynamische Angaben sind möglich für Feldnamen und Platzhalter, Standardwert und ID.

Wer sich ein wenig mit HTML auskennt erkennt rasch, dass es sich hier nicht um reine Feldtypen sondern um eine Mischung von Feldtypen und Funktionen handelt.

Das erste Feld in unserem Beispiel ist vom **Typ** Textfeld, es heisst *Name*, was auch als **Feldname** angezeigt werden kann. Der **Platzhalter**, also was im Eingabebereich des Feldes vorab stehen kann, ist ebenfalls mit *Name* vorausgefüllt. Wenn die Option **Pflichtfeld** gewählt wird wird beim Absenden auf das Vorhandensein eines Wertes geprüft.

Die **Spaltenbreite** ist ein Prozentwert der sich auf den zur Verfügung stehenden Platz bezieht, wer also 2 Felder mit 50% definiert wird diese Felder nebeneinander dargestellt bekommen.

Ein **Standardwert** kann in den erweiterten Feldoptionen eingetragen werden. Der Unterschied zwischen Platzhalter und Standardwert ist übrigens, dass der erster nur eine für den User sichtbare Eingabehilfe ist während das Zweite wirklich im Feld eingetragen wird, also im Falle des Absendens weiter gegeben.

Die **ID** muss unbedingt eindeutig sein, damit der Formularversand klappt. Aus technischen Gründen sind hier die erlaubten Zeichen beschränkt (keine Um-

The image shows two screenshots of the Elementor form field configuration interface. The left screenshot shows the 'Inhalt' (Content) tab selected, with a pink arrow pointing to it. The right screenshot shows the 'Erweitert' (Advanced) tab selected, with a pink arrow pointing to it. Both screenshots show a form field named 'Name' with various configuration options like Type, Field names, Placeholder, Required, and Column width.

Jedes Formularfeld hat 2 Bereiche zur Konfiguration: Inhalt und Erweitert. Beide müssen sorgfältig ausgefüllt werden, damit brauchbare Ergebnisse verschickt werden.

- | | | |
|-----------------|---------------------|------------------|
| 1. Text | 8. Auswahlbox | 15. HTML |
| 2. Email | 9. Akzeptanz | 16. Versteckt |
| 3. Textbereich | 10. Zahl | 17. reCaptcha |
| 4. URL | 11. Datum | 18. reCaptcha V3 |
| 5. Telefon | 12. Zeit | 19. HoneyPot |
| 6. Radio | 13. Datei hochladen | 20. Step |
| 7. Auswahlliste | 14. Passwort | |

Elementor bietet 20 verschiedene **Typen** für Formularfelder an, wobei manche Typen keine Inhalte transportieren sondern Funktionen auslösen (Step, reCaptcha).

laute etc.). Wieder erkennbare IDs werden dringend empfohlen. Der **Shortcode** wiederum dient dazu den Wert des Felds im erzeugten E-Mail zu platzieren. Dieser wird von Elementor automatisch generiert und kann hier kopiert werden.

Schaltflächen

Diesen tauchen bei mehrstufigen Formularen und beim Senden-Button auf. Im Tab Inhalt kann eine grundsätzliche **Größe** eingestellt werden (mehr Design im Stil-Tab). Wie auch bei den Formularfeldern kann eine **Spaltenbreite** in Prozent angegeben werden, responsive Angaben möglich. Wie auch bei der **Ausrichtung** des jeweiligen Buttons, wobei hier die mit Blocksatz bezeichnete Option *Volle Breite* bedeutet.

Für die als **Step Buttons** bezeichneten Bedienelemente für mehrstufige Darstellung können die Beschriftungen eingetragen werden.

Für die **Absenden Schaltfläche** steht neben der Beschriftung auch ein Icon zur Verfügung das entsprechend festgelegt und konfiguriert werden kann.

Eine **ID** für den Button kann ebenfalls eingetragen werden.

Aktionen nach dem Versenden

Im Dritten Tab wird festgelegt, was denn überhaupt mit dem Formular geschehen soll, sobald es abgeschickt wurde. Die Möglichkeiten sind:

- Collect Submissions (vorausgewählt)
- E-Mail (vorausgewählt)
- E-Mail 2
- Weiterleiten
- ActiveCampaign, ConvertKit, Discord, Drip, GetResponse, Mailchimp, MailerLite, Slack
- Webhook
- Popup

Die Schaltflächen eines Formulars werden in einem eigenen Tab konfiguriert. Neben der Äusseren Erscheinung (Größe, Spaltenbreite) und Beschriftung ist beim Submit-Button auch ein Icon möglich.

▼ Schaltflächen

Größe	Schmal	▼
Spaltenbreite	100%	▼
Step Buttons		
Nächster	Nächster	☰
Voriger	Voriger	☰
Absenden Schaltfläche		
Submit	Senden	☰
Icon	🚫 🏠 ●	
Schaltflächen ID		☰

Please make sure the ID is unique and not used elsewhere on the page. This field allows A-z 0-9 & underscore chars without spaces.

Für jede hier eingetragene Option steht dann *ein eigener Tab zur Konfiguration* der jeweiligen Option bereit. Je nachdem, ob die Daten selber oder über einen Drittanbieter verarbeitet werden soll, werden hier die Einstellungen getroffen. Der allgemeine **Webhook** erlaubt die Weiterleitung an andere Dienste (zB. Zapier), **Weiterleiten** bietet ein Redirect, also einen Aufruf einer beliebigen URL, zB. einer DANKE-Seite.

Die Option **Collect Submission** speichert die Formulardaten innerhalb von Elementor wo sie dann inkl. der Angaben wann und wo das Formular verschickt wurden abrufbar bleiben. Abgerufen können die Einträge über

Dashboard > Elementor > Submissions

Die voreingestellte Funktion **E-Mail** leitet das Form an eine E-Mail-Adresse weiter, wobei hierfür die vom Webhost eingestellte SMTP-Technik verwendet wird. Wenn man also über einen bestimmten Account versenden will benötigt man ein **SMTP-Plugin**.

Neben der *Empfänger-Adresse (An)* und *Betreff* ist natürlich die Nachricht selbst wichtig. Voreingetragen ist der Shortcode **[all-fields]**, damit werden einfach alle Formularfelddaten übermittelt. Alternativ dazu kann hier eine eigene Nachricht formuliert werden wo dann die einzelnen Feldinhalte über die Feld-Shortcodes, also zB. **[field id="name"]** eingebaut werden.

Die weiteren Felder *Von E-Mail*, *Von Name*, *Antwort an*, *Cc*, *Bcc* sind die Daten für den Mail-Header, sind also bei der Spam-Erkennung ein Thema. Im Feld *Antwort an* kann man E-Mail-Feld auswählen. Wenn man das tut (und natürlich ein solches Feld existiert und auch korrekt ausgefüllt wird) kann dann auf das generierte E-Mail direkt an den Versender geantwortet werden.

Weiters kann man diverse Meta Daten dem Mail hinzufügen und auswählen, ob es HTML-Codiert oder als reiner Text versendet werden soll.

Step Settings

Wer ein mehrstufiges Formular anbietet kann hier die Gestaltung der verschie-

Die wohl wichtigste Aktion nach dem Versenden ist E-Mail. Hier konfiguriert man an wen, von wo und was versendet werden soll.

▼ E-Mail

An

Betreff

Nachricht

By default, all form fields are sent via [all-fields] shortcode. To customize sent fields, copy the shortcode that appears inside each field and paste it above.

Von E-Mail

Von Name

Antwort an

Cc

Bcc

Meta Daten
 Datum Zeit Seiten URL Benutzer Agent
 Remote IP Credit

Senden als

denen Stufen vornehmen. Damit generiert man eine Anzeige/Navigation zwischen den einzelnen Formularstufen. Von *None* bis *Icon & Text* sind eine Reihe verschiedener Optionen wählbar. Einige Optionen bieten dann auch Gestaltungsmöglichkeiten an.

Weitere Optionen in Formularen

Eine eindeutige Formular ID zu vergeben ist immer eine gute Idee! Ausserdem können hier eigene Texte für die Nachrichtenmeldungen angelegt werden.

11.2. Besondere Feldtypen

Die verschiedenen im HTML möglichen Feldtypen sollten bekannt sein, an dieser Stelle werden nur einige Besonderheiten im Elementor beschrieben.

Feldtyp: Datei hochladen

Damit kann der Benutzer ein File mit dem Formular mitschicken. Das kann naturgemäss ein wenig kritisch werden, man muss also ein paar Einschränkungen vornehmen:

1. Recht neu ist das Auswahlfeld *Send Files*, wo man definiert, was mit den Dateien geschehen soll. Ursprünglich wurden die Dateien am Server gespeichert (auch in den Form-Submissions). Allerdings bekommen sie neue Filenamen. Jetzt ist auch auswählbar, dass die Files als E-Mail-Attachment versandt werden.
2. *Maximale Dateigröße*: damit verhindern man, dass der User zu große Files mit dem Formular mitschickt. **Achtung**: die Filegrößen die der Webhost bzw. die Mailprogramme akzeptieren können hier nicht beeinflusst werden.
3. *Erlaubte Datei-Typen*: hier kann man zumindest den Namen des Files auf die Fileart checken. Ob in einem .jpg genannten File auch wirklich ein Bild


Für mehrstufige Formulare gibt es wenige Einstellungen.

▼ Steps Settings

Typ

Gestaltung


▼ Weitere Optionen


Formular ID 


Please make sure the ID is unique and not used elsewhere on the page. This field allows A-z 0-9 & underscore chars without spaces.


Form Validation


Eigene Nachrichten Ja


Nachricht bei erfolgreich versenden 





Form Error 





Server Error 



Ungültiges Formular 



Required Field 



Eine eigene ID kann notwendig sein, eigene Nachrichten sinnvoll. Leicht verwirrend ist der Schalter bei Form Validation, ist er aktiviert taucht eine zusätzliche Eingabemöglichkeit Required Field auf.

drinnen ist und nicht ein .zip mit Schadsoftware kann man so jedoch nicht überprüfen, es werden die Dateieindungen gecheckt.

4. *Mehrere Dateien:* damit ermöglicht man es, mehr als nur ein File mitzuschicken. Ein Beispiel für eine Anwendung wäre ein Formular für eine Kostenanfrage einer Reparatur eines Möbelstückes, wo der User seine Bilder des Schadens mit der Anfrage mit schickt. Allerdings sind viele User mit der Bedienung eines solchen Feldes überfordert.

Feldtyp: Akzeptanz

Damit erzeugt man eine Checkbox die sicher stellt, dass nur Formulare abgeschickt werden wo der Benutzer auch den Bedingungen zugestimmt hat. Normalerweise wird man es als Pflichtfeld ausführen. Leider kann man die Feldbeschriftung nicht einzeln ausblenden, oft wird dieses Feld ohne Feldnamen verwendet weil dieser eher stört. Im Text ist HTML erlaubt, es können also Links gesetzt werden, zB. zur Seiten mit der Datenschutzvereinbarung.

Hinweis: Wenn man so ein Feld einsetzt um eine DSGVO-konforme Zustimmung zu erhalten darf die Option standardmäßig ausgewählt *nicht* gesetzt werden.

Feldtypen: Datum und Zeit

Sehr praktische Feldtypen, wenn man vom User eine Datums- und/oder Zeitangabe benötigt (zB. für eine Terminvereinbarung). Es öffnet sich beim Verwenden ein kleines Modalfenster mit Datumsauswahl.

Die Elementor-Umsetzung bietet die Optionen für eine untere bzw. obere Datumsgrenze. Das Standardformat ist YYYY-MM-DD, wenn man den Schalter *Normales HTML5* setzt ändert sich das Datumsformat auf DD.MM.YYYY. Gleiches gilt auch für den Feldtyp Zeit. Auch hier bringt der etwas unglückliche Begriff *Normales HTML5* das im deutschen Sprachraum üblichere Format HH:MM.

Fileupload mit Formularen ist immer kritisch: Filehandling und Sicherheit müssen genau überdacht werden,

Files

Inhalt | Erweitert

Typ: Datei hochladen

Feldnamen anzeigen: Files

Pflichtfeld: Nein

Send files: Email with link, Email with attachment, Email with both

Max Dateigröße: [Dropdown]

Kontaktiere deinen Host, um die maximale Datei-Upload-Größe zu erhöhen

Erlaubte Datei-Typen: [Textfeld]

Mehrere Dateien: Nein

Spaltenbreite: 100%

Feldtyp: HTML

Für weitere Anwendungen bietet Elementor die Möglichkeit mit einem HTML-Feld Code im Formular unter zu bringen. Zum Beispiel kann man damit die leider nicht unterstützte Funktion des Schiebereglers (Input type=range) umsetzen.

Der Feldtyp wird auch verwendet um zwischen den Eingabefeldern Text unterzubringen der ev. eine Erklärung für ein Feld oder einen Link für ein Info-Pop-up o.ä. aufnimmt.

Feldtypen: reCaptcha, Honeypot

Zur Einbindung von Captchas bietet Elementor die Typen reCAPTCHA sowie reCAPTCHA V3 an.

Ausserdem gibt es auch eine Honeypot-Funktion an. Die Idee dieser Technik ist es, Formularspam dadurch zu verhindern, dass Spambots auf die schlichte Existenz eines solchen Feldes reagieren und sich dadurch verraten, denn Menschen sehen dieses Feld gar nicht.

11.3. Gestaltung der Formularelemente

Die Gestaltung der Formulare wird über die Website-Grundeinstellungen und über den Tab *Stil* durchgeführt.

Panel > Website-Einstellungen > Theme Stil: Formularfelder

Formular

Bei jedem einzelnen *Formularfeld* ist die relative Breite (Voreinstellung = 100%) angegeben. Wenn man eine mehrspaltige Gestaltung anwendet (durch Änderungen der Breiten auf zB. 50%) kann man nun den **Spaltenabstand** angeben, weiters auch den **Reihenabstand** der unter den einzelnen Formularfeldern wirksam sein soll.

Bei angezeigten *Feldnamen* kann ein **Abstand** zum Feld, **Textfarbe**, **Markierungsfarbe** und **Typographie** eingestellt werden. Als Markierungsfarbe wird jene Farbe bezeichnet, mit welcher das * eingefärbt wird, welches bei Pflichtfeldern angezeigt wird.

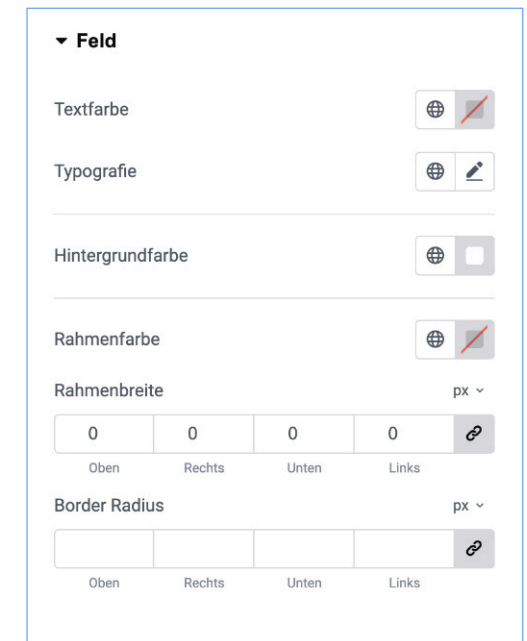
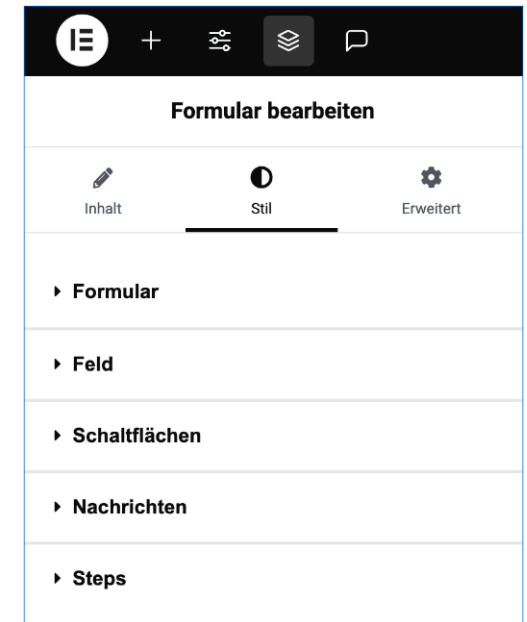
Abstände, **Farbe** und **Typographie** kann für *HTML Felder* extra eingestellt werden.

Feld

Im Bereich **Feld** werden die Einstellungen für die Eingabefelder vorgenommen. Hier können eingestellt werden:

- Textfarbe
- Typographie
- Hintergrundfarbe
- Rahmenfarbe
- Rahmenbreite
- Border Radius

Die Formulgestaltung wird über den Stil-Tab in 5 Bereichen durchgeführt.



Angaben für die Feldgestaltung bei Formularen.

Schaltflächen

Unter **Position** kann man wählen, ob der Submit-Button links, mittig, rechts unter dem Formular oder mittels *strecken* über die ganze Breite dargestellt werden soll. Mittels *Ausrichtung* kann die Beschriftung des Buttons verändert werden (nur bei aktivem *strecken*). Die *Typographie*-Optionen sind wie überall einzustellen, ebenso die *Rahmen*.

Weiters können **Hintergrund-** und **Textfarbe** getrennt für den Buttontyp *Nächster und Abschicken* bzw. den Buttontyp *Voriger* eingegeben werden.

Dazu Angaben zum Rahmen und auch ein Innenabstand des Textes zum Rahmen.

Nachrichten

Bei den Angaben zu den Nachrichten kann die Typographie eingestellt werden. Ausserdem können 3 verschiedene Farben für erfolgreiche Durchführung, Fehler und die Nachricht selber angegeben werden.

Steps

Mehrstufige Formulare benötigen zwischen den Stufen einige Designangaben. Neben Typographie gibt es Abstände und Innenabstände. Es gibt 3 Stauseinstellungen für *Inaktiv*, *Aktiv* und *Abgeschlossen* die wiederum jeweils eine erste und eine zweite Farbe haben können. Dazu kommen noch 2 Einstellungen für Trenner.

11.4. Google reCaptcha V3 verwenden

Wer eine Wette abschliessen will, ob über ein Webformular zuerst eine echte Anfrage oder eine Spammnachricht in der Mailbox landet, sollte auf's Zweite setzen. Auch wenn die Erfolge gering sind: anscheinend kann immer noch ir-

Die neue Positionsangabe bei den Schaltflächen erlaubt flexiblere Gestaltung, die Anzeige für die Mehrstufigen Formulare sind nicht übermässig flexibel.

▼ Steps

Typografie

Abstände px 20

Innenabstand px 30

Inactive Aktiv Completed

Erste Farbe

Zweite Farbe

Divider Width px 1

Divider Gap px 10

▼ Schaltflächen

Position

Typografie

Border Type Vorgabe

Normal Hover

Next & Submit Button

Hintergrundfarbe

Textfarbe

Previous Button

Hintergrundfarbe

Textfarbe

Border Radius px

Oben Rechts Unten Links

Innenabstand des Textes px

Oben Rechts Unten Links



Dir gefällt dieses Skriptum?

Kaffeespende hier!

jetzt spenden

Spende Skriptum 3.0



Anzeige

gendwer einen Nutzen daraus ziehen sinnlose Nachrichten über Webformulare zu verschicken. Ein Mittel um sich dagegen zu wehren besteht im Einsatz von Captchas, also kleinen Suchspielen, Rätseln oder Aufgaben die ein Mensch vermeintlich einfach, ein Robot aber nur schwer lösen kann. Daraus weiter entwickelt hat Google eine Technik die versucht Spam über Formulare zu verhindern. Ohne, dass der User belästigt wird indem er einen Haken setzen muss und bestätigt, kein Roboter zu sein. Das kostenfreie Produkt das daraus entstanden ist heißt **Google reCAPTCHA V3**. (<https://www.google.com/recaptcha/>).

Man benötigt ein Google-Konto, mit dem man sich beim Service einloggt. Dort kann man in einem Panel verschiedene Webseiten registrieren. Dabei muss man jeweils festlegen, ob man die Version 2 oder 3 verwenden will. Die beiden Dienste benötigen unterschiedliche Schlüssel, dh. ein Umstieg in der reCaptcha-Version bedarf eine erneuten Registrierung der Seite.

Für jede Webseite (bei der man übrigens mehrere Domainnamen eintragen kann) erhält man ein Schlüsselpaar, bestehend aus einem Webseitenschlüssel und einem geheimen Schlüssel.

Die Schlüssel werden nun auf der eigenen Webseite eingetragen, dazu

Dashboard > Elementor > Einstellungen > Integr. > reCAPTCHA V3

aufrufen und die beiden Zeichenketten entsprechend hinein kopieren. Damit das Ganze nun funktioniert muss nun auch das Formular darüber informiert werden. Dazu gibt es das entsprechende Formularfeld reCAPTCHA V3. Das Besondere an dieser Lösung ist, dass die reCaptcha-Plakette konfiguriert werden kann, ob sie links, rechts oder innerhalb der Formularelemente auftauchen soll.

Diese Plakette ist Teil der Nutzungsvereinbarung, die man mit Google eingeht sobald man dieses Service verwendet. Die mit Elementor umgesetzte Lösung zeigt die Plakette wirklich nur dann an, wenn auch ein Formular verwendet wird. Man kann sich den Umweg über eine CSS-Ausblendung der Plakette und dafür eine Einblendung des Textes sparen.

Nach dem Speichern ist nun das Formular abgesichert. Über das reCaptcha-

Nach der Registrierung der Website bei Google erhält man das Schlüsselpaar, mit dem man den Dienst auf der eigenen Website autorisiert.

Im Dashboard werden die Schlüssel hinterlegt, im Formular ein Feld reCaptcha V3 eingefügt.

Adminkonto bekommt man nach einiger Zeit Informationen über die letzten 90 Tage, wieviele Anfragen über das Tool analysiert wurden.

11.5. Mehrstufige Formulare

Die Idee beim mehrstufigen Formular ist es, dem User die Bedienung zu vereinfachen indem eine ev. lange Liste an Eingabefeldern in zusammenhängende Abschnitte geteilt wird. Technisch bedeutet das, dass bis zum Absenden des Formulars die vom User eingegebenen Daten serverseitig aufgehoben werden müssen und erst nach dem Senden-Klick weitergeleitet werden (zB. als E-Mail, siehe oben).

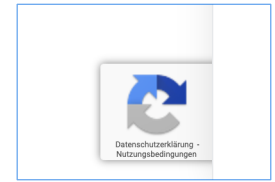
In dem Beispiel hier wurde das Formular in 3 Schritte geteilt, zuerst sollen die persönlichen Daten eingeben werden, im 2. Schritt kann eine Nachricht geschrieben werden und im 3. Schritt kann eine Datei zur Nachricht gehängt werden, die dann mit dem Formular übertragen werden soll. Zwischen den einzelnen Schritten kann navigiert werden, dh es ist möglich zB. im dritten Schritt zurück zum Nachrichtenfenster in Schritt 2 zu wechseln und dort seine Nachricht zu überarbeiten.

Die Umsetzung dieser Funktionalität erfolgt mittels dem Feldtyp **Step**, der jeweils am Anfang (wird automatisch erstellt) bzw. zwischen den zu trennenden Feldern eingesetzt wird.

Die Gestaltungsmöglichkeiten sind 2-geteilt. Zum einen gibt es den Abschnitt Step Settings in dem angelegt wird, ob und wie die einzelnen Schritte grafisch dargestellt werden sollen. Es gibt:

- none
- Text
- Icon
- Number
- Progress Bar
- Number & Text
- Icon & Text

Auf der Website erscheint auf der Formularseite das Recaptcha-Logo von Google.



Mehrstufige Formulare können die Benutzererfahrung verbessern und zu mehr Kundenkontakt führen.

Ausserdem kann man die einzelnen Schritte einstellen, in

- Circle
- Square
- Rounded
- none

Dann kommen wie üblich im Bereich Stil eine Reihe von Einstellungsmöglichkeiten dazu. Beispielsweise kann man die Icons je nach Stufe verschieden einfärben.

11.6. Formulardaten Speichern

Die Funktion **Collect Submissions** kann bei einem Formular im Menüpunkt *Aktion nach dem Versenden* hinzugefügt werden (und ist auch aktuell voreingestellt). Damit werden Formulardaten innerhalb vom WordPress gespeichert. Sollte also zB. die E-Mail-Zustellung nicht klappen kann man so auf diese Daten trotzdem zugreifen.

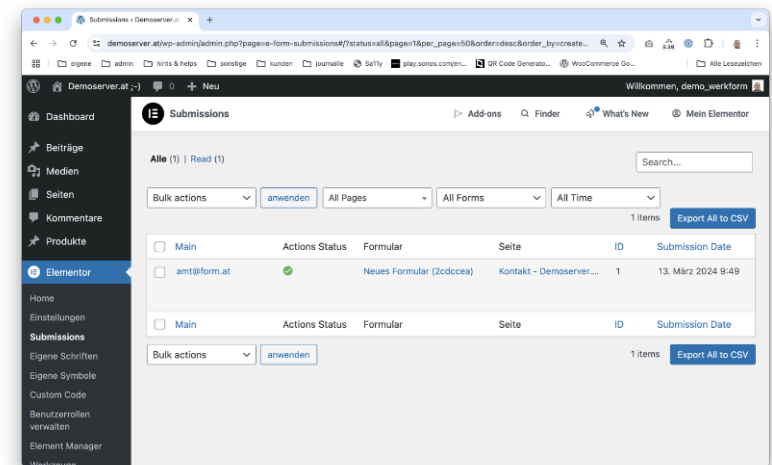
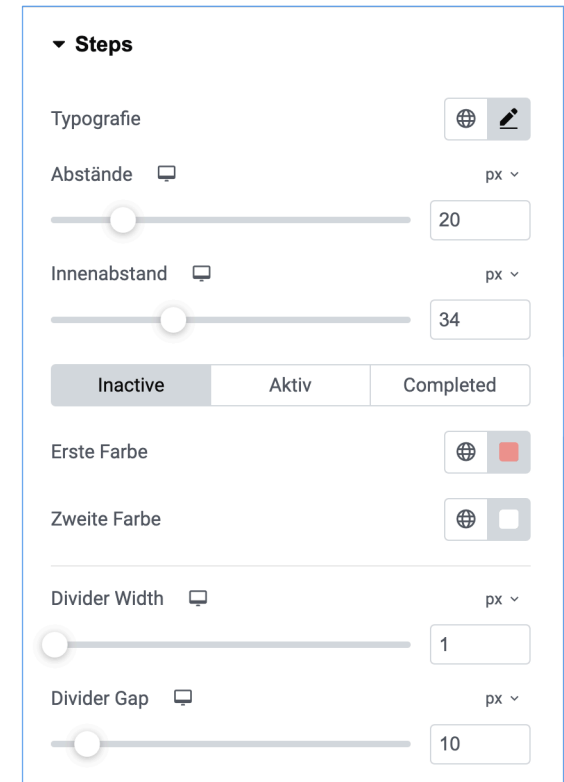
Verschiedene Filter helfen im Backend beim Finden. Sehr praktisch sind die Angaben der Form-ID und auch der Seite, wo das Formular eingebaut wurde. Eine Exportfunktion der Daten darf natürlich auch nicht fehlen (CSV).

Eine mit dem Formular übermittelte Datei wird in einem Unterverzeichnis `.../elementor/forms/` gespeichert.

Dashboard > Elementor > Submissions

Hier findet man die Datensätze, hier sind sie auch zu löschen.

Im Stil lassen sich die einzelnen Stufen konfigurieren.



Submissions im Dashboard.

11.7. Probleme beim E-Mail-Versand

Es kommt häufig zu Fehlern beim Formularversand über E-Mail. Hier einige öfter auftretende Problem und Lösungsansätze:

Problem: Das E-Mail landet im Spam-Ordner.

Lösung: entweder die Absenderadresse oder die IP des Webservers auf die Whitelist setzen. Besonders bei Nachrichtefeldern die freie Texteingabe ermöglichen werden Spammer versuchen, irgendwelchen Müll über das Formular zu versenden. Dh. die Spamssoftware erkennt eigentlich korrekt, dass es sich um Spam handelt, leider kann so die Absenderadresse fälschlich unter Spamverdacht geraten.

Problem: E-Mails werden gar nicht erst verschickt.

Lösung 1: Eine gültige Absenderadresse eintragen. Sinnvoll ist hier, eine E-Mailadresse mit der selben Domain wie die Webseite einzutragen.

Lösung 2: Es liegt nicht an der Webseite sondern am Webserver. Der Lösungsweg ist:

Checken, ob der Webserver überhaupt E-Mails verschickt. (also zB. im WordPress eine Passwortanforderung absetzen oder das Plugin Check-Emails verwenden.

Elementor verwendet die WordPress-Funktion wp_mail die wiederum die PHP-Funktion send_mail verwendet. Das kann man umgehen, indem man einen anderen Mailserver verwendet, indem man ein **SMTP-Plugin** verwendet.

Lösung 3: Sicherstellen, dass die **IDs** der Felder und des Formulars passen und eindeutig sind. Jedes Feld benötigt eine eindeutige ID, damit man das Feld auch zuordnen kann ist natürlich eine aussagekräftige ID ratsam.

12. Dynamik mit Elementor

Um dynamische Inhalte (=Attribute) zu verstehen ist es hilfreich, sich zB. das Widget *Titel* nicht als simple Sammlung von Buchstaben vorzustellen, die gestaltet und mit einem HTML-Tag umschlossen werden, sondern als Platzhalter, der zwar Tag und Form festlegt, aber den Inhalt flexibel zulässt.

In Kapitel 4 wurde im Theme Builder Footer ein Beispiel gebracht: Eine Jahreszahl soll immer das aktuelle Jahr zeigen. Man kann also nicht nur einfach 2025 in den Inhaltsbereich schreiben sondern das Widget *Titel* anweisen, an dieser Stelle die aktuelle Jahreszahl anzuzeigen.

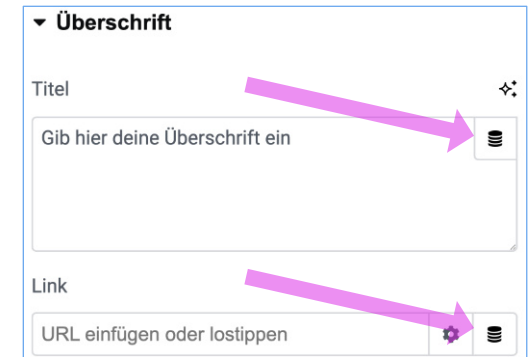
Es gibt eine Reihe von Dingen, die man dynamisch anzeigen kann. Elementor bietet in den verschiedenen Popups immer nur jene Inhalte an, die auch möglich sind. Das sind bei Texteingaben andere als bei Links oder bei Bildern oder Farben. Unterteilt nach Anwendung sind das:

Beiträge

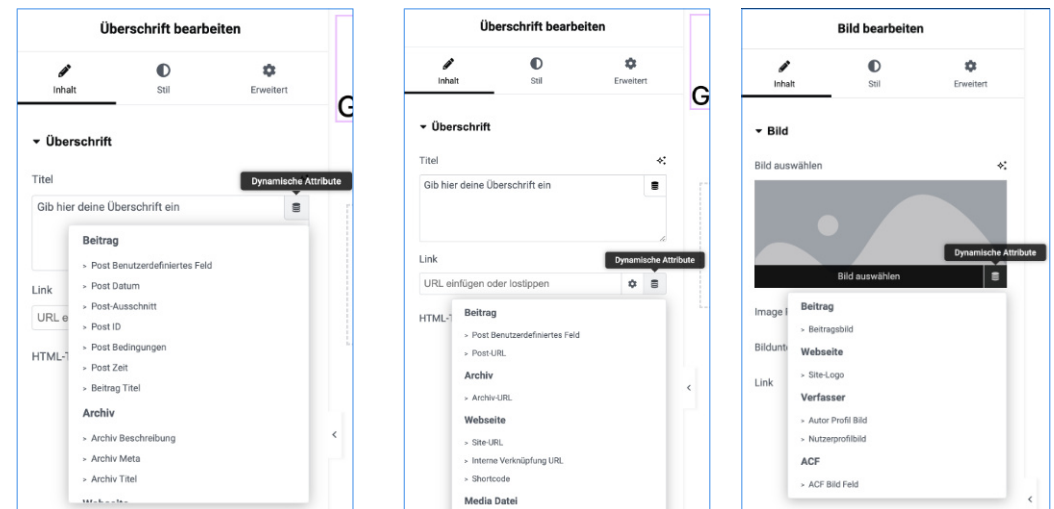
- Post Benutzerdefiniertes Feld* - Inhalt eigener Felder
- Post Datum* - Datum des aktuellen Posts
- Post Ausschnitt* - Textauszug des aktuellen Posts
- Post ID* - die WordPress ID des aktuellen Posts
- Post Bedingungen* - ausgewählte Taxonomien für den aktuellen Post (Kategorien, Schlagworte, etc.)
- Post Zeit* - die Zeitangabe, zu der der Post erstellt wurde
- Beitrags Titel* - Titel des aktuellen Posts
- Post URL* - die URL des aktuellen Posts
- Beitragsbild* - das Beitragsbild zur Post ID

Archive

- Archiv Beschreibung* - die Beschreibung des aktuellen Archives
- Archiv Meta* - zeigt alle Infos zum Archiv abhängig vom Meta-Key
- Archiv Titel* - Titel des aktuellen Archives



Um Dynamische Attribute auszuwählen muß man auf das Icon in der Eingabemaske klicken.



Die Auswahlmöglichkeiten sind vom Inhalt anhängig: Texte erlauben andere dynamische Inhalte als Links und andere als Bilder.

| *Archiv URL* - die URL des aktuellen Archives

Website

Seiten Titel - Titel der Seite oder des Beitrages

Website Tagline - Untertitel der Website

Seitentitel - Name der Website (des Blogs)

Aktuelles Datum und Uhrzeit - aktuelle Uhrzeit und/oder das Datum

Request Parameter - Zeigt die Parameter an, die mittels GET, POST oder als Query-Variablen übergeben wurden

Shortcode - beliebiger Shortcode

User Info - Zeigt eine Info aus der Auswahl möglicher Userinformationen an, zB. ID, Anzeigename, Username, Vorname, Nachname, Biographie, E-Mail, Website oder User Meta-Daten

Site URL - URL der Webseite

Interne Verknüpfung URL - zeigt die interne URL für einen Link, ausgehend von Inhalt, Taxonomy, Media oder Autor.

Site Logo - Logo der Webseite

Actions

Popup - öffnet, schliesst oder toggelt ein Popup das im Theme Builder erstellt wurde.

Off-Canvas - öffnet/schliesst bei Klick ein Off-Canvas-Widget

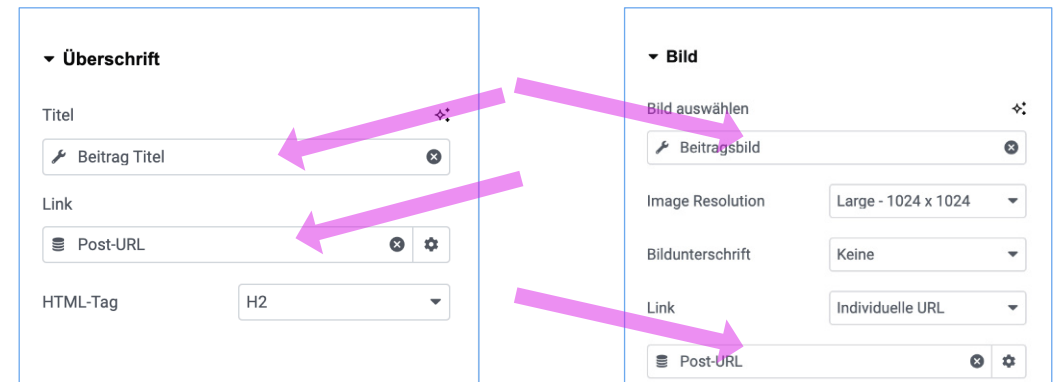
Lightbox - öffnet/schliesst bei Klick eine Lightbox

Kontakt URL - erzeugt einen Link für zB: Telefon, WhatsApp, Skype, Google Kalender, ...

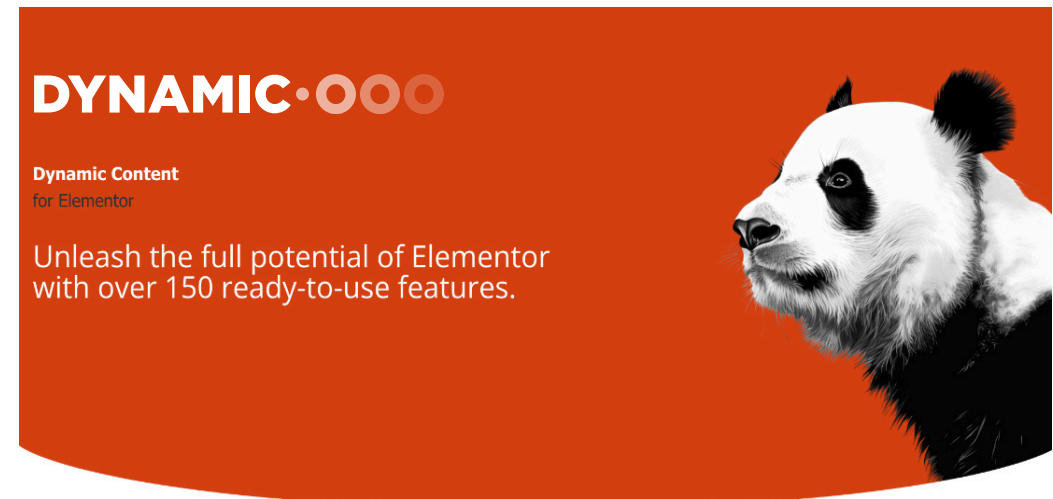
Media Datei

Beitragsbild - zeigt das Beitragsbild an, mit Fallback-Einstellung

Featured Image Data - zeigt die Metadaten des Beitragsbildes an, auswählbar sind Titel, Alt, Überschrift, Beschreibung, File URL und Attachment URL



Beispiel: in einem Loop-Template werden der Beitrag Titel, und das Beitragsbild angezeigt. Beides wird mit der Post-URL verlinkt, wodurch ein Klick auf den Beitrag selbst führt.



Verfasser / Author

- Autor Informationen* - zeigt Informationen über den Autor des aktuellen Beitrages, wählbar sind Biographie, E-Mail oder Webseite
- Autor Meta* - zeigt Metadaten des Autors ein, der Schlüssel muss eingegeben werden (Land, Geburtsdatum)
- Autorenname* - zeigt den Displaynamen des Autors
- Autor-Url* - URL des Autors, also der Archivaufruf aller Beiträge des Autors
- Autor Profil Bild* - Bild des Autors
- Nutzerprofilbild* - Bild des Nutzers

Kommentare / Comments

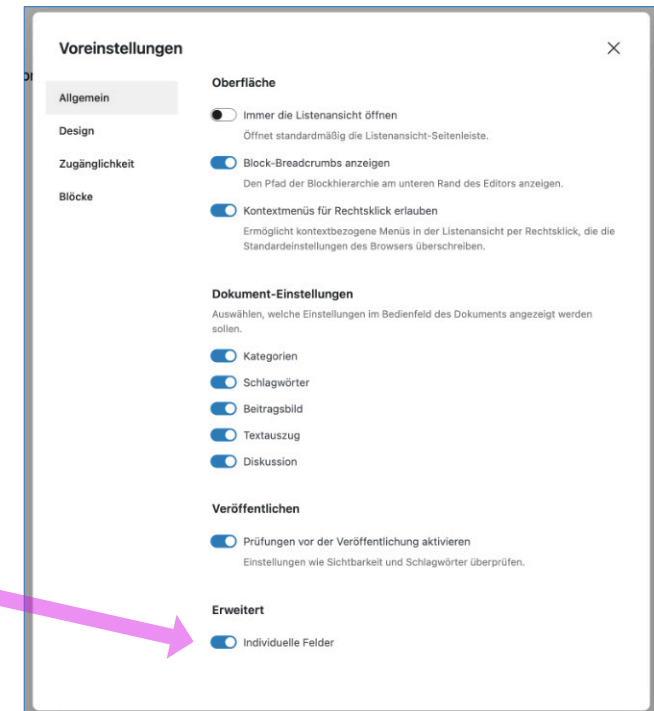
- Kommentarnummer* (sic!) - Anzahl der Kommentare des aktuellen Beitrages
- Kommentar-URL* - URL zum Kommentarform für den aktuellen Beitrag

Diese Einbindungen sind besonders bei der Gestaltung von Templates im Rahmen des Theme Builders sinnvoll. Zum Beispiel um den Beitragstitel anzuzeigen: man kann einfach das entsprechende Widget verwenden um den Titel anzuzeigen, man kann aber genauso gut jedes andere Widget verwenden das dynamischen Text zulässt und dann über die dynamischen Inhalte den Beitragstitel ausgeben lassen.

12.1. Benutzerdefinierte Felder

Gleich die erste Auswahl unter den Dynamischen Inhalten ist *Post Benutzerdefiniertes Feld*. Viele - auch erfahrene - WordPress-User wissen das nicht: seit Version 5.5. unterstützt WordPress von Haus aus - auch ohne Plugin - eigene Felder bei Beiträgen. Das einzige, was man dafür machen muss, ist diese zu aktivieren. Praktisch geschieht dies so, dass man einen Beitrag bearbeitet und ein entsprechendes Feld definiert. Dieses steht dann auch bei den anderen Beiträgen bereit.

WordPress unterstützt keine unterschiedlichen Feldtypen sondern baut



WordPress eigene Individuelle Felder können in den Voreinstellungen aktiviert werden.

einfach Text-Felder ein. Diese müssen zuerst aktiviert werden: dazu muss man einen beliebigen Beitrag in WordPress bearbeiten, dort über *Optionen* (das 3 Punkte-Menü rechts oben) ganz unten auf die *Voreinstellungen* gehen. Dort kann man unter *Allgemein* > *Individuelle Felder* diese Funktion aktivieren.

Ein Individuelles Feld besteht immer aus dem Namen und dem Wert. Der Name ist der Schlüssel, mit dessen Hilfe man den Wert - also den Inhalt - auslesen und anzeigen kann. Da WordPress nur Texte zulässt können nur Textinhalte, URLs oder Farben (also Codes) eingetragen werden.

Trotz dieser Einschränkungen kann es durchaus sinnvoll sein, diese Art Feld zu verwenden. Wer zB. ein Feld für Untertitel zu einem Beitrag benötigt kann damit auskommen. Da die eingegebenen Inhalte aber nicht so einfach überprüft werden können, werden die meisten Anwender sehr schnell auf eine andere Umsetzung des Felderkonzeptes wechseln: **ACF**.

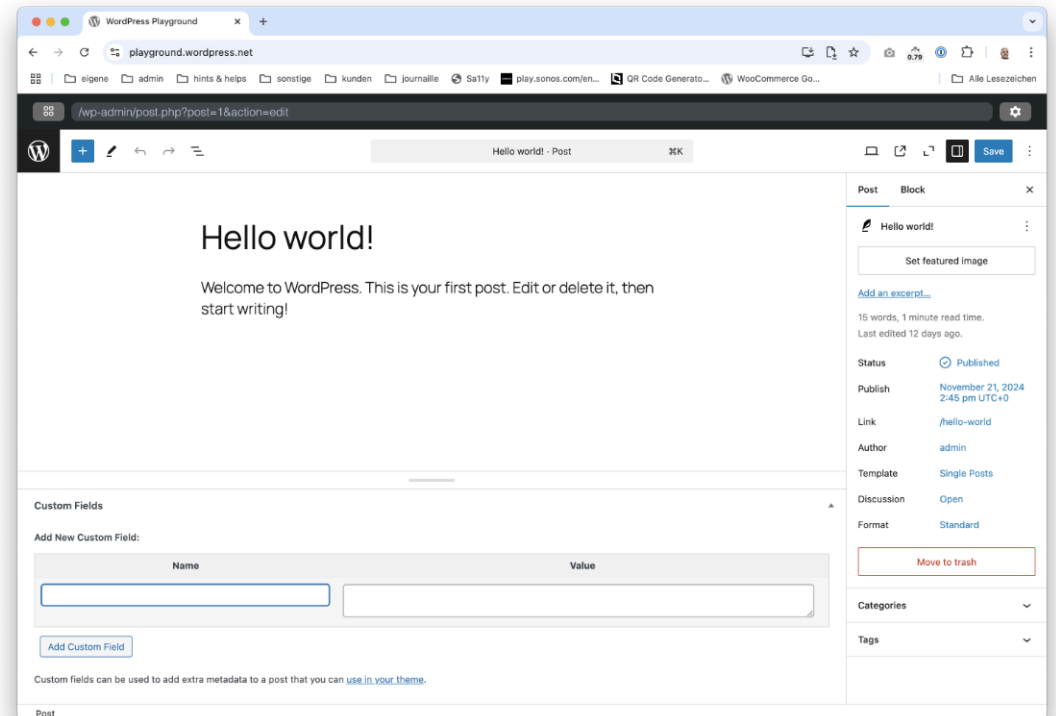
12.2. Advanced Custom Fields - ACF

Das Plugin **ACF** kann bereits in der kostenlosen Version fast alles. Nur wenn man Galerie-, Wiederholungsfelder oder andere spezielle Inhalte benötigt wird man auf die Pro-Version wechseln. Elementor unterstützt auch 2 Alternativen: **Toolset** und **Pods**. Die Empfehlung lautet hier aber eindeutig ACF zu verwenden, auch nicht das von Automattic geforkte **SCF** (#WPDrama).

Wo liegt nun der Vorteil von ACF gegenüber den WordPress-eigenen Feldern? In der Definition von **Feldgruppen** und den **Feldtypen**.

Feldgruppen

Wenn man sich zum Beispiel eine Website oder ein Blog zum Thema *Kochen&Essen* vorstellt, dann könnte es Beiträge mit Rezepten geben und andere mit Restaurant-Besprechungen. Die Beiträge könnten jeweils eigenen Kategorien zugewiesen werden. Je nach Kategorie können dann unterschiedliche Feldgruppen zugewiesen werden: für die Rezepte könnte es Angaben zu Zuta-



Nach erfolgreicher Aktivierung können eigene Felder angelegt und eigene Inhalte für jedes Feld in jedem Beitrag gespeichert werden. Angezeigt werden diese Inhalte über die dynamische Elementor-Funktion *Post Benutzerdefiniertes Feld*.

ten, Kochdauer und zur Schwierigkeit der Zubereitung geben. Für die Restaurantbesprechungen Felder für die Google-Maps, Sterne-Bewertung und ähnliches...

Feldtypen

ACF unterteilt die Felder in folgende Typen:

Grundlegend

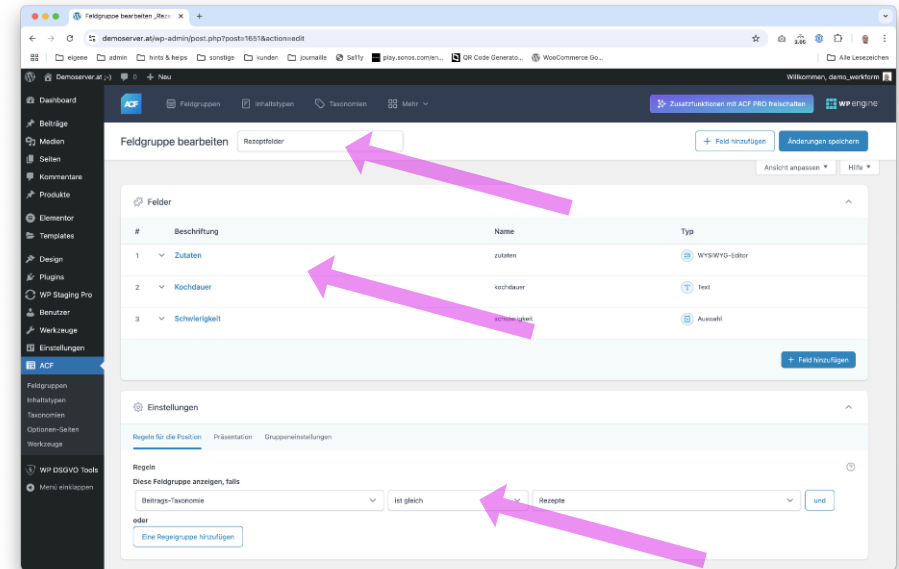
- Text
- Textfeld
- Numerisch
- Bereich
- E-Mail
- URL
- Passwort

Inhalt

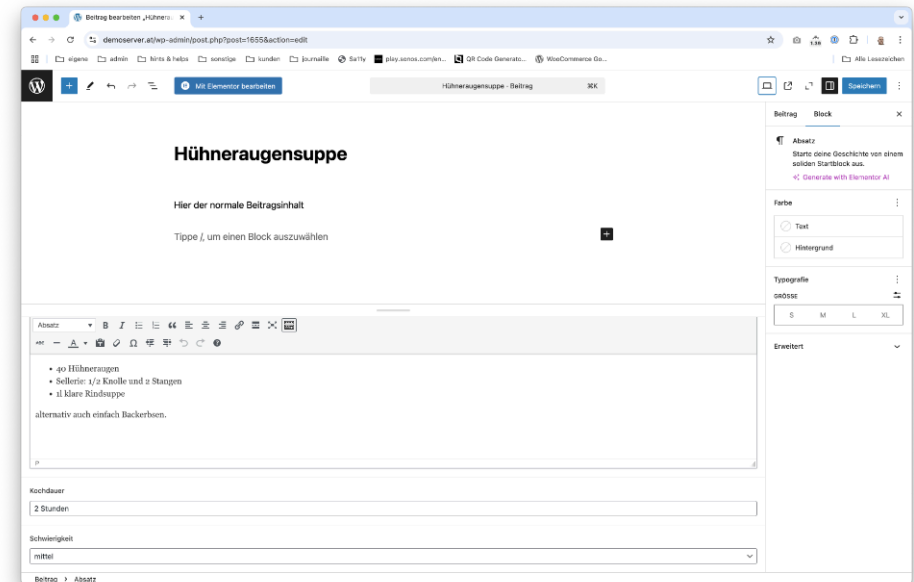
- Bild
- Datei
- WYSIWYG-Editor
- oEmbed
- Galerie (Pro)

Auswahl

- Auswahl
- Auswahlkästchen
- Radio-Button
- Button-Gruppe
- Wahr/Falsch



Ein Beispiel: eine Feldgruppe namens *Rezeptfelder* hat 3 Felder mit verschiedenen Feldtypen bekommen (WYSIWYG-Editor, Text und Auswahl). Diese Felder stehen im WordPress-Backend nur zur Verfügung, wenn der Beitrag in der Kategorie *Rezepte* gelistet ist.



Die einzelnen Rezepte werden als Blockeditor-Beiträge mit ACF-Feldern angelegt.

Relational

- Link
- Beitrags-Objekt
- Seiten-Link
- Beziehung
- Taxonomie
- Benutzer

Erweitert

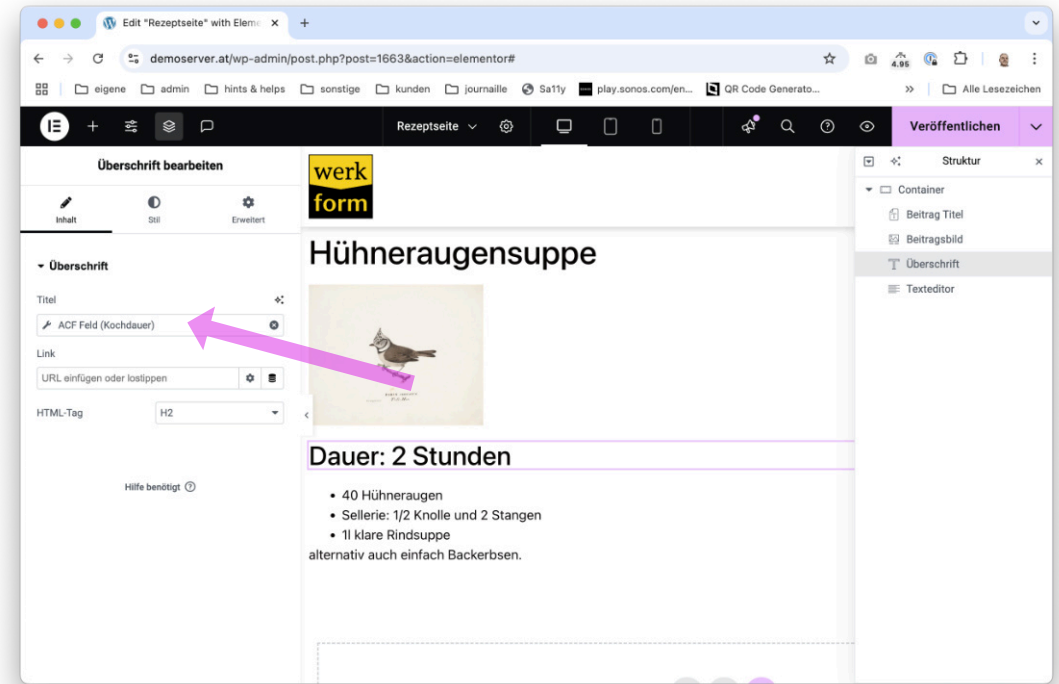
- Google Maps
- Datumpicker
- Datums und Zeitauswahl
- Zeitauswahl
- Farbauswahl
- Icon-Wähler

Layout

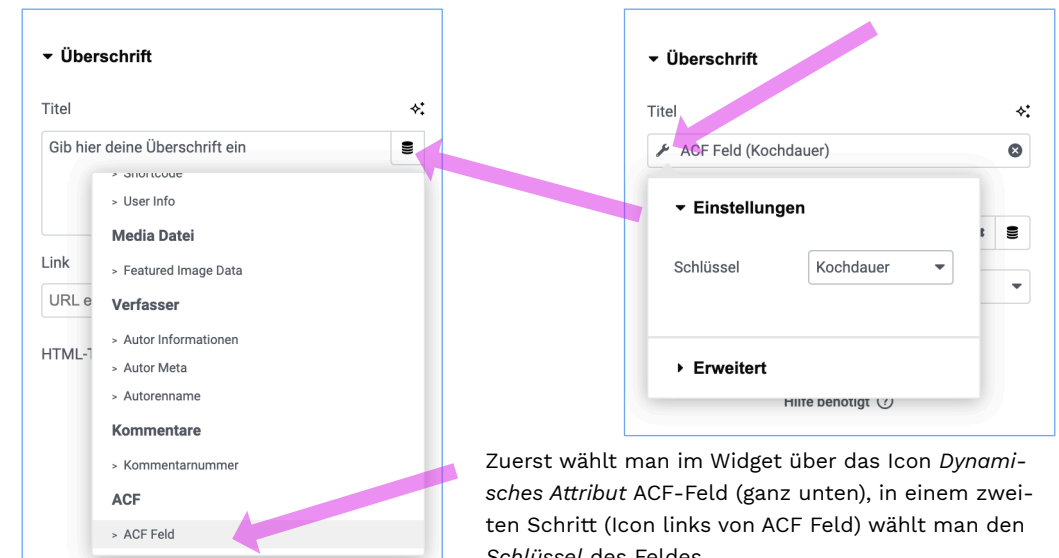
- Mitteilung
- Akkordion
- Tab
- Gruppe
- Wiederholung (Pro)
- Flexible Content (Pro)
- Klon (Pro)

Durch die Wahl des geeigneten Feldtypes erleichtert man die Dateneingabe und sichert das richtige Format. Die Ausgabe der eigenen Felder erfolgt sinnvoller Weise über die Templates für Single Post und Loop.

Je nach Inhaltstyp können nur bestimmte Felder in bestimmten Widgets ausgewählt werden, also Textfelder nur bei Texteingaben, Links nur in Linkfeldern etc.



Im Theme-Builder wird dann das Single-Post Template gestaltet. Während die Widgets Beitrag Titel und Beitragsbild von sich aus dynamisch sind, wird die Überschrift darunter über ein dynamisches Attribut (ACF-Feld) angesteuert. Ebenso der Texteditor darunter.



Zuerst wählt man im Widget über das Icon *Dynamisches Attribut* ACF-Feld (ganz unten), in einem zweiten Schritt (Icon links von ACF Feld) wählt man den *Schlüssel* des Feldes.

12.3. Eigene Post Typen

Wenn nun Kategorien nicht ausreichen oder sinnvoll sind um unterschiedliche Inhalte voneinander zu unterscheiden kann man auch den Weg wählen eigene Post-Typen (Inhaltstypen, CPT [Custom Post Type]) zu definieren. Auch das ist in ACF möglich.

In der Praxis ist es sehr oft der Fall, dass man Inhalte über Post-Typen verwaltet um die Übersichtlichkeit zu bewahren oder auch um unterschiedliche Abfragen einfach durchführen zu können. Taxonomien wie Kategorien oder Schlagworte (Tags) sind oft umständlicher zu handhaben, wogegen es sehr einfach ist Abfragen nach Post-Typen zu definieren und über die Bedingungen im Theme Builder unterschiedliche Templates verschiedenen Typen zuzuordnen. Auch WordPress selbst verwendet diese Technik, zB. sind alle Medien in der Mediathek dem Post-Typ *Attachment* zugeordnet.

Dashboard > ACF > Inhaltstypen

Ist der Weg zur Verwaltung. Wird ein neuer Inhaltstyp angelegt muss man die Beschriftung (in Einzahl und Mehrzahl) angeben, ACF schlägt dann den slug (=Inhaltstyp-Schlüssel) gleich vor, der aber auch geändert werden kann.

Ohne weitere *Erweiterte Konfiguration* kann der neue Inhaltstyp das selbe wie ein Beitrag. Will man Dinge ändern (kein Beitragsbild, Editor, Revisionen etc.) ist das dort zu finden.

Sind die Einstellungen geschafft (keine Sorge, alles ist auch im Nachhinein änderbar) kann man eigene ACF-Feldgruppen diesen neuen Inhaltstypen zuordnen. Es ist also sehr schnell erledigt eigene Inhalte für eine WordPress-Umgebung zu definieren: 1. einen eigenen Typ erstellen und 2. eigene Felder diesem Typ zuordnen.

Wenn nun neue Posts mit dem neuen Post-Typ erstellt werden muss Elementor nicht verwendet werden, of ist es sogar besser, keinen PageBuilder zu verwenden. Statt dessen kommen die Templates des Theme-Builders zum Einsatz. Will man jedoch auch eigene Post-Typen mit dem Elementor Page-BUILDER gestalten können muss unter

Im einfachsten Fall ist das Anlegen eines eigenen Typs mit der Benennung bereits erledigt. Führt man keine *Erweiterte Konfiguration* durch erhält man einen Typ der die selben Merkmale wie der Standardtyp Beitrag hat.

die Checkbox vor dem jeweiligen Typ aktiviert werden.

Nutzen kann man den neuen Typ für **Archive**. Über die Bedingungen kann man den neuen Post-Typ auswählen und so Archivseiten gestalten, die nur für diesen einen Typ aufgerufen werden.

Was für Archive gilt muss auch für **Single-Post** Templates gelten. Auch hier wird über die Conditions (beim ersten Speichern automatisch aufgerufen *oder* über das Popup neben dem Speicherbutton zu finden *oder* über das neue Theme Builder Interface aufzurufen) die Möglichkeit geschaffen einen bestimmten Typ mit einem Template zu verknüpfen. Man kann sogar nur bestimmte Posts eines bestimmten Autors verknüpfen, damit kann man unterschiedliche Designs automatisch unterschiedlichen Bereichen und Autoren zuweisen. Wer's braucht ...

Häufiger zum Einsatz kommen sicher die Abfrage im **Loop-Grid**, den **Beiträgen** oder im **Portfolio-Widget**. Alle drei haben einen Tab *Abfrage* der u.A. dazu da ist, die Quelle der anzuzeigenden Daten einzustellen. Eigene Post-Typen werden dieser Auswahlliste automatisch hinzugefügt.

Es ist die Kombination aus *Dynamischem Content*, *eigenen Post-Typen* mit *eigenen Feldern*, die aus dem Blogsystem WordPress mit Hilfe von Elementor Pro ein vollwertiges und hoch flexibles Content Management System machen kann.