

Theme erstellen in WP mit KI

Bitte beachten

Wir erstellen mit chatGPT in mehreren Stufen ein Theme. Achten Sie darauf, jede Stufe zu behalten, sowohl lokal auf Ihrem Rechner (*als Sicherungsbackup!*) als auch in Wordpress (wir können es dann immer wieder aktivieren und den Fortschritt visualisieren). Kopieren Sie also immer die Ergebnisse einer Stufe und benennen Sie sie um (von Stufe 1 zu 2 und so weiter). Nach jeder Stufe machen Sie sich drei bis fünf kurze Notizen, um das Wichtigste zusammenzufassen, z.B. für Stufe 1:

- `<? get_header() ?>` am Anfang in der index.php und `<? get_footer() ?>` am Ende fügen header, index und footer zu einer einzigen php-Datei zusammen.
- html-head und header-Bereich landen in der header.php, der main-Bereich landet in index.php und der footer-Bereich in footer.php
- In styles.css werden die css-Anweisungen geschrieben, in der functions.php wird die styles.css „geladen“.
- Es erscheinen noch keine Inhalte aus der Datenbank auf der Webseite.
- Es ergeben sich zwei Wege der Installation des Themes:
 - per ftp auf dem Server in themes einen order stufe_1 erstellen und dort die Dateien hochladen
 - oder lokal zippen und dann im Backend (Design → Themes → Hochladen) hochladen und aktivieren

Wir legen den Fokus zunächst auf die Funktionalität. Das Layout bearbeiten wir gesondert in bestimmten Terminen. Wir fügen sonst nur css-Befehle hinzu, die der Übersichtlichkeit oder einer grundsätzlichen Struktur(Grid-Layout)/Funktionalität(Responsivität) dienen (vergleiche Aufgabenstellungen).

Stufe 1 Statische Seite

Erstellen Sie ein supereinfaches/einfachstes Theme für Wordpress mit einer statischen Seite mit Header, Main und Footer und verschiedenen Hintergrundfarben und Platzhaltertext und/oder minimaler Höhe zur Sichtbar-Machung.

Stufe 2 Trennung der Bereiche

Wenn noch nicht in Stufe 1 geschehen, splitten Sie index.php in unterschiedliche php-Dateien für Header, Main und Footer.

Stufe 3 Beiträge, die erste

Lassen Sie im Mainbereich alle Ihre Beiträge anzeigen und untersuchen Sie, welche Bestandteile der Beitrag hat und wie darauf zugegriffen wird.

Stufe 4 Styles und Google Fonts

Sie dürfen ein paar erste Styles für body, main, header und footer anlegen. Verwenden Sie dafür (zumindest testweise) auch Google Fonts.

Achten Sie darauf, grundlegende Styles in übergeordneten Elementen zu definieren und in den Kindelementen nur zu überschreiben, wenn Sie sie diese ändern wollen. (Negativbeispiel: Es ist fehleranfällig überall die gleiche Schrift zu setzen.)

Hier zur Erinnerung:

◆ Übersicht mit KI

In HTML erben alle Elemente von `body` standardmäßig nur textbezogene Eigenschaften wie `font-family` und `color`. Die Hierarchie des HTML-Dokuments bedeutet, dass der `body`-Tag die Wurzel für alle sichtbaren Inhalte bildet und seine textbezogenen Eigenschaften an seine Kinder und deren Nachfahren weitergibt. [🔗](#)

- **Standardmäßige Vererbung:** CSS-Eigenschaften wie `color` (Schriftfarbe), `font-family` (Schriftart) und `line-height` (Zeilenhöhe) werden standardmäßig von übergeordneten Elementen an untergeordnete geerbt, wenn sie nicht explizit überschrieben werden.
- **Beispiel `font-family`:** Wenn Sie dem `body`-Tag eine Schriftart zuweisen, erhalten fast alle anderen Elemente diese Schriftart automatisch, da sie sie vom `body` erben.
- **Beispiel `font-size`:** Wenn Sie dem `body`-Element eine Schriftgröße von `20px` geben, wird diese Größe auch für ein `p`-Element übernommen, da es sie vom `body` erbt. Ein nachfolgendes `strong`-Element innerhalb des `p`-Tags erbt dann die Größe von `20px`.
- **Vererbung von nicht-textbezogenen Eigenschaften:** Nicht alle Eigenschaften sind vererbbar. Eigenschaften wie `background-color` oder `border` werden nicht von `body` an untergeordnete Elemente weitergegeben.
- **Manueller Erzwingen der Vererbung:** Das CSS-Schlüsselwort `inherit` kann verwendet werden, um die Vererbung explizit zu erzwingen, falls das Standardverhalten nicht ausreicht oder überschrieben wurde. [🔗](#)

Stufe 5 Grid-Layout für Posts

Lassen Sie die Posts in einem Grid-Layout ausgeben.

Stufe 6 Menü (komplex)

Erstellen Sie ein responsives/aufklappendes (css + javascript) Menü, in das Sie Ihre Kategorien und Unterkategorien sowie Seiten (im Backend) hinzufügen können. Bei Klick auf eine (Unter-)Kategorie sollen alle zugehörigen Beiträge ausgegeben werden (loop).

(Später/Stufe 9): Überlegen Sie ein sinnvolles Verhalten für einen Klick auf Oberkategorien im Menü (dafür haben wir keine Beiträge angelegt).

Stufe 7 loop au lagern

Einige Codesnippets doppeln sich, vor allem die loop. Daher sollte diese in eine loop.php im Unterordner eine template_parts ausgelagert werden. Ggf. haben Sie unterschiedliche Loops und daher keine Dopplungen. Lagern Sie diese Loops trotzdem aus (damit man sie woanders wiederverwenden könnte) und rufen Sie in Ihren php-Dateien dann jeweils die richtige Loop auf.

Stufe 8 Startseite

Wir haben jetzt eine Menü, über das wir alle Beiträge und alle Unterseiten erreichen können. Sorgen Sie dafür, dass bei der Standard-URL wieder Ihre Startseite ausgegeben wird.

Stufe 9 Kategorien mit Unterkategorien

Wir verfeinern das Verhalten bei Aufruf einer Kategorie. Es sollen nicht mehr alle Beiträge dieser und aller Unterkategorien ausgegeben werden, sondern beim Klick auf eine Kategorie

- a) sollen nur ihre „eigenen“ Beiträge ausgeben – und nicht auch die Beiträge etwaiger Unterkategorien.
- b) Optional: Gibt es kein Beiträge in dieser Kategorie, kleine Notiz „keine aktuellen Beiträge vorhanden“ ausgeben.
- c) Ferner sollen Kacheln mit Links zu allen Unterkategorien ausgegeben werden. Am besten vor den Beiträgen, damit man sie nicht übersieht.

Stufe 10 Ausgabe der Beiträge

Falls das bei Ihnen nicht sowieso zufällig schon genau so geschehen: Beim Anzeigen aller Beiträge einer Kategorie, sollen nicht die gesamten Beiträge angezeigt werden, sondern nur eine Vorschau. Der Beitrag selber soll durch einen Link in dieser Vorschau (entweder im Titel oder ein „weiterlesen“/„mehr“ am Ende) erreichbar sein und dann angezeigt werden.

Optional: In der Beitragsvorschau kann man auch ein Vorschaubild anzeigen lassen. (Dieses Vorschaubild ist aber kein Bild aus dem Beitrag, sondern muss als Vorschaubild im Backend (in jedem Beitrag individuell) hinterlegt werden.)

Stufe 11 Layout-Entwurf

Erstellen Sie mit einem/-r Mitschüler/in zwei Entwürfe für das Webseitenlayout, z.B. mithilfe von figma.

Entscheiden Sie sich anschließend für ein Design und setzen Sie dieses in Ihrem WP-Template um.

Stufe 12 Individuell

Nehmen Sie nun nach eigenem Ermessen wenn gewünscht weitere funktionale Anpassungen vor.