

Prompts aus dem Video: Claude Code Website erstellen

1. Prompt um Astro Website mit Startseite (ggf Unterseiten) von Claude Code anlegen zu lassen

Astro-Website: Startseite von null aufbauen

Du bist mein Entwickler-Assistent. Du bist direkt in dem Ordner geöffnet, in dem meine Website entstehen soll.

Schritt 1 – Kurze Abfrage (noch nichts anlegen)

Stelle mir zuerst diese Fragen und warte auf meine Antworten:

1. ****Thema:**** Worum geht es auf der Website? (Thema / Angebot in 1–2 Sätzen)
2. ****Name:**** Wie heißt das Unternehmen bzw. die Marke?
3. ****Farben:**** Welche Farben soll die Seite haben? (z. B. Hauptfarbe + Akzentfarbe, gern als Hexcode – wenn ich keine nenne, schlage passende vor)
4. ****Stil:**** Welcher Stil soll es sein? (z. B. modern, minimalistisch, elegant, verspielt)
5. ****Weitere Seiten:**** Sollen neben der Startseite noch weitere Seiten angelegt werden? (z. B. Über mich, Leistungen, Kontakt – wenn ja, welche?)

Mehr brauchst du nicht – stelle keine weiteren Rückfragen und triff sinnvolle Annahmen

auf Basis meiner Antworten.

Schritt 2 – Strukturvorschlag (immer noch nichts anlegen)

Schlage mir auf Basis meiner Antworten eine Startseiten-Struktur vor. Orientiere dich an diesem bewährten Aufbau und passe ihn an mein Thema an:

1. **Header:** Logo links, Navigation rechts
2. **Hero-Bereich:** Überschrift, kurzer Einleitungstext, Call-to-Action-Button
3. **Trustbar als Logo-Marquee:** Kundenlogos/Partner laufen langsam in einer Endlosschleife durch – nutze Platzhalter-Logos
4. **Ein-Zeilen-Sektion:** Was ist das Unternehmen und für wen ist es?
5. **Vorteile-Sektion**
6. **„So funktioniert es“-Sektion** (3 Schritte)
7. **Kurze Über-mich/Über-uns-Sektion**
8. **Testimonial-Sektion** (mit Beispiel-Testimonials)
9. **FAQ-Sektion** (4–5 Fragen mit Antworten)
10. **Bottom-Call-to-Action-Sektion**
11. **Footer**

Wichtig: Der Haupt-Call-to-Action gehört in den Hero **und** wird an weiteren passenden Stellen wiederholt (z. B. nach den Vorteilen, nach „So funktioniert es“ oder nach den Testimonials), damit Besucher an jedem Punkt der Seite handeln können.

Wenn du für mein Thema eine andere Reihenfolge oder das Weglassen einzelner Sektionen sinnvoller findest, schlage das begründet vor. Dann **STOPP** – warte auf mein „OK“.

Schritt 3 – erst nach meinem OK

1. **Astro-Projekt anlegen:** Lege direkt in diesem Ordner ein minimales Astro-Projekt an (leere Minimal-Vorlage, keine Zusatz-Integrationen, Paketmanager npm).
2. **BaseLayout erstellen:** Erstelle `src/layouts/BaseLayout.astro` – HTML-Grundgerüst mit `<head>` (Titel + Meta-Description als Props) und `<slot />` für den Seiteninhalt. Header (Logo links, Navigation rechts) und Footer gehören ins BaseLayout, damit sie auf **allen** Seiten erscheinen.
3. **Styles anlegen:** Erstelle `src/styles/global.css` mit meinen Farben und dem gewählten Stil als CSS-Variablen und binde sie im BaseLayout ein.
4. **Beispielbilder besorgen:** Lade passende Beispielbilder herunter, bevor du die Startseite baust. Ziehe dir kostenlose Beispielbilder in passenden Größen (z. B. von Unsplash oder picsum.photos) und speichere sie als Dateien unter `public/images/`. Wähle Motive, die zum Thema passen (z. B. Hero-Bild, Über-mich-Porträt, Porträts für die Testimonials). Für die Logo-Marquee erzeuge schlichte Platzhalter-Logos. Falls der Download nicht möglich ist, erzeuge stattdessen selbst einfache Platzhalter-Bilder per Skript.
5. **Startseite bauen:** Baue die Startseite unter `src/pages/index.astro` exakt nach der abgestimmten Struktur. Die Texte darfst du selbst formulieren – kurz, professionell, zum Thema passend. Binde die heruntergeladenen Bilder aus `public/images/` ein.
6. **Weitere Seiten anlegen:** Lege die gewünschten weiteren Seiten als einfache Platzhalter-Seiten unter `src/pages/` an: BaseLayout nutzen, sinnvollen Titel + Meta-Description setzen, im Body nur eine Überschrift mit dem Seitennamen plus Hinweis „Inhalt folgt“. Verlinke alle Seiten in der Navigation.
7. **Konfiguration:** Setze in `astro.config.mjs` die `site`-Option auf `https://PLATZHALTER-DOMAIN.de`.
8. **Git:** Sorge für eine `.gitignore` (node_modules, dist) und initialisiere ein Git-Repository in diesem Ordner.

9. ****Dev-Server:**** Starte den Dev-Server und nenne mir die lokale Vorschau-Adresse sowie eine kurze Baumübersicht der angelegten Dateien.

2. Prompt um die Website lokal abzurufen

Starte den Dev Server und gib mir die Localhost URL

3. Prompt um Git Repository für dieses Projekt initialisieren (macht Claude meist schon automatisch)

Ich will ein git repository für dieses projekt initialisieren.

4. Prompt um Git Identity zu setzen

Die Git Identity soll nur für dieses repo sein (set it for this repo only). Du sollst mit diesem Namen das repo konfigurieren: [DEIN NAME] und meine Email ist [DEINE EMAIL]

5. Prompt um Github Repository zu verbinden

Link it with the online repo with the following commands:

[HIER DEN COMMAND EINGEBEN, DEN DU IM GITHUB REPOSITORY FINDEST]

6. Prompt um neue Änderungen an Github zu pushen

Committe und pushe die Änderungen zu GitHub