

PhantomWP im Detail: Was du damit machen kannst, ob dynamische Websites möglich sind, und wie es sich im Vergleich zu Bricks Builder und Divi verhält

Wenn du bisher mit **Bricks Builder** oder **Divi** arbeitest und dir **PhantomWP** anschaust, dann ist die wichtigste Erkenntnis gleich am Anfang:

“ **PhantomWP ist kein klassischer WordPress Page Builder.**

Es ist eher eine **Headless-/Static-Site-Plattform auf Basis von Astro**, die **WordPress als Content-Backend** nutzen kann.

Das bedeutet: Du bewegst dich bei PhantomWP in einer **anderen Architekturklasse** als bei Bricks oder Divi.

Bricks und Divi sind in erster Linie Werkzeuge, mit denen du **direkt im WordPress-Frontend bzw. innerhalb des WordPress-Themes** Seiten baust.

PhantomWP trennt dagegen stärker:

- **WordPress = Content-Verwaltung**
- **PhantomWP/Astro = Frontend, Rendering, Deployment**

Dadurch entstehen andere Stärken, andere Grenzen und auch ein anderer Workflow.

1. Was ist PhantomWP überhaupt?

PhantomWP ist laut Doku ein System, mit dem du Websites auf Basis von **Astro** bauen kannst, während du **WordPress als CMS** weiterverwendest.

Die Grundidee ist:

- du bearbeitest Inhalte in WordPress
- du baust das Frontend in PhantomWP
- PhantomWP generiert daraus eine **statische oder moderne serverseitig ausgelieferte Website**
- das Ergebnis kann auf Plattformen wie **Vercel** oder **Cloudflare** deployed werden

Im Kern ist PhantomWP also interessant für alle, die:

- WordPress für Content mögen
- aber das klassische WordPress-Frontend, die Performance oder die Wartung nicht mögen
- moderne Frontend-Technologie einsetzen wollen
- trotzdem nicht komplett von WordPress wegmigrieren möchten

2. Was kann man mit PhantomWP alles machen?

Die kurze Antwort ist:

“ Du kannst damit sehr viele moderne Websites bauen — vor allem schnelle Content-, Marketing- und Headless-Websites — und je nach Setup auch dynamische Funktionen integrieren.

Damit das greifbar wird, schauen wir uns das nach Kategorien an.

2.1 Klassische Unternehmenswebsites

PhantomWP ist sehr gut geeignet für:

- Unternehmensseiten
- Agentur-Websites
- Freelancer-Seiten
- Kanzleien
- Ärzte / Praxen
- Handwerksbetriebe
- lokale Dienstleister
- B2B-Websites

Typische Inhalte:

- Startseite
- Leistungen
- Über uns
- Team
- Referenzen
- Kontakt
- FAQ
- Landingpages
- Karriereseiten

Warum das gut passt:

- solche Seiten sind oft überwiegend **contentgetrieben**
- sie brauchen selten schwere Laufzeit-Logik
- SEO und Ladegeschwindigkeit sind wichtig
- Inhalte ändern sich nicht minütlich

Genau dort spielt PhantomWP seine Stärken aus.

2.2 Landingpages und Marketing-Seiten

Das ist wahrscheinlich einer der stärksten Anwendungsfälle.

Du kannst damit bauen:

- Produkt-Landingpages
- SaaS-Marketingseiten
- Kampagnenseiten
- Sales Pages
- Event-Seiten
- Webinar-Seiten
- Produkt-Launch-Seiten

Vorteile dabei:

- sehr schnelle Ladezeiten
- gute Core Web Vitals
- wenig unnötiges JavaScript
- sehr gute technische Basis für SEO
- saubere Deployments über Git/Vercel

Wenn du Performance und saubere Auslieferung priorisierst, ist PhantomWP hier oft deutlich attraktiver als ein schweres Page-Builder-Setup in WordPress.

2.3 Blogs und Magazine

Mit WordPress als Backend und PhantomWP als Frontend kannst du sehr gut bauen:

- Blogs
- News-Seiten
- Online-Magazine
- Content-Hubs
- redaktionelle Websites
- Wissensportale

Typische Features:

- Blog-Listen
- Kategorien
- Tags
- Autoren-Seiten
- Detailseiten
- Featured Images
- SEO-Metadaten

- Pagination
- Suchseiten

Da WordPress in der Redaktion weiterhin vertraut bleibt, ist das für viele Teams attraktiv:

- Autoren schreiben in WordPress
 - Redakteure pflegen Inhalte in WordPress
 - die Besucher sehen aber ein modernes, schnelles Astro-Frontend
-

2.4 Dokumentation, Resource Center, Wissensseiten

Wenn Inhalte strukturiert sind, eignet sich PhantomWP ebenfalls gut für:

- Dokumentationsseiten
- Help Center
- Knowledge Bases
- Ressourcen-Seiten
- Tutorials
- Glossare
- Content Libraries

Hier kannst du WordPress als redaktionelle Quelle einsetzen oder Inhalte direkt im Projekt strukturieren.

2.5 Portfolios und Showcase-Websites

Sehr gut geeignet für:

- Designer-Portfolios
- Entwickler-Portfolios
- Fotografen
- Agentur-Showcases
- Architekten / Kreative

Gerade hier hilft die Kombination aus:

- guter Medienverwaltung
 - hoher Performance
 - sauberem UI
 - modernen Komponenten
 - Theme-/Font-/Icon-Management
-

2.6 Headless WordPress-Websites

Das ist eigentlich die Kernidee.

PhantomWP kann WordPress anbinden und Inhalte per REST API nutzen, z. B.:

- Posts
- Pages
- Media
- Kategorien
- Tags
- Autoren
- Custom Post Types

Das bedeutet:

- du brauchst WordPress nicht mehr als klassisches Frontend
- du verwendest WordPress nur noch als CMS
- das Frontend wird in Astro/PhantomWP gebaut

Das ist besonders spannend, wenn du WordPress magst, aber Theme-/Plugin-/Template-Chaos reduzieren willst.

2.7 WooCommerce-Headless-Shops

Ja, laut Doku geht auch das — mit der wichtigen Einschränkung:



Die WooCommerce-Integration ist vorhanden, aber als **alpha** beschrieben.

Möglich sind laut Doku u. a.:

- Shop-Startseite
- Produktlisten
- Produktdetailseiten
- Kategorien
- Warenkorb
- Checkout
- Order Complete
- Login / Register
- Passwort zurücksetzen
- Account-Bereich
- Bestellungen
- Adressen
- Einstellungen

Das ist also deutlich mehr als „nur statische Produktseiten“.

Aber: Für einen produktiven, umsatzkritischen Shop würde ich aktuell sehr genau prüfen:

- wie stabil die Flows in deinem konkreten Setup laufen
- welche Zahlungsanbieter du brauchst
- ob Sonderlogik nötig ist
- wie viel Individualentwicklung du einplanen musst

Für experimentelle oder frühe Headless-Woo-Projekte kann das sehr spannend sein. Für hochkritische, komplexe Commerce-Projekte würde ich konservativer evaluieren.

3. Kann man mit PhantomWP auch dynamische Websites

bauen?

Kurze Antwort

Ja — aber nicht im selben Sinn wie bei klassischem WordPress mit PHP-Templates und Plugins auf jeder Anfrage.

PhantomWP ist **nicht nur für komplett starre Webseiten** da.

Du kannst damit **dynamische Funktionen** bauen, aber die Dynamik wird anders erzeugt.

3.1 Was „dynamisch“ überhaupt bedeuten kann

Der Begriff ist unscharf. „Dynamische Website“ kann heißen:

1. Inhalte kommen aus einem CMS
2. Seiten werden serverseitig generiert
3. Nutzer können sich einloggen
4. Formulare senden Daten
5. Warenkörbe und Checkouts existieren
6. Inhalte sind personalisiert
7. Suche, Filter, Live-Interaktionen laufen im Browser
8. Daten ändern sich häufig

PhantomWP kann viele dieser Dinge abdecken, aber die Mechanik ist moderner und entkoppelter.

3.2 Arten von Dynamik, die mit PhantomWP gut funktionieren

A) CMS-getriebene Dynamik

Sehr gut geeignet.

Beispiele:

- Blogposts aus WordPress laden
- Kategorien dynamisch ausgeben
- Autorenprofile bauen
- neue Inhalte in WordPress veröffentlichen und im Frontend darstellen

Das ist eine typische Headless-Dynamik.

B) Clientseitige Interaktivität

Auch möglich.

Beispiele:

- Suchfelder
- Filter
- Tabs
- Accordions
- Sliders
- Formularinteraktionen
- Mobile Menüs
- Cart-State im Browser

Da PhantomWP auf Astro basiert, kannst du interaktive Komponenten gezielt einbauen, statt die ganze Website unnötig mit JS zu beladen.

C) Serverseitige / API-basierte Dynamik

Ebenfalls möglich, je nach Setup.

Beispiele:

- Form-Handling
- Auth-Flows
- Passwort-Reset
- Benutzerstatus
- Checkout-Prozesse
- geschützte Account-Bereiche
- API-Routen

- SSR/Server-Rendering in bestimmten Fällen

Die Doku zeigt z. B.:

- Auth-Endpunkte
 - WooCommerce-Flows
 - serverseitige Logik
 - Deployments auf Plattformen, die serverseitige Funktionen unterstützen
-

D) Personalisierte Inhalte

Teilweise möglich.

Beispiele:

- User-Account-Bereiche
- eingeloggte Kunden
- Bestellansichten
- passwortgeschützte Funktionen

Hier wird es aber schnell komplexer als bei „einfacher statischer Website“. Das ist machbar, aber nicht der primäre No-Code-Komfort-Anwendungsfall.

3.3 Wo PhantomWP dynamisch ist — aber anders als WordPress

Klassisches WordPress:

- bei jedem Request läuft PHP
- Datenbank wird abgefragt
- Plugins greifen ineinander
- Theme rendert das HTML zur Laufzeit

PhantomWP:

- viel wird **vorab generiert**
- Inhalte kommen aus APIs oder vorbereiteten Daten
- Interaktivität wird gezielt ergänzt
- dynamische Teile können über moderne Frontend-/Servermechanismen laufen

Das ist architektonisch meist sauberer und schneller, aber weniger „einfach mal irgendein Plugin aktivieren und es läuft im Theme-Kontext“.

3.4 Was mit PhantomWP schwieriger sein kann

PhantomWP ist **nicht automatisch die beste Wahl**, wenn du stark pluginabhängige, hochdynamische WordPress-Seiten baust, etwa:

- Mitgliederportale mit vielen WordPress-Plugins
- LMS mit engem WP-Plugin-Ecosystem
- Foren
- BuddyPress-artige Community-Plattformen
- komplexe Buchungssysteme
- extrem pluginzentrierte Business-Logik
- stark personalisierte Dashboards
- Portale mit vielen Rollen, Rechten und serverseitigen Prozessen

Nicht unmöglich — aber dann verlässt du das Feld, in dem PhantomWP maximal elegant ist.

4. Für welche Arten von dynamischen Websites ist PhantomWP sinnvoll?

Hier eine praktische Einordnung.

4.1 Sehr sinnvoll

- Unternehmensseiten mit Formularen

- Blogs mit Kategorien, Tags, Autoren
 - News-Seiten
 - Marketingseiten mit Lead-Formularen
 - Resource Center
 - Produktseiten mit Content-Management in WordPress
 - kleinere bis mittlere Headless-Setups
 - einfache bis mittlere dynamische Nutzerinteraktionen
 - schnell ladende, SEO-starke Websites
-

4.2 Sinnvoll, aber genauer prüfen

- WooCommerce-Headless-Shops
 - Account-Bereiche
 - Login-/Register-Flows
 - Search/Filter-Funktionen
 - Sites mit API-gestützten Funktionen
 - redaktionelle Plattformen mit spezifischen Datenmodellen
-

4.3 Eher mit Vorsicht

- sehr komplexe Membership-Plattformen
 - hochgradig pluginbasierte WordPress-Webapps
 - Portale mit vielen Echtzeit- oder Session-Funktionen
 - tiefe Integration in klassische WP-Plugin-Ecosysteme
 - Projekte, bei denen Redakteure pixelgenau visuell im WordPress-Frontend bauen müssen
-

5. Wie ist PhantomWP im Vergleich zu Bricks Builder?

Jetzt zur wahrscheinlich wichtigsten Praxisfrage für dich.

Grundunterschied

Bricks Builder ist ein **visueller Builder innerhalb von WordPress**.

PhantomWP ist eine **separate moderne Frontend-/Build-Umgebung**, die WordPress als CMS nutzen kann.

Bricks = WordPress-native visuelle Seitenerstellung

PhantomWP = Headless-/Astro-basierte Site-Entwicklung

Das ist kein kleiner Unterschied, sondern ein fundamentaler.

5.1 Bricks: Wofür es stark ist

Bricks ist stark, wenn du willst:

- direkt in WordPress bauen
- visuell arbeiten
- Templates innerhalb von WP verwalten
- dynamische WP-Daten direkt im Builder nutzen
- Theme Builder, Query Loops, Conditions etc. im WP-Kontext nutzen
- viele Websites mit einem visuellen Builder schnell umsetzen
- komplexe WP-Strukturen bauen, ohne komplett in ein Frontend-Framework zu wechseln

Bricks ist aus Entwicklersicht oft „sauberer“ und performanter als viele ältere Builder und wird deshalb von vielen Profis bevorzugt.

5.2 PhantomWP gegenüber Bricks: Stärken

PhantomWP ist stärker, wenn du Wert legst auf:

Performance

Astro-/Static-/Headless-Ansatz ist häufig leichter und schneller auszuliefern als ein WordPress-Builder-Frontend.

Sicherheit

Kein klassisches öffentliches WordPress-Frontend nötig. Teilweise kann WordPress sogar stärker abgeschirmt werden.

Trennung von Content und Frontend

WordPress bleibt CMS, Frontend wird separat sauber entwickelt.

Moderne Entwicklungsprozesse

- Git
- Deployments
- Vercel/Cloudflare
- Components
- Theme Tokens
- Build-Prozesse
- Headless-Architektur

Geringere Frontend-Abhängigkeit von WP-Themes/Plugins

Du baust das Frontend nicht „im Theme“, sondern eigenständiger.

5.3 Wo Bricks für viele praktischer bleibt

Bricks ist oft praktischer, wenn:

- du direkt in WordPress visuell bauen willst
- Kunden selbst Layouts innerhalb von WP pflegen sollen
- du viele klassische WP-Dynamiken direkt mit Plugins kombinierst
- du keine Git-/Deploy-/Headless-Workflows willst
- du in einer reinen WordPress-Agenturlogik arbeitest
- du eine „alles in einem System“-Arbeitsweise bevorzugst

Bricks hat hier eine niedrigere Reibung.

5.4 PhantomWP vs Bricks in einem Satz

Bricks ist meist besser für **klassische professionelle WordPress-Websites mit visuellem Builder-Workflow**.

PhantomWP ist meist besser für **moderne, headless, performanceorientierte Websites mit sauber getrenntem Frontend**.

6. Wie ist PhantomWP im Vergleich zu Divi?

Divi ist nochmal etwas anders als Bricks.

Kurz gesagt:

- **Divi** ist stärker auf einen breiten Markt ausgerichtet: Designer, Freelancer, Agenturen, Einsteiger, visuelles Arbeiten
 - **Bricks** wirkt technischer, strukturierter und für viele Profis performanter
 - **PhantomWP** geht architektonisch noch weiter raus aus dem klassischen WordPress-Builder-Modell
-

6.1 Divi: Wofür es stark ist

Divi ist stark, wenn du willst:

- visuell und direkt bauen
- viele Layouts schnell zusammenklicken

- mit einem reifen Ökosystem arbeiten
- Kunden einen gut verständlichen visuellen Workflow geben
- klassische WordPress-Websites ohne Headless-Komplexität umsetzen

Für viele kleine bis mittlere Business-Seiten ist Divi „bequem genug“, solange die Performanceanforderungen nicht extrem sind.

6.2 PhantomWP gegenüber Divi: Vorteile

PhantomWP hat gegenüber Divi oft Vorteile bei:

Performance

Divi kann schwer werden. PhantomWP/Astro ist strukturell auf hohe Geschwindigkeit ausgelegt.

Technische Sauberkeit

Weniger Builder-Overhead, weniger Theme-Abhängigkeit.

Moderne Deployment-Workflows

Git, branchbasierte Deployments, Hosting auf modernen Plattformen.

Sicherheit

Kein klassisches öffentliches WordPress-Rendering.

Headless-Flexibilität

Besser geeignet, wenn WordPress nur Content-Backend sein soll.

6.3 Wo Divi für viele einfacher bleibt

Divi bleibt einfacher, wenn du willst:

- direkt im Browser pixeln
- innerhalb von WordPress bleiben
- ohne Headless-Architektur arbeiten
- ohne Git-/Build-/Deploy-Denke auskommen
- Kunden ein klassisches visueller-Builder-Setup geben

Wenn dein Hauptziel ist: „eine Website schnell in WordPress zusammenbauen und dort komplett verwalten“, ist Divi oft einfacher zugänglich als PhantomWP.

7. PhantomWP vs Bricks vs Divi – der ehrliche Praxisvergleich

Hier die Unterschiede nicht theoretisch, sondern aus echter Projektlogik betrachtet.

7.1 Wenn du schnelle Marketingseiten willst

Beste Optionen:

- PhantomWP
- Bricks

Warum?

- Divi ist hier oft technisch schwerer
- Bricks ist leistungsfähig im WP-Kontext
- PhantomWP ist stark, wenn du maximale Performance willst

Entscheidung:

- willst du im WordPress-Builder bleiben → **Bricks**
 - willst du headless/modern/deploymentorientiert arbeiten → **PhantomWP**
-

7.2 Wenn du klassische Kundenwebsites baust

Beste Optionen:

- Bricks
- Divi
- PhantomWP nur unter bestimmten Bedingungen

Entscheidung:

- Kunde erwartet WordPress-Editor-/Builder-Nähe → **Bricks oder Divi**
 - du willst technisch bessere Basis und bist builderaffin → **Bricks**
 - du willst komplett modernes Frontend und der Kunde redigiert nur Content → **PhantomWP**
-

7.3 Wenn du Blog + SEO + Performance willst

Sehr stark:

- PhantomWP
- Bricks

Schwächer im Vergleich:

- Divi, wenn Performance wirklich Priorität 1 ist

PhantomWP ist hier besonders spannend, weil die Kombination aus:

- WordPress als Redaktion
 - Astro als Frontend
 - statischer Auslieferung
sehr stark ist.
-

7.4 Wenn du WooCommerce willst

Am konservativsten / praktikabelsten heute meist:

- Bricks + WooCommerce
- Divi + WooCommerce

Interessant, aber mit mehr Prüfbedarf:

- PhantomWP + WooCommerce headless

Warum?

WooCommerce ist im klassischen WP-Setup nativ zuhause.

Sobald du headless gehst, wird vieles flexibler — aber auch anspruchsvoller.

Wenn du Standard-Shopflows möglichst unkompliziert willst, bist du mit klassischem WP/Woo oft schneller.

Wenn du maximale Frontend-Kontrolle willst und Headless bewusst suchst, kann PhantomWP spannend sein.

7.5 Wenn du maximale visuelle Freiheit ohne viel Code willst

- **Divi**: gut für visuelles Arbeiten
- **Bricks**: sehr gut, oft professioneller
- **PhantomWP**: weniger „klassischer Page Builder“, stärker strukturierte Entwicklungsumgebung

PhantomWP hat zwar visuelle Hilfen, Component Libraries, Theme Studio usw.

Aber es fühlt sich **nicht** wie ein klassischer Drag-and-Drop-Builder im selben Sinn wie Divi oder

Bricks an.

Das ist wichtig für die Erwartung.

8. Ist PhantomWP ein Ersatz für Bricks oder Divi?

Ja — für manche Projekte.

Nein — für andere.

Die ehrlichste Antwort ist:

“ PhantomWP ersetzt Bricks oder Divi **nicht 1:1**, sondern verfolgt ein anderes Modell.

PhantomWP kann ein Ersatz sein, wenn:

- du WordPress nur noch als CMS willst
 - du das Frontend sauber modernisieren willst
 - du Performance priorisierst
 - du Deployment/Git/Headless ok findest
 - Kunden Inhalte pflegen, aber nicht selbst Layouts im Builder umbauen müssen
 - du technisch kontrollierter arbeiten willst
-

PhantomWP ist eher kein direkter Ersatz, wenn:

- du einen echten visuellen WordPress-Page-Builder suchst
 - Kunden Seiten im WP-Frontend selbst layouten sollen
 - du viele WP-Plugins direkt im Theme-/Template-Kontext verwendest
 - du auf klassische WordPress-Dynamik setzt
 - du „alles in WordPress“ behalten willst
 - du möglichst wenig technische Komplexität im Setup willst
-

9. Was sind die größten Vorteile von PhantomWP?

Hier die wichtigsten Punkte in klarer Form.

9.1 Sehr gute Performance-Basis

Wenn Seiten statisch oder sehr schlank ausgeliefert werden, ist das oft deutlich schneller als klassische Builder-Seiten.

9.2 Bessere Sicherheit

Wenn WordPress nicht mehr das öffentliche Frontend ist, sinkt die Angriffsfläche deutlich.

9.3 WordPress kann als Redaktion bleiben

Das ist ein riesiger Vorteil für Teams, die WordPress als CMS schätzen.

9.4 Moderne Frontend-Architektur

Du bewegst dich in Richtung:

- Komponenten
 - Git
 - Deployments
 - Headless
 - moderne Hosting-Plattformen
 - klare Trennung von Daten und UI
-

9.5 Gute Eignung für SEO- und Content-Projekte

Gerade bei Blogs, Magazinen, Unternehmensseiten und Landingpages ist das sehr attraktiv.

9.6 Bessere Hosting-Optionen

Static/modern-deployed Sites können oft günstiger, robuster und einfacher betrieben werden.

10. Was sind die größten Nachteile oder Hürden?

Genauso wichtig.

10.1 Anderes Denken als bei klassischen Page Buildern

Wenn du von Divi oder Bricks kommst, musst du umdenken.

PhantomWP ist kein „Theme Builder in WordPress“, sondern eine andere Ebene.

10.2 Mehr technische Struktur

Auch wenn es visuelle Tools gibt, ist das System stärker entwicklungsorientiert:

- Files
 - Komponenten
 - Build-Prozesse
 - Deployments
 - Git
 - API-/Headless-Denke
-

10.3 Nicht jedes WP-Plugin-Modell passt gut

Je mehr deine Website direkt vom klassischen WP-Rendering und WP-Plugin-Ausgabemodell lebt, desto schwieriger kann die Migration werden.

10.4 WooCommerce und komplexe Dynamik brauchen genauere Planung

Einfach nur „geht irgendwie“ reicht bei Shops und Member-Bereichen nicht. Da muss sauber geprüft werden.

10.5 Kundenerwartungen können anders sein

Wenn ein Kunde gewohnt ist, in WordPress visuell Seiten zusammenzuschieben, ist PhantomWP u. U. nicht das passende Modell.

11. Für wen ist PhantomWP ideal?

PhantomWP ist besonders sinnvoll für:

Agenturen und Freelancer, die:

- moderne performante Websites bauen wollen
- WordPress als CMS behalten möchten
- aber das Frontend technisch sauberer lösen wollen

Entwicklernahe Teams, die:

- Git und Deployments mögen
- Headless-Architektur verstehen oder lernen wollen
- Astro/modernes Frontend nutzen möchten

Content-getriebene Projekte, die:

- SEO wichtig finden
 - hohe Ladegeschwindigkeit brauchen
 - viele Inhalte aus WordPress ziehen
 - keine extreme Plugin-Abhängigkeit haben
-

12. Für wen sind Bricks oder Divi wahrscheinlich die bessere Wahl?

Bricks eher dann, wenn:

- du professionelle WordPress-Seiten baust
- du innerhalb von WordPress bleiben willst
- du hohe Flexibilität im WP-Kontext brauchst
- du visuell arbeiten willst, aber technisch sauberer als in vielen anderen Buildern

Divi eher dann, wenn:

- du ein bequemes, etabliertes visuelles WordPress-System willst
 - du eher designer-/anwendernah arbeitest
 - Performance nicht absolute Höchstpriorität hat
 - du viele klassische kleine bis mittlere WP-Projekte umsetzt
-

13. Meine klare Einordnung für deinen Fall als Bricks-/Divi-Nutzer

Wenn du bereits **Bricks Builder** und **Divi** nutzt, dann würde ich PhantomWP so einordnen:

PhantomWP ist spannend für dich, wenn du:

- öfter an die Grenzen von WordPress-Performance stößt
- WordPress nur als CMS brauchst
- moderne Frontends aufbauen willst
- technische Qualität, Geschwindigkeit und saubere Deployments wichtiger findest als „alles direkt im Builder“
- bereit bist, einen etwas anderen Workflow zu lernen

PhantomWP ist eher nichts für dich, wenn du:

- den klassischen visuellen Page-Builder-Workflow liebst
 - Kunden aktiv am Seitenlayout bauen lässt
 - stark pluginzentrierte WordPress-Projekte umsetzt
 - möglichst wenig technische Reibung willst
 - hauptsächlich „schnell in WordPress bauen und fertig“ arbeitest
-

14. Praktische Entscheidungshilfe

Hier eine einfache Matrix.

Nimm eher PhantomWP, wenn ...

- du eine schnelle Website mit modernem Stack willst
- WordPress nur CMS sein soll
- SEO/Performance sehr wichtig sind
- du Git/Deployment nicht scheust
- das Frontend sauber getrennt sein soll

Nimm eher Bricks, wenn ...

- du in WordPress bleiben willst
- du einen starken professionellen Builder willst
- du dynamische WP-Daten direkt im Builder nutzt
- du Kunden ein flexibles visuelleres WP-System geben willst

Nimm eher Divi, wenn ...

- du möglichst bequem visuell arbeiten willst
- du im klassischen WP-Ökosystem bleiben willst
- du schnell Seiten für typische Business-Cases bauen möchtest
- Performance nicht dein einziges Hauptkriterium ist

15. Schlussfazit

PhantomWP ist **kein besseres Divi** und auch **kein Bricks-Klon**.

Es ist eher eine **moderne Alternative zum klassischen WordPress-Frontend**.

Du kannst damit sehr viel machen:

- Unternehmensseiten
- Landingpages
- Blogs
- redaktionelle Sites
- Headless WordPress-Projekte
- teilweise auch dynamische Websites
- sogar WooCommerce-Headless-Setups

Ja, dynamische Websites sind möglich, aber anders umgesetzt als im klassischen WordPress-Modell.

PhantomWP ist am stärksten, wenn du **Content, Performance, SEO und moderne Frontend-Architektur** kombinieren willst.

Wenn du dagegen einen **klassischen visuellen WordPress-Builder** suchst, bei dem du komplett innerhalb von WordPress bleibst, dann sind **Bricks** oder **Divi** oft die naheliegendere Wahl.

Die einfachste ehrliche Zusammenfassung ist:

- **Divi** = bequemes visuelles WordPress-Bauen
- **Bricks** = professioneller, moderner WordPress-Builder
- **PhantomWP** = WordPress als CMS, modernes Astro-Frontend, performance- und architekturorientiert

Revision #1

Created 2026-06-29 12:14:23 UTC by art10m

Updated 2026-06-29 12:19:04 UTC by art10m