



Das erweiterte Web-Entwickler Glossar

100+ Begriffe & Definitionen

zs.studio

Ziel dieses Glossars:

Schnelle, verständliche Erklärungen der wichtigsten Begriffe aus Webentwicklung, Design und Digitalstrategie – für Kunden, Neueinsteiger und Profis als Nachschlagewerk.



Technische Grundlagen

HTML (HyperText Markup Language)

Auszeichnungssprache

Basis

Strukturaufbau des Web. Definiert semantische Elemente wie <header>, <nav>, <main>. HTML5 ist aktueller Standard mit eingebauten APIs.

CSS (Cascading Style Sheets)

Styling-Sprache

Basis

Kontrolliert visuelle Präsentation von HTML. Inkl. Layout (Flexbox, Grid), Animationen und Responsive Design. CSS3 ist aktueller Standard.

JavaScript (JS)

Skriptsprache

Basis

Client-seitige Programmierung für Interaktivität. Ermöglicht dynamische Inhalte, API-Kommunikation und moderne SPAs. Basis fast aller Frontend-Frameworks.

HTTPS

Verschlüsselungsprotokoll

Basis

HTTP mit TLS/SSL-Verschlüsselung. Sichere Datenübertragung, Pflicht seit 2017 für gute Rankings und Nutzertrauen. Erkennbar am grünen Schloss im Browser.

DNS

Namensauflösung

Basis

Domain Name System übersetzt Domains in IP-Adressen. Besteht aus A-Records, CNAMEs, MX-Records und TTL-Werten. Kritisch für Erreichbarkeit.

API

Schnittstelle

Mittel

Application Programming Interface. Definiert, wie Programme miteinander kommunizieren. Basis für moderne Webanwendungen und Microservices.

JSON

Datenformat

 Basis

JavaScript Object Notation. Leichtgewichtiges, lesbares Datenformat für API-Kommunikation. Standard im modernen Web.

JWT

Authentifizierung

 Mittel

JSON Web Token. Sichere Methode zur Benutzeroauthentifizierung im Web. Enthält Nutzerdaten als verschlüsselten Token.

DOM

Dokument-Modell

 Mittel

Document Object Model. Baumstruktur der HTML-Elemente. JavaScript manipuliert den DOM für dynamische Inhalte.

TCP/IP

Netzwerk-Protokoll

 Basis

Transmission Control Protocol/Internet Protocol. Grundlagenprotokolle für Datenübertragung im Internet.

SSL/TLS

Verschlüsselung

 Basis

Secure Sockets Layer/Transport Layer Security. Verschlüsselt die Verbindung zwischen Client und Server.

Frontend-Architektur

React

JS-Bibliothek

 Mittel

Komponenten-basierte UI-Bibliothek von Meta. Nutzt Virtual DOM, JSX und Hooks. Marktführer für komplexe, interaktive Webanwendungen.

Vue.js

JS-Framework

 Mittel

Progressives Framework mit deklarativer Syntax. Einfach zu erlernen, flexibel einsetzbar – von Widgets bis zu großen SPAs.

Next.js

React-Framework

 Mittel

Fullstack-Framework für React mit SSR, SSG, Image-Optimierung und API-Routing. Branchenstandard für performante, SEO-freundliche React-Apps.

Svelte/SvelteKit

Kompilier-Framework

 Mittel

Kompiliert Komponenten zu purem JavaScript – kein Virtual DOM. Extrem kleine Bundle-Größen und hohe Performance.

Tailwind CSS

Utility-Framework

 Mittel

Utility-First-Ansatz mit atomaren Klassen (`flex`, `mt-4`, `bg-blue-500`). Schnelle Entwicklung, konsistentes Design, kleine CSS-Bundles.

SASS/SCSS

CSS-Präprozessor

 Basis

Erweitert CSS um Variablen, Nesting und Mixins. Erhöht Wartbarkeit. Wird zu nativem CSS kompiliert.

TypeScript

Typisierte JS

 Mittel

JavaScript mit statischen Typen. Erkennt Fehler bereits im Code-Editor. De-facto Standard für große Codebases und Teams.

WebAssembly

Binär-Format

 Hoch

Low-level-Binärformat für Web. Ermöglicht nahe-native Performance für rechenintensive Aufgaben (3D, Video-Editing).

SPA

Architektur

 Mittel

Single Page Application. Lädt Seite einmal, dann dynamisch. Schnelle Navigation, aber SEO-Herausforderungen.

SSR

Rendern

 Mittel

Server-Side Rendering. HTML wird auf dem Server generiert. Besser für SEO und Ladezeit als reine SPAs.

SSG

Rendern

 Mittel

Static Site Generation. HTML wird bei Build-Time generiert. Extrem schnell und sicher. Ideal für Blogs/Docs.

Hydration

Prozess

 Mittel

Initialisierung von JavaScript auf server-generiertem HTML. Übernimmt Interaktivität nach dem Laden.

PWA

App-Technologie

 Mittel

Progressive Web App. Webanwendung mit App-Funktionen (Offline, Push, Installation). Brücke zwischen Web und Native.

Service Worker

Browser-Skript

 Mittel

Hintergrund-Skript im Browser. Ermöglicht Offline-Funktion, Push-Benachrichtigungen und Caching. Kern von PWAs.

Web Components

Standard

 Basis

Browser-eigene API für wiederverwendbare Komponenten ohne Framework. Shadow DOM, Custom Elements, HTML Templates.

Shadow DOM

DOM-Isolation

 Basis

Kapselt Styling und Struktur von Komponenten. Verhindert CSS-Konflikte. Basis für Web Components.

Virtual DOM

Abstraktion

! Mittel

Virtuelle Kopie des DOM. Frameworks vergleichen Änderungen und rendern nur das Notwendige. Performance-Trick.

JSX

Syntax-Erweiterung

i Basis

JavaScript XML. Ermöglicht HTML-ähnliche Syntax in JavaScript. Wird zu React.createElement() transpiliert.

Hooks

React-Funktion

! Mittel

React-Funktionen wie useState, useEffect für State und Lifecycle. Moderner Ansatz statt Class Components.

**Build-Tools Performance****Vite**

Build-Tool

! Mittel

Moderner Ersatz für Webpack. Nutzt ES-Module für instant HMR und schnelle Produktionsbuilds. Standard für neue Vue/React-Projekte.

Webpack

Bundler

! Mittel

Traditioneller Module Bundler für JS, CSS, Assets. Hochgradig konfigurierbar, aber komplexe Konfiguration.

ESLint/Prettier

Code-Qualität

! Mittel

ESLint: Statische Code-Analyse für Fehler. Prettier: Automatische Code-Formatierung. Gemeinsam garantieren sauberen, einheitlichen Code.

Core Web Vitals

Google-Metriken

⚡ Hoch

Drei Performance-Signale für Ranking: LCP (Ladezeit), FID (Interaktivität), CLS (Stabilität). Pflicht seit 2021.

LCP (Largest Contentful Paint)

Web-Vital

! Mittel

Zeit bis größtes sichtbares Element geladen ist. Ziel: <2,5s. Verbesserung durch Bildoptimierung, Caching, schnelleren Server.

FID (First Input Delay)

Web-Vital

! Mittel

Zeit bis erste Interaktion möglich ist. Ziel: <100ms. Verbesserung durch Reduzierung von JavaScript-Blocking.

CLS (Cumulative Layout Shift)

Web-Vital

! Mittel

Visuelle Stabilität während Ladevorgang. Ziel: <0,1. Verhindert durch feste Bildgrößen und stabile Werbe-Plätze.

Lazy Loading

Lade-Optimierung

! Mittel

Nachladen von Inhalten unter dem Fold erst beim Scrollen. Reduziert initiale Ladezeit und Traffic.

Browser Cache

Speicher-Mechanismus

! Mittel

Speichert statische Assets lokal im Browser. Vermeidet erneutes Herunterladen und beschleunigt wiederholte Besuche.

Server Cache

Backend-Optimierung

! Mittel

Temporäre Datenspeicherung auf Server-Ebene (Redis, Memcached). Reduziert Datenbank-Abfragen und Antwortzeiten.

Tree Shaking

Code-Optimierung

! Mittel

Entfernt ungenutzten Code aus Bundles. Reduziert Bundle-Größe und Ladezeit. Funktioniert mit ES-Modules.

Code Splitting

Bundle-Technik

! Mittel

Aufteilung des Codes in kleinere Chunks. Lädt nur das Notwendige initial. Rest nach Bedarf nachgeladen.

Minification

Code-Komprimierung

i Basis

Entfernt Whitespace, Kommentare, kurze Variablennamen. Reduziert Dateigröße ohne Funktionsverlust.

Source Map

Debug-Hilfe

i Basis

Mapping von kompliziertem Code auf Original-Quelle. Ermöglicht Debugging von minifiziertem Code im Browser.

**API Backend****REST API**

API-Standard

i Basis

Representational State Transfer. HTTP-Standard mit GET, POST, PUT, DELETE. Zustandslos, skalierbar, universell.

GraphQL

Query-Sprache

! Mittel

Flexible API-Abfragen – Clients fragen genau benötigte Felder an. Reduziert Overfetching. Alternative zu REST.

Node.js

JS-Runtime

! Mittel

Server-seitige JavaScript-Umgebung. Hochperformant durch asynchrone Event-Loop. Basis für Express, Next.js.

Express.js

Node-Framework

! Mittel

Minimalistisches Web-Framework für Node.js. Schneller Aufbau von APIs und Servern. De-facto Standard im Node-Ökosystem.

SQL

Datenbank-Sprache

! Mittel

Structured Query Language. Standard für relationale Datenbanken (MySQL, PostgreSQL). Datenabfrage und -manipulation.

NoSQL

Datenbank-Typ

! Mittel

Nicht-relationale Datenbanken (MongoDB, Redis). Flexibles Schema, hohe Skalierbarkeit. Ideal für unstrukturierte Daten.

MongoDB

NoSQL-Datenbank

! Mittel

Dokument-basierte Datenbank mit JSON-ähnlichen Dokumenten. Flexibel, horizontal skalierbar.

PostgreSQL

SQL-Datenbank

! Mittel

Fortgeschrittene relationale Datenbank. Unterstützt JSONB, Volltextsuche, ACID-Compliance. Beliebt bei modernen Stacks.

Redis

In-Memory-DB

! Mittel

Extrem schnelle Key-Value-Datenbank im Arbeitsspeicher. Ideal für Caching, Sessions, Real-time-Features.

OAuth 2.0

Auth-Protokoll

! Mittel

Standard für sichere API-Autorisierung. "Login with Google/Facebook". Delegiert Authentifizierung an Dritte.

Session

Server-Status

! Mittel

Server-seitige Speicherung von Nutzer-Status. Identifiziert wiederkehrende Besucher über Cookies.

Cookie

Client-Speicher

! Mittel

Kleine Textdatei im Browser. Speichert Session-IDs, Präferenzen, Tracking-Daten. Unterliegt DSGVO.

Webhook

Event-API

! Mittel

HTTP-Callback bei Ereignissen. Server benachrichtigt andere Systeme in Echtzeit (z.B. Payment-Status).

Middleware

Filter-Schicht

! Mittel

Software-Schicht zwischen Anfrage und Handler. Prüft Auth, Logging, Rate Limiting. Express.js-Konzept.

ORM

Datenbank-Mapper

! Mittel

Object-Relational Mapping. Abstrahiert SQL-Queries in Objekt-Methoden. Beispiele: Prisma, Sequelize, TypeORM.

Migration

DB-Versionierung

! Mittel

Versionierte Datenbank-Schema-Änderungen. Erlaubt reproduzierbare DB-Updates über alle Umgebungen.

Cron Job

Zeitgesteuerte Aufgabe

! Mittel

Automatisierte Tasks auf dem Server nach Zeitplan (z.B. tägliches Backup, Berichte).

Rate Limiting

API-Schutz

! Mittel

Begrenzung von Anfragen pro IP/Zeitraum. Schützt vor Abuse und DDoS. 429 Too Many Requests.

CORS

Browser-Sicherheit

! Mittel

Cross-Origin Resource Sharing. Regelt, welche Domains auf Ressourcen zugreifen dürfen. Pflicht für APIs.

**Infrastruktur DevOps****CDN**

Content-Network

! Mittel

Globales Verteilnetzwerk für statische Assets. Liefert Inhalte aus geo-nah gelegenen PoPs. Drastische Ladezeitverbesserung.

Docker

Container-Technologie

! Mittel

Isolierte Container für Anwendungen mit allen Dependencies. Garantiert gleiche Umgebung über alle Stages.

CI/CD

Automatisierung

⚡ Hoch

Continuous Integration/Continuous Deployment. Automatisiertes Testen und Deployen bei Code-Änderungen. Standard für professionelle Entwicklung.

Git

Versionskontrolle

ℹ Basis

Distributed Version Control System. Standard für Code-Verwaltung, Team-Kollaboration und Deployment-Workflows.

GitHub/GitLab

Code-Plattform

ℹ Basis

Git-Repository-Hosting mit Issue-Tracking, CI/CD und Kollaboration-Tools. Zentrale Infrastruktur für moderne Entwicklung.

Vercel

Hosting-Plattform

! Mittel

Edge-First-Hosting für moderne Webanwendungen. Integriert mit Next.js, GitHub. Automatische Deployments und globale Edge-Network.

Netlify

Hosting-Plattform

Mittel

JAMstack-fokussiertes Hosting. Continuous Deployment, Form-Handling, Serverless Functions. Beliebt für static Sites.

AWS

Cloud-Plattform

Hoch

Amazon Web Services. Umfassende Cloud-Infrastruktur (S3, EC2, Lambda). Industry Standard für skalierbare Anwendungen.

Nginx

Webserver

Mittel

High-Performance Webserver und Reverse Proxy. Beliebt für Load Balancing, Caching und SSL-Terminierung.

Load Balancer

Verteilung

Hoch

Verteilt Traffic auf mehrere Server. Verhindert Überlastung, erhöht Verfügbarkeit. Grundlage für Skalierung.

Reverse Proxy

Gateway

Mittel

Server zwischen Client und Backend. Handhabt SSL, Caching, Rate Limiting. Versteckt interne Infrastruktur.

SSL/TLS

Verschlüsselung

Basis

Verschlüsselungsprotokolle für sichere Verbindungen. TLS ist Nachfolger von SSL. Pflicht für HTTPS.

SSH

Secure Shell

Basis

Protokoll für sicheren Fernzugriff auf Server. Grundlage für Deployment und Server-Verwaltung.

Environment Variables

Konfiguration

Basis

Umgebungsvariablen für API-Keys, DB-Credentials. Trennt Konfiguration von Code. Sicherer als hartcodierte Werte.

Serverless

Architektur

Hoch

Backend-Lösung ohne Server-Verwaltung. Code wird on-demand ausgeführt (z.B. AWS Lambda). Kosteneffizient und skalierbar.

FaaS

Cloud-Modell

Hoch

Function as a Service. Einzelne Funktionen als Services. Kern von Serverless. Beispiel: AWS Lambda, Vercel Functions.

Edge Computing

Infrastruktur

Hoch

Code-Ausführung an Edge-Locations (nahe beim Nutzer). Reduziert Latenz. Vercel, Cloudflare Workers.

Monorepo

Repo-Strategie

! Mittel

Alle Projekte in einem Git-Repository. Erleichtert Code-Sharing und CI/CD. Tools: Nx, Turborepo, Lerna.

Microservices

Architektur

⚡ Hoch

Anwendung als Sammlung kleiner, unabhängiger Services. Erhöht Komplexität, aber hohe Skalierbarkeit.

Kubernetes

Orchestrierung

⚡ Hoch

Container-Orchestrierungsplattform. Automatisiert Deployment, Skalierung und Verwaltung von Docker-Containern.

**UX Design****Mobile First**

Design-Strategie

! Mittel

Entwicklung beginnt beim kleinsten Screen. Verbessert Performance und UX auf allen Geräten. Progressive Enhancement für Desktop.

Design System

Komponenten-System

⚡ Hoch

Zentralisierte UI-Bibliothek mit Tokens, Patterns und Komponenten. Sicherstellung visueller Konsistenz bei Skalierung.

Atomic Design

Design-Methodologie

! Mittel

Aufteilung von UI in Atome, Moleküle, Organismen, Templates und Pages. Strukturierte Herangehensweise an Design Systems.

Component

UI-Baustein

i Basis

Wiederverwendbares UI-Element wie Button, Card, Modal. Kernkonzept moderner Frameworks (React, Vue, Svelte).

Props/State

Daten-Fluss

! Mittel

Props: Read-only Daten, die an Komponenten übergeben werden. State: Interne, veränderbare Daten der Komponente.

User Journey

UX-Konzept

! Mittel

Kartierung der Schritte, die Nutzer durchlaufen, um Ziel zu erreichen. Identifiziert Reibungspunkte und Optimierungspotenzial.

Wireframe

UX-Prototyp

i Basis

Schematischer Layout-Entwurf ohne Design. Fokus auf Struktur und Inhaltshierarchie.

Mockup	Design-Prototyp	Basis
Hochwertige visuelle Darstellung des finalen Designs. Zeigt Farben, Typografie, Bilder. Noch nicht interaktiv.		
Figma	Design-Tool	Basis
Cloud-basiertes Design- und Prototyping-Tool. Echtzeit-Kollaboration, Components, Auto-Layout. Industry Standard.		
Accessibility	Barrierefreiheit	Mittel
Gestaltung für Menschen mit Behinderungen. WCAG-Richtlinien, Screenreader, Tastatur-Navigation. Pflicht für öffentliche Projekte.		
WCAG	Barrierefreiheits-Standard	Mittel
Web Content Accessibility Guidelines. Internationale Standards für zugängliche Webinhalte. Level A, AA, AAA.		
Dark Mode	Design-Modus	Basis
Dunkles Farbschema als Alternative zum hellen Design. Reduziert Eye-Strain, spart Batterie bei OLED.		
Micro-Interaction	UX-Detail	Mittel
Kleine, funktionale Animationen (Button-Hover, Lade-Spinner). Verbessert Usability und gibt Feedback.		
Onboarding	User-Einführung	Mittel
Geführte Einführung neuer Nutzer in eine Anwendung. Reduziert First-Use-Frustration und churn.		
Design Token	Design-Variablen	Mittel
Wiederverwendbare Design-Werte (Farben, Abstände, Typography). Zentraler Single Source of Truth für Design Systems.		
Component Library	UI-Bibliothek	Mittel
Wiederverwendbare, dokumentierte UI-Komponenten (z.B. Material-UI, Chakra UI). Beschleunigt Entwicklung.		
Storybook	Entwicklungs-Tool	Mittel
Isolierte Umgebung zum Entwickeln und Testen von UI-Komponenten. Dokumentation, Visual Regression Testing.		
Prototyping	Validierung	Mittel
Interaktiver Prototyp zum Testen von Konzepten vor der Entwicklung. Spart Zeit und Ressourcen.		
User Flow	Prozess-Karte	Mittel
Visualisiert den Weg durch alle Screens zur Zielerreichung. Zeigt Entscheidungspunkte und Pfade.		

Usability Testing

Evaluierung

! Mittel

Beobachtung echter Nutzer bei der Bedienung. Identifiziert UX-Probleme frühzeitig.

Persona

User-Archetyp

i Basis

Fiktive Darstellung der Zielgruppe. Hilft beim User-Centered Design und Priorisierung von Features.

Information Architecture

Struktur

! Mittel

Strukturierung und Organisation von Inhalten. Erleichtert Finden und Navigieren. Grundlage für gute UX.

Marketing Analytics

SEO

Suchmaschinen-Optimierung

i Basis

Maßnahmen zur organischen Sichtbarkeit in Suchmaschinen. Technisches SEO + Content-SEO + Backlinks.

Analytics

Daten-Analyse

i Basis

Messung von User-Verhalten mit Tools wie Google Analytics, Plausible, Matomo. Basis für datengetriebene Entscheidungen.

KPI

Erfolgsmessung

i Basis

Key Performance Indicator. Messgrößen wie Conversion Rate, Ladezeit, Bounce Rate zur Erfolgsbewertung.

ROI

Rentabilität

! Mittel

Return on Investment. Verhältnis von Gewinn zu Investition. Kern-Metrik für Geschäftserfolg digitaler Projekte.

Bounce Rate

User-Metrik

! Mittel

Prozent der Nutzer, die Seite ohne Interaktion verlassen. Hohe Rate = schlechte Relevanz oder UX.

Session

Besuchs-Metrik

i Basis

Zeitraum einer Nutzer-Aktivität auf der Website. Standard: 30 Minuten Inaktivität = neue Session.

Pageview

Metrik

i Basis

Zählung jedes einzelnen Seitenaufrufs. Grundlage für Traffic-Analyse.

Event Tracking

Analytics-Methode

! Mittel

Messung spezifischer Nutzer-Interaktionen (Klicks, Downloads, Video-Plays). Basis für Conversion-Optimierung.

UTM-Parameter

Tracking-Code

! Mittel

URL-Parameter für Kampagnen-Tracking (?utm_source=...). Identifiziert Herkunft von Traffic in Analytics.

Landing Page

Marketing-Seite

! Mittel

Zielgerichtete Seite für Kampagnen. Optimiert auf eine einzige Conversion-Action. Kern-Element von CRO.

Sales Funnel

Customer Journey

! Mittel

Trichter-Modell der Kundenreise: Awareness → Interest → Decision → Action. Identifiziert Drop-off-Punkte.

Heatmap

Visualisierung

! Mittel

Farbliche Darstellung von Nutzer-Interaktionen (Klicks, Scrolls). Zeigt, welche Bereiche genutzt werden.

Pixel

Tracking-Code

! Mittel

1x1-Transparentes Bild oder JavaScript für Conversion-Tracking (Meta Pixel, TikTok Pixel). Erfasst User-Actions.

Retargeting

Marketing-Taktik

! Mittel

Werbung an Nutzer, die Website besucht haben. Steigert Conversion durch wiederholte Touchpoints.

DSGVO

Datenschutz

! Mittel

Datenschutz-Grundverordnung. EU-Recht für Nutzerdaten. Pflicht für Cookies, Tracking, personenbezogene Daten.

Conversion Funnel

Ziel-Trichter

! Mittel

Spezifische Steps bis zur Conversion (z.B. Produkt → Warenkorb → Checkout → Kauf). Identifiziert Abbruchpunkte.

Churn Rate

Retention

! Mittel

Prozent der Nutzer, die einen Service kündigen. Wichtige Metrik für SaaS-Subscription-Modelle.

CLV

Kundenwert

! Mittel

Customer Lifetime Value. Prognostizierter Gesamtertrag pro Kunde. Steuerungsgröße für Marketing-Budget.

CPA

Kosten

! Mittel

Cost per Acquisition. Durchschnittliche Kosten pro Kunden-Gewinnung. Kern-Metrik für Kampagnen-Effizienz.

Google Tag Manager

Tag-Management

! Mittel

Zentrale Verwaltung von Tracking-Codes ohne Code-Änderungen. Vereinfacht Implementierung von Analytics und Pixeln.

Wichtigkeits-Level im Glossar

Basis	Wichtig	Essenziel
Jeder sollte kennen	Für Entwickler essenziell	Spezialisten-Wissen