Moderne Webentwicklung mit HTML5, JavaScript und CSS3

Webentwicklung-Basics lernen mit dem Training von ETC

Trainings-ID: **SD-HTML5**
Wissensgarantie: **12 Monate**
Ort: **Wien, Online**

Zum Seminar →

Das nehmen Sie mit

Dieses Training bietet eine umfassende Einführung in moderne Webentwicklung mit HTML5, aktuellen JavaScript-Standards (ES2024) und CSS3. Die vermittelten Grundlagen sind essentiell für die Entwicklung moderner Web-Anwendungen, Progressive Web Apps (PWAs) und Single Page Applications (SPAs).

Es werden aus dem Bereich moderner Frontend-Entwicklung folgende Themen abgedeckt: Semantisches HTML5, moderne CSS-Layout-Techniken (Grid, Flexbox), ES2024+ JavaScript-Features, Komponenten-basierte Entwicklung, State Management, Build-Tools und moderne Entwicklungsworkflows, Performance-Optimierung, Accessibility (a11y) und moderne Web-APIs.

Das Training verwendet aktuelle Best Practices und moderne Entwicklungstools wie Vite, TypeScript und moderne Browser-APIs. Es wird gezeigt, wie Prinzipien und Technologien der modernen Webentwicklung angewendet werden, um skalierbare und wartbare Webanwendungen zu erstellen.

Nach Abschluss dieses Trainings haben die Teilnehmer*innen Wissen zu folgenden Themen:

- Erstellen semantischer und zugänglicher HTML5-Strukturen
- Moderne CSS-Layout-Techniken (CSS Grid, Flexbox, Container Queries)
- JavaScript ES2024+ Features und moderne Sprachkonstrukte

- Verwenden von GitHub Copilot zur Code-Generierung und -Vervollständigung
- Asynchrone Programmierung mit Promises, async/await und moderne APIs
- Komponenten-basierte Architektur und State Management
- Progressive Web Apps (PWAs) und Service Workers
- Performance-Optimierung und Core Web Vitals
- Moderne Build-Tools und Entwicklungsworkflows
- TypeScript Grundlagen für typsichere Entwicklung
- Testing und Debugging moderner Webanwendungen
- Web Security und HTTPS
- Responsive Design und Mobile-First Entwicklung

Programm

Moderne HTML5 und Semantic Web

- Semantische HTML5-Elemente und Zugänglichkeit (a11y)
- Web Components und Custom Elements
- Progressive Enhancement und Graceful Degradation
- SEO-optimierte HTML-Strukturen
- ARIA und Accessibility Best Practices
- Moderne Entwicklungsumgebung (VS Code, Extensions, DevTools)

JavaScript ES2024+ und Moderne Sprachfeatures

- ES2024+ Syntax: Optional Chaining, Nullish Coalescing, Top-level await
- Modules (ESM) und moderne Import/Export-Patterns
- Template Literals und Tagged Templates
- Destructuring, Spread/Rest Operators
- Arrow Functions und Lexical Scoping
- Classes, Inheritance und Private Fields
- Promises, async/await und moderne Asynchronous Patterns
- Iterators, Generators und Async Iterators

Moderne CSS3 und Layout-Techniken

- CSS Grid Layout für komplexe Layouts
- Flexbox für flexible Komponenten-Layouts

- CSS Custom Properties (CSS Variables)
- Container Queries und Element Queries
- CSS Logical Properties für Internationalisierung
- CSS Cascade Layers und @layer
- Modern CSS Selectors (:has(), :is(), :where())
- CSS Houdini und Paint API
- PostCSS und CSS-in-JS Ansätze

Responsive Design und Mobile-First

- Mobile-First Design Patterns
- Advanced Media Queries und Breakpoint-Strategien
- Viewport Meta Tag und moderne Viewport-Einheiten
- Touch-Gesten und Mobile Interaction Patterns
- Performance für Mobile Geräte
- Progressive Web App (PWA) Design Principles

Moderne Web-APIs und Browser-Features

- Fetch API und moderne HTTP-Client-Patterns
- Aufruf von OpenAPI aus dem Browser
- Web Storage (localStorage, sessionStorage, IndexedDB)
- Intersection Observer für Performance-optimiertes Scrolling
- Resize Observer und Mutation Observer
- Web Workers und Shared Workers
- Service Workers für Offline-Funktionalität
- Push Notifications und Background Sync
- WebAssembly (WASM) Integration
- File System Access API
- Web Share API und Navigator APIs

Asynchrone Programmierung und State Management

- Promise Patterns und Error Handling
- async/await Best Practices
- RxJS und Reactive Programming
- State Management Patterns (Redux, Zustand, Context API)
- Event-driven Architecture

- WebSockets und Server-Sent Events (SSE)
- Real-time Data Synchronization

Component-Based Architecture

- Vanilla JavaScript Components
- Web Components (Custom Elements, Shadow DOM, HTML Templates)
- Component Lifecycle und State Management
- Props, Events und Component Communication
- Styling Strategies für Components
- Testing von Komponenten

Progressive Web Apps (PWAs)

- Service Worker Lifecycle und Caching Strategies
- App Manifest und Installation
- Offline-First Design Patterns
- Background Sync und Push Notifications
- PWA Performance Optimization
- App Shell Architecture

Performance und Optimierung

- Core Web Vitals (LCP, FID, CLS)
- Critical Rendering Path Optimization
- Lazy Loading und Code Splitting
- Image Optimization (WebP, AVIF, Responsive Images)
- Bundle Analysis und Tree Shaking
- Performance Monitoring und Analytics
- Lighthouse und Web Vitals Measurement

Moderne Entwicklungstools und Workflows

- Vite und moderne Build-Tools
- Aspire CLI für lokale Entwicklung
- Einsatz von GitHub Copilot für HTML, CSS, und JavaScript
- Package Managers (npm, yarn, pnpm)
- TypeScript Grundlagen und Integration
- ESLint, Prettier und Code Quality Tools

- Git Workflows und Collaboration
- CI/CD für Frontend-Projekte
- Testing (Jest, Vitest, Playwright, Testing Library)

Web Security und Best Practices

- Content Security Policy (CSP)
- Cross-Origin Resource Sharing (CORS)
- Authentication Patterns (JWT, OAuth)
- HTTPS und TLS Best Practices
- XSS, CSRF und andere Sicherheitslücken
- Secure Coding Practices

TypeScript Fundamentals

- TypeScript Typen und Interfaces
- Generics und Advanced Types
- Type Guards und Type Assertions
- Module System und Declaration Files
- Integration mit JavaScript-Projekten
- Migration Strategies von JavaScript zu TypeScript

Testing und Debugging

- Unit Testing mit Jest/Vitest
- Integration Testing mit Testing Library
- End-to-End Testing mit Playwright/Cypress
- Browser DevTools Advanced Features
- Performance Profiling und Memory Debugging
- Error Monitoring und Logging

Deployment und DevOps

- Static Site Hosting (Netlify, Vercel, GitHub Pages)
- CDN und Edge Computing
- Environment Management
- Monitoring und Analytics
- Performance Budgets und Continuous Performance

Zielgruppen

- Entwickler*innen mit grundlegenden Programmierkenntnissen, die moderne Webentwicklung lernen oder ihre Frontend-Skills auf den neuesten Stand bringen möchten. Ideal für:
- Fullstack-Entwickler, die ihre Frontend-Kenntnisse vertiefen wollen
- Backend-Entwickler, die in die Frontend-Entwicklung einsteigen
- Bestehende Frontend-Entwickler, die moderne Technologien und Best Practices lernen möchten

Vorkenntnisse

- Grundlegende Programmiererfahrung** in einer beliebigen Sprache
- Verständnis von Web-Grundlagen**: HTTP, URLs, Browser-Funktionsweise
- Basis HTML-Kenntnisse**: Grundlegende Tags und Dokumentstruktur
- Basis CSS-Kenntnisse**: Selektoren, Properties, Box Model
- Grundlegendes Verständnis von Versionskontrolle** (Git)
- Kommandozeile/Terminal** Grundkenntnisse

Nach diesem Training können Teilnehmer*innen folgende spezialisierte Trainings besuchen:

- Programming Web Applications with ASP.NET Core and C#
- Programming Web Applications with ASP.NET Core Blazor
- Angular
- smart Implementing Reactive Web Apps using React and Redux
- Vue Fundamentals

Trainings zur Vorbereitung

- Programmierung von Webanwendungen mit ASP.NET Core und Blazor
- ITS: Software Development Fundamentals
- ITS: HTML5 Application Development Fundamentals
- Programmierung für Einsteiger

Termine & Optionen

Datum	Dauer	Ort	Angebot	Preis
23.11.2026-27.11.2026	5 Tage	Wien	Trainingspreis (Vor Ort)	€ 2.620,-
23.11.2026-27.11.2026	5 Tage	Wien	Trainingspreis (Online)	€ 2.620,-
31.05.2027-04.06.2027	5 Tage	Wien	Trainingspreis (Vor Ort)	€ 2.620,-
31.05.2027-04.06.2027	5 Tage	Wien	Trainingspreis (Online)	€ 2.620,-
13.12.2027-17.12.2027	5 Tage	Wien	Trainingspreis (Vor Ort)	€ 2.620,-
13.12.2027-17.12.2027	5 Tage	Wien	Trainingspreis (Online)	€ 2.620,-