

# **ORGANISATION UND EINFÜHRUNG IN HTML5, CSS3 UND JS**

**MMWP2024 - LV01**

## INHALTSVERZEICHNISS

- Organisation
- Aktuelle und historische Daten
- Strukturen von Webseiten
- Tools und Entwicklungshilfen

## VERANSTALTUNG

- Wöchentliche Vorlesung  
(DO 15:30-17:00, Sebastian Pötter)
- Wöchentliche Übungen  
(FR 09:30-15:00, Sebastian Pötter, Ben Beckschäfer)
- 3 Belege für Prüfungszulassung
- Alternative Prüfungsform: Projektarbeit
- Sonst wie im Modulux: schriftliche Prüfung (120min)

# OPAL - MODULKURS

Wählen Sie sich in den Kurs über Opal, den QR-Code oder dem Link in den Kurs ein.

<https://bildungspotrait.sachsen.de/opal/auth/RepositoryEntry/46557921282>



## OPAL - SEMINARGRUPPEN

Teilen Sie sich in den Seminaren ein (nur so viel Platz wie im Seminarraum).

[https://bildungsportal.sachsen.de/opal/auth/  
RepositoryEntry/46557921282/  
CourseNode/1728441152683815006](https://bildungsportal.sachsen.de/opal/auth/RepositoryEntry/46557921282/CourseNode/1728441152683815006)

# VORLESUNGSFOLIEN

Link zu den Vorlesungsfolien im Opal (PDF, Link) oder über:

<https://poetter-sebastian.github.io/MMWP-2024/>



## MODULSCHWERPUNKTE

- Einführung in Webprogrammierung (HTML5, CSS3, ECMAScript-Javascript)
- Einführung in die Browser API
- Vorstellung von HTML/CSS/JS-Frameworks
- Verwendung von Medien im Browser (Text, Bild, Ton, Video)
- Barrierefreiheit, SEO und Ladezeitenoptimierung
- Selbständige Vertiefung, Erwerb praktischer Erfahrungen

## 4 PRÜFUNGSVORLEISTUNGEN (BELEGE) - 1

- Erstellung einer OnePage-Seite mit Grid- und Flexbox-Design
  - Verschiedene Layouts
  - 4 Container mit passend gewählten Layout und Inhalten
- Erstellung einer responsiven OnePager-Seite mithilfe eines Frameworks
  - Wahlweise Bootstrap5 oder ähnlichen Framework/Bibliothek
  - Header mit Navigationsmenü (Sprunganker zu Inhalten)
  - 6 Container mit passend gewählten Layout und Inhalten
  - Einbindung von SVG- oder Canvas-Elementen
  - Footer-Element



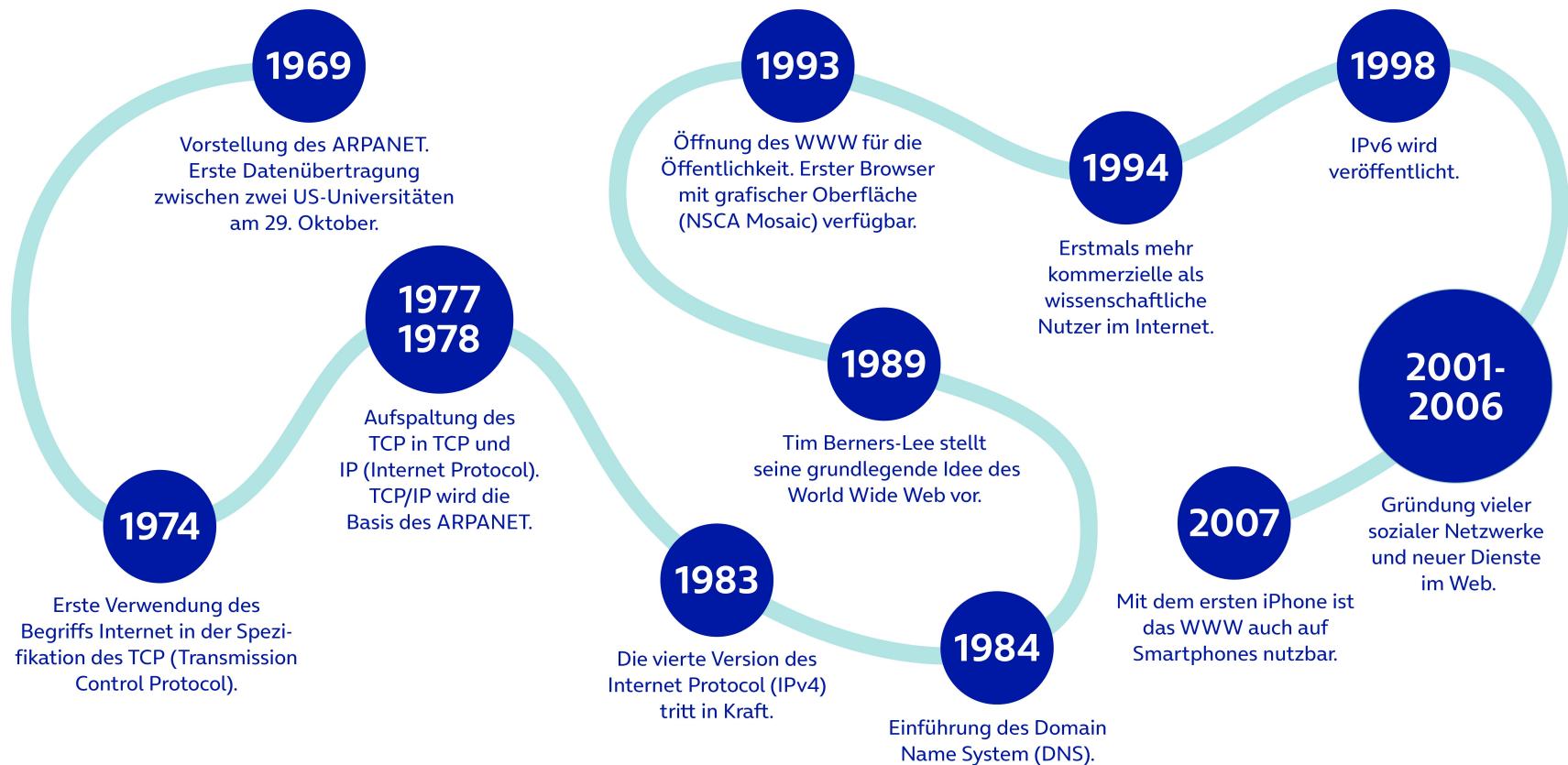
- Erstellungen von verschiedenen User-Interaktionen mit Webseiten über Javascript
  - Verschiedene Aufgaben zu Inputs und anderen Interaktionen
  - Verwendung von asynchronen Javascript Elementen
  - Zufällige Belegaufgaben
- Vortrag zu Webtechnologien oder Web-frameworks in den Seminaren
  - 3 Personen je Gruppe (5min pro Person)
  - 5 Minuten für Fragen
  - Vorstellung und Eingliederung der Technologie
  - Beispiele oder Anwendungsfälle präsentieren
  - Zukünftige Entwicklungen und Möglichkeiten aufzeigen

## ALTERNATIVE PRÜFUNGSFORM

- Erarbeitung eines Web-Portfolios (SEO Optimierung, möglichst barrierefrei und Ladezeitenoptimiert)
- Alle Übungen und Belege gesammelt, aufbereitet auf einer kleinen eigenen Seiten
- Einsatz von verschiedenen Medienformaten zur Darstellung
- Bericht (2-3 Seiten) über eingesetzte Medien, Erfahrung und Lernerkenntnissen

# ANFÄNGE DES INTERNETS

## Wichtige Stationen der Internetgeschichte



Source

# DIE ERSTE INTERNETSEITE

## World Wide Web

The WorldWideWeb (W3) is a wide-area [hypermedia](#) information retrieval initiative aiming to give universal access to a large universe of documents.

Everything there is online about W3 is linked directly or indirectly to this document, including an [executive summary](#) of the project, [Mailing lists](#), [Policy](#), November's [W3 news](#), [Frequently Asked Questions](#).

### [What's out there?](#)

Pointers to the world's online information, [subjects](#), [W3 servers](#), etc.

### [Help](#)

on the browser you are using

### [Software Products](#)

A list of W3 project components and their current state. (e.g. [Line Mode](#), X11 [Viola](#), [NeXTStep](#), [Servers](#), [Tools](#), [Mail robot](#), [Library](#))

### [Technical](#)

Details of protocols, formats, program internals etc

### [Bibliography](#)

Paper documentation on W3 and references.

### [People](#)

A list of some people involved in the project.

### [History](#)

A summary of the history of the project.

### [How can I help ?](#)

If you would like to support the web..

Source

## Tools und Entwicklungshilfen

# WEBENTWICKLUNG - 1

Web 1.0



Web 2.0



Digital Content

<https://itapro.co.th/index.php/services/-innovation-services>

Web 3.0



Semantic Web

<https://datanova.com.au/what-is-the-semantic-web/>

Web 4.0



Internet Of Things

<https://www.alibabacloud.com/de/solutions/iot>



## Tools und Entwicklungshilfen

# WEBENTWICKLUNG - 2

## Web 1.0

Erstellung von Informationen  
Medieninhalten  
Content-Management /DRM  
Responsive Design/Accessibility  
Services  
Bewegung  
Maschinen  
Digital Content

## Web 2.0

- Kommunikation
- Data-/Media- Sharing
- Mobility
- Internet-Sicherheit
- Social-Analytics/Privacy
- Citizen Participation
- Open Science
- E-Services/Collaboration

## Social Web

## Web 3.0

- Einsatz von KI
- Interoperabilität von E-Services
- Big-Data-Analytics/Wissensextraktion
- Cloud-Services
- Performance und Grid-Computing

## Semantic Web

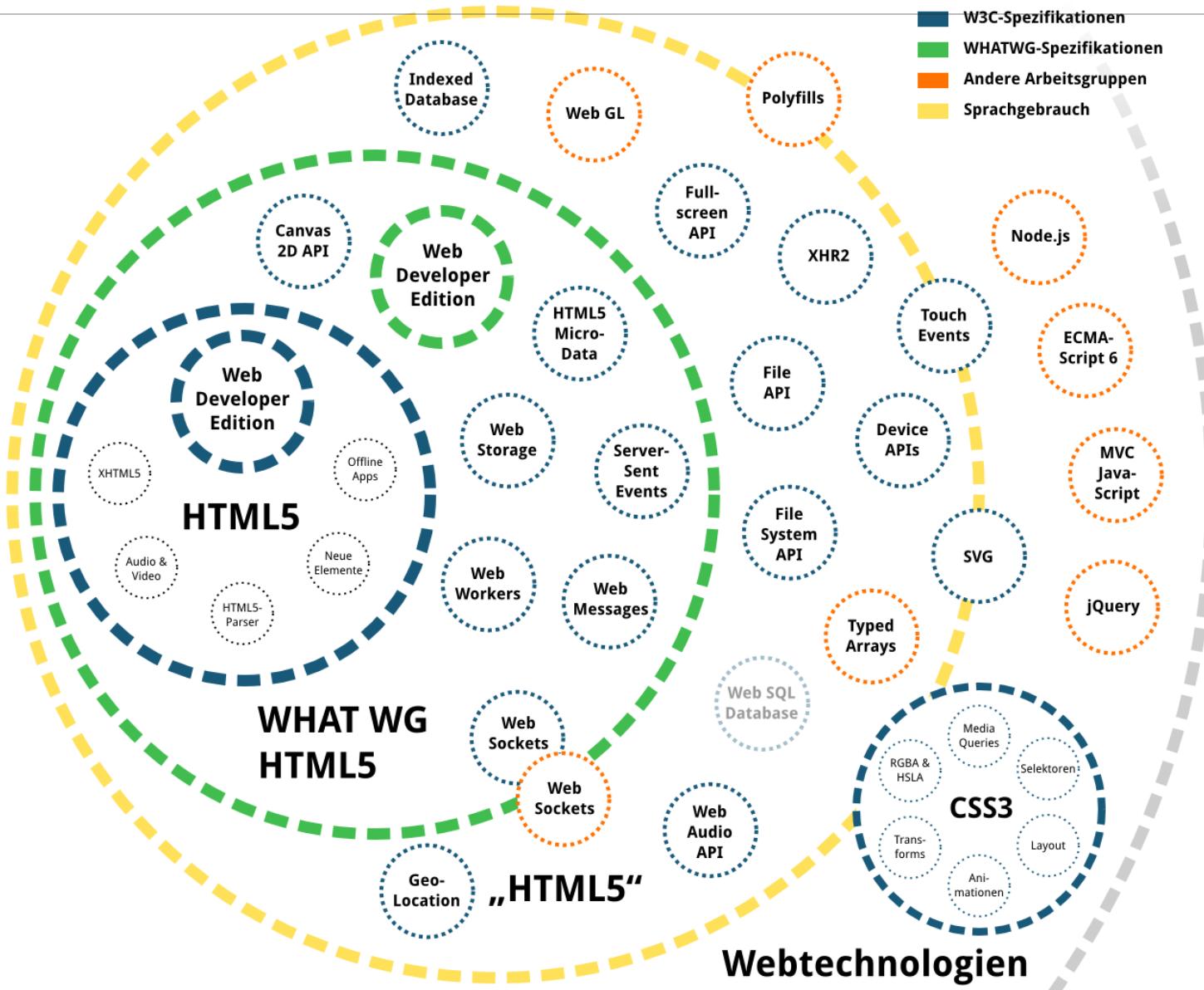
## Web 4.0

- Sensoren-Daten
- Big-Data-Analytics
- KI/Autonomy
- Kommunikation von Geräten/Collaboration
- Cyber-Sicherheit

## Internet Of Things

# ÜBERSICHT VON WEBTECHNOLOGIEN

Tools und Entwicklungshilfen



Source

# BEGRIFFLICHKEITEN RUND UM WEBTECHNOLOGIEN

## Tools und Entwicklungshilfen

<b>Library</b>	<b>Framework</b>	<b>IDE – Integrated Development Environment</b>	<b>SDK – Software Development Kit</b>	<b>API – Application Programming Interface</b>
ist eine Programm-Code-Kollektion (Bibliothek), die spezifische Objekte/Funktionen implementiert und in einem Softwareprogramm genutzt werden kann. z.B. grafische oder mathematische Bibliothek.	ist ein Gerüst von spezifischen Funktionen und Designpatterns auf einer vorgegebenen Softwarearchitektur und dient als Umgebung zur Entwicklung von spezifischen Anwendungen. z.B. Java Swing Framework für GUI oder Model-View-Controller Framework für Web-Anwendungen, oder MS .NET	ist eine Umgebung zur Entwicklung von Software mit Editor-, Debugging-, Compilern- und Ausführungsfunktion. z.B. Eclipse, MS-Visual Studio  <b>Toolkit</b> ist ein Entwicklungsinstrument	ist eine Sammlung aus Programmierwerkzeugen und Bibliotheken. Ermöglicht auch die Erschließung von Systemfunktionen und Ausführung der Anwendungen. z.B Java-SDK	ist eine Schnittstelle/Sammlung von Funktionen zur Kommunikation und Nutzung von existierenden Anwendungen und Services

[oholicconfessions.wordpress.com/2017/06/13/difference-between-ide-library-framework-api-and-sdk/](http://oholicconfessions.wordpress.com/2017/06/13/difference-between-ide-library-framework-api-and-sdk/)

## Source

## AKTUELLE BEWEGUNG

- Accessible Rich Internet Applications (ARIA)
- Mobile first (neuer Stand 2024)
- Dezentrale Interoperabilität (ab ca. 2014)
  - ActivityPUP (Protokoll)
  - Distributed Web (Blockchain)
- VR/XR/AR Webpages
- Single-Page Application (SPAs)
- Serverless Architecture
- Progressive Web Apps (PWA)
- Low-/No-Code Webpages

## AKTUELLE BEWEGUNG - ACTIVITYPUP

- [Mastodon](#), [BlueSky](#), [Threads](#) als Microblogging
- [Lemmy](#), [KBin](#) als Forumservice
- [Peertube](#), [Funkwhale](#), [Castopod](#) oder [Pixelfed](#) für Multimedia-Anwendungen
- [OwnCast](#) für Livestreaming
- "[Flohmarkt](#)" als dezentraler (lokaler) Marktplatz
- "[Goldfish Social](#)" als Kurzvideo Platform

## AKTUELLE BEWEGUNG - DISTRIBUTED WEB

- Blockchain Domain Name Systems (B-DNS)
- [Brave Ad system](#) als Cryptocurrency ADs-System
- Decentralized applications (dapps)

## AKTUELLE BEWEGUNG - VR/XR/AR WEBPAGES

- Immersive Web Toolkit (VR, XR, AR)
- Three.js (WebGPU und Canvas)
- Needle (Unity with WebAssembly und WebGPU)

## AKTUELLE BEWEGUNG - SINGLE-PAGE APPLICATION (SPAS)

- Kein Neuladen der Seite, alle Inhalte werden vorab oder asynchron geladen
- Schlechtes SEO, wenn keine Inhalte vorab compiliert wurden
- [React](#)
- [Svelte](#)
- [Vue](#)
- [Angular](#)

## AKTUELLE BEWEGUNG - PROGRESSIVE WEB-APPS (PWA)

- Installierbare Website als Browser-App
- Serviceworker und Webmanifest zur Einstellung und Synchronisierung der Web-App
- Viele SPAs unterstützen PWAs

## AKTUELLE BEWEGUNG - LOW-/NO-CODE WEBPAGES

- Keine/wenige Programmierkenntnisse nötig
- Baukastensystem für Webseitenerstellung, z.B.:
- [Webflow](#)
- [Wordpress](#)
- [Wix](#)

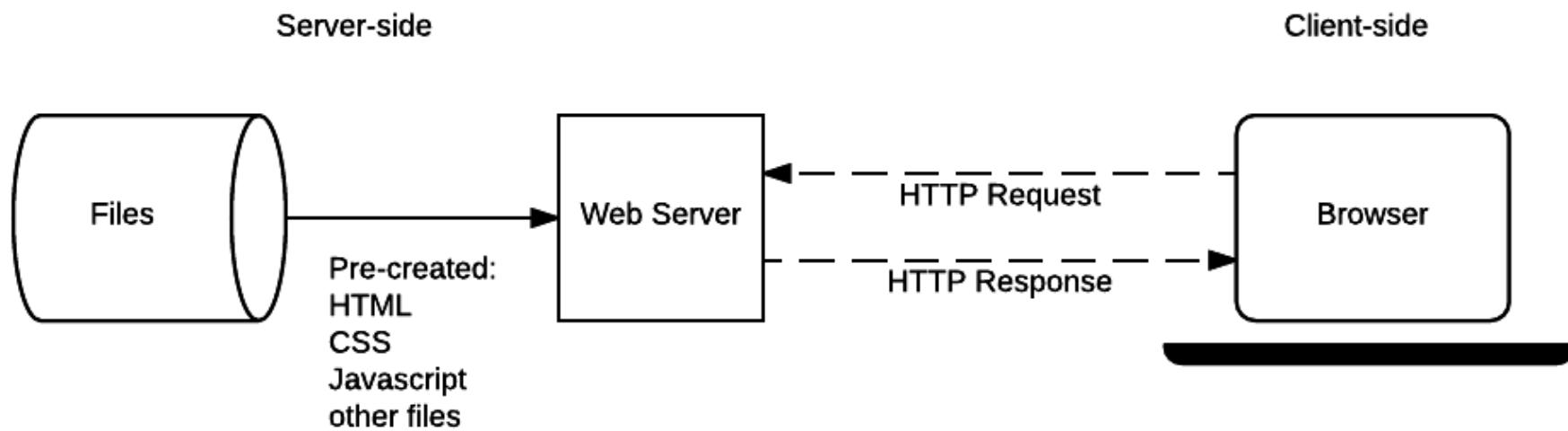
# ÜBERSICHT VON MODERNEN BROWSERN

- Chromium/Blink-Based Browser:
  - Google Chrome
  - Microsoft Edge
  - Opera
  - Brave
- Gecko/Quantum-Based (Free HTML layout engine):
  - Mozilla Firefox
  - TOR (-Browser)
  - Mullvad
  - Waterfox

## AUFBAU VON WEBSEITEN

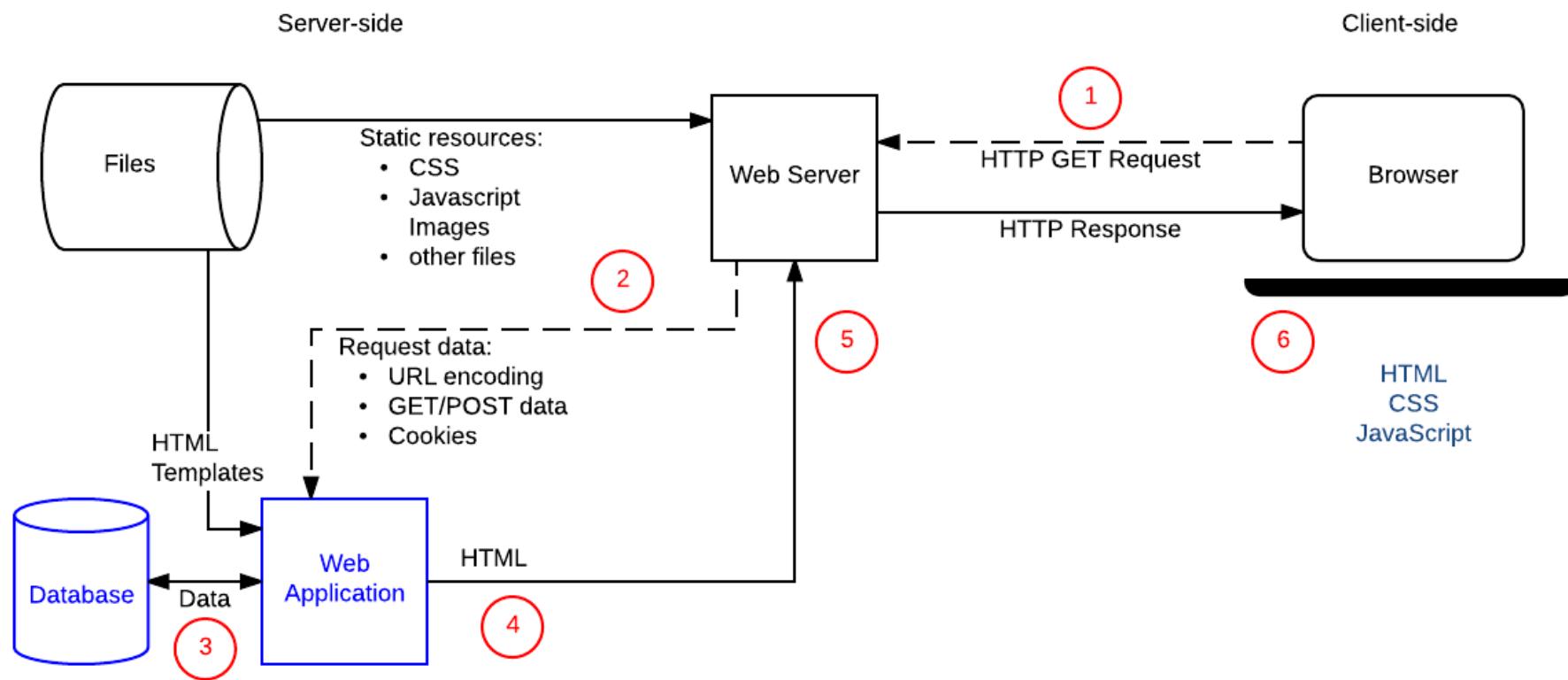
- Serverstruktur zur Bereitstellung
- HTML Dokument
  - Wie ein Netzwerkpaket aufgebaut
    - Header (Meta-Daten und Bibliotheken)
    - Body (Struktur, Asynchrone Bibliotheken)

# STATISCHE WEBSEITEN



Source

# DYNAMISCHE WEBSEITEN



Source

## DEKLARATIVE VERSUS IMPERATIVE PROGRAMMIERUNG

Bei deklarativer Programmierung wird vorgegeben, mit welchen Eingabeparametern welche qualitative Anweisung vom Computer auszuführen ist und (implizit) was als Ergebnis erwartet wird. => „Was ist umzusetzen.“

Dabei wird es der ausführenden Applikation überlassen, welche Algorithmen angewandt werden und wie das Ergebnis dann dargestellt wird. „Wie ist es umzusetzen.“ (z.B. durch Browser Engines bei HTML5)

Für das dynamische Verhalten der Applikationen (z.B. nach Interaktion mit dem Nutzer) sind imperative Programmanteile zulässig, um die meist festgelegte Syntax deklarativer Programmiersprachen um neue Funktionalitäten erweitern zu können oder um den Quellcode (z.B. via DOM) dynamisch zu verändern. Bei HTML5 ist es JavaScript, das eingesetzt wird.

## BEISPIELE FÜR IMPERATIVE SPRACHEN - 1

- HTML, CSS mit JavaScript als imperativem Teil
- VRML, X3D, Collada und andere 3D-Sprachen
- XSLT und XSL -FO als Darstellungssprachen zu XML
- SQL, XQuery, SPARQL als Datenbank- oder XML-Abfragesprachen
- Funktionale Sprachen (Gofer, Haskell, Erlang, F#, Joy, Julia, Scala, Swift)

## BEISPIELE FÜR IMPERATIVE SPRACHEN - 2

- Logische Sprachen (z. B. Prolog) und funktional-logische Sprachen (z. B. Babel, Escher, Curry, Oz)
- Datenflusssprachen (z. B. Valoder Linda)
- Synchrone Programmiersprachen (z. B. Lustre)
- Aufbauwerkzeuge (z. B. make oder Ant)

## VORTEILE

- Die Programme sind kürzer, leichter zu verstehen
- Es besteht Zugriff auf den Quellcode zum Zeitpunkt der Ausführung – die Semantik des Programmes kann erkannt werden, Screenreader können auf Metadaten und Programmlogik zur Laufzeit zugreifen
- Die Programme sind leichter der Programmverifikation zugänglich. Es kann mit standardisierten Abarbeitungsapplikationen gearbeitet werden (Browser, Player, Parser, Runtimes, Prozessoren)

## NACHTEILE

- Verschiedene Abarbeitungsapplikationen können intern verschiedene Algorithmen und Darstellungsvorschriften verwenden – es ergeben sich evtl. unterschiedliche Resultate bei gleichem Code (Webseite auf Firefox und Chrom anders)
- Die Performanz unterschiedlicher Abarbeitungsapplikationen unterscheidet sich mitunter sehr, wobei Verbesserungen nur experimentell ermittelt werden können
- Existenz von Hacks, Workarounds, Tricks jenseits der standardisierten Syntax der Sprache

# XML

- XML (Extensible Markup Language) ist eine Struktursprache, die HTML ähnelt, aber keine vordefinierten Tags verwendet. Stattdessen werden eigenen Tags benutzt, die selbst definiert werden müssen.
- XML wurde für die Speicherung und den Transport von Daten entwickelt und ist weiterhin ein gängiger Standard.
- Dokument mit einer Baumstruktur, bei dem es ein Root-Element gibt. Die Kodierung von XML-Dokumenten erfolgt in einem Format, das sowohl für Menschen als auch für Maschinen lesbar ist.

## XML - SYNTAX

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <note>
3   <to>reciever</to>
4   <from>sender</from>
5   <heading>Heading</heading>
6   <body>Add Content Here</body>
7 </note>
```

# WAS HYPERTEXT/HYPERMEDIA BEDEUTET

- Systeme zur Verarbeitung von Dokumenten auf der Basis nicht-linear verknüpfter Daten werden Hypertext-Systeme genannt. Anders, als vom Namen her zu erwarten, behandeln diese Systeme nicht nur reine Textdokumente.
- Hypertext-Systeme gehen in der Regel von einer Darstellung der Dokumentstruktur als gerichteter Graph aus (Baumstruktur oder DOM-Struktur).
- Die Knoten verkörpern die grundlegenden Informationseinheiten, und die gerichteten Kanten stellen Bezüge zwischen den Informationseinheiten (Links, Verweise) her

## Tools und Entwicklungshilfen

# XHTML

```
1 <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN"
2 "http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">
3 <html xmlns="http://www.w3.org/1999/xhtml">
4 <head>
5 <title>Title of document</title>
6 </head>
7 <body>
8 Zeilenumbruch
9
10 <br></br>
11
12 <p>some content here...</p>
13
14 </body>
15 </html>
```

Tools und Entwicklungshilfen

---

**XML**

XHTML vs HTML

## AUFBAU VON HTML5

- HTML-Dokumente besitzen HEADER
- Struktur von HTML5

## CASCADING STYLE SHEETS

Cascading Style Sheets (CSS) ist eine Stylesheet-Sprache, mit der die Darstellung eines in HTML oder XML beschrieben wird.

CSS beschreibt, wie Elemente auf dem Bildschirm, auf Papier, in Sprache oder auf anderen Medien dargestellt werden sollen.

Beschriebe Styles kaskadieren von der obersten Ebene bis zum beschriebenen Element.

Tools und Entwicklungshilfen

---

## **HTML STYLING**

[\*\*HTML5 Styling\*\*](#)

## TRENNUNG VON CONTENT UND STYLING

- Konzept von HTML als Struktur und CSS als Styling
- Auslagerung des Styles in separate Datei (CSS-Datei)

## CSS - KASKADIERUNG - 1

```
<style>
  body {
    color: lightblue;
  }
  h1 {
    color: white;
  }
  .h1 {
    font-family: red;
  }
</style>
<h1 class="h1" style="color:black">My First CSS Example</h1>
```

## CSS - KASKADIERUNG - 2

```
<style>
  body {
    color: lightblue;
  }
  h1 {
    color: white !important;
  }
  .h1 {
    font-family: red;
  }
</style>
<h1 class="h1" style="color:black">My First CSS Example</h1>
```

## CSS-COMPILESPRACHEN

- Preprocessor-Sprachen zum Erzeugen von CSS über Scriptsprachen
- Compiler erzeugt aus Preprocessor die eigentliche Sprache
- [SASS \(SCSS\)](#)
- [Less](#)
- [Stylus](#)
- Auch Angular, Vue und ähnliche SPAs erzeugen eigenes CSS

## SASS ZU CSS

- SASS oder SCSS ist eine Preprocessor-Sprache, welche in CSS compiliert und miniert wird
- Erweiterung der CSS-Syntax durch Funktionen, Variablen und mehr
- Siehe: [SASS-lang.com](http://SASS-lang.com)

## DYNAMISCHES ÄNDERN VON INHALTEN - 1

- Statische Seiten mit HTML, CSS bieten (bis jetzt) wenig Interaktionsmöglichkeiten
- HTML und CSS sind zustandslos, keine Möglichkeit zur Speicherung von Zuständen
- Kein Zugriff auf BrowserAPI über HTML
- Was könnten Interaktionsmöglichkeiten sein?

## DYNAMISCHES ÄNDERN VON INHALTEN - 2

- Laden von (neuen) Inhalten
- Speichern von Daten (z.B. bei Logins)
- Auf User-Interaktionen reagieren (z.B. Mausklicks)
- Auf Endgeräten installieren (PWAs)
- DOM-Tree Manipulation ausführen

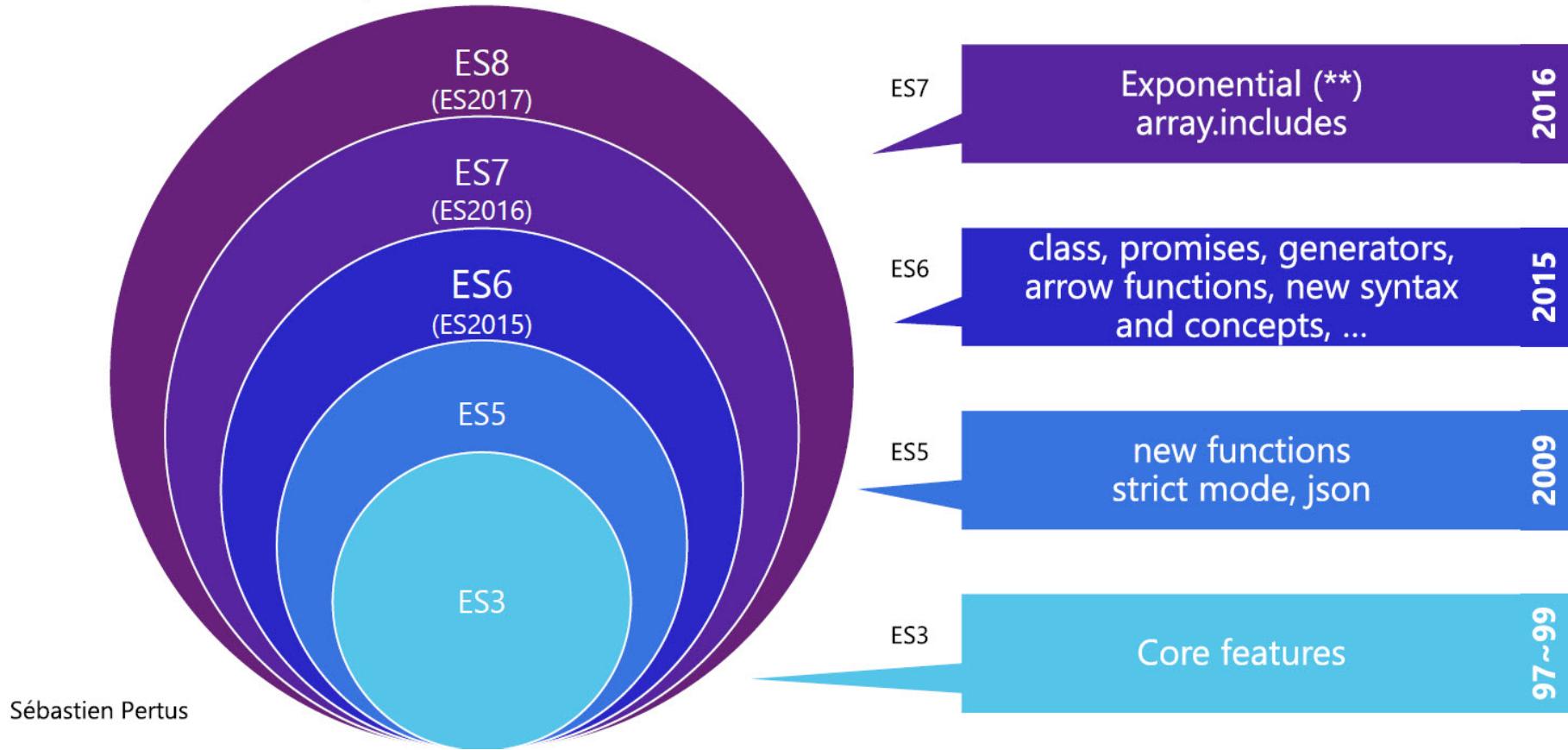
# ECMASCRIPT

- Ist ein Standard für Skriptsprachen, einschließlich JavaScript, JScript und ActionScript
- Standard zum Ausführen und Interpretieren von ECMA-Skriptsprachen (z.B. javascript)
- Möglichkeit der dynamischen Manipulation während der Laufzeit (Beim Aufrufen des HTML-Dokuments)
- JavaScript ist somit eine high-level oft "just-in-time" Interpreter- oder Compiler Sprache, die dem ECMAScript-Standard entspricht
- Mehr dazu in späteren Veranstaltungen

# ECMASCRIPT VERSIONEN - 1

Tools und Entwicklungshilfen

## EcmaScript



Source

## ECMASCRIPT VERSIONEN - 2

- ES2017 (Object-Entries, String operations)
- ES2018 (Async iterators, Promise API)
- ES2019 (FlatMap, Optional catch binding)
- ES2020 (BigInt, Dynamic library import, For-in order)
- ES2022 (Logical assignment operator, Promise Any)
- ES2023 (Top-level async await, Private instance fields, methods, and accessors)
- ES2024 (Array Finding Methods, Array Sorting Algorithm)

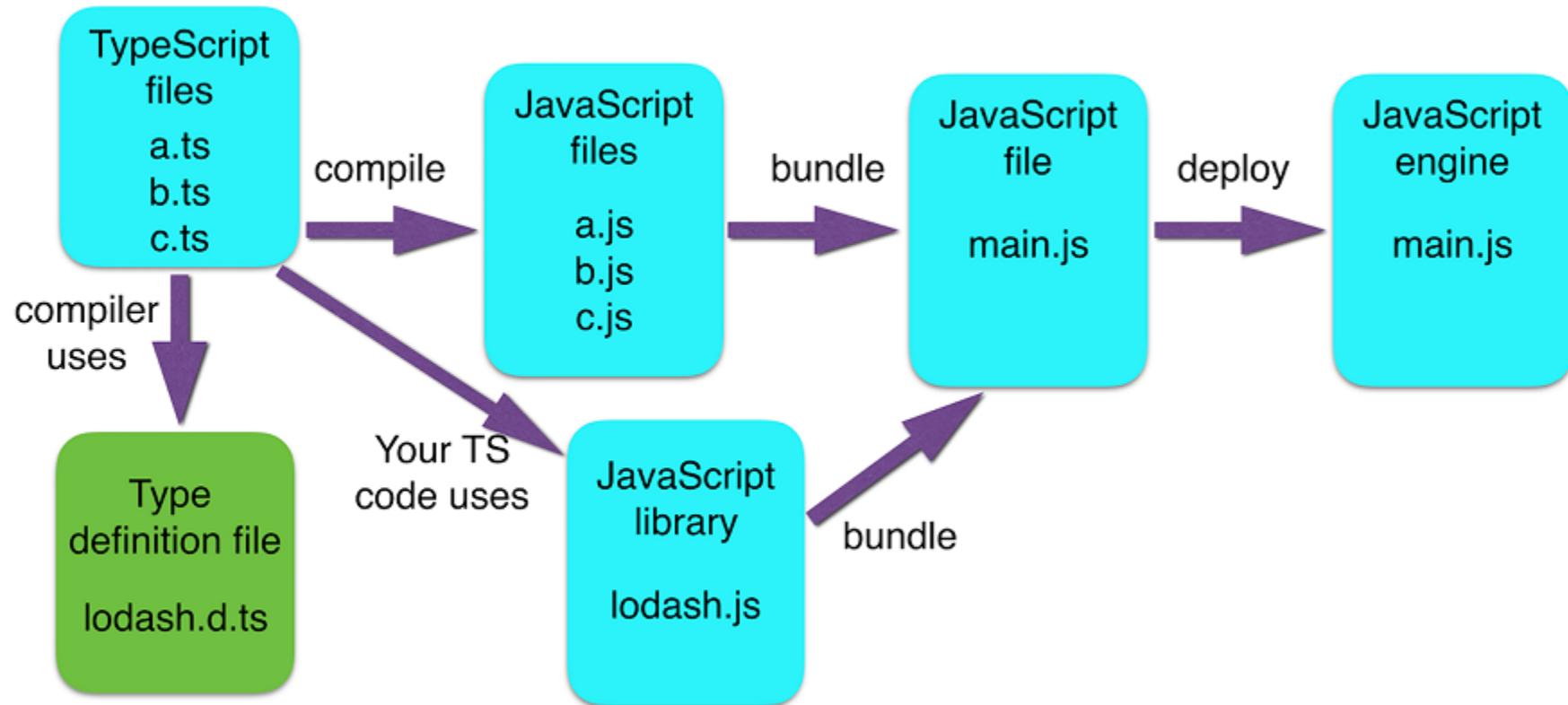
## SUPERSCRIPTS UND COMPILESPRACHEN

- Compilersprachen zur Erweiterung von Javascript auf Basis von ECMAScript-Standard
- [TypeScript](#)
- [Scala](#)
- [Coffee-Script](#)

## TYPESCRIPT

- TypeScript ist eine von ECMAScript abgeleitete Skriptsprache, die mehr als JavaScript ES6-13 kann, unter anderem ist sie typensicher
- Es kann so zu JavaScript kompiliert werden, dass es für jedes Betriebssystem und jeden Webbrowser lauffähigen Javascript-Code ergibt
- Man kann TypeScript-Code aus JavaScript heraus aufrufen, man kann JavaScript-Bibliotheken in TypeScript-Code einbinden

# ECMASCRIPT VERSIONEN - 1



Source

## Tools und Entwicklungshilfen

## IDE'S

- Texteditoren, Notepad++, Sublime Text, oder anderen Texteditoren
- Online: [Codepen](#)
- [Pulsar Edit \(Opensource\)](#)
- [Eclipse \(Eclipse Foundation\)](#)
- [Web-/PHP-Storm \(Jetbrains\)](#)
- [Adobe Dreamweaver \(Adobe\)](#)
- [Visual Studio code \(Microsoft\)](#)
- [Cloud 9 \(Amazon\)](#)

## Tools und Entwicklungshilfen

---

### BROWSERCONSOLE

Jeder Browser besitzt eine Browserconosole mit nützlichen Tools, welche meist mit F12 geöffnet werden kann.

Tools und Entwicklungshilfen

---

## WEBTOOLS

- Google Search console
- Lighthouse
- PageSpeed-Tools
- Firefox Webdeveloper
- Firefox Devoper tools

Tools und Entwicklungshilfen

---

## SEHR HILFREICHE SEITEN

- [W3schools](#)
- [Stack-Overflow <3](#)
- [Mozilla Dev](#)
- [Google Dev](#)
- [Web Archive](#)

Tools und Entwicklungshilfen

---

## GUIDES UND TUTORIALS

- [Google Learn \(Web.dev\)](#)
- [Mozilla Web Docs](#)
- [W3C-Schools HTML](#)
- [W3C-Schools CSS](#)
- [W3C-Schools JS](#)

## LITERATUR

- R. Steinmetz, Multimedia-Technologie. Grundlagen, Komponenten und Systeme., Springer-Verlag, 2000
- H. Erlenkötter, XML – Extensible Markup Language von Anfang an, Rowohlt Taschenbuch Verlag, Reinbek bei Hamburg, 2001/2003
- G. Born, Jetzt lerne ich XML, Verlag Markt+ Technik, München, 2001
- Juan Diego Gauchat, „HTML5 for Masterminds“, 3rd edition, Eigenverlag, März 2017
- Ranjan, Alok, Abhilasha Sinha, and Ranjit Battewad. JavaScript for modern web development: building a web application using HTML, CSS, and JavaScript. BPB Publications, 2020.
- Herron, David. Node.js Web Development: Server-side web development made easy with Node 14 using practical examples. Packt Publishing Ltd, 2020.
- Iskandar, Taufan Fadhilah, et al. "Comparison between client-side and server-side rendering in the web development." IOP Conference Series: Materials Science and Engineering. Vol. 801. No. 1. IOP Publishing, 2020.

## **ABSPANN**

Erstes Level geschafft, weitere Folgen!

Fragen und Feedback?