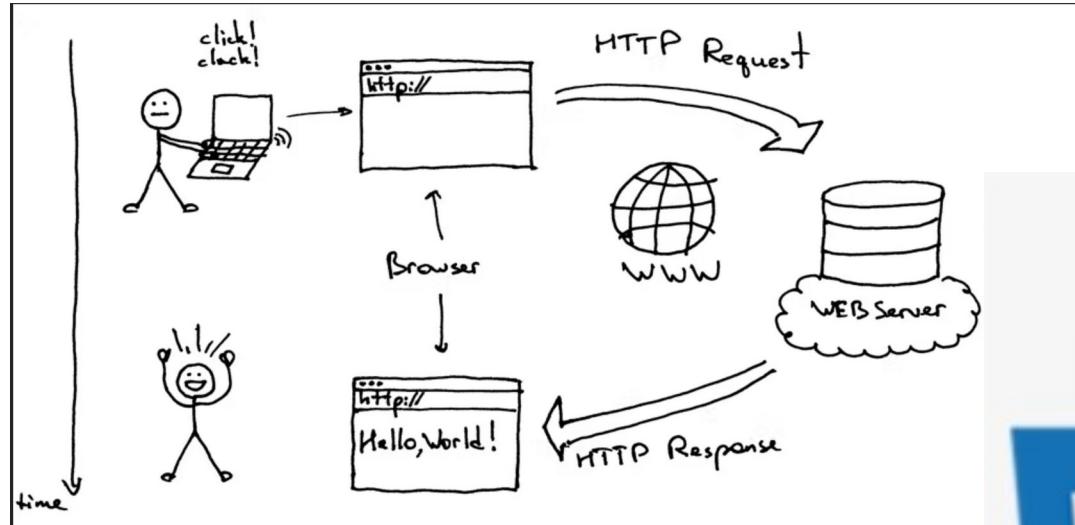


Web- and Mobile Computing (WMC)



Web- and Mobile Computing (WMC)

Erstes Jahr:

WS: html, CSS

SS: Javascript

Zweites Jahr:

- Single Page Application
- Typescript
- React
- Next.JS



Kapitel: Geschichte des Internets

Pre-Internet

1969 - ARPANET war das erste erfolgreiche Networking-Projekt. Universitäten und Forschungszentren, Regierungen, Firmen.

Ursprung: ARPANET wurde finanziert und initiiert vom US-Verteidigungsministerium.

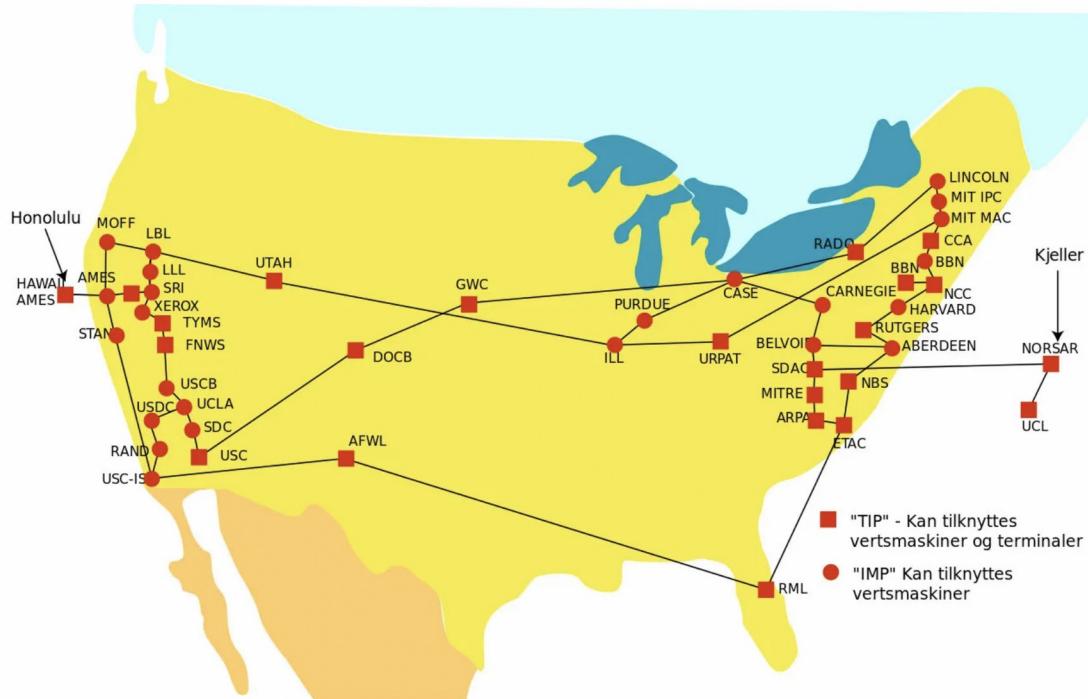
Erste Verbindung: Die erste ARPANET-Verbindung wurde am 29. Oktober 1969 zwischen University of California, Los Angeles (UCLA) und dem Stanford Research Institute (SRI) hergestellt.

Protokolle: Die anfänglichen Protokolle, die von ARPANET verwendet wurden, waren das Network Control Program (NCP). Anfang der 1980er Jahre wechselte ARPANET zu TCP/IP, welches die Grundlage für das heutige Internet legte.

Expansion: Im Laufe der Jahre erweiterte sich das Netzwerk und schloss verschiedene Universitäten und Forschungsinstitute in den USA ein.

Außenbetriebnahme: ARPANET wurde 1990 offiziell außer Betrieb genommen.

ARPANET, September 1974



US-Verteidigungsministerium und MIT

Dezentrales Netz

Verbindung über Telefonleitungen

Packet Switching

Vorläufer des heutigen Internets

Vinton Cerf (geb. 1943) & Robert Kahn (geb. 1938)



Cerf und Kahn entwickelten TCP/IP, das bis heute grundlegende Protokoll des Internets

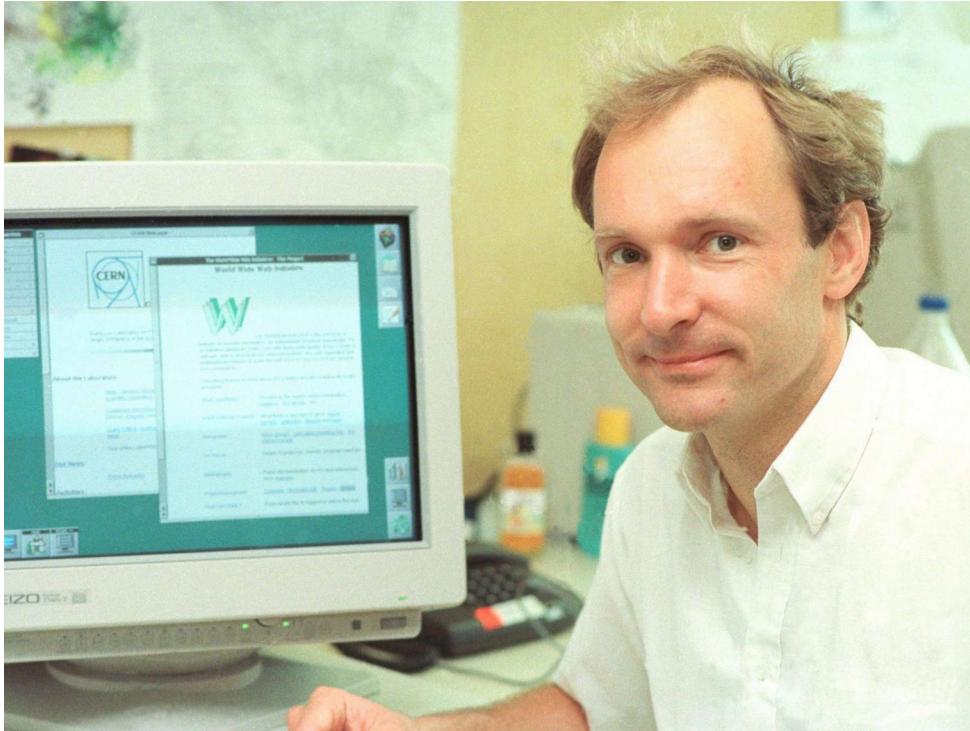
2019: Besuch von Vint Cerf ander Spengergasse (mit BP van der Bellen)

<https://www.spengergasse.at/?p=2135>

TCP/IP:

- Application-Layer
(http, ftp, smtp, pop, ...)
- Transport-Layer
(TCP, UDP, ...)
- Internet-Layer
(IPv4, IPv6, ...)
- Link Layer

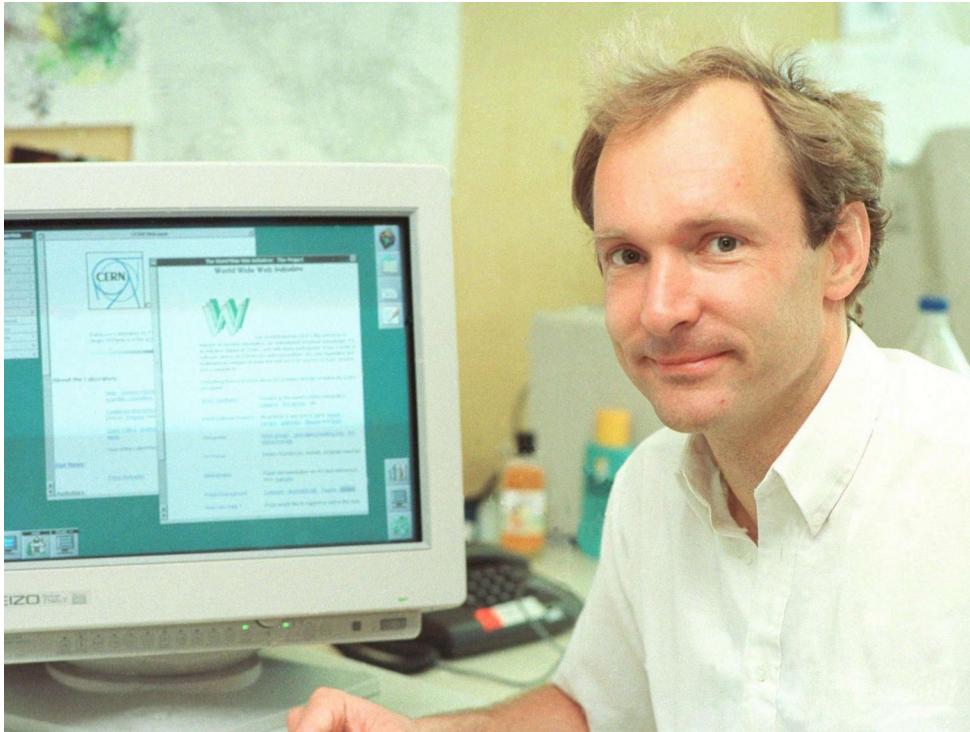
Sir Tim Berners Lee, CERN 1990



Ende der **1980er** Jahre erkannte Tim Berners-Lee bei **CERN** die Notwendigkeit, Forschungsdaten effizient zu teilen.

Er schlug das Konzept eines dezentralen Informationssystems mit **Hypertext** vor, welches später als **World Wide Web** bekannt wurde.

Sir Tim Berners Lee, CERN 1990



Tim Berners-Lee erfand die Grundbausteine des heutigen Web:

- **HTTP-Protocol**
- **HTML**
- **URL**

Er programmierte auch den ersten:

- **Browser**
- **HTTP-Server**

Modernes Internet

1990 - Tim Berners-Lee/CERN: **HTML, HTTP, URL, HTTP-Server, Browser**

1991 - **HTML Draft**

1995 - **HTML 2**: Erster offizieller HTML Standard

1994 - Tim Berners-Lee: **World Wide Web Consortium (W3C)**

1994 - Marc Mosaic: **Netscape Navigator**

1994 - Håkon Lie: **CSS**

1995 - Brendan Eich/Netscape: **Javascript**

1995 - Microsoft: **Internet Explorer** (1999: 99% Market Share)

1996 - Browser Wars: **Internet Explorer vs Netscape**

1997 - **HTML 4**

2000 - Netscape: **Mozilla Foundation**

2002 - Mozilla: **Firefox**

2003 - Apple: **Safari**

2008 - Google: **Chrome**

2010 - **Responsive Web Design** (Ethan Marcotte, A List Apart)

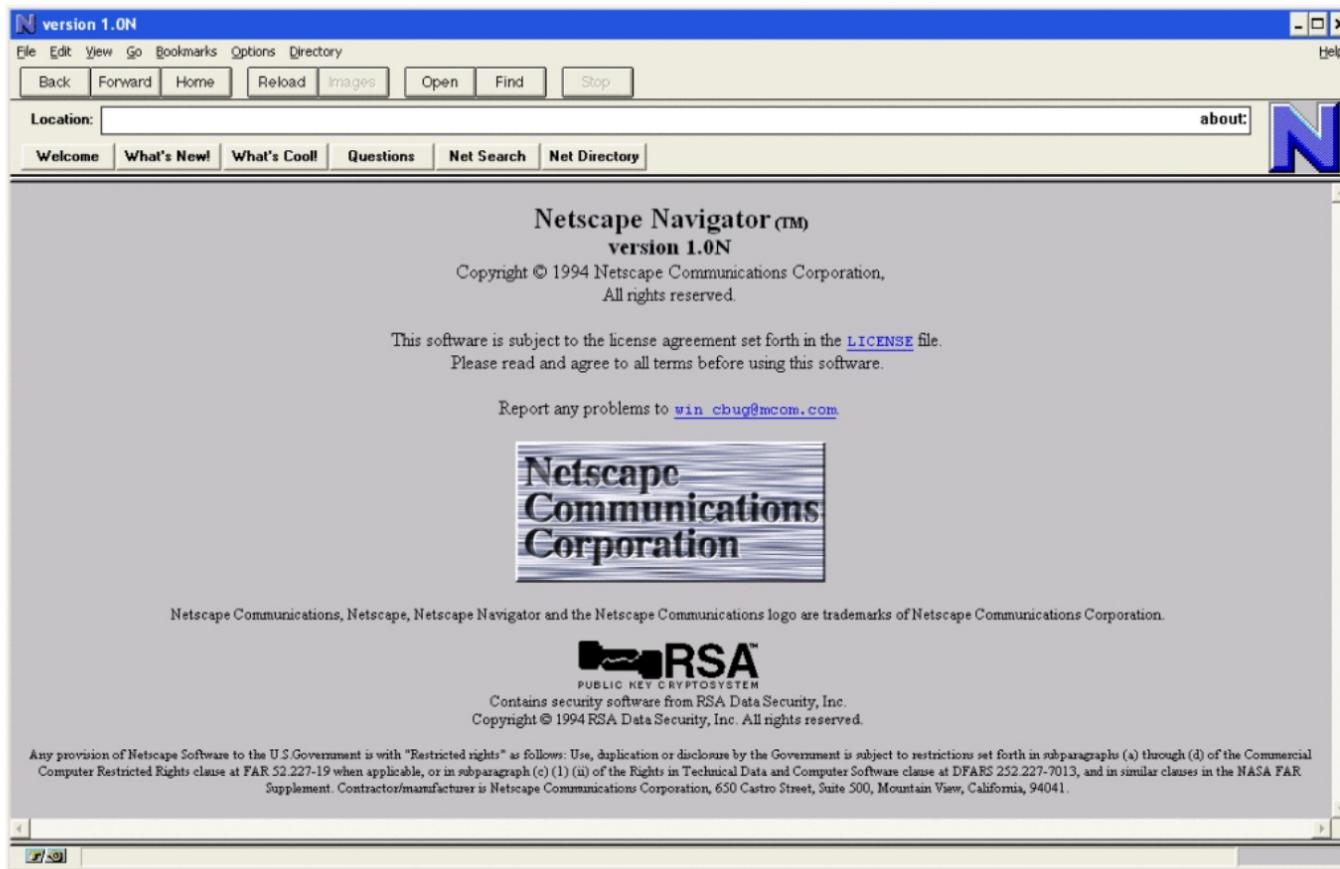
2014 - **HTML 5**

2015 - **ECMAScript 2015**

2017 - **PWA**: Progressive Web Apps

2023 - **ECMAScript 2023**

Netscape Navigator 1994



Screenshot of the 1.0N version of Netscape Navigator, released in 1994

Yahoo 1994

Netscape: Yahoo!

Back Forward Home Reload Images Open Print Find Stop

Location: <http://www.yahoo.com/>

What's New? What's Cool? Handbook Net Search Net Directory Software

[Reuters News Headlines](#) [Click here for breaking technology news first.](#) [Web Launch](#)

Search Options

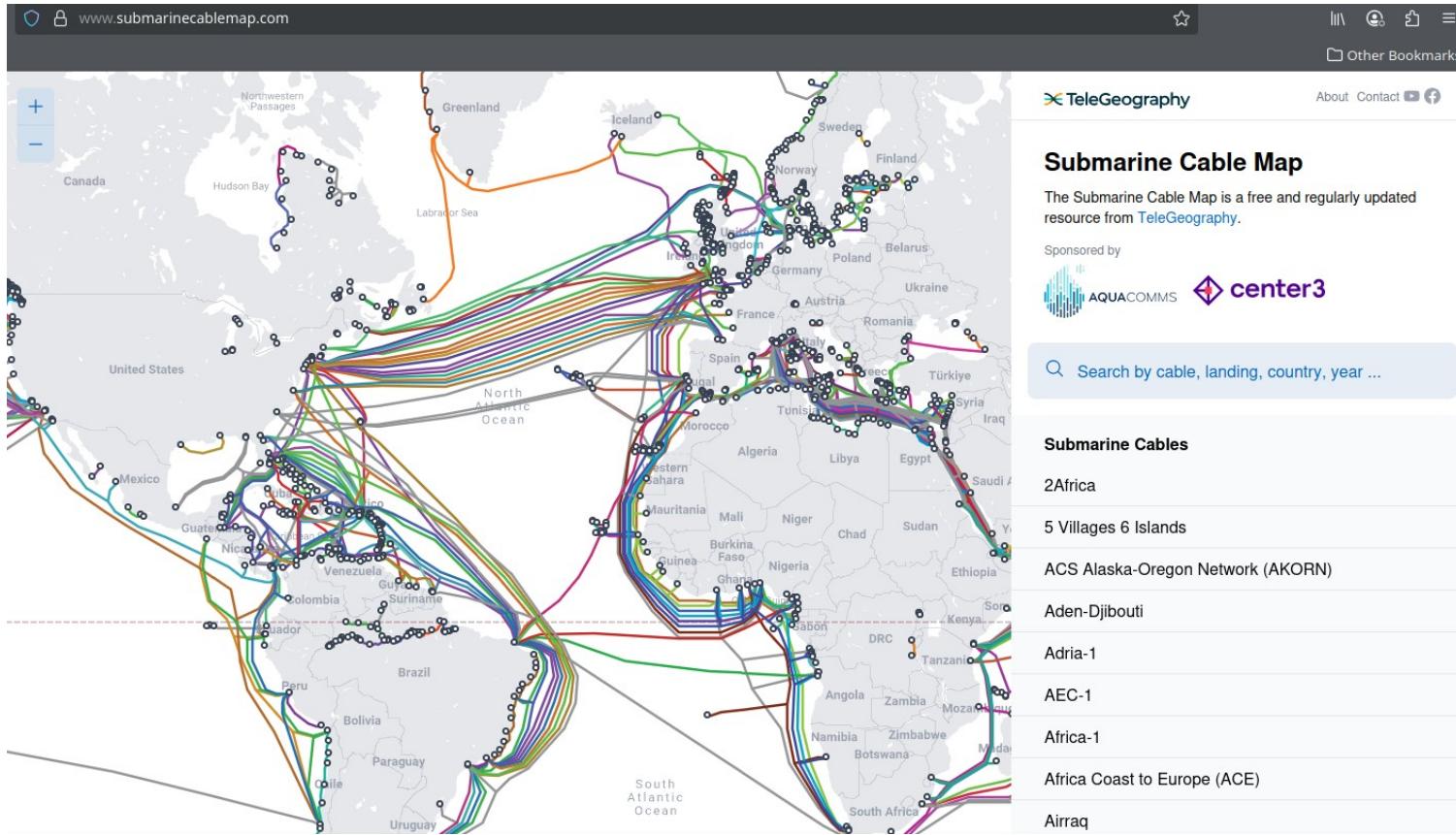
- **Arts**
[Humanities](#), [Photography](#), [Architecture](#), ...
- **Business and Economy [Xtra!]**
[Directory](#), [Investments](#), [Classifieds](#), ...
- **Computers and Internet [Xtra!]**
[Internet](#), [WWW](#), [Software](#), [Multimedia](#), ...
- **Education**
[Universities](#), [K-12](#), [Courses](#), ...
- **Entertainment [Xtra!]**
[TV](#), [Movies](#), [Music](#), [Magazines](#), ...
- **Government**
[Politics \[Xtra!\]](#), [Agencies](#), [Law](#), [Military](#), ...
- **Health**
[Medicine](#), [Drugs](#), [Diseases](#), [Fitness](#), ...
- **News [Xtra!]**
[World \[Xtra!\]](#), [Daily](#), [Current Events](#), ...
- **Recreation**
[Sports \[Xtra!\]](#), [Games](#), [Travel](#), [Autos](#), ...
- **Reference**
[Libraries](#), [Dictionaries](#), [Phone Numbers](#), ...
- **Regional**
[Countries](#), [Regions](#), [U.S. States](#), ...
- **Science**
[CS](#), [Biology](#), [Astronomy](#), [Engineering](#), ...
- **Social Science**
[Anthropology](#), [Sociology](#), [Economics](#), ...
- **Society and Culture**
[People](#), [Environment](#), [Religion](#), ...

[Text-Only Yahoo](#) ~ [Contributors](#)

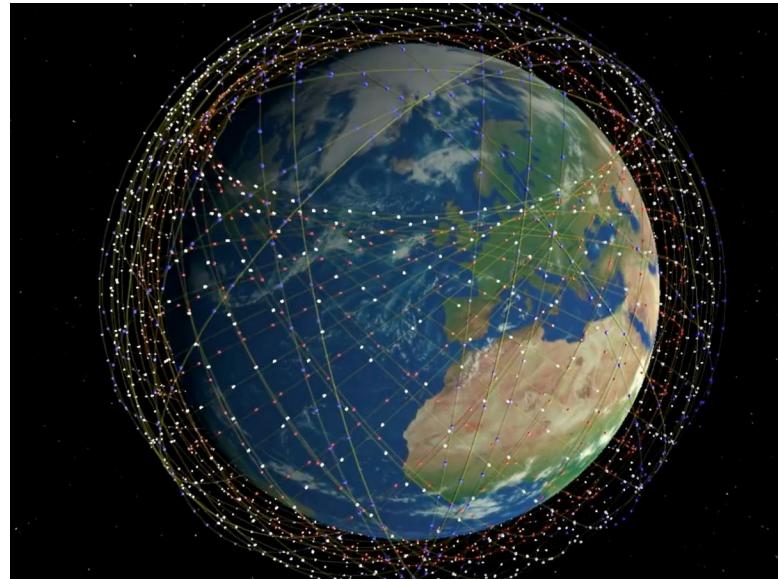
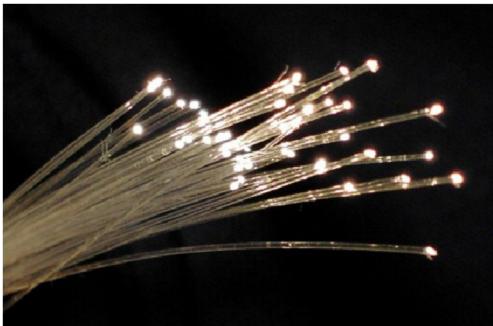
Document Done.



Modernes Internet



Modernes Internet

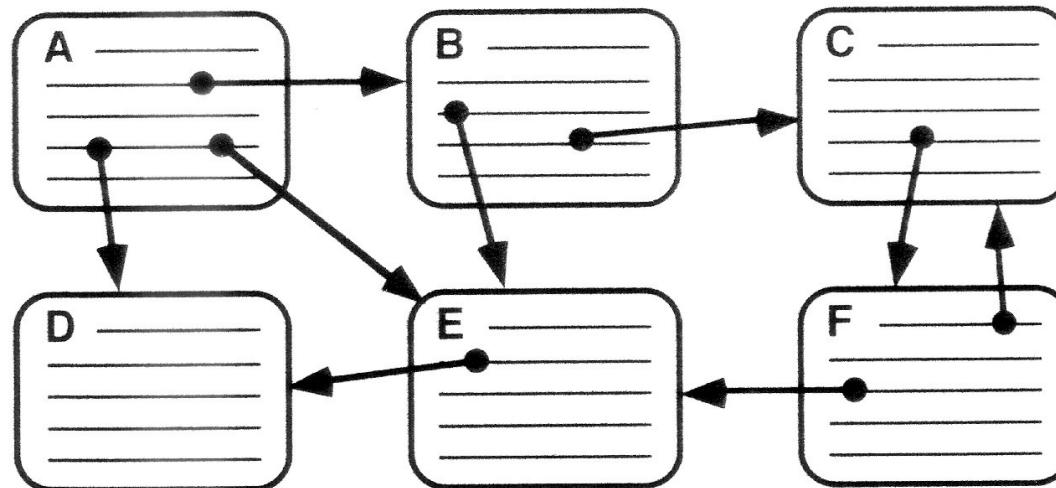


Kapitel: Grundlagen HTML

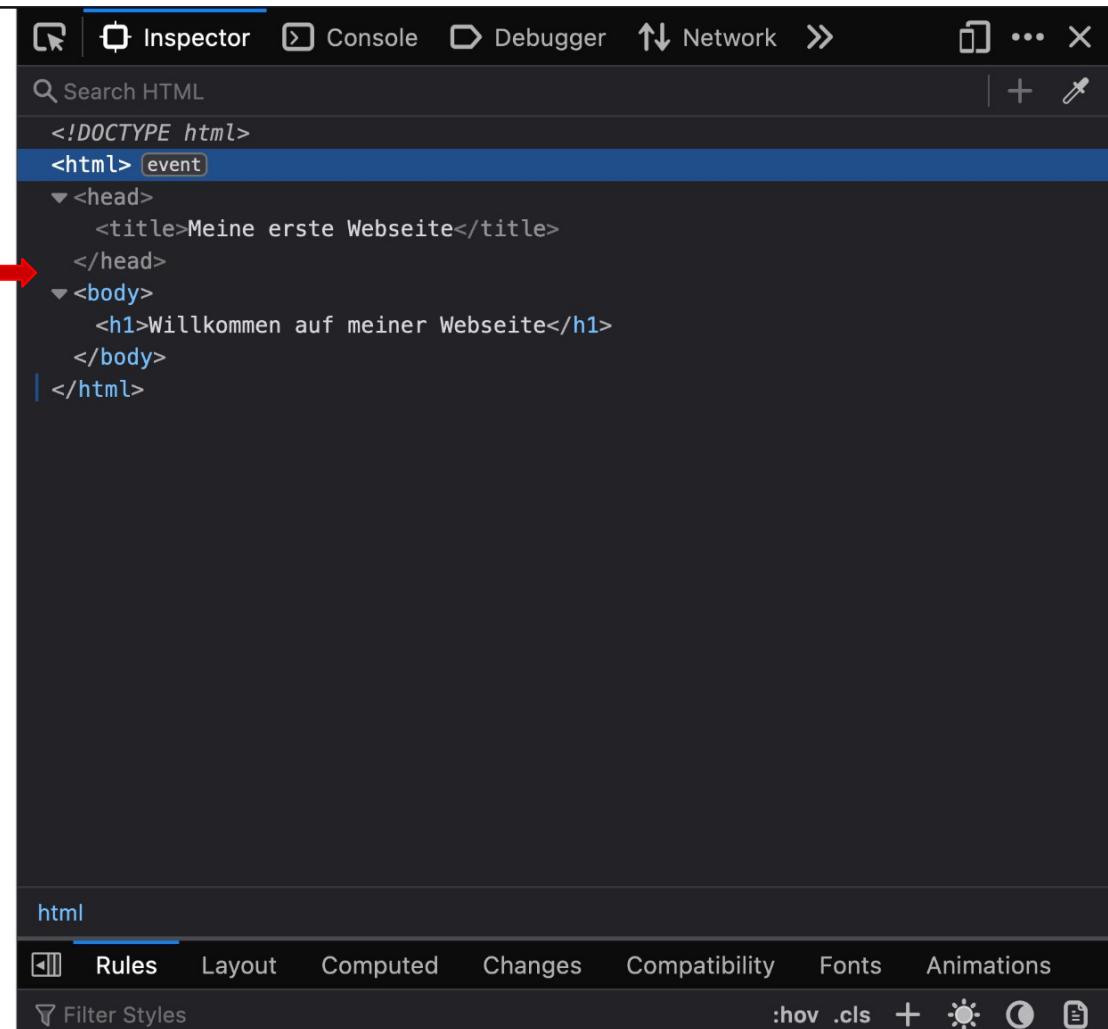
HTML - Hypertext Markup Language

Hypertext Markup Language (HTML) ist eine textbasierte Auszeichnungssprache
zur Strukturierung elektronischer Dokumente wie Text mit Hyperlinks (...)

- Source: https://de.wikipedia.org/wiki/Hypertext_Markup_Language



Willkommen auf meiner Webseite



The screenshot shows the developer tools of a web browser with the "Inspector" tab selected. A red arrow points from the left towards the "event" node in the HTML tree.

```
<!DOCTYPE html>
<html> event
  <head>
    <title>Meine erste Webseite</title>
  </head>
  <body>
    <h1>Willkommen auf meiner Webseite</h1>
  </body>
</html>
```

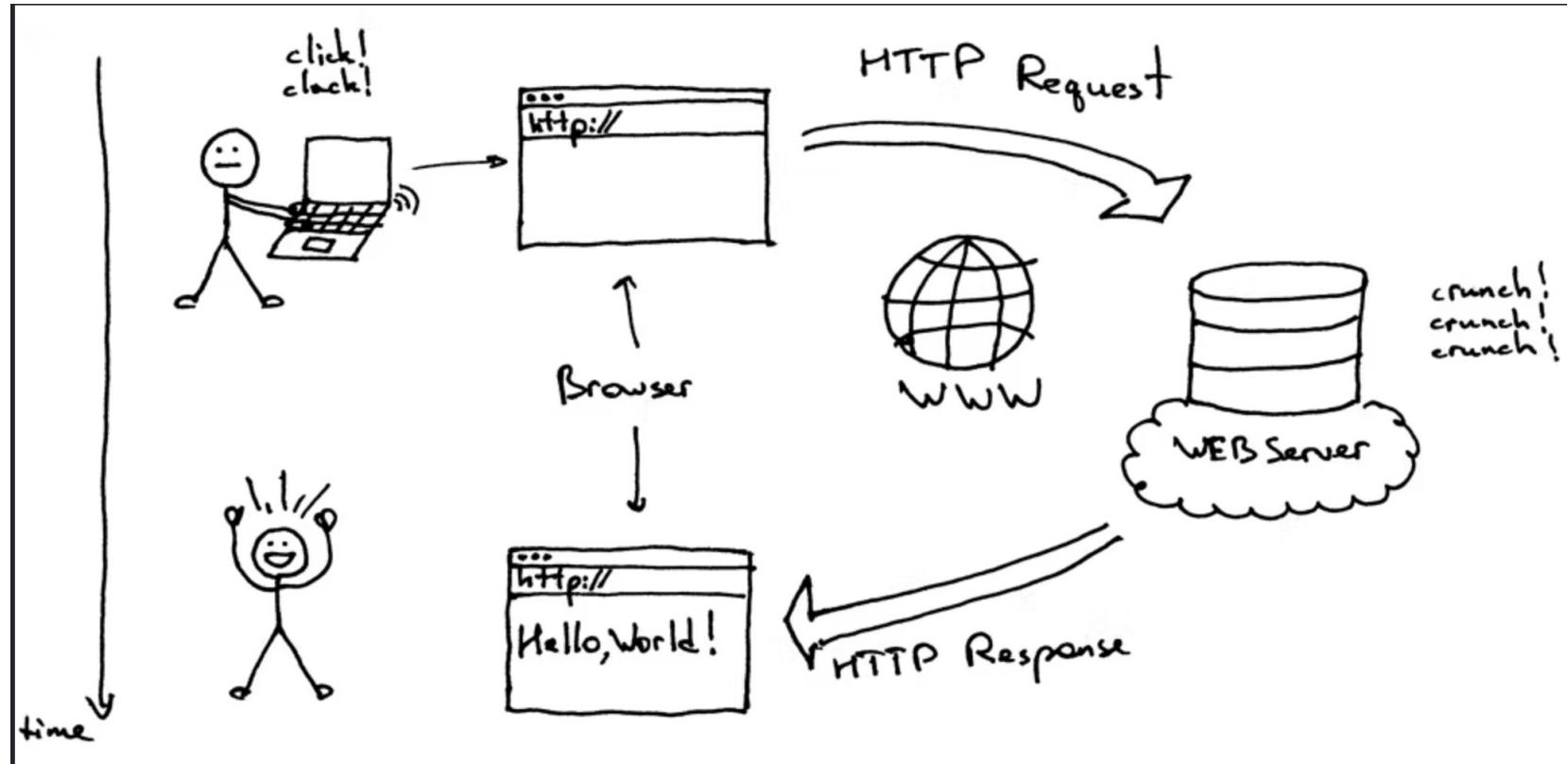
The bottom navigation bar shows the "html" tab is active, along with other tabs: Rules, Layout, Computed, Changes, Compatibility, Fonts, and Animations. There is also a "Filter Styles" button and some icons for selecting pseudo-classes like ":hov" and ".cls".

HTTP - Hypertext Transfer Protocol

Hypertext Transfer Protocol (HTTP) (..) **Protokoll zur Übertragung** (..) von Webseiten (Hypertext-Dokumente) aus dem World Wide Web (WWW) in einen Webbrowser zu laden.

- Source: https://de.wikipedia.org/wiki/Hypertext_Transfer_Protocol

HTTP - Hypertext Transfer Protocol



URL - Uniform Resource Locator

`http://example.com/path/index.html?_cr=google#article3`

Protocol

Host

Path

Query String

Fragment

URL - Uniform Resource Locator

Eine URL ("Uniform Resource Locator") ist die Webadresse einer speziellen **Ressource** im Internet. **Ressourcen** wie: Webseiten, Mediadaten, Dokumente, etc.

<https://www.orf.at/stories/123/?query=beispiel>

Scheme / Protocol ([https](https://www.orf.at/stories/123/?query=beispiel))

Das Protokoll für die Kommunikation (hier: sicheres HTTP).

Host ([www.orf.at](https://www.orf.at/stories/123/?query=beispiel))

Der Server, von dem die Ressource abgerufen wird (hier: die Website von ORF).

Path ([/stories/3206630/](https://www.orf.at/stories/123/?query=beispiel))

Der spezifische Ort der Ressource auf dem Server.

Query-String ([?query=beispiel](https://www.orf.at/stories/123/?query=beispiel))

Enthält zusätzliche Parameter für die Anfrage, die dem Server übermittelt werden. In diesem Fall wird ein Suchbegriff ("beispiel") übergeben.

Exkurs - Host vs Hostname

Host

Im Kontext einer URL bezieht sich "Host" auf die Kombination aus der Domain und optional dem Subdomain (Hostname) und der Portnummer.

Der Host gibt an, mit welchem Server im Internet der Browser eine Verbindung herstellen soll, um eine Webseite abzurufen.

Beispiel:

<https://www.orf.at/stories/123>

<https://api.orf.at/stories/123>

Host:

www.orf.at:443 (Port ist optional)

api.orf.at:443

Hostname

Im Kontext einer URL ist der "Hostname" der Teil der URL, der den spezifischen Server identifiziert, mit dem der Browser eine Verbindung herstellen soll.

Er ist ein Teil des "Hosts", der auch die Domain beinhaltet.

Beispiel:

<https://www.orf.at/stories/123>

<https://api.orf.at/stories/123>

Hostname:

[www](#)

[api](#)

Exkurs - Fully Qualified Domain Name (FQDN)

FQDN

Der Fully Qualified Domain Name (FQDN) ist eine vollständige und spezifische Referenz zu einem einzelnen Host oder Server im Internet.

Er beinhaltet den Hostnamen, den Domainnamen und die Top-Level-Domain (TLD), die einen vollständigen Pfad durch die DNS-Hierarchie zu einem spezifischen Netzwerkhost bilden.

Beispiel:

<https://www.orf.at/stories/123>

Hostname: [www](https://www.orf.at/stories/123)

Domain: [orf](https://www.orf.at/stories/123)

TLD: [at](https://www.orf.at/stories/123)

FQDN: [www.orf.at](https://www.orf.at/stories/123)

Exkurs - Beispiel orf.at

<https://www.orf.at/>

Scheme: https

Host: www.orf.at

Hostname: www

Domain: orf

TLD: at

Path: /

The screenshot shows the Network tab of a browser developer tools interface, specifically the Network tab in Chrome DevTools. The URL in the address bar is <https://www.orf.at>. The main content area displays the homepage of ORF.at with several news thumbnails. A red arrow points from the bottom of the main content area to the bottom of the developer tools window, highlighting the detailed network request information.

Network Tab Headers:

Header	Value
Address	194.232.104.140:443
Status	200 OK
Version	HTTP/2
Transferred	27,55 kB (112,16 kB size)
Request Priority	Highest
DNS Resolution	System

Network Tab Requests (Table):

Status	Method	Domain	File	Initiator	Type	Transferred	Size
200	GET	orf.at	/	document	html	27,55 kB	112,16 kB
200	GET	orf.at	ORFUniversalVF_W_Wg font	w...	font	107,05 kB	106,49 kB
200	GET	orf.at	ORFUniversalVF_Italic_ font	w...	font	127,71 kB	127,16 kB
200	GET	orf.at	ORFUniversalSCVF_W_ font	w...	font	31,10 kB	30,54 kB

Network Tab Summary: 64 requests | 4,79 MB / 2,46 MB transferred | Finish: 1,84 s | DOMContentLoaded: 593 ms | load: 1,63 s

Network Tab Details:

- Headers
- Cookies
- Request
- Response
- Timings
- Security

Network Tab Filter Headers:

Block Resend

GET

Scheme: https
Host: www.orf.at
Filename: /

Address: 194.232.104.140:443

Status: 200 OK
Version: HTTP/2
Transferred: 27,55 kB (112,16 kB size)
Request Priority: Highest
DNS Resolution: System

HTML for structure

CSS for design and layout

Javascript for interaction



What is the latest ?

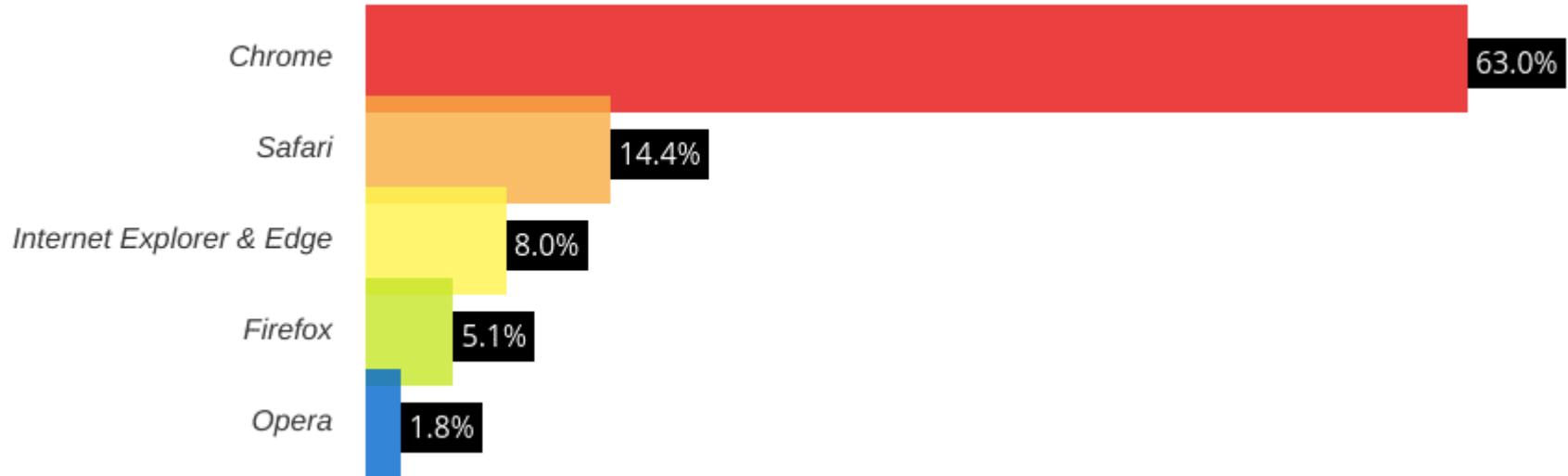
HTML 5.2

CSS 3

Javascript ES2024



Browser Market Share 2023



Ionic Framework (Native Apps with JavaScript)



Search



Docs

Components

Community

Enterprise



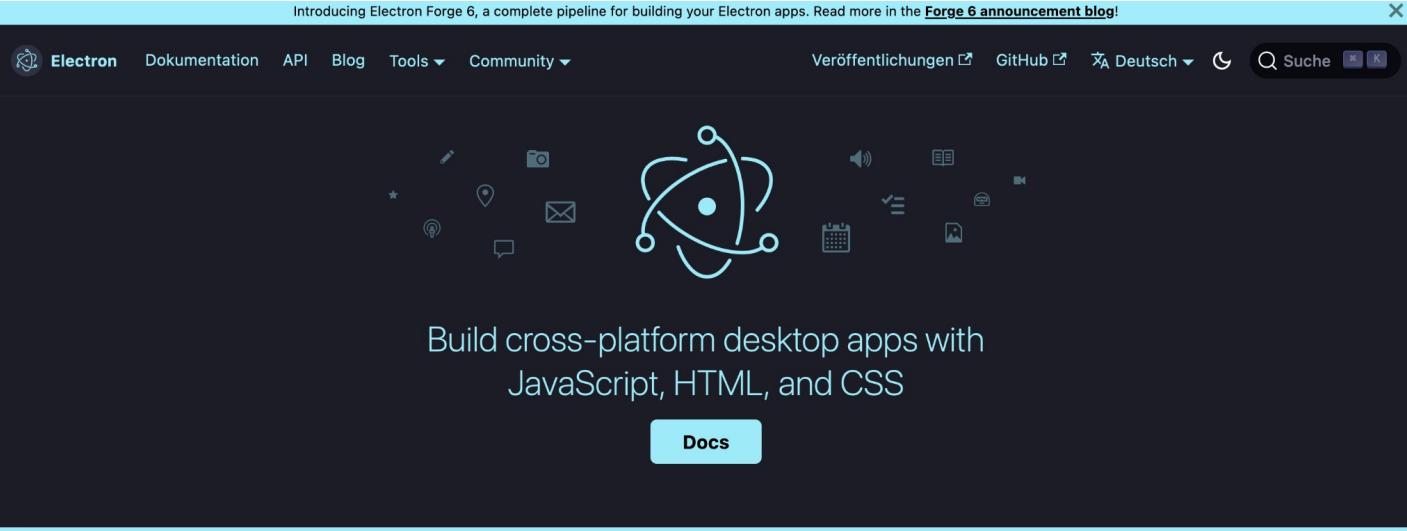
The mobile SDK for the Web.

An open source mobile UI toolkit for building modern, high quality cross-platform mobile apps from a single code base in [Angular](#).

Get started →

\$ npm i -g @ionic/cli

Elektron Framework (Desktop Apps with JavaScript)



The screenshot shows the official Electron website. At the top, there's a blue header bar with the text "Introducing Electron Forge 6, a complete pipeline for building your Electron apps. Read more in the [Forge 6 announcement blog!](#)". Below the header is a dark navigation bar with links for "Electron", "Dokumentation", "API", "Blog", "Tools ▾", "Community ▾", "Veröffentlichungen ▾", "GitHub ▾", "Deutsch ▾", a search bar, and user profile icons.

The main content area features a large central icon of a planet with orbiting rings, surrounded by various small application icons like a file, a mail, a calendar, etc. Below this icon, the text "Build cross-platform desktop apps with JavaScript, HTML, and CSS" is displayed. A prominent blue button labeled "Docs" is centered below the text.

At the bottom of the page, there's a dark footer section with three columns:

- Web Technologies**: Shows a small square icon and the text "Electron embeds Chromium and Node.js to enable web developers to create desktop applications."
- Cross Platform**: Shows a cube icon and the text "Compatible with macOS, Windows, and Linux, Electron apps run on three platforms across all supported"
- Open Source**: Shows a stylized "Y" icon and the text "Electron is an open source project maintained by the OpenJS Foundation and an active community of"

NodeJs (JavaScript on the Server)

The screenshot shows the official Node.js website. At the top, there's a dark blue header bar with the Node.js logo on the left, followed by navigation links: HOME, ABOUT, DOWNLOADS, DOCS, GET INVOLVED, SECURITY, CERTIFICATION, and NEWS. On the right side of the header are icons for a sun, a person, and a gear.

The main content area has a dark background. At the top, it says "Node.js® is an open-source, cross-platform JavaScript runtime environment." Below this, there's a section titled "Download for macOS" featuring two green buttons: "18.17.1 LTS" (labeled "Recommended For Most Users") and "20.6.1 Current" (labeled "Latest Features").

Below the download buttons, there are links for "Other Downloads | Changelog | API Docs" and "Other Downloads | Changelog | API Docs". A note below the buttons says, "For information about supported releases, see the [release schedule](#)."

At the bottom of the page, there's a footer with copyright information: "Copyright OpenJS Foundation and Node.js contributors. All rights reserved. The OpenJS Foundation has registered trademarks and uses trademarks. For a list of trademarks of the OpenJS Foundation, please see our [Trademark Policy](#) and [Trademark List](#). Trademarks and logos not indicated on the list of OpenJS Foundation trademarks are trademarks™ or registered® trademarks of their respective holders. Use of them does not imply any affiliation with or endorsement by them." It also includes links to "The OpenJS Foundation | Terms of Use | Privacy Policy | Bylaws | Code of Conduct | Trademark Policy | Trademark List | Cookie Policy".

JavaScript

JavaScript Everywhere



Referenzen

MDN - <https://developer.mozilla.org/de/>

CAN I USE - <https://caniuse.com/>

W3Schools - <https://www.w3schools.com/>

Editoren (installed)

IntelliJ, Visual Studio Code, Any Text Editor

Editoren (online)

CodePen, JSFiddle, Stackblitz

CSS References

CSS Tricks - <https://css-tricks.com/>

CSS Learning Games

CSS Flexbox - <https://flexboxfroggy.com/#de>

CSS Grid - <https://cssgridgarden.com/#de>

<https://www.w3.org/>

Gründung

1994 von Tim Berners-Lee, dem Erfinder des World Wide Webs.

Organisation

Eine internationale Gemeinschaft, in der Mitgliedsorganisationen, ein Vollzeit Stab und die Öffentlichkeit gemeinsam daran arbeiten, Webstandards zu entwickeln.

Standards

Entwickelt eine Vielzahl von Standards für das World Wide Web, einschließlich HTML, CSS, XML, und viele andere.

Dokumentation

Die Spezifikationen durchlaufen verschiedene Stadien, bevor sie als Empfehlungen veröffentlicht werden.

WHATWG

Web Hypertext Application Technology Working Group

<https://whatwg.org/>

Gründung

2004 von den Mitarbeitern von Apple, Mozilla, und Opera, als Reaktion auf die Wahrnehmung, dass die W3C die Entwicklung von Web Technologien, die für die Entwicklung von Webanwendungen erforderlich sind, nicht ausreichend unterstützte.

Organisation

Eine lose organisierte, offene Gemeinschaft von Einzelpersonen und Organisationen, die sich auf die Verbesserung der Web Technologien konzentrieren.

Standards

Fokussiert sich vor allem auf die Entwicklung von HTML und verwandten Webtechnologien.

Dokumentation

Die Spezifikationen werden als "lebende Standards" betrachtet, die kontinuierlich aktualisiert werden, anstatt in festgelegten Versionen veröffentlicht zu werden.

HTML5 Spezifikation

Die **HTML5-Spezifikation** der WHATWG ist ein "Living Standard".

Dies bedeutet, dass sie ständig aktualisiert wird, um aktuelle Technologien und Best Practices widerzuspiegeln. Dies fördert eine moderne und effiziente Webentwicklung.

Hier einige wichtige Punkte der **HTML5-Spezifikation**:

- HTML-Tags
- DOM (Document Object Model)
- APIs (Schnittstellen)
- Security Policies

<https://www.w3.org/TR/2011/WD-html5-20110405/>

<https://html.spec.whatwg.org/multipage/>

HTML5 Tags

HTML-Tags repräsentieren die Bausteine von HTML-Dokumenten. Sie sagen dem Browser, wie er den Inhalt anzeigen soll.

Dokument-Metadaten

<html><head><body>, <title>, <link>, <meta>, <style>

Struktur und Sektionierung

<header>, <footer>, <main>, <nav>, <section>, <article>, <aside>

Gruppierungs-Elemente

<div>, <p>, <hr>, <pre>, <blockquote>, , ,

Text-Semantik-Elemente

<a>, , , <small>, <cite>, <abbr>, <time>, <code>, <address>

HTML5 Tags

Formulare und Formelemente

<form>, <input>, <textarea>, <button>, <select>, <option>, <label>, <fieldset>, <legend>

Eingebettete Inhalte-Elemente

, <video>, <audio>, <iframe>, <embed>, <object>

Tabellen-Elemente

<table>, <thead>, <tbody>, <tr>, <td>, <th>

Scripting-Elemente

<script>, <noscript>, <canvas>

HTML 5 “Hello World”

```
<!DOCTYPE html>  
<html>
```



*Definiert den Dokumenttyp
(in diesem Fall HTML5).*

```
<head>  
  <title>Hello World</title>  
</head>
```



<head>
*Hier stehen Informationen
über die Webseite.*
</head>

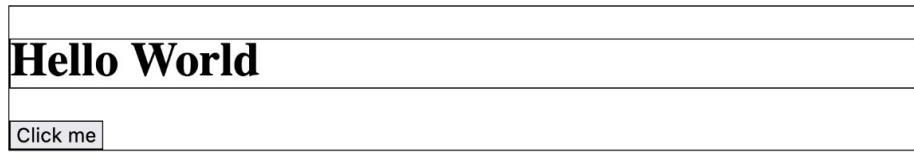
```
<body>  
  <h1>Hello World</h1>  
  <button>Click me</button>  
</body>
```



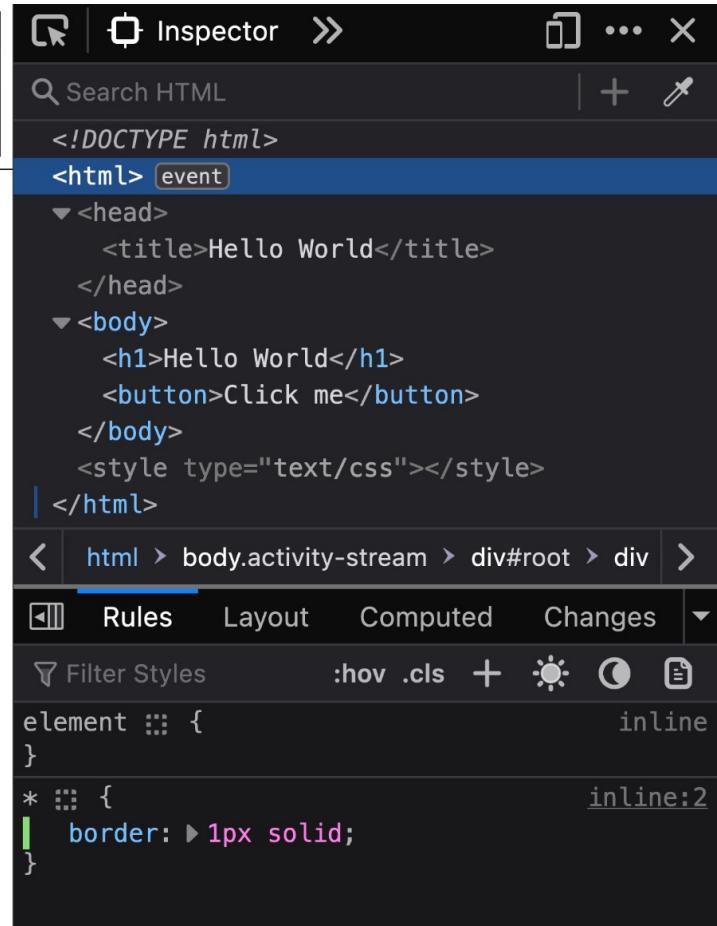
<body>
*Hier steht der sichtbare Teil
der Webseite.*
</body>

```
</html>
```

HTML erzeugt Boxes (“Kästen”)



A screenshot of a web browser window. The title bar says "Inspector". The main content area shows the text "Hello World" in a large font and a button labeled "Click me". A red arrow points from the text "Gibt allen HTML-Elementen eine sichtbare Umrandung." to the "Click me" button.



A screenshot of a browser's developer tools Inspector panel. The DOM tree shows the following structure:

```
<!DOCTYPE html>
<html> [event]
  <head>
    <title>Hello World</title>
  </head>
  <body>
    <h1>Hello World</h1>
    <button>Click me</button>
  </body>
  <style type="text/css"></style>
</html>
```

The "Rules" tab of the Inspector panel is selected. It shows the following CSS rule for the button element:

```
element ::{ border: 1px solid; }
```

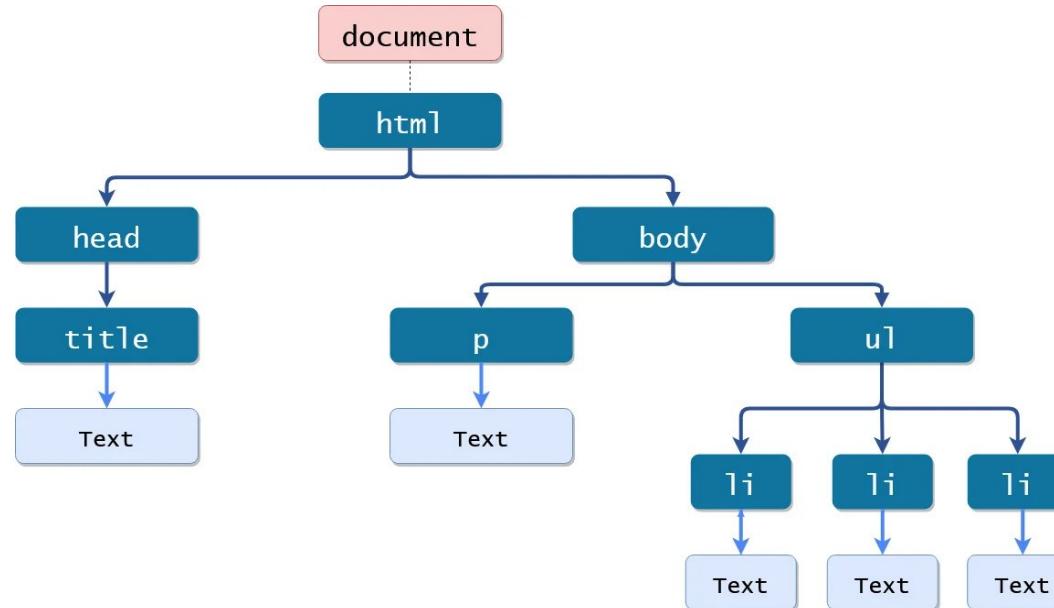
The "inline" style is highlighted in green.

Gibt allen HTML-Elementen
eine sichtbare Umrandung.

HTML5 DOM

DOM (Document Object Model) ist eine Schnittstelle, die Webseiten **als Baumstruktur** aus Objekten repräsentiert, wobei jedes Objekt einen Teil der Webseite darstellt.

Über **JavaScript** können diese Objekte manipuliert werden.



```
<!DOCTYPE html>
<html>
  <head>
    <title>Hello World</title>
  </head>
  <body>
    <h1>Hello World</h1>
    <button>Click me</button>

    <script>
      // Document Object Model (DOM) API
      const button = document.querySelector('button');

      button.addEventListener('click', function() {
        console.log('Button was clicked');
      });

    </script>
  </body>
</html>
```

Exkurs DOM API

Fügt den Button einen
`Click`-Handler hinzu.

HTML5 APIs

HTML5 APIs bieten Entwicklern Schnittstellen, mit denen sie **fortschrittliche Funktionen** und interaktive Inhalte in Webanwendungen integrieren können.

Beispielsweise APIs für Drag-and-Drop, Offline-Anwendungen, Lokalisierung und die Interaktion mit Multimedia-Elementen wie Video und Audio.

<https://developer.mozilla.org/en-US/docs/Web/API>

- Document Object Model (DOM) API
- Geolocation API
- Web Storage API
- Fetch API
- Canvas API
- Web Audio API
- WebRTC API
- Media Capture and Streams API
- WebSockets API
- Web Animations API
- File API
- Notification API
- Payment Request API
- IndexedDB API
- History API
- Drag and Drop API

```
<!DOCTYPE html>
<html>
  <head>
    <title>Hello Camera</title>
  </head>
  <body>
    <video autoplay playsinline></video>

    <script>
      const video = document.querySelector('video');

      // Media Capture and Streams API
      navigator.mediaDevices.getUserMedia({ video: true })
        .then((stream) => {
          video.srcObject = stream;
        });
    </script>
  </body>
</html>
```

Exkurs Media Capture API

Öffnet die Kamera und streamt den Inhalt in das Video-Element.

HTML5 Security Policies

Sicherheitsrichtlinien (Security Policies) dienen dazu, die Sicherheit beim Browsen von Webseiten zu erhöhen. Sie **schränken schädliche Aktionen ein**, die über Webseiten ausgeführt werden könnten, indem sie bestimmte Regeln und Beschränkungen für die Ausführung von Scripts und den Zugriff auf Ressourcen festlegen.

Same-Origin-Policy (SOP)

Eine Sicherheitsrichtlinie, die besagt, dass ein Webbrowser eine Webseite daran hindert, auf Daten zuzugreifen, die von einer anderen Webseite stammen.

https://developer.mozilla.org/en-US/docs/Web/Security/Same-origin_policy

Content Security Policy (CSP)

Eine zusätzliche Sicherheitsebene, die es Webentwicklern ermöglicht, bestimmte Sicherheitsrichtlinien durchzusetzen.

<https://developer.mozilla.org/en-US/docs/Web/HTTP/CSP>

Exkurs Same Origin Policy

The screenshot shows a Firefox browser window. The main content area displays the news website orf.at. The page features a large image of Putin and Kim Jong-un, several smaller news thumbnails, and a sidebar with sections for EU and Ukraine/Krieg. The developer tools are open, specifically the Console tab, which shows the following JavaScript code and its execution results:

```
>> fetch('https://www.orf.at')
<-> Promise { <state>: "pending" }
>> fetch('https://www.google.com')
<-> Promise { <state>: "pending" }

! Cross-Origin Request Blocked: The Same Origin Policy
disallows reading the remote resource at https://www.google.com/. (Reason: CORS header 'Access-Control-Allow-Origin' missing). Status code: 200. [Learn More]

! Ignoring error without stack
> Object { documentUrl: "https://orf.at/", errorKey: "f668f059cfa6f5ca61125833abb2e81a", sourceFile: "", stackTrace: "", errorMessage: "NetworkError when attempting to fetch resource.", lineNumber: "", project: "newssport", domain: "orf.at", browserName: "Firefox", browserVersion: "117.0", ... }
```

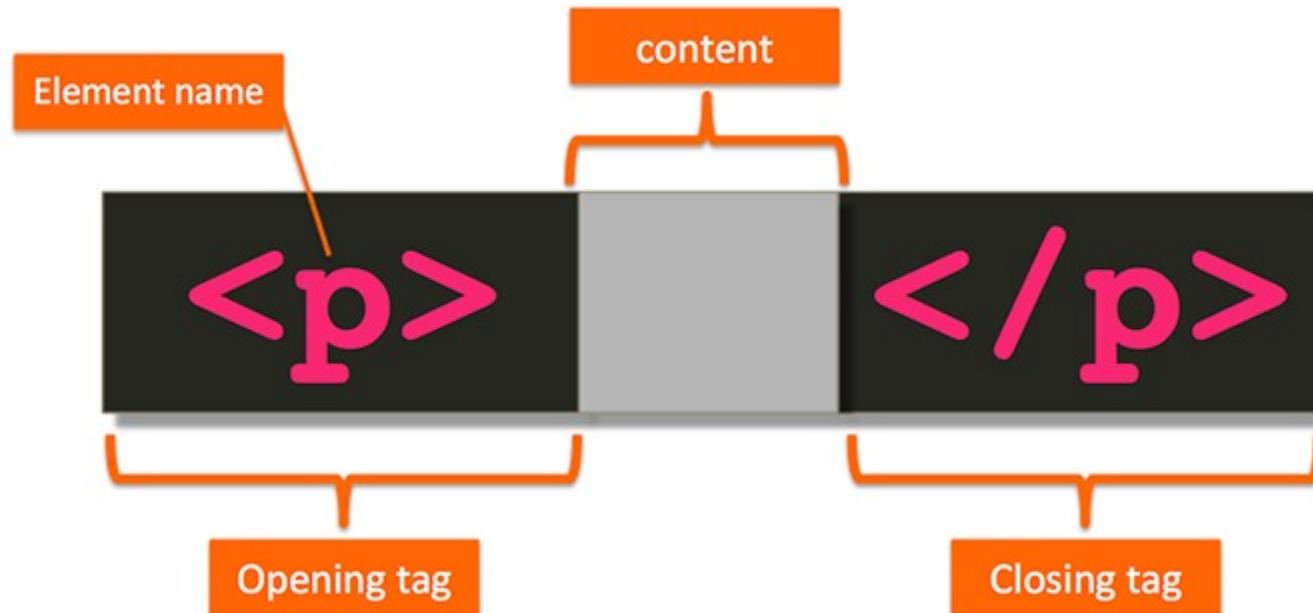
The error message in the console indicates that a cross-origin request was blocked due to the absence of a 'Access-Control-Allow-Origin' header in the response from Google.

Die **fetch-API** führt einen HTTP-Request in JavaScript aus.

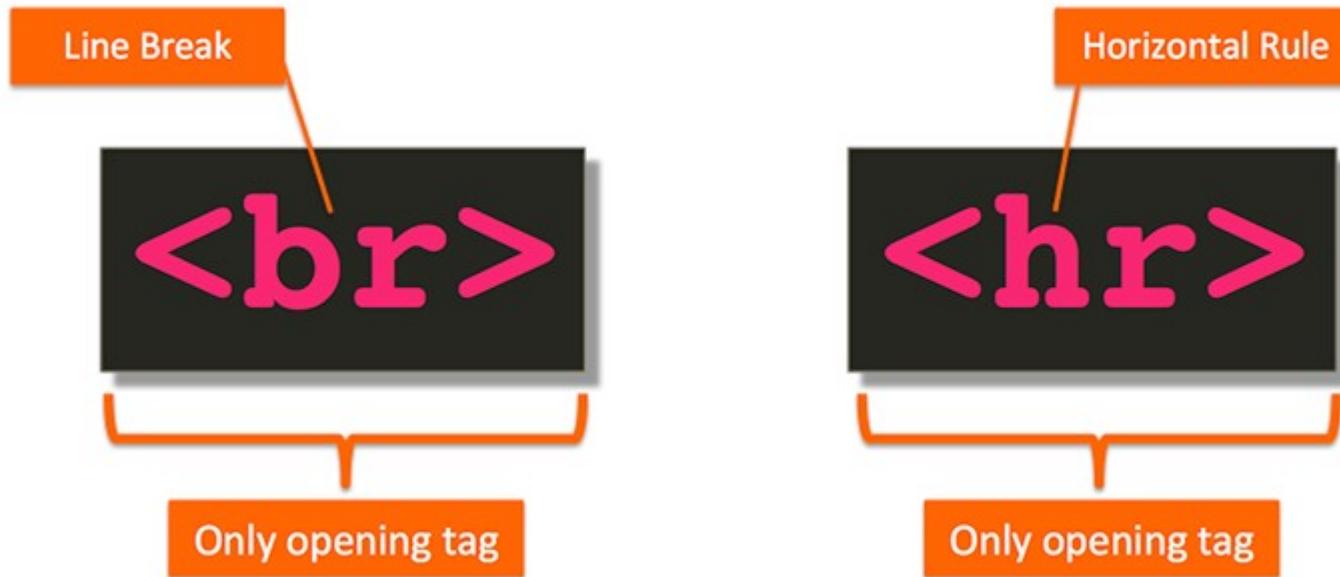
www.orf.at → www.orf.at
Same Origin
Erlaubt

www.orf.at → www.google.com
Cross Origin
Der Browser blockiert die Antwort von Google

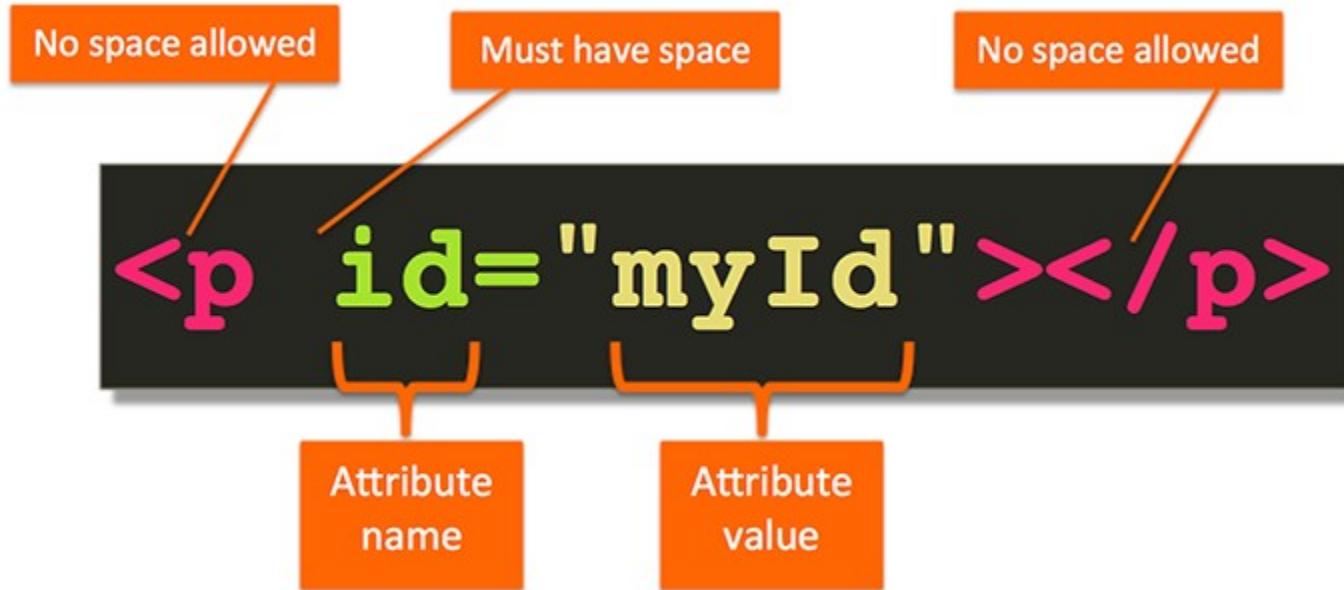
Anatomy of an HTML-Element



Anatomy of an HTML-Element



Anatomy of an HTML-Element



Anatomy of an HTML-Element

Double quotes are ok

```
<p id="myId"></p>
```

Single quotes are ok

```
<p id='myId'></p>
```

Anatomy of an HTML-Element

```
<h1>  
  <div>Hello World!</h1>  
</div>
```



```
<h1>  
  <div>Hello World!</div>  
</h1>
```



Exkurs: Link Navigation (1)

```
<!-- Anchor mit Text-Content -->
<a href="https://www.orf.at">
    ORF
</a>
```

```
<!-- Anchor mit HTML-Content -->
<a href="https://www.orf.at">
    <div>ORF</div>
</a>
```

Exkurs: Button Navigation (2)

```
<!-- Button mit JavaScript onclick-Handler -->
<button onclick="location.href='https://www.orf.at'" type="button"> ORF
</button>
```

```
<!-- Button ohne JavaScript -->
<form>
  <button formaction="https://www.orf.at">ORF</button>
</form>
```

```
<!-- Button ohne JavaScript; nicht valides HTML5 :-( -->
<a href="https://www.orf.at">
  <button>ORF</button>
</a>
```

W3C Validator

Das W3C (World Wide Web Consortium) bietet einen **Validator**, der es Entwicklern ermöglicht, HTML- (und CSS-) Markup auf Konformität mit den Standards zu überprüfen.

Wenn das Markup diese Validierung besteht, bedeutet das, dass es den **etablierten Webstandards** entspricht.

<https://validator.w3.org/>

The screenshot shows the W3C Markup Validation Service interface. At the top, there's a blue header bar with the W3C logo and the text "Markup Validation Service". Below it, a sub-header says "Check the markup (HTML, XHTML, ...) of Web documents". There are three tabs: "Validate by URI" (which is selected), "Validate by File Upload", and "Validate by Direct Input". Under the "Validate by URI" tab, there's a section for "Validate a document online:" with a text input field labeled "Address:" containing a placeholder URL. Below this is a link "► More Options". At the bottom center is a button labeled "Check".

W3C Validator

```
<!-- Button ohne JavaScript; nicht valides HTML5 :-( -->
<a href="https://www.orf.at">
  <button>ORF</button>
</a>
```

Error The element `button` must not appear as a descendant of the `a` element.

[From line 2, column 2; to line 2, column 9](#)

`orf.at"><button>ORF</b`

HTML5 Semantic Markup



HTML5 Semantic Markup

<header>

Kopfbereich einer Webseite oder eines Abschnitts.

Könnte den Seitentitel, das Logo, die Hauptnavigation oder andere einleitende Inhalte enthalten.

<nav>

Hauptnavigation einer Webseite.

Sollte nur für Hauptnavigation Bereiche verwendet werden und nicht für jede Gruppe von Links auf der Seite.

<main>

Hauptinhalt der Webseite.

Sollte einzigartig für das Dokument sein und nicht für Inhalte wiederholt werden, die über das gesamte Set von Dokumenten hinweg wiederholt werden
(z.B. Seitennavigation, Footer-Inhalte).

HTML5 Semantic Markup

<article>

Ein selbstständiger Abschnitt der Webseite.

Ist thematisch ist der von den umgebenden Inhalten getrennt und eigenständig verständlich (z.B. ein Blogpost, News einer Nachrichtenseite).

Kann <header>, <footer>, und andere semantische Elemente innerhalb enthalten.

<aside>

Gehören indirekt zum Hauptinhalt und werden oft als Sidebar präsentiert.

Zum Beispiel Werbung, Nutzerhinweise, oder Verweise auf verwandte Inhalte.

<footer>

Fußbereich einer Webseite oder eines Abschnitts.

Könnte Informationen wie den Autor, Copyright-Informationen, Links zu verwandten Dokumenten usw. enthalten.

HTML5 Semantic “Hello World”

```
<header>
  <h1>Rezept</h1>
</header>

<nav>
<ul>
  <li><a href="#">Home</a></li>
  <li><a href="#">Über mich</a></li>
</ul>
</nav>

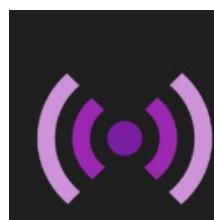
<article>
  <h2>Kochrezept</h2>
</article>

<footer>
  <p>Copyright © 2023 Chefkoch</p>
</footer>
```

Toolset für den Unterricht



Visual Studio Code



Live Server

Ritwick Dey | ⬤ 67,796,441 | ★★★★★(509)

Launch a development local Server with live reload feature for sta...

[Disable](#)

[Uninstall](#)

Auto Update



W3C Offline HTML Validator

FGMNTS | ⬤ 1,832 | ★★★★★(2)

A VSCode extension for offline HTML validation using W3C standa...

[Installing](#)

Auto Update

Links und Quellen

- Referenz:
<https://developer.mozilla.org/en-US/docs/Web/HTML/Reference/Elements/html>
- HTML Validator:
<https://validator.w3.org/detailed.html>
- Tutorials: <https://www.w3schools.com/>
- W3C Konsortium: <https://www.w3.org/>
- Web Hypertext Application Technology Working Group
<https://whatwg.org/>