



# Einführung in Web Engineering

Teil 2: HTML und CSS

*Manfred Kaul*

# Wiederholung der ersten Woche

Mit welchen fachlichen Argumenten wurde das  
WWW-Proposal von TBL abgelehnt?

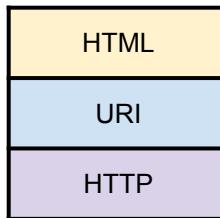
Was sind die fachlichen Argumente, warum das  
WWW dennoch ein Erfolg wurde?

Was wäre der Preis für die garantierter Verhinderung  
von “broken links”?

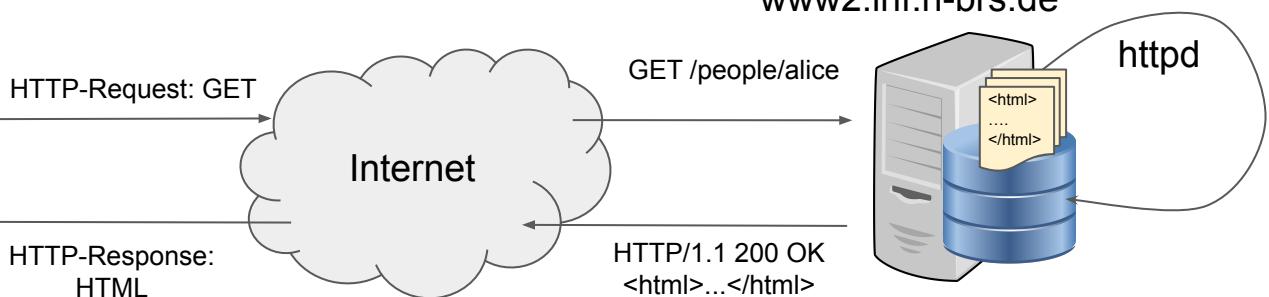


# Ihr persönlicher Server: www2.inf.h-brs.de

Persönliche Homepage



Browser als HTTP-Client



Klartext

```
GET /people/alice HTTP/1.1
Host: www.example.com
Accept: text/html, application/xhtml+xml
Accept-Language: en, de
```

The server could answer:

```
HTTP/1.1 200 OK
Content-Type: text/html
Content-Language: en
Content-Location: http://www.example.com/people.en.html
```

Quelle: <https://www.w3.org/TR/cooluris/>



# Technische Anleitung zur Erstellung einer eigenen Homepage in der Infrastruktur des Fachbereiches Informatik der Hochschule Bonn-Rhein-Sieg

1. Einloggen mit ssh auf www2.inf.h-brs.de.
2. Ordner 'public\_html/' in Ihrem eigenen Homeverzeichnis erstellen
  - o cd # Wechseln ins Heimverzeichnis
  - o chmod a+x . # oder chmod a+x \$HOME
  - o mkdir public\_html # Verzeichnis neu erstellen
  - o chmod a+rx public\_html
3. Homepage 'index.html' neu anlegen im Ordner 'public\_html/' mit dem Inhalt 'Hallo Welt'
  - o cd public\_html
  - o cat > index.html
  - o Hallo Welt
  - o STRG-D
4. Webseite <http://www2.inf.h-bonn-rhein-sieg.de/~astudi2s> im Browser aufrufen



# Semesterprojekt Web Engineering

## Aufgabenstellung

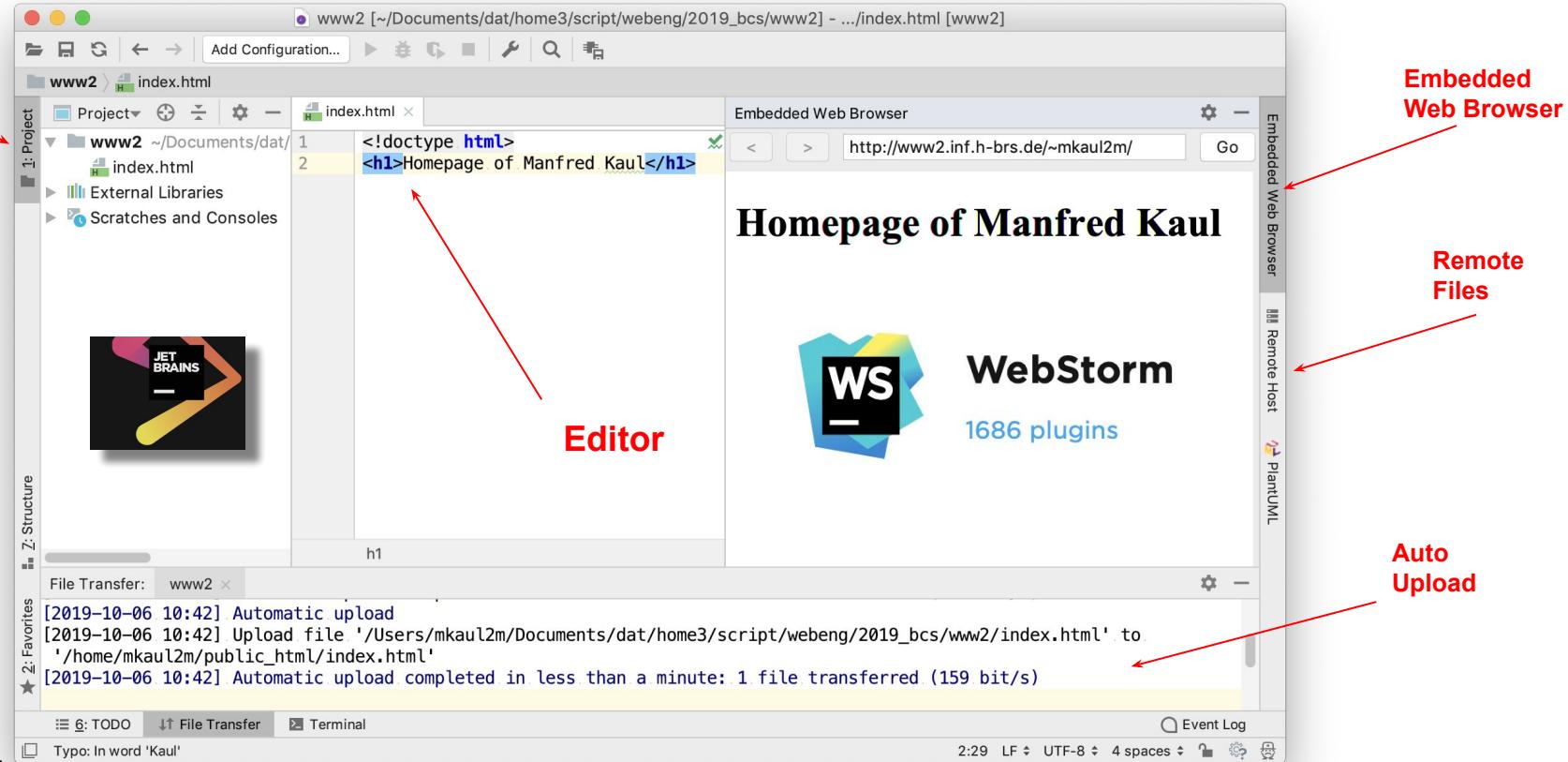
Die Aufgabe im Semesterprojekt ist die Erstellung und Pflege einer eigenen **Homepage** im Rahmen der **wöchentlichen Praktikumsaufgaben**. D.h. dass Sie wöchentlich etwas zu Ihrer Homepage hinzufügen oder ändern sollen nach dem aktuellen Kenntnisstand aus der Vorlesung. Die besprochenen Technologien sollen auf Ihrer Homepage auch tatsächlich zum Einsatz kommen. Machen Sie sich also darauf gefasst, dass die Homepage viele Unterseiten umfasst, auf denen die verschiedenen Themen behandelt werden.

Stellen Sie die Lösungen zu Ihren wöchentlichen Praktikumsaufgaben auf Ihre Homepage **nach Ablauf der Deadline**.

Sie werden in den Praktikumsaufgaben auch eigene Web-Komponenten erstellen, die Sie in Ihre Homepage einbauen sollen.



# Integrated Development Environment (IDE)





# Visual Studio Code

# von Microsoft

The screenshot shows the Visual Studio Code interface with a dark theme. The Explorer sidebar on the left lists 'OPEN EDITORS' (index.html) and a folder 'www2' containing '.idea' and 'index.html'. The main editor area displays the file 'index.html' with the following content:

```
<!doctype html>
<h1>Homepage of Manfred Kaul</h1>
```

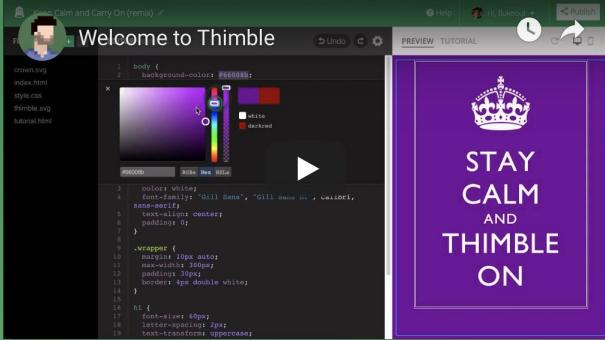
The status bar at the bottom shows: Ln 2, Col 34, Spaces: 4, UTF-8, LF, HTML, Prettier, Formatting: ✓, and icons for issues (0), warnings (0), and notifications.

# Thimble von Mozilla

## Schreiben Sie das Web Im Web

Thimble ist ein Online-Quelltext-Editor, mit dem Sie ganz einfach Ihre eigenen Webseiten erstellen und veröffentlichen können, wobei Sie gleichzeitig HTML, CSS & JavaScript lernen können.

Fangen Sie ein ganz neues Projekt an



The screenshot shows the Thimble interface. On the left is a file tree with files: crown.svg, index.html, style.css, and thimble.svg. The main area has a color palette at the top and a code editor below it. The code editor contains the following CSS:

```
body {  
    background-color: #5555EE;  
}  
  
.wrapper {  
    width: 100%; max-width: 300px;  
    padding: 10px;  
    border: 1px double white;  
}  
  
h1 {  
    font-size: 60px;  
    letter-spacing: 2px;  
    text-transform: uppercase;  
}
```

To the right is a preview window showing a purple page with a white crown icon and the text "STAY CALM AND THIMBLE ON".

## We're moving! Time to check out Glitch

Mozilla has made a strategic decision to sunset Thimble over the course of 2018-2019. We encourage you to check out [Glitch](#), a platform similar to Thimble. [Read our blog post](#) to learn more about this partnership.



If you use Thimble without an account, you may continue to do so until **16. Dezember 2019**.

If you have an account on Thimble, please sign in and migrate your projects over to Glitch by **16. Dezember 2019**.





**wordart**

a glorious tribute to powerpoint

Show View Source

assets

public/

- client.js
- style.css
- wordart.css

views/

- index.html**
- .env

README.md

package.json

server.js

```

16 <meta charset="utf-8">
17 <meta http-equiv="X-UA-Compatible" content="IE=edge">
18 <meta name="viewport" content="width=device-width, initial-scale=1">
19 <link rel="stylesheet" href="/style.css">
20 <link rel="stylesheet" href="/wordart.css">
21 </head>
22 <body>
23 <header>
24 <h1>
25     let's make some ✨ ART ✨
26 </h1>
27 </header>
28
29 <main>
30     pick your <span id="poison"><a href="#">poison</a></span>
31     <div id="picker">
32         <div class="wrapper"><div class="wordart rainbow"><span class="text">WordArt</span></div></div>
33         <div class="wrapper"><div class="wordart blues"><span class="text">WordArt</span></div></div>
34         <div class="wrapper"><div class="wordart superhero"><span class="text">WordArt</span></div></div>
35         <div class="wrapper"><div class="wordart purple"><span class="text">WordArt</span></div></div>
36         <div class="wrapper"><div class="wordart radial"><span class="text">WordArt</span></div></div>
37         <div class="wrapper"><div class="wordart horizon"><span class="text">WordArt</span></div></div>
38         <div class="wrapper"><div class="wordart tilt"><span class="text">WordArt</span></div></div>
39         <div class="wrapper"><div class="wordart slate"><span class="text">WordArt</span></div></div>
40         <div class="wrapper"><div class="wordart italic-outline"><span class="text">WordArt</span></div></div>
41
42
43

```

Remix to Edit

Edit your Project Here

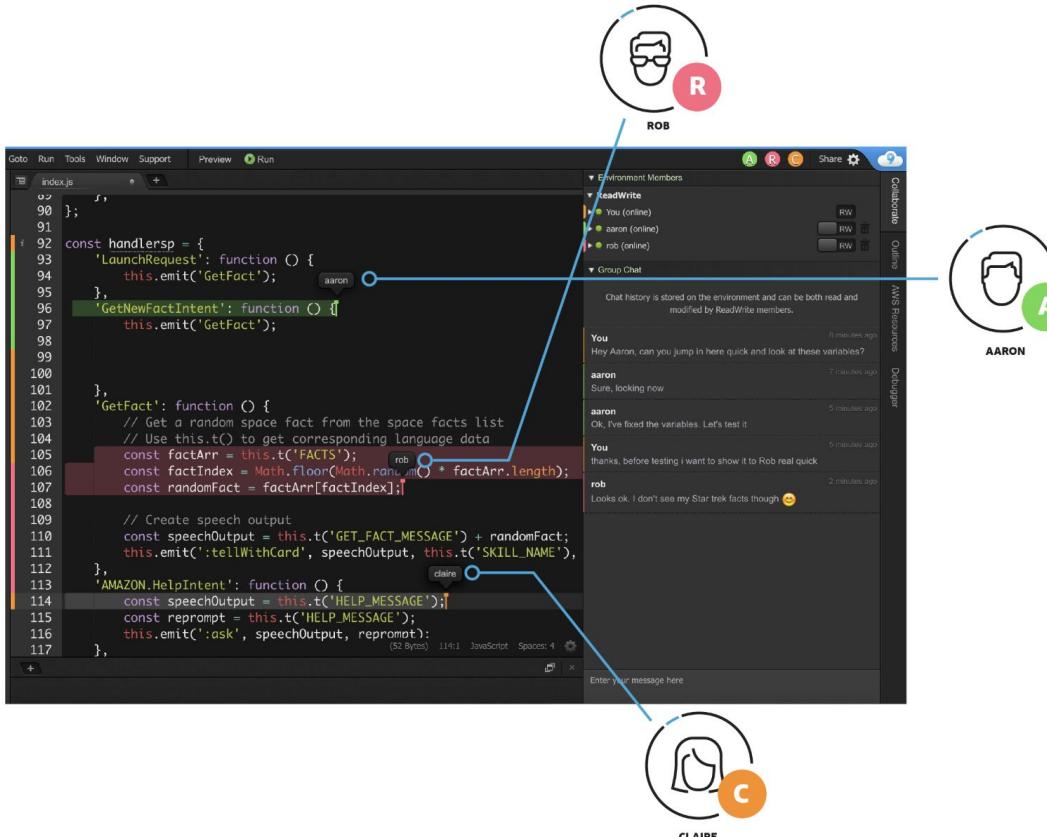
Share View App

wordart by

Report Abuse

Remix your own

# Amazon Cloud9: Das Web als IDE für das Web

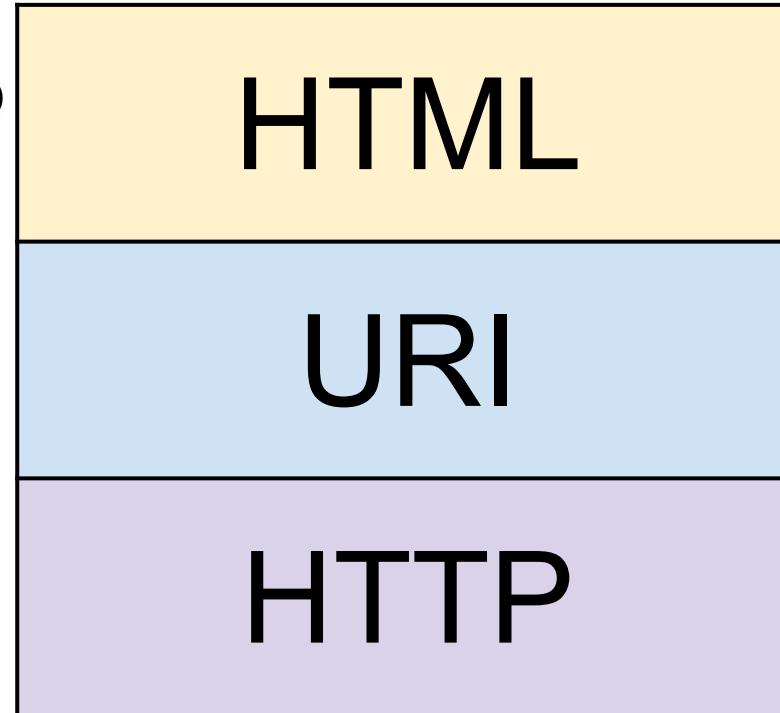
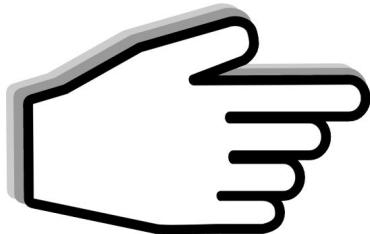


# Digital Makerspace: Das Web als Werkzeug UND Material

The screenshot shows a web browser window titled "Digital Makerspace" with the URL "kaul.inf.h-brs.de/dms/#dms=components". The page displays a grid of 24 components, each with a small icon, name, creator, description, and rating.

Component	Creator	Description	Rating
Analog Clock	Manfred Kaul	Analog Clock	★★★★★ (0)
Content	André Kless	Shows predefined content	★★★★★ (0)
Exercise	André Kless	Create a free text task	★★★★★ (0)
Fill in the Blanks	André Kless	Create a task with missing words in a text	★★★★★ (0)
Guess Picture	André Kless	Guess a picture before it becomes completely visible.	★★★★★ (0)
JSON Builder	André Kless		★★★★★ (0)
Listing	André Kless	Listing of items	★★★★★ (0)
Live Poll	André Kless	Create a spontaneous live survey	★★★★★ (0)
Mark Words	André Kless	Create a task where users highlight words	★★★★★ (0)
Paper Generator	mkaul2m	Paper Generator for surveys (opinion polls) with time measurements.	★★★★★ (0)
PDF Viewer	André Kless	Upload a PDF to offer it without download	★★★★★ (0)
Quick Decision	André Kless	Create a quiz where the user has to decide quickly	★★★★★ (0)
Quill Editor	Tea Kless	Quill Editor	★★★★★ (0)
Quiz	André Kless	Create single or multiple choice quizzes	★★★★★ (0)
Star Rating	André Kless	Create a five star rating	★★★★★ (0)
Submit	André Kless		★★★★★ (0)
Team Building	André Kless	Create a real time team building	★★★★★ (0)
Thumb Rating	André Kless	Create a thumb rating	★★★★★ (0)



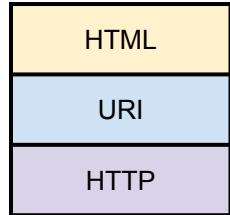


```
<!DOCTYPE html>
<html>
<head>
  <title>Titel wird in Kopfzeile angezeigt</title>
  <meta charset="utf-8">
</head>
<body>
  <h1>Überschrift</h1>
  
  <p> Inhalt eines Paragraphen mit
    <a href="seite2.html">Hyperlink</a>
  </p>
  <ul>
    <li>erster Listeneintrag</li>
    <li>zweiter Listeneintrag</li>
  </ul>
</body> https://www.w3.org/TR/html52/
</html> https://html.spec.whatwg.org
```

## HTML Tags

- <h1> ... <h6> header level 1 ... 6
- <a href="URL"> hyperlink anchor
- <img> image
- <p> paragraph
- <br> line break
- <hr> horizontal rule
- <pre> preformatted
- <ul> unordered list
- <ol> ordered list
- <li> list item
- <table> table
- <th> table header
- <tr> table row
- <td> table cell
- <strong>
- <em>
- <div>
- <span>
- ...

<https://wiki.selfhtml.org/wiki/HTML/Tutorials/HTML5-Grundger%C3%BCst>



Grundsätzliche Unterscheidung:

# Block versus Inline Element

Gruppierung von Content mit  
Absatz (*block element*)

z.B. <h1>, <p>, <div>

Block-Elemente werden  
untereinander angeordnet und  
belegen die gesamte Breite.

Gruppierung von Content ohne  
Absatz (*inline element*)

z.B. <a>, <em>, <b>, <span>

Inline-Elemente werden  
nebeneinander angeordnet im  
Fluss des Textes mit  
Zeilenumbruch wie jeweils  
erforderlich.



# HTML-Elemente

## Eine Auswahl

### Kommentare

<!-- Kommentar -->

- Überschriften: <h1> bis <h6>
  - 1,2,3 größer als Normal
  - 4 Normal
  - 5,6 kleiner als Normal

### Abschnitte (*sections*)

- <h1> Größte Überschrift </h1>
- <h2> bis <h6> Unterüberschriften
- neu in HTML5: Layout-Elemente:  
*<header>, <footer>, <nav>, <address>, <article>, <section>, <aside>, <hgroup>, ...*

### Gruppierung von Content

- <p> Paragraph
- <ol> Ordered List
- <ul> Unordered List
- <li> List Item

### Block-Elemente

### Block-Elemente

### Gruppierung von Content (Fortsetzung)

- <pre> a block of preformatted text
- <dl> description list
- <dt> description term (inside dl)
- <dd> description, definition, or value (inside dl)
- <figure> mit <figcaption>
- <div> Gruppierung von Content mit Absatz (*block content*), neutrales Tag, wenn kein anderes Tag passen würde, ohne Semantik.

### Inline-Elemente

### Text-level semantics

- <a> Anker-Element, meist für Hyperlinks
- <em> emphasis
- <strong> Betonung, "*strong importance*"
- <mark> markiert (*marked or highlighted*)
- <small> small print
- <sub> Tieferstellen (*subscript*)
- <sup> Höherstellen (*superscript*)
- <i> Kursiv (*italics*)
- <b> Fett (*bold*)
- <u> Unterstrichen (*underline*)
- <br> Zeilenumbruch (*break*)
- <hr> Horizontaler Trenner (*horizontal ruler*)
- <span> Gruppierung von Content ohne Absatz (*inline content*), neutrales Tag, wenn kein anderes Tag passen würde, ohne Semantik.

<https://www.w3.org/TR/html52/>

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

# HTML5 Layout Elements

Neuere HTML5-Tags zur Auszeichnung der Struktur

```
<!doctype html>
<header>
  <nav>
    <ul>
      <li><a href="#link_1.html">Startseite</a></li>
      <li><a href="#link_2.html">Unterseite 1</a></li>
    </ul>
  </nav>
</header>

<main>
  <article>
    <h1>CSS-basierte Layouts</h1><p>...some text</p>
  </article>

  <aside>
    <h2>Weiterführende Links</h2>
    <ul>
      <li><a href="#link_1.html">Wiki</a></li>
    </ul>
  </aside>
</main>

<footer>
  <a href="#kontakt.html">Kontakt</a>
</footer>
```

<header>

<footer>

<nav>

<main>

<article>

<section>

<aside>

Falls IE unterstützt werden soll,  
wird davon abgeraten



gleiche  
Wirkung wie  
<div>



# Weitere HTML-Elemente

Wichtig für Umlaute

Eine Auswahl (Fortsetzung)

## Document Metadata

- <head> Kopfbereich, gruppiert Metadaten
- <title> Titel, erscheint im Tabulator
- <base> Basis-URL siehe [∞](#)
- <link> Verweis auf Ressourcen, (*link document to other resources*)
- <style> Stil, Layout in CSS
- <script> meist JavaScript
- <meta> sonstige Metadaten

## Edits

- <ins> insert
- <del> delete

## Embedded Content

- <img>
- <audio>
- <video>

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Titel wird im Tab-Kopf angezeigt</title>
  <link rel="stylesheet" href="stylesheet.css" type="text/css">
  <link rel="icon" type="image/x-icon" href="favicon.ico">
  <link rel="contents" title="Inhaltsverzeichnis" href="../inhalt.html">
</head>
<body>
  <!-- Wie man sieht, sieht man nichts. -->
</body>
</html>
```



# HTML Text Formatting

## Beispiele

- <strong>wichtiger Text</strong> → **wichtiger Text**
- <i>kursiver Text</i> → *kursiver Text*
- <em>betonter Text</em> → **betonter Text**
- <mark>hervorgehobener Text</mark> → **hervorgehobener Text**
- <small>kleiner Text</small> → *kleiner Text*
- <del>gelöschter Text</del> → ~~gelöschter Text~~
- <ins>hinzugefügter Text</ins> → hinzugefügter Text
- <u>unterstrichener Text</u> → unterstrichener Text
  - hochgestellter Text
  - tiefgestellter Text
- <sup>hochgestellter Text</sup>
- <sub>tiefgestellter Text</sub>

<https://www.w3.org/TR/html52/>

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

Bei HTML steht die Gruppierung von Content und semantische Auszeichnung (*markup*) im Vordergrund. Stil und Layout sind Themen für CSS.

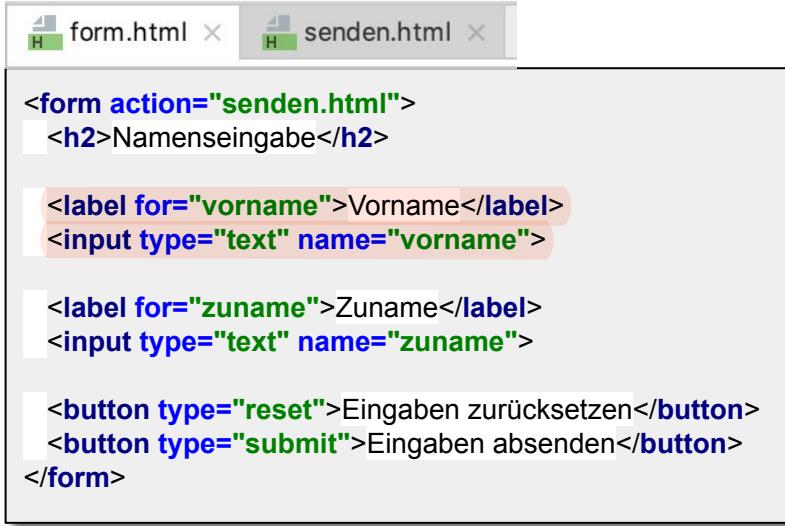


# HTML-eigene Zeichen 🔗

Zeichen	Beschreibung	Name in HTML	NCR hex.	NCR dez.
>	schließende spitze Klammer	&gt;	&#x3E;	&#62;
<	öffnende spitze Klammer	&lt;	&#x3C;	&#60;
'	Apostroph, einfaches Anführungszeichen oben (nur bei XHTML definiert)	&apos;	&#x27;	&#39;
&	Ampersand-Zeichen, kaufmännisches Und	&amp;	&#x26;	&#38;
"	Anführungszeichen oben	&quot;	&#x22;	&#34;



# HTML-Formulare



```
<form action="senden.html">
  <h2>Namenseingabe</h2>

  <label for="vorname">Vorname</label>
  <input type="text" name="vorname">

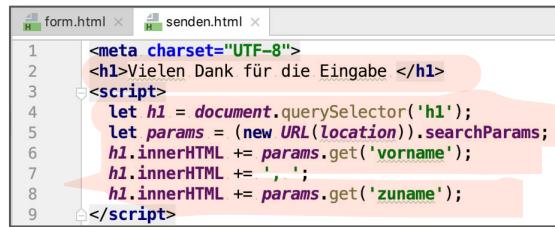
  <label for="zuname">Zuname</label>
  <input type="text" name="zuname">

  <button type="reset">Eingaben zurücksetzen</button>
  <button type="submit">Eingaben absenden</button>
</form>
```

## Namenseingabe

Vorname

Zuname



```
1  <meta charset="UTF-8">
2  <h1>Vielen Dank für die Eingabe </h1>
3  <script>
4    let h1 = document.querySelector('h1');
5    let params = (new URL(location)).searchParams;
6    h1.innerHTML += params.get('vorname');
7    h1.innerHTML += ', ';
8    h1.innerHTML += params.get('zuname');
9  </script>
```

Vielen Dank für die Eingabe Wolfgang, Müller

<https://wiki.selfhtml.org/wiki/HTML/Formulare/form>



# HTML-Formular-Elemente

Das `form`-Element kann ein oder mehrere dieser Kind-Elemente enthalten

- `input` (Eingabefeld)
- `textarea` (mehrzeilige Textfelder)
- `button` (Button, Schalter)
- `select` (Auswahllisten)
- `option` (Auswahllisten)
- `optgroup` (verschachtelte Auswahllisten)
- `fieldset` (Gruppierungen)
- `label` (Beschriftung für Eingabefelder)

<https://wiki.selfhtml.org/wiki/HTML/Formulare/form>



## HTML Input Types

- `<input type="button">`
- `<input type="checkbox">`
- `<input type="color">`
- `<input type="date">`
- `<input type="datetime-local">`
- `<input type="email">`
- `<input type="file">`
- `<input type="hidden">`
- `<input type="image">`
- `<input type="month">`
- `<input type="number">`
- `<input type="password">`
- `<input type="radio">`
- `<input type="range">`
- `<input type="reset">`
- `<input type="search">`
- `<input type="submit">`
- `<input type="tel">`
- `<input type="text">`
- `<input type="time">`
- `<input type="url">`
- `<input type="week">`

[https://www.w3schools.com/html/html\\_forms.asp](https://www.w3schools.com/html/html_forms.asp)

# Metadata für Web App

```
<meta charset="utf-8">  
  
<!-- Make the app title different than the page title. -->  
<meta name="apple-mobile-web-app-title" content="WE">  
  
<meta name="apple-mobile-web-app-capable" content="yes">  
  
<meta name="apple-mobile-web-app-status-bar-style" content="black">  
  
<meta name="viewport" content="initial-scale=1.0, user-scalable=no">  
  
<!-- Disable automatic phone number detection. -->  
<meta name="format-detection" content="telephone=no">  
  
<!-- retina iPad iOS 7 für iPhone 6 Plus -->  
<link rel="apple-touch-icon" href="img/hbrs_logo180.png" sizes="180x180">
```

<https://www.w3.org/TR/appmanifest/>  
⇒ In Zukunft ersetzt durch [WebApp-Manifest-Datei](#)



# HTML ist eine Auszeichnungssprache (*Markup-Sprache*)

1986: **SGML** = *Standard Generalized Markup Language* ist eine Metasprache, mit der Auszeichnungssprachen (engl. markup languages) für Dokumente definiert werden können, ISO-Standard: ISO 8879:1986. In SGML beschreibt eine Dokumenttypdefinition (DTD), was ein valides Dokument ist und was nicht.

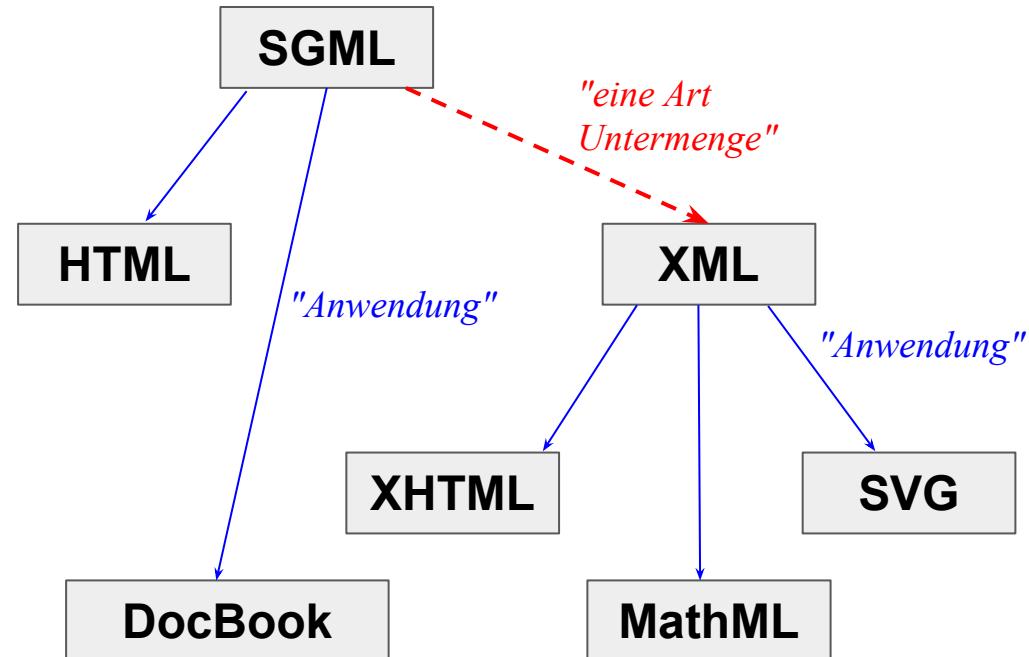
1998: **XML** = *Extensible Markup Language* ist eine Metasprache, auf deren Basis durch strukturelle und inhaltliche Einschränkungen anwendungsspezifische Sprachen (Domain Specific Languages, DSL) definiert werden.

1999: **XMLSchemata** = Alternative zur DTD, die selbst in XML geschrieben wird.

2001: **SVG** = Scalable Vector Graphics

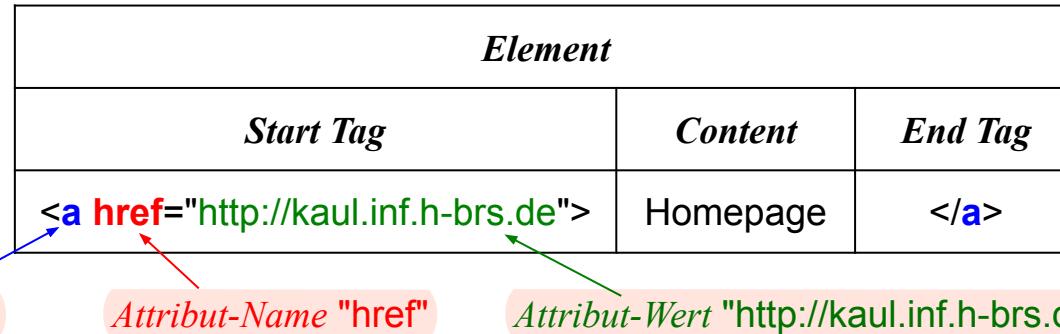
**wohlgeformt**: korrekt als XML-Dokument

**valide**: konform zur Grammatik (in DTD oder XMLSchemata)



# Hauptunterschiede zwischen SGML und XML

SGML	XML
Schließende Klammern können als optional definiert sein: <code>&lt;img src="image.png" &gt;</code> <code>&lt;p&gt;Lorem...&lt;p&gt;Ipsum...</code>	Schließende Klammern sind immer verpflichtend: <code>&lt;img src="image.png" /&gt;</code> <code>&lt;p&gt;Lorem...&lt;/p&gt;&lt;p&gt;Ipsum...&lt;/p&gt;</code>
Bei Element- und Attribut-Namen wird zwischen Groß- und Kleinschreibung NICHT unterschieden ( <i>not case-sensitive</i> ).	Bei Element- und Attribut-Namen wird zwischen Groß- und Kleinschreibung unterschieden ( <i>case-sensitive</i> ).
Attributwerte müssen nur dann mit Apostrophen begrenzt werden, wenn sie auch andere als alphanumerische Zeichen enthalten (z.B. Leerzeichen).	Attributwerte müssen immer mit Apostrophen begrenzt werden.



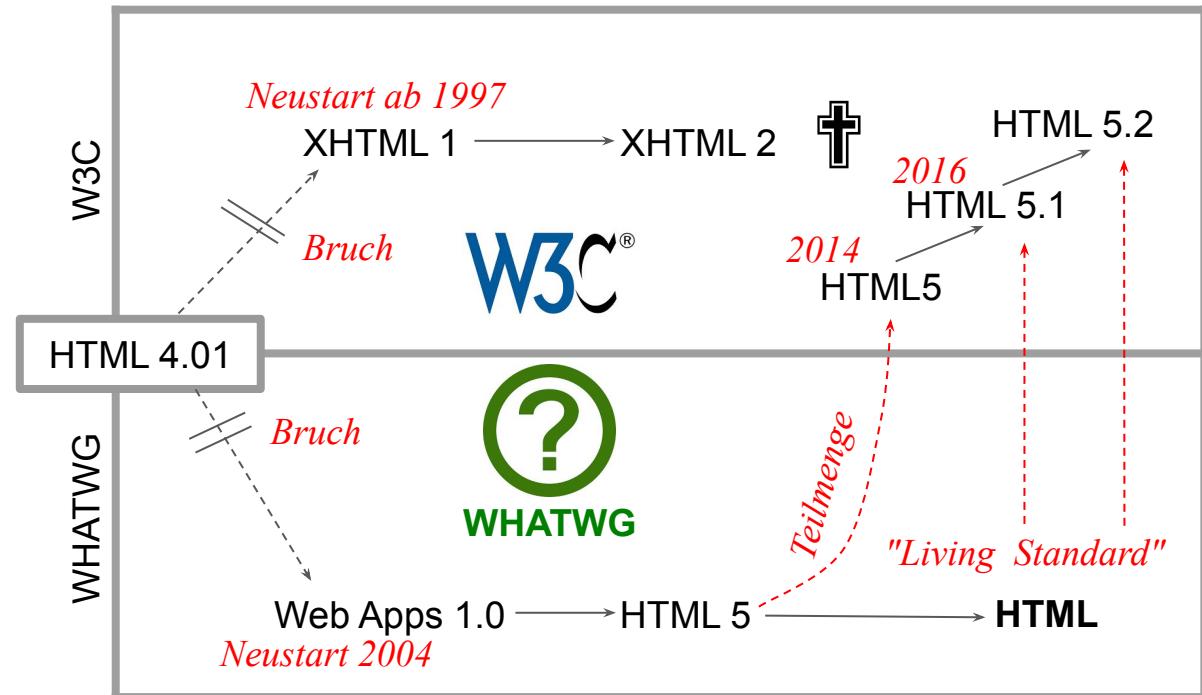
# ⇒ Hauptunterschiede zwischen HTML und XHTML

HTML	XHTML
HTML kann als Anwendung von SGML verstanden werden. HTML wird durch eine DTD definiert.	XHTML ist eine Anwendung von XML
Schließende Klammern können als optional definiert sein: <pre>&lt;img src="image.png" &gt; &lt;p&gt;Lorem...&lt;p&gt;Ipsum...</pre>	Schließende Klammern sind immer verpflichtend: <pre>&lt;img src="image.png" /&gt; &lt;p&gt;Lorem...&lt;/p&gt;&lt;p&gt;Ipsum...&lt;/p&gt;</pre>
Bei Element- und Attribut-Namen wird zwischen Groß- und Kleinschreibung NICHT unterschieden ( <i>not case-sensitive</i> ).	Bei Element- und Attribut-Namen wird zwischen Groß- und Kleinschreibung unterschieden ( <i>case-sensitive</i> ).
Attributwerte müssen nur dann mit Apostrophen begrenzt werden, wenn sie auch andere als alphanumerische Zeichen enthalten (z.B. Leerzeichen).	Attributwerte müssen immer mit Apostrophen begrenzt werden.



# HTML Timeline: "Standards War"

- 1989 TBL erfindet HTML
- 1990er Jahre: Browser implementieren HTML direkt und ignorieren SGML
- ab 1997: XHTML
- 2009: W3C erfindet XHTML 2
- ab 2004: Browserhersteller ignorieren XML und XHTML und gründen eigene Arbeitsgruppe, die WHATWG und entwerfen HTML 5 als "Living Standard HTML" (auch ohne Versionsnummer)
- 2012: W3C übernimmt HTML-Anteile aus HTML 5



<https://en.wikipedia.org/wiki/HTML>

# XHTML 1.0 ist nicht tot

Heute noch verbliebene Anwendungen von XHTML

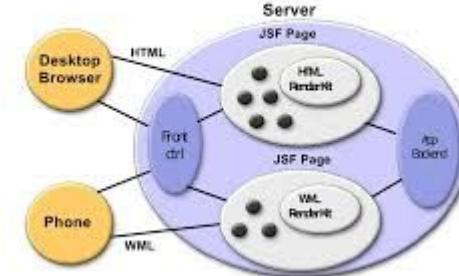
```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0
Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml"
      xmlns:h="http://xmlns.jcp.org/jsf/html">
<head>
    <title>Hello JSF World</title>
</head>
<body>
    <h3><h:outputText value="Hello JSF World" /></h3>
    <p> <h:outputText value="Host: #{header['Host']}' /><br/>
        <h:outputText value="User-Agent: #{header['User-Agent']}' /></p>
</body>
</html>
```

<http://www.torsten-horn.de/techdocs/jsf.htm>

Manche Technologien passen  
sehr gut zu dem XML-Ansatz  
von HTML: XHTML.



JSF Architecture

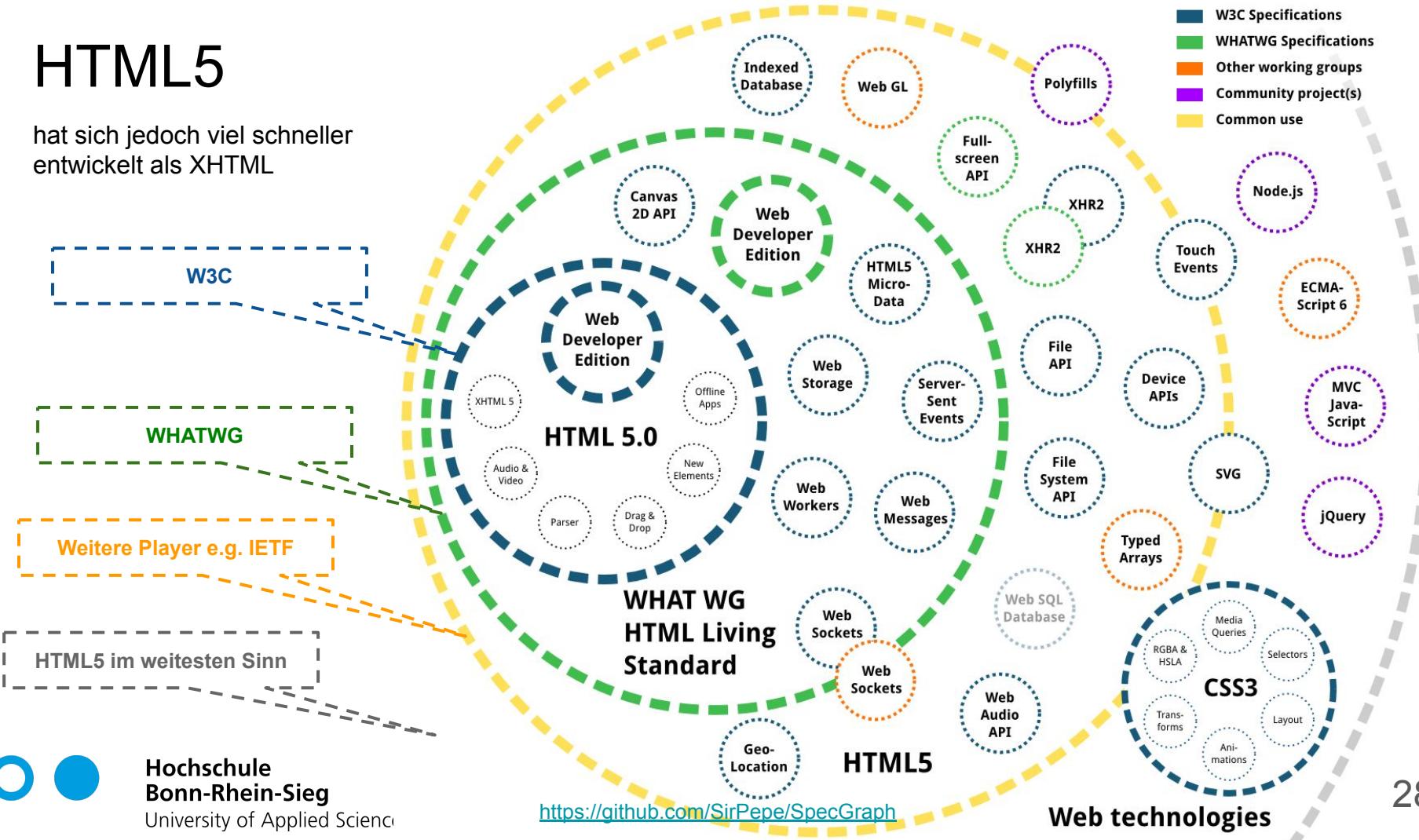


<http://www.oracle.com/technetwork/java/javaee/javaserverfaces-139869.html>



# HTML5

hat sich jedoch viel schneller entwickelt als XHTML

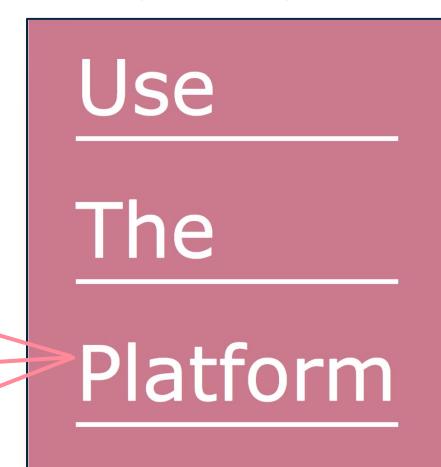


# Das WWW ist die erfolgreichste Integrationsplattform

- für vernetzte Informationen
- für vernetzte Informationsarchitekturen
- für Technologien (HTML, CSS, JavaScript, WASM, XML, JSON, SVG, WebGL, ...)

Gemeint ist das WWW.

Gemeint ist der Browser-Standard.



# HTML 6 ?

Yes and no.

Several answers have stated that the Worldwide Web Consortium (W3C) and the Web Hypertext Application Technology Working Group (WHATWG) are probably not thinking about HTML 6 since HTML 5 is still a work in progress.

However, though the W3C is still using version numbers, while the WHATWG, made an announcement that **they will no longer use version numbers for HTML**, so several answers have stated that the question is somewhat irrelevant.

<https://www.quora.com/Are-they-preparing-HTML-6>

## Are they preparing HTML 6?



# Wie lernt man HTML?



<https://wiki.selfhtml.org/wiki/HTML>

w3schools.com

# HTML

The language for building web pages

LEARN HTML    HTML REFERENCE

Ad closed by Google

Result Size: 336 x 279

```
<!DOCTYPE html>
<html>
<head>
<title>Page Title</title>
</head>
<body>
<h1>This is a Heading</h1>
<p>This is a paragraph.</p>
</body>
</html>
```

This is a Heading

This is a paragraph.



# Erste Woche: HTML und CSS lernen

The screenshot shows the homepage of w3schools.com. The top navigation bar includes links for TUTORIALS, REFERENCES, EXAMPLES, and EXERCISES. Below the navigation, there are three main sections: **HTML**, **CSS**, and **JavaScript**. Each section contains a brief description and a "Try It Yourself" button. The **HTML** section shows an example of HTML code:

```
<!DOCTYPE html>
<html>
<title>HTML Tutorial</title>
<body>
<h1>This is a heading</h1>
<p>This is a paragraph.</p>
</body>
</html>
```

The **CSS** section shows an example of CSS code:

```
body {
background-color: lightblue;
}

h1 {
color: white;
text-align: center;
}

p {
font-family: verdana;
font-size: 20px;
}
```

The **JavaScript** section shows an example of JavaScript code:

```
<button onclick="myFunction()">Click Me!</button>
```

This screenshot shows the "Tryit Editor v3.3" interface within a web browser window. The URL is <https://www.w3schools.com/html/tryit.asp?...>. The editor displays the HTML code from the previous screenshot. A green "Run" button is visible above the code. To the right, the generated output is shown in a large text area:

```
<!DOCTYPE html>
<html>
<head>
<title>Page Title</title>
</head>
<body>
<h1>This is a Heading</h1>
<p>This is a paragraph.</p>
</body>
</html>
```

The output text area contains the rendered HTML content: "This is a Heading" and "This is a paragraph."

# Was sind die primären Nachschlagequellen?

# W3C

# WHATWG

# MDN

The screenshot shows a browser window for the W3C Working Draft of HTML 5.2. The title bar says "w3.org" and "HTML Standard". The main content area displays the draft document with sections like "HTML 5.2", "This version:", "Latest published version:", "Latest version of HTML:", "Editor's Draft:", "Previous Versions:", and "Editors:". It also lists names such as Steve Faulkner, Aron Eichholz, Travis Leithead, Alex Danilo, and Sangwhan Moon. At the bottom, there are links for "Former Editors:" and "Erika Doyle Navara". A W3C logo is visible in the top right corner.

<https://www.w3.org/TR/html52/>

The screenshot shows a browser window for the WHATWG HTML Living Standard. The title bar says "html.spec.whatwg.org" and "HTML Standard". The main content area displays the living standard document with sections like "HTML", "Living Standard – Last Updated 19 April 2017", and a large "WHATWG" logo. Below the logo is a search bar with a question mark icon. The document structure includes sections for "Semantics, structure, and APIs of HTML documents", "Table of Contents", and "Embedded content". A detailed table of contents for the "The elements of HTML" section is shown, starting with "4.1 The document element" and "4.1.1 The `html` element".

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

The screenshot shows a browser window for the MDN Web Documentation. The title bar says "MDN-Web-Dokumentation" and "Sicher". The main content area features a large blue header with the text "Informationen für Entwickler von Entwicklern.". Below the header are navigation links for "Web-Technologien", "Web-Entwicklung lernen", and "Entwicklerwerkzeuge". There are also sections for "Auf Sauce Labs testen", "Lernen Sie das Beste aus dem Bereich Web-Entwicklung", and a newsletter sign-up form.

<https://developer.mozilla.org>



# WWW-Deployment-Modelle

- Es reicht nicht, wenn es in meinem Browser läuft.
  - Es muss dem Standard genügen. ⇒ W3C
  - Es muss zukunftsfähig sein ⇒ WHATWG
  - Woran sich die meisten Entwickler orientieren ⇒ MDN
- Es soll "in allen" Browsern laufen
  - häufig nicht erreichbar wegen IE-Altversionen (Internet-Explorer)
- Es soll "in allen modernen" Browsern laufen
  - Was heißt "modern"?
  - z.B. Firefox beherrscht auch in seiner neuesten Variante noch nicht Shadow DOM
- HTML ≠ PDF
  - Was soll das Ergebnis sein? Wann ist die WebApp "korrekt"?
  - Responsives HTML ⇒ "Auf jedem Device anders" ist gewünscht
  - ⇒ HTML ist eine abstraktere Seitenbeschreibung als PDF



# Die Lösung des Deployment-Problems: Can I use?

<https://caniuse.com>

Shadow DOM v1  - WD

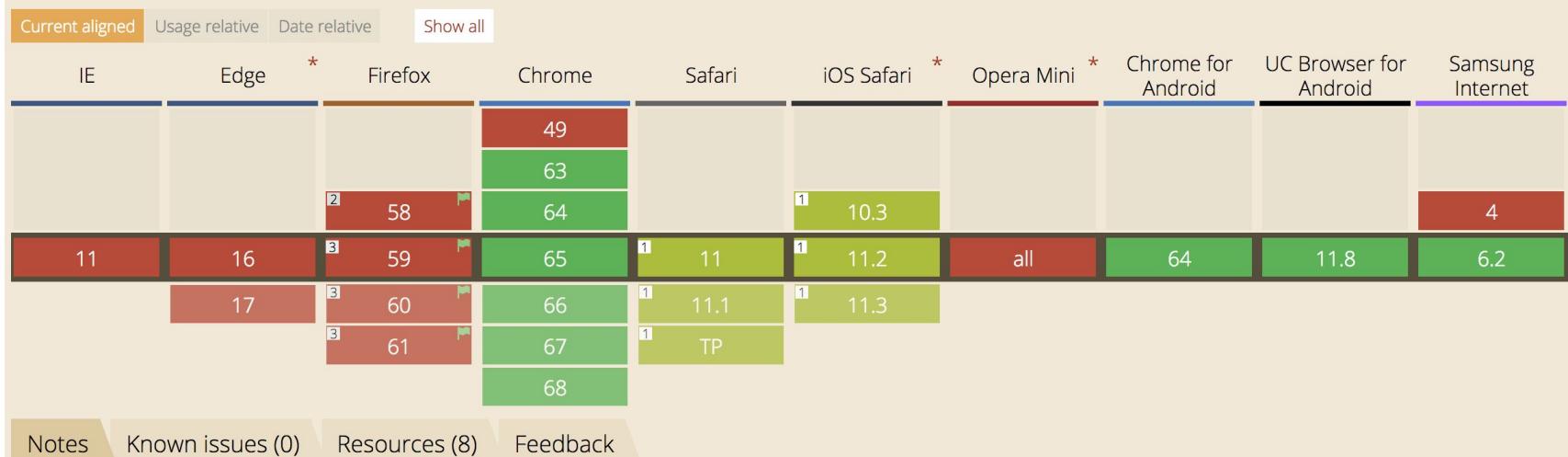
Usage

Global

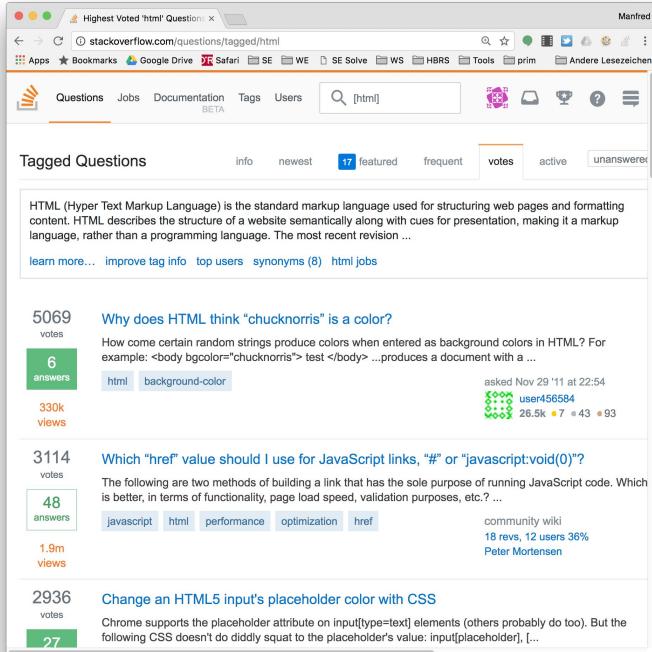
% of all users

66.89% + 12.27% = 79.16%

Method of establishing and maintaining functional boundaries between DOM trees and how these trees interact with each other within a document, thus enabling better functional encapsulation within the DOM & CSS.



# Wo bekommt man Antworten?



A screenshot of a web browser window titled "Highest Voted 'html' Questions". The URL is stackoverflow.com/questions/tagged/html. The browser has several tabs open, including "SE Solve", "WS", "HBRs", "Tools", and "prim". The main content area shows a list of tagged questions under the heading "Tagged Questions". The first question is "Why does HTML think "chucknorris" is a color?", with 5069 votes, 6 answers, and 330k views. The second question is "Which "href" value should I use for JavaScript links, "#" or "javascript:void(0)"?", with 3114 votes, 48 answers, and 1.9m views. The third question is "Change an HTML5 input's placeholder color with CSS", with 2936 votes, 27 answers, and 1.9m views.



<http://stackoverflow.com/questions/tagged/css>

Hochschule  
Bonn-Rhein-Sieg

University of Applied Sciences

# HTML Table

```
<!DOCTYPE html>
<html lang="de">
<head>
  <meta charset="UTF-8">
  <title>Table</title>
</head>
<body>
  <table>
    <tr>
      <th>A</th><th>B</th><th>C</th>
    </tr>
    <tr>
      <td>a</td><td>b</td><td>c</td>
    </tr>
    <tr>
      <td>x</td><td>y</td><td>z</td>
    </tr>
  </table>
</body>
</html>
```

A	B	C
a	b	c
x	y	z

HTML 4:

```
<table border="1">
```

HTML 5:

CSS style

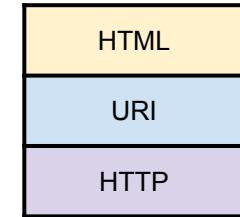
```
<head>
  <meta charset="UTF-8">
  <title>Table</title>
  <style>
    table, th, td {
      border: thin solid black;
    }
    table {
      border-collapse: collapse;
    }
  </style>
</head>
<body>
  <table> ... </table>
```

A	B	C
a	b	c
x	y	z

A	B	C
a	b	c
x	y	z



Diese 3 Konzepte sind die Basis für das WWW:



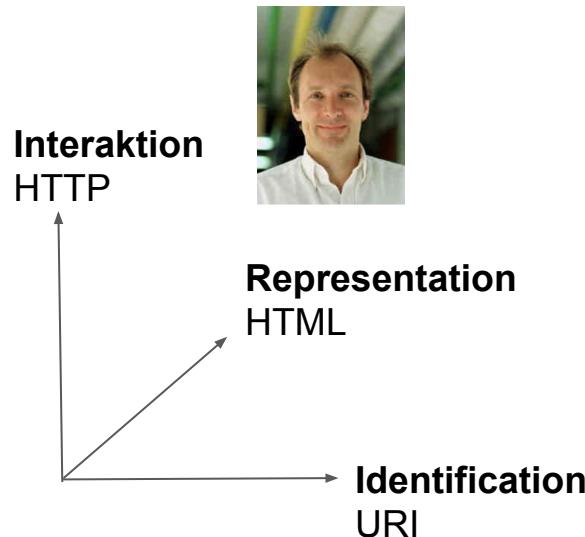
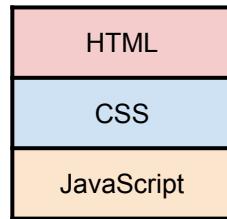
Bis hierher  
die anfänglichen Ideen von Tim-Berners Lee.

**Das erste Web war sicher, performant, skalierbar, ...**

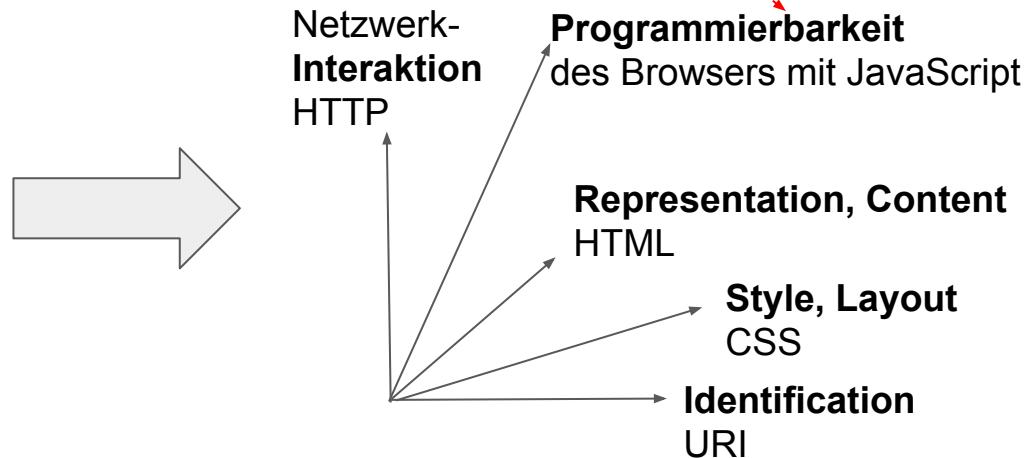
Was kam danach?



# Weitere Dimensionen der Emergenz ...



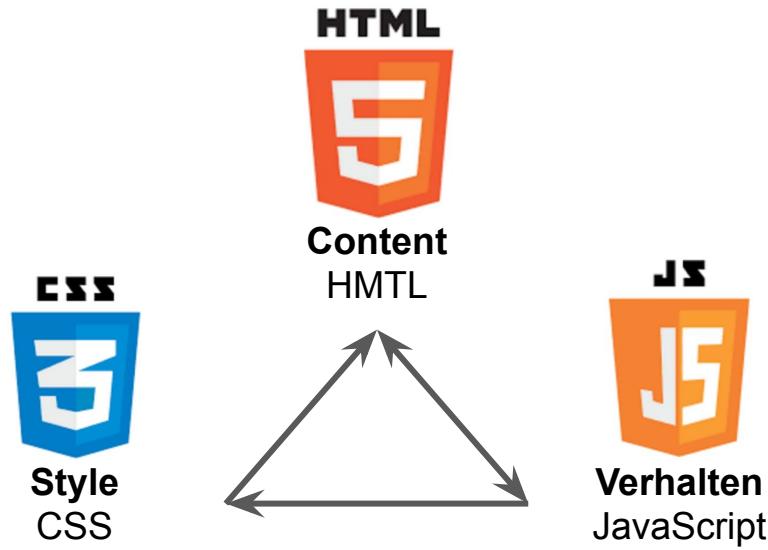
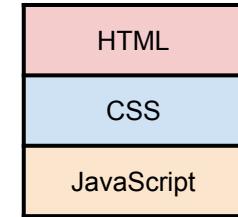
alte WWW-Triade



Das heutige WWW



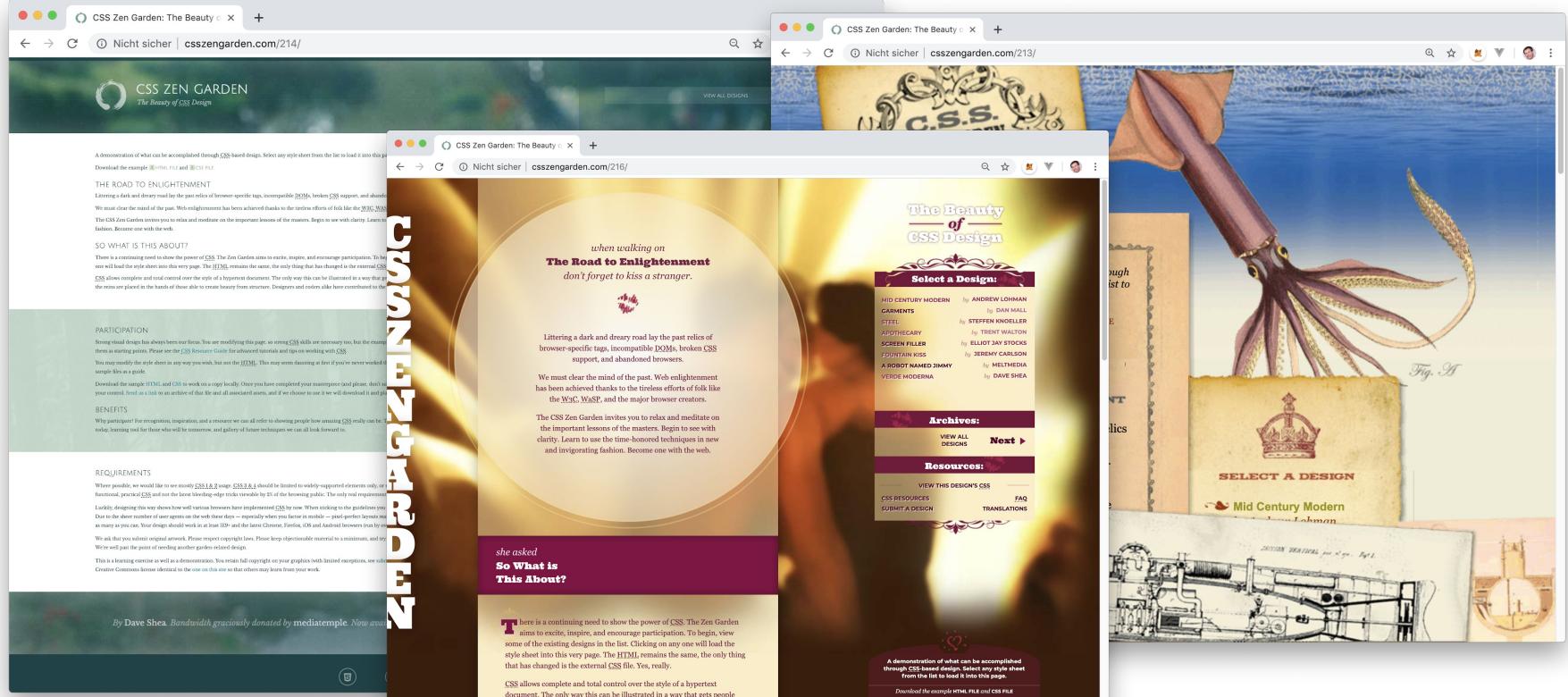
# Heutige Browser-Triade



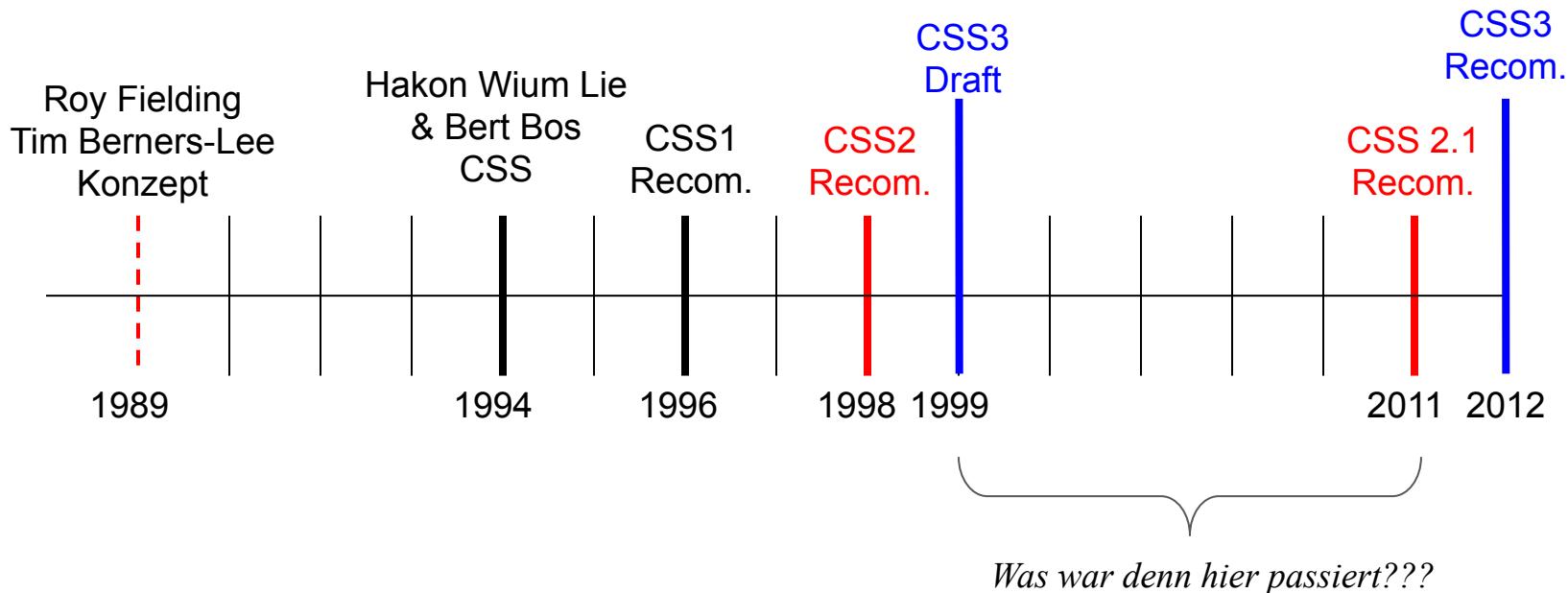
MVC = Model View Controller im Browser möglich, aber nicht zwingend



# Trennung von Struktur und Layout



# CSS Timeline

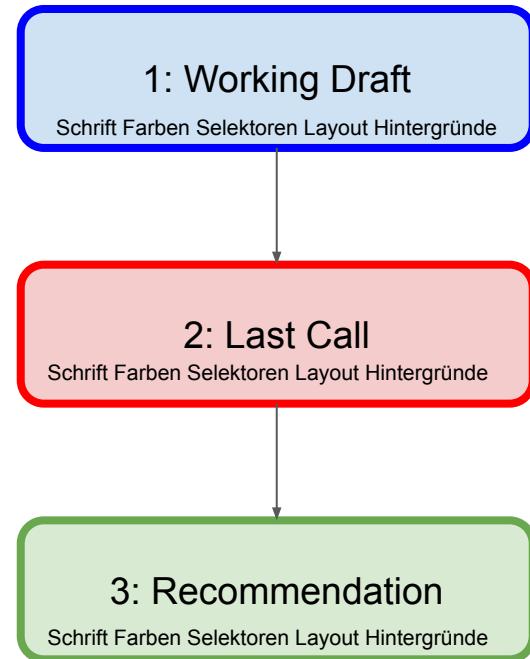


[https://de.wikipedia.org/wiki/Cascading\\_Style\\_Sheets](https://de.wikipedia.org/wiki/Cascading_Style_Sheets)

# CSS 1 und 2 waren monolithische Standards

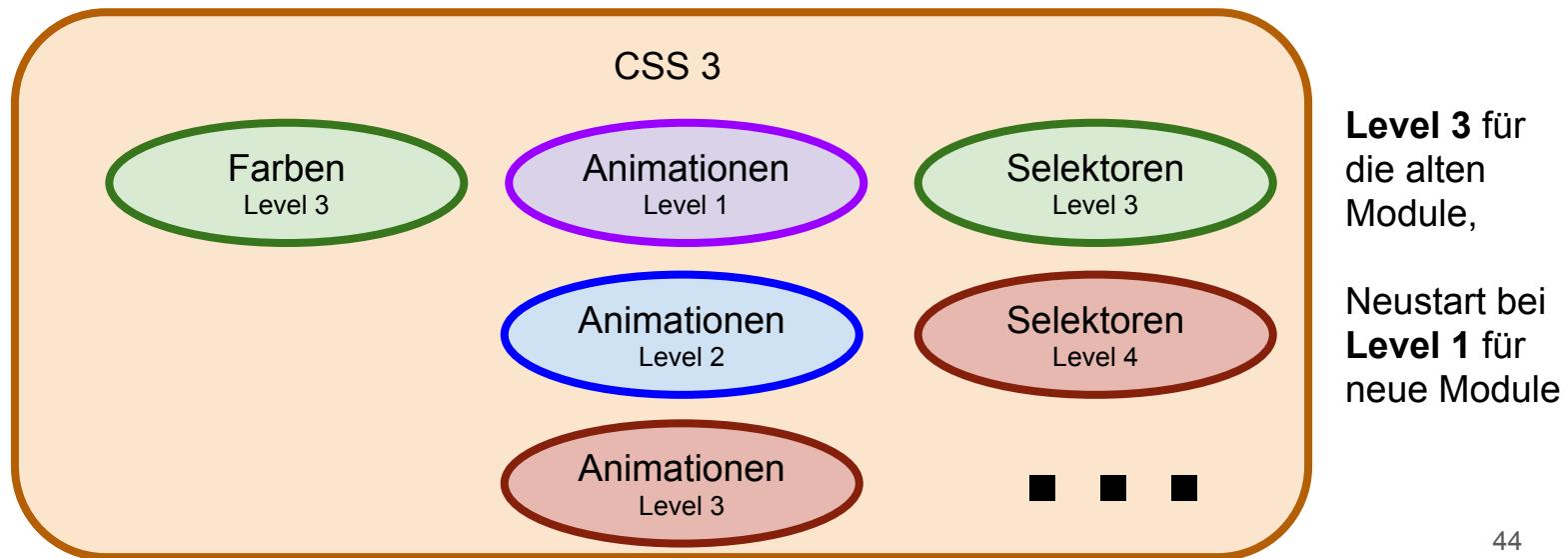
Alle CSS-Regeln zu Schrift, Farben, Selektoren, Layout, Hintergründe wurden in einem Standard *gemeinsam* verabschiedet

- ⇒ Alles musste fertig sein, damit der nächste Schritt möglich wurde.
- ⇒ Verletzung des Prinzips der Orthogonalität verschiedener Entwicklungslinien
- ⇒ CSS-Standardisierung zu schwerfällig.

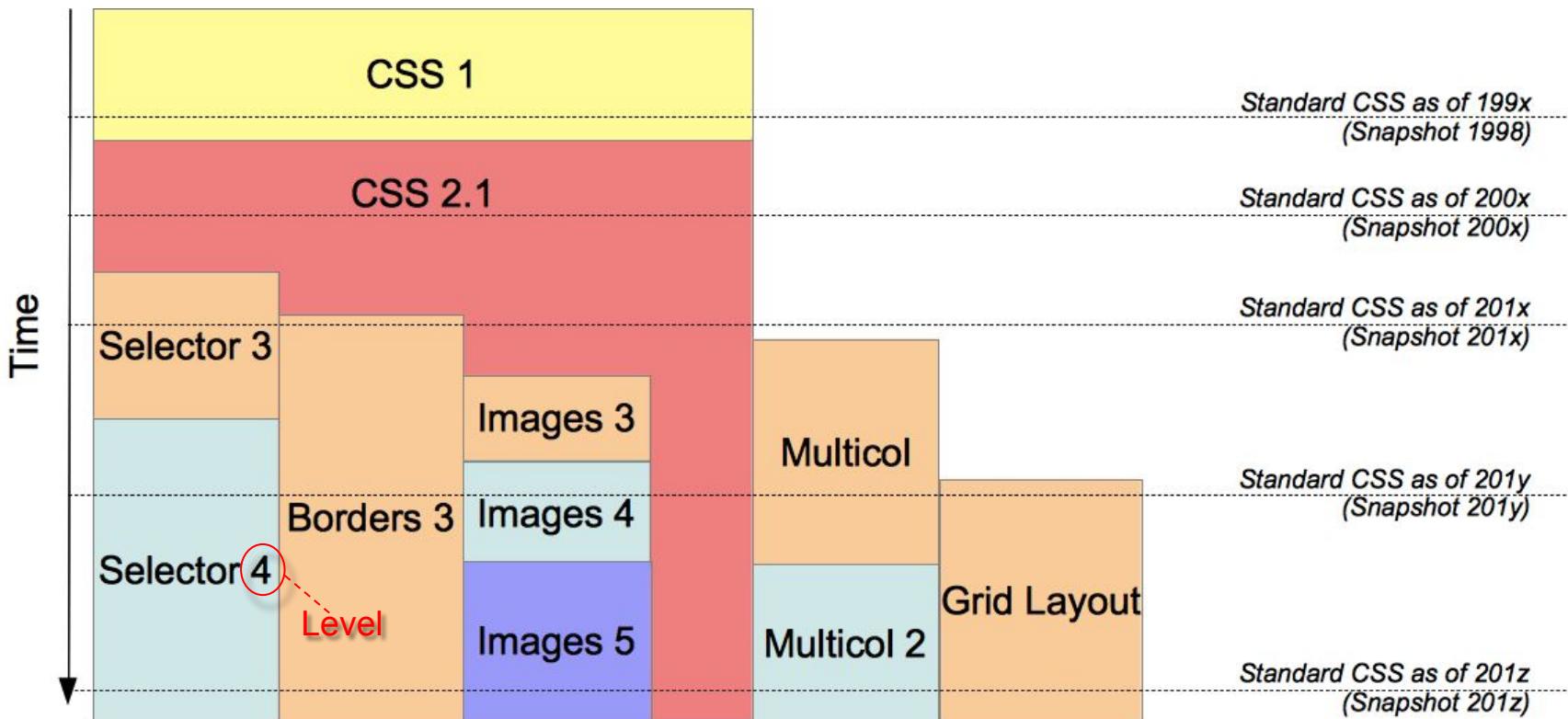


# CSS3 ⇒ "Living Standard"

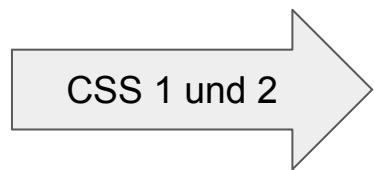
- Zerlegung der Spezifikation in Module
- **Unabhängige** Weiterentwicklung der Module



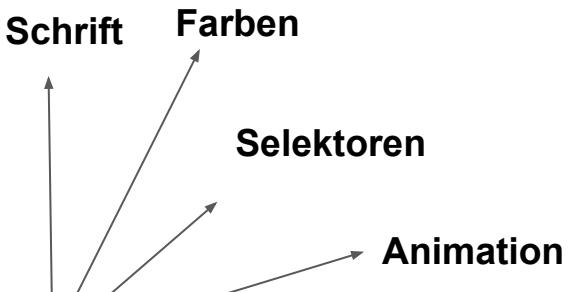
# Zerlegung des CSS-Standards in Module



# Seit das Prinzip der Orthogonalität in CSS angewendet wird, hat sich der Fortschritt wieder beschleunigt



monolithisch



*CSS Living Standard*  
als Sammlung von Modulen, die  
unabhängig weiterentwickelt  
werden

TABELLARISCHE ÜBERSICHT DER SPEZIFIKATIONEN

		Aktuell	Zukünftig	
Abgeschlossene Werke				Latest stable CSS
CSS Snapshots 2017	NOTE			
CSS Snapshots 2015	NOTE			
CSS Snapshots 2010	NOTE			
CSS Snapshots 2007	NOTE			
CSS Color Level 3	REC	REC		
CSS Namespaces	REC	REC		
Selectors Level 3	REC	REC		
CSS Level 2 Revision 1	REC	REC		
CSS Level 1	REC			Unmaintained, see Snapshot
CSS Print Profile	NOTE			
Media Queries	REC	REC		
CSS Style Attributes	REC	REC		
Hohe Priorität		Aktuell	Zukünftig	
CSS Backgrounds and Borders Level 3	CR	PR		
CSS Conditional Rules Level 3	CR	CR		
CSS Multi-column Layout	CR	CR		
CSS Values and Units Level 3	CR	PR		
CSS Cascading and Inheritance Level 3	CR	PR		
Hohe Priorität		Aktuell	Zukünftig	
CSS Image Values and Replaced Content Level 3	CR	CR		
CSS Speech	CR	CR		
CSS Flexible Box Layout	CR	PR		
CSS Text Decoration Module Level 3	CR	CR		
CSS Fonts Level 3	CR	CR		
CSS Writing Modes Level 3	CR	CR		
CSS Shapes Level 1	CR	CR		
CSS Masking Level 1	CR	CR		
CSS Counter Styles Level 3	CR	PR		
CSS Fragmentation Level 3	CR	PR		
Compositing and Blending Level 1	CR	CR		
CSS Syntax Level 3	CR	CR		
CSS Grid Layout Level 1	CR	PR		
CSS Basic User Interface Level 3	CR	PR		
Geometry Interfaces Module Level 1	CR	CR		
Mittlere Priorität		Aktuell	Zukünftig	
CSS Animations	WD	WD		
Web Animations 1.0	WD	WD		
CSS Text Level 3	LC	CR		
CSS Transforms	WD	WD		
CSS Transitions	WD	CR		
CSS Cascading Variables	CR	PR		
CSS Box Alignment Module Level 3	WD	WD		
CSS Box Alignment Level 2	CR	PR		
Preview of CSS Level 2	FPWD	NOTE		
Mittlere Priorität		Aktuell	Zukünftig	
CSS Paged Media Level 3	WD	WD		
CSS View	WD	WD		
Selectors Level 4	WD	CR		
CSS Intrinsic & Extrinsic Sizing Module	WD	WD		
CSS Overflow Level 3	FPWD	WD		
CSS Display Level 4	WD	WD		
CSS Pseudo-Elements Module Level 4	FPWD	WD		
Mittlere Priorität		Aktuell	Zukünftig	
CSS Backgrounds and Borders Level 4	WD	WD		
CSS Device Adaptation	WD	WD		
CSS Exclusions	WD	WD		
Filter Effects	WD	WD		
CSS Generated Content for Paged Media	WD	WD		
CSS Page Floats	FPWD	WD		
CSS Template Layout	NOTE	NOTE		
CSS Line Grid	WD	WD		
CSS Lists Level 3	WD	WD		
CSS Positioned Layout Level 3	WD	WD		
CSS Regions	WD	WD		
CSS Ruby Level 1	WD	WD		
CSS Tables Level 3	FPWD	WD		
CSS Object Model	WD	WD		
CSS Font Loading	LC	WD		
CSS Scoping Level 1	FPWD	WD		
Media Queries Level 4	WD	WD		
CSS Image Values Level 4	FPWD	WD		
CSS Compositing and Blending Level 3	FPWD	WD		
CSS Syntax Level 4	FPWD	WD		
CSS Grid Layout Level 1	FPWD	WD		
Motion Path Module Level 1	FPWD	WD		
CSS Round Display Level 1	FPWD	WD		
CSS Basic User Interface Module Level 4	FPWD	WD		
CSS Text Module Level 4	FPWD	WD		
CSS Painting API Level 4	FPWD	WD		
CSS Properties and Values API Level 1	FPWD	WD		

# Syntaktischer Aufbau von CSS-Regeln

Allgemein:

**Selektor1 [, Selektor2 [, ...] ] {**

Eigenschaft-1: Wert-1;

...

Eigenschaft-n: Wert-n[;]

$n \geq 0$

}

CSS-Deklaration

*/\* Kommentar \*/*

*/\* In eckigen Klammern stehen optionale Angaben \*/*

Beispiel:

**h1, h2 {**  
**color: red; /\* rot \*/**  
**border: thin solid black;**  
**}**

Grouped Selectors

Grouped Declaration

# 3 Arten der Einbettung CSS in HTML

1

2

3

<link>	<style>	<tag>
<pre>&lt;!DOCTYPE html&gt; &lt;head&gt;   &lt;title&gt; Demo &lt;/title&gt;   &lt;link rel="stylesheet" href="s.css"&gt; &lt;/head&gt;  &lt;h1&gt; Header &lt;/h1&gt;</pre>	<pre>&lt;!DOCTYPE html&gt; &lt;style&gt;   h1 {     color: red;   } &lt;/style&gt;  &lt;h1&gt; Header &lt;/h1&gt;</pre>	<pre>&lt;!DOCTYPE html&gt; &lt;h1 style="color:red;"&gt; Header &lt;/h1&gt;</pre>

external style sheet

internal style sheet

inline style



# CSS External Style Sheet

HTML-Datei

```
<!DOCTYPE html>
<html>
  <head>
    <title>Titel wird in Kopfzeile angezeigt</title>
    <meta charset="utf-8">
    <link rel="stylesheet" href="stylesheet.css">
  </head>
  <body>
    <h1>Überschrift</h1>
    
    <p>Inhalt eines Paragraphen mit
      <a href="seite2.html">Hyperlink</a>
    </p>
    <ul>
      <li>erster Listeneintrag</li>
      <li id="second">zweiter Listeneintrag</li>
    </ul>
  </body>
</html>
```

CSS-Datei

```
body {
  background-color: lightblue;
}

h1 {
  color: red;
  text-align: center;
}

a {
  background-color: white;
}

img.center {
  display: block;
  margin-left: auto;
  margin-right: auto;
}

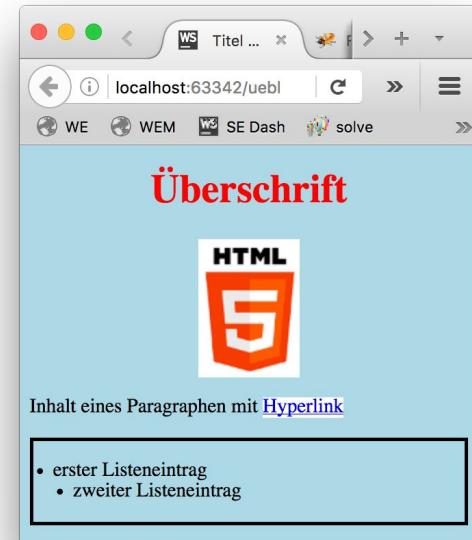
ul {
  border: solid;
  padding: 1em;
}

li#second {
  margin-left: 1em;
}
```

CSS-Syntax:

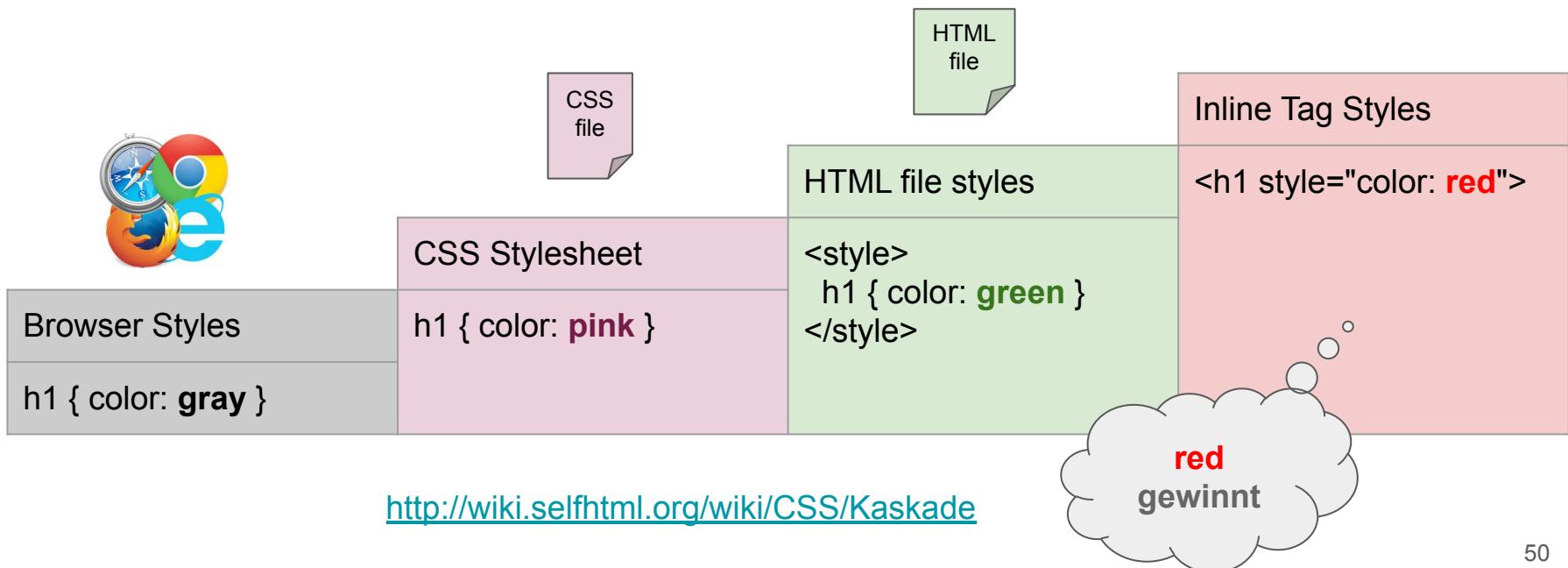
```
selector {
  property: value;
}
```

Browser



# Die CSS-Kaskade

- Styles der höheren Ebene überschreiben Styles darunter liegenden Ebenen



# Die CSS-Kaskadierungsregeln

- Die letzte Deklaration gewinnt.
- Inline-Styles > Webseiten-Styles > externe CSS-Dateien
- Ablauf der Kaskade (mit steigender Wichtigkeit):
  - Deklarationen im Browser-Stylesheet
  - Deklarationen des Benutzers
  - Deklarationen des Autors
  - Wichtige Deklarationen (mit !important) des Autors
  - Wichtige Deklarationen (mit !important) des Benutzers

<http://wiki.selfhtml.org/wiki/CSS/Kaskade>

<https://stackoverflow.com/questions/37910944/set-multiple-values-for-one-css-attribute>

# Selektoren stellen die Beziehung zwischen HTML und CSS her

Selektoren	HTML	CSS
Typ	<p>	p { regel: wert; }
Identifikator	<p id="eindeutig">	#eindeutig { color: red }
Klasse	<p class="klasse">	.klasse
Attribute	<img width="100">	[width="100"]
Pseudoklassen	<div><p><p></div>	:first-child { font-weight: bold } :hover { color: red }
Pseudoelemente	<p>Am Anfang ...</p>	::first-letter { font-size: large }

# CSS-Selektoren (*eine Auswahl*)

Selektor	Selektion: alle Elemente mit ...
h1	<b>Tag</b> <h1>
tag	<b>Tag</b> <tag>
.myClass	<b>Klasse</b> myClass
#myId	<b>ID</b> myId
[foo]	<b>Attribut</b> foo
[foo=bar]	Attribut foo hat Value bar
[foo~=bar]	foo enthält bar
[foo^=bar]	foo beginnt mit bar
[foo\$=bar]	foo endet mit bar
[foo*=bar]	foo enthält bar

## Kombinatoren

Kombi	Selektion: alle Elemente
E F	alle Elemente F, die <b>Nachfahren</b> eines Elements E sind
E > F	alle Elemente F, die ein <b>direktes Kind</b> eines Elements E sind
E ~ F	alle Elemente F, die einen <b>Vorgänger E auf gleicher Ebene</b> haben
E + F	alle Elemente F, die einen <b>direkten Vorgänger E auf gleicher Ebene</b> haben

# CSS-Selektoren (*Fortsetzung*)

## Pseudoklassen

Selektor	Selektion: alle Elemente mit ...
:link	noch nicht besuchte Links
:visited	bereits besuchte Links
:active	das aktiv gerade angeklickte Element
:hover	Cursor über Element
:focus	Element im Fokus
:enabled	auswählbar
:checked	ausgewählt
:not(x)	x trifft nicht zu

## strukturelle Pseudoklassen

Selektor	Selektion: alle Elemente mit ...
	...
:first-child	erstes Kind
:last-child	letztes Kind
:nth-child(3)	drittes Kind
:nth-last-child(3)	dritt-letztes Kind

# Vollständige Liste aller CSS-Selektoren Level 3

## einfache Selektoren

- Universalselektor: \*
- Typselektoren: element
- Klassenselektoren: .class
- ID-Selektoren: #id
- Attributselektoren
  - Attributpräsenz: [a]
  - Attributwert: [a=v], [a~v], [a|=v]
  - Teilübereinstimmung: [a^=v], [a\$=v], [a\*=v]

## Kombinatoren

- Kindselektor: e > f
- Nachfahrenselektor: e f
- Nachbarselektor: e + f
  - The lobotomized owl selector: \* + \*
- Geschwisterselektor: e ~ f

## Pseudoelemente

- ::first-line
- ::first-letter
- ::before, ::after
- ::backdrop
- ::selection

## Pseudoklassen

- strukturelle Pseudoklassen
  - :root
  - :empty
  - :first-child
  - :last-child
  - :nth-child()
  - :nth-last-child()
  - :only-child
  - :first-of-type
  - :last-of-type
  - :nth-of-type()
  - :nth-last-of-type()
  - :only-of-type
- dynamische Pseudoklassen
  - :any-link, :link, :visited
  - :hover, :active, :focus
  - :target
  - :disabled, :enabled, :checked
  - :valid, :invalid
  - :in-range, :out-of-range



# CSS-Selektoren Level 4

# Selectors Level 4

Editor's Draft, 21 March 2018

<https://drafts.csswg.org/selectors-4/>

```
/* The following selector matches all p elements immediately followed by an img element. */



p:has(+img) {
color: red;
}



/* The following selector matches all section elements which contain h1, h2, and/or h3 elements. */

section:has(h1, h2, h3) {
background-color: green;
}

/* The following selector matches anything containing a span element with a class urgent. */

:has(span.urgent) {
background-color: red;
}
```

<http://css4.rocks/>



# caniuse css has

:has() CSS relational pseudo-class WD

Only select elements containing specified content. For example, `a:has(>img)` selects all `<a>` elements that contain an `<img>` child.

Usage % of all users Global 0%

IE	Edge	Firefox	Chrome	Safari	Opera	iOS Safari	Opera Mini	Android Browser	Opera Mobile	Chrome for Android	Firefox for Android	UC Browser for Android	Samsung Internet	QQ Browser	Baidu Brow
6-10	12-17	2-68	4-76	3.1-12.1	10-60	3.2-12.3		2.1-4.4.4	12-12.1				4-9.2		
11	18	69	77	13	62	13.1	all	76	46	76	68	12.12	10.1	1.2	7.1
		76	70-71	78-80	13.1-TP										

Notes Known issues (0) Resources (4) Feedback

- Specification [drafts.csswg.org] ref
- Firefox support bug [bugzilla.mozilla.org] info
- MDN Web Docs - :has [developer.mozilla.org] ref info
- IE/Edge platform status: Under Consideration [developer.microsoft.com] official status

# Aufgabe mit Selektoren

Schreiben Sie einen Selektor zur Selektion des folgenden Elementes:

```
<!doctype html>
<html>
<head>
<meta charset="utf-8">
<meta name="viewport" content="w
<style>
  selector {
    border: thin solid black;
    background-color: red;
  }
</style>
</head>
```

```
<body>
  <h1>Finden Sie den richtigen CSS-Selektor, um ...</h1>
  <article>
    <p>Nein, nicht dieses!</p>
    <ul>
      <li>Nein, nicht dieses!
      <li>Nein, nicht dieses!
        <ul>
          <li>Nein, nicht dieses!
          <li>
            ... dieses Element hier zu selektieren
          </li>
        Nein, nicht dieses!
        </ul>
      </li>
    </ul>
  </article>
  <p>Nein, nicht dieses!</p>
</body>
</html>
```



# Spezifität = Prioritätsregeln in CSS

Je spezifischer, desto höher die Priorität

- A = ID-Selektor
- B = Anzahl der Attribut-, Klassenselektoren und Pseudoklassen
- C = Anzahl der Typselektoren und Pseudoelemente
- **!important** höchste Prio

Beispiel: - Spezifität von Selektoren von gering nach hoch geordnet

*	/* A=0, B=0, C=0, Spezifität 0 0 0 */
h1	/* A=0, B=0, C=1, Spezifität 0 0 1 */
ul li	/* A=0, B=0, C=2, Spezifität 0 0 2 */
a::after	/* A=0, B=0, C=2, Spezifität 0 0 2 */
p:first-child	/* A=0, B=1, C=1, Spezifität 0 1 1 */
a:not([href])	/* A=0, B=1, C=1, Spezifität 0 1 1 */
ul.nav [href]	/* A=0, B=2, C=1, Spezifität 0 2 1 */
#author	/* A=1, B=0, C=0, Spezifität 1 0 0 */
#editor p	/* A=1, B=0, C=1, Spezifität 1 0 1 */

```
h1 { color: green !important; }  
h1 { color: red; }
```

<https://wiki.selfhtml.org/wiki/Spezifität>

body #content .alternative p has a specificity of **112** (HTML selector + id selector + class selector + HTML selector, 1+100+10+1)

<http://www.htmldog.com/guides/css/intermediate/specification/>

# CSS Specificity Calculator

## Specificity Calculator

A visual way to understand *CSS specificity*. Change the selectors or paste in your own.

`li:first-child h2 .title`

0

Inline styles

0

IDs

2

Classes, attributes  
and pseudo-classes

2

Elements and  
pseudo-elements

+ Duplicate

`#nav .selected > a:hover`

0

Inline styles

1

IDs

2

Classes, attributes  
and pseudo-classes

1

Elements and  
pseudo-elements

+ Duplicate

<https://specificity.keegan.st/>

# Aufgaben zur Spezifität von CSS-Selektoren

Selektor	Spezifität
body #content .alternative p	112
li:first-child h2 .title	22
ul ul li :nth-child(2)	4
div > div > ul > li + li	5
section > .my > ul > li .li	23

# Wie lernt man CSS?



<https://wiki.selfhtml.org/wiki/CSS>

The screenshot shows the w3schools.com website. The top navigation bar includes links for Home, HTML, CSS (which is highlighted), JAVASCRIPT, SQL, PHP, BOOTSTRAP, JQUERY, and MORE. Below the navigation is a sidebar with a 'CSS Tutorial' section containing links to CSS HOME, CSS Introduction, CSS Syntax, CSS How To, CSS Colors, CSS Backgrounds, CSS Borders, CSS Margins, CSS Padding, CSS Height/Width, CSS Box Model, CSS Outline, and CSS Text. The main content area features a 'Code-free prototyping tool' banner from Infragistics. The 'CSS Tutorial' page itself has a heading 'CSS Tutorial', a 'Home' link, and three text snippets: 'CSS is a language that describes the style...', 'CSS describes how HTML elements should...', and 'This tutorial will teach you CSS from basic...'. To the right, a 'TryIt-Editor' window is open in a browser, showing a code editor with the following HTML and CSS:

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

<h2>Color Names Examples</h2>
<p>Note: You will learn more about the background-color and the color property later in our tutorial.</p>

<h2 style="background-color:red">
Red background-color
</h2>

<h2 style="background-color:green">
Green background-color
</h2>

<h2 style="background-color:blue">
Blue background-color and white
text color
</h2>

<h2 style="background-color:orange">
Orange background-color
</h2>

</body>
</html>
```

The editor interface includes tabs for Home, Run, and Stop seeing this ad. On the right, there's a 'Color Names Examples' section with examples for Red, Green, Blue, and Orange background colors.

<https://www.w3schools.com/css/default.asp>

weitere Tutorial-Angebote am Ende ...

WE\_BCS - Google Dr. x 02\_CSS - Google Pra. x CSS Syntax and Selectors x CSS - SELFHTML-WI x CSS | MDN x W Cas

[Sicher https://www.w3schools.com/css/css\\_syntax.asp](#)

HTML **CSS** JAVASCRIPT SQL PHP BOOTSTRAP JQUERY ANGULAR MORE ▾

CSS Tutorial  
CSS HOME  
CSS Introduction  
**CSS Syntax**  
CSS How To  
CSS Colors  
CSS Backgrounds  
CSS Borders  
CSS Margins  
CSS Padding  
CSS Height/Width  
CSS Box Model  
CSS Outline  
CSS Text  
CSS Fonts  
CSS Icons  
CSS Links  
CSS Lists  
CSS Tables  
CSS Display  
CSS Max-width  
CSS Position  
CSS Overflow  
CSS Float  
CSS Inline-block  
CSS Align  
CSS Combinators  
CSS Pseudo-class  
CSS Pseudo-element  
CSS Opacity  
CSS Navigation Bar  
CSS Dropdowns  
CSS Tooltips  
CSS Image Gallery  
CSS Image Sprites  
CSS Attr Selectors  
CSS Forms  
CSS Counters

## CSS Syntax and Selectors

< Previous

### CSS Syntax

A CSS rule-set consists of a selector and a declaration block:

Selector	Declaration	Declaration
<b>h1</b>	<pre>{ color:blue; font-size:12px; }</pre>	
Property	Value	Property
Value		Value

The selector points to the HTML element you want to style.

The declaration block contains one or more declarations separated by semicolons.

Each declaration includes a CSS property name and a value, separated by a colon.

A CSS declaration always ends with a semicolon, and declaration blocks are surrounded by curly braces.

In the following example all `<p>` elements will be center-aligned, with a red text color:

**Example**

```
p {
    color: red;
    text-align: center;
```

CSS Tutorial
<b>CSS HOME</b>
CSS Introduction
CSS Syntax
CSS How To
CSS Colors
CSS Backgrounds
CSS Borders
CSS Margins
CSS Padding
CSS Height/Width
CSS Box Model
CSS Outline
CSS Text
CSS Fonts
CSS Icons
CSS Links
CSS Lists
CSS Tables
CSS Display
CSS Max-width
CSS Position
CSS Overflow
CSS Float
CSS Inline-block
CSS Align
CSS Combinators
CSS Pseudo-class
CSS Pseudo-element
CSS Opacity
CSS Navigation Bar
CSS Dropdowns
CSS Tooltips
CSS Image Gallery
CSS Image Sprites
CSS Attr Selectors
CSS Forms
CSS Counters

## CSS3

CSS3 Introduction
CSS3 Rounded Corners
CSS3 Border Images
CSS3 Backgrounds
CSS3 Colors
CSS3 Gradients
CSS3 Shadows
CSS3 Text
CSS3 Fonts
CSS3 2D Transforms
CSS3 3D Transforms
CSS3 Transitions
CSS3 Animations
CSS3 Images
CSS3 Buttons
CSS3 Pagination
CSS3 Multiple Columns
CSS3 User Interface
CSS3 Box Sizing
CSS3 Flexbox
CSS3 Media Queries
CSS3 MQ Examples

## CSS Responsive

RWD Intro
RWD Viewport
RWD Grid View
RWD Media Queries
RWD Images
RWD Videos
RWD Frameworks
RWD Templates
CSS Examples
CSS Examples
CSS Quiz
CSS Exercises
CSS Certificate
CSS References
CSS Reference
CSS Selectors
CSS Functions
CSS Reference Aural
CSS Web Safe Fonts
CSS Animatable
CSS Units
CSS PX-EM Converter
CSS Colors
CSS Color Values
CSS Default Values
CSS3 Browser Support

# CSS Tricks

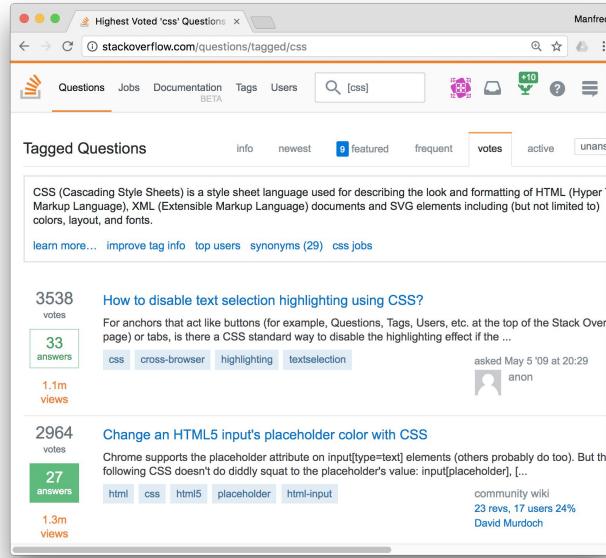


The screenshot shows the CSS-Tricks website's "Snippets" section. On the left is a vertical sidebar with icons for Home, Back, Forward, File, Image, Snippets (highlighted), Chat, and Email. The main content area has a blue header with the word "Snippets" and the tagline "Little useful bits.". A sidebar on the right lists categories: HTML, CSS (highlighted), Sass, SVG, htaccess, and JavaScript. To the right of each category is a snippet title.

Category	Snippet Title
HTML	"Shake" CSS Keyframe Animation
CSS	"Stitched" Look
Sass	A Complete Guide to Flexbox
SVG	A Complete Guide to Grid
htaccess	Absolute Center (Vertical & Horizontal) an Image
JavaScript	Accessibility/SEO Friendly CSS Hiding

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

# Wo bekommt man Antworten?



A screenshot of a web browser window titled "Highest Voted 'css' Questions". The URL in the address bar is "stackoverflow.com/questions/tagged/css". The page shows a list of tagged questions under the heading "Tagged Questions". The first question is "How to disable text selection highlighting using CSS?", which has 3538 votes, 33 answers, and 1.1m views. The second question is "Change an HTML5 input's placeholder color with CSS", which has 2964 votes, 27 answers, and 1.3m views.



<http://stackoverflow.com/questions/tagged/css>