

## HTML (Hypertext Markup Language)

- Auszeichnungssprache für Inhalte des Internets
- Auszeichnung von Textelementen durch Tags `<tag>text</tag>`

### Skeleton

```
<!DOCTYPE html>
<html lang="de">
  <head> <!--Metadaten für Browser, Suchmaschinen, ... -->
    <!-- Typography --> <b><i><u><sub>↓<sup>↑<mark><del><ins></*
    <meta charset="UTF-8"> <!-- author, description, keywords, ... -->
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Cheatsheet</title>
  </head>
  <body></body> <!--Inhalt der Webseite -->
</html>
```

### Grundlegende HTML-Tags

```
<div>Bereich</div><p>Absatz<span>spez. form. Textbereich</span></p>
<!-- Typography --> <b><i><u><sub>↓<sup>↑<mark><del><ins></*
<a href="/example.html#id" target="_blank" title="tooltip">Link</a>

<ol><li>itemA</li><li value="4">itemD</li></ol> <!-- sortierte Liste -->
<!-- Attr.^ start=1 type=A reversed -->
<ul><li>itemA</li><li>itemB</li></ul> <!-- unsortierte Liste -->
<dl><dt>term</dt><dd>description</dd></dl> <!-- Beschreibungsliste -->
newline<br>breaking-none-space<wbr>sil&shy;ben&shy;tren&shy;nung
```

### Semantisches HTML

```
<header> <-- Seitenkopf (Titel, Logo, Navigation,...)
<nav> <-- Navigationslinks
<main> <-- Hauptinhalt (max. 1x pro Seite)
<aside> <-- ergänzende Inhalte
<footer> <-- Seitenende (Kontakt, Impressum, Copyright, Sitemap,...)
<article> <-- in sich geschlossener Inhalt (eigener header, footer mögl.)
<section> <-- Abschnitt eines Inhaltes (z.B. in article)
```

### Tabellen

```
<table> <!-- in Tabelle strukturierte Daten -->
  <caption>Title</caption> <!-- zentrierter Titel -->
  <thead> <!-- Tabellen-Kopf -->
    <tr> <!-- Zeile -->
      <th>Time</th> <!-- Kopfzelle -->
      <th>Subject</th> <!-- another Kopfzelle -->
    </tr>
  </thead>
  <tbody> <!-- Tabellen-Körper -->
    <tr><th>08:00</th><td rowspan=2>Web</td></tr>
    <tr><th>09:45</th></tr> <!-- td^ normale Zelle -->
  </tbody> <!-- row/colspan um Zellen zu verbinden -->
  <tfoot> <!-- Tabellen-Fuß -->
    <tr><td colspan="2">Footer</td></tr>
  </tfoot>
</table>
```

### Formulare

```
<form action="index.html" method="post" enctype="multipart/form-data">
  <fieldset>
    <legend>Allgemein</legend>
    <label for="name">Name:</label>
    <input id="name" type="text" name="name" maxlength="20" required>
    <label for="age">Alter:</label>
    <input type="number" id="age" name="age" min="0" required>
  </fieldset><br>
  <input type="submit" value="Senden" name="send">
  <input type="reset" value="Zurücksetzen" name="reset">
</form>
```

- Input-Types: password,email,color,range,radiobutton,checkbox,file
- Input-Attr: name,minlength,pattern,value,placeholder,min,step

### Sonstige Inputs

```
<select name="select" multiple> <-- Auswahlliste (Dropdown)
  <option value="A">A!</option><option value="B">B!</option></select>
<textarea cols="80" rows="15" name="notes"></...> <-- mehrzeilige Textbox
```

### CSS einbinden

```
style="..." <-- inline CSS-Attribute
<style></style> <-- HTML-Style tag
<link rel="stylesheet" type="text/css" href="style.css"> <-- ext. file(h)
```

### JS einbinden

```
<script></script> <-- HTML-Script tag
<script src="script.js"> <-- ext. file (mögl. Ende des Bodys)
```

## JS ECMA-Script=Standard → Javascript=Implementation des Standards; Engines: V8, SpiderMonkey

- Variablen: const, let; Operatoren wie in C (\*\* für Potenzieren); === → Vergleich mit Typ-Prüfung
- Datentypen: String, Number, Boolean, undefined, ... ; komplexe Typen: Array, Object, function
- DOM: Schnittstelle zwischen HTML und JS → abstrakte Repräsentation der HTML-Objektstruktur
- DOM-Manipulation: elem = document.querySelector(); {innerText,innerHTML,parentNode,src,href,classList,style}=elem; document.addEventListener("DOMContentLoaded", () => {});

## CSS (Cascading Style Sheets)

### Selektoren (nach Spezifität)

- \* → Universalselektor
- tag → Typselektor (HTML-Tag)
- .class → Klassenselektor
- #id → ID-Selektor

### :Pseudoklassen

- :empty,:first-child,:last-child,:nth-child(mx+n)
- :hover,:link,:visited,:active,:focus
- :disabled,:enabled,:checked
- :valid,:invalid,:optional,:in-range,:out-of-range

### ::Pseudoelemente

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

### Verbundselektoren

- Kindkombinator: elem > kind (eine Ebene darunter)
- Nachfahrenkombinator: elem kind (belieb. Ebene darunter)
- Nachbarkombinator: elem + nachfolger
- Geschwisterkombinator: bruder ~ schwester

### Einheiten

- physisch: in,cm,px Winkel: deg,rad Zeit: s,ms
- relativ: em&rem, vw&vh, vmin&vmax, %, fr

Farben: rgb(100%,0%,0%); rgba(255,0,128,0.5); #ff0080

### Display

- inline: kein Zeilenumbruch, Breite durch Inhalt bestimmt
- block: Zeilenumbruch (gesamte Breite) → untereinander
- flex: eindimensionales Layoutmodell (Zeile, Spalte)
- grid: zweidimensionales Layoutmodell (vgl. Tabellen)

### Display:flex

- Hauptachse: flex-direction: (row|col)[-reverse]
- Anordnung-Hauptachse: justify-content: center|space-(between|around)|flex-(start|end)
- Anordnung-Nebenachse: align-items: center|stretch|flex-(start|end)
- Line-Wrap: flex-wrap: wrap|no-wrap|wrap-reverse
- Gesamt-Anord. Nebenachse bei Line-Wrap: align-content

### Display:grid

- grid-template-(columns|rows): Zeilen/Spalten und deren Aufteilung
- grid-template-areas: Zellen-Bereich-Zuordnung
- grid-area: Element-Bereich-Zuordnung

body{

```
  display: grid;
  grid-template-columns: 1fr 2fr 2fr 1fr;
  grid-template-rows: 2fr 4fr 4fr 1fr;
  grid-template-areas: "head head head head"
                      "nav main main aside"
                      "nav main main aside"
                      "foot foot foot foot";
}
```

header{ grid-area: head; }

### Media Queries (untersch. Styles für versch. Ausgabemedien)

```
@media (max-width: 500px) {/* css for small screens */}
@media print {/* css for print layout */}
```

### Random CSS-Attributes

...in no particular order  
text-decoration: none|(under|over)line|line-through red;  
margin: t r b l; padding: y x; color: #f00; z-index: 0;  
border: xy solid|dotted|dashed red; border-radius: 0;  
text-align: center|start|justify|right; opacity: .4;  
font-family: "Comic Sans MS", sans-serif; font-size: 2em;  
background: blue url("test.jpg") repeat-xy;  
visibility: visible|hidden; cursor: auto|pointer|...;

