

HTML

Was ist HTML?

HTML (Hypertext Markup Language) ist fundamentaler Bestandteil jeder Website.

HTML ist eine Sprache, aber *keine Programmiersprache*. Es gibt keine Variablen, Logik und Funktionen.

Stattdessen ist es eine **Auszeichnungssprache**, die spitze Klammern verwendet, um die Struktur einer Webseite semantisch zu definieren.

Webbrowser interpretieren den HTML-Code und rendern ihn (ggf. leicht unterschiedlich).

Einstiegsbeispiel

```
<html>

  <head>
    <title>
      Hallo Welt
    </title>
  </head>

  <body>
    Welt, hallo
  </body>

</html>
```

Einstiegsbeispiel

```
<html>

  <head>
    <title>
      Hallo Welt
    </title>
  </head>

  <body>
    Welt, hallo
  </body>

</html>
```

Einstiegsbeispiel

```
<html>

  <head>
    <title>
      Hallo Welt
    </title>
  </head>

  <body>
    Welt, hallo
  </body>

</html>
```

Einrückungen sind optional, aber sehr hilfreich

```
<html><head><title>Hallo  
Welt</title></head><body>Welt, hallo</body></html>
```

Ohne Einrückungen: Schlecht lesbar!

Minifier sind Tools, die alle Einrückungen und Zeilenumbrüche entfernen, um die zu übertragende Datenmenge zu reduzieren.

Viele Editoren können Code (auch nachträglich) hierarchisch eingerückt formatieren.

Vollständiges Beispiel mit empfohlenen Ergänzungen

```
<!DOCTYPE html>
```

```
<html lang="de">
```



Sprache des Dokuments
für Screenreader und
Silbentrennung

```
<head>
```

```
<meta charset="UTF-8">
```

```
<title>Hallo Welt</title>
```

```
</head>
```

```
<body>
```

```
Welt, hallo
```

```
</body>
```

```
</html>
```

Vollständiges Beispiel mit empfohlenen Ergänzungen

```
<!DOCTYPE html>  
<html lang="de">
```

```
  <head>
```

```
    <meta charset="UTF-8">
```

```
    <title>Hallo Welt</title>
```

```
  </head>
```

```
  <body>
```

```
    Welt, hallo
```

```
  </body>
```

```
</html>
```

← Angabe des String-Encodings der Datei:
vermeidet Darstellungsfehler; muss dem
Encoding entsprechen, mit dem die Datei
gespeichert wurde (UTF-8, ISO-8859-1, ...)

Vollständiges Beispiel mit empfohlenen Ergänzungen

```
<!DOCTYPE html>
<html lang="de">

  <head>
    <meta charset="UTF-8">
    <title>Hallo Welt</title>
  </head>

  <body>
    Welt, hallo
  </body>

</html>
```

Beispiel für Darstellungsfehler

Mit richtigem charset="UTF-8":
Schöne Grüße aus Köln! 🙌

Ohne Hinweise zum Encoding
kann es zu "Mojibake" kommen:
SchÃ¶ne GrÃ¼Ãe aus KÃ¶ln! ð

Umfangreiche Möglichkeiten

Die **HTML-Tags** (Auszeichnungen) ermöglichen es uns, Informationen über den Inhalt zu vermitteln.

Es gibt über 100 verschiedene HTML-Elemente und viele Online-Ressourcen, um diese zu finden (z.B. *selfhtml.org*).

Wir behandeln hier nur eine sehr kleine Auswahl.

Lernen durch Abschauen: Im Browser kann man den HTML-Code mit dem Shortcut **Strg+U** (**Cmd+Option+U** in macOS) ansehen.

Noch mehr Möglichkeiten: **Entwickler-Tools** (**F12** / **Strg+Shift+I** / **Cmd+Option+I**).

Tags zur Hervorhebung

`...` und `...`

Text zwischen diesen Tags wird **fett** dargestellt.

`<i>...</i>` und `...`

Text zwischen diesen Tags wird *kursiv* dargestellt.

Semantische vs. typografische Auszeichnung

`` und `` markieren die Bedeutung (Betonung),
`` und `<i>` markieren hingegen nur das Aussehen.

Tags zur Strukturierung

`<p>...</p>`

Text wird als Absatz dargestellt,
mit Abstand davor und danach.

`<hX>...</hX>`

X = 1, 2, 3, 4, 5 oder 6

Text wird als Überschrift der
X-ten Ebene dargestellt.

`<h1>` ist die wichtigste Überschrift,
`<h6>` die unwichtigste.

Tags zur Strukturierung: Beispiel

`<h1>Ein Text</h1>`

`<p>`

Das ist ein Text mit mehreren Sätzen.

`</p>`

`<h2>Abschnitt 1: Absätze</h2>`

`<p>`

Dieses Beispiel zeigt:

Man kann mehrere Sätze in einen Absatz schreiben.

`</p>`

`<p>`

Zeilenumbrüche spielen dabei keine Rolle.

`</p>`

Ein Text

Das ist ein Text mit mehreren Sätzen.

Abschnitt 1: Absätze

Dieses Beispiel zeigt: Man kann mehrere Sätze in einen Absatz schreiben.

Zeilenumbrüche spielen dabei keine Rolle.

Tags für Listen

`...`

Begrenzt eine ungeordnete
Aufzählungsliste (Bullet Points)

`...`

Begrenzt eine geordnete
(nummerierte) Liste

`...`

Markiert einzelne Listenelemente
innerhalb einer Liste

Tags für Listen: Verschachtelte Listen

```
<ul>
  <li>Programmiersprachen</li>
  <ul>
    <li>Markup-Sprachen
      <ul>
        <li>HTML</li>
        <li>XML</li>
      </ul>
    </li>
    <li>Echte Programmiersprachen</li>
    <ul>
      <li>Python</li>
      <li>C</li>
    </ul>
  </li>
</ul>
```

- Programmiersprachen
 - Markup-Sprachen
 - HTML
 - XML
 - Echte Programmiersprachen
 - Python
 - C

Tags für Tabellen

`<table>...</table>`

Begrenzt eine Tabellendefinition

`<tr>...</tr>`

Begrenzt eine Tabellenzeile (*table row*)

`<td>...</td>`

Begrenzt eine Tabellenzelle (*table data*)

Tags für Tabellen: Beispiel

```
<table>  
  <tr>  
    <td>1</td>  
    <td>2</td>  
    <td>3</td>  
  </tr>  
  <tr>  
    <td>4</td>  
    <td>5</td>  
    <td>6</td>  
  </tr>  
</table>
```

1	2	3
4	5	6

Tags für Formulare

`<form>...</form>`

Begrenzt ein HTML-Formular

`<input name=X type=Y>`

Definiert ein Eingabefeld

X ist ein eindeutiger Name für das Feld

Y bestimmt die Art der Eingabe:

text (Default), *password*, *checkbox*, *submit*, ...

Tags für Formulare

```
<form>
  <div>
    Benutzername:
    <input name="username" type="text">
  </div>
  <div>
    Passwort:
    <input name="password" type="password">
  <div>
    Angemeldet bleiben?
    <input name="remember" type="checkbox">
    <input name="login" type="submit">
  </div>
</form>
```

Benutzername:

Passwort:

Angemeldet bleiben? ☐

Tags für Formulare

```
<form>
  <div> ← Universeller Container
    Benutzername:
    <input name="username" type="text">
  </div>
  <div>
    Passwort:
    <input name="password" type="password">
    <div>
      Angemeldet bleiben?
      <input name="remember" type="checkbox">
      <input name="login" type="submit">
    </div>
  </div>
</form>
```

Tags für Formulare

```
<form>
  <div>
    Benutzername:
    <input name="username" type="text">
  </div>
  <div>
    Passwort:
    <input name="password" type="password">
    <div>
      Angemeldet bleiben?
      <input name="remember" type="checkbox">
      <input name="login" type="submit">
    </div>
  </div>
</form>
```

Früher gebräuchlich (heute unnötig):
↓
<input name="username" type="text" />

Tags für Formulare

```
<form>
  <div>
    Benutzername:
    <input name="username" type="text">
  </div>
  <div>
    Passwort:
    <input name="password" type="password">
    <div>
      Angemeldet bleiben?
      <input name="remember" type="checkbox">
      <input name="login_button" type="submit">
    </div>
  </div>
</form>
```

Tags für Formulare: weitere Details

```
<form action="/login" method="post">
```

action: Wohin gehen die Daten?

method: Wie werden sie gesendet?

- get: Daten in URL sichtbar übertragen
- post: Daten im Hintergrund übertragen

Tags für besser benutzbare Formulare

```
<form action="/submit" method="post">
  <div>
    <label for="name">Name:</label>
    <input type="text" id="name" name="name">
  </div>
  <div>
    <label for="email">E-Mail:</label>
    <input type="email" id="email" name="email">
  </div>
  <button type="submit">Absenden</button>
</form>
```


Tags für besser benutzbare Formulare

```
<form action="/submit" method="post">
  <div>
    <label for="name">Name:</label>
    <input type="text" id="name" name="name">
  </div>
  <div>
    <label for="email">E-Mail:</label>
    <input type="email" id="email" name="email">
  </div>
  <button type="submit">Absenden</button>
</form>
```

Tags für Links und Bilder

```
<a href="X">...</a>
```

Erzeugt einen Hyperlink zur Webseite X

Text zwischen den Tags wird als klickbarer Link dargestellt

```

```

Fügt ein Bild ein, das unter Adresse X abrufbar ist

Weitere Attribute wie *width*, *height*, *alt* sind möglich

Tags für Links und Bilder

```
<a href="https://www.example.com">Besuchen Sie example.com!</a>  

```



[Besuchen Sie example.com!](https://www.example.com)

Vollständige URLs (Uniform Resource Locator) bestehen aus Protokoll, Domain und optionalen Elemente, etwa dem Pfad:

```
<a href="https://www.example.com/">Homepage</a>  
<a href="https://www.example.com/seite.html">Eine Unterseite</a>  

```

↑
Protokoll

↑
Domain

↑
Pfad

Relative URLs (beginnen *nicht* mit /)
sind relativ zur URL der aktuellen Seite

`verlinkt auf Seite in Unterordner im akt. Verzeichnis`

`Ein Ordner höher (doppelter Punkt)`

``

``

Anwendung:

URL der aktuellen Seite `https://www.example.com/seite.html`

`bilder/foto.jpg` entspricht

`https://www.example.com/bilder/foto.jpg`

URL der aktuellen Seite `https://www.example.com/static/seite.html`

`bilder/foto.jpg` entspricht

`https://www.example.com/static/bilder/foto.jpg`

`../seite.html` entspricht

`https://www.example.com/seite.html`

Absolute URLs (beginnen mit /):

aktuelles Protokoll und Domain wird autom. vorne angefügt

```
<a href="/">Homepage</a>
```

```
<a href="/login.html">Eine Datei im obersten Verzeichnis</a>
```

```

```

Anwendung:

URL der aktuellen Seite z.B. `https://www.example.com/seite.html`

`/login.html` entspricht

`https://www.example.com/login.html`

URL der aktuellen Seite z.B. `https://www.example.com/static/seite.html`

`/img/cat.jpg` entspricht

`https://www.example.com/img/cat.jpg`

Weitere Details zum Dateianfang und Header

```
<!DOCTYPE html>
```

Teilt dem Browser mit, dass HTML5-Standard verwendet wird

Es gibt weitere Meta-Tags für den Header, z.B.

```
<head>
```

```
  <meta charset="UTF-8">
```

```
  <meta name="description" content="Kursbeschreibung: HTML-Grundlagen">
```

```
  <meta name="viewport" content="width=device-width, initial-scale=1">
```

```
  <title>HTML-Kurs</title>
```

```
</head>
```

Kommentare in HTML und Wohlgeformtes HTML

`<!--` Kommentare werden durch Start- und Endmarkierungen begrenzt.
Sie können aus einer oder mehreren Zeilen bestehen. `-->`

Wohlgeformtes HTML

Jeder geöffnete Tag muss in der richtigen Reihenfolge geschlossen werden
(Ausnahme: einige Tags wie *input*, *img*, *br*, *hr* brauchen keinen schließenden Tag)

```
<!-- Dies ist ein Kommentar -->
<div>
  <p>Dies ist <strong>sehr</strong> wichtig!</p>
  <!-- Richtig geschachtelt -->
</div>
```

```
<!-- Nicht wohlgeformt: -->
<p><strong>falsch</p></strong>
```


EXTRAS IN 3 MINUTEN
FRAGEN – ANTWORTEN – RÄTSEL
UND KURZE ZUSAMMENFASSUNG

```
<!DOCTYPE html>
<html>
  <head>
    <title>Meine Rezeptsammlung</title>
  </head>
  <body>
    <h1>Lieblingskuchen</h1>
    <div>
      
      <p>Ein <strong>fantastisches</strong>
        Rezept von Oma!</p>
    </div>
    <h2>Zutaten</h2>
    <ul>
      <li>300g Mehl</li>
      <li>2 Eier</li>
      <li>200g Schokolade</li>
    </ul>
    <a href="rezepte.html">Weitere Rezepte</a>
  </body>
</html>
```

alt-Text wird angezeigt →
da Bild nicht abrufbar

Lieblingskuchen

Schokoladenkuchen

Ein **fantastisches** Rezept von Oma!

Zutaten

- 300g Mehl
- 2 Eier
- 200g Schokolade

[Weitere Rezepte](#)

Aufgabe 1

Vervollständigen Sie den Code für diesen Blog-Eintrag.

```
<!-- Fügen Sie vorher und nachher alles ein, was fehlt -->  
<h1>Mein Blog</h1>  
<div>  
  <!-- Fügen Sie hier einen Artikel mit Überschrift,  
        Datum und zwei Absätzen Fließtext ein.  
        Verwenden Sie dafür die Tags div, h2 und p. -->  
</div>  
<!-- Fügen Sie vorher und nachher alles ein, was fehlt -->
```

Aufgabe 1

Mögliche Lösung

```
<!DOCTYPE html>
<html lang="de">
<head>
  <meta charset="UTF-8">
  <title>Mein Blog</title>
</head>
<body>
  <h1>Mein Blog</h1>
  <div>
    <h2>Meine erste Webseite</h2>
    <div>Datum: 8. Januar 2025</div>
    <p>Heute habe ich gelernt, wie man HTML schreibt...</p>
    <p>Morgen will ich mir CSS anschauen...</p>
  </div>
</body>
</html>
```

Aufgabe 2

Ein Bild ist mit folgender vollständigen URL abrufbar.
Sie möchten das Bild in einer Webseite anzeigen, die unter der angezeigten URL abrufbar ist.

Bild:

<https://www.bank17.de/static/img/logo.png>

URL der aktuellen Webseite:

<https://www.bank17.de/static/login/login.html>

Ist das die richtige relative Adresse?

[../img/logo.png](#)

Würde auch diese relative Adresse funktionieren?

[../../static/img/logo.png](#)

Ist das die richtige absolute Adresse?

[/static/img/logo.png](#)

Aufgabe 3

Finden und korrigieren Sie alle Fehler in diesem Code!

```
<html>
<head><title>Feler finden</head></title>
<div>
<h1>Willkommen!
<p>Hier stimmt was nicht.<p>
<p>Finden Sie die Fehler!<br/>
</div>
</html>
```

Aufgabe 3

Mögliche Lösung

```
<!DOCTYPE html>
<html>
<head>
  <title>Fehler finden</title>
</head>
<body>
  <div>
    <h1>Willkommen!</h1>
    <p>Hier stimmt was nicht.</p>
    <p>Finden Sie die Fehler!</p>
  </div>
</body>
</html>
```

HTML ist eine
Auszeichnungssprache
(keine Programmiersprache)
und beschreibt die Struktur
von Webseiten.

Wichtige Tags sind
Überschriften (h1–h6),
Absätze (p), Listen (ul/ol),
Tabellen (table),
Formulare (form)
und Links (a).

HTML-Elemente werden
durch Tags in spitzen
Klammern definiert.
Elemente bestehen meist
aus öffnendem und
schließendem Tag.

HTML soll die Bedeutung
der Inhalte auszeichnen
(Semantik), aber nicht
deren Aussehen.
Für das Design ist
später CSS zuständig.

Die Grundstruktur besteht
aus `<!DOCTYPE html>`,
gefolgt von `html`-, `head`-
und `body`-Tags.
Der `head` enthält Metadaten,
der `body` den sichtbaren Inhalt.

Wohlgeformtes HTML ist
wichtig für Wartbarkeit und
Zugänglichkeit, auch wenn
Browser fehlerhaftes HTML
oft noch darstellen können.