

[zurück](#)

HTML - Elemente und Struktur

HTML besteht aus **Elementen**, die den Inhalt einer Webseite strukturieren und semantisch beschreiben.

Jedes HTML-Element hat eine klar definierte Aufgabe. Die richtige Auswahl ist entscheidend für:

- Lesbarkeit
 - Barrierefreiheit
 - Wartbarkeit
 - sauberes CSS-Layout
-

1. Aufbau eines HTML-Elements

Ein typisches HTML-Element besteht aus:

```
<tag attribut="wert">Inhalt</tag>
```

Beispiel:

```
<p class="text">Das ist ein Absatz.</p>
```

Bestandteil	Bedeutung
<p>	öffnendes Tag
class="text"	Attribut
Inhalt	Elementinhalt
</p>	schließendes Tag

2. Block- vs. Inline-Elemente

2.1 Block-Elemente

Block-Elemente:

- beginnen **immer in einer neuen Zeile**
- nehmen standardmäßig die **volle Breite** ein

Typische Block-Elemente:

Element	Zweck
<div>	allgemeiner Container
<p>	Absatz
<h1>-<h6>	Überschriften
, 	Listen
<section>	inhaltlicher Abschnitt

Beispiel:

```
<div>Block 1</div>
<div>Block 2</div>
```

2.2 Inline-Elemente

Inline-Elemente:

- bleiben **in der Textzeile**
- nehmen nur so viel Platz ein wie nötig

Typische Inline-Elemente:

Element	Zweck
	Inline-Container
<a>	Link
	Betonung
	Hervorhebung
	Bild

Beispiel:

```
<p>Das ist <strong>wichtig</strong>er Text.</p>
```

3. Warum diese Unterscheidung wichtig ist

Viele Layout-Probleme entstehen durch:

- falsche Elementwahl
- falsche Erwartungen an Inline- oder Block-Verhalten

Beispiel eines klassischen Fehlers:

```
<span>
  <div>Das funktioniert nicht sauber</div>
</span>
```

□ **Block-Elemente dürfen nicht in Inline-Elementen stecken.**

4. <div> und - richtig eingesetzt

4.1 <div>

- Block-Container **ohne eigene Bedeutung**
- Wird für Layout und Gruppierung verwendet

```
<div class="container">
  <p>Inhalt</p>
</div>
```

△ <div> ist **kein Ersatz für semantische Elemente.**

4.2

- Inline-Container **ohne eigene Bedeutung**
- Für gezielte Textauszeichnung

```
<p>Status: <span class="warnung">kritisch</span></p>
```

5. Verschachtelung (Nesting)

HTML-Elemente können ineinander verschachtelt werden:

```
<section>
  <h2>Titel</h2>
  <p>Text</p>
</section>
```

Regel:

- **Saubere Hierarchie**
- **Keine wilden Container-Kaskaden**

Je tiefer die Verschachtelung, desto schwerer:

- CSS
- Wartung
- Fehlersuche

6. Attribute - Zusatzinformationen

Attribute erweitern HTML-Elemente.

Häufige Attribute:

Attribut	Bedeutung
id	eindeutige Kennung
class	CSS-Zuweisung
href	Linkziel
src	Quelle
alt	Alternativtext

Beispiel:

```

```

7. Best Practices (bewährt, nicht modisch)

- HTML **beschreibt Bedeutung**, nicht Aussehen
- Möglichst **wenige <div>**
- Lieber korrektes Element als cleveres CSS
- Erst Struktur, dann Styling

Merksatz:

HTML ist das Skelett -CSS ist die Kleidung.

From:
<http://wiki.nctl.de/dokuwiki/> - □ **Veni. Vidi. sudo rm -rf / vici.**



Permanent link:
<http://wiki.nctl.de/dokuwiki/doku.php?id=it-themen:webdesign:html:elemente>

Last update: **27.12.2025 00:36**