

Grundgerüst als Kopiervorlage

```
<!doctype html>
<html lang="de">
  <head>
    <meta charset="utf-8">
    <meta name="viewport"
      content="width=device-width">
    <title>Seitentitel</title>
  </head>
  <body>
    <!-- Sichtbarer Inhalt -->
  </body>
</html>
```

<!doctype html> Aktiviert den standardkonformen HTML-Modus.

lang="de" Sprache des Dokuments, wichtig für Vorlesen und Übersetzung.

<head> Metadaten, Seitentitel und Verweise auf weitere Ressourcen.

<body> Sichtbarer und bedienbarer Dokumentinhalt.

Elemente, Tags und Attribute

```
<p class="hinweis">
  Elementinhalt
</p>

```

Globale Attribute: `id` ist eindeutig, `class` gruppiert Elemente, `data-*` speichert eigene Daten.

Element Start-Tag, Inhalt und End-Tag bilden zusammen ein Element.

Attribut Zusatzangabe im Start-Tag, meist als `name="wert"`.

Leeres Element Besitzt keinen Inhalt und kein End-Tag, etwa `img` oder `meta`.

Verschachtelung Elemente vollständig ineinander notieren, nicht überkreuzen.

Semantische Seitenstruktur

```
<header>...</header>
<nav aria-label="Hauptnavigation">...</nav>
<main>
  <article>
    <h1>Seitentitel</h1>
    <section>...</section>
  </article>
  <aside>...</aside>
</main>
<footer>...</footer>
```

header Einleitung eines Dokuments oder Abschnitts.

nav Bereich mit wichtigen Navigationslinks.

main Einmaliger Hauptinhalt der Seite.

article Eigenständig verständlicher Beitrag.

section Thematischer Abschnitt, meist mit Überschrift.

aside Ergänzender, nur indirekt zugehöriger Inhalt.

Text und Listen auszeichnen

Grundelemente

h1–h6 Überschriften in hierarchischer Reihenfolge.

p Textabsatz.

strong Starke inhaltliche Bedeutung.

em Betonung, die den Sinn beeinflussen kann.

br Erforderlicher Zeilenumbruch, nicht für Abstände.

```
<h2>Wunschliste</h2>
<ol>
  <li>Gesundheit</li>
  <li>Glück</li>
</ol>
<dl>
  <dt>HTML</dt>
  <dd>Auszeichnungssprache</dd>
</dl>
```

Überschriften strukturieren Inhalte; Ebenen nicht für die Optik überspringen.

Links und Medien

Link

```
<a href="preise.html">Preise</a>
```

Der Linktext beschreibt das Ziel.

Video und Audio

```
<video controls><source src="film.mp4"></video>
```

`controls` stellt Bedienelemente bereit; Untertitel können über `track` eingebunden werden.

Tabellen für tabellarische Daten

```
<table>
  <caption>Preise</caption>
  <tr>
    <th scope="col">Produkt</th>
    <th scope="col">Preis</th>
  </tr>
  <tr><td>Tasse</td><td>12 €</td></tr>
</table>
```

Formulare und Beschriftungen

```
<form action="/senden" method="post">
  <label for="email">E-Mail</label>
  <input id="email" name="email"
    type="email" autocomplete="email"
    required>
  <button>Absenden</button>
</form>
```

Interaktion und Einbindung

details

```
<details><summary>Mehr</summary>...</details>
```

Einblendbarer Abschnitt ohne eigenes JavaScript.

Skripte laden

```
<script src="app.js" defer></script>
```

`defer` wartet auf das HTML; `type="module"` lädt Module ebenfalls verzögert.

Kurze Prüfliste

- Dokumentsprache und aussagekräftigen Seitentitel angeben.
- Elemente nach Bedeutung statt nach Aussehen wählen.
- Überschriften logisch gliedern.
- Bilder, Formulare und eingebettete Inhalte beschriften.

Bild

```

```

`alt` beschreibt informative Bilder; `figure` ergänzt Beschriftungen.

Responsive Bilder

```
<img srcset="klein.jpg 480w, groß.jpg 960w"
sizes="100vw" ...>
```

`srcset` bietet Varianten an; `picture` erlaubt Formate oder Bildausschnitte.

caption Überschrift oder Beschreibung der Tabelle.

tr Tabellenzeile.

th Kopfzeile; `scope` ordnet sie Zeilen oder Spalten zu.

td Datenzeile.

Nicht verwenden: Tabellen sind kein Werkzeug für Seitenlayouts.

label Sichtbare Beschriftung; `for` verweist auf die passende `id`.

name Name, unter dem ein Wert übertragen wird.

type Passender Eingabetyp, etwa `email`, `number` oder `date`.

required Boolesches Attribut: Eingabe ist erforderlich.

button Schaltfläche zum Absenden oder Auslösen einer Aktion.

dialog

```
<dialog id="info">...</dialog>
```

Semantischer Dialog; per JavaScript öffnen und schließen.

tabindex

```
tabindex="-1"
```

Nur gezielt einsetzen; natürliche Tab-Reihenfolge bevorzugen.

Häufige Stolperstellen

- `div` und `span` nur verwenden, wenn kein passendes semantisches Element existiert.
- `id`-Werte im Dokument nicht wiederholen.
- Links nicht mit unklarem Text wie „hier“ beschriften.
- HTML mit einem Validator prüfen.

