HTML 3 bis 5 – Zusammenfassung mit IHK-Bezug

Block und Inline, Links und Bilder, Tabellen

🌐 HTML 3 bis 5 – Zusammenfassung mit IHK-Bezug

📚 Grundlage sind drei Skripte der GBS Fachschule:

- "HTML 3: Block und Inline"
- "HTML 4: Links und Bilder"
- "HTML 5: Tabellen"

Die Zusammenfassung folgt diesen drei Blöcken.

🧭 So liest du dieses Dokument

- 🔑 **Fett** gedruckte Begriffe sind Schlagwörter. Diese Begriffe solltest du erklären können.
- 🎯 Jeder Abschnitt hat eine Zeile "IHK-Relevanz". Das ist eine eigene Einschätzung. Es ist keine amtliche Angabe.
- ✏️ Am Ende jedes Blocks stehen Übungsfragen. Die Lösungen stehen gesammelt am Ende.
- 🛠️ Manche Skripte sind ungenau. Dort steht ein Hinweis "Korrektur".

ℹ️ Hinweis: "HTML 5" im Titel ist die Nummer des Skripts. Es geht dort um Tabellen. Die HTML-Version heißt **HTML5** (ohne Leerzeichen). Sie kommt in Block 1 vor.

⚖️ Ehrliche Einordnung: Wie wichtig ist HTML für die FiSi-Prüfung?

Das ist eine Einschätzung, keine Garantie:

- 📉 Webgrundlagen zählen in der FiSi-Prüfung **gering**. Die ausgewerteten Quellen nennen HTML nicht als Schwerpunkt. Höchstens kommt eine Randfrage vor.
- 🚫 Kein Prüfungsbereich verlangt, eine Webseite zu programmieren. Das ist eher Thema für Anwendungsentwickler.

👍 Trotzdem lohnt sich der Stoff als Grundwissen:

Wo HTML dir als FiSi hilft	Beispiel
📄 Skripte und Berichte (AP2 Konzeption/Administration)	Ein Skript schreibt einen Statusbericht als HTML-Tabelle, z. B. mit PowerShell `ConvertTo-Html`
🖧 Web-Oberflächen von Geräten	Router, Switches, Drucker und NAS stellt

Wo HTML dir als FiSi hilft	Beispiel
	man oft über eine Webseite ein
🔍 Fehlersuche	Quelltext einer Seite lesen, Pfade und Links prüfen
🧩 Projektarbeit	Doku-Seiten, Intranet-Anleitung oder ein kleines Dashboard
♿ Barrierefreiheit	Das `alt`-Attribut bei Bildern; laut Prüflingsforum ein mögliches AP1-Thema (nicht amtlich)

💡 Empfehlung: Begriffe verstehen. Einfachen Code lesen können. Nicht alle Tags auswendig lernen.

🧱 Block 1: Block und Inline (HTML 3)

1.1 📦 Das Grundprinzip

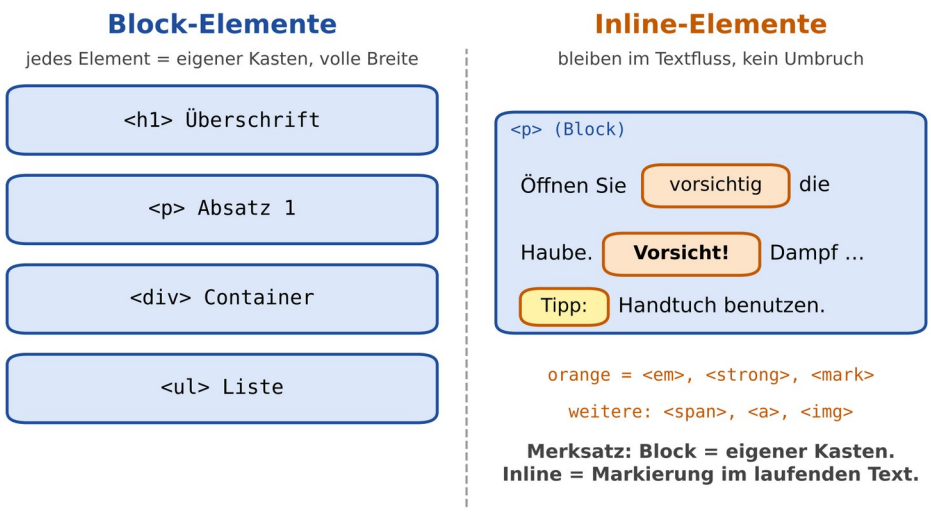
🎯 IHK-Relevanz: gering – Grundbegriff der Webtechnik; hilft beim Lesen von HTML-Berichten und Weboberflächen.

Stell dir eine Zeitungsseite vor. Es gibt ganze Kästen, z. B. eine Überschrift oder einen Absatz. Und es gibt Markierungen im Text, z. B. ein fettes Wort.

Genau so teilt HTML seine Elemente in zwei Gruppen:

- Ein Kasten mit eigener Zeile heißt **Block-Element**.
- Eine Markierung im laufenden Text heißt **Inline-Element**.

Gruppe	Verhalten	Beispiele
Block-Element	beginnt in einer neuen Zeile und endet mit einem Umbruch; bildet einen eigenen Absatz	`<p>`, `<h1>`, `<div>`, `<ul>`, `<table>`
Inline-Element	bleibt im Textfluss, kein Umbruch; markiert Text innerhalb eines Blocks	`<em>`, `<strong>`, `<span>`, `<a>`, `<img>`



Block-Elemente stehen als eigene Kästen untereinander, Inline-Elemente markieren Wörter im Text

📌 **Merksatz:** Block = eigener Kasten. Inline = Markierung im laufenden Text.

1.2 ✨ Betonung und Hervorhebung

🎯 IHK-Relevanz: gering – Detailwissen ohne Prüfungsbezug.

Im Gespräch betonst du manche Wörter lauter. In HTML gibt es dafür eigene Tags.

Tag	Bedeutung	Standarddarstellung
`<em>`	emphasis, Betonung	kursiv
`<strong>`	starke Betonung, Wichtigkeit	fett
`<mark>`	Textmarker, Hervorhebung	schwarz auf gelb

```
<p>
  Öffnen Sie <em>vorsichtig</em> die Abdeckhaube.
  <strong>Vorsicht!</strong> Es könnte heißer Dampf austreten.
  <mark>Tipp:</mark> Benutzen Sie ein sauberes Handtuch.
</p>
```

⚠️ Wichtig: Die Tags sagen, was etwas **bedeutet**. Sie sagen nicht nur, wie es aussieht. Diese Bedeutung heißt **Semantik**.

- 🔊 Ein Screenreader kann `<strong>` zum Beispiel betont vorlesen.
- 🎨 Das Aussehen legt man später mit CSS fest.

1.3 🔢 Hochstellen, Tiefstellen, klein

🎯 IHK-Relevanz: gering – Detailwissen.

Denk an die Formel für Wasser: H_2O. Die 2 steht tiefer. Bei r^2 steht die 2 höher.

Tag	Bedeutung
`<sup>`	superscript, hochgestellt
`<sub>`	subscript, tiefgestellt
`<small>`	Kleingedrucktes, z. B. Anmerkungen

```
<p>Die chemische Formel für Wasser lautet H<sub>2</sub>O.</p>
<p>
  Fläche eines Kreises: A = &#960; &times; r<sup>2</sup><br>
  <small><em>Anmerkung:</em> A = Fläche, r = Radius</small>
</p>
```

🔍 Erklärung zum Code:

- `π` steht für das Zeichen π. Das ist die Unicode-Nummer 960. So ein Code heißt **Zeichenreferenz**.
- `×` steht für das Malzeichen ×. Das ist eine benannte Zeichenreferenz.
- `<br>` erzwingt einen Zeilenumbruch. Es hat kein schließendes Tag.

🛠️ Korrektur: Das Skript nennt für `<small>` "70 %". Browser zeigen `<small>` aber eine Schriftstufe kleiner an. Der CSS-Wert dafür heißt `smaller`. Eine feste Prozentzahl legt der Standard nicht fest.

1.4 🏷️ Weitere Inline-Elemente

🎯 IHK-Relevanz: gering – Detailwissen.

Tag	Zweck	Standarddarstellung
`<cite>`	Titel eines Werks (Buch, Film, Artikel)	kursiv
`<code>`	Programmcode	Festbreitenschrift (z. B. Courier)
`<del>`	gelöschter Text	durchgestrichen
`<ins>`	eingefügter Text	unterstrichen

```
<p>Descartes' zentrale Erkenntnis: <cite>Cogito, ergo sum</cite></p>
<p>Mit <code>let</code> werden in JavaScript Variablen deklariert.</p>
<p>
  Änderungen in den AGB haben wir <del>durchgestrichen</del>,
  Hinzufügungen werden <ins>unterstrichen</ins>.
</p>
```

🛠️ Korrektur: Laut Skript markiert `<cite>` Zitate. Laut HTML-Standard markiert `<cite>` aber den **Titel eines Werks**.

- Für ein kurzes Zitat im Text gibt es `<q>`.
- Für ein langes Zitat gibt es `<blockquote>`.

1.5 🪆 Verschachteln (Nesting)

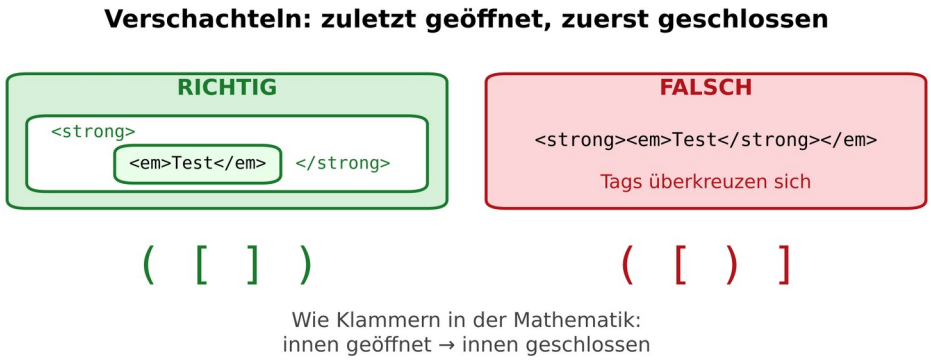
🎯 IHK-Relevanz: gering bis mittel – die Regel "zuletzt geöffnet, zuerst geschlossen" gilt auch für XML und andere Formate, die in der Administration vorkommen.

Denk an russische Puppen. Die kleine Puppe steckt ganz in der großen. Tags in Tags zu stecken heißt **Nesting** (Verschachteln).

📏 Regeln:

1. Block-Elemente dürfen Inline-Elemente enthalten.
2. Block-Elemente dürfen auch Block-Elemente enthalten, z. B. `<li>` in `<ul>`.
3. Inline-Elemente stehen in Block-Elementen. Sie dürfen Inline-Elemente enthalten. In der Regel dürfen sie keine Block-Elemente enthalten.
4. Das zuletzt geöffnete Tag wird zuerst geschlossen.

Code	Bewertung
`<strong><em>Test</em></strong>`	✅ richtig: `em` innen geöffnet, innen geschlossen
`<strong><em>Test</strong></em>`	❌ falsch: Tags überkreuzen sich



Richtig und falsch verschachtelt, verglichen mit Klammern

📌 **Merkhilfe:** Tags sind wie Klammern in der Mathematik. ([ ]) ist richtig. ([ )] ist falsch.

🛠️ Korrektur zu den Folgen: Das Skript spricht von "Fehlermeldungen". Ein Browser zeigt aber keine Fehlermeldung an.

- Der Browser repariert den Code still nach festen Regeln.
- Das Ergebnis sieht aber nicht immer so aus wie gedacht.
- Fehler meldet erst ein Prüfprogramm. Es heißt **Validator**. Ein Beispiel ist der W3C-Validator unter `validator.w3.org`.

1.6 🗃️ Neutrale Container: div und span

🎯 IHK-Relevanz: gering – nützlich zum Lesen von Webseiten-Quelltext.

Stell dir leere Umzugskartons vor. Sie haben keine Aufschrift. Sie halten nur Dinge zusammen. So arbeiten `<div>` und `<span>`.

Tag	Art	Zweck
`<div>`	Block	neutraler Container zum Gruppieren; ohne CSS keine sichtbare Wirkung außer dem Umbruch
`<span>`	Inline	neutraler Container im Text; dient als Ziel für CSS und JavaScript

```
<div>
  <p>Status des Servers: <span>online</span></p>
</div>
```

💡 Beide Tags haben keine eigene Bedeutung. Interessant werden sie erst mit CSS und JavaScript.

- 🎨 CSS regelt die Gestaltung.
- ⚙️ JavaScript regelt das Verhalten.

1.7 🏗️ Strukturelemente in HTML5

🎯 IHK-Relevanz: gering – Seitenaufbau als Grundwissen; hilft bei Barrierefreiheit, weil Screenreader die Bereiche erkennen.

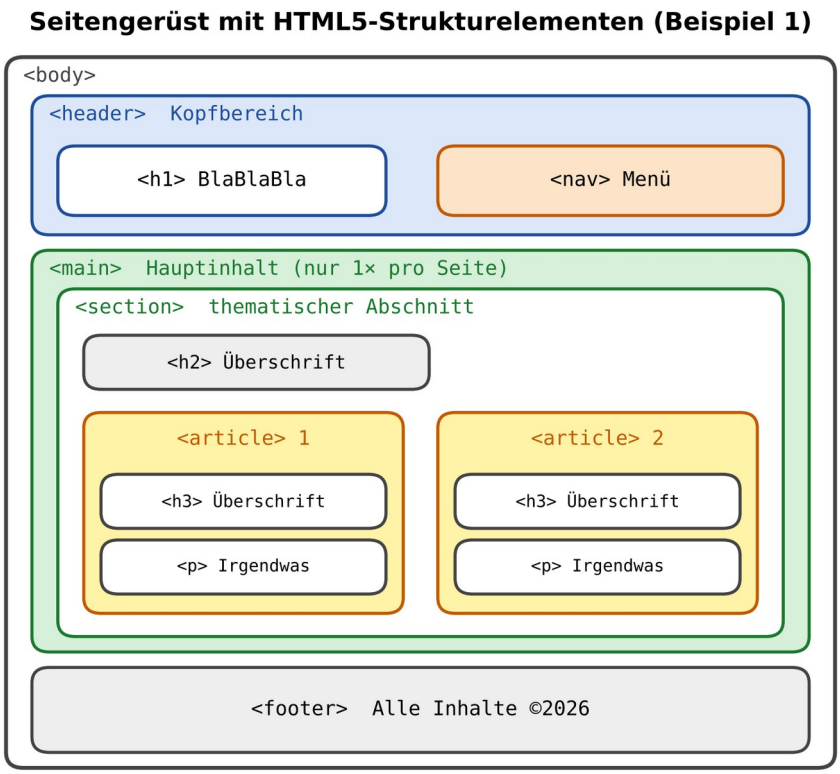
Ein Haus hat Dach, Wohnräume und Keller. Eine Webseite hat auch feste Bereiche.

Mit **HTML5** kamen Block-Elemente mit fester Bedeutung dazu. Sie gliedern die Seite besser als viele `<div>`.

Tag	Verwendung
`<header>`	Kopfbereich, z. B. Logo und Titel
`<nav>`	Navigation (Menü), oft im `<header>`
`<main>`	Hauptinhalt der Seite, nur einmal pro Seite
`<section>`	thematischer Abschnitt, meist mit Überschrift
`<article>`	abgeschlossener Beitrag, z. B. ein Blogeintrag
`<aside>`	Randbereich, z. B. Seitenleiste oder Werbung
`<footer>`	Fußbereich, z. B. Copyright und Impressum

Beispiel 1: Seitengerüst

```html
<body>
  <header>
    <h1>BlaBlaBla</h1>
    <nav>
      <!-- Das Menü -->
    </nav>
  </header>
  <main>
    <section>
      <h2>Überschrift</h2>
      <article>
        <h3>Überschrift</h3>
        <p>Irgendwas</p>
      </article>
      <article>
        <h3>Überschrift</h3>
        <p>Irgendwas</p>
      </article>
    </section>
  </main>
  <footer>
    <p>Alle Inhalte &copy;2026</p>
  </footer>
</body>
```



Aufbau von Beispiel 1 als Kastenskizze: header, main mit section und zwei article, footer

🔍 Aufbau in Worten:

1. `header` enthält die Hauptüberschrift `h1` und das Menü `nav`.
2. `main` enthält eine `section` mit Überschrift `h2`. Darin stehen zwei `article` mit Überschrift `h3`.
3. `footer` enthält das Copyright. `©` ist die Zeichenreferenz für ©.

✏️ Übungsfragen Block 1

Ü1.1 Erklären Sie den Unterschied zwischen Block- und Inline-Elementen. Nennen Sie je zwei Beispiele.

Ü1.2 Finden und korrigieren Sie den Fehler: `<p>Bitte <strong>nicht <em>ausschalten</strong></em>!</p>`

Ü1.3 Wozu dienen `<div>` und `<span>`, obwohl sie ohne CSS kaum sichtbar sind?

🔗 Block 2: Links und Bilder (HTML 4)

2.1 🏷️ Attribute

🎯 IHK-Relevanz: gering – Grundbegriff; Attribute braucht man zum Lesen jeder HTML- oder XML-Datei.

Ein Tag ist wie ein Paket. Ein Zettel daran gibt Zusatzinfos, z. B. die Adresse. Dieser Zettel heißt **Attribut**.

📏 Das Attribut steht im öffnenden Tag. Die Form ist `name="wert"`.

```
<html lang="de">
<meta charset="utf-8">
```

- `lang="de"` sagt: Die Seite ist deutsch.
- `charset="utf-8"` sagt: Der Text ist in UTF-8 codiert.

ℹ️ Leerzeichen um das = sind erlaubt. Die Skripte schreiben so: `lang = "de"`. Üblich ist die Schreibweise ohne Leerzeichen.

2.2 🔗 Links mit dem a-Tag

🎯 IHK-Relevanz: gering – Grundwissen Web; Hyperlinks machen aus Dokumenten das World Wide Web.

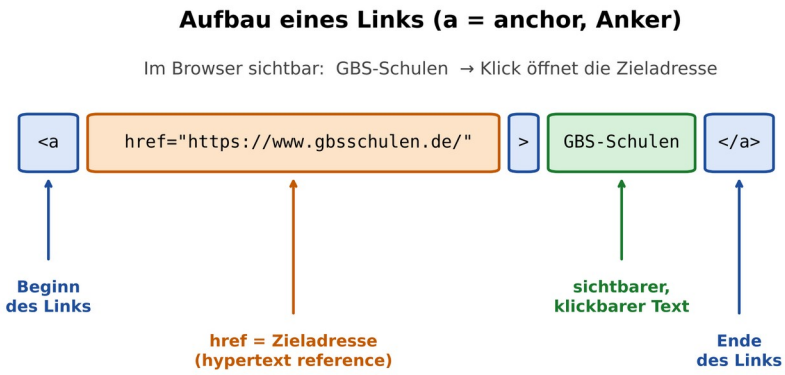
Ein Link ist wie ein Wegweiser. Ein Klick bringt dich zu einem anderen Ort. Solche Verweise heißen **Links** (Hyperlinks).

- 🕸️ Links verbinden Dokumente zu einem Netz. Dieses Netz ist das Web.
- 📁 Ein Link kann auf jede Datei zeigen: HTML-Seite, Bild, PDF, Video.

Das **<a>-Tag** umschließt den klickbaren Inhalt. “a” steht für anchor, auf Deutsch Anker. Das Ziel steht im Attribut **`href`** (hypertext reference).

```
<a href="https://www.gbsschulen.de/">GBS-Schulen</a>
```

Teil	Bedeutung
`<a ...>`	Beginn des Links
`href="https://www.gbsschulen.de/"`	Zieladresse
`GBS-Schulen`	sichtbarer, klickbarer Text
`</a>`	Ende des Links



Die Teile eines Links: Beginn, href mit Zieladresse, sichtbarer Text, Ende

2.3 🎯 Interne Links (Sprungmarken)

🎯 IHK-Relevanz: gering – Detailwissen; nützlich für lange Dokumentationsseiten.

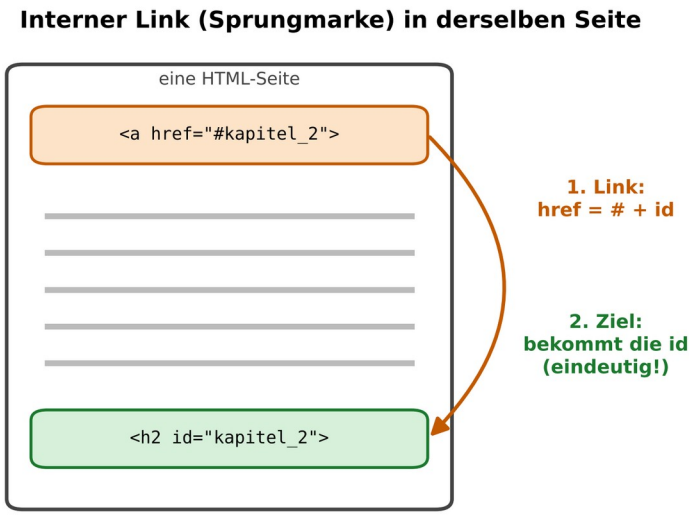
Denk an ein Inhaltsverzeichnis in einem Buch. Du springst damit direkt zu einem Kapitel. Das geht auch innerhalb einer Webseite.

📏 So geht es in zwei Schritten:

1. Das Ziel bekommt ein **`id`**-Attribut. Den Namen darfst du frei wählen. Er muss aber auf der Seite eindeutig sein.
2. Der Link bekommt als `href` das Zeichen # und die id.

```
<!-- oben auf der Seite -->
<a href="#kapitel_2">Hier geht's zum zweiten Kapitel</a>

<!-- weiter unten -->
<h2 id="kapitel_2">Kapitel 2</h2>
```



2.4 🖼️ Bilder mit dem img-Tag

🎯 IHK-Relevanz: gering bis mittel – das `alt`-Attribut ist ein Beispiel für Barrierefreiheit; laut Prüflingsforum ein mögliches AP1-Thema (Einschätzung, nicht amtlich).

```
<img src="bild.png" alt="Ein Bild" width="250" height="140">
```

Teil	Bedeutung
`img`	image; Inline-Element **ohne schließendes Tag**
`src`	source, Pfad zur Bilddatei
`alt`	alternative text: Ersatztext, wenn das Bild fehlt oder nicht gesehen wird
`width`	optional: Breite in Pixel
`height`	optional: Höhe in Pixel

❓ Warum ist `alt` wichtig?

- 🖼️ Fehlt das Bild, zeigt der Browser den Text an.
- ♿ Ein Vorleseprogramm liest blinden Menschen den Text vor. Es heißt **Screenreader**. So wird die Seite für alle nutzbar. Das nennt man **Barrierefreiheit**.
- 🔎 Suchmaschinen verstehen das Bild über den Text.

⚠️ Bilder mit `width` und `height` zu verkleinern, ist laut Skript "in aller Regel keine gute Idee".

- Grund: Ein großes Bild wird trotzdem ganz geladen. Es wird nur kleiner angezeigt.
- Besser: Das Bild vorher passend verkleinern.

2.5 📂 Pfadangaben

🎯 IHK-Relevanz: mittel – relative und absolute Pfade funktionieren in Linux und Windows genauso; das Prinzip ist prüfungsrelevant.

Ein Pfad ist wie eine Wegbeschreibung zu einer Datei. Man kann vom aktuellen Standort aus beschreiben. Oder man startet ganz oben.

- Ein Pfad ab der aktuellen Seite heißt **relativer Pfad**.
- Ein Pfad ab dem obersten Verzeichnis heißt **absoluter Pfad**. Er beginnt mit `/`.

Angabe	Bedeutung
`href="daten.html"`	Datei liegt im selben Verzeichnis wie die aktuelle Seite
`href="include/datei.html"`	Datei liegt im Unterordner `include`
`href="../daten.pdf"`	Datei liegt eine Ebene höher
`href="/index.html"`	Datei liegt im obersten Verzeichnis der Website
`href="/sonstiges/diagramm.pdf"`	Datei liegt im Ordner `sonstiges` im obersten Verzeichnis
`href="https://example.org/a.html"`	vollständige Adresse (URL) auf einem anderen Server

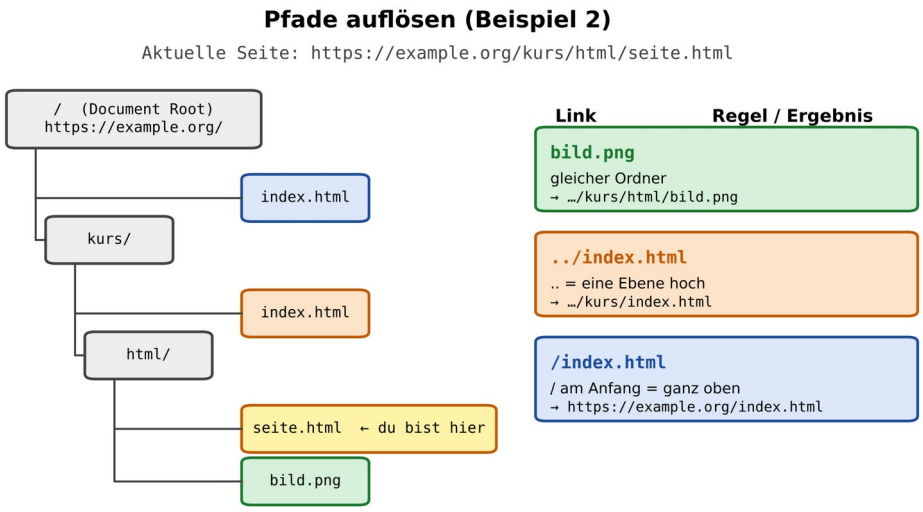
🛠️ Korrektur zum “Root-Verzeichnis”: Gemeint ist das oberste Verzeichnis der Website. Es heißt **Document Root** des Webservers.

- ⚠️ Das ist nicht das Linux-Wurzelverzeichnis `/` des Servers.
- 🐧 Beispiel: Beim Webserver Apache unter Debian liegt das Document Root standardmäßig in `/var/www/html`.

Beispiel 2: Pfad auflösen

Die aktuelle Seite ist `https://example.org/kurs/html/seite.html`.

Link	Ergebnis
`bild.png`	`https://example.org/kurs/html/bild.png`
`../index.html`	`https://example.org/kurs/index.html`
`/index.html`	`https://example.org/index.html`



Ordnerbaum zu Beispiel 2: welcher Link zu welcher Datei führt

📌 **Merksatz:** Das Prinzip ist dasselbe wie bei Linux-Pfaden. `..` geht eine Ebene hoch. Ein `/` am Anfang startet ganz oben.

✏️ Übungsfragen Block 2

Ü2.1 Schreiben Sie den HTML-Code für einen Link mit dem Text “Handbuch”, der auf die Datei `handbuch.pdf` im Unterordner `docs` zeigt.

Ü2.2 Warum sollte jedes `<img>`-Tag ein `alt`-Attribut haben? Nennen Sie zwei Gründe.

Ü2.3 Die Seite liegt unter `/projekt/doku/start.html`. Wohin zeigt `../bilder/netz.png`?

📊 Block 3: Tabellen (HTML 5)

3.1 🧮 Aufbau einer Tabelle

🎯 IHK-Relevanz: gering bis mittel – HTML-Tabellen sind das häufigste Format für automatisch erzeugte Berichte aus Skripten (AP2 Konzeption/Administration); als eigene Prüfungsfrage eher selten.

Denk an eine Excel-Tabelle oder einen Dienstplan.

- ↔ Die waagerechten Linien heißen **Reihen** (Zeilen).
- ↕ Die senkrechten Linien heißen **Spalten**.
- 🔲 Wo sie sich kreuzen, liegen die **Zellen**.

Tag	Bedeutung
`<table> … </table>`	umschließt die ganze Tabelle
`<tr> … </tr>`	table row, eine Reihe

Tag	Bedeutung
`<td> ... </td>`	table data, eine Datenzelle
`<th> ... </th>`	table header, Kopfzelle; standardmäßig fett und zentriert
`<caption>`	Beschriftung der Tabelle, meist zentriert darüber
`<thead>`, `<tbody>`, `<tfoot>`	Kopf-, Haupt- und Fußbereich für komplexere Tabellen

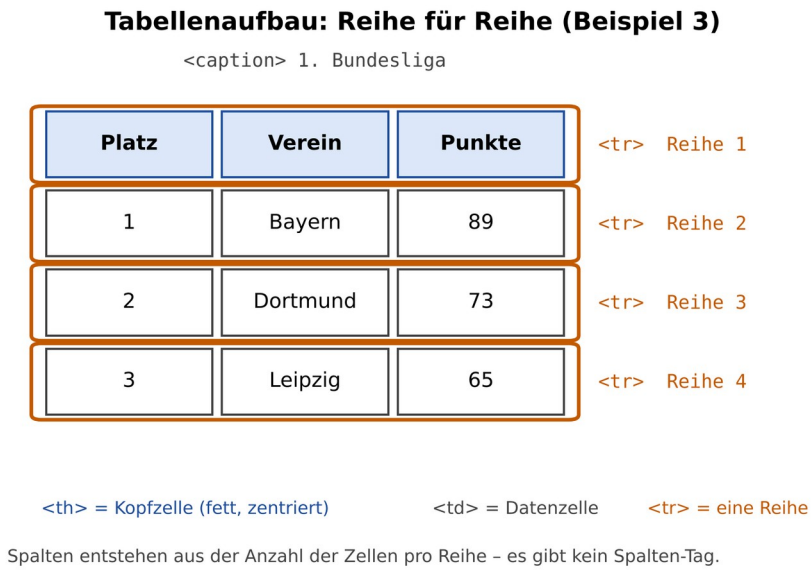
🛠️ Korrektur: Im Skript steht bei `<tr>` einmal "`<tr> ... <tr>`". Richtig ist `<tr> ... </tr>`.

⚠️ Wichtig: HTML baut Tabellen **zeilenweise** auf.

- Es gibt kein Tag für eine Spalte.
- Die Spalten ergeben sich aus der Anzahl der Zellen pro Reihe.

Beispiel 3: Bundesliga-Tabelle

```html
<table>
  <caption>1. Bundesliga</caption>
  <tr>
    <th>Platz</th>
    <th>Verein</th>
    <th>Punkte</th>
  </tr>
  <tr>
    <td>1</td>
    <td>Bayern</td>
    <td>89</td>
  </tr>
  <tr>
    <td>2</td>
    <td>Dortmund</td>
    <td>73</td>
  </tr>
  <tr>
    <td>3</td>
    <td>Leipzig</td>
    <td>65</td>
  </tr>
</table>
```



Beispiel 3 als Skizze: jede Reihe ist ein tr, oben Kopfzellen th, darunter Datenzellen td

✅ Ergebnis: 4 Reihen (1 Kopfreihe, 3 Datenreihen) und 3 Spalten.

Beispiel 4: Dieselbe Tabelle mit thead und tbody

```
<table>
  <caption>1. Bundesliga</caption>
  <thead>
    <tr><th>Platz</th><th>Verein</th><th>Punkte</th></tr>
  </thead>
  <tbody>
    <tr><td>1</td><td>Bayern</td><td>89</td></tr>
    <tr><td>2</td><td>Dortmund</td><td>73</td></tr>
    <tr><td>3</td><td>Leipzig</td><td>65</td></tr>
  </tbody>
</table>
```

💡 Der Browser zeigt beide Varianten gleich an. Die zweite ist klarer gegliedert. Man kann sie leichter mit CSS gestalten.

3.2 🧩 Zellen verbinden: colspan und rowspan

🎯 IHK-Relevanz: gering – Detailwissen; als Übung im logischen Denken trotzdem nützlich.

Denk an einen Stundenplan. Eine Doppelstunde belegt zwei Felder untereinander. Genau so kann eine Zelle mehrere Felder belegen.

Attribut	Wirkung	Beispiel
`colspan`	Zelle überspannt mehrere **Spalten**	`<td colspan="2">über 2 Spalten</td>`
`rowspan`	Zelle überspannt mehrere	`<td rowspan="2">über 2 Reihen</td>`

Attribut	Wirkung	Beispiel
	Reihen	
beides	Zelle überspannt Spalten und Reihen	`<td colspan="4" rowspan="2">...</td>`

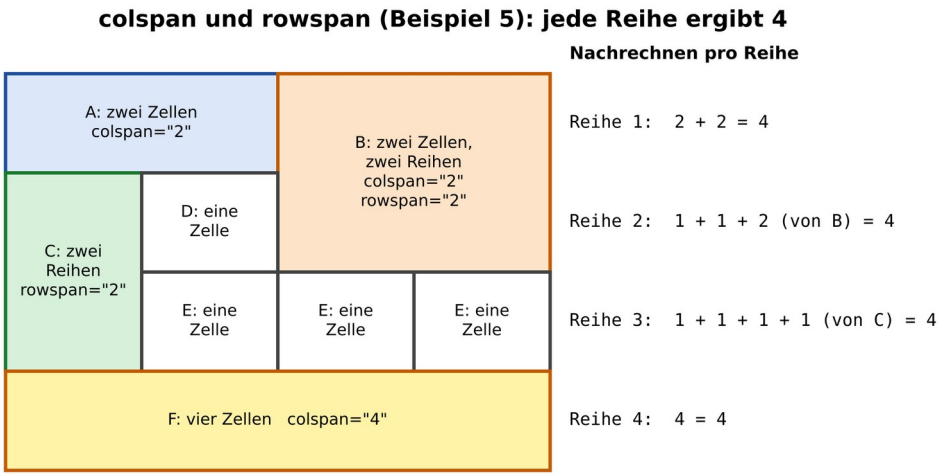
📏 **Grundregel:** Jede Reihe muss rechnerisch **gleich viele Zellen** haben.

- `<th>` zählt dabei als Zelle.
- Eine Zelle mit `colspan="2"` zählt als zwei Zellen.
- Eine Zelle mit `rowspan` belegt auch in den folgenden Reihen ihren Platz.

Beispiel 5: Komplettbeispiel aus dem Skript, nachgeprüft

```
<table>
  <tr>
    <td colspan="2">zwei Zellen</td>
    <td colspan="2" rowspan="2">zwei Zellen,<br>zwei Reihen</td>
  </tr>
  <tr>
    <td rowspan="2">zwei Reihen</td>
    <td>eine Zelle</td>
  </tr>
  <tr>
    <td>eine Zelle</td>
    <td>eine Zelle</td>
    <td>eine Zelle</td>
  </tr>
  <tr>
    <td colspan="4">vier Zellen</td>
  </tr>
</table>
```

So sieht das Raster aus. A bis F sind die Zellen:



Raster von Beispiel 5 mit den Zellen A bis F und der Summe pro Reihe

🧮 Nachrechnen pro Reihe:

Reihe	Zellen im Code	Belegt durch Reihe darüber	Summe
1	2 (colspan) + 2 (colspan)	–	4
2	1 + 1	2 (rowspan von B)	4
3	1 + 1 + 1	1 (rowspan von C)	4
4	4 (colspan)	–	4

✅ Jede Reihe ergibt 4. Die Tabelle ist korrekt aufgebaut.

🛠️ Korrektur:

- Im Skript fehlt vor der dritten Reihe das `</tr>`. Laut HTML-Standard darf `</tr>` fehlen, wenn direkt ein neues `<tr>` folgt. Der Browser ergänzt es dann.
- Sauberer ist es, immer zu schließen. So ist es im Code oben.
- Außerdem steht im Skript einmal “rowpan”. Richtig ist `rowspan`.

3.3 🤖 Praxis: HTML-Tabellen aus Skripten

🎯 IHK-Relevanz: mittel – Automatisierung mit Skripten nennt die Verordnung für AP2 Konzeption/Administration; HTML ist dabei nur das Ausgabeformat.

➕ Ergänzung, nicht im Skript: Administratoren erzeugen Berichte oft automatisch als HTML. Jeder Browser kann sie öffnen.

Beispiel 6: PowerShell

```
Get-Service | ConvertTo-Html -Property Name, Status | Out-File
dienste.html
```

🔍 Was passiert hier?

1. Der Befehl liest alle Windows-Dienste.
2. Er baut daraus eine HTML-Tabelle mit den Spalten Name und Status.
3. Das Ergebnis landet in `dienste.html`.

Beispiel 7: Bash

```
#!/bin/bash
echo "<table>" > bericht.html
echo "<tr><th>Datei</th><th>Größe</th></tr>" >> bericht.html
for f in /var/log/*.log; do
  echo "<tr><td>$f</td><td>$(du -h "$f" | cut -f1)</td></tr>" >>
bericht.html
done
echo "</table>" >> bericht.html
```

🔍 Das Skript schreibt eine Tabelle. Sie zeigt alle `.log`-Dateien in `/var/log` mit ihrer Größe.

- > legt die Datei neu an.
- >> hängt jede weitere Zeile an.

✏️ Übungsfragen Block 3

Ü3.1 Schreiben Sie eine HTML-Tabelle mit Kopfreihe “Gerät” und “IP-Adresse” und zwei Datenreihen: Router 192.168.1.1, Drucker 192.168.1.20.

Ü3.2 Eine Tabelle hat 3 Spalten. In der ersten Reihe steht `<td colspan="2">A</td>`. Wie viele `<td>` ohne Attribute fehlen noch in dieser Reihe?

Ü3.3 Warum gibt es in HTML kein Tag für eine Spalte?

✅ Lösungen

Block 1

Ü1.1 Block-Elemente beginnen in einer neuen Zeile. Sie bilden einen eigenen Absatz, z. B. `<p>`, `<div>`, `<h1>`, `<table>`. Inline-Elemente bleiben im Textfluss ohne Umbruch, z. B. `<em>`, `<strong>`, `<span>`, `<a>`.

Ü1.2 Die Tags `<strong>` und `<em>` überkreuzen sich. `<em>` wurde zuletzt geöffnet. Es muss also zuerst geschlossen werden. Richtig:

```
<p>Bitte <strong>nicht <em>ausschalten</em></strong>!</p>
```

Ü1.3 Sie gruppieren Inhalte. `<div>` fasst Blöcke zusammen. `<span>` markiert Text im Fließtext. Beide dienen als Ziel für CSS (Gestaltung) und JavaScript (Verhalten).

Block 2

Ü2.1

```html
<a href="docs/handbuch.pdf">Handbuch</a>
```

Ü2.2 Zwei von diesen Gründen:

- Fehlt das Bild, erscheint der Ersatztext.
- Screenreader lesen den Text vor (Barrierefreiheit).
- Suchmaschinen können den Bildinhalt erfassen.

Ü2.3 Start: `/projekt/doku/`. `..` führt zu `/projekt/`. Dann `bilder/netz.png`. Ergebnis: **`/projekt/bilder/netz.png`**.

Block 3

Ü3.1

```html
<table>
  <tr>
    <th>Gerät</th>
    <th>IP-Adresse</th>
  </tr>
  <tr>
    <td>Router</td>
    <td>192.168.1.1</td>
  </tr>
  <tr>
    <td>Drucker</td>
    <td>192.168.1.20</td>
  </tr>
</table>
```

Ü3.2 `colspan="2"` belegt 2 von 3 Spalten. Es fehlt **ein** `<td>`.

Ü3.3 HTML baut Tabellen Reihe für Reihe auf. Spalten entstehen aus der Position der Zellen in jeder Reihe. Deshalb braucht es kein eigenes Spalten-Tag. (Ergänzung: Für die Gestaltung ganzer Spalten gibt es `<colgroup>` und `<col>`.)

🔑 Schlagwortliste

Begriff	Kurzdefinition	Block
Block-Element	eigener Absatz mit Umbruch davor und danach	1
Inline-Element	im Textfluss, ohne Umbruch	1
`<em>` / `<strong>`	Betonung (kursiv) / starke Betonung (fett)	1
`<mark>`	Textmarker	1

Begriff	Kurzdefinition	Block
`<sup>` / `<sub>`	hochgestellt / tiefgestellt	1
`<cite>`	Titel eines Werks	1
`<code>`	Programmcode, Festbreitenschrift	1
`<del>` / `<ins>`	gelöschter / eingefügter Text	1
Zeichenreferenz	Sonderzeichen im Code, z. B. `×`, `π`	1
Nesting	Verschachteln; zuletzt geöffnet, zuerst geschlossen	1
Validator	Werkzeug, das HTML auf Fehler prüft	1
`<div>` / `<span>`	neutraler Block- / Inline-Container	1
HTML5-Strukturelemente	header, nav, main, section, article, aside, footer	1
Semantik	Tags beschreiben die Bedeutung, CSS das Aussehen	1
Attribut	Zusatzangabe im öffnenden Tag: `name="wert"`	2
`<a>` / `href`	Link / Linkziel	2
`id` / `#`	Sprungziel / Verweis auf ein Sprungziel	2
`<img>` / `src` / `alt`	Bild / Bildquelle / Ersatztext	2
Barrierefreiheit	Inhalte für alle nutzbar, z. B. mit Screenreader	2
relativer Pfad	ab aktueller Seite, z. B. `../daten.pdf`	2
absoluter Pfad	ab Document Root, beginnt mit `/`	2
Document Root	oberstes Verzeichnis einer Website auf dem Webserver	2
`<table>`	umschließt eine Tabelle	3
`<tr>` / `<td>` / `<th>`	Reihe / Datenzelle / Kopfzelle	3
`<caption>`	Tabellenbeschriftung	3
`<thead>`, `<tbody>`, `<tfoot>`	Kopf-, Haupt-, Fußbereich einer Tabelle	3
`colspan` / `rowspan`	Zelle über mehrere Spalten /	3

Begriff	Kurzdefinition	Block
	Reihen	
ConvertTo-Html	PowerShell-Cmdlet, erzeugt HTML-Tabellen aus Objekten	3