2. Veranstaltung

Inhalt

- White-Space-Zeichen
- Textumbruch
- Textabsätze
- Überschriften
- Logische Textauszeichnungen
- Verweise
- Grafiken
- Kommentare

White-Space-Zeichen

Zeilenumbruch-Zeichen, **Tabulator**-Zeichen und einfache **Leerzeichen** bilden in HTML die Klasse der so genannten White-Space-Zeichen (white space = "weißer Raum").

Die Browser setzen in der Regel ein **Tabulator-Zeichen** oder **Zeilenumbruch-Zeichen** im **Editor** als **Leerzeichen** im **HTML-Text** um.

White-Space-Zeichen

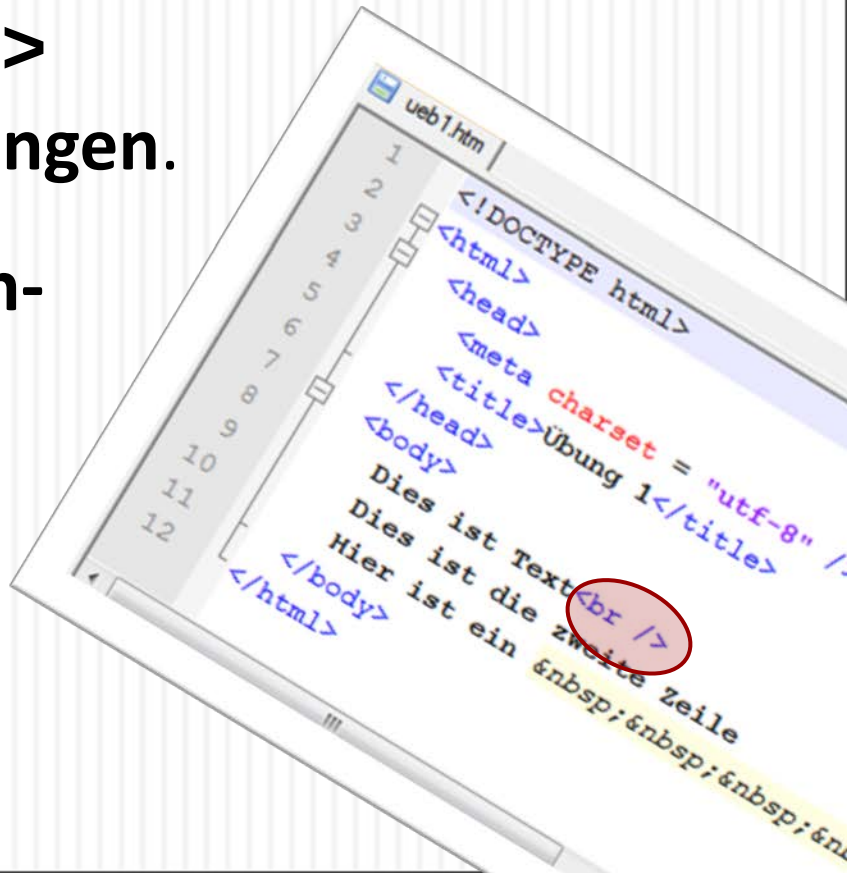
Mehrere solcher White-Space-Zeichen hintereinander werden **ignoriert** und zu **einem einzigen** Leerzeichen zusammengefasst.

Um **mehrere** Leerzeichen **hintereinander** zu **erzwingen**, können Sie ein **geschütztes** Leerzeichen ` ` (nonbreaking space) oder auch ` ` mehrmals hintereinander eingeben.

Textumbruch

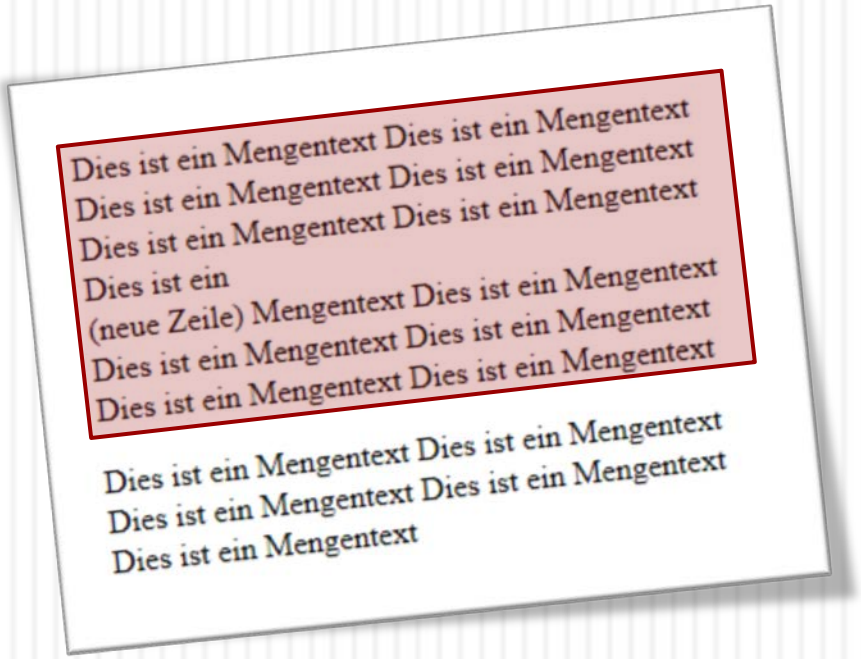
Text wird vom Web-Browser bei der Anzeige meist **automatisch umbrochen**. Sie können jedoch einen Zeilenumbruch auch mit **
** (br = break = Umbruch) **erzwingen**.

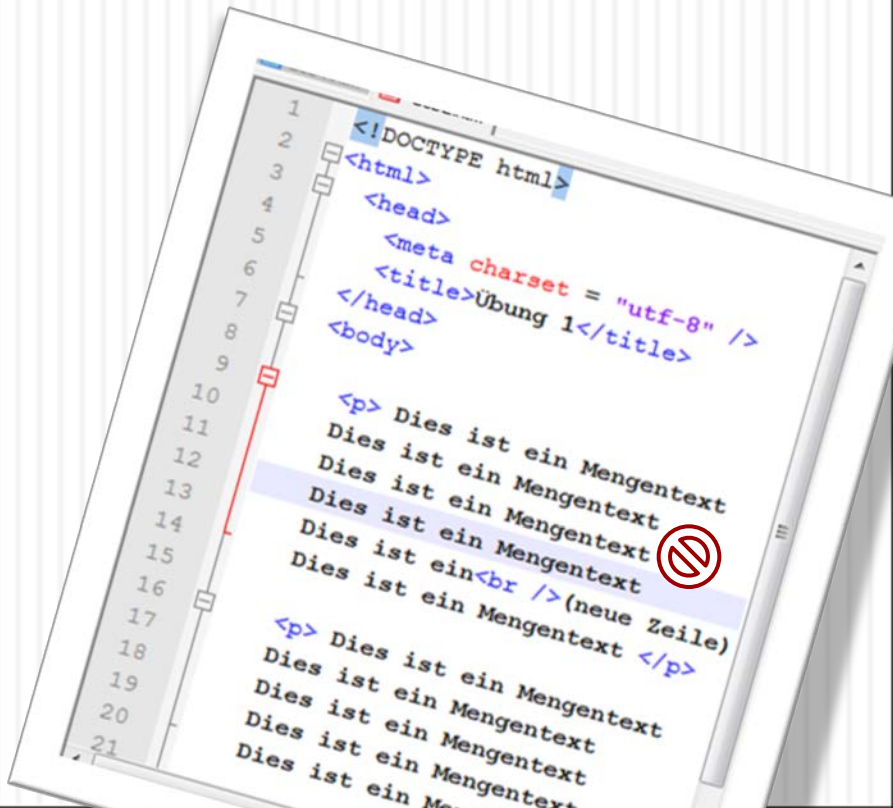
Dabei ist es egal, ob das **allein-stehende** Tag am **Ende** der **vorherigen** Zeile steht oder am **Anfang** der **folgenden** Zeile.



Textabsätze

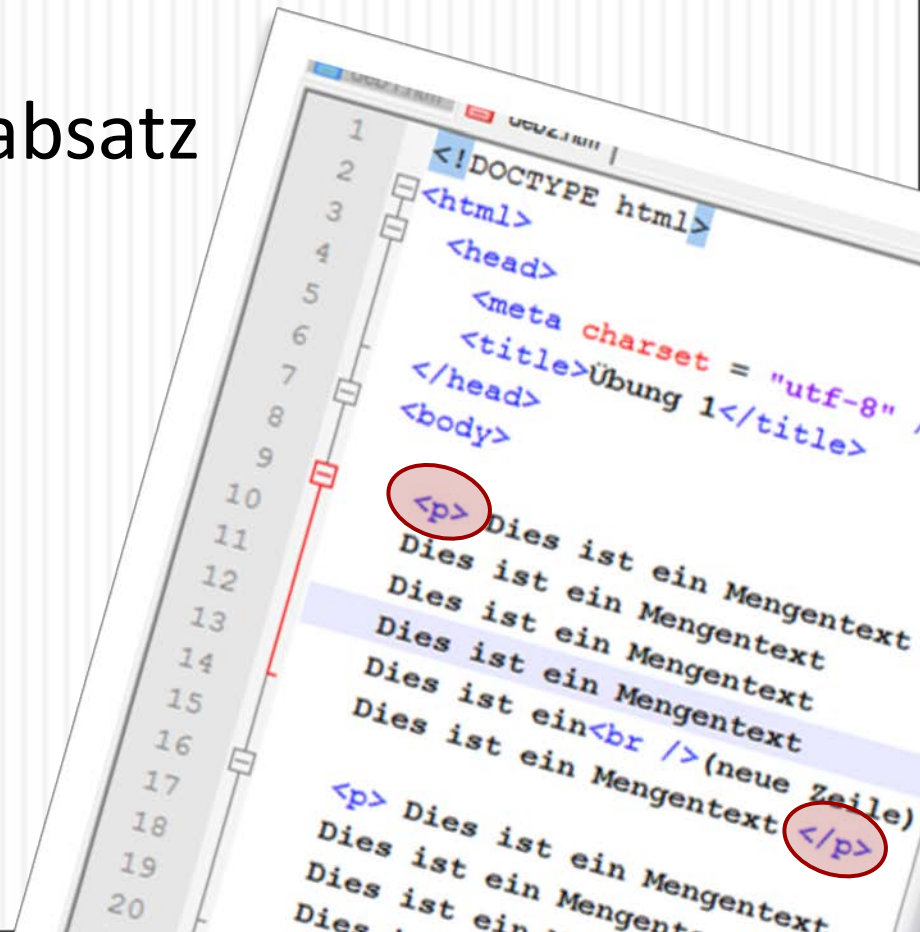
Absätze dienen der optischen **Gliederung** eines Textes. WWW-Browser **ignorieren** jedoch harte **Umbrüche** im **Editor**.





Textabsätze

** ** (p = paragraph = Absatz) leitet einen Textabsatz ein.
** ** beendet den Textabsatz und steht am Ende des Absatztextes.



Textabsätze

Das ** **-Element darf **keine** anderen **block-erzeugenden** Elemente wie z.B. Überschriften oder Listen enthalten:

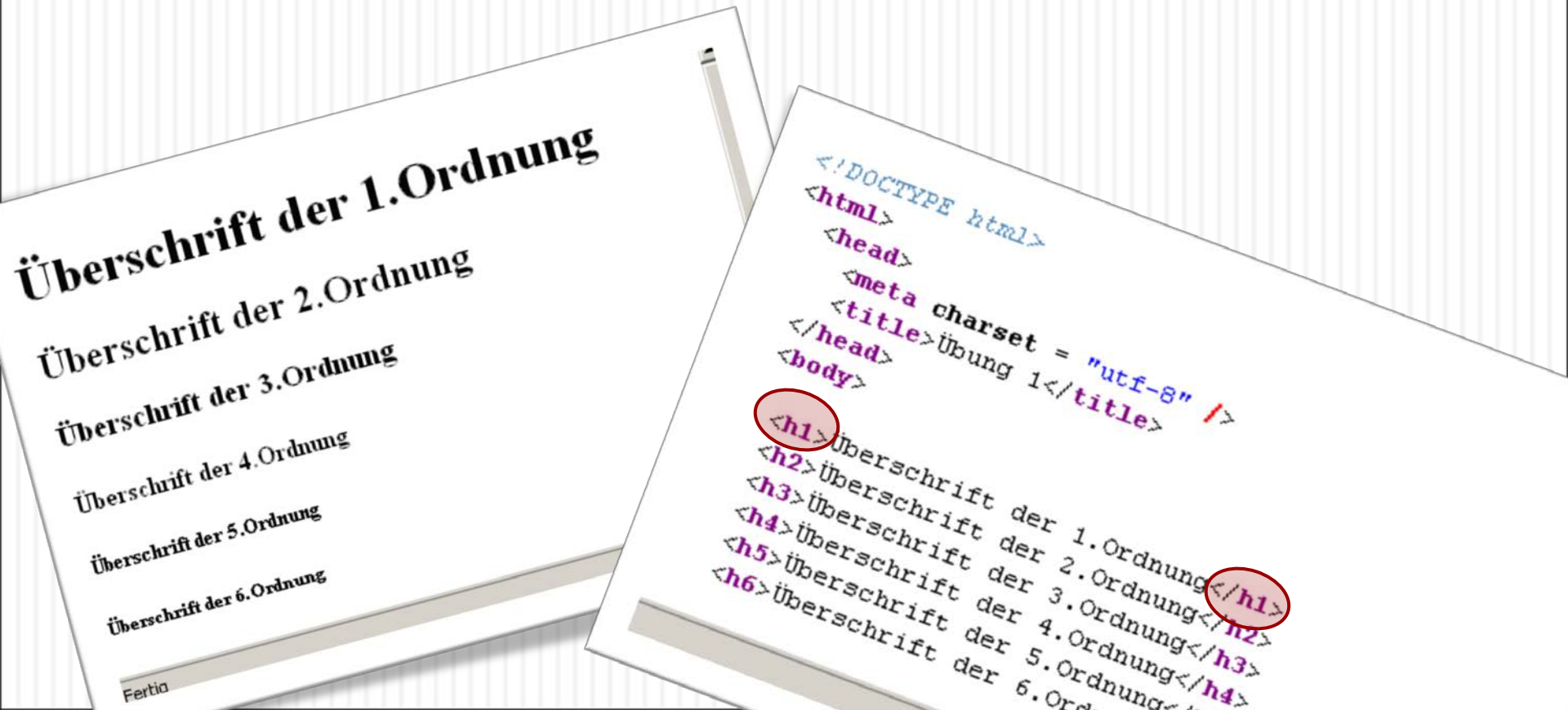
```
<p>Abschnitt<h1>Überschrift</h1></p>
```

Das schließende ** ** muss **vor** den blockerzeugenden **anderen** Elementen kommen:

```
<h1>Überschrift</h1>
<p>Abschnitt</p>
```

Überschriften

HTML unterscheidet **6 Überschriftenebenen**, um Hierarchieverhältnisse abzubilden.



Überschriften

<h[1-6]> (h = heading = Überschrift) leitet eine Überschrift ein.

Dabei ist 1 die **höchste Überschriftenebene** , und 6 ist die **niedrigste Überschriftenebene**.

</h[1-6]> steht am Ende der Überschrift.

Vor und nach Überschriften sind **keine Absatz-schaltungen** nötig.

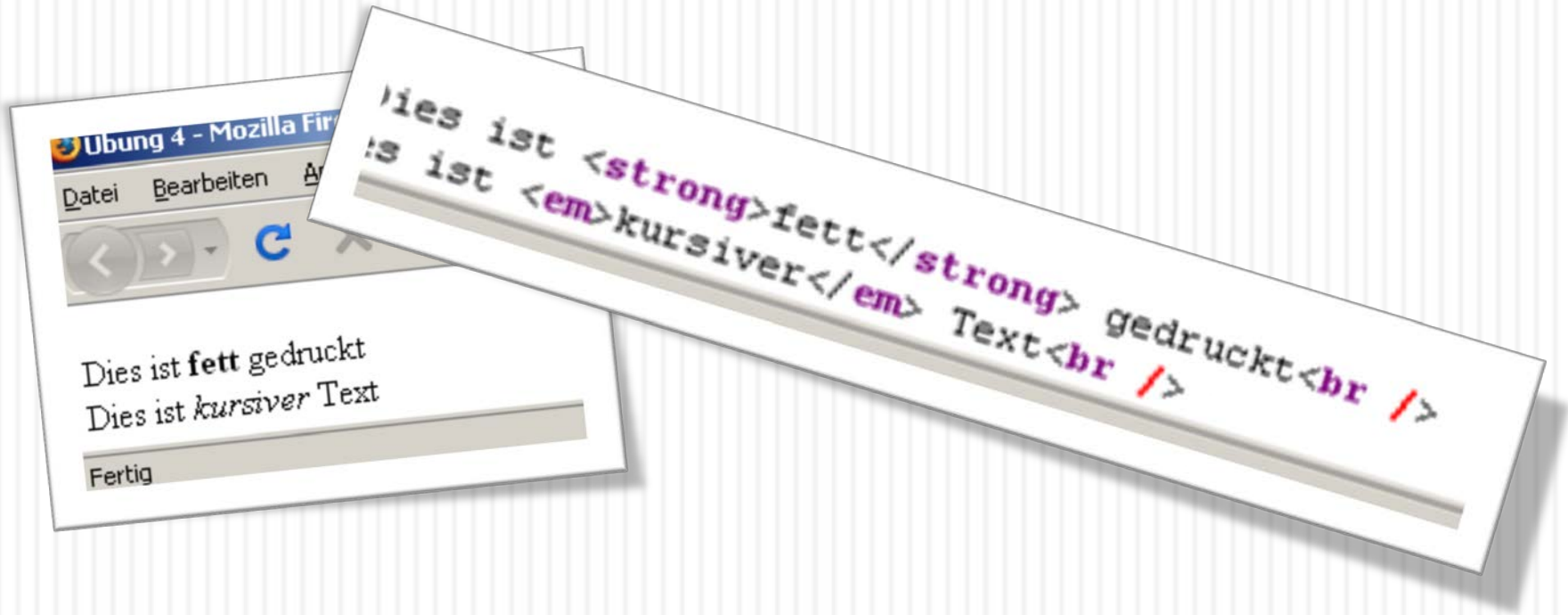
Logische Textauszeichnungen

In HTML gibt es **logische** und **physische** Elemente zur Tesxtauszeichnung. **Logische** Textauszeichnungen haben **Bedeutungen** wie "betont" oder "emphatisch".

Bei **logischen** Elementen **entscheidet** der **Web-Browser**, wie ein solcher Text **hervorgehoben** wird (z.B. fett, kursiv oder andersfarbig).

Logische Textauszeichnungen

Mit **Stylesheets** können logische Textauszeichnungen jedoch **nach Wunsch** formatiert werden.



Logische Textauszeichnungen

Es stehen verschiedene HTML-Elemente zur Verfügung, um Text logisch auszuzeichnen.

Hier zwei häufig verwendete Beispiele:

`<em>...</em>` betonter Text ("emphatisch")
`<strong>...</strong>` stark betonter Text

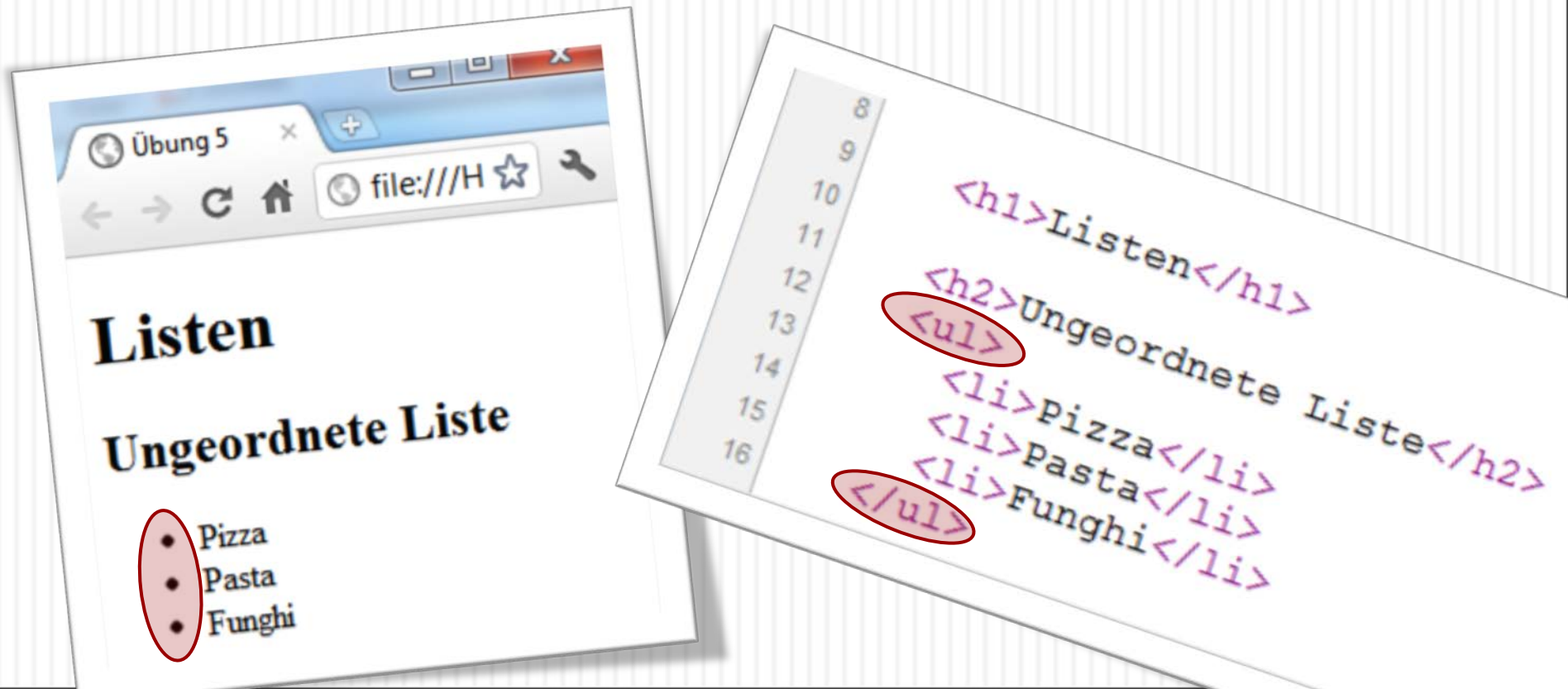
Logische Textauszeichnungen

Inline-Elemente für Auszeichnungen im Text müssen **innerhalb** anderer **Block-Elemente** (z.B. p-tag oder h1- tag) vorkommen.

```
<p> Dies ist <strong>fett</strong>
gedruckt und dies ist
<em>kursiver</em> </p>
```

Listen - Aufzählungslisten

Bei einer **Aufzählungsliste** werden alle Listeneinträge mit einem **Aufzählungszeichen** (Bullet) versehen.



Listen - Aufzählungslisten

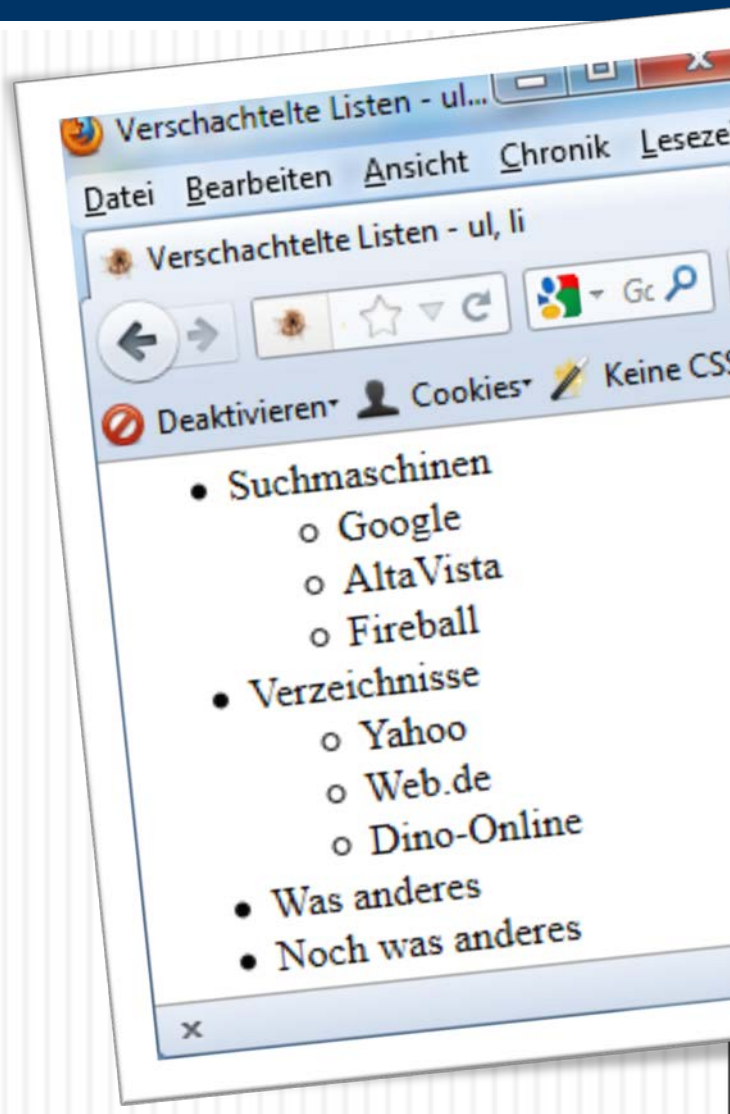
** **	leitet eine **Aufzählungsliste** ein (ul = unordered list = unsortierte Liste).
** **	eröffnet einen neuen **Punkt** innerhalb der Liste (li = list item = Listeneintrag).
** **	beendet den Listeneintrag.
** **	beendet die Liste.

Listen - Aufzählungslisten

Wie das Bullet **dargestellt** wird, **bestimmt** dabei der **Web-Browser**.

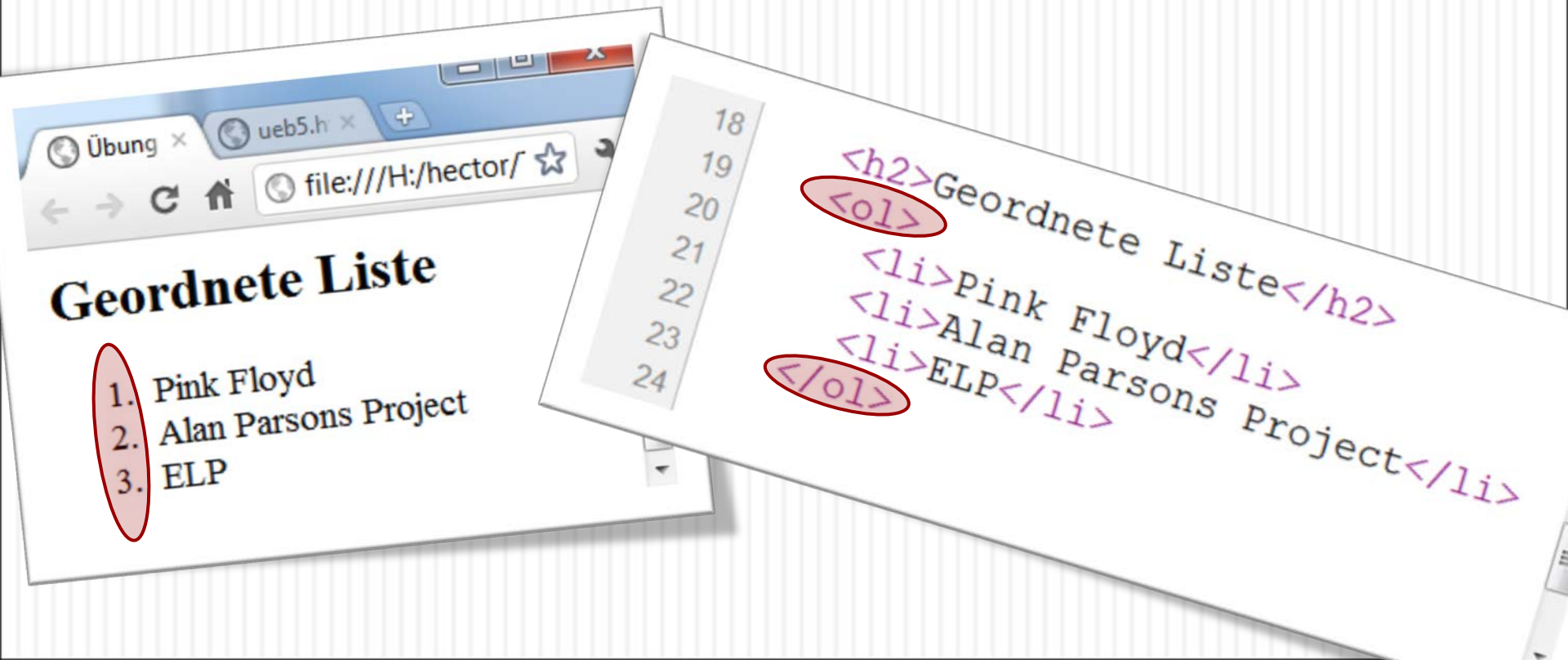
Das **Verschachteln** von Listen ist ebenfalls möglich.

Zwischen **
** und **
** darf eine komplette **weitere** Liste stehen.



Listen - Nummerierte Listen

Bei einer **nummerierten Liste** werden alle Listeneinträge **automatisch durchnummeriert**.



Listen - Nummerierte Listen

** **	leitet eine **nummerierte Liste** ein (ol = ordered list = nummerierte Liste).
** **	eröffnet einen neuen **Punkt** innerhalb der Liste (li = list item = Listeneintrag).
** **	beendet den Listeneintrag.
** **	beendet die Liste.

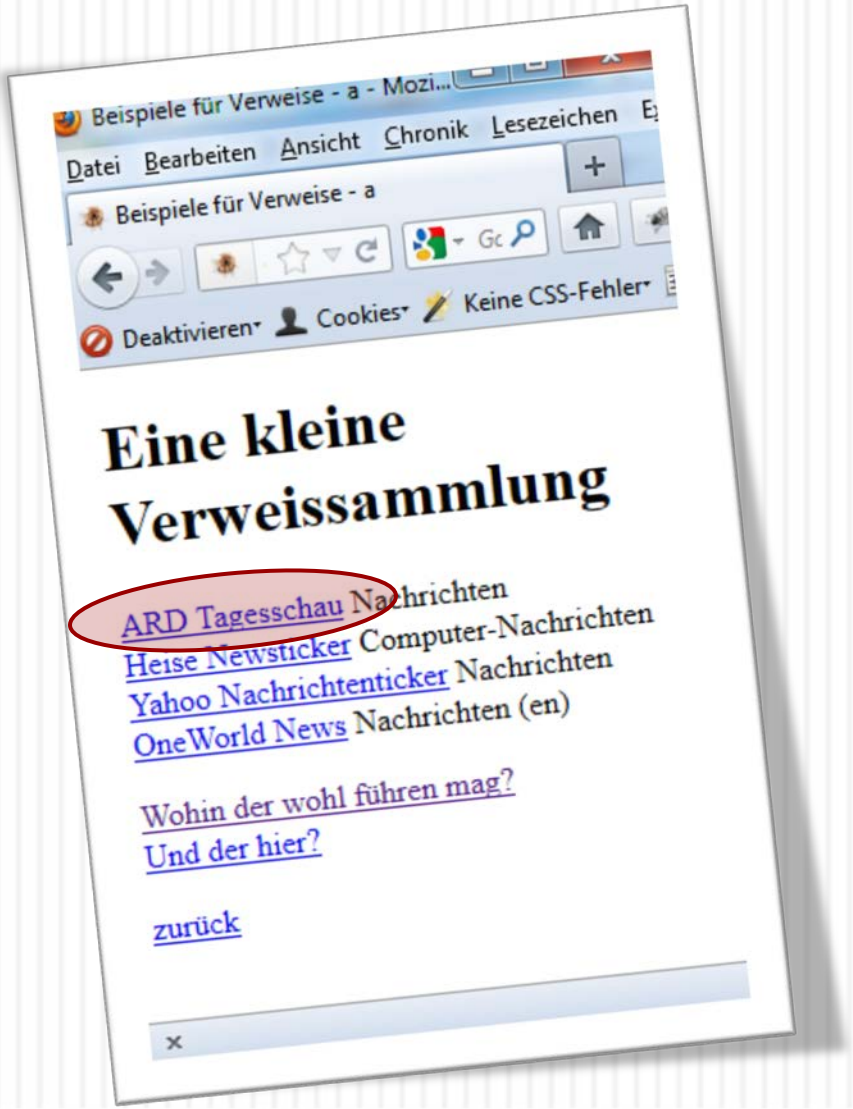
Listen – Nummerierte Listen

Verschachteln von nummerierten Listen ist ebenfalls möglich, bewirkt aber **keine** Gesamtnummerierung.

Automatische **Nummerierungs-Hierarchien** wie 1, 1.1, 1.1.1, sind mit HTML alleine **nicht** möglich.

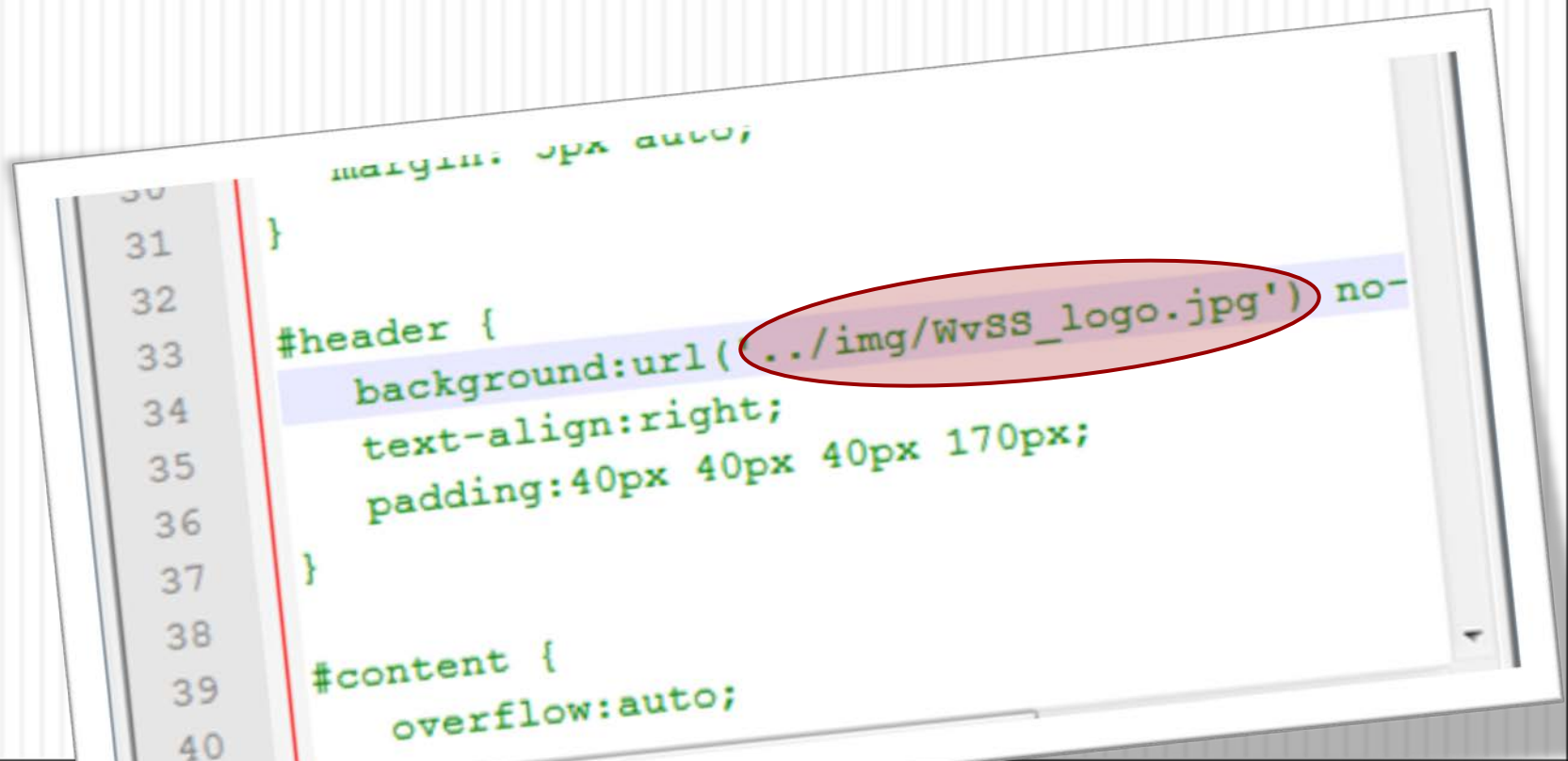
Verweise

Ein **Verweis** (link) zu einer anderen Web-Seite ist nur ausführbar, wenn er sein **Verweis-ziel** benennt.



Verweise

Ebenso gibt es in **Ergänzungssprachen** wie **Stylesheets** oder **JavaScript** Stellen, an denen Sie andere Datenquellen **referenzieren** müssen.



Verweise

HTML-Dateien bestehen bekanntlich **nur aus Text**. Dennoch enthalten viele Web-Seiten Elemente wie

- Grafiken bzw. Hintergrundgrafiken,
- Multimedia-Elemente,
- Java-Applets,
- Flash-Animationen etc.

Auch diese Elemente werden in Form einer **Referenz** auf eine entsprechende **Datenquelle** notiert.

Verweise

Für all diese Zwecke wird das **Referenzieren** in HTML benötigt.

Die **Regeln** zum Referenzieren gehen dabei aus dem zentralen und einheitlichen **Adressierungsschema** im Web hervor (unabhängig von der Syntax einzelner Betriebssysteme).

Verweise

Alle Verweise in HTML haben einen **einheitlichen Aufbau**, egal ob sie

- zu einem **Verweisziel** in der **gleichen** Datei,
- zu einer **anderen** HTML-Datei im **eigenen** Projekt,
- zu einer beliebigen **WWW-Adresse** oder
- zu einer **beliebigen Datei** eines **anderen Dateityps** im **Internet** oder **lokal** auf dem eigenen Rechner führen.

Verweise

Für **Verweise** in HTML gibt es das **a-Element** (a = anchor = Anker).

Zusätzlich ist das **Attribut** `href` erforderlich (href = hyper reference = Hyper(text)-Referenz) mit dem **Verweisziel** als **Wert** für das href-Attribut.

Als Inhalt zwischen **<a>** und **** steht der für den **Anwender** als **Verweis sichtbare** Text.

```
<a href="seite_2.htm">Seite 2</a>
```

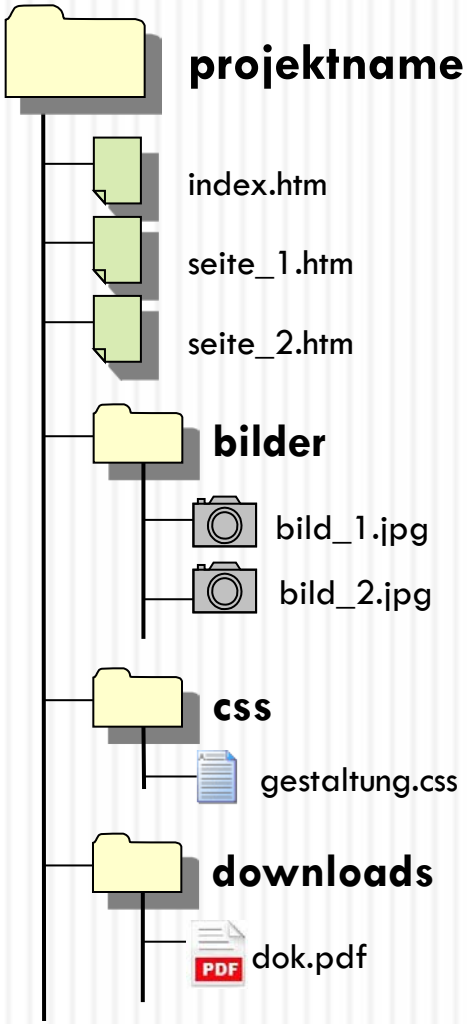
Projekt-interne Verweise

Ein Web-Projekt besteht typischerweise aus **mehreren** Einzelseiten, die miteinander **verlinkt** sind.

Verweise auf andere Projektdateien sollten mit **relativen** Angaben zum **Verweisziel** definiert werden.

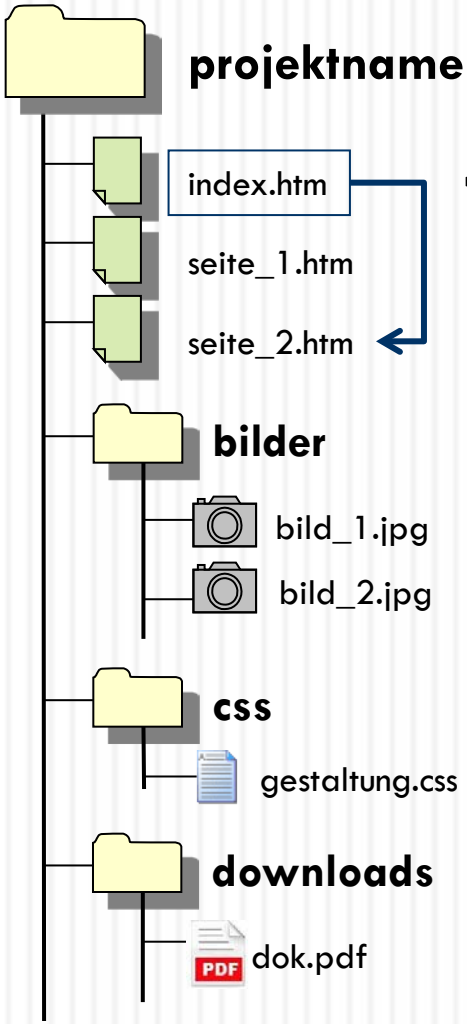
Das Projekt bleibt dadurch **flexibler**, und die Verweise **funktionieren** auch in **anderen** Umgebungen.

Projekt-interne Verweise



Um **projekt-interne** Verweise mit **relativen** Angaben besser verstehen zu können, sollte man die **typische Ordner-Struktur** innerhalb eine **Web-Projekts** vor Augen haben.

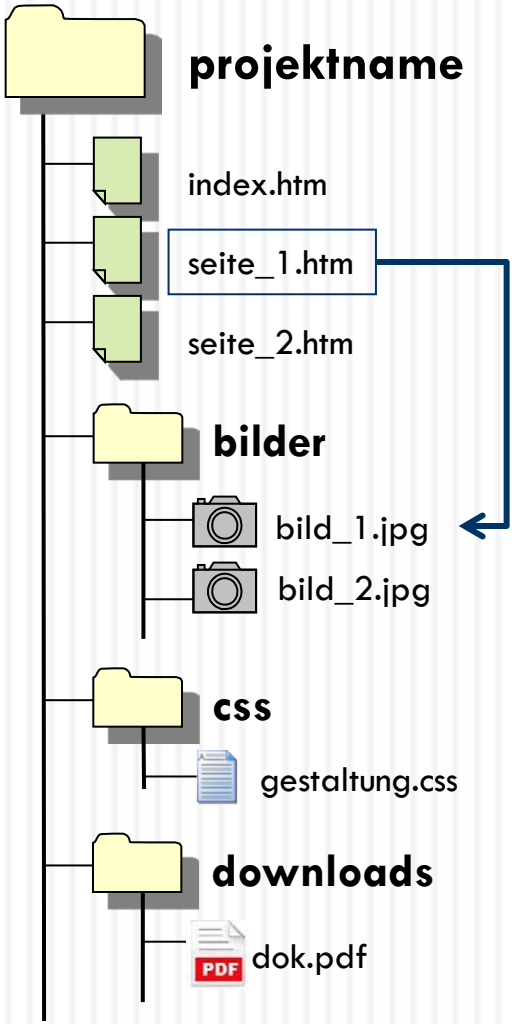
Projekt-interne Verweise




```
<a href="seite_2.htm">Seite 2</a>
```

Beide Dateien sind im **gleichen** Verzeichnis abgelegt.
Deshalb genügt bei **href** die Angabe des Dateinamens **ohne** weitere Zusätze.

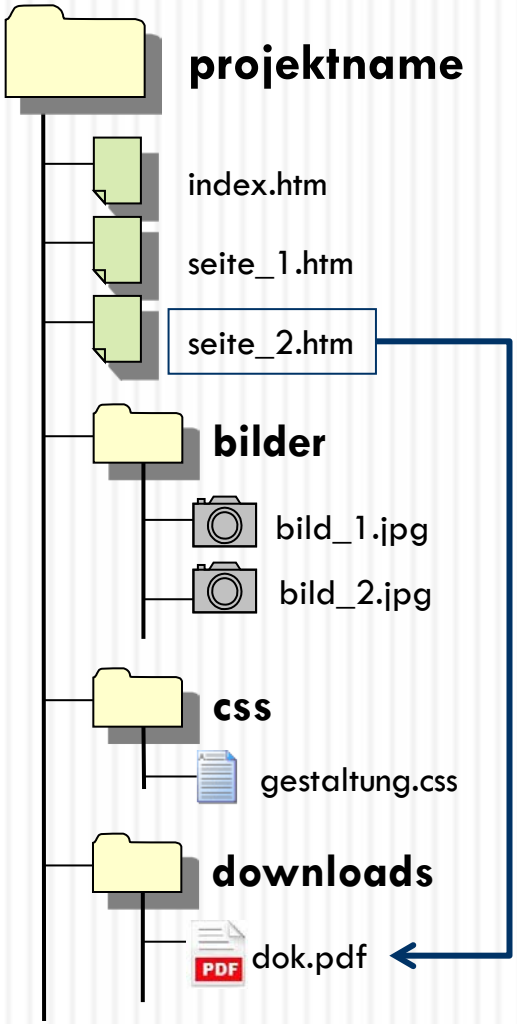
Projekt-interne Verweise




```
<a href="bilder/bild_1.jpg">Bild 1</a>
```

Die referenzierte Datei ist in einem **Unterverzeichnis** abgelegt. Deshalb muss bei href der **Ver-zeichnisname** gefolgt von einem / (slash) **vorangestellt** werden.

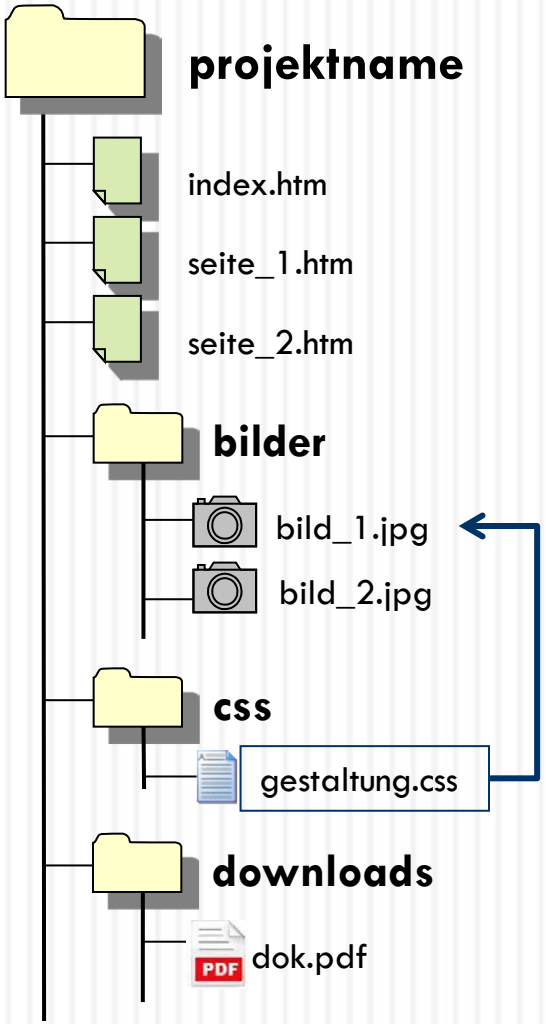
Projekt-interne Verweise




```
<a href="downloads/dok.pdf">Bild 1</a>
```

Auch **andere Dateiformate**, die der Web-Browser **erlaubt**, sind möglich (z.B.: .jpg oder .pdf)

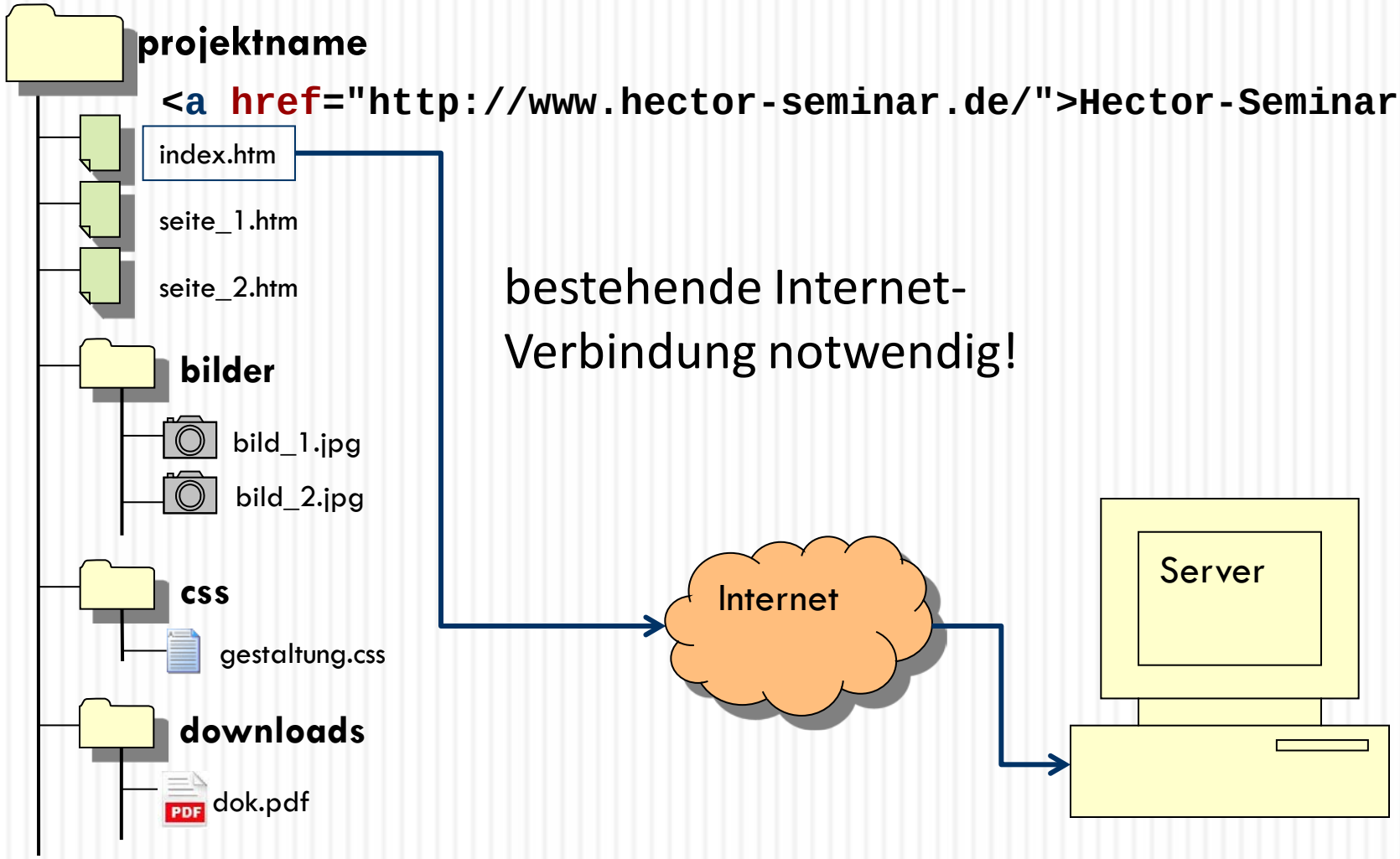
Projekt-interne Verweise



Beide Dateien liegen in **unterschiedlichen Verzeichnissen**, die ihrerseits im **selben Stamm** liegen. Deshalb muss bei href dem **Verzeichnisnamen** ein **../** (eine Ebene nach oben) **vorangestellt** werden.

```
<a href="../bilder/bild_1.jpg">Bild 1</a>
```

Projekt-externe Verweise



Projekt-externe Verweise

```
<a href="http://www.hector-seminar.de/">Hector-Seminar</a>
```

Es sollte **immer** ein **slash** angefügt werden, falls **nicht** eine **einzelne**, spezielle Datei referenziert wird, sondern das **Stammverzeichnis** oder ein **Unterverzeichnis**

Dort sucht der Server dann **automatisch** nach einer Datei mit einem **speziellen** Namen (u.a. index.htm).

Inhalt von Verweisen

Als Inhalt des a-Elements (der Verweistext) ist nicht **nur** reiner Text erlaubt.

Sie können im Verweistext auch andere **Inline-Elemente** (z.B.) notieren.


```
<p><a  href = "unterseite_1.htm"><b>Unterseite 1</b></a></p>
<p><a  href = "unterseite_2.htm"><b>Unterseite 2</b></a></p>
```

Inhalt von Verweisen

Unter anderem können Sie anstelle von Text auch **Grafiken** als Verweise fungieren lassen.


```
<a href="flashgmbh_home.html"><img src="flashgmbh/m_home.gif" border="0"></a>
<a href="flashgmbh_prod.html"><img src="flashgmbh/m_prod.gif" border="0"></a>
```

Zielfenster für Verweise

Per **Voreinstellung** werden alle Verweise im **aktuellen** Fenster geöffnet.

Sie können aber entscheiden, ob ein **Verweisziel** im aktuellen Browser-Fenster ausgegeben werden soll, oder ob ein **neues Fenster** dafür geöffnet werden soll (bei Verweisen zu **fremden** Web-Angeboten meist **sinnvoll**).

Zielfenster für Verweise

Mit dem Attribut **target** im einleitenden <a>-Tag können Sie ein **Zielfenster** für den Verweis festlegen:

`_blank`	um den Verweis in einem **neuen** Fenster zu öffnen
`_self`	um den Verweis im **aktuellen** Fenster zu öffnen (Standard, kann entfallen)

```
<a href="http://www-hector-seminar.de/" target="_blank">
  Hector-Seminar
</a>
```

Grafiken

Um eine **Grafik** in HTML-Dateien einzubinden, **referenziert** man die **Grafikdatei** im HTML-Quelltext.

Geeignete Dateiformate für Web-gerechte Grafiken sind vor allem GIF und JPEG, und PNG

Grafiken

Grafiken sind **Inline-Elemente** und müssen innerhalb von Block-Elementen vorkommen (z.B. einem Text-absatz).

Für Grafikreferenzen gibt es in HTML das **`<img>`-**Tag (*img = image = Bild, src = source = Quelle*).

Es handelt sich um ein Standalone-Tag (ohne Element-inhalt und ohne End-Tag).

Grafiken

Mit Hilfe von **Attributen** bestimmen Sie nähere Einzelheiten der Grafikreferenz.
Zwei Attribute sind **Pflicht** und müssen immer angegeben werden:

`src` und **`alt`**

Grafiken

Mit dem Attribut **`src`** wird die gewünschte Grafikdatei bestimmt. Dabei gelten die **Regeln** wie bei den zuvor besprochenen **Verweisen**.

Das Attribut **`alt`** ist Pflichtangabe für jede Grafikeinbindung. Geben Sie darin einen **Alternativtext** an für den Fall, dass die Grafik **nicht** angezeigt werden kann.

```
<img src="img/bild.gif" alt="Grafik">
```

Kommentare (Nicht angezeigter Text)

HTML beinhaltet die Möglichkeit, innerhalb einer HTML-Datei **Kommentare** einzufügen.
Diese werden von Web-Browsern **nicht** angezeigt.

```
<h1>Willkommen!</h1>
    <!--
        Kommentar:
        das obendrüber ist
        eine Überschrift
    -->
```

Kommentare (Nicht angezeigter Text)

- Kommentare sind z.B. sinnvoll,
  - um interne Angaben zu Autor/Erstelldatum in einer Datei zu platzieren,
  - um interne Anmerkungen zu bestimmten Textstellen zu machen oder
  - um verwendete HTML-Tags intern "auszukommentieren".

Neue HTML5-Elemente

Viele Webseiten enthalten bzw. enthielten HTML-Code wie zum Beispiel...

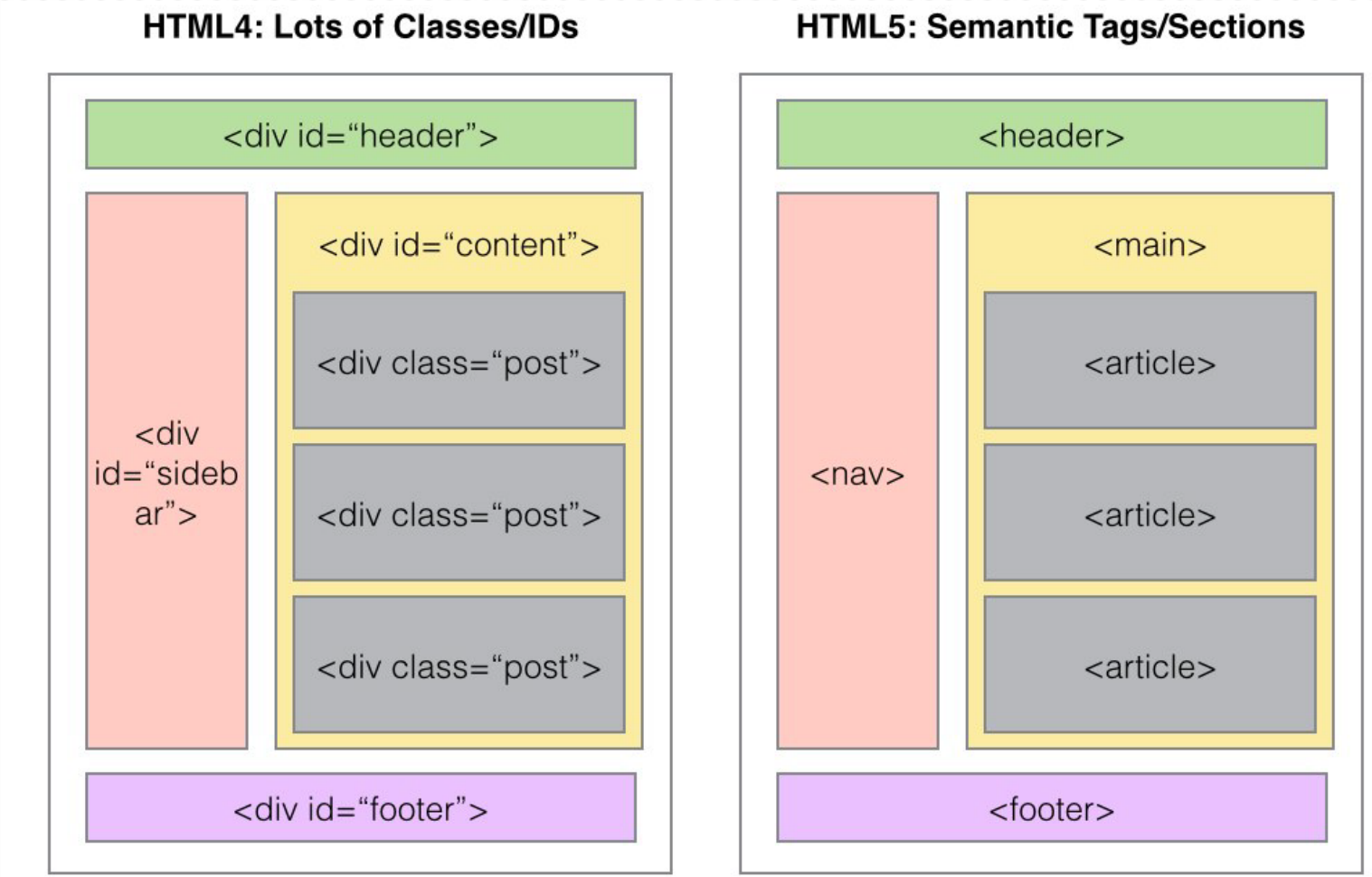
```
<div id = "nav">
<div class = "header">
<div id = "footer">
```

...um einen Bereich als Navigation, Header und Footer zu kennzeichnen.

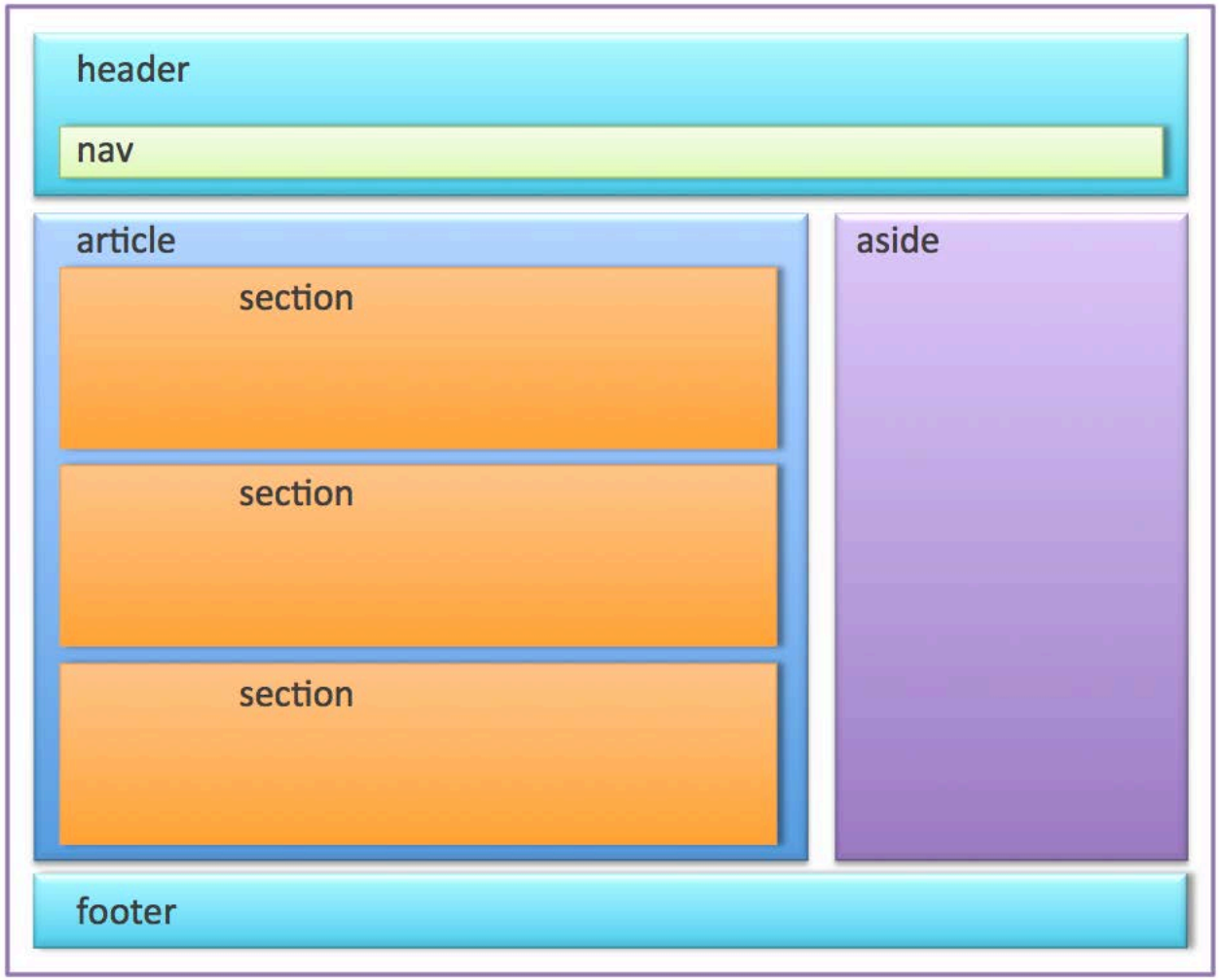
Neue HTML5-Elemente

- HTML5 bietet jetzt viele neue semantische Elemente an, um verschiedene Bereiche einer Webseite zu definieren:
  - <article>
  - <aside>
  - <details>
  - <figcaption>
  - <figure>
  - <summary>
  - <time>
  - <footer>
  - <header>
  - <main>
  - <mark>
  - <nav>
  - <section>

Neue HTML5-Elemente



Neue HTML5-Elemente



Neue HTML5-Elemente