



# WEB ANWENDUNGEN I

HTML

# ÜBERSICHT GESAMT

EINFÜHRUNG

HTML

CSS

JAVA SCRIPT

JQUERY

BOOTSTRAP

# HTML – BESCHREIBUNGSSPRACHE VON HYPERTEXT-DOKUMENTEN

- Struktur und Konvention
- HTML-Elemente
- Hypertext
- Tabellen
- Grafiken
- Formulare

# STRUKTUR VON HTML

- HTML Dokumente
  - Sind reine Textdateien
  - Entsprechen einer HTML-Konvention
  - Bestehen aus
    - Darzustellenden Text
    - Strukturinformation (Markup)
      - Elementen und
      - Attributen
    - Entitäten

# HTML ELEMENTE

```
<!DOCTYPE html>
<html lang="de">
  <head>
    <title> Titel des Dokuments</title>
  </head>
  <body>

    <h1>Überschrift</h1>
    <p>Absatz oder Paragraph</p>

  </body>
</html>
```

Elemente werden mit Tags definiert (Start-Tag, End-Tag)

Elemente können Texte umschließen (Elementinhalt)

Hierarchisch gegliedert

Genau ein Wurzelelement

Elemente ohne Inhalt können abgekürzt werden: <br />

# HTML KOMMENTARE

```
<!DOCTYPE html>
<html lang="de">
  <head>
    <title> Titel des Dokuments</title>
  </head>
  <body>
    <!-- Der Datensatz von German Nemirovski -->
    <h1>Überschrift</h1>
    <p>Absatz oder Paragraph</p>

  </body>
</html>
```

Kommentare erscheinen nicht in der Ausgabe  
Können aber im Quellcode an der Seite ausgelesen werden

# HTML DEKLARATION

```
<!DOCTYPE html>
```

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

```
<head>
```

```
<title> Titel des Dokuments</title>
```

```
</head>
```

```
<body>
```

```
<!-- Der Datensatz von German Nemirovski -->
```

```
<h1>Überschrift</h1>
```

```
<p>Absatz oder Paragraph</p>
```

```
</body>
```

```
</html>
```

Deklariere ein HTML Dokument

# <HTML>

```
<!DOCTYPE html>
```

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

Wurzel-Element eine HTML Dokuments

```
<head>
```

```
<title> Titel des Dokuments</title>
```

```
</head>
```

```
<body>
```

```
<!-- Der Datensatz von German Nemirovski -->
```

```
<h1>Überschrift</h1>
```

```
<p>Absatz oder Paragraph</p>
```

```
</body>
```

```
</html>
```



# <HEAD>

```
<!DOCTYPE html>
```

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

```
<head>
```

```
<title> Titel des Dokuments</title>
```

```
</head>
```

```
<body>
```

```
<!-- Der Datensatz von German Nemirovski -->
```

```
<h1>Überschrift</h1>
```

```
<p>Absatz oder Paragraph</p>
```

```
</body>
```

```
</html>
```

Das Element beinhaltet Metadaten über das Dokument

# <BODY>

```
<!DOCTYPE html>
```

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

```
<head>
```

```
<title> Titel des Dokuments</title>
```

```
</head>
```

```
<body>
```

beinhaltet Dokumenteninhalte, der im Browserfenster sichtbar ist

```
<!-- Der Datensatz von German Nemirovski -->
```

```
<h1>Überschrift</h1>
```

```
<p>Absatz oder Paragraph</p>
```

```
</body>
```

```
</html>
```

# <HEADER>

```
<!DOCTYPE html>
<html lang="de">
  <head>
    <title> Titel des Dokuments</title>
  </head>
  <body>
    <!-- Der Datensatz von German Nemirovski -->
    <h1>Überschrift</h1>
    <p>Absatz oder Paragraph</p>

  </body>
</html>
```

Eine große Überschrift

<P>

```
<!DOCTYPE html>
<html lang="de">
  <head>
    <title> Titel des Dokuments</title>
  </head>
  <body>
    <!-- Der Datensatz von German Nemirovski -->
    <h1>Überschrift</h1>
    <p>Absatz oder Paragraph</p>

  </body>
</html>
```

Ein Mit Abstand unten und oben getrennter Absatz

# ATTRIBUTE

```
<!DOCTYPE html>
<html lang="de">
  <head>
    <title> Titel des Dokuments</title>
  </head>
  <body>

    <h1>Überschrift</h1>
    <p>Absatz oder Paragraph</p>
    Klicken Sie<a href="https://www.hs-albsig.de">hier</a>.

  </body>
</html>
```

Attribute können nur in einleitenden Tags verwendet werden  
Attribute bestehen aus einem Namen und einem Wert  
Attribute erscheinen nicht in der Ausgabe  
Attribute verfeinern Element-Definition

# VALIDATOREN

- Dokument auf die Einhaltung von Standards überprüfen
- W3C Markup Validator [validator.w3.org](https://validator.w3.org)
- nicht-valide Dokumente verlassen sich auf Fehlertoleranz des Browsers!

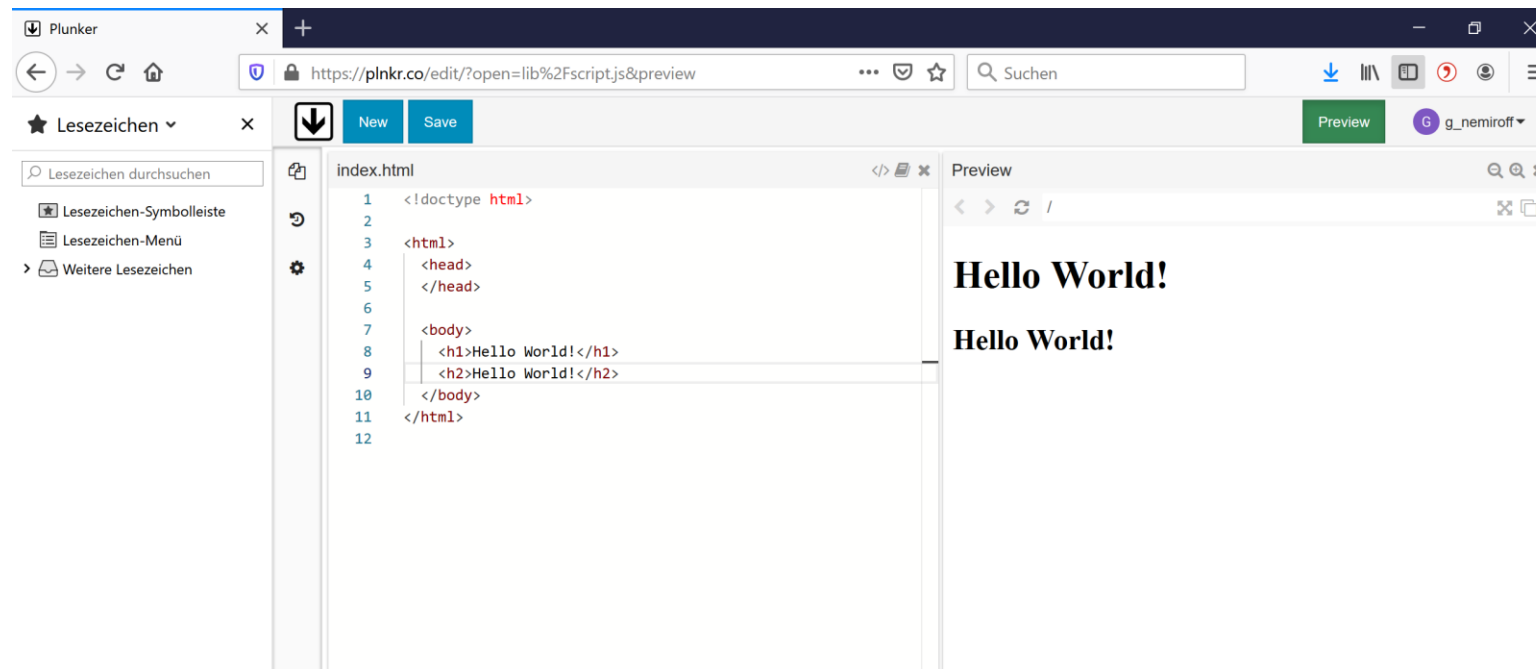
The screenshot shows the W3C Markup Validation Service interface in a web browser. The browser's address bar displays the URL [https://validator.w3.org/#validate\\_by\\_input](https://validator.w3.org/#validate_by_input). The page header features the W3C logo and the text "Markup Validation Service" with the subtitle "Check the markup (HTML, XHTML, ...) of Web documents". Below the header, there are three tabs: "Validate by URI", "Validate by File Upload", and "Validate by Direct Input". The "Validate by Direct Input" tab is selected. The main content area is titled "Validate by direct input" and contains a text box labeled "Enter the Markup to validate:". The text box contains the following HTML code: 

```
<!DOCTYPE html>
<html>
<head>
<title> Titel des Dokuments</title>
</head>
<body>
<h1>Überschrift</h1>
<p>Absatz oder Paragraph</p>
Klicken Sie<a href="https://www.meinserver.de">hier</a>.
</body>
</html>
```

 Below the text box, there is a link "More Options" and a "Check" button.

# HTML CODE PLAYGROUNDS

- Ermöglichen real time at place Ausführung vom Source-Code
- Rasche Fehlerkorrekturen mit dem direkten Ergebnis-Ansehen ist möglich
- Z.B. Plunker (plnkr.co)
- [https://en.wikipedia.org/wiki/Comparison\\_of\\_online\\_source\\_code\\_playgrounds](https://en.wikipedia.org/wiki/Comparison_of_online_source_code_playgrounds)



# HTML – BESCHREIBUNGSSPRACHE VON HYPERTEXT-DOKUMENTEN

- Struktur und Konvention
- HTML-Elemente
- Hypertext
- Tabellen
- Grafiken
- Formulare



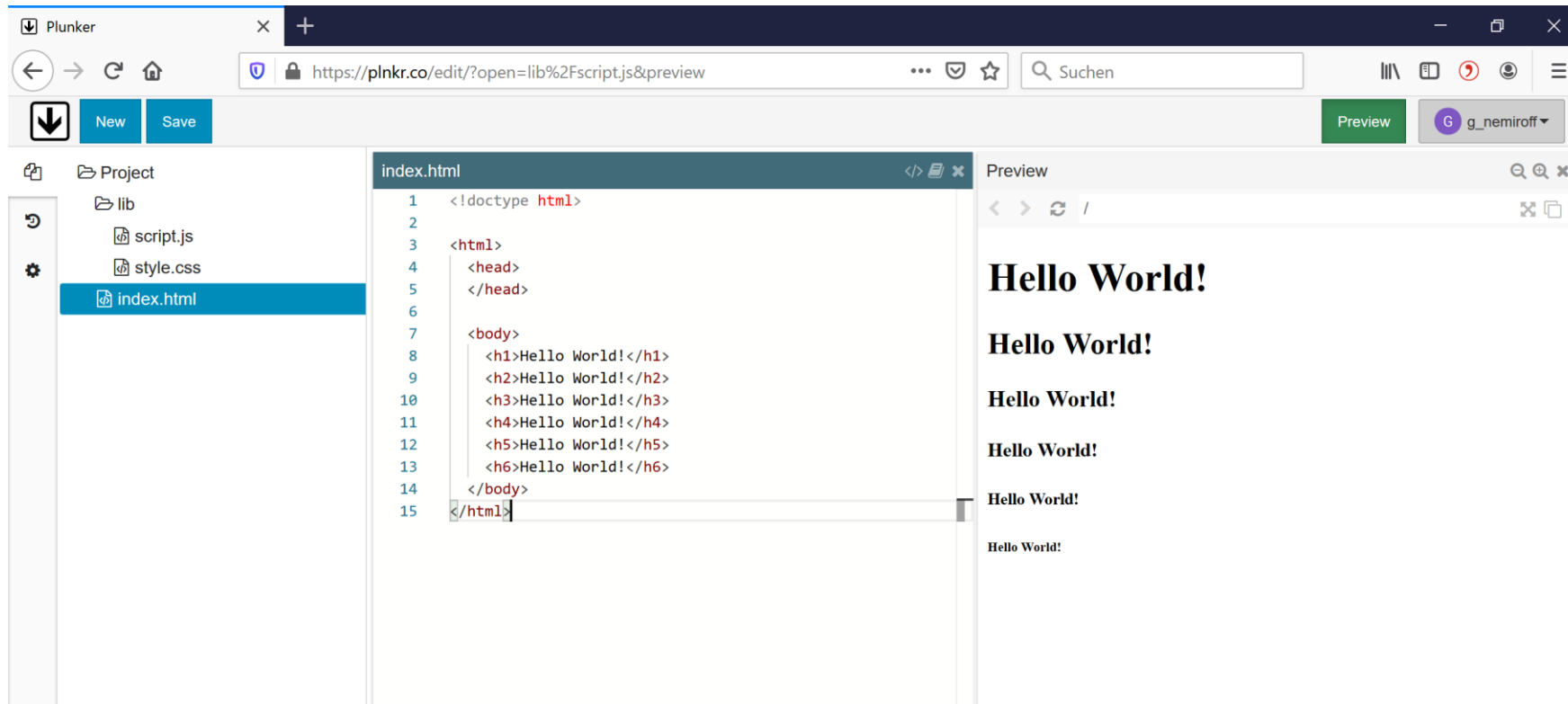
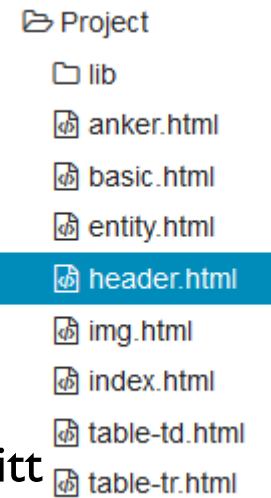
# ENTITÄTEN

- Project
  - lib
  - anker.html
  - basic.html
  - entity.html
  - img.html
  - index.html
  - table-td.html
  - table-tr.html

Entität	UNICODE DEC	UNICODE HEX	Zeichen
&lt;			<
&auml;	&#228;	&#x00E4;	Ä
&copy;	&#169;	&#x00A9;	©
&gt;			>
&Auml;	&#196;	&#x00C4;	Ä
&frac12;	&#189;	&#x00BD;	1/2
&quot;	&#171;	&#x00AB;	“
&Ouml;	&#214;	&#x00D6;	Ö
&divide;	&#247;	&#x00F7;	÷

# HEADER

- Auszeichnungselement für Überschriften
- Elementname “h” von “header“
- Sechs mögliche Verschachtelungsebenen (h1 bis h6) z. B.: Kapitel, Abschnitt, Unterabschnitt



# TEXTABSÄTZE

- Browser ignorieren Formatierungen im Quelltext
  - Leerzeichen, Tabs, neue Zeile werden zu einem einzelnen Leerzeichen zusammengefasst
- Absätze müssen mit einem Element ausgezeichnet werden
- Elementname “p“ von “paragraph“

## Project

### lib

anker.html

basic.html

entity.html

header.html

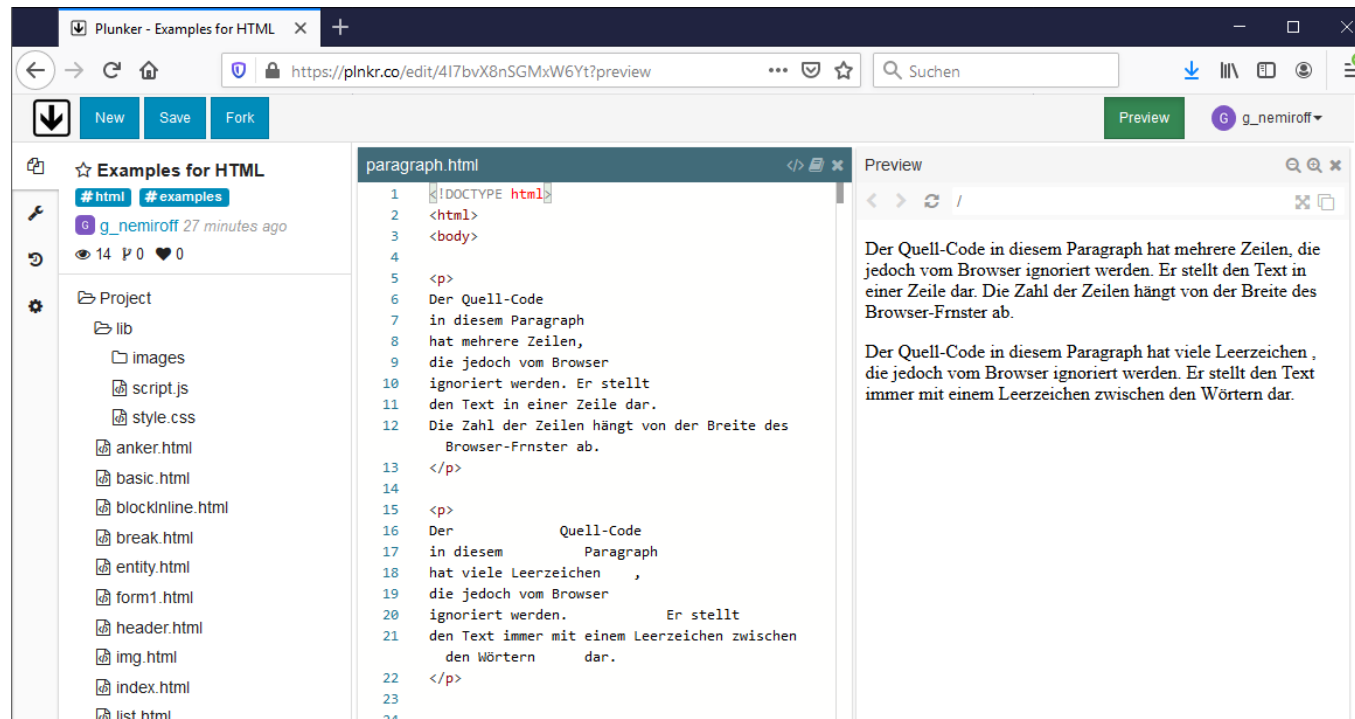
img.html

index.html

paragraph.html

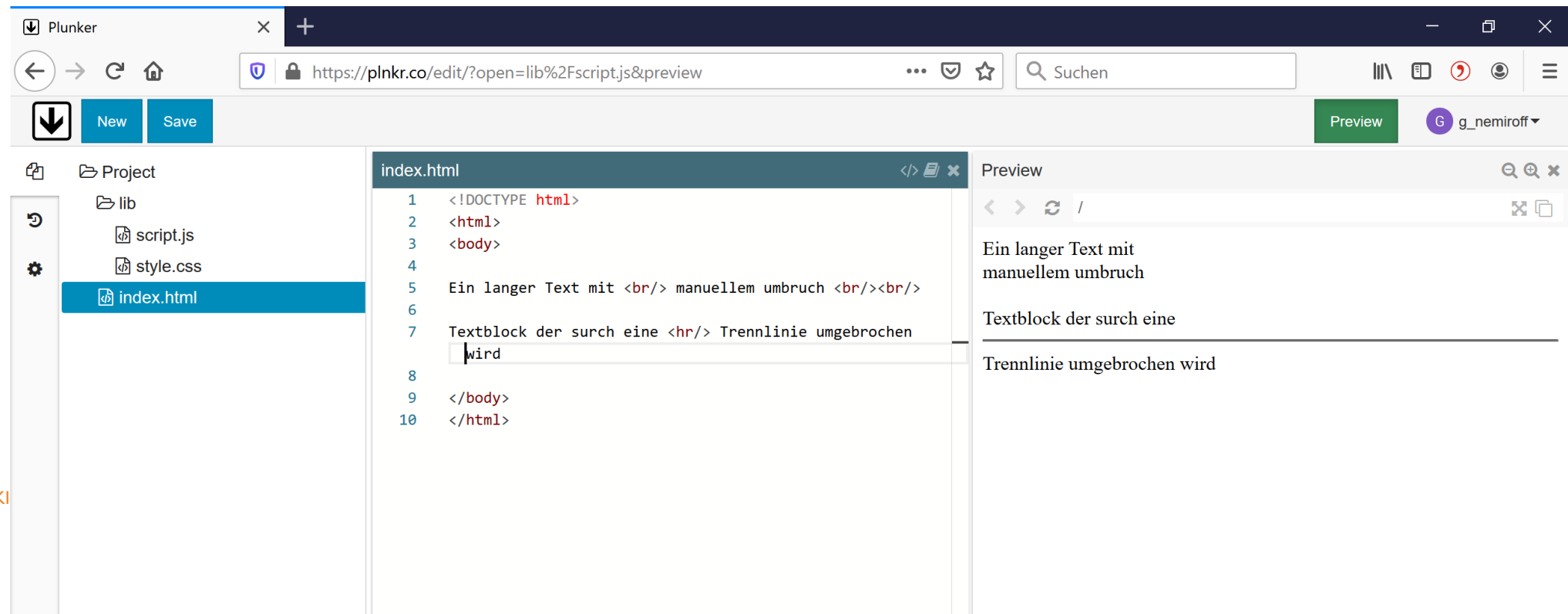
table-td.html

table-tr.html



# ZEILENUMBRUCH

- Browser brechen Absätze automatisch um
- HTML-Autor kann Zeilen manuell umbrechen: Elementname “br” von “line break”
- Alternativ: sichtbare Trennlinie über gesamte Textbreite: Elementname “hr” von “horizontal rule”



The screenshot shows the Plunker web editor interface. The browser address bar displays `https://plnkr.co/edit/?open=lib%2Fscript.js&preview`. The left sidebar shows a file tree with `Project`, `lib`, `script.js`, `style.css`, and `index.html`. The main editor area shows the `index.html` file with the following code:

```
1 <!DOCTYPE html>
2 <html>
3 <body>
4
5 Ein langer Text mit <br/> manuellem umbruch <br/><br/>
6
7 Textblock der such eine <hr/> Trennlinie umgebrochen
  wird
8
9 </body>
10 </html>
```

The right sidebar shows the `Preview` of the HTML code. It displays the rendered output:

Ein langer Text mit  
manuellem umbruch

Textblock der such eine

---

Trennlinie umgebrochen wird

# LISTEN

- Aufzählungen können als Listen ausgezeichnet werden
  - Elementname “ul“ von “unordered list“ → Symbole
  - Elementname “ol“ von “ordered list“ → Ziffern
- Innerhalb der Listen Einträge als Unterelemente
  - Elementname “li“ von “list item“

break.html

entity.html

header.html

img.html

index.html

list.html

paragraph.html

table-td.html

table-tr.html

index.html

```
1  <!DOCTYPE html>
2  <html>
3  <body>
4
5  <ul>
6    <li>Kafee</li>
7    <li>Tee</li>
8    <li>Cola</li>
9  </ul>
10
11 <ol>
12   <li>Grün</li>
13   <li>Gelb</li>
14   <li>Rot</li>
15 </ol>
16
17
```

</>  x

Preview

- < > ↺ /
- Kafee
  - Tee
  - Cola
- 
1. Grün
  2. Gelb
  3. Rot

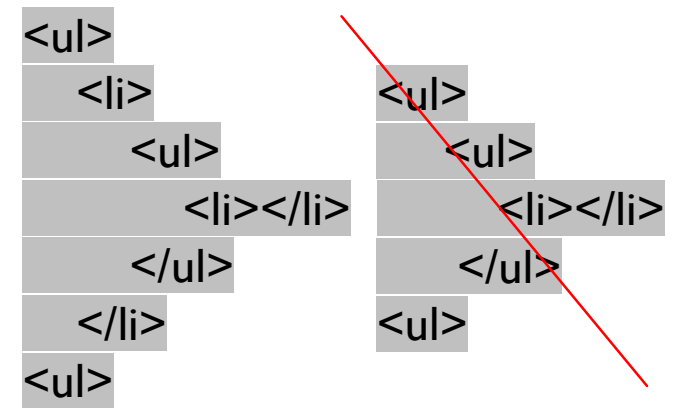
# LISTEN

- Aufzählungen können als Listen ausgezeichnet werden
  - Elementname “ul” von “unordered list” → Symbole
  - Elementname “ol” von “ordered list” → Ziffern
- Innerhalb der Liste Einträge als Unterelemente
  - Elementname “li” von “list item”
- Verschachtelungen sind möglich, nur so ----->
- type Attribut (1,a,A,I)

```
<ol type=„a“>
  <li>Kaffee</li>
  <li>Milch</li>
</ol>
```

- Beschreibungsliste

```
<dl>
  <dt>Kaffee</dt>
  <dd>- ein dunkles heißes Getränk</dd>
  <dt>Milch</dt>
  <dd>- weiss und kalt</dd>
</dl>
```



# LOGISCHE AUSZEICHNUNGEN

- ...sind die Elemente, welche einem Text eine “Bedeutung“ geben
- `<em>` von “emphasis“ → Betonung des Inhalts → kursive Darstellung
- `<strong>` von “strong emphasis“ → starke Betonung des Inhalts → fette Darstellung
- `<code>` → Inhalt ist Programmcode → Schreibmaschinenschrift
- `<cite>` → Beinhaltet eine Zitate oder einen Verweis → kursive Darstellung sources.
- `<samp>` → Beinhaltet eine Zitate oder einen Verweis → Schreibmaschinenschrift
- `<kbd>` → keyboard → Schreibmaschinenschrift
- `<var>` → Variable → kursive Darstellung.
- `<abbr>`
- `<acronym>`
- Verschachtelungen sind möglich

- 📄 break.html
- 📄 entity.html
- 📄 header.html
- 📄 img.html
- 📄 index.html
- 📄 list.html
- 📄 logicMarkups.html
- 📄 paragraph.html
- 📄 table-td.html
- 📄 table-tr.html

# PHYSISCHE AUSZEICHNUNGEN

- Elemente um den Stil eines Texts zu definieren
- `<i>` von “italic” → kursive Darstellung
- `<b>` von “bold” → fette Darstellung
- `<tt>` von “teletype” → Schreibmaschinenschrift
- Logische Elemente sollten bevorzugt werden
- Gestaltung sollte mittels CSS definiert werden

- break.html
- entity.html
- header.html
- img.html
- index.html
- list.html
- logicMarkups.html
- paragraph.html
- table-td.html
- table-tr.html

index.html

```
1 <!DOCTYPE html>
2 <html>
3 <body>
4 Text beginn
5   <em>emphasis-text</em>
6   <strong>strong-Text</strong>
7   <code>code-text</code>
8   <cite>citation</cite>
9   <samp>sample</samp>
10  <kbd>keyboard</kbd>
11  <var>variable</var>
```

Preview

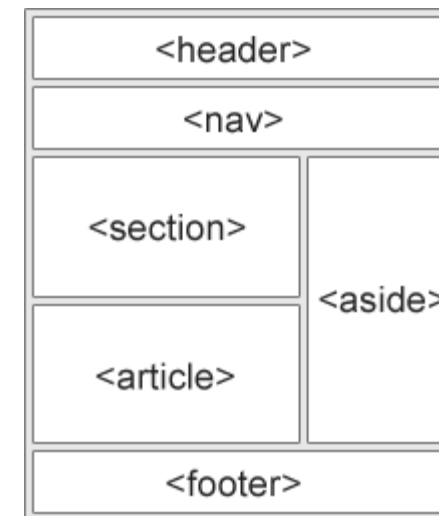
Text beginn *emphasis-text* **strong-Text** `code-text` *citation* sample  
keyboard *variable* typetext Text end



# SEMANTISCHE ELEMENTE (HTML 5)

- list.html
- logicMarkups.html
- paragraph.html
- semanticElements.html**
- table-td.html
- table-tr.html

- `<article>`
- `<aside>`
- `<details>`
- `<figcaption>`
- `<figure>`
- `<footer>`
- `<header>`
- `<main>`
- `<mark>`
- `<nav>`
- `<section>`
- `<summary>`
- `<time>`
- `<main>`



# SEMANTISCHE ELEMENTE (HTML 5)

- `<section>` Element definiert einen Abschnitt in einem Dokument, eine semantische (nach Bedeutung) Gruppierung des Inhaltes, normalerweise unter einer gemeinsamen Überschrift `<h1,2,3...>`. `<header>` ist nicht erforderlich. Z.B: Einführung, Inhalt, und Fazit.
- `<article>`-Element definiert unabhängiges Stück vom Inhalt. Zum Verständnis von diesem Inhalt wird keinen Anderen Inhalt auf der Webseite erforderlich. Z.B. Forum-Post, Blog-Post, Zeitungsartikel. Wenn alle Elemente außerhalb `<article>` gelöscht werden soll der Inhalt wird immer noch sinn machen.
- `<article>` kann in einer `<sections>` nesten und eine `<section>` in einem `<article>`. Ein Buch, z.B., kann in `<section>` (Kapitel) aufgeteilt werden, jede `<section>` würde mehrere `<article>` umfassen. Jeder `<article>` kann mehrere `<section>` haben.
- Header/ Footer bilden den Rahmen einer Webseit, z.B. mit der Navigation, Kontakt, Impressum, aber auch den Rahmen von semantischen Elementen, z.B. eines Artikles.
- Aside stellt zusätzliche Information am Rande des Browser Fensters dar.


NEMIROVSKI


- list.html
- logicMarkups.html
- paragraph.html
- semanticElements.html**
- table-td.html
- table-tr.html

```
<!DOCTYPE html>
<html>
<body>
  <header>
    <nav>
      <ul>
        <li><a href="#">link</a></li>
        <li><a href="#">link</a></li>
        <li><a href="#">link</a></li>
      </ul>
    </nav>
    <h1>
      Page Title
    </h1>
  </header>


  <section>
    <h2>
      My Blog Posts
    </h2>
    <article>
      <header>
        <p>
          Article Title
        </p>
      </header>
      <p>
        content
      </p>
    </article>
    <article>
      <header>
        <p>
          Article Title
        </p>
```


# ALLGEMEINE AUSZEICHNUNGEN

 anker.html

 basic.html

 blockInline.html

 break.html

 entity.html

- Allgemeine Auszeichnungen
  - umschließen Texte
  - (noch) keine Semantik, nützlich für CSS und JavaScript
- Allgemeines Block-Element
  - Elementname `<div>` von “division“
- Allgemeines Inline-Element
  - Elementname `<span>`

# BLOCK/INLINE

anker.html  
basic.html  
blockInline.html  
break.html  
entity.html

- Alle Elemente von HTML gehören zu einer Kategorie
  - Block → Erzeugt einen rechteckigen Bereich
  - Inline → Erzeugt einen Bereich, der dem Textfluss folgt
- Verschachtelungen sind möglich
- Aber: Inline-Elemente dürfen keine Block-Elemente enthalten

```
index.html  </> [icon] x
1  <!DOCTYPE html>
2  <html>
3  <body>
4
5  Dieser Satz zeigt, <div>wie unterschiedlich
    sich Block- und Inline-Elemente </div>
    verhalten.
6  <br/><br/>
7  Dieser Satz zeigt, <span>wie unterschiedlich
    sich Block- und Inline-Elemente </span>
    verhalten.
8  <br/><br/>
9
10 Dieser Satz zeigt, <div>wie
    <span>unterschiedlich</span> sich Block- und
```

Preview

< > ↺ / [icon] [icon]

Dieser Satz zeigt,  
wie unterschiedlich sich Block- und Inline-Elemente  
verhalten.

Dieser Satz zeigt, wie unterschiedlich sich Block- und Inline-  
Elemente verhalten.

Dieser Satz zeigt,  
wie unterschiedlich sich Block- und Inline-Elemente  
verhalten.

# BLOCK/INLINE

- Weitere Block-Elemente

<address>	<article>	<aside>	<blockquote>	<canvas>	<dd>
<div>	<dl>	<dt>	<fieldset>	<figcaption>	<figure>
<footer>	<form>	<h1>-<h6>	<header>	<hr>	<li>
<main>	<nav>	<noscript>	<ol>	<p>	<pre>
<section>	<table>	<tfoot>	<ul>	<video>	

# BLOCK/INLINE

- Weitere Inline-Elemente

<a>	<abbr>	<acronym>	<b>	<bdo>	<big>
 	<button>	<cite>	<code>	<dfn>	<em>
<i>	<img>	<input>	<kbd>	<label>	<map>
<object>	<output>	<q>	<samp>	<script>	<select>
<section>	<span>	<strong>	<sub>	<sup>	<textarea>
<time>	<tt>	<var>			

# UNIVERSALATTRIBUTE

- Attribut “id”
  - Individueller Name des Elements `<p id="x"> ...</p>`
    - Muss in der HTML-Seite eindeutig sein
- Attribut “title” `<p title="ttttttt"> ...</p>`
  - kurze individuelle Beschreibung des Elements
  - wird als Tooltip angezeigt
- Attribut “style” und “class” `<p style="color:red"> ...</p>`
  - CSS spezifische Attribute, Erklärung später

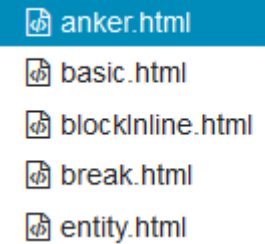
# HTML – BESCHREIBUNGSSPRACHE VON HYPERTEXT-DOKUMENTEN

- Struktur und Konvention
- HTML-Elemente
- Hypertext
- Tabellen
- Grafiken
- Formulare





# VERWEISE



anchor.html  
basic.html  
blockInline.html  
break.html  
entity.html

- Erlauben die Verknüpfung von einem HTML-Dokument auf ein anderes
- Sind unidirektional (kein zurück)
- Bestehen aus:
  - Verweistext
  - Verweisziel
- Elementname `<a>` von “anchor”
- Inhalt des Elements ist der Verweistext `<a>inhalt</a>`
- Attribut “href” von “hypertext reference” ist die URL des Verweisziels `<a href=„...“>`
  - relativ `href=„../index.html“`
  - absolut `href=„/home/user2/index.html“`
  - extern (URL) `href=„http://www.hs-albsig.de/index.html“`
  - Email `href=„mailto:nemirovski@hs-albsig.de“`

# UNIFORM RESSOURCE LOCATOR (URL)

anker.html

basic.html

blockInline.html

break.html

entity.html

`<a href=„https://www.hs-albsig.de:80/studienangebot/bachelorstudiengaenge/it-security-studium#top“>hier klicken</a>`

https://www.hs-albsig.de:80/studienangebot/bachelorstudiengaenge/it-security-studium#top

Protokoll      Host      Port      Pfad      Element ID

# ANKER

anker.html

basic.html

blockInline.html

break.html

entity.html

- Verweise können auch auf bestimmte Stellen auf Seiten verweisen
- Anker müssen definiert werden
- Ebenfalls mittels Element “a”
  - Attribut “id” definiert Ziel einer Sprungstelle

index.html

```
7
8  <h1>Übersicht</h1>
9  <p><a href="#section1">Einführung</a><br/>
10 <a href="#section2">Grundlagen</a><br/>
11 <a href="#section2.1">Vertiefung</a><br/>
12 <br/>
13 <br/>
14 <br/>
15 <br/>
16 <br/>
17 <br/>
18 <br/>
19 <br/>
20 <br/>
21 <br/>
22 <br/>
23 <br/>
24 <br/>
25 <br/>
26 <br/>
27 <h2 id="section1">Einführung</h2>
28 ...section 1...<br/>
29 ...section 1...<br/>
```

Preview

## Übersicht

[Einführung](#)

[Grundlagen](#)

[Vertiefung](#)

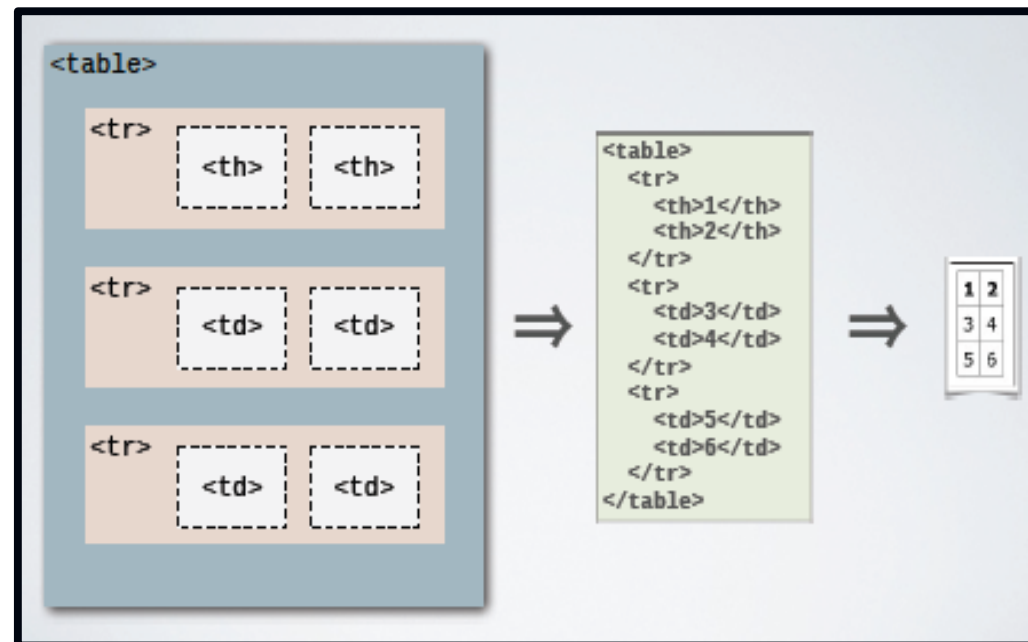
# HTML – BESCHREIBUNGSSPRACHE VON HYPERTEXT-DOKUMENTEN

- Struktur und Konvention
- HTML-Elemente
- Hypertext
- Tabellen
- Grafiken
- Formulare

# TABELLEN

[logicMarkups.html](#)  
[paragraph.html](#)  
[semanticElements.html](#)  
[table-td.html](#)  
[table-tr.html](#)

- Element `<table>` → Definition einer Tabelle → umfasst keinen Text, nur Unterelemente
- Element `<tr>` von “table row” → Definition einer Tabellenzeile → auch nur Unterelemente
- Element `<td>` von “table data” → Definition einer Tabellenzelle → umfasst einen Text
- Element `<th>` von “table header” → Titelizele → Spezialform von `<td>` → zentriert und fett



# TABELLENELEMENTE

[logicMarkups.html](#)

[paragraph.html](#)

[semanticElements.html](#)

[table-td.html](#)

[table-tr.html](#)

## Attribute:

### width

Tabellenbreite

### border

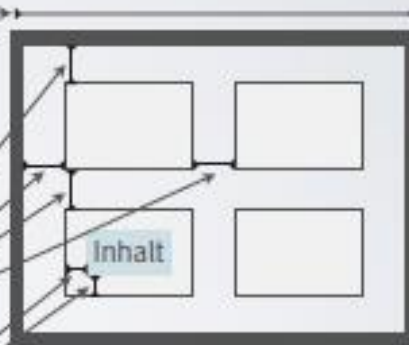
Rahmenbreite

### cellspacing

Abstand zwischen Zellen

### cellpadding

Abstand Zelle zu Inhalt



```
<table border="5"
        cellspacing="12"
        cellpadding="1">
  <tr>
    <th>1</th>
    <th>2</th>
  </tr>
  <tr>
    <td>3</td>
    <td>4</td>
  </tr>
</table>
```



```
<table border="5"
        cellspacing="1"
        cellpadding="9">
  <tr>
    <th>1</th>
    <th>2</th>
  </tr>
  <tr>
    <td>3</td>
    <td>4</td>
  </tr>
</table>
```



<TR>

- Attribute:
  - align = left | center | right  
Richtet Zellen einer Zeile horizontal aus
  - valign = top | middle | bottom  
richtet Zellen einer Zeile vertikal aus

# <TD>

- logicMarkups.html
- paragraph.html
- semanticElements.html
- table-td.html
- table-tr.html

- Attribute:
  - align = left | center | right - richtet Zellen einer Zeile horizontal aus
  - valign = top | middle | bottom - richtet Zellen einer Zeile vertikal aus
  - width Breite der Zelle (und damit der Spalte) in Pixel
  - colspan - Anzahl der Spalten, über die sich eine Zelle erstreckt
  - rowspan - Anzahl der Zeilen, über die eine die Zelle erstreckt

```
table-td.html
1  <!doctype html>
2
3  <html>
4    <head>
5    </head>
6    <body>
7
8    <table border="1" width="100">
9      <tr>
10       <th>1</th>
11       <th colspan="2">2</th>
12     </tr>
```

Preview

1	2	
3	4	5
	6	7 8



# HTML – BESCHREIBUNGSSPRACHE VON HYPERTEXT-DOKUMENTEN

- Struktur und Konvention
- HTML-Elemente
- Hypertext
- Tabellen
- **Grafiken**
- Formulare

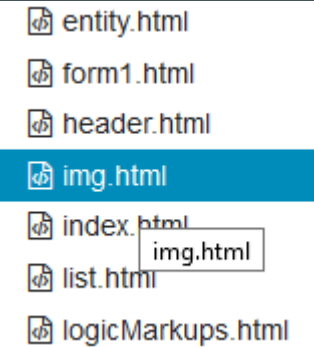
# GRAFIKELEMENT

- Elementname `<img>` von “image“
- Attribute “src“ von “source“ definiert Bild-URL
- Attribute “alt“ von “alternative“ definiert Alternativtext
  - Falls Bild nicht geladen werden kann
  - Für barrierefreie Webseiten
- HTML definiert keine Grafikformate
  - unterstützt werden: GIF, PNG, JPG

- entity.html
- form1.html
- header.html
- img.html
- index.html
  - img.html
- list.html
- logicMarkups.html

# GRAFIKELEMENT MIT STYLE

- Größe mittels “width“ und “height“
  - sollten immer angegeben bzw. über CSS gesetzt werden
- Mit dem Style –Attribute richtig die Größe setzen
  - `<img src=„chirch.gif“ alt=„eine Kirche“ style="width:50px;height:75px;">`



entity.html  
form1.html  
header.html  
img.html  
index.html  
list.html  
logicMarkups.html

# IMAGE MAP

- entity.html
- form1.html
- header.html
- img.html
- index.html
- list.html
- logicMarkups.html

- Ein image map ist ein image mit Bereichen die der Benutzer anklicken kann.

```

```

```
<map name="windows">
```

```
  <area shape="rect" coords="40,70,100,200" alt="Musik" href="https://www.gorillaz.com/">
```

```
  <area shape="rect" coords="120,70,200,200" alt="Spass"
    href="https://www.youtube.com/watch?v=H333s_d6MYE">
```

```
  <area shape="circle" coords="250,150,40" alt="Kunst"
    href="https://strassenkunst.piyafo.com/exterior-dabsmyla/">
```

```
</map>
```

## Bereiche:

rect – Rechteck

circle - Kreis

poly - Polygon

default – das ganze Image

# PICTURE

- Das Element `<picture>` umfasst mehrere `<source>` -Elemente. Jedes Element referenziert jeweils ein Image. Der Browser selektiert ein Source in Abhängigkeit von der aktuellen Präsentationsbedingungen, zum Beispiel von der Browse-Fensterbreite.

```
<picture>
  <source media="(min-width: 650px)" srcset="chirch1.jpg">
  <source media="(min-width: 465px)" srcset="chirch2.jpg">
  
</picture>
```

# HTML – BESCHREIBUNGSSPRACHE VON HYPERTEXT-DOKUMENTEN

- Struktur und Konvention
- HTML-Elemente
- Hypertext
- Tabellen
- Grafiken
- **Formulare**



# FORMULARE

- blockInline.html
- break.html
- entity.html
- form1.html
- header.html
- img.html
- index.html

- Formularbereich: Element `<form>`
  - Attribut “action“ definiert Ziel-URL für die Formulardaten
  - Attribut “method“ definiert HTTP-Request-Methode
    - GET oder POST
- Zur Auswertung muss unter der Ziel-URL ein Service stehen der die Daten verarbeitet


# GET ODER POST


GET	POST
Daten werden als Parameter in der URL übermittelt.	Daten werden im Body des Requests übermittelt.
Daten müssen kodiert werden. Hallo Günter → Hallo+G%Fcnter	Daten können binär übertragen werden.
Länge der URL (und damit der Daten) ist beschränkt.	Daten können beliebig lang sein.
URL enthält Adresse der Webseite und Daten der Eingabe, bleiben in Bookmarks und Emails enthalten.	URL enthält nur die Adresse der Webseite, Eingaben werden in Bookmarks und Emails nicht gespeichert.




# FORMULARE


- Element `<input ...>` zur Definition eines Eingabefeldes
- Attribut `“name“` zur Definition des Parameternamens
- Attribut `“type“` um eine Variante auszuwählen:
  - `“text“` → einzeliges Eingabefeld
  - `“password“` → einzeliges verdecktes Eingabefeld
  - `“file“` → Feld für das Hochladen von lokalen Dateien
  - `“radio“`
  - `“checkbox“`
  - `“button“`
  - `“reset“`
  - `“submit“`


 blockInline.html


 break.html

 entity.html

 form1.html

 header.html

 img.html

 index.html

# INPUT-TYPEN EINGABEFELDER

- blockInline.html
- break.html
- entity.html
- form1.html
- header.html
- img.html
- index.html

- `<input type="text" ...>`
  - Attribut "size" bestimmt die Länge des Eingabefeldes
  - Attribut "maxlength" bestimmt die maximale Länge der Eingabe
  - Attribut "value" bestimmt die Vorbelegung des Eingabefeldes
- `<input type="password" ...>`
  - Gleiche Attribute wie einzeliges Texteingabefeld
  - Eingabetext wird nur im Browser verschleiert
  - In der Netzwirkommunikation unverschlüsselt
- `<input type="file" ...>`
  - Attribut "accept" um den Medientyp einzuschränken
    - text/plain → Textdatei
    - image/jpeg → JPEG-Datei
  - Aussehen sehr browserspezifisch
    - häufig nur ein Eingabeknopf
  - Nur mit Post-Formularen
- `<textarea ...>`
  - Mehrzeiliges Texteingabefeld
  - Attribut "rows" bestimmt die Anzahl der Zeilen
  - Attribut "cols" bestimmt die Anzahl der Spalten (Zeichen pro Zeile)

# INPUT-TYPE TEXT ATTRIBUT PATTERN

blockInline.html

break.html

entity.html

form1.html

header.html

img.html

index.html

- `<input type="text" pattern="...">`

Attribut "pattern" bestimmt mithilfe regulären Ausdrücken die für das aktuelle Feld erlaubte Eingabe (<https://html.spec.whatwg.org/multipage/input.html#attr-input-pattern>) und (<https://tc39.es/ecma262/#sec-regexp-regular-expression-objects>)

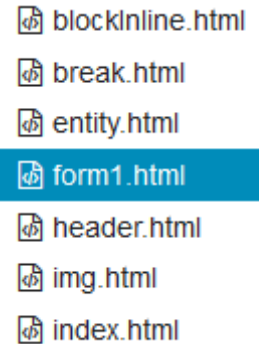
So kann man die Eingabe auf drei beliebige Buchstaben (klein oder groß) einschränken

```
<form action="/action_page">
  <input type="text" id="Amerikanische PLZ" name="country_code"
    pattern="[A-Za-z]{3}" title="drei Buchstaben klein oder gross"><br><br>
  <input type="submit">
</form>
```

So kann man die Eingabe auf fünf beliebige Zahlen einschränken

```
<form action="/action_page">
  <input type="text" id="PLZ" name="deutsche_PLZ"
    pattern="[0-9]{5}" title="fünf Zahlen"><br><br>
  <input type="submit">
</form>
```

# INPUT-TYPEN AUSWAHLELEMENTE



blockInline.html  
break.html  
entity.html  
form1.html  
header.html  
img.html  
index.html

- **Auswahllisten**
  - Aus einer Liste einen oder mehrere Einträge auswählen
  - Element `<select>` definiert Liste
  - Attribut `"name"` definiert den Parameternamen
  - Attribut `"size"` definiert die Menge der angezeigten Einträge
    - `"size" > 1` → Drop-Down-Liste
  - Attribut `"multiple"` erlaubt die Mehrfachauswahl
  - Element `"option"` definiert Einträge
  - Attribut `"value"` zur Definition eines Rücksendewerts
  - Attribut `"selected"` zur Vorselektierung
  - Beliebig viele Einträge in der Auswahlliste
- **Auswahlfelder**
  - "Radio Buttons" → Eins aus einer Auswahl an Feldern
    - `<input type="radio" ...>`
  - "Checkbox" → Mehrere aus einer Auswahl an Feldern
    - `<input type="checkbox" ...>`
  - Attribut `"value"` zur Definition eines Rücksendewerts
  - Attribut `"checked"` zur Vorselektierung
  - Auswahlfelder werden mit dem Attribut `"name"` gruppiert

# EINGABE-BUTTONS

- blockInline.html
- break.html
- entity.html
- form1.html
- header.html
- img.html
- index.html

- Absende-Button
  - `<input type="submit" ...>`
  - Attribut "value" bestimmt die Beschriftung
- Rückstell-Button
  - `<input type="reset" ...>`
- Button (simple)
  - `<input type="button" ...>`
  - Erst mit JavaScript nutzbar

# BESCHRIFTUNGEN

- Element `<label>` erlaubt Beschriftung von Eingabefeldern
- Attribut „for“ definiert die Verbindung zum Eingabefeld
  - muss auf die „id“-Attribut eines Eingabefelds verweisen
- Bei einem Mausklick auf die Beschriftung wird das Eingabefeld fokussiert/ausgewählt
  - Hilfreich z. b. bei Checkboxes

blockInline.html

break.html

entity.html

form1.html

header.html

img.html

index.html

# GRUPPIERUNGEN

- Zusammengehörige Eingabefelder gruppieren
  - mit einem Rahmen und einer Überschrift versehen
- Element `<fieldset>` umschließt die Gruppe
- Element `<legend>` definiert die Beschriftung des Rahmens

blockInline.html

break.html

entity.html

form1.html

header.html

img.html

index.html