



Kontakt, Skripte, Übungen, Medien

Skripte, Aufgabenblätter:

\\ads\dfs\imm\Public\Professoren\Schober Public\T3B250-Internet-Technologie\

Vorlesungsskripte im PDF-Format

Übungen HTML, CSS, JavaScript, PHP, MySQL

Kontakt:

Prof. Dipl.-Ing. Martin Schober

e-Mail: martin.schober@hs-karlsruhe.de

Lerninhalte – Behandelte Technologien

HTML5

CSS (Cacading StyleSheets (Layout))

Adobe Dreamweaver (Editor für HTML, CSS, Javascript, PHP)

Javascript (Programmiersprache für Client-Seitige Interaktionen)

PHP (Programmiersprache für Server-Seitige Interaktionen)

MySQL (Datenbank)

AJAX → Asynchronous JavaScript + XML, wichtig im Zusammenhang mit Web 2.0

Klausur und geforderten Zeitaufwand für 5CP's

EDV-Bez.	Lehrveranstaltung	Sem.	CP	SWS	Art	Voraus.	SL/ PV	Dauer	PL	Dauer	GFN	FP	Bem.
T3B250	Internet-Technologien	2	5	4	V,Ü		Ü	1S	KI	90		08	
T3B251	Internet-Technologien		3	2	V				KI	90			
T3B252	Internet-Technologien, Übung		2	2	Ü		Ü	1S	8 bPL 1 üPL	60		8 FP	

Klausur Internet-Technologien 90 Min.

Prüfungsvorleistung: Erfolgreiche Teilnahme an den Übungen.

Die Prüfung

KURSBEZEICHNUNG: Internet-Technologien		LN – NUMMER: T2B251
DOZENT: Prof. Dipl.-Ing. Martin Schober		
NAME:	MATRIKELNR:	DATUM: 02.02.2009

PRAKTISCHER TEIL DER PRÜFUNG. ZEIT: 120 MINUTEN FÜR 4 AUFGABEN

Bitte beachten Sie, dass Sie eine Aufgabe streichen müssen. Sie dürfen nur vier Aufgaben abgeben.

Aufgabe 1 (HTML/CSS) (8 Punkte)

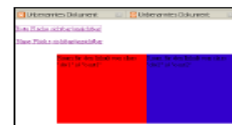
1. Erzeugen Sie eine HTML-Seite. Legen Sie mit CSS einen Navigationsbereich, einen Inhaltsbereich und einen Bannerbereich an. Der Navigationsbereich soll links sitzen. Der Bannerbereich soll oben sitzen. Der Inhaltsbereich soll rechts vom Navigationsbereich und unter dem Bannerbereich sitzen.
2. Der Bannerbereich soll 800 Pixel breit und 200 Pixel hoch sein und die Hintergrundfarbe rot erhalten. Der Navigationsbereich soll 200 Pixel breit und 400 Pixel hoch sein und die Hintergrundfarbe grün erhalten. Die Hintergrundfarbe des Inhaltsbereichs soll grau sein. Der gesamte Anzeigebereich soll in der Breite 800 Pixel und in der Höhe 600 Pixel groß sein. Richten Sie die Web-Seite mittig aus.
3. Stellen Sie den Abstand zum oberen Fensterrand auf 15 Pixel ein. Stellen Sie die Abstände der Texte innerhalb der div-Container zum div-Container-Rand auf 15 Pixel ein.
4. Setzen Sie die Hintergrundfarbe der Seite mittels CSS auf blau. Fügen Sie einen Fußbereich am unteren Ende ein, der 15 Pixel hoch und 800 Pixel breit ist. Der Abstand des Textes zum Rand soll 0 Pixel sein. Die Gesamthöhe der Webseite darf 600 Pixel nicht überschreiten. Die Gesamtbreite der Webseite darf 800 Pixel nicht überschreiten.



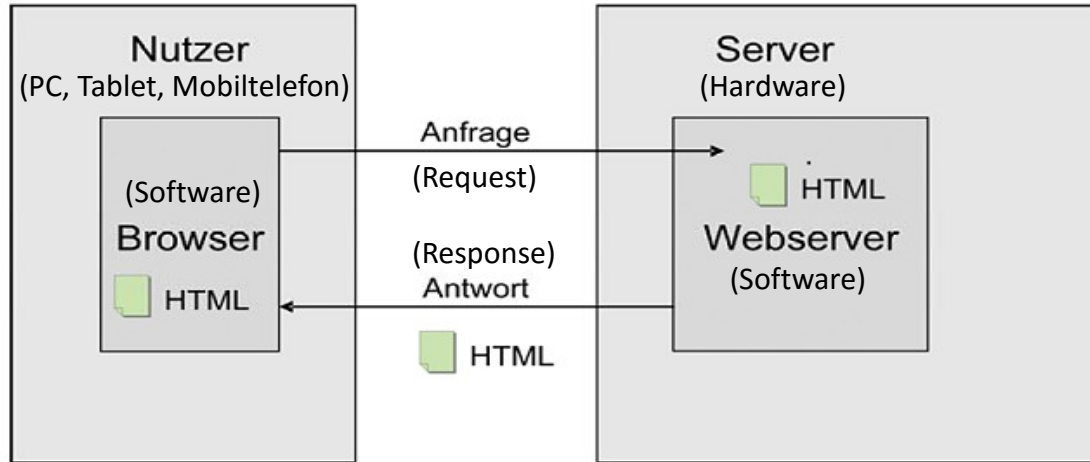
Speichern Sie die Dateien (A1.htm und A1.css) auf dem Rechner im Verzeichnis D:\htdocs\Ihr_Name\A1.

Aufgabe 2 HTML/CSS/Javascript (8 Punkte)

1. Erzeugen Sie eine HTML-Seite. Legen Sie mit CSS zwei Bereiche (`<div>`) mit der Größe 200x200 Pixel an. Die Bereiche sollen nebeneinander sitzen und zwei verschiedene Hintergrundfarben haben.
2. Erzeugen Sie zwei Text-Links, um mit einer Javascript-Funktion die div-Container unsichtbar zu machen. (Tipp: ... `.style.visibility = ...`)
3. Sind die Container unsichtbar, so sollen sie beim erneuten Klick auf den entsprechenden Link wieder sichtbar werden.



Client-Server-Modell



Client-Server-Modell - Definitionen

Server (*deutsch*: Bediener, Anbieter, Dienstleister, *englisch*: to serve)

Ein Server ist ein Programm (Prozess), das mit einem anderen Programm (Prozess), dem Client (*deutsch*: Kunde), kommuniziert, um ihm Zugang zu einem Dienst zu verschaffen. Hierbei muss abgrenzend beachtet werden, dass es sich bei "Server" um eine Rolle handelt, nicht um einen Computer an sich. Ein Computer kann nämlich ein Server und Client zugleich sein, siehe: Peer-to-Peer.

Client (*deutsch*: Kunde, Dienstnutzer)

Ein Client kann einen Dienst bei dem Server anfordern, welcher diesen Dienst bereitstellt.

Dienst (*englisch*: Service)

Vereinbarung einer festgelegten Aufgabe, die der Server anbietet und der Client nutzen kann.

Request (*deutsch*: Anforderung, Anfrage)

Anforderung eines Clients an den Server, dessen Dienst er benötigt.

Response (*deutsch*: Antwort)

Antwort eines Servers auf eine Anforderung eines Clients.

vgl. Wikipedia, <http://de.wikipedia.org/wiki/Client-Server-Modell>, Zugriff 13.03.2014

Domains, URLs und IP-Adressen

http:// www . hs-karlsruhe . de / support / index.htm

Protokoll Subdomain Domain Toplevel Verzeichnis Datei

Domain: technischeredaktion.com

IP-Adresse: 213.174.55.61

Die Verbindung von IP-Adresse und Domain erledigen sogenannte Domain Name Server (DNS).

Festlegung einer Ordnerstruktur (Server / lokal)

Welche Dateiformate werden eingesetzt?

HTML-Dateien (*.htm, *.html)

Pixelbilder (*.gif, *.jpg, *.png)

Vektorgrafik(*.svg)

CSS-Dateien (*.css)

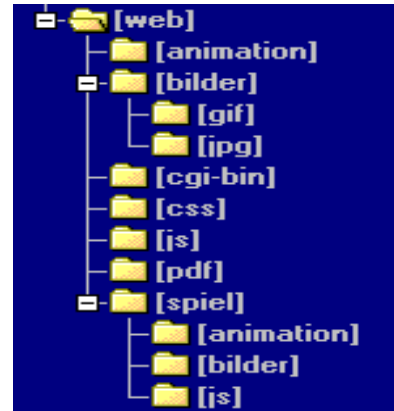
Javascript (*.js)

Videos (*.mp4)

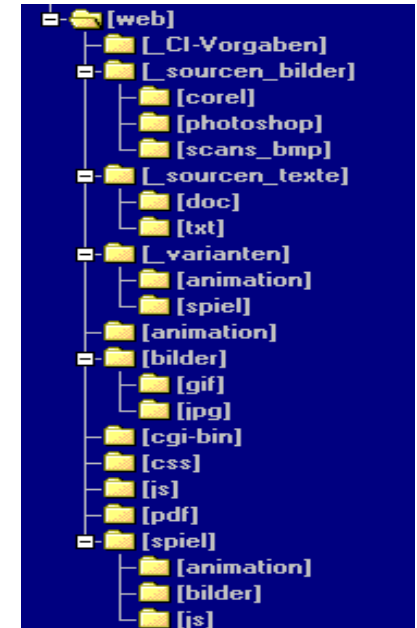
Sounds (*.wav, *.mpeg3)

weitere (*.pl, *.cgi, *.php)

Ordnerstruktur Web-Server



Ordnerstruktur Arbeitsplatz-PC



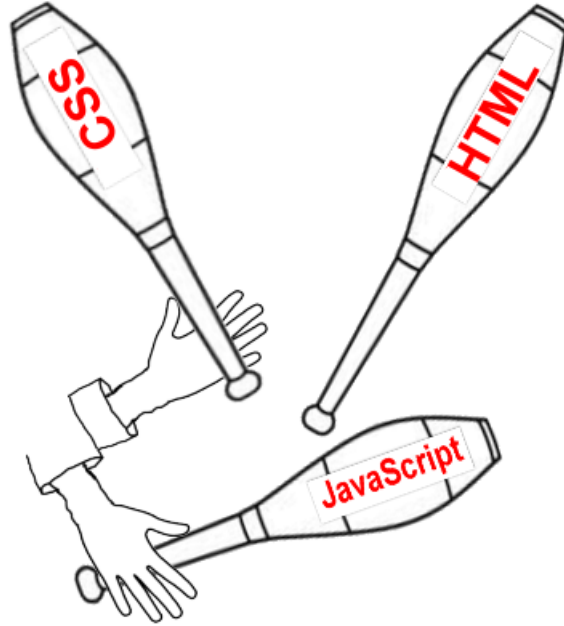
HTML5 → Wichtige Erweiterung für TR und E-Learning!



Lesen Sie den Artikel: [Mobil, mehrsprachig und multimedial](#)

HTML5 - Basistechnologien

HTML5 =



- + neue Tags
- + neue Styles
- + neue JS-Methoden

Interessante HTML5-Neuerungen für die TD

HTML5 wird von einer Arbeitsgruppe ([Microsoft](#), Apple, Mozilla, Opera und Adobe) betrieben und vom W3C toleriert und anerkannt.

Neuerungen von HTML5, die sich für die Technische Dokumentation eignen:

- Native Unterstützung von Audio- und [Videodateien](#).
- [Canvas](#): Grafiken mit [JavaScripts](#) erstellen und animieren.
- [SVG](#) kann über ein <svg>-Tag einfach eingebunden werden. (Neue Animationsbibliothek [snap.svg](#))
- Native [3D-Unterstützung](#) mit [WebGL](#) (Mozilla, Google, AMD, Nvidia, ...).
- [WebVR](#) (JavaScript u. Three.js) ist ein Open-Source-Standard um VR- und 360°-Inhalte auf Browsern (auch mobil) darzustellen.

Webseite, Website und Homepage

Eine Webseite ist ein HTML-Dokument, also eine Seite aus einem Webangebot.

Die Website ist der Begriff für das komplette Webangebot.

Die Homepage bezeichnet die Eingangsseite des Webangebots, wird aber häufig auch als Gattungsbegriff für das gesamte Webangebot verwendet.

CSS → Cascading Style Sheets

Technologie wird wie HTML/XHTML auch vom W3C verwaltet

CSS ist eine Sprache um Layout-Befehle zu definieren

Diese werden den Tags und Bereichen im HTML-Dokument zugewiesen

Layout von Inhalt und Struktur trennen

```
h1 {  
    color: red;  
}
```

Javascript

Relevant in allen Browsern, im PDF und SVG, in Adobe Captivate und Adobe Animate CC

JavaScript wird z.B. verwendet um

- neue Fenster ohne Menüleiste zu öffnen (PopUps)
- Formulare auf Vollständigkeit zu prüfen
- Interaktionen mit dem Nutzer zu ermöglichen

JavaScript spielt bei Web2.0/AJAX eine wichtige Rolle

JavaScript spielt bei HTML5 eine wichtige Rolle

Serverseitige Technologien

PHP

ASP.NET

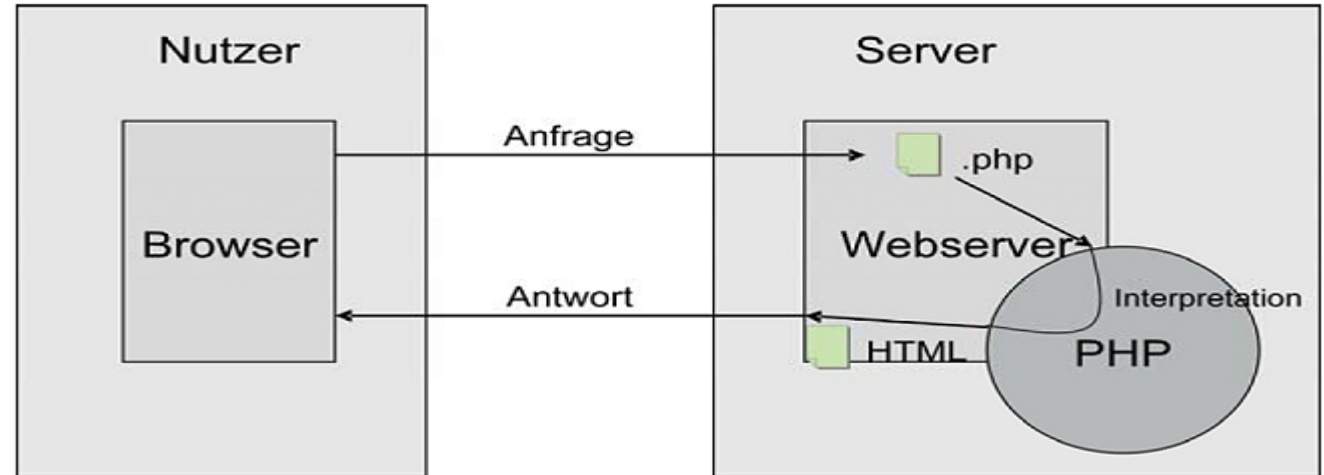
Java

Perl

Python

Ruby

ColdFusion



Clientseitige Technologien

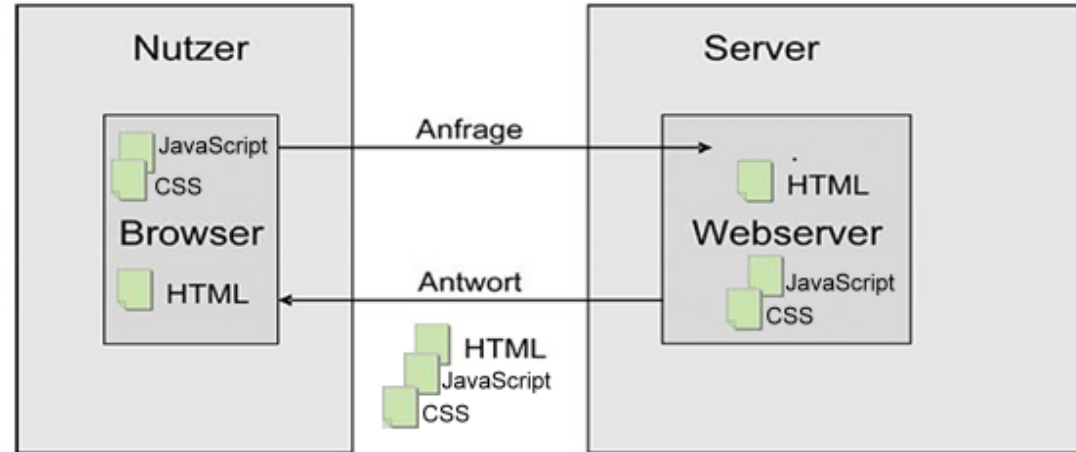
HTML

JavaScript

CSS

SVG

webGL

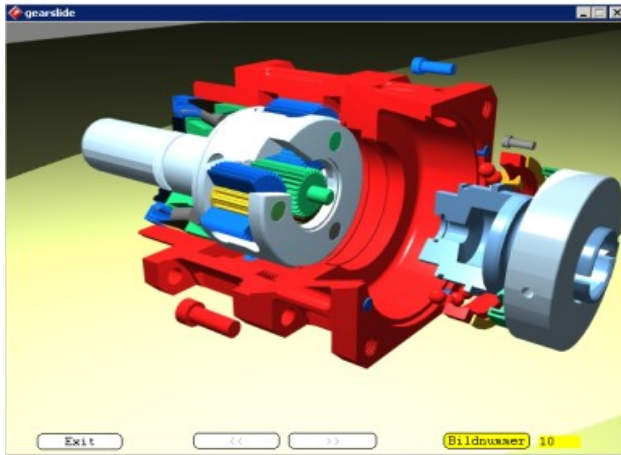


Einsatz von Webtechnologien in E-Learning und TD

3D-CAD-Modelle als Rohstoff für die TD

Bilder: 3D-Modell + Rendering → Photorealistisches Bild

Interaktion: Bildfolge + Scriptsprache → Ablaufdarstellung



Bsp. 1: [HTML + Javascript](#)

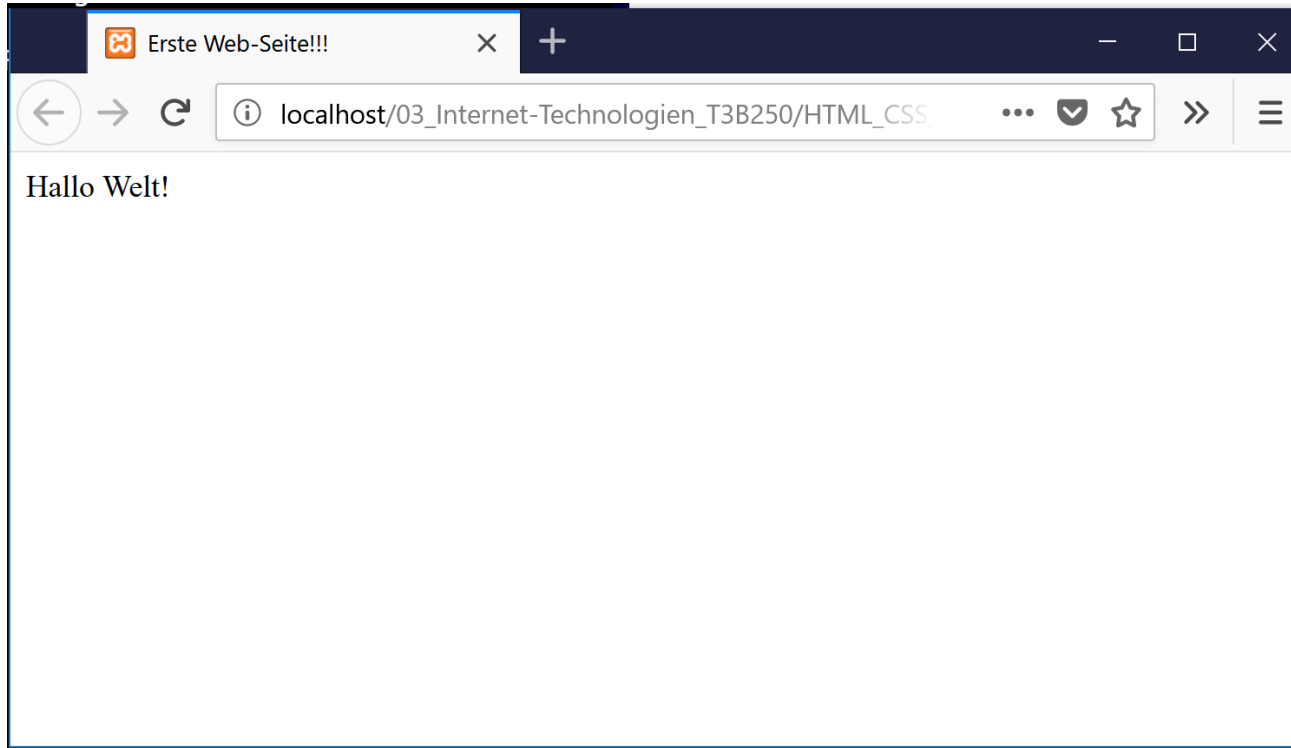
Bsp. 2: [HTML + VRML](#)

Bsp. 3: [HTML + Video](#)

Multilingual, Auditiv, Interaktiv

Format	Multilingual	Auditiv	Interaktiv	Mögliche Interaktionen
<u>WebGL</u>	JS, AJAX, XML	JS, TTS	JS	explore, transform, rotate, control time
X3D	JS, AJAX, XML	JS, TTS	JS	explore, transform, rotate, control time
<u>gITF</u>	JS, AJAX, XML	JS, TTS	JS	explore, transform, rotate, control time
<u>U3D</u>	JS, AJAX, XML	JS, TTS	JS	explore, transform, rotate, control time
FBX (AR, VR)	C#, C++, XML	C#, C++	C#, C++	explore, transform, rotate, control time, immersive
Vektorgrafik	JS, AJAX, XML	JS, TTS	JS	explore, zoom
Pixel-Grafik	JS, AJAX, XML	JS, TTS	JS	explore
<u>Video</u>	JS, AJAX, XML	JS, TTS	JS	explore, control time
WebVR	JS, AJAX, XML	JS, TTS	JS	explore, transform, rotate, control time, immersive

Einfache HTML-Seite



Grundelemente einer HTML-Seite

<html>

Kennzeichnet Anfang und Ende eines HTML-Dokuments

Zwei Teile

<head>

Angaben zum Dokument

Vorlagen

Dokument-Logik: JavaScript-Programmierung

</head>

<body>

Ausgestaltung der Seite

</body>

Einfache HTML-Seite

```
<!doctype html>
<html>
  <head>
    <meta charset="utf-8">
    <title>Erste Web-Seite!!!</title>
  </head>

  <body>
    Hallo Welt!

  </body>
</html>
```

Tags in HTML

HTML-Dateien bestehen aus ASCII-Text

Ein „Tag“ ist ein HTML-Befehl

kann vom Browser interpretiert werden

hat ein Anfang und ein Ende, weitere Informationen stehen zwischen ihnen

`<tag>` Anfang-Tag

`</tag>` End-Tag

Klein- und Großschreibung wird nicht berücksichtigt

Tags in HTML

Verschachtelung von Tags

Tags können ineinander verschachtelt werden

Richtig

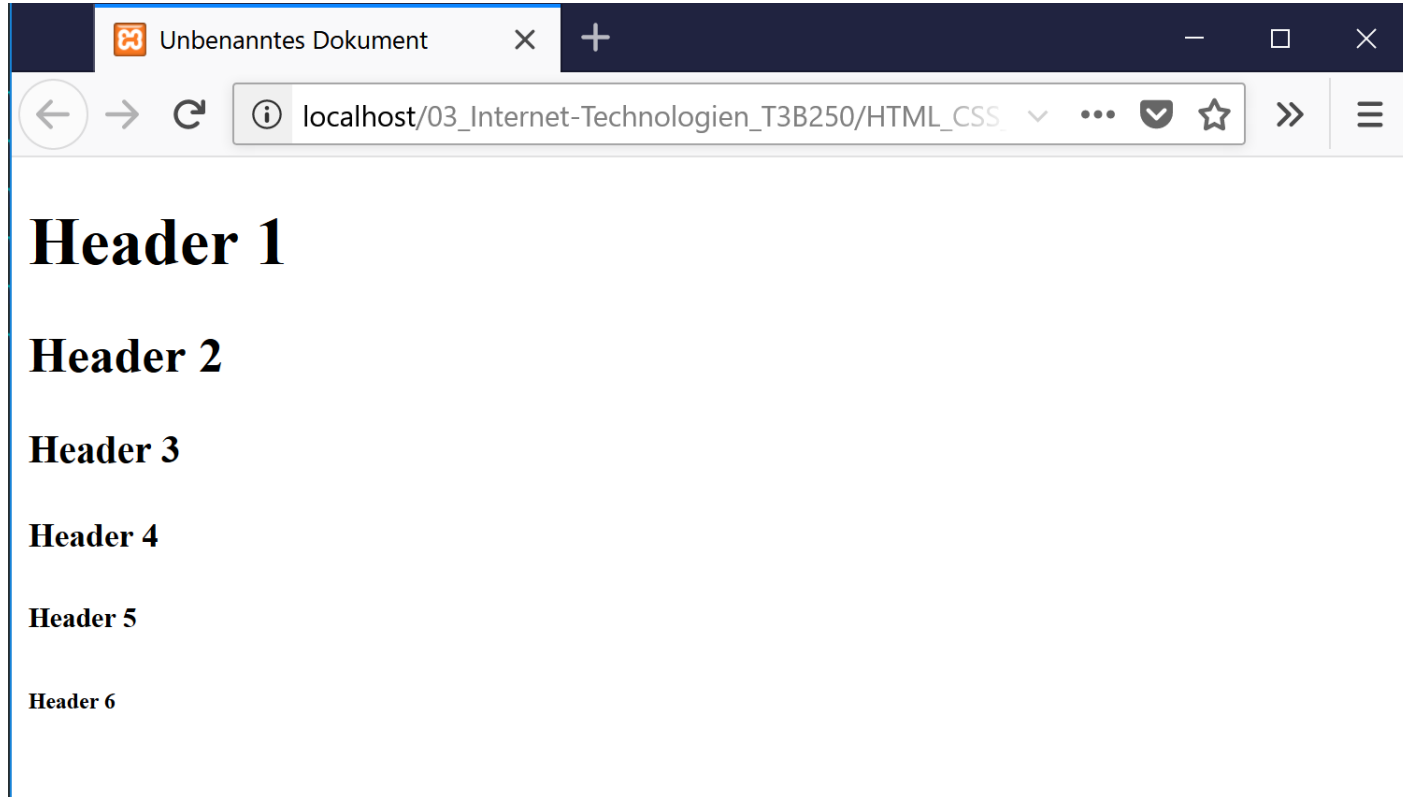
```
<head>  
  <title>  
    </title>  
  </head>
```

Falsch

```
<head>  
  <title>  
  </head>  
  </title>
```

Das Einrücken der Tags ist nicht notwendig, erleichtert jedoch das Lesen der Dokumente

HTML-Elemente



HTML-Elemente

```
<html>
  <head>
    <title>Verschiedene Header</title>
  </head>
  <body>
    <h1>Header 1</h1>
    <h2>Header 2</h2>
    <h3>Header 3</h3>
    <h4>Header 4</h4>
    <h5>Header 5</h5>
    <h6>Header 6</h6>
  </body>
</html>
```

Der Absatz

Absätze dienen der logischen und optischen Gliederung eines Textes

Erleichtern das Bearbeiten eines Dokuments, da sich die vorgenommenen Einstellungen auf dem ganzen Absatz beziehen

Definition

<p>Absatz</p>

Kennzeichnet den Anfang und Ende eines Absatzes

p = paragraph

Formatierung HTML-Seiten



Formatierung HTML-Seiten

```
...
<body>
<h2>Die 3 Schritte zur mobilen Optimierung der Website</h2>
PUBLIZIERT AM 15. APRIL 2015 VON HTML5-ADMIN
<p>Bereits im November 2014 hat Google im offiziellen Webmaster-Blog darauf hingewiesen, dass es ab
    sofort in den Suchergebnissen anzeigt, wenn Websites „mobile friendly“ sind. Am 21. April wird es
    dann richtig ernst: die Mobilitätsfähigkeit der Website wird ab diesem Stichpunkt nicht nur
    angezeigt, sondern hat auch nachhaltige Auswirkungen auf die Platzierung der Seite in den
    Suchergebnissen – zumindest wenn von einem mobilen Endgerät gesucht wird.</p>
<p>Die Vorgaben von Google, was dort unter einer „mobile friendly“ Website verstanden wird sind klar:
</p>
<ul>
<li>Keine Software, die nicht auf allen Endgeräten verfügbar ist, z.B. Flash</li>
<li>Texte müssen ohne Zoomen lesbar dargestellt werden</li>
<li>Die Seitengröße passt sich an die Gerätegröße an, sodass nicht horizontal gescrolled oder gezoomt
    werden muss</li>
<li>Links müssen weit genug auseinander liegen, um gut tap-bar zu sein</li>
<li>Wir besprechen nachfolgend die 3 wichtigsten Schritte für die mobile Optimierung der Website, um
    diesen Anforderungen gerecht zu werden.</li>
</ul>
</body>
```

Die Aufzählungsliste

**** leitet eine Aufzählungsliste mit Punkten ein

unordered list

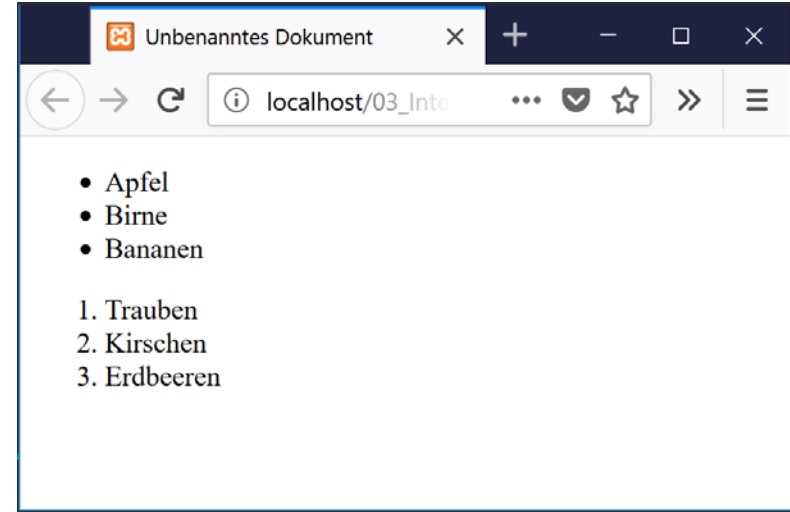
**** leitet eine Aufzählungsliste mit Zahlen ein

ordered list

**** kennzeichnet ein Listeneintrag
list item

```
<ul>
  <li>Apfel</li>
  <li>Birne</li>
  <li>Bananen</li>
</ul>
```

```
<ol>
  <li>Trauben</li>
  <li>Kirschen</li>
  <li>Erdbeeren</li>
</ol>
```



Die Verweise

Dienen der Navigation in WWW

Verbinden HTML-Seiten untereinander

Definition

`<a href="Verweisziel">Mein Link</a>`

Das Verweisziel ist die relative oder absolute Pfadangabe des Dokumentes auf das verwiesen wird.

z.B.: `<a href="http://www.google.de">Suchmaschine</a>`

Die Verweise

Verweise um Bilder

Verbinden HTML-Seiten untereinander über Links auf Bildern

Definition

```
<a href="Verweisziel"></a>
```

Das Verweisziel ist die relative oder absolute Pfadangabe des Dokumentes auf das verwiesen wird.

```
z.B.: <a href="http://www.google.de"></a>
```

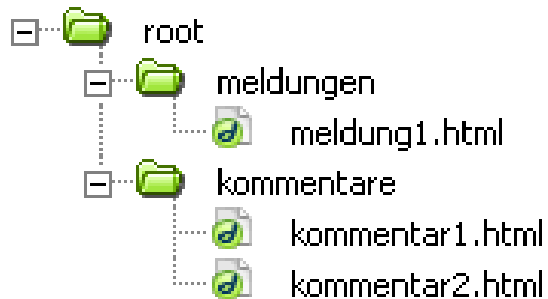
Pfadangaben

Relative Pfadangabe

gibt die Position einer Datei von einer anderen Datei ausgehend an

Verweis von meldung1.html auf kommentar1.html

von der Festplatte und im Internet: **".../kommentare/kommentar1.html"**



Pfadangaben

Absolute Pfadangabe

gibt die Position einer Datei vom Wurzelverzeichnis ausgehend an

Verweis von meldung1.html auf kommentar1.html

von der Festplatte: "**c:/kommentare/kommentar1.html**"

im Internet: "**http://www.adresse.de/kommentare/kommentar1.html**"



Weitere HTML-Elemente

`<p>...</p>`

Paragraph = Absatz

`<br />`

Line Break (Zeilenumbruch)

`<h1>...</h1>`

Überschrift in der Größe 1 (größte Überschrift)

`<h2>...</h2>`

...

`<h6>....</h6>`

weitere Überschriften von 2-6

`<b>...</b>`

Bold (schreibt etwas fett)

`<i>...</i>`

Italics (schreibt etwas kursiv)

`<hr>`

Horizontal Rule (zieht eine horizontale Linie)

Attribute in Tags

Einleitende Tags und Standalone-Tags können zusätzliche Angaben enthalten.

Attribute beschreiben Eigenschaften des Tags.

Attribute haben Auswirkung auf die Informationen zwischen den Tags

z.B. **`<h1 align="center">HTML - die Sprache des Web</h1>`**

Das Attribut beschreibt die Eigenschaft *Ausrichtung* des kommenden Textes

Das Attribut

Der Attributsname

HTML unterscheidet nicht zwischen Klein- und Großbuchstaben bei Attributsnamen

in XHTML müssen Attributsnamen immer klein geschrieben werden

Der Wert eines Attributes

der Wert eines Attributes muss in Anführungszeichen stehen

Arten von Attributen in HTML-Elementen

Attribute mit Wertzuweisung (erlaubte Werte)

z.B. bei **<h1 align="center">...**

Überschrift 1. Ordnung zentriert ausgerichtet

nur die Werte **left**, **center**, **right** und **justify** erlaubt

Arten von Attributen in HTML-Elementen

Attribute mit freier Wertzuweisung

erwartet ein bestimmter Datentyp oder eine bestimmte Konvention

z.B. **<style type="text/css">**

Bereich für Stylesheets definieren

Mime-Type als Wert erwartet

Mime-Typen haben immer den Aufbau *Typ/Untertyp*

z.B. **<table border="1">**

Tabelle mit Rahmen von 1 Pixel Stärke

numerische Angabe

Arten von Attributen in HTML-Elementen

Attribute mit freier Wertzuweisung ohne weitere Konventionen

z.B. **`<p title="Aussage mit Vorbehalt">`**

hier kann ein ganzer Text zugewiesen werden.

Alleinstehende Attribute

z.B. **`<hr noshade>`**

Trennlinie ohne Schatten

alleinstehende Attribute gibt es nur in HTML

in XHTML

`<hr noshade="noshade">`

Kommentare

Schreibweise:

```
<!-- Dies ist ein Kommentar -->
```

zum internen Auskommentieren von Inhalten oder Befehlen
wird im Browser nicht angezeigt

Ein einfaches HTML-Dokument

```
<!DOCTYPE HTML>
<html>
  <head>
    <title>Mein HTML</title>
  </head>
  <body>
    Hallo Welt
  </body>
</html>
```

Dokumenttyp-Angaben

<!DOCTYPE html>

DOCTYPE sagt dem Browser drei Dinge:

- wie das Stammelement des folgenden Dokuments heißt (**html**)
- das Ausrufezeichen vor **DOCTYPE** wird nicht als Kommentar interpretiert

HTML-Kopfdaten - Meta-Angaben

Autor

```
<meta name="author" content= "Jesse James">
```

content="xxx" mit Namen des Urhebers ersetzen

Beschreibung

```
<meta name="description" content="Text">
```

Text mit Kurzbeschreibung des Inhalts ersetzen

Stichwörter

```
<meta name="keywords" content="Wort, Wort, Wort">
```

für ***Wort*** je ein Stichwort zum Inhalt notieren

Anzahl ist flexibel

Stichwörter durch Kommata trennen

Weitere Informationen: <https://www.seo-kueche.de/lexikon/meta-tags/>

HTML-Kopfdaten - Meta-Angaben

Zeitstempel

<meta name="date" content="yyyy-MM-ddThh:mm:ss+hh:mm">

yyyy = Jahr, **MM** = Monat, **dd** = Tag, **T** = fixer Buchstabe, **hh** = Stunden,
mm = Minuten, **ss** = Sekunden, **+/-** = Abweichung gegenüber Greenwich in Stunden und Minuten

Datum der Publizierung

z.B: **content="2003-10-14T08:49:37+01:00,,**

Robots

<meta name="robots" content="index|noindex|follow|nofollow">

Anweisung an Suchmaschinen:

index = Suchmaschine darf die Seite durchsuchen

noindex = Suchmaschine darf die nicht Seite durchsuchen

follow = Suchmaschine darf die Links von dieser Seite verfolgen und die neuen Seiten durchsuchen

nofollow = Suchmaschine darf nicht die Links von dieser Seite verfolgen und die neuen Seiten durchsuchen

HTML-Kopfdaten - Meta-Angaben

Zeichensatz

<!-- HTML5 -->

<meta charset="utf-8">

<!-- HTML 4.x -->

<meta http-equiv="content-type" content="text/html; charset=utf-8">

<http://de.selfhtml.org/inter/zeichenkodierungen.htm>

ASCII-Zeichensatz										
+	0	1	2	3	4	5	6	7	8	9
30				!	"	#	\$	%	&	'
40	(	)	*	+	,	-	.	/	0	1
50	2	3	4	5	6	7	8	9	:	;
60	<	=	>	?	@	A	B	C	D	E
70	F	G	H	I	J	K	L	M	N	O
80	P	Q	R	S	T	U	V	W	X	Y
90	Z	[	\	]	^	_	`	a	b	c
100	d	e	f	g	h	i	j	k	l	m
110	n	o	p	q	r	s	t	u	v	w
120	x	y	z	{		}	~			

iso-8859-1										
+	0	1	2	3	4	5	6	7	8	9
160		í	φ	£	¥	ı	š	¨		
170	ª	«	¬	-	®	-	°	±	²	³
180	´	µ	¶	·	¸	¹	º	»	¼	½
190	¾	¿	À	Á	Â	Ã	Ä	Å	Æ	Ç
200	È	É	Ê	Ë	Ì	Í	Î	Ï	Ð	Ñ
210	Ò	Ó	Ô	Õ	Ö	×	Ø	Ù	Ú	Û
220	Ü	Ý	Þ	ß	à	á	â	ã	ä	å
230	æ	ç	è	é	ê	ë	ì	í	î	ï
240	ð	ñ	ò	ó	ô	õ	ö	÷	ø	ù
250	ú	û	ü	ý	ÿ					

<http://www.utf8-zeichentabelle.de/unicode-utf8-table.pl?number=1024>

HTML-Kopfdaten - Meta-Angaben

Cache-Verhalten

```
<meta http-equiv="expires" content="0">
```

definiert, ob ein bestimmtes Dokument zwischengespeichert werden darf und für welche Zeitspanne

Eintrag in Sekunden oder ein Datum im internationalen Format

```
content="Tue, 14 Oct 2003 12:00:00 GMT"
```

Eintrag "0" schaltet die Verwendung des Caches ab

Weiterleitung

```
<meta http-equiv="refresh" content="0; URL=URI">
```

veranlasst die Weiterleitung zu einer anderen Adresse

gibt an nach wie viel Sekunden die Weiterleitung gestartet werden soll z.B: **content="5** nach 5 Sekunden
0=sofort

URI bestimmt die Ziel-Adresse, die nach der angegebenen Zeit automatisch geladen werden soll

HTML-Kopfdaten - Meta-Angaben

Adressbasis

<base href="*URI*">

URI gibt die Ziel-Adresse an, die als Basis für relative Adressierung dienen soll

Zielfensterbasis

<base target="*Name*">

Name ist der Name des gewünschten Frame

gibt das Frame an, in dem Verweise standardmäßig geladen werden sollen

beim Verweisen ist dann keine Angabe zu **target** mehr erforderlich

HTML-Body

Farben für Text, Hintergrund und Verweise

```
<body bgcolor="#XXXXXX" text="#XXXXXX" link="#XXXXXX" vlink="#XXXXXX"
alink="#XXXXXX">
```

bgcolor = Hintergrundfarbe des Dokumentes

text = Schriftfarbe des Dokumentes

link = Farbe noch nicht besuchter Verweise

vlink = Farbe besuchter Verweise

alink = Farbe aktivierter Verweise (mouse over)

Für **#XXXXXX** eine hexadezimal notierte RGB-Farbe angeben oder einen erlaubten Farbnamen

Format: **#RRGGBB** (Rot, Grün, Blau)

HTML-Body

Hintergrundbild (Wallpaper)

<body background="*URI*">

URI gibt die Ziel-Adresse der gewünschten Grafikdatei an

Elemente zur Textstrukturierung

Textabsätze ausrichten

<p align="Ausrichtung">Text</p>

Ausrichtung kann durch einen der folgenden Werte ersetzt werden:

left = linksbündig,
center = zentriert,
right = rechtsbündig,
justify = Blocksatz

kann ebenfalls für **<h1></h1>... <h6></h6>** benutzt werden

Das allgemeine Block-Element

`<div>Inhalte</div>`

schließt mehrere Elemente in einen gemeinsamen Bereich ein

Text, Grafiken, Tabellen usw.

formatiert Elemente gemeinsam mit Hilfe von Attributen

- Blockelemente erzeugen einen »Absatz« und **nehmen die gesamte verfügbare Breite ein.**
- Blockelemente können weitere Block- und Inline-Elemente enthalten.

z.B. **`<div align="center">`**

→ besser mit CSS **`<div style="text-align: center;">`**

Ideal für die Formatierung mittels CSS!!!!

Das allgemeine Block-Element



Das allgemeine Block-Element <div>

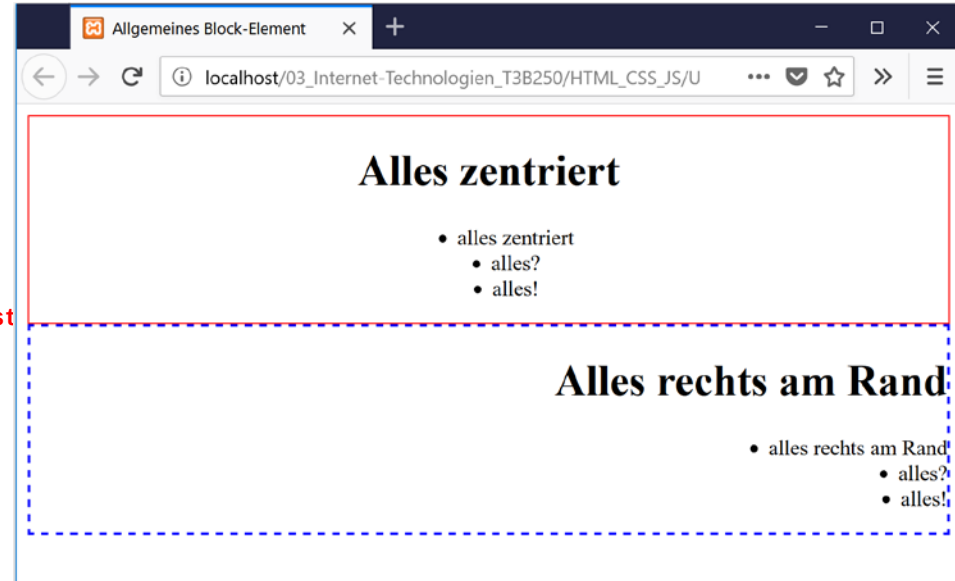
```
<!DOCTYPE HTML>
<html>
  <head>
    <title>Allgemeines Block-Element</title>
  </head>
  <body>
    <div align="center">
      <h1>Alles zentriert</h1>
      <ul>
        <li>alles zentriert</li>
        <li>alles?</li>
        <li>alles!</li>
      </ul>
    </div>
    <div align="right">
      <h1>Alles rechts am Rand</h1>
      <ul>
        <li>alles rechts am Rand</li>
        <li>alles?</li>
        <li>alles!</li>
      </ul>
    </div>
  </body>
</html>
```

Das allgemeine Block-Element <div> mit CSS-Formatierung

```
<!DOCTYPE HTML>
<html>
  <head>
    <title>Allgemeines Block-Element</title>
  </head>
  <body>
    <div style="text-align: center;">  <!-- CSS-Formatierung -->
      <h1>Alles zentriert</h1>
      <ul>
        <li>alles zentriert</li>
        <li>alles?</li>
        <li>alles!</li>
      </ul>
    </div>
    <div style="text-align: right;">  <!-- CSS-Formatierung -->
      <h1>Alles rechts am Rand</h1>
      <ul>
        <li>alles rechts am Rand</li>
        <li>alles?</li>
        <li>alles!</li>
      </ul>
    </div>
  </body>
</html>
```

Das allgemeine Block-Element <div> mit CSS-Formatierung

```
<!-- CSS-Formatierung -->
<style>
#div1 {
    text-align: center;
    border: solid thin red;
}
#div2 {
    text-align: right;
    border: dashed 2px blue;
}
ul {
    list-style-position: inside;
    /* Spezieller CSS-Befehl für den Abstand des List
}
</style>
<!-- CSS-Formatierung -->
<body>
    <div id="div1">
        <h1>Alles zentriert</h1>
        <ul>
            <li>alles zentriert</li>
            <li>alles?</li>
            <li>alles!</li>
        </ul>
    </div>
    <div id="div2">
```



Das allgemeine Inline-Element

`<span>Text</span>`

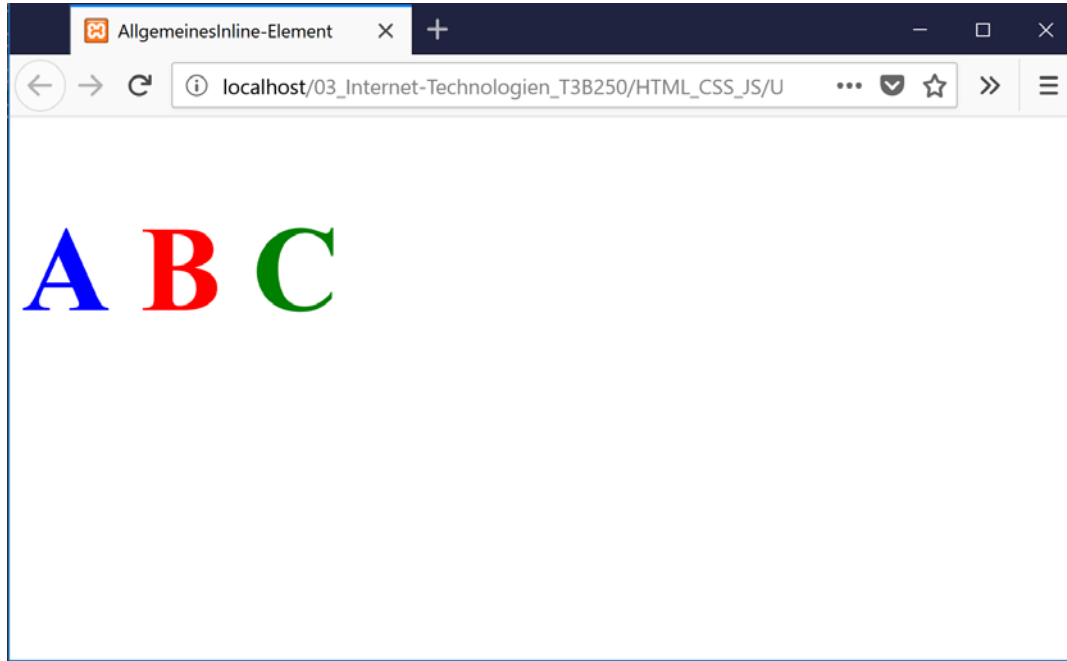
schließt mehrere Elemente in einen gemeinsamen Bereich ein

für Text und andere Inline-Elemente gedacht, im Gegensatz zu `<div>`

besitzt selbst keinerlei Eigenschaften

bewirkt nichts

Das allgemeine Inline-Element



Das allgemeine Inline-Element

```
<!DOCTYPE HTML>
<html>
  <head>
    <title>Allgemeines Inline-Element</title>
  </head>
  <body>
    <h1 style="font-size:500%">
      <span style="color:blue">A</span>
      <span style="color:red">B</span>
      <span style="color:green">C</span>
    </h1>
  </body>
</html>
```

Trennlinien

`<hr>`

fügt eine Trennlinie ein (*hr = horizontal rule*)

optische Abgrenzung nicht unmittelbar zusammengehöriger Textabschnitte

eine Trennlinie erzeugt einen eigenen Absatz

Tabelle

Darstellung tabellarischer Daten. Positionierung von Text und Grafiken auf der HTML-Seite.

```
<table width="500" border="1" cellspacing="5" cellpadding="2">
  <tr>
    <td>&nbsp;</td>
    <td>&nbsp;</td>
    <td>&nbsp;</td>
  </tr>
  <tr>
    <td>&nbsp;</td>
    <td>&nbsp;</td>
    <td>&nbsp;</td>
  </tr>
  <tr>
    <td>&nbsp;</td>
    <td>&nbsp;</td>
    <td>&nbsp;</td>
  </tr>
</table>
```


Tabellenaufbau

```
<table border="1">  
  <tr>  
    <th>Kopfzelle</th>  
    <th>Kopfzelle</th>  
    <th>Kopfzelle</th>  
  </tr>  
  <tr>  
    <td>Datenzelle</td>  
    <td>Datenzelle</td>  
    <td>Datenzelle</td>  
  </tr>  
</table>
```

<table>				
<tr>	<th> </th>	<th> </th>	<th> </th>	</tr>
<tr>	<td> </td>	<td> </td>	<td> </td>	</tr>
<tr>	<td> </td>	<td> </td>	<td> </td>	</tr>
				</table>

Kopfzelle	Kopfzelle	Kopfzelle
Datenzelle	Datenzelle	Datenzelle

Spalten vordefinieren

vor dem Beginn der eigentlichen Tabelle kann

die Breite der Spalten einer Tabelle in Pixel bestimmt werden

```
<table>  
  <colgroup>  
    <col width="80">  
    <col width="100">  
    <col width="320">  
  </colgroup>  
  ...  
</table>
```

Kopfzelle	Kopfzelle	Kopfzelle
Datenzelle	Datenzelle	Datenzelle

Spalten vordefinieren

vor dem Beginn der eigentlichen Tabelle kann

die Breite einer Spalte in Prozent von 100% der gesamten Tabelle angegeben werden

```
<table>
  <colgroup>
    <col width="20%">
    <col width="50%">
    <col width="30%">
  </colgroup>
  ...
</table>
```

Kopf, Körper und Fuß einer Tabelle

eine Tabelle ist in 3 logische Bereiche aufgeteilt

einen Kopfbereich

einen oder mehrere Datenbereiche

einen Fußbereich

```
<table border="1">  
  <thead>  
    ...  
  </thead>  
  <tfoot>  
    ...  
  </tfoot>  
  <tbody>  
    ...  
  </tbody>  
</table>
```

Zellen verbinden

Zellen in einer Zeile spaltenweise verbinden z.B. **<td colspan=„3">**

Mehrere Zellen können innerhalb einer Zeile miteinander verbunden werden, so dass sich eine Spalte in dieser Zeile über mehrere Spalten hinweg erstreckt.

```
<table width="500" border="1" cellspacing="5"
cellpadding="2">
  <tr>
    <td colspan="3">&nbsp;</td>
  </tr>
  <tr>
    <td>&nbsp;</td>
    <td>&nbsp;</td>
    <td>&nbsp;</td>
  </tr>
  <tr>
    <td>&nbsp;</td>
    <td>&nbsp;</td>
    <td>&nbsp;</td>
  </tr>
</table>
```


Zellen verbinden

Zellen in einer Spalte zeilenweise verbinden z.B. **<th rowspan="2">**. Mehrere Zellen können innerhalb einer Spalte miteinander verbunden werden, so dass sich eine Zeile in dieser Spalte über mehrere Zeilen hinweg erstreckt.

```
<table width="500" border="1" cellspacing="5" cellpadding="2">
  <tr>
    <td colspan="3">&nbsp;</td>
  </tr>
  <tr>
    <td rowspan="2">&nbsp;</td>
    <td>&nbsp;</td>
    <td>&nbsp;</td>
  </tr>
  <tr>
    <td>&nbsp;</td>
    <td>&nbsp;</td>
  </tr>
</table>
```


Zellen verbinden

Zellen spalten- und zeilenweise zugleich verbinden

z.B: `<th colspan="2" rowspan="2">`



Listen

Aufzählungsliste

z.B:

```
<ul>
  <li>Autos</li>
  <li>Fahrräder</li>
    <ul>
      <li>Rennräder</li>
      <li>Pixies</li>
    </ul>
  <li>Motorisierte Zweiräder</li>
    <ul>
      <li>Motorrad</li>
      <li>Elektroscooter</li>
    </ul>
  <li>LKW</li>
</ul>
```

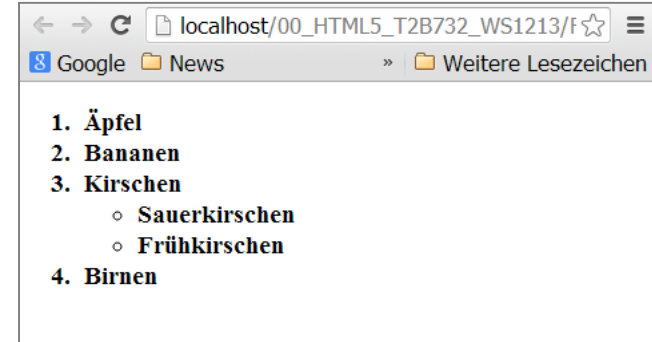


Listen

Nummerierte Listen

z.B:

```
<ol>
  <li>Äpfel</li>
  <li>Bananen</li>
  <li>Kirschen</li>
    <ul>
      <li>Sauerkirschen</li>
      <li>Frühkirschen</li>
    </ul>
  <li>Birnen</li>
</ol>
```



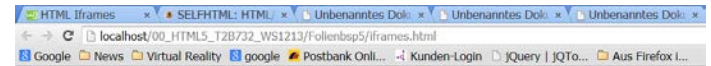
iframes

Iframes sind eingebettete Flächen. Diese Flächen sind wie Grafiken Bereiche innerhalb einer HTML-Datei, in denen fremde Quellen, vor allem andere HTML-Dateien angezeigt werden können.

```
<style>
iframe {
    position: absolute;
    left: 50px;
    top: 200px;
}
</style>
</head>

<body>

<iframe src="http://www.hs-karlsruhe.de" name="if1"
width="500" marginwidth="1" height="500"
marginheight="1" align="middle" scrolling="auto">
</iframe>
</body>
```



Iframe - Attribute

Attribut	Wert	Beschreibung
align	left right top middle bottom	Nicht unterstützt in HTML5. Spezifiziert die Ausrichtung eines <iframe>
frameborder	1 0	Nicht unterstützt in HTML5. Spezifiziert ob ein Rahmen um den <iframe> gezeigt wird.
height	<i>pixels</i>	Spezifiziert die Höhe des <iframe>
longdesc	URL	Nicht unterstützt in HTML5. Spezifiziert eine Seite mit einer langen Beschreibung des <iframe>-Inhalts
marginheight	<i>pixels</i>	Nicht unterstützt in HTML5. Spezifiziert den oberen und unteren Abstand des <iframe>-Inhalts
marginwidth	<i>pixels</i>	Nicht unterstützt in HTML5. Spezifiziert den linken und rechten Abstand des <iframe>-Inhalts
name	<i>text</i>	Spezifiziert den Namen eines <iframe>
sandbox	"" allow-forms allow-same-origin allow-scripts allow-top-navigation	Ermöglicht spezielle Restriktionen für den Inhalt des <iframe>
scrolling	yes no auto	Nicht unterstützt in HTML5. Spezifiziert ob im <iframe> Scrollbars angezeigt werden oder nicht
seamless	seamless	Spezifiziert das der <iframe> aussieht wie ein Teil des Dokuments
src	URL	Spezifiziert die Adresse (URI) des Inhalts <iframe>
srcdoc	HTML_code	Spezifiziert den HTML-Content der Seite, um ihn im <iframe> zu zeigen.
width	<i>pixels</i>	Spezifiziert die Breite eines <iframe>

Verweise

Verweis definieren

z.B: **Verweistext**

Für **URI** die Ziel-Adresse oder ein Ziel mit oder ohne Pfad der Datei oder Quelle angeben, das verlinkt werden soll

URI kann sein:

Anker in der gleichen Datei (z.B. #Ankername)

Andere Datei (z.B. datei.htm)

Andere Datei in anderem Verzeichnis (z.B. ../verzeichnis/datei.htm)

Anker in der anderer Datei (z.B. datei.htm#Ankername)

WWW-Adresse (http://...)

FTP-Adresse (ftp://...)

Telnet-Adresse (telnet://...)

Newsgroup-Adresse (news:...)

Absolute lokale Adresse (file://...)

Verweise

Zielfenster für Verweise

z.B: `<a href="URI" target="Zielfenster">Verweistext</a>`

Für ***Zielfenster*** den Namen eines definierten Framefensters angeben oder einen der folgenden reservierten Namen:

_blank = Verweis in neuem Fenster öffnen

_self = Verweis im gleichen Fenster öffnen

_parent = Besagt, dass das Dokument in dem Zustand des Fensters geöffnet werden soll, wie es vor dem Erscheinen des Framesets das Fall war. Dies ist dann nützlich, wenn z.B. zu einer bestimmten Ausgangsposition wieder zurück gelangt werden soll.

_top = Besagt, dass sich die Datei/Seite im gesamten Fenster öffnen soll. Dies ist dann nützlich, wenn Frames definiert wurden und die Datei auf die verwiesen wird nicht in einem speziellen Frame sondern im gesamten Fenster erscheinen soll.

E-Mail-Verweise

`<a href="mailto:name@domain.xy">Verweistext</a>`

`<a href="mailto:name@domain.xy?subject=Thema">Verweistext</a>`

`<a href="mailto:name@domain.xy?cc=name2@domain.xy">Verweistext</a>`

`<a href="mailto:name@domain.xy?bcc=name2@domain.xy">Verweistext</a>`

`<a href="mailto:name@domain.xy?body=Text">Verweistext</a>`

`<a href="mailto:name@domain.xy?Zusatz1&Zusatz2">Verweistext</a>`

HTML5 – Erweiterungen → article, address

...

`<article>`

`Lehrbeauftragter für UX-Design<br />`

`an der Hochschule Karlsruhe<br />`

`Wirtschaft & Technik`

`<address>`

`<h4>Malte Müller</h4>`

`Kaiserstraße 123<br />`

`76133 Karlsruhe<br />`

`</address>`

`</article>`

...

Beispiel

HTML5 – Erweiterungen → pre, code

```
<pre>
  <code>
    if( x == 3) {
      document.getElementById("Name").style.color = "white";
    }
  </code>
</pre>
```

Das <pre> -Tag definiert vorformatierten Text.

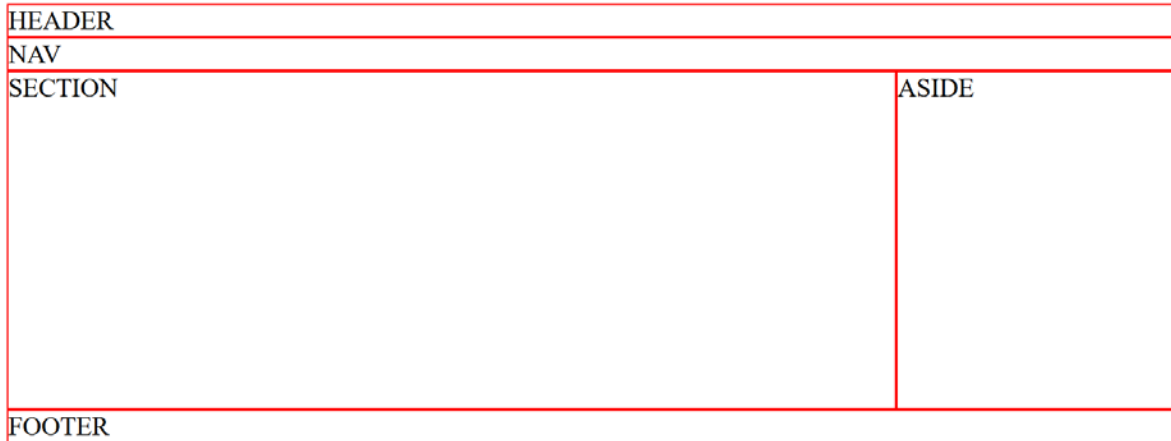
Text in einem <pre> -Element wird in einer Schriftart mit fester Breite (normalerweise Courier) angezeigt und behält Leerzeichen und Zeilenumbrüche bei.

Das <code> -Tag ist ein Phrasen-Tag. Es definiert ein Stück Computercode.

Beispiel

HTML5 – Semantische Tags

header, nav, section, aside, footer...



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

Beispiel

HTML5 – Textfeldkontrolle/Validierung von Formularen

z. B. E-Mail-Adresse kontrollieren

```
<form>  
  <input id="email" name="email" type="email" required="required" />  
  <input type="submit" />  
</form>
```

Beispiel

HTML5 – SVG

```
<!DOCTYPE html>
```

```
<html>
```

```
<body>
```

```
<svg width="100" height="100">
```

```
  <circle cx="50" cy="50" r="40" stroke="green" stroke-width="4"  
    fill="yellow" />
```

```
</svg>
```

```
</body>
```

```
</html>
```

Beispiel

HTML5 - Schieberegler

```
<style>
body, html { width: 100%; height: 100%; }
#Rechteck { width: 200px; height: 200px; border: solid thin red; }
</style>
<body>
  <input type="range" min="0" max="10" step="2" value="6" />
  <div id="Rechteck">
</div>

</body>
<script>
document.getElementsByTagName("input")[0].onmousemove = function() {
  document.getElementById("Rechteck").style.width =
    document.getElementsByTagName("input")[0].value + "%";
}
</script>
```

Beispiel

Quellen und Literatur

Quellen/Tutorials:

[1] W3Schools → <https://www.w3schools.com>

[2] SelfHTML 8.1.2 → <https://de.selfhtml.org/>

Literatur:

Ertel, A./ Laborenz, K. (2015): Responsive Webdesign. Galileo Press, Bonn

Hauser, T./ Wenz, C. / Maurice, F. (2018): Das Website Kompendium. Markt + Technik Verlag, München

Müller P. (2007): little boxes. Webseiten gestalten mit CSS. Grundlagen; Markt + Technik Verlag, München