

 Designing Education
Connecting People



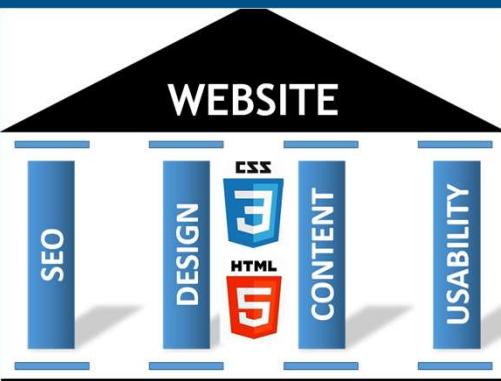
Das erwartet Sie:

- Web-Design Grundlagen
- Benutzeroberflächen gestalten

 Lernfeld 10

Benutzerschnittstellen gestalten und entwickeln

1



Benutzerschnittstellen gestalten und entwickeln

HTML/CSS/PHP/SQL

Lernziel

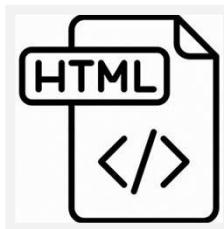
Gestalten und Entwickeln von Benutzeroberflächen für softwarebasierte

- Arbeitsabläufe
- Geschäftsprozesse

 2

2

Themen



HTML

Grundlagen
Tabellen, Listen,
Formular etc.



CSS

Cascading
Style Sheets

Responsive
Webdesign

Anpassen der Webseite
für verschiedene
Ausgabemedien



SEO

Suchmaschinen-
optimierung



PHP-SQL

Einbinden von
Datenbanken mittels
PHP



3

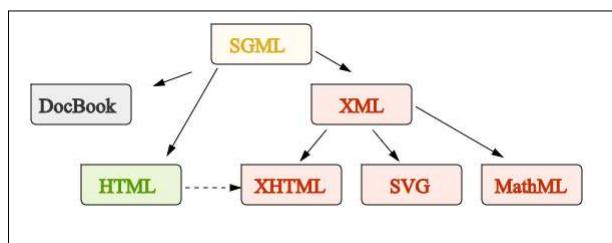
3

HTML - Historie

Hypertext Markup Language wurde etwa 1990 vom Web-Gründer Tim Berners-Lee als **Auszeichnungssprache (Markup Language)** entwickelt, die auf SGML basierte



Ein System für CERN-Forscher zur Verwendung und Weitergabe von Dokumenten



4

4

2

Standard Generalized Markup Language - SGML

SGML

- Idee
 - Computerverarbeitung von Dokumenten
 - Trennung von Struktur, Inhalt und Darstellung
 - System- und herstellerunabhängig
- Nachteile
 - Komplex, viele Optionen für spezielle Anwendungen
 - Syntax schwer für Menschen lesbar
 - Komplizierte und teure Tools

Geschichte

- Metasprache, die ab 1960 Texte standardisieren und für Computer lesbar machen sollte
- In der ISO 8879:1986 Information Processing – Text and Office Systems – wurden Standards für die Auszeichnung von Texten festgelegt
- Diese Normen wurden aber 2012 ersatzlos zurückgezogen



5

5

... MLs (Markup Language)

SGML

- SGML (Standard Generalized Markup Language) als Urvater von XML schon 1986 als Standard verabschiedet

HTML

- HTML (Hypertext Markup Language), das ebenfalls auf SGML basiert, erschien 1992 (HTML DTD 1.1)

XML

- Die erste XML-Spezifikation wurde 1998 vom World Wide Web Consortium (W3C) veröffentlicht

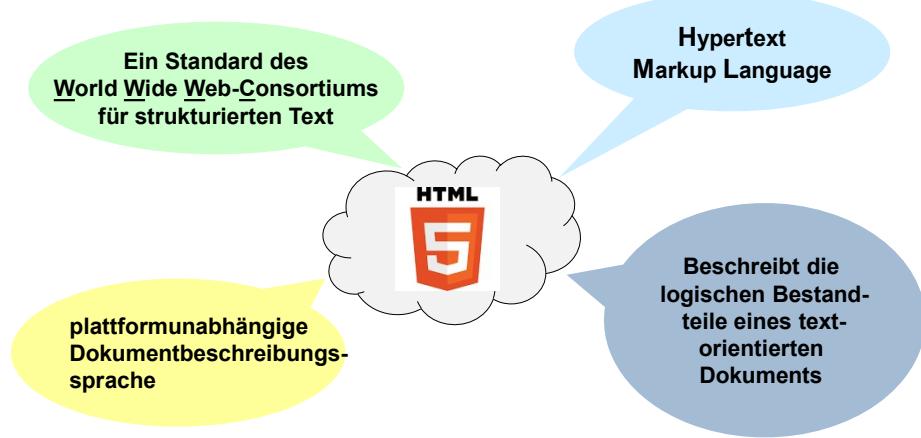


6

6

3

Hyper Text Markup Language - HTML



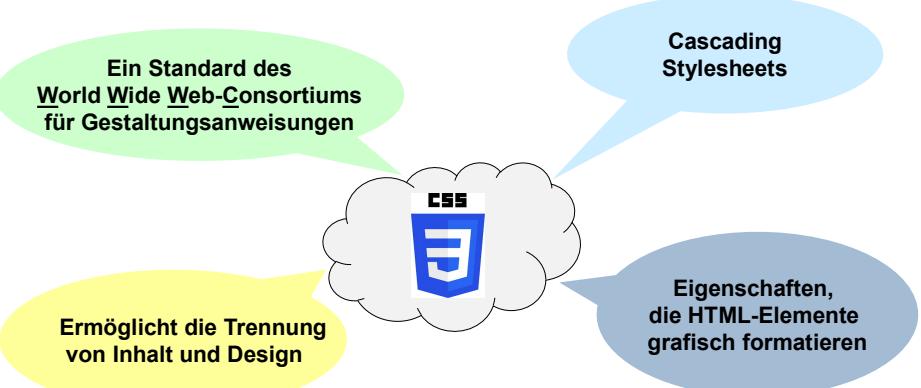
Spezielle Programme (Browser) wie Mozilla Firefox, Google Chrome oder Internet Explorer (Microsoft) führen die Formatierungsbefehle des HTML-Dokuments aus und zeigen es auf dem Bildschirm an



7

7

Hyper Text Markup Language - HTML



Sie ist ein sogenannter „living standard“ (lebendiger Standard)

https://de.wikipedia.org/wiki/Cascading_Style_Sheets



8

8

HTML

- K E I N E Programmiersprache
- Formatierungssprache für Web-Browser
- Einfach zu verstehen und zu erstellen
- Weit verbreitet und angewendet
- Keine inhaltliche Semantik
- Inhalt kann nicht von Programmen erkannt und bearbeitet werden
- Seiten sind aufwendig zu warten

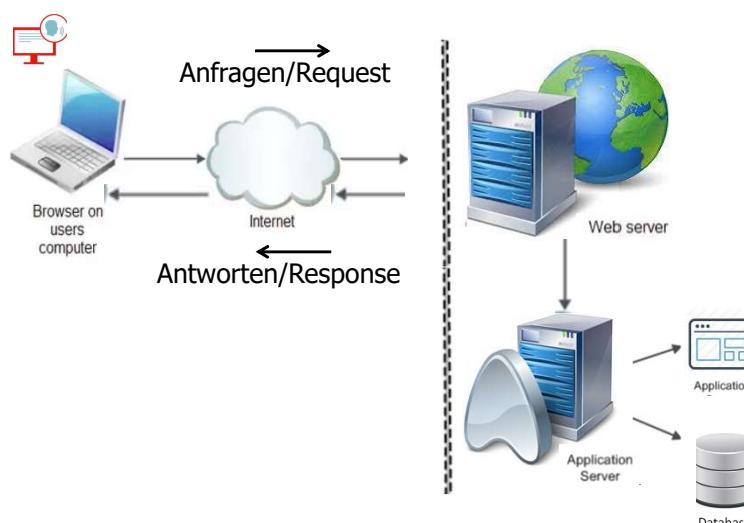
<https://wiki.selfhtml.org>



9

9

Szenario

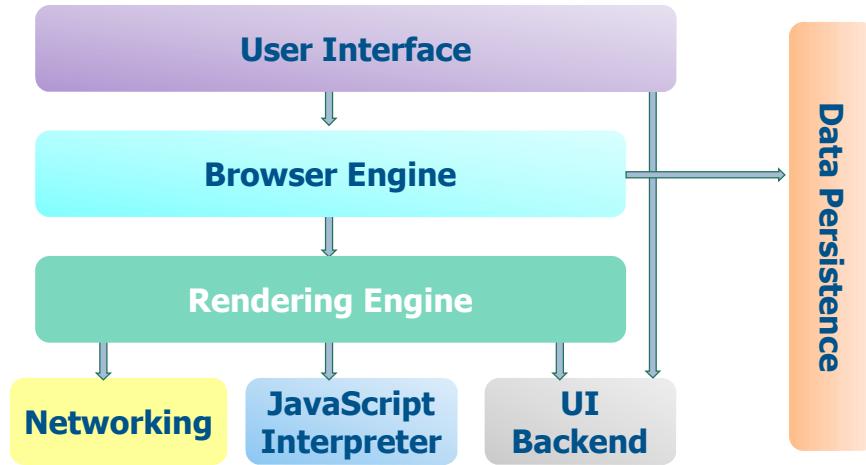


10

10

5

Hauptkomponenten eines Browsers



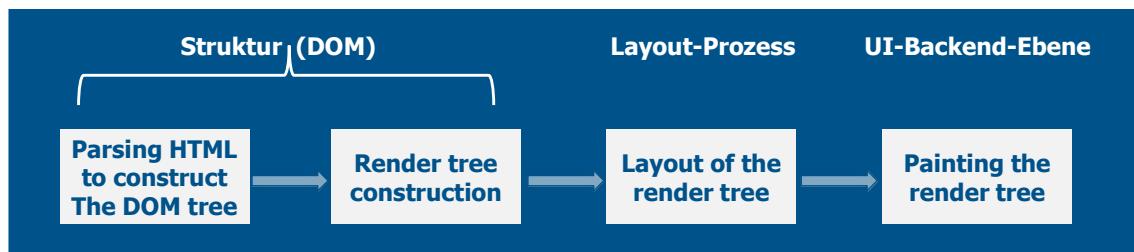
11



11

Rendering Modul

Grundlegender Ablauf des Rendering-Moduls



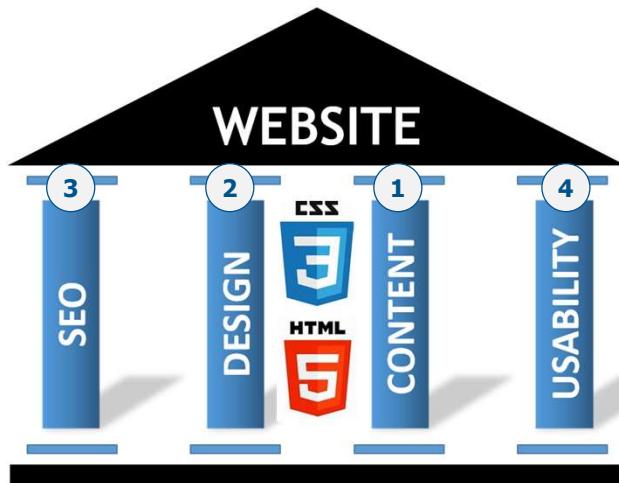
Das Rendering-Modul beginnt mit dem Abrufen der Inhalte des angeforderten Dokuments aus der Netzwerkebene. Dies erfolgt üblicherweise in 8 K-Blöcken



12

12

Grundlegendes zur Erstellung einer Website



SEO = Suchmaschinenoptimierung



13

13

HTML

- ist eine reduzierte XML-Notation, enthält eine festgelegte, begrenzte Anzahl von Befehlen, um Web-Seiten zu erstellen
- enthält strukturierten Text
- besteht aus einzelnen Elementen, sogen. "Tags" `<p> Text </p >`
- HTML kann mit einem beliebigen Texteditor erstellt werden (z. B. *Microsoft Notepad*)



14

14

Grundlagen - HTML



Typische Elemente eines textorientierten Dokuments:

Kopfdaten

+

- Überschriften (<h1>, <h2>, <h3>,...)
- Textabsätze (<p>)
- Listen (,...)
- Tabellen (<table>,...)
- Grafikreferenzen (<img...> ,<href>,...)



15

15

HTML - head

Kopfdaten - für nicht menschliche Besucher einer Internetseite

- base (Basis für relative Verweise)
- link (Beziehungen zu anderen Dokumenten im Web-Space)
- meta (enthalten zusätzliche Angaben, die beispielsweise die Verwaltung des Dateiinhalts erleichtern sollen)
- style (für Stylesheet-Angaben möglich, die einheitlich für das aktuelle Dokument gelten)
- title (Titel der Seite, die in der Registerkarte (Reiter) angezeigt wird)



16

16

Ordnerstruktur

- Auf der Festplatte einen eigenen Ordner, z. B. mit dem Namen "home" anlegen.
- Die Startseite heißt häufig index.html.
- Im Unterordner "html" sind alle weiteren HTML-Dateien bzw. weitere Unterordner.
- Im Unterordner namens "img" befinden sich alle Bilder, die verwendet werden, aber auch nur die, auf die von den Seiten verwiesen wird. Andere Grafiken sind in einem anderen Ordner.



Wenn viele Dateien vorhanden sind, sollte jedes HTML-Dokument einen "sprechenden Namen" haben: also z. B. "bilder.html", wenn dort Fotos angezeigt werden, oder "hobbies.html", wenn darin z. B. über Hobbies geschrieben wird



17

17

Live Demonstration



HTML-Datei schreiben und ausführen

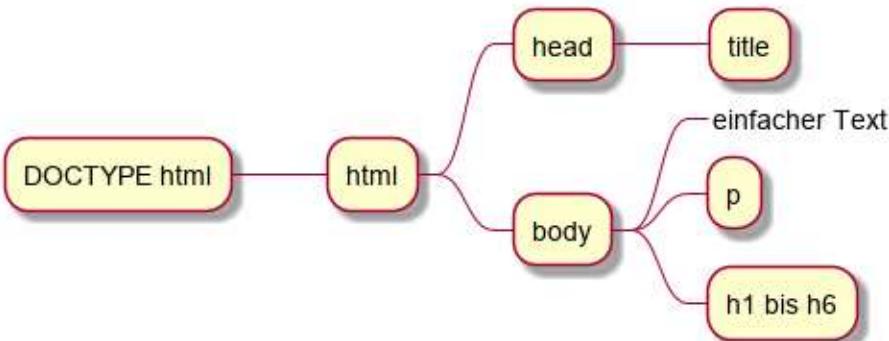


18

18

HTML - Struktur

So sieht das Grundgerüst aus:



19

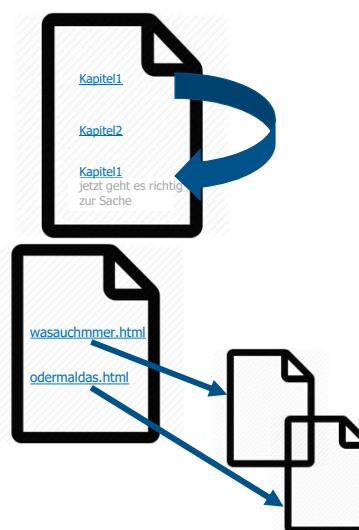
19

HTML - Verweise

Wozu Verweise?

- Navigation innerhalb einer Seite/
eines Dokuments
 - → Anker

- Navigation zur einer anderen Seite/
einem anderen Dokument
 - → Links



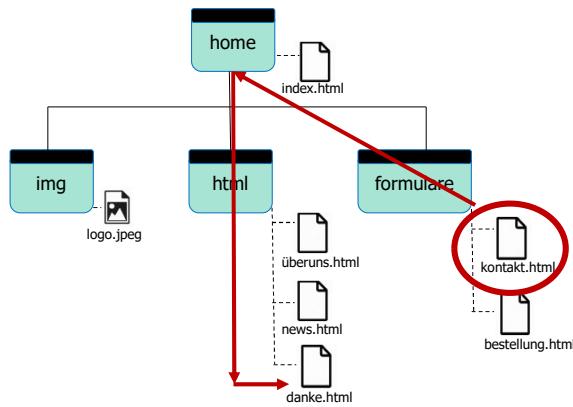
20

10

HTML – Pfadangaben

Beispiel:

Die Verwendung eines Kontaktformulars, das auf einer Unterseite eingebaut wurde. Anschließend soll zu der Erfolgsseite "danke.html" verwiesen werden



Die relative Pfadangabe würde in diesem Skript entsprechend so aussehen:

-/ gehe aus dem Verzeichnis "formulare" ein Verzeichnis höher in das Stammverzeichnis (dort, wo sich auch die Startseite (index.html) befindet)
- von dem Stammverzeichnis weiter in den Ordner "html/ "
- rufe dort die Seite "danke.html" auf



21

21

Verlinkung - Verweise

Verlinkung (Verweise)

```
<a href="zielseite">Ausgabe auf Bildschirm für Besucher</a>
```

Interne Links:

Link nach "zielseite" – intern dann nur der Dateiname der HTML-Seite, z. B. "hobbies.htm", "index.htm" etc.

```
<a href="hobbies.htm">Meine Hobbies</a>
```

Meine Hobbies

```
<a href="hobbies.htm">Meine Hobbies</a>
|
<a href="musik.htm">Meine Musik</a>
```

Meine Hobbies | Meine Musik



22

22

Verlinkung - Verweise

Externer Link

mit voller URL, z. B. <http://www.html-seminar.de/> (der letzte / gehört dazu!)

```
<a href="http://www.html-seminar.de/">HTML-Tutorial</a>
```

Externer Link in neuem Fenster

mit voller URL, z. B. <http://www.html-seminar.de/> und zusätzlich mit target="_blank", damit eine neue Seite/Tab aufgeht.

Beispiel:

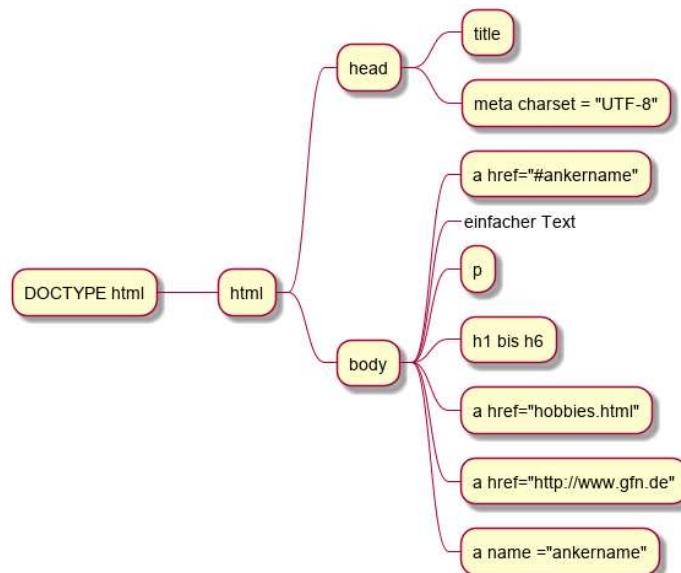
```
<a href="http://www.html-seminar.de/" target="_blank">HTML-Tutorial</a>
```



23

23

Referenzierungen (Verweise)

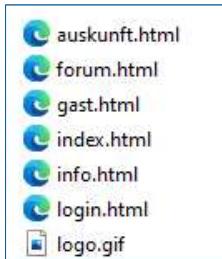


24

24

Aufgabe: 6 Formulare erstellen

- Home → index.html
- Reisesuche → auskunft.html
- Touristeninfo → info.html
- Anmeldung → login.html
- Gästebuch → gast.html
- Forum → forum.html



Reisebüro h1

[Home | Reisesuche | Touristeninfo | Anmeldung | Gästebuch | Forum]

Herzlich willkommen! h2

Wir sind Ihr Online-Reisebüro, spezialisiert auf Flugreisen in europäische Metropolen und die entsprechenden Hotelaufenthalte. Wir wünschen Ihnen einen angenehmen Aufenthalt - auf unserer Website und in Ihrem Urlaub!

index.html

Aktuelle Seite ist kein Link
(Einfacher Text)



25

25

Aufgabe

Reisesuche → auskunft.html

Reisebüro h1

[Home | Reisesuche | Touristeninfo | Anmeldung | Gästebuch | Forum]

Reiseauswahl h2

Aktuelle Seite ist kein Link



26

26

Aufgabe

Touristeninfo → info.html

Reisebüro

[Home | Reisesuche | Touristeninfo | Anmeldung | Gästebuch | Forum]

Touristeninfo

Info über Reiseziel: [Bitte wählen] ↗ OK

Aktuelle Seite ist kein Link

27



27

Aufgabe

Anmeldung → login.html

 EuroCityTravel

[Home | Reisesuche | Touristeninfo | Anmeldung | Gästebuch | Forum]

Kundenanmeldung

Bitte geben Sie hier Ihre E-Mail-Adresse ein, falls Sie bereits registriert sind:

Aktuelle Seite ist kein Link

img → logo.gif

28



28

Aufgabe

Gästebuch → gast.html



Aktuelle Seite ist kein Link



29

29

Aufgabe

Forum → forum.html



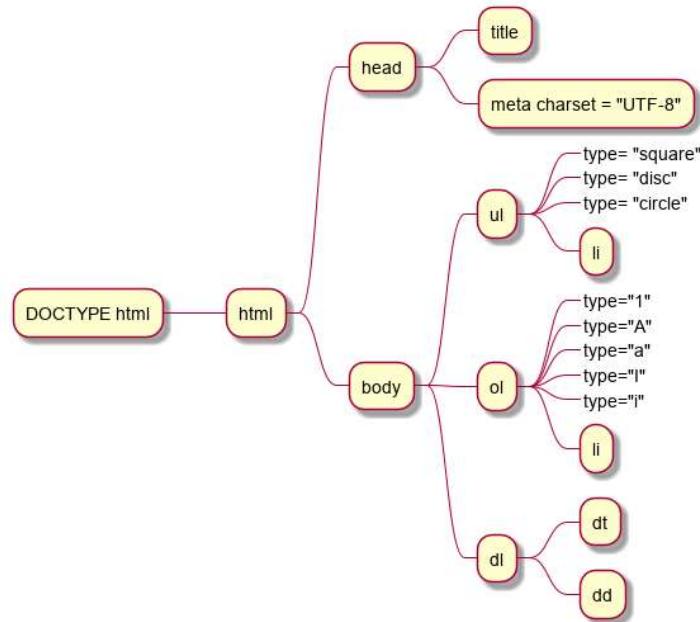
Aktuelle Seite ist kein Link



30

30

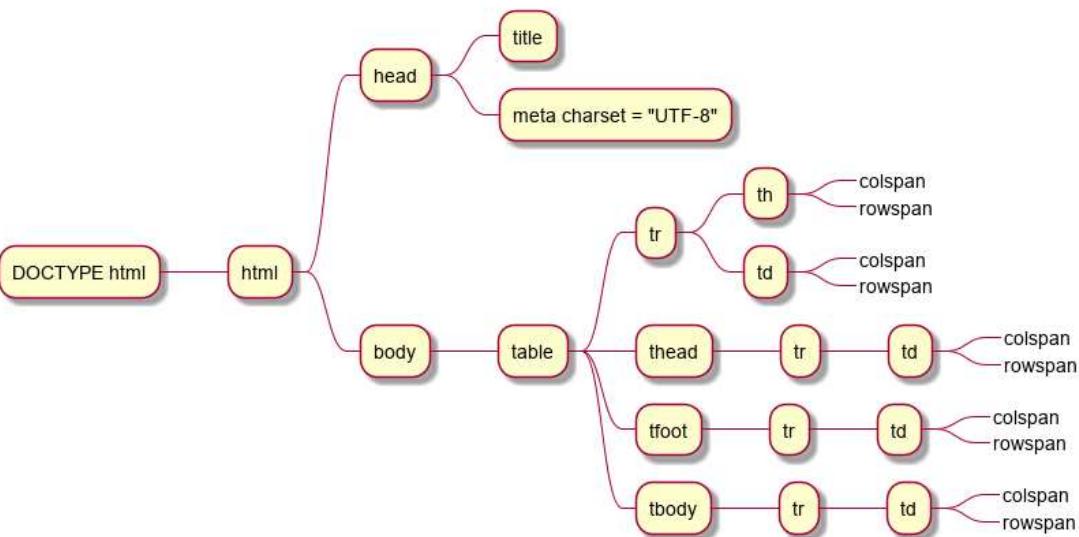
Listen



31

31

Tabellen



32

32

HTML Tags

Container

Beispiel:

- <title> </title>
- <body> </body>
- <table> <table>
- <h1> </h1>
- <h2> </h2>
- <p> </p>
- <pre> </pre>

NON Container

Beispiel:

- <meta/>
- <link...../>
- <img...../>
- <input.....>
-

- <hr/>



33

33

HTML - Validator

<https://validator.w3.org/>

W3C® Markup Validation Service
Check the markup (HTML, XHTML, ...) of Web documents

Validate by URI Validate by File Upload Validate by Direct Input

Validate by URI
Validate a document online:
Address:
[More Options](#)

This validator checks the [markup validity](#) of Web documents in HTML, XHTML, SMIL, MathML, etc. If you wish to validate specific content such as RSS/Atom feeds or CSS stylesheets, [MobileOK content](#), or to [find broken links](#), there are [other validators and tools](#) available. As an alternative you can also try our [non-DTD-based validator](#).



Interested in "developing" your developer skills? In W3Cx's hands-on Professional Certificate Program, learn how to code the right way by creating Web sites and apps that use the latest Web standards. [Find out more!](#)

[Donate](#) and help us build better tools for a better web.

[Home](#) [About...](#) [News](#) [Docs](#) [Help & FAQ](#) [Feedback](#) [Contribute](#)



COPYRIGHT © 1994-2013 W3C® (MIT, ERIC, KEIO, BEIHANG). ALL RIGHTS RESERVED. W3C LIABILITY, TRADEMARK, DOCUMENT USE AND SOFTWARE LICENSING RULES APPLY. YOUR INTERACTIONS WITH THIS SITE ARE IN ACCORDANCE WITH OUR PUBLIC AND MEMBER PRIVACY STATEMENTS.



34

34

HTML und CSS - Cascading Style-Sheet



+

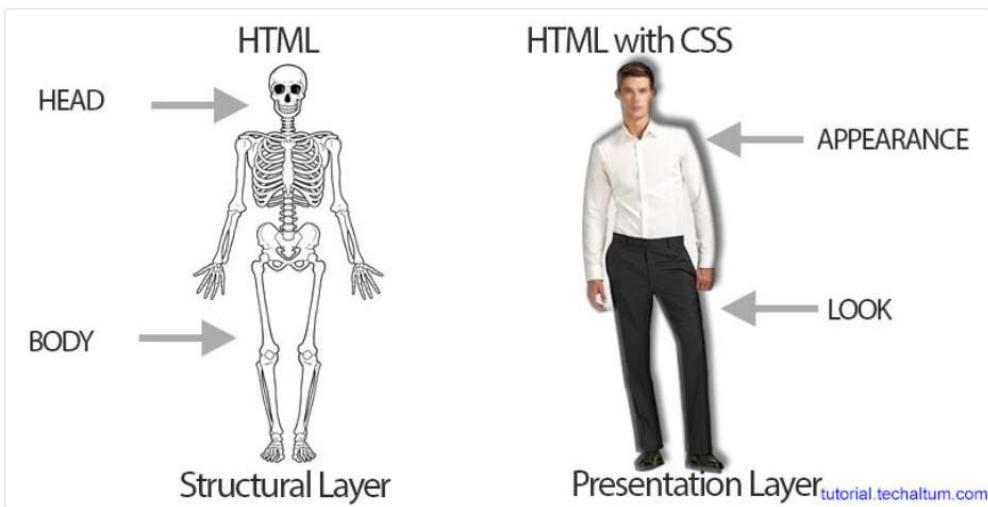


Cascading – kaskadierend, stufenartig

Style Sheets – „Stilblätter“

... enthalten Codes zum Formatieren eines HTML-Elements

Warum CSS?



HTML und CSS



37



37

HTML und CSS

Globale Einstellung des Browsers

Stilvorlage als CSS-Datei

Integrierte Stilvorlage im HEAD

Lokale Angabe im HTML-Tag

Lokale Angabe im HTML-Tag



38

38

19

HTML und CSS

Vorteile

- Das Design ist konsistent zwischen den Browsern
- **Kürzere Zeiten zur Erstellung und Pflege von Websites**
- CSS macht zahlreiche HTML-Wiederholungen überflüssig
- Weniger Code bedeutet schnelleren Download
- Auch bei Browsern, die CSS nicht unterstützen, wird die Website noch angezeigt (dann allerdings ohne CSS-Design)

Nachteile

- **Unterschiedliche Unterstützung von CSS3 durch Browser:**

Ein kleiner Nachteil ist die teilweise uneinheitliche Unterstützung von CSS3 durch Browser

Deshalb sollten nicht alle CSS3-Definitionen eingesetzt bzw. mit Präfixen gearbeitet werden



39

39

CSS - Kaskade

Stylesheets

Browserstylesheets

- Erstes Stylesheet, das angewendet wird
- Lehnt an den Initialwerten von CSS an
- Jeder Browser hat eigene Standardeinstellungen
- Browser besitzen vorgegebene Formatierungen für HTML-Dokumente

Userstylesheets

- Browser-Konfigurations-einstellungen der Benutzer
- Beispielsweise.
 - Schriftarten
 - Vorder- oder Hintergrundfarbe
 - Darstellung besuchter oder unbesuchter Links
- Überschreibt die Browdereinstellungen

Autorenstylesheets

- Referenzierte Stylesheets im HTML-Dokument
- Eingebundene Stylesheets durch Import-Regel
- Kaskade im Gesamtstylesheet
- Überschreiben Browser- und Benutzer-Stylesheets



40

40

CSS – Cascading Style Sheets

4 Varianten:

1. direkt im Quellcode, beim <tag>
2. am Anfang der HTML-Datei (style-Block)
3. Import einer externen CSS-Datei innerhalb des Style-Tags
4. ausgelagert in eine separate CSS-Datei



41

CSS - Kaskade

1.

```
<h4 style="background-color: silver">Inline-CSS</h4>
```
2.

```
<style>
h4 { background-color: silver; }
</style>
```
3.

```
<style>
@import url("style.css") screen;
</style>
```
4.

```
<link rel="stylesheet" type="text/css" href="style.css" media="screen" />
```

Hier: Datei „style.css“ muss in demselben Verzeichnis sein, wie das HTML-Dokument!



42

42

CSS am Anfang der HTML-Datei/Inline CSS

```
<!DOCTYPE html>
<head>
<style>
→ h1 {
  color: white;
  background-color: #006699;
}
</style>
</head>
→ <h1> jetzt html lernen mit CSS </h1>
→ <h2 style="background-color: silver"> Inline CSS </h2>
<body>
</body>
</html>
```

jetzt html lernen mit CSS

Inline CSS



43

43

CSS - Externe CSS

```
<!DOCTYPE html>
<head>
  <link rel="stylesheet" href="design.css"/>
</head>
  <h1> jetzt html lernen mit CSS </h1>
  <h2> Inline CSS </h2>
<body>
</body>
</html>
```

design.css

```
h1 {
  color: white;
  background-color: #006699;
}
h2 {
  background-color: silver;
}
```

jetzt html lernen mit CSS

Inline CSS



44

44



Live Demonstration


HTML und CSS


45

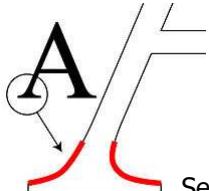
45

CSS - Styles für Textauszeichnungen

- font-family
- font-size
- font-style
- font-weight
- text-decoration
- line-height

1.
2.
3.

```
body{
    font-family: "Helvetica Neue", Helvetica, sans-serif;
}
```

Priorisierung, Suche nach der vorh. Schriftart
in unterschiedlichen Umgebungen, serifenos


Serifen: (umgangssprachlich auch „Füßchen“ genannt)
sind die **häkchenartigen Enden an Buchstaben** vieler Antiquaschriften



46

Aufgabe: CSS – Kaskadierend !

Welche Farbe hat der Text ?

```
h1 {
    color: navy;
}
```

Externes Stylesheet

```
<head>
<link rel="stylesheet" type="text/css" href="mystyle.css">
<style>
h1 {
    color: orange;
}
</style>
</head>
```

Internes
Stylesheet



47

Aufgabe: CSS – Kaskadierend !

Welche Farbe hat der Text ?

```
h1 {
    color: navy;
}
```

Externes Stylesheet

```
<head>
<style>
h1 {
    color: orange;
}
</style>
<link rel="stylesheet" type="text/css" href="mystyle.css">
</head>
```

Internes
Stylesheet

Es spielt eine Rolle, wo die Verlinkung zum Stylesheet platziert wird.
Deshalb besser im Head-Bereich, damit es keine Überraschungen gibt!



48

CSS – Pseudoformate für Hyperlinks :focus / :hover

```

    ::link
      Nichtbesuchter Hyperlink.
    ::visited
      Besuchter Hyperlink
    ::active
      Aktiver Hyperlink (Wenn der User mit der Maus auf den Link klickt.)
    ::hover
      Mouseover-Effekt. Wenn der User den Mauszeiger über einen Hyperlink oder ein anderes Element bewegt.
    ::focus
      Wenn ein Element den Fokus hat. Z.B. Texteingabefelder, in denen der Cursor steht.
    ::first-child
      Ein Element, das das erste Kind eines anderen Elements ist.
  
```

Heres a cool way to remember the order of the four
selectors, "LVHA (link visited hover active)".....

Just think 'LOVE' (LV) and 'HATE' (HA)....

```

<style type="text/css">
  a:link {color:#0000ff;}
  a:visited {color:#ddddd;}
  a:hover {color:#ff0000;}
  a:active {color:#ff0000;}

</style>

<title>Irgendein Titel</title>
<body>
  <a href="datei.htm">Klick</a>
</body>
</html>
  
```



49

49

Aufgabe: Externe CSS anlegen - Teil 1

CSS-Datei mit dem Namen: **main.css** anlegen und in allen 6 html-Dokumenten „verlinken“

Folgende Vorgaben gelten:

body: background-color: #000066; color: #000066;  blau
 font-family: Verdana, Arial, Helvetica, sans-serif; font-size: 12px;

td: background-color: #3399CC; color: #000066;
 font-family: Verdana, Arial, Helvetica, sans-serif; font-size: 12px;

a: color: #003300; text-decoration: none



50

50

Aufgabe: Externe CSS anlegen - Teil 2

CSS-Datei mit dem Namen: **main.css** erweitern

Folgende Vorgaben gelten:

```
a:link      color: #003300
a:visited   color: #000000
a:hover     color: #3399CC; background-color: #000066
a:active    color: #000000
```



51

51

CSS Selektoren



Mehrfache Selektoren (tags) mit Komma getrennt möglich:
p, h1, h2 { : ...; }

Jede Deklaration muss mit ; beendet werden

```
Selektor1 [, Selektor2 [, ...] ] {
  Eigenschaft-1 : Wert-1;
  ...
  Eigenschaft-n: Wert-n[;]
}
/* CSS-Kommentar */
/* In eckigen Klammern stehen optionale Angaben */
```



52

52

26

CSS Selektoren

- Universal-Selektor
- Typ-Selektor
- Class-Selektor (Klassen)
- ID-Selektor

	HTML-Datei	CSS-Datei
ID-Selektor	<h1 id = "farbe1">	#farbe1 { color:blue; }
Klassen-Selektor	<h1 class = "farbe1">	.farbe1 { color:blue; }
Typ-Selektor	<h1>	h1 id = "farbe1"



53

53

CSS Universal-Selektor

Universal-Selektoren verknüpfen jedes vorhandene Element mit Stylesheetangaben

```

<html><head>
<style type="text/css">

  * {
    font-size:14px;
    color:#ff0000;}

</style>

<title>Irgendein Titel</title>
</head><body>

<h1>Ein "h1"-Tag mit roter Schrift</h1>
<h2>Ein "h2"-Tag mit roter Schrift</h2>
<h3>Ein "h3"-Tag mit roter Schrift</h3>
<p>Ein "p"-Tag mit roter Schrift</p>
<div>Ein "div"-Tag mit roter Schrift</div>

</body></html>

```



54

54

CSS Typ-Selektor

Typ-Selektoren verknüpfen einen Elementtyp, also ein HTML-Tag mit einer Formatierung.

```

<html><head>
<style type="text/css">

h1
{
  font-size:14px;
  font-style:italic;
}
p {color:#ff0000;}

</style>

<title>Irgendein Titel</title>
</head><body>

  <h1>Eine kursive Überschrift</h1>
  <p>Und roter Text</p>

</body></html>

```



55

55

CSS Class-Selektor

Klassen sind Selektoren mit beliebigen Namen

```

<html><head>
<style type="text/css">

  h1 { font-size:14px; font-style:italic; }
  h1.gerahmt { border:1px solid #aaaaaa; }

</style>

<title>Irgendein Titel</title>
</head><body>

  <h1>Eine kursive Überschrift</h1>
  <h1 class="gerahmt">Eine kursive Überschrift mit Rahmen</h1>

</body></html>

```



56

56

CSS Class-Selektor

Man darf das Tag vor dem Punkt aber auch weglassen, sodass die Klasse für alle Tags gültig ist

```

<html><head>
<style type="text/css">
h1 { font-size:14px; font-style:italic; }
.gerahmt { border:1px solid #aaaaaa; }

.einfachertext { font-size:14px; }

</style>

<title>Irgendein Titel</title>
</head><body>

<h1>Eine kursive Überschrift</h1>
<h1 class="gerahmt">Eine kursive Überschrift mit Rahmen</h1>
<p>Ein Text</p>

<p class="gerahmt">Ein Text mit Rahmen</p>
<p class="einfachertext">Ein Text, 14 Pixel groß</p>
<div class="einfachertext">Ein Text, 14 Pixel groß</div>
</body></html>

```



57

57

CSS Class-Selektor

Zuweisung mehrerer Klassen zu einem Element

```

<html><head>
<style type="text/css">
.rot{color:#ff0000;}
.ul {text-decoration:underline}
.fett{font-weight:bold}

</style>

<title>Irgendein Titel</title>
</head><body>

<p class="rot">Ein roter Text</p>
<p class="fett">Ein fetter Text</p>
<p class="ul">Ein unterstrichener Text</p>
<p class="ul rot">Ein Text, rot und unterstrichen</p>
<p class="fett ul">Ein Text, fett und unterstrichen</p>
<p class="rot fett ul">Ein Text, rot, fett und unterstrichen</p>

</body></html>

```



58

58

CSS ID-Selektor und DIV-Tag

- Über die beiden Befehle `<div>` und `` können Bereiche festgelegt werden, die keine semantische Bedeutung haben, sprich geschmacksneutral sind
- Die beiden neutralen Befehle können andere HTML-Befehle beherbergen. Dadurch können Container gebildet werden, die dann einen bestimmten Zweck haben: Diese Bereiche werden dann gerne für Design genutzt

```
<div> ... </div>
```

div = engl. Division=Bereich

```
<div id="kopf">
<div id="navi">
<div id="inhalt">
```

Die Benennung sollte mit a-z, 0-9 und ohne Leerzeichen und Sonderzeichen erfolgen



59

59

CSS ID-Selektor und DIV-Tag

ID-Selektoren verknüpfen Elemente über das ID-Attribut mit einer Formatierung

```
<html><head>
<style type="text/css">
div {background-color:#efefef; border:1px solid #aaaaaa;}
div#a
{
position:absolute;
top:100px; left:100px;
width:120px; height:80px;
}
#b
{
position:absolute;
top:150px; left:300px;
width:120px; height:80px;
}

</style>
<title>Irgendein Titel</title>
</head><body>
<div id="a">100 Pixel von oben und 100 Pixel von links</div>
<div id="b">150 Pixel von oben und 300 Pixel von links</div>
</body></html>
```



60

60

CSS - Validator

<http://jigsaw.w3.org/css-validator/>

The screenshot shows the W3C CSS Validation Service interface. At the top, there's a navigation bar with links to various languages. Below it is a main form area with tabs for "per URI", "per File-Upload", and "per Direkteingabe". A sub-section titled "Validierung per URI" asks for a URL to validate. There's also a "Weitere Optionen" link and a "Prüfen" button. In the center, there's a small circular logo with "W3C" and some text around it. Below the logo, there's a message about developer skills and a "Find out more!" link. Further down, there's a "Donate" link. The footer contains links for "About", "Documentation", "Download", "Feedback", and "Credits". It also includes the W3C Quality Assurance logo and a "VALIDATOR" link. On the right side of the footer, there's a "GFN" logo.

61

61

HTML – Formular-Elemente

The diagram illustrates various HTML form elements arranged around a central blue cloud containing the word "Formular".

- Textarea:** A large text input area labeled "Text eingeben".
- Textbox:** A standard text input field labeled "irgendein Text".
- Checkbox:** A group of checkboxes with one checked: "Wasser" (checked), "Kaffee" (unchecked), and "Tee" (unchecked).
- Radio Button:** A group of radio buttons with one selected: "Java" (selected), "PHP" (unchecked), and "XML" (unchecked).
- Button:** A standard button labeled "Speichern".
- Listbox:** A dropdown list box containing the numbers 1, 2, and 3.

62

62

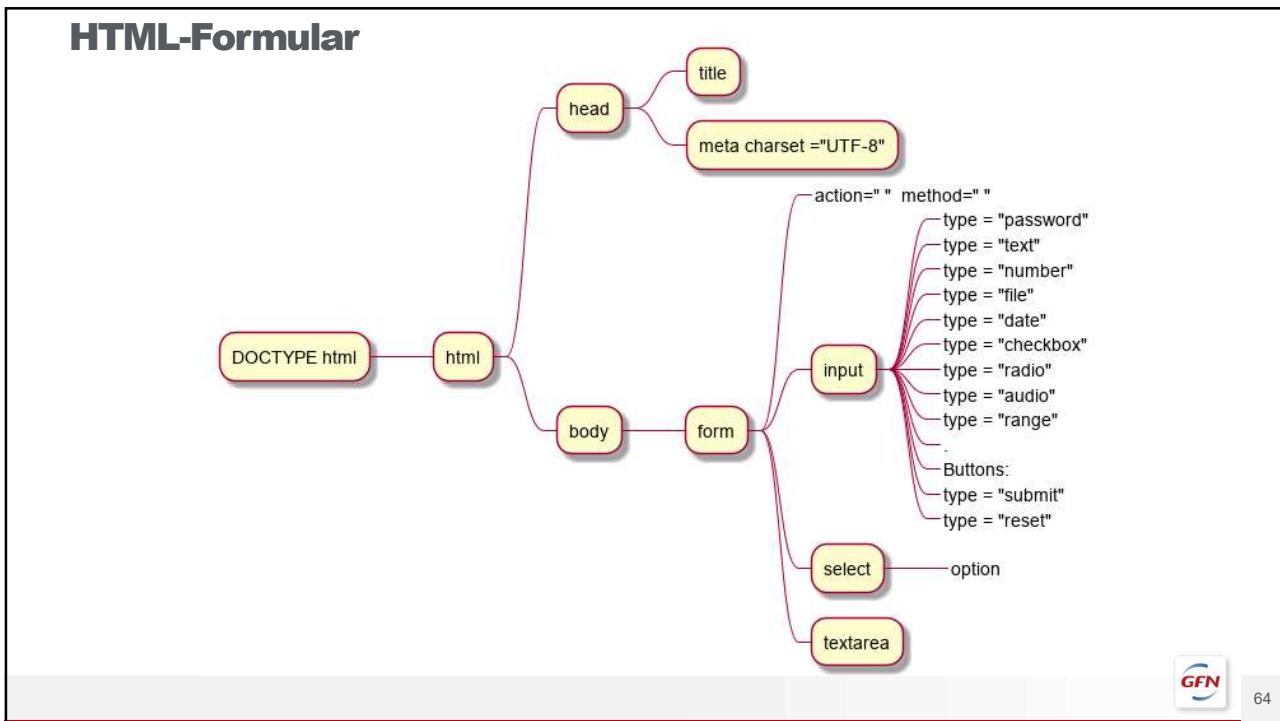


Live Demonstration

 HTML Formular

GFN 63

63



64

Method GET/POST

```
<form action ="sendenhtmlphp.php" method = "GET">
  <table>
    <tr>
      <td> Vorname: </td>
      <td> <input type = "text" name = "vorname" /> </td>
    </tr>
    <tr>
      <td>Nachname:</td>
      <td><input type = "text" name = "nachname" /> </td>
    </tr>
    <tr>
      <td> <input type = "submit" value = "Senden" /> </td>
    </tr>
  </table>
</form>
```

In PHP können Variablen von einer Seite zu einer anderen Seite übertragen werden. Für diesen Anwendungsfall gibt es zwei Methoden, die sich in der Art der Übertragung an die nächste Seite unterscheiden. Die beiden Methoden sind **GET** und **POST**



65

65

Methode GET

- Überträgt die Variablen in Adresszeile (URL) sichtbar (Vor- und Nachteil)

- Limitierte Anzahl / Länge an Variablen

```
file:///C:/xampp/htdocs/test/sendenhtmlphp.php?vorname=Klara&nachname=Fall
```

- Die Ergebnisseite eines GET-Formulars kann man bookmarken, da alle nötigen Informationen in der URL enthalten sind



66

66

Methode POST

```

1 <?php
2 echo "eingegebener Vorname: ". $_POST['vorname'];
3 echo "";
4 echo "eingegebener Nachname: ". $_POST['nachname'];
5 ?>
6
7
8 <form action ="sendenPOSThtml.php" method = "post">
9   <table>
10    <tr>
11      <td> Vorname: </td>
12      <td> <input type = "text" name = "vorname" /> </td>
13    </tr>
14    <tr>
15      <td>Nachname:</td>
16      <td><input type = "text" name = "nachname" /> </td>
17    </tr>
18    <tr>
19      <td> <input type = "submit" value = "Senden" /> </td>
20    </tr>
21  </table>
22 </form>
23

```

Notice: Undefined index: vorname in C:\xampp\htdocs\test\sendenPOSThtml.php.php on line 2
eingegebener Vorname:
Notice: Undefined index: nachname in C:\xampp\htdocs\test\sendenPOSThtml.php.php on line 4
eingegebener Nachname:
Vorname: Klara
Nachname: Fall
Senden

eingegebener Vorname: Klara eingegebener Nachname: Fall
Vorname:
Nachname:
Senden

```

1 eingegebener Vorname: Klara eingegebener Nachname: Fall
2
3 <form action ="sendenPOSThtml.php" method = "post">
4   <table>
5     <tr>
6       <td> Vorname: </td>
7       <td> <input type = "text" name = "vorname" /> </td>
8     </tr>
9     <tr>
10      <td>Nachname:</td>
11      <td><input type = "text" name = "nachname" /> </td>
12    </tr>
13    <tr>
14      <td> <input type = "submit" value = "Senden" /> </td>
15    </tr>
16  </table>
17 </form>

```



67

67

Methode POST

- Überträgt die Variablen für den Client unsichtbar im Hintergrund
- Unlimitierte Anzahl / Länge an Variablen
- Die Ergebnisseite eines POST-Formulars kann man weder bookmarken, noch im Browser aktualisieren, da die Daten nicht mehr zur Verfügung stehen
- File-Upload ist nur mit POST möglich



68

68

Aufgabe

- Reisesuche → auskunft.html, action = "ergebnis.php", method= "post"

Reisebüro

[[Home](#) | [Reisesuche](#) | [Touristeninfo](#) | [Anmeldung](#) | [Gästebuch](#) | [Forum](#)]

Reiseauswahl

Abreiseort:	<input type="text" value="[[Bitte wählen] ▾"/>	Die meisten Flugangebote gehen von/nach Köln/Bonn, einige auch von/nach Frankfurt oder Düsseldorf.
Zielort:	<input type="text" value="[[Bitte wählen] ▾"/>	
Abreisedatum:	Datum:	
Rückreisedatum:	Datum:	
Mit Hotellangebot?	<input checked="" type="radio"/> ja <input type="radio"/> nein	



69

69

Aufgabe

- Touristeninfo → info.html, action = "info.php", method= "get"

Reisebüro

[[Home](#) | [Reisesuche](#) | [Touristeninfo](#) | [Anmeldung](#) | [Gästebuch](#) | [Forum](#)]

Touristeninfo

Info über Reiseziel:



70

70

Aufgabe

2 Formular-Bereiche!

- Anmeldung → login.html
- action = "testlog.php", method= "post,"
- Neuer Kunde
- action = "newuser.php",
- method= "post"

EuroCityTravel

Kundenanmeldung

Bitte geben Sie hier Ihre E-Mail-Adresse ein, falls Sie bereits registriert sind:

Anmerkung: Hier fehlt zur Praxistauglichkeit eine Passwortüberprüfung.

Neuer Kunde

Bitte geben Sie hier Ihre Daten ein, falls Sie ein neuer Kunde sind (die mit * markierten Felder sind Pflichtangaben):

* E-Mail:

Dies wird Ihre Anmelde-ID.

* Name:

* Vorname:

* Geburtsdatum:

Straße:

Hausnummer:

Postleitzahl:

Wohnort:

* Land: [Bitte wählen]

Telefon:



71

71

Aufgabe

- Gästebuch → gast.html
- action = "gast.php",
- method= "post"

Reisebüro

[Home | Reisesuche | Touristeninfo | Anmeldung | Gästebuch | Forum]

Neuer Gästebucheintrag

[Einträge lesen](#)

Nickname:

E-Mail (optional):

Titel:

Ihr Eintrag:



72

72

Aufgabe

- Forum → forum.html
- action = "forum.php",
- method= "post"

Reisebüro

[[Home](#) | [Reisesuche](#) | [Touristeninfo](#) | [Anmeldung](#) | [Gästebuch](#) | [Forum](#)]

Forumsbeitrag verfassen

Einträge lesen

Nickname:

E-Mail (optional):

Titel:

Ihr Eintrag:



73

73

Aufgabe

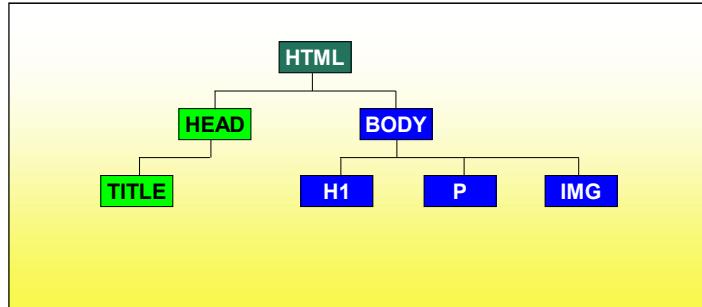
- Alle 6 html-Dokumente als *.php's speichern
- Verlinkungen entsprechend anpassen



74

74

Vererbung mit CSS



Vererbung gilt auch für die meisten Eigenschaften bei CSS



75

Vererbung mit CSS

```

<html>
  <head>
    <title>Mein Titel</title>
    <link rel="stylesheet" href="styles.css" type="text/css">
  </head>
  <body>
    <h1>Hier kommt eine Überschrift</h1>
    <p>Es folgt ein Absatz mit ein wenig Text</p>
    <table>
      <tr>
        <td>Text Spalte 1</td>
        <td>Text Spalte 2</td>
        <td>Text Spalte 3</td>
      </tr>
    </table>
    <!--Hier folgt noch eine Liste-->
    <ul>
      <li>Krä</li>
      <li>Bla</li>
      <li>Schwall</li>
      <li>Blubb</li>
      <li>Blubber</li>
    </ul>
  </body>
</html>
  
```

Eltern- und Kind-Elemente?

Also <body> ist das Eltern-Element von <h1>, <p>, <table> und .

<table> ist das Eltern-Element von den Kind-Elementen <tr> und <td>,

 ist das Eltern-Element von den Kind-Elementen .

Man kann sich so ein HTML-Dokument also als eine Art von Stammbaum vorstellen



76

Einfache Vererbung mit CSS

```
td, p
{
  font-family: Arial, Helvetica, sans-serif;
  ...
}
a:link, a:active, a: visited, a:hover
{
  font-family: Arial, Helvetica, sans-serif;
  ...
}
.fliesstext
{
  font-family: Arial, Helvetica, sans-serif;
  ...
}
```

Dank des Vererbungsprinzips reicht in diesem Fall eine einzige Anweisung aus

```
body
{
  font-family: Arial, Helvetica, sans-serif;
}
```

body-Element vererbt die Eigenschaft Schriftart Arial, Helvetica, sans-serif auf alle ihre Kindelemente weiter.



Vererbung mit CSS

Beispiel:

```
<div id="box1">
  <p>Hier ein Text.</p>
  <ul>
    <li>Eintrag 1</li>
    <li>Eintrag 2</li>
  </ul>
  <p> Hier ein weiterer Text.</p>
</div>
```

Wunsch: Alle Elemente in box1 sollen anders aussehen (Schriftart, -größe), als der Rest des Dokuments

Weitere Elementtypen?

```
#box1 p {
  font-family: Arial, sans-serif;
  font-size: 85%;
}
#box1 li {
  font-family: Arial, sans-serif;
  font-size: 85%;
}
```



Vererbung mit CSS

```
#box1 {
font-family: Arial, sans-serif;
font-size: 85%;
}
```

Die definierte Schriftart und Schriftgröße wird an alle Kindelemente des Elements mit der id="box1" vererbt, in diesem Falle an die Elemente <p> und .

Hier können Sie schon erkennen, dass es vorteilhaft ist, gewisse Basiseigenschaften wie Schriftarten, Schriftgrößen und Farben möglichst früh, zum Beispiel für das <body>-Element, zu bestimmen, so dass diese nicht für jedes Element im Dokument wiederholt werden müssen.



79

Erweiterte Vererbung mit CSS

```
p
{
  color: #000;
  font-size: 11px;
}
```

+ an diversen Stellen Absätze nach rechts um 20 px einrücken.

Klassendefinition:

```
.einrueck
{
  margin-left: 20px;
}

<p class="einrueck">Dieser Text rückt um 20 Pixel ein</p>
```

Damit erbt dieser Absatz die Eigenschaften der allgemeinen Absatz-Definition **und** die zusätzliche Eigenschaft der Klasse .einrueck.



80

Mehrfach-Vererbung mit CSS

```
.kasten
{
  width: 400px;
  height: 200px;
  border: 1px solid #000;
  background-color: #fff;
}
...
<div class="kasten">...</div>
```

Wir benötigen auf bestimmten Seiten eine fette Schrift für das <div>-Element kasten

Weitere Klasse

```
.fett
{
  font-weight: bold;
}
```

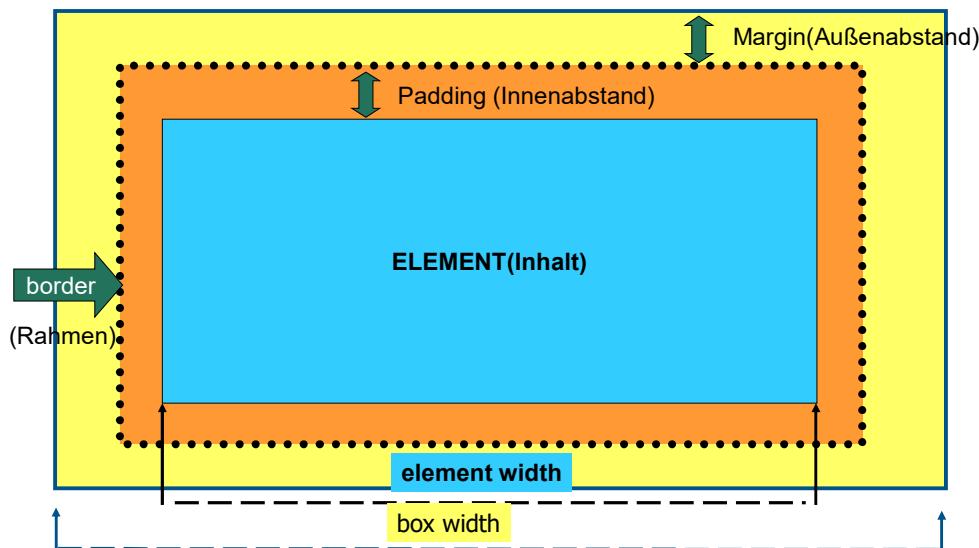
```
<div class="kasten fett">
```

Mögliche Zuweisungen

```
<div class="fett kasten">
```



Boxmodell



Boxmodell

Layout mit BoxModell

DIV-Container

- Breite (width)
- Höhe (height)
- Hintergrundfarbe (background-color)
- Innenabstand (padding)
- Rahmen (border)
- Außenabstand (margin)

Breite und Höhe ?

Die Gesamtbreite bzw. die Gesamthöhe einer Box ergibt sich aus der Addition von:

- der Breite bzw. Höhe des Elementinhalts (width)
- des Innenabstands (padding)
- der Rahmenstärke (border-width) und
- des Außenabstands (margin)



83

83

CSS – Border

Rahmenarten

- solid = durchgezogen
- double = doppelt
- none = kein Rahmen
- hidden = kein Rahmen
- dotted = gepunktet
- dashed = gestrichelt
- groove = 3D-Effekt
- ridge = 3D-Effekt
- inset = 3D-Effekt
- outset = 3D-Effekt

Aufgabe: CSS und Rahmen (border)

Überschrift h1

Hier kommt ein normaler Absatz

Hier kommt Überschrift 2

Kurzschreibweise

```
border-color: green;
border-width: 5px;
border-style: solid;
```

```
border: green 5px solid;
```



84

84

CSS – Border left, right...

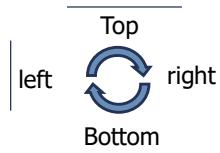
Angabe der Position des Rahmens: top, right, bottom und left.

```
border-top-style: ...;
border-right-style: ...;
border-bottom-style: ...;
border-left-style: ...;
```

Kurzschreibweise

```
border-top: green 20px solid;
```

```
border-top: ...;
border-right: ...;
border-bottom: ...;
border-left: ...;
```



85

85

CSS – Border Color

```
<!DOCTYPE HTML>
<html>
<head>
<title>Rahmen um einen Text</title>
<style>
h1{
    border:2px solid Violet;
}

h2{
    border-top:10px solid lightgrey;
    border-left:10px solid lightgrey;
}
p{
    border-style: none dotted dotted none;
    border-color: orange;
    border-width:4px;
}
</style>
</head>
<body>
    <h1> Wichtigste „Überschrift h1“</h1>
    <h2> „Überschrift h2“ </h2>
    <p>
        Und nun ein normaler Absatz mit
        mehreren Zeilen damit die Texteinrückung in der ersten Zeile gut
        zur Geltung kommt.
    </p>
</body>
</html>
```

Wichtigste Überschrift h1

Überschrift h2

Und nun ein normaler Absatz mit mehreren Zeilen damit die Texteinrückung in der ersten Zeile gut zur Geltung kommt.



86

86

Background - ..., border, margin, ...

```
<!DOCTYPE html>
<html>
<head>
    <title>Box-Modell</title>
    <link href="style.css" type="text/css" rel="stylesheet">
</head>
<body>
    <p class="rot">
        Lorem ipsum dolor sit amet, consetetur sadipscing elitr,
        sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat,
        sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet
        clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet.
        Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod
        tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua.
        At vero eos et accusam et justo duo dolores et ea rebum.
        Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet.
    </p>
    <p>
        Lorem ipsum dolor sit amet, consetetur sadipscing elitr,
        sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat,
        sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet
        clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet.
        Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod
        tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua.
        At vero eos et accusam et justo duo dolores et ea rebum.
        Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet.
    </p>
</body>
</html>
```

style.css

```
p {
    background-color: yellow;
    border-color: black;
    border-width: 5px;
    border-style: dotted;

    padding: 50px;
    margin: 60px;

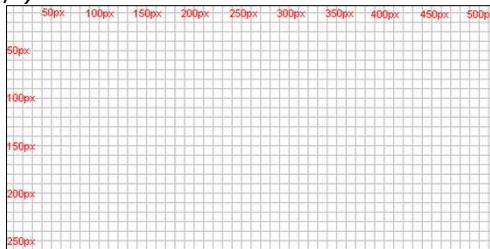
    width: 640px;
}

.rot {
    color: red;
    font-weight: bold;
}
```

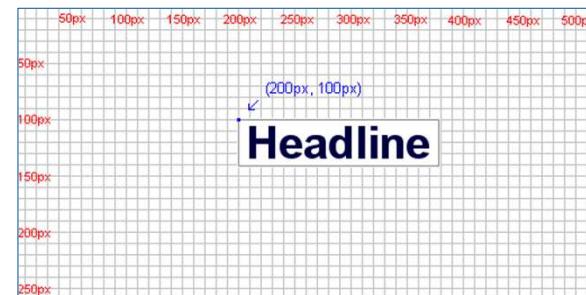


Absolute Positionierung von Elementen

(0,0)

**Headline**

```
h1 {
    position: absolute;
    top: 100px;
    left: 200px;
}
```



Relative Positionierung von Elementen

```
<!DOCTYPE html>
<html>
<head>
    <title>Box-Modell</title>
    <link href="style.css" type="text/css" rel="stylesheet">
</head>
<body>
    <p class="rot">
        Lorem ipsum dolor sit amet, consetetur sadipscing elitr,  

        sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat,  

        sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet  

        clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet.  

        Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod  

        tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua.  

        At vero eos et accusam et justo duo dolores et ea rebum.  

        Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet.
    </p>
    <p class="weiss">
        Lorem ipsum dolor sit amet, consetetur sadipscing elitr,  

        sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat,  

        sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet  

        clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet.  

        Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod  

        tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua.  

        At vero eos et accusam et justo duo dolores et ea rebum.  

        Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet.
    </p>
</body>
</html>
```

style.css

```
p { background-color: yellow;  
border-color: black;  
border-width: 5px;  
border-style: dotted;  
padding: 50px;  
margin: 10px;  
width: 640px; }  
.rot { position: relative;  
left: 150px;  
bottom: 10px;  
color: red;  
font-weight: bold; }  
.weiss { background-image: url("hintergrund.jpg");  
color: white;  
font-weight: bold; }
```



Float

Die float Eigenschaft gibt an, wie ein Element „schweben“ soll

Werte: none, left, right, initial, inherit

```
<!DOCTYPE html>
<html>
<head> No Float</head>
<style>
.img-nofloat{width:200px;height:150px;float: none;}
</style>
<body>
<p>Image will be displayed just where it occurs .</p>
<p></p>
Image displayed float none, Image displayed float none, Image disp  
displayed float none, Image displayed float none, Image  
displayed float none,Image displayed float none,Image disp  
displayed float none,Image displayed float none,Image float to the  
left,Image displayed float none,Image displayed float none ,Image  
,Image displayed float none,Image displayed float none ,Image  
displayed float none ,Image displayed float none ,Image displayed  
displayed float none ,Image displayed float none,Image float to t  
left ,Image displayed float none ,Image displayed float none ,Im  
</body>
</html>
```



Image displayed float none, Image displayed float none, Image displayed float none,
Image displayed float none, Image displayed float none, Image displayed float none,Image
displayed float none,Image displayed float none,Image displayed float none,Image float to the left,Image displayed
float none,Image displayed float none,Image displayed float none,Image displayed float none,Image displayed float
none,Image displayed float none,Image displayed float none,Image displayed float none,Image displayed float none
Image displayed float none,Image float to the left ,Image displayed float none ,Image displayed float none,Image
displayed float none.



Float

```
<style>
.img-right{width:200px; height:150px; margin-left:15px; float: right;}
</style>
<body>
```

Float Right

The image will float to the right.

Image float to the right. Image float to the right.



```
<style>
.img-left{width:200px; height:150px; margin-right:15px; float: left;}
</style>
```

Float Left

The image will float to the left.



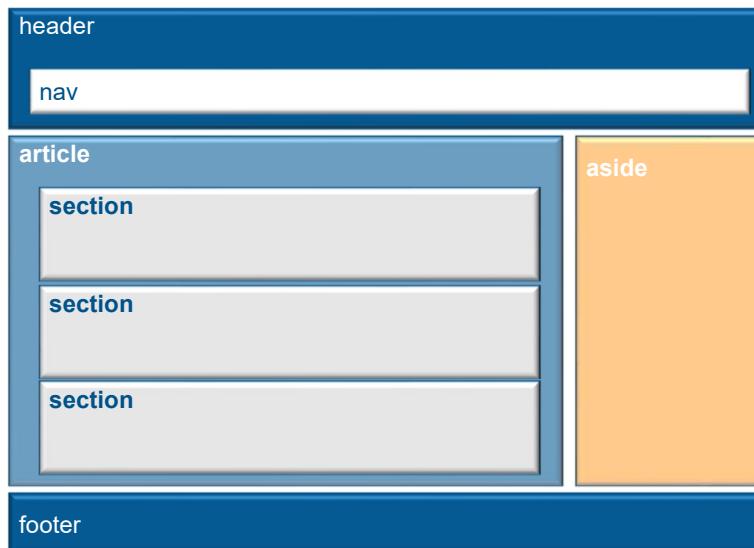
Image float to the left. Image float to the left.



91

91

Layout mit HTML5



92

92

Layout mit HTML5

- <footer> Fußbereich (bisher über div id= "fuß")
- <header> Kopfbereich (bisher über div id= "kopf")
- <nav> Steuerung (bisher über div id= "nav")
- <section> Inhalt (bisher über div id= "inhalt")

- <article>
- <aside>
- <dialog>
- <figure>



Aufgabe

Gruppenaufgaben:

3-spaltiges Layout mit Navigation,
Fülltext: lorem ipsum



Viewport

Der Viewport bezeichnet den Anzeigebereich für eine Website.

Meist ist damit der sichtbare Bereich innerhalb des Browserfensters gemeint.

Alles außerhalb des Viewports ist erst durch Scrollen der Website sichtbar.

Die untere Kante des Viewports wird als Falz (engl. Fold) bezeichnet.

Viewport als Meta-Tag in HTML

```
<meta name=viewport content="width=device-width, initial-scale=1">
```

Viewport in der CSS-Datei

```
@viewport {  
    width: device-width;  
}
```

Breite des Bildschirms des jeweiligen Gerätes

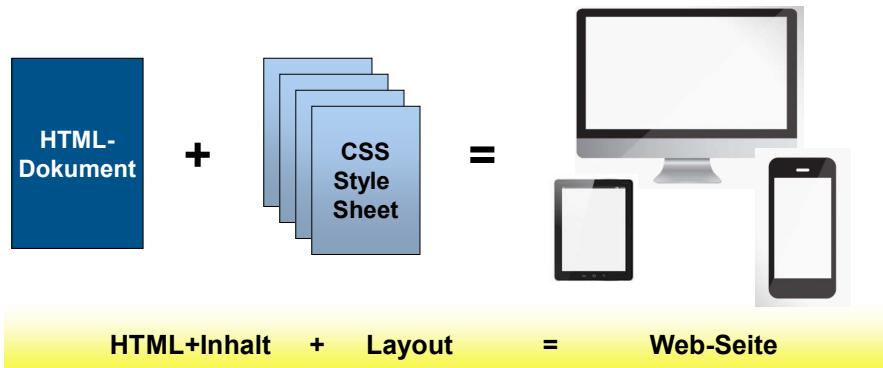
Verhältnis 1:1



95

95

Responsive Webdesign



Verlinkung/Verweis

```
<link rel="stylesheet" type="text/css" href="formate.css">
```



96

96

Media Query

Die Media Queries sind ein wichtiger Bestandteil beim Responsive Webdesign. Sie sind dafür zuständig, die spezifischen Eigenschaften eines Ausgabemediums / Displays abzufragen. So wird jedem Gerät das genau passende Stylesheet zugeordnet und die Webseite wird damit immer optimal angezeigt.



```
h1 {  
    font-size: 24px;  
    line-height: 1.25;  
}
```

CSS-Codeblocks über eine spezielle '@media'-Regel

```
/* Nicht-druckbare Elemente verbergen */  
@media print {  
    video, audio {  
        display: none;  
    }  
}
```

```
@media screen and (min-width: 1024px) {  
    h1 {  
        font-size: 36px;  
    }  
}
```



97

97

Media Query

Parameter	Beschreibung
all	Jegliche Ausgabemedien. Dies ist der Standardwert, wenn kein anderer Parameter gesetzt wird.
aural	Ausgabemedien, die sich künstlicher Sprachausgabe bedienen.
braille	Geräte, auf denen die Braille-Schrift (Blindsight) tastbar wiedergegeben wird.
embossed	Drucker, die die Braille-Schrift tastbar auf Papier einprägen.
handheld	Geräte, die in der Hand gehalten werden. Sie besitzen üblicherweise einen kleinen Bildschirm.
print	Drucker, Seiten mit festen Dimensionen
projection	Beamer und andere Projektoren, die Inhalte projizieren.
screen	Displays / Computer-Bildschirme (scrollend)
speech	Dieser Parameter ist für Sprachsynthesizer / Screenreader gedacht.
tty	Der Parameter tty ist für Geräte mit feststehenden Zeichentypen.
tv	Geräte, die Informationen nur beschränkt wiedergeben können, wie zum Beispiel TV-Geräte.



98

98

Media Query – Flexbox Layout

```
.container {
    display: flex;
}
/* Querformat */
@media screen and (orientation: landscape) {
    .container {
        flex-direction: row;
    }
}
/* Hochformat */
@media screen and (orientation: portrait) {
    .container {
        flex-direction: column;
    }
}
```



99

99

Toggle-Menü ohne Java Script

- Menü, das sich auch bei kleineren Auflösungen für Handy und Tablet automatisch ändert
- Ein **Toggle-Menü** blendet eine Navigation ein- oder aus, um dem Inhaltsbereich auf schmalen Viewports so viel Platz wie möglich zu geben
- Umwandlung in ein Pulldown-Menü
[<a> → <select><option>]



100

100

Aspekte Design

- Was? Ziele für die Seite
 - Wen? Zielgruppe
 - Wie? Informationen/Funktionen, um Personen der Zielgruppe(n) zu überzeugen
 - Womit? Medienformate (Text/Bild)
- bestimmen
bestimmen



101

101

Aspekte Design

- Wer ist die Zielgruppe
 - (Personengruppe, Alter, Bildung, Berufsgruppe) meiner Website?
Danach richten sich: Inhalte inkl. der Bilder, Layout, Formate (z. B. Schriftgröße)
- Fotos vorbereiten
- Stammordner ev. Unterordner anlegen
- Layout planen
- Stylesheets planen
- Namenskonventionen festlegen
- Texte und Inhalte (Bilder etc.) in die Seiten einfügen
- Verlinkungen umsetzen und testen

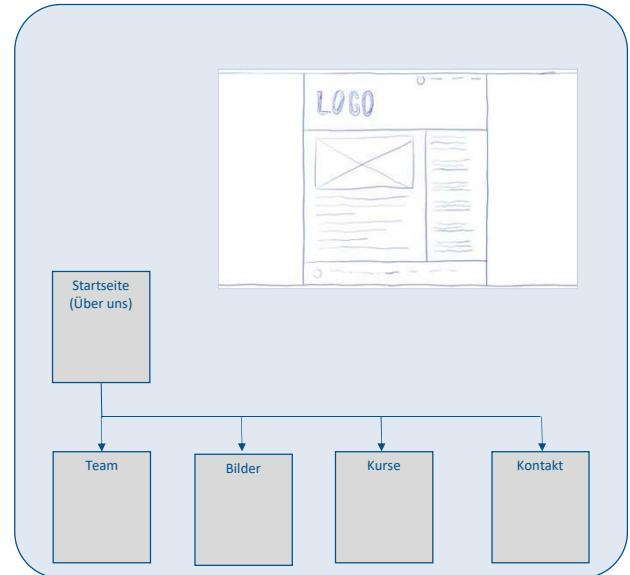


102

102

Regeln für gutes Webdesign

1. Einfachheit
2. Visuelle Hierarchie
3. Einfache Navigation
4. Einheitlichkeit
5. Zugänglichkeit
6. Konventionen
7. Glaubwürdigkeit
8. Der Nutzer im Mittelpunkt



103

103

Gestaltpsychologie

Gesetz

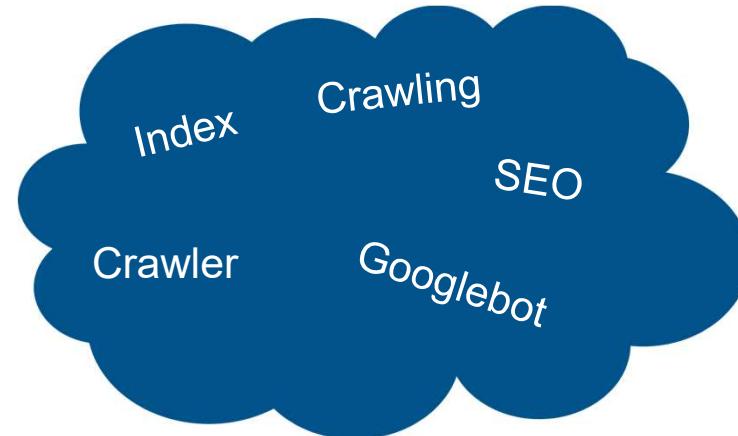
- der Nähe
- der Gleichheit
- der Kontinuität für Webdesign
- der Geschlossenheit
- der Erfahrung
- des gemeinsamen Schicksals
- der Symmetrie
- der guten Gestalt
- [Gestaltpsychologie angewendet bringt gutes Webdesign \(html-seminar.de\)](#)



104

104

SEO – Search Engine Optimization Suchmaschinenoptimierung



105

105

Title, Description, Keywords

- Mit meta-Tags besseres Ranking/Auffinden ermöglichen
 - title
 - Kurzbeschreibung, ist Rankingfaktor bei Suchmaschinen
 - animiert möglicherweise zum Anklicken
 - 1. anklickbare Zeile
 - bookmarken möglich
 - description
 - Beschreibung 160-180 Zeichen, die unter Titel und URL erscheint
 - keywords
 - damit wird die Website von den richtigen Nutzern gefunden

```
<head>
<meta name="description" content="Free Web tutorials">
<meta name="keywords" content="HTML,CSS,XML,JavaScript">
</head>
```



106

106

Robots (Webcrawler) unterstützen

```
<meta name="robots" content="index"/>
```

```
<meta name="robots" content="noindex"/>
```

```
<meta name="robots" content="follow"/>
```

```
<meta name="robots" content="nofollow"/>
```

Anweisung an die Suchmaschine,

- diese Seite zu indizieren
- wie mit verlinkten Seiten verfahren wird

Kombination

```
<meta name="robots" content="index,nofollow" />
```



107

107

Sitemap

Bei der HTML-Sitemap handelt es sich um eine HTML-Seite, auf der alle Unterseiten einer Website aufgelistet werden. Sie wird meist im Footer der Website verlinkt und ist somit für alle Besucher des Webauftritts sichtbar.

- Hohe Bedeutung bzgl. Usability einer Webseite
- Vorteilhaft im Hinblick auf Suchmaschinenoptimierung, da sie den Crawlern das Auffinden der Unterseite erleichtern (GoogleBot)

Beispiel Amazon:

The screenshot shows the footer of the Amazon.de website. It features several service links arranged in two rows. The top row includes: Amazon Advertising (Kunden finden, gewinnen und binden), Amazon Music (Streamen Sie Millionen von Songs), AbeBooks (Bücher, Kunst & Sammelobjekte), Amazon Web Services (Cloud Computing Dienste von Amazon), and Audible (Hörbücher herunterladen). The bottom row includes: Book Depository (Bücher mit kostenfreier Lieferung weltweit), IMDb (Filme, TV & Stars), Kindle Direct Publishing (Ihr E-Book veröffentlichen), Prime Now (1-Stunden-Lieferung Tausender Produkte), AbeBooks (Reduzierte B-Ware), ZVAB (Zentrales Verzeichnis Antiquarischer Bücher und mehr), Amazon Business (Kauf auf Rechnung, PD-Nummern, Für Unternehmen), Amazon Second Chance (Geben Sie es weiter, tauschen Sie es ein, geben Sie Ihnen ein zweites Leben), Amazon Warehouse (Reduzierte B-Ware), and Shopbop (Designer Modemärkte).

[Amazon.de: Günstige Preise für Elektronik & Foto, Filme, Musik, Bücher, Games, Spielzeug & mehr](#)



108

108

Usability

Marke und Design

1. Ist Nutzern innerhalb von **5 Sekunden** klar, worum es auf der Website geht?
2. Ist die Website im Vergleich zur Konkurrenz optisch ansprechend und **zeitgemäß**?
3. Steht die Gestaltung der Website im Einklang mit Ihrem **Corporate Design**?

User Experience (UX)

4. Wird die Startseite innerhalb **weniger Sekunden** geladen?
5. Ist die **Navigation** intuitiv und verständlich?
6. Sind **Links** so gestaltet, dass Nutzern unmittelbar klar ist, wohin sie führen?



109

109

Usability

Content & SEO

7. Ist die Website unter den wichtigsten **Keywords** bei Google auffindbar?
8. Sind Texte **fehlerfrei** und übersichtlich strukturiert?
9. Sind Inhalte aktuell und mit **Social Media** verzahnt?
10. Gibt es einladende Kontakt- und **Interaktionsmöglichkeiten**?

Technik & Mobile

11. Ist Ihre Website auf allen **Endgeräten** (Desktop, Tablet, Smartphone) gut zu benutzen?
12. Können **Redakteure** Inhalte komfortabel pflegen und verwalten?
13. Sind Tools wie eine **Volltextsuche** vorhanden und funktionieren sie einwandfrei?



110

110

http-Fehlercodes

- **HTTP Statuscode 200** - alles ok
- **HTTP Statuscode 304** - es wurde nichts geändert

- **HTTP Status 404** Seite nicht gefunden
- **HTTP Status 500** Interner Serverfehler
- **HTTP Status 401** Authentifizierung fehlgeschlagen

- → eigene Fehlerseite erstellen
ErrorDocument 404 / fehlermeldung-404.html



111

111

Passwortschutz über .htpasswd/.htaccess

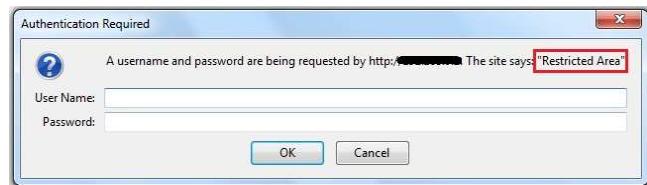
- **.htpasswd** (einfache Textdatei, Punkt nicht vergessen) wird bei Apache HTTP-Servern dazu genutzt, um Usernamen und Passwörter zu speichern, die später für einen Verzeichnisschutz als Logindaten genutzt werden
- Liste aller möglichen Zugangsdaten
- **htpasswd** Passwörter können per md5(), crypt() oder sha1() hinterlegt werden
- Die .htaccess Datei übernimmt die eigentliche Aufgabe, das Verzeichnis und die Unterverzeichnisse mit einem Passwortschutz versehen.
Es spielt daher nur eine Rolle, wo die .htaccess Datei abgelegt wird.

```

AuthType Basic
AuthName "Passwortgeschützter Bereich"
AuthUserFile /pfad/zur/datei/.htpasswd
Require valid-user

```

.htaccess Datei



112

112