2018

HTML und CSS

Karl Steinam

Klara-Oppenheimer-Schule

28.11.2018

Inhalt

sf

Historie von HTML und css

(Quell: Selfhtml, https://wiki.selfhtml.org/wiki/HTML/Entstehung_und_Entwicklung)

https://wiki.selfhtml.org/wiki/Grundlagen/Einstieg/Entstehung_des_Internet#1993_-_Das_World_Wide_Web_.28WWW.29

HTML (HyperText Markup Language) wurde 1990 vom Web-Gründer Tim Berners-Lee als Auszeichnungssprache (Markup Language) entwickelt, die auf SGML basierte.

Eine solche Auszeichnungssprache hat die Aufgabe, die logischen Bestandteile eines textorientierten Dokuments zu beschreiben. Als Auszeichnungssprache bietet HTML daher die Möglichkeit an, typische Elemente eines textorientierten Dokuments, wie Überschriften, Textabsätze, Listen, Tabellen oder Grafikreferenzen, als solche auszuzeichnen.

Das Auszeichnungsschema von HTML geht von einer hierarchischen Gliederung aus. HTML zeichnet Inhalte von Dokumenten aus. Dokumente haben globale Eigenschaften wie zum Beispiel Kopfdaten. Der eigentliche Inhalt besteht aus Elementen, zum Beispiel einer Überschrift 1. Ordnung, Textabsätzen, Tabellen und Grafiken. Einige dieser Elemente haben wiederum Unterelemente. So enthält ein Textabsatz zum Beispiel eine als betont markierte Textstelle, eine Aufzählungsliste besteht aus einzelnen Listenpunkten, und eine Tabelle gliedert sich in einzelne Tabellenzellen.

In der ursprünglichen Version von HTML gab es 18 Elemente, von denen TITLE und die Textstrukturierungselemente (H1-H6, P, A, ADDRESS, UL,OL und LI, sowie DL,DT und DD) immer noch verwendet werden, während Elemente wie PLAINTEXT, ISINDEX und LISTING heute nicht mehr gültig, sondern als obsolet angesehen werden. [1] [2]

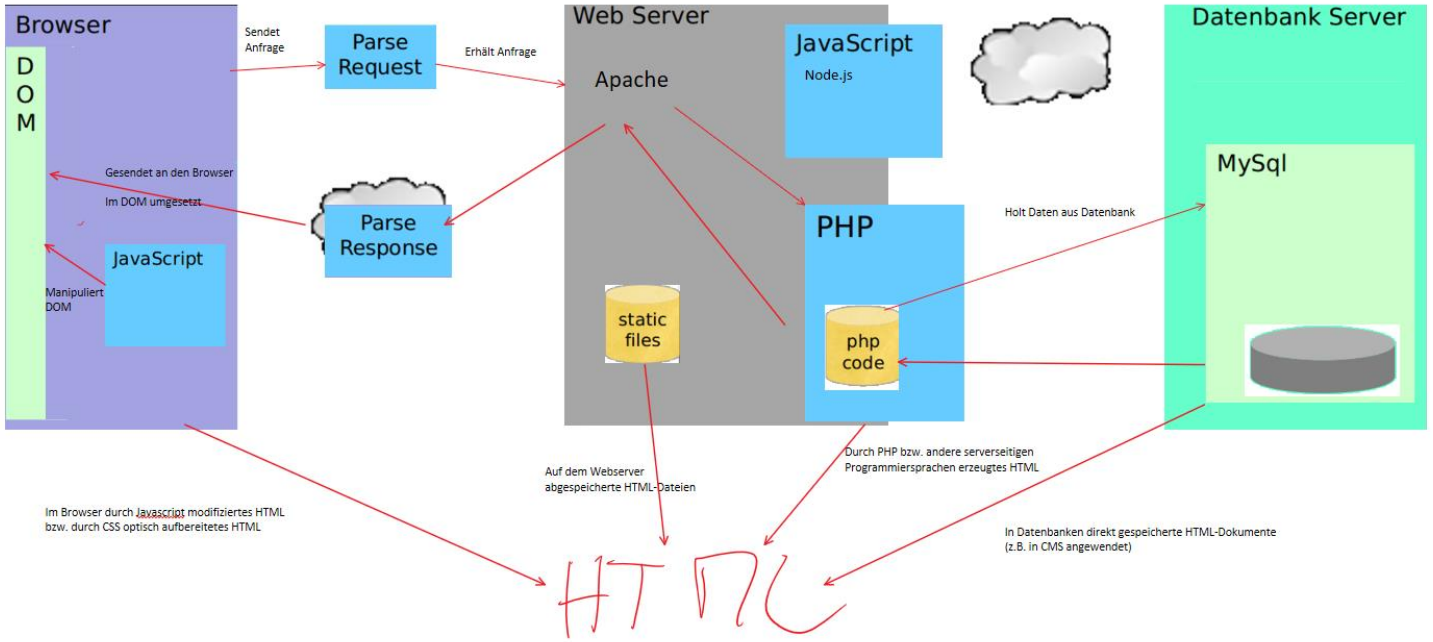
Eine der wichtigsten Eigenschaften von HTML ist die Möglichkeit, Verweise zu definieren. Verweise („Hyperlinks“) können zu anderen Stellen im eigenen Projekt führen, aber auch zu beliebigen anderen Adressen im World Wide Web und sogar zu Internet-Adressen, die nicht Teil des Webs sind.[3]. Durch diese einfache Grundeigenschaft eröffnet HTML völlig neue Welten. Das Bewegen zwischen räumlich weit entfernten Rechnern wird bei modernen grafischen Web-Browsern auf einen Mausklick reduziert. In Ihren eigenen HTML-Dateien können Sie Verweise notieren und dadurch inhaltliche Verknüpfungen zwischen Ihren eigenen Inhalten und denen anderer Anbieter herstellen. Auf dieser Grundidee beruht letztlich das gesamte World Wide Web, und dieser Grundidee verdankt es seinen Namen

Wichtigste Erkenntnis:
HTML soll Dokumente strukturieren, nicht formatieren! Dafür setzte sich mit den Cascading Style Sheets (CSS) gegenüber angedachten Alternativen eine Erweiterung durch, die die Präsentation festlegen konnte

- 1990 von Tim Berners-Lee als Auszeichnungssprache (Markup Language) entwickelt, die auf SGML basierte.
- Beschreibt die logischen Elemente eines Textdokumentes, z.B. Überschriften, Textabsätze, Listen, Tabellen, Bilder
- Dokument ist hierarchisch gegliedert
- Besteht aus Kopfdaten (Metadaten) und den eigentlichen Dokumenten

Zusammenhang HTML / Webserver

Die Kommunikation zwischen Browser und serverseitigen Komponenten kann wie folgt aussehen.




```
<html>
<body>
        <h1>Hello</h1>

        <p> Steinam ist toll</p>

        <h2>Gibt es noch andere</h2>

        Nein

</body>
</html>
```

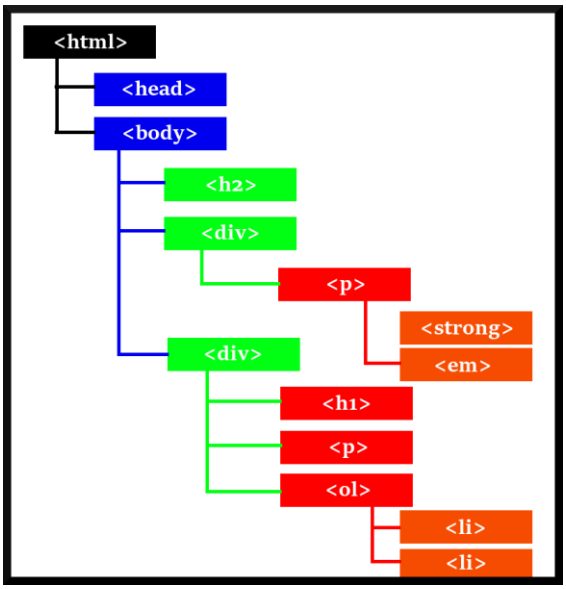
Eigenschaften

Die Hypertext Markup Language (englisch für Hypertext-Auszeichnungssprache), abgekürzt HTML, ist eine textbasierte Auszeichnungssprache zur Strukturierung digitaler Dokumente wie Texte mit Hyperlinks, Bildern und anderen Inhalten. HTML-Dokumente sind die Grundlage des World Wide Web und werden von Webbrowsern dargestellt. Neben den vom Browser angezeigten Inhalten können HTML-Dateien zusätzliche Angaben in Form von Metainformationen enthalten, z. B. über die im Text verwendeten Sprachen, den Autor oder den zusammengefassten Inhalt des Textes.

HTML5 ist die fünfte Fassung der Hypertext Markup Language (engl. für Hypertext-Auszeichnungssprache), einer Computersprache zur Auszeichnung und Vernetzung von Texten und anderen Inhalten elektronischer Dokumente, vorwiegend im World Wide Web.

HTML dient als Auszeichnungssprache dazu, einen Text semantisch zu strukturieren, nicht aber zu formatieren. Die visuelle Darstellung ist nicht Teil der HTML-Spezifikationen und wird durch den Webbrowser und Gestaltungsvorlagen wie CSS bestimmt. Ausnahme sind die als veraltet (englisch deprecated) markierten präsentationsbezogenen Elemente.

Das World Wide Web Consortium (W3C) hat am 28. Oktober 2014 die fertige HTML5-Spezifikation („W3C Recommendation“) vorgelegt. HTML5 wird damit als Nachfolger von HTML4 die Kernsprache („core language“) des Webs. Sie ersetzt die Standards HTML 4.01, XHTML 1.0 und DOM HTML Level 2. Sie bietet neue Funktionen wie Video, Audio, lokalen Speicher und dynamische 2D- und 3D-Grafiken, die von HTML4 nicht direkt unterstützt wurden und sich ohne HTML5 nur mit zusätzlichen Plugins (z. B. Adobe Flash) umsetzen ließen. Zukunftsweisend sind weiterhin neue Elemente, die eine verbesserte semantische Struktur ermöglichen.

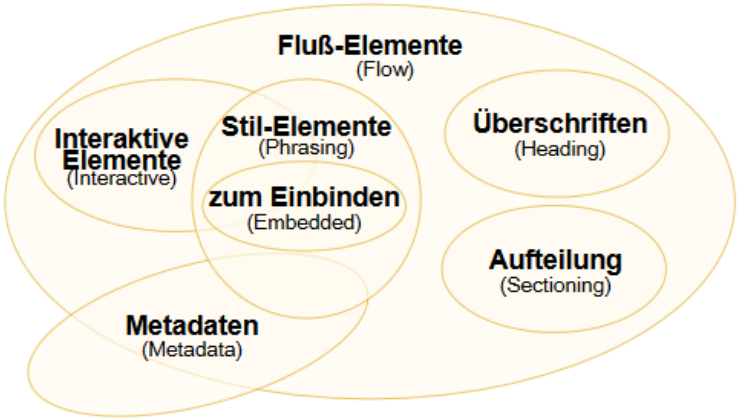


Elemente

HTML-Elemente können hinsichtlich des Inhalts, den sie enthalten dürfen, kategorisiert werden.

Seit HTML5 existieren hierzu sieben grundsätzliche Kategorien:

- Metadaten: Beeinflussen das gesamte Dokument, beispielsweise das Einbinden externer CSS-Dateien
- Fluss-Elemente: Enthalten in der Regel Text oder Aussagen-Elemente
- Überschriften: Beschreiben einen Abschnitt
- Elemente zur Aufteilung: Teieln ein Dolkument in verschiedene logische Bereiche (article, aside, nav, section)
- Stil-Elemente: differenzieren Formulierung und Darstellungsebene ihres Inhalts, s.B. sup, sub, em, b
- Elemente zur Einbindung von Ressourcen und Interaktive Elemente, z.B. audio, canvas, iframe, img, svg und video



HTML-Tags

Das grundlegende Element von HTML ist das sog. **TAG**. Es definiert eine semantischr Aussage ohne die optische Presentation festzulegen. Alle Browser haben allerdings eine Statndardinterpretation für jeden Tag.

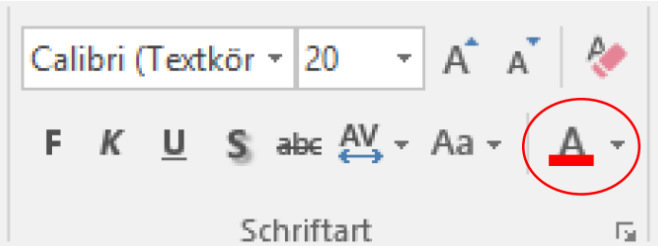
Die Struktur eines Tags ist wie folgt:

- In HTML gibt es fest definierte Tags, die alle eine bestimmte Aufgabe haben, z.B. Inhalt für fettgedruckten Text.
- Tags sind gekennzeichnet durch die <…>-Zeichen.

- Es gibt immer (zumindest fast) ein öffnendes und ein schließendes Tag, wobei das schließende Tag noch einen Schrägstrich vor dem Namen des Tags hat.
- Tags können geschachtelt seien, wichtig ist jedoch die korrekte Schachtelung
    - Tags müssen in der umgekehrten Reihenfolge geschlossen werden
    - Das Tag, welches als letztes geöffnet wurde, muss erstes wieder geschlossen werden
    - Fehler: <u>Inhalt</u>
- Es gibt einige Tags, die kein schließendes Tag besitzen (z.B.
 oder <hr>)
  Das sind sogenannte selbstschließende Tags (engl. self-closing tags)
  Selbstschließende Tags sind Inhaltsleer (was nicht bedeutet, dass sie keine Informationen besitzen)
-
 ist ein erzwungener Zeilenumbruch (markiert nur eine Stelle)
- <hr> ist eine horizontale Linie (Linie hat keinen Inhalt)

HTML-Attribute

Attribute können die Tags eines HTML-Elementes erweitern. Unterschiedliche Tags können verschiedene Attribute haben. Das Konzept von Attributen ist auch in anderen Textverarbeitungen / Programmen vorhanden



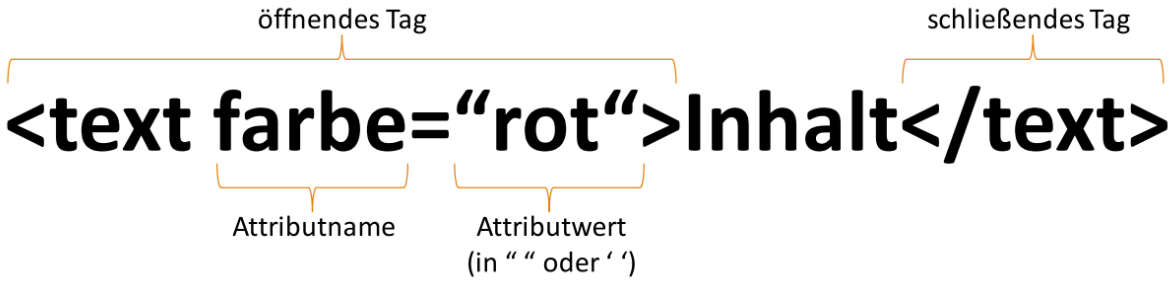
Im konkreten Fall einer Farbe wäre es schwierig, für jedes Element verschiedene Versionen des Tags vorzuhalten.

- <textMitFarbe>Inhalt</textMitFarbe> Problem: Keine Information welche Farbe!
- <textRot>Inhalt</textRot> Problem: Für jede Farbe muss ein eigenes Tag definiert werden!

Die Lösung sind Attribute:

- <text farbe="rot">Inhalt</text>

Attribute haben folgenden Aufbau:



Das HTML-Grundgerüst

Eine minimale HTML-Datei hat folgenden Aufbau

```
<!DOCTYPE html>
<html lang="de">
<head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Titel</title>
</head>
<body>
</body>
</html>
```

Übungsaufgabe:

A1.

Erzeugen Sie eine HTML-Datei, die folgende Struktur abbildet. Ignorieren Sie die unterschiedlichen Farben.

Überschriften

Allgemeines

Überschriften richtig einsetzen

Hierarchien

Untertitel

Zitate und Hervorhebungen

Überschriften mit CSS formatieren

Quellen

siehe auch

Referenzen

Weblinks

Erweiterung:

Welche Änderungen müssen Sie vornehmen, um die Überschriften in farbiger Schrift erscheinen zu lassen.

A2:

Erzeugen sie eine HTML-Datei mit folgendem Aussehen

Hilfe gibt es unter https://wiki.selfhtml.org/wiki/HTML/Textauszeichnung

Fettgedrucker Text

Fettgedrucker Text
Fettgedrucker Text

Kursivgedruckter Text

Kursivgedruckter Text
Kursivgedruckter Text

Unterstrichener Text

Unterstrichener Text

span-Tag

Aufgabe 3

Folgende HTML-Formular zeigt beispielhaft die Manipulation des DOM mit Javascript.

http://bioub.github.io/d3.DOMVisualizer/

Aufgabe 4:

Erstellen Sie ein DOM (Document Object Model) des folgenden HTML-Dokumentes

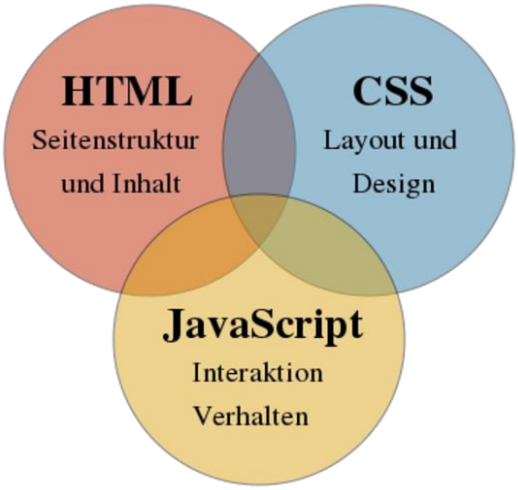
```
<!doctype html>
<html>
  <head>
    <meta charset="utf-8"/>
    <title>Steckbrief von Charly</title>
    <link rel="stylesheet" type="text/css" href="charly.css">
  </head>
  <body>
    <h1>Mein Hund</h1>
    <p><img alt="Foto von Charly" src="charly.jpg"/></p>
    <p>
      Ich heiße <em>Charly</em> und wohne mit meinem Frauchen in der Nähe von
Würzburg
    </p>
    <p>
      Ich mag ganz gerne <em>Agility-Sport</em>.
      Meine Lieblingsdisziplinen sind:
    </p>
    <ul>
      <li><em>Sprung</em> durch einen Reifen</li>
      <li><em>Lauf</em> über eine Wippe</li>
      <li><em>Slalomlauf</em> zwischen Stangen</li>
    </ul>
    <p>
      Ich gehe jedes Wochenende zum
      <a href="https://www.hundesportverein-giebelstadt.de/">Hundesportverein
Giebelstadt</a>
      .
    </p>
  </body>
</html>
```

Lösung:

em em em

Die Grundlagen von CSS

Im modernen Webdesign kommt den Webtechniken HTML, CSS und JavaScript jeweils eine bestimmte Rolle zu.



- HTML legt fest, was auf der Seite stehen soll (struktureller Aufbau einer Webseite)
- CSS legt fest, wie es dargestellt werden soll (Formatierung & Gestaltung)
- JavaScript legt fest, was passieren soll. (interaktive Elemente)

CSS (Cascading Style Sheets, zu deutsch „Mehrstufige Formatvorlagen“) ist eine Formatierungssprache für HTML-, SVG- und XML-Dokumente. Der „große Sinn von CSS“ besteht in der Trennung von Inhalt und Design. Das hört man oft, stellt sich nichts drunter vor und bastelt dann doch eine besondere Überschrift mit einem inline-style, einen neuen div-container mit einer ganz speziellen id oder Klasse, die später nirgends wieder auftaucht. Das funktioniert auch mehr oder weniger - man wird ohne viel Mühe eine fixe Seite zusammenstellen, die ihren Zweck erfüllt - und vergisst das ganze.

Es ist aber beinahe unmöglich, einen so über die Zeit gewucherten Internetauftritt umzugestalten - man müsste dutzende oder mehr Einzeldateien umschreiben und im Gewusel der Klassen und Elemente wird man sich schnell verlieren. Letztlich dauert die Änderung beinahe länger als die Neuerstellung. Und genau darin liegt die Stärke von CSS: uneingeschränkte Flexibilität, wenn z.B. das Layout nicht mehr zeitgemäß ist oder wenn neue Strukturen, vor allem bei dynamischen Seiten, Änderungen erfordern. Dabei wäre man ohne CSS buchstäblich „verloren im Quelltext“.

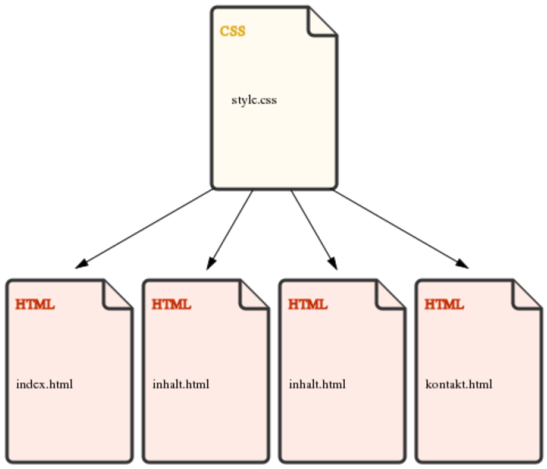
Das erste Gebot von CSS lautet: „am Anfang ohne.“

Soll heißen - man gestaltet die Seite erst komplett ohne CSS, nur die Seitenstrukturierung in Kapitel und Absätze, bei diesen dann die Textauszeichnung von Überschriften und Ähnlichem. Es stellt auch die „allgemeine Gültigkeit“ des Layouts sicher - ist die Grundlage an sich stimmig, kann man die Seite beliebig umgestalten, jeder Fehler schränkt dies wieder ein.

Eine solche Webseite sieht nicht gut aus, ist aber uneingeschränkt nutzbar. Die einzelnen Elemente werden vom Browser einfach untereinander auf dem Bildschirm angeordnet, und zwar in der Reihenfolge, in der sie in der HTML-Datei aufgeschrieben sind. Die Elemente selbst werden gemäß den Voreinstellungen des Browsers dargestellt; so ist z.B. festgelegt, dass der Hintergrund des Dokuments weiß oder grau darzustellen ist, dass Überschriften in fett und einer Größe von soundsoviel Punkten darzustellen sind.

An diesem Punkt setzen die Cascading Stylesheets ein. Es handelt sich dabei um eine unmittelbare Ergänzungssprache, die vorwiegend für HTML (aber auch für SVG) entwickelt wurde. Sie klinkt sich nahtlos in HTML ein und erlaubt das beliebige Formatieren einzelner HTML-Elemente. Mit Hilfe von Stylesheets können Sie beispielsweise festlegen, dass alle Überschriften 24 Punkt groß sind und mit einem Nachabstand von 16 Punkt und mit einer grünen doppelten Rahmenlinie oberhalb dargestellt werden. Schematisch würde dies etwa so aussehen:

CSS erlaubt es, Stile, Farben und Formen zu definieren, beispielsweise für alle Überschriften, oder für alle Textabsätze mit einem bestimmten Klassennamen, oder für kursiv ausgezeichneten Text, der innerhalb einer Tabellenzelle vorkommt.



Die zentralen Formate können sich auf eine HTML-Datei beziehen, aber auch in eine externe Style-Datei ausgelagert werden, die man in beliebig viele Seiten einbinden kann. So werden einheitliche Formatvorgaben möglich, und der HTML-Code wird von unnötigem Ballast befreit. Spätere Änderungen am Design können so leicht durchgeführt werden.

In der folgenden Abbildung dient eine einzige CSS-Datei beispielsweise vier HTML-Dateien als Formatvorlage:

Aufbau von CSS-Regeln

Eine CSS-Regel besteht aus

- der Bezeichnung für das Element, auf das die Regel zielt (**CSS-Selektor**),
- aus Eigenschaften (Properties), die dem Element zugewiesen werden.
- CSS-Eigenschaften stehen in geschweiften Klammern und sind durch Semikolons voneinander getrennt.
- Klassen- und ID-Namen sind case-sensitiv


```
selector { Eigenschaft: Wert;
                  Eigenschaft: Wert;
                  Eigenschaft: Wert
}

h1 {
            font-family: Helvetica;
            font-size: 1.4em;
            color: red
}
```

Einbinden von CSS

Styleinformationen können auf verschiedene Arten in ein HTML-Dokument eingebunden werden

INLINE

Mit dem Universalattribut **style** können Sie Eigenschaften einem Element direkt zuweisen. Dabei sind nur Deklarationen, also Eigenschaft-Wert-Zuweisungen möglich.

Beispiel:

```
<p style="text-align: center; color: green;">
   Dieser Absatz wird über ein style-Attribut formatiert.
</p>
```

Bewertung:

- Direktes Festlegen von Formaten pro Element
- Verlust vieler Vorteile
- Hoher Wartungsaufwand
- verringerte Flexibilität
- keine zentrale Bearbeitung

INTERNAL

Das HTML-Element **style** legt Formate zentral im Head des HTML-Dokuments fest.

```
<html lang="de">
<head>
    <style>
        h1 {
            background-color: green;
            color: blue;
         }

        p {
            text-align: center;
            color: red;
          }
    </style>
</head>
<body>


<h1>Überschrift</h1>
<p>Ein text halt</p>
</body>
</html>
```

EXTERNAL

Das CSS wird einem eigenen externen Stylesheet mit der Endung .css abgespeichert und mit dem HTML-Element link direkt im Head eingebunden.

```
<!--Einbinden eines externen Stylesheets -->

<!doctype html>
<html lang="de">
<head>
   <link rel="stylesheet" href="stylesheet.css">
```

CSS und Selektoren

Damit Formateigenschaften auf ein Element angewendet werden können, muss definiert werden, welche Elemente angesprochen werden. Dies geschieht über Selektoren. Als Selektoren bezeichnet man die Teile einer CSS-Regel, die vor dem Abschnitt aus geschweiften Klammern stehen. Voneinander unabhängige Selektoren, denen dieselben Eigenschaften zugewiesen werden, können mit Kommata getrennt werden. Es gibt verschiedene Arten von Selektoren.

TypSelektoren

Der Element- bzw. Typselektor besteht aus dem Namen des Elements, das angesprochen werden soll. Mit diesem Selektor werden alle Elemente eines Typs angesprochen.

```
<!doctype html>
    <html>
        <head>
            <meta charset="utf-8">
            <title>CSS-Beispiel: Typselektor</title>
            <style> p {border: 3px solid green;} </style>
        </head>
        <body>
            <p>Dieses Beispiel demonstriert die Wirkung des Typselektors auf
<abbr>HTML</abbr>-Dokumente.</p>
        </body>
    </html>
```

Typselektoren geben manche Eigenschaften(z.B. Schriftgestaltung) an ihre Kinder weiter und manche nicht (z.B. background-color, border). Dies kann erwünscht bzw. unerwünscht sein. Man kann dies innerhalb von CSS durch den Attributwert **inherit** verändern.

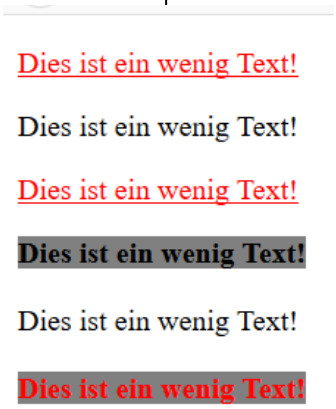
Klassenselektor

- spricht Elemente an, die einer bestimmten Klasse zugehörend sind
- mehrere HTML-Elemente können die gleichen Klasse zugeordnet bekommen
- Klassenselektoren können mit anderen Selektoren verbunden werden
- Klassenselektor kann auch mit dem Typselektor verbunden sein
- **Ein Klassenselektor wird gebildet, wenn vor dem Klassennamen ein Punkt notiert wird**

Übung:

Erzeugen Sie aus dem folgenden HTML das nebenstehende Aussehen mit Hilfe von Klassenselektoren

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

<head>
<meta charset="utf-8">
<title>Klassenselektoren</title>
<link rel="stylesheet" type="text/css" href="klassenselektoren.css">
</head>

<body>
<p>Dies ist ein wenig Text!</p>
<p>Dies ist ein wenig Text!</p>
<p>Dies ist ein wenig Text!</p>
<span>Dies ist ein wenig Text!</span><br><br>
<span>Dies ist ein wenig Text!</span><br><br>
<span>Dies ist ein wenig Text!</span>
</body>
</html>
```



ID-Selektor

- Spricht ein Element an, dem eine ID zugeordnet wurde
- Gebildet durch Voranstellen des Gatterzeichens vor den ID-Namen
- ID-Selektoren können mit anderen Selektoren verbunden werden
    - Mit Elementselektoren: elementname#id
    - Mit Klassenselektoren: .klassenname#id bzw. #id.klassenname

Beispiel:

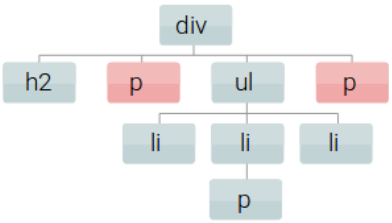
```
<!doctype html>
<html>
  <head>
    <meta charset="utf-8">
    <title>CSS-Beispiel: IDSelektor</title>
    <style>

      your code here.....

    </style>
  </head>
  <body>
    <div id="beispiel">
      <h1>ID-Selektoren</h1>
      <p>Dieses Beispiel demonstriert die
Wirkung des ID-Selektors.</p>
      </div>
  </body>
</html>
```

Kontextselektor

Ein Nachfahren-Selektor oder Nachbar-Selektor wählt Elemente aus ihrer Position innerhalb von anderen Elementen aus – aus dem **Kontext** der HTML-Struktur. Sie sind also abhängig von der Struktur, in der sie auftreten: von ihren Nachbarn, Vorfahren und Nachfahren.

- Ein Nachfahren-Selektor (**Descendant Selector)** filtert Elemente unterhalb bestimmter Elemente.
- Ein Nachbar-Selektor (**Adjacent Selector)** filtert aufeindnerfolgende Elemente der gleichen Ebene,

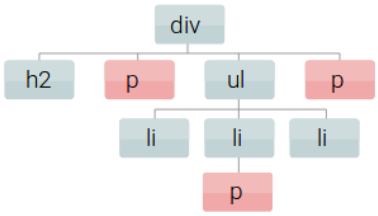
Kindselektor



- ">" zwischen den beiden Selektoren
- Trifft nur auf Elemente zu, die direkt innerhalb des div-Elementes liegen

d

Nachfahrenselektor

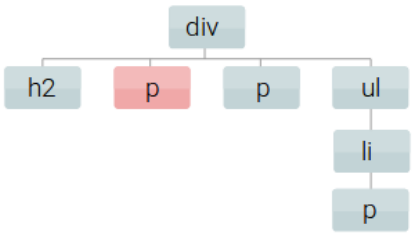


- Leerzeichen " " zwischen den Selektoren

Alle p-Elemente, die innerhalb von div-Elementen liegen, werden durch die Regel angesprochen

d

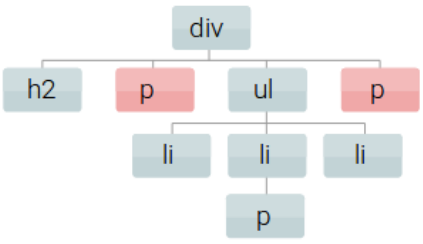
Direkte Nachbar-Selektoren (Adjacent)



- Absteigender Selektor, angewendet auf ein p-Element, das dem h2-Element direkt folgt.
- h2 und p haben dabei immer dieselben Eltern

h

Indirekte Nachbar-Selektoren



- Angewendet auf alle p-Elemente, die dem h2-Element folgen.
- h2 und p haben dabei immer dieselben Eltern – liegen also in derselben Ebene.

h

Aufgaben zu Selektoren

1. Folgende HTML-Datei ist vorhanden:

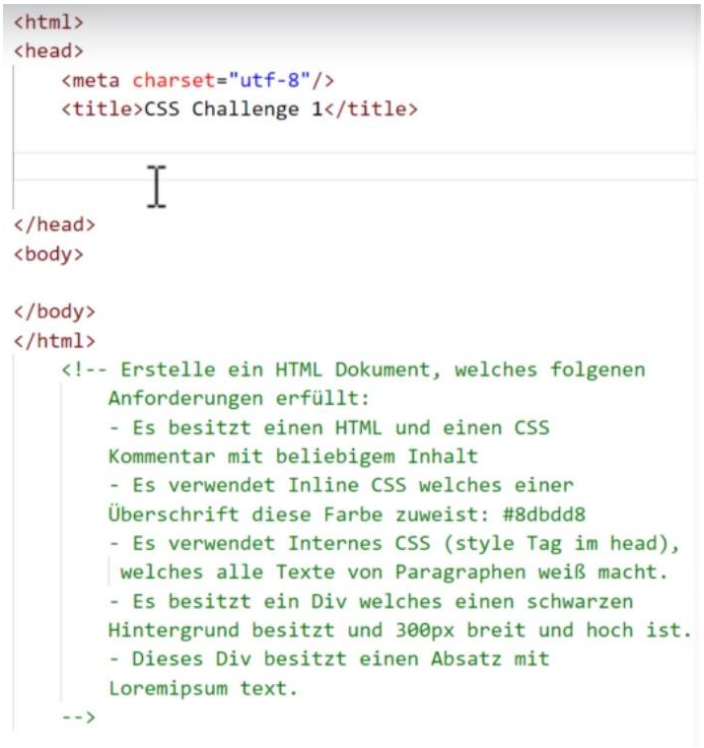
```
<!DOCTYPE html>
<html lang="de">

  <head>
    <meta charset="utf-8">
    <title>Typselektoren</title>
    <link rel="stylesheet" type="text/css" href="typselektoren.css">
  </head>

  <body>
    <h1>Leggings kickstarter</h1>
    <p>
      Coloring book vaporware heirloom cloud bread man braid, ramps organic
umami distillery.
      <span>
        Kale chips retro literally locavore migas raw denim.
        <b>Live-edge austin taiyaki four dollar toast, cray shoreditch.</b>
        DIY meh wayfarers retro banh mi selfies food truck.
      </span>
      Craft beer chicharrones gluten-free, intelligentsia humblebrag.
    </p>
  </body>

</html>
```

Erstellen Sie untenstehende HTML-Datei mit Hilfe von Typselektoren
Arbeiten Sie die Aufzählungspunkte hintereinander ab.

- Die Farbe des body-Tags soll einheitlich rot sein

- Die Farbe des span-tags soll grau sein
- Die Farbe des h1-Tags soll ausnahmsweise blau sein
- Der body-Tag soll einen roten Rahmen erhalten
  Warum erhalten die Kindelemente des body-Tags keinen Rahmen ?
- Geben Sie dem h1-Tag ebenfalls einen Rahmen, indem Sie von body den Rahmen erben.

Aufgabe 2:



```
<html>
<head>
    <meta charset="utf-8"/>
    <title>CSS Challenge 1</title>


</head>
<body>

</body>
</html>
    <!-- Erstelle ein HTML Dokument, welches folgenen
        Anforderungen erfüllt:
        - Es besitzt einen HTML und einen CSS
        Kommentar mit beliebigem Inhalt
        - Es verwendet Inline CSS welches einer
        Überschrift diese Farbe zuweist: #8dbdd8
        - Es verwendet Internes CSS (style Tag im head),
         welches alle Texte von Paragraphen weiß macht.
        - Es besitzt ein Div welches einen schwarzen
        Hintergrund besitzt und 300px breit und hoch ist.
        - Dieses Div besitzt einen Absatz mit
        Loremipsum text.
    -->
```

Aufgabe 3

Erstelle eine HTML Datei, welche folgenden Bedingungen erfüllt:

- Grundstruktur einer HTML Datei
- Verwendet einen Tag Selektor (z.B. p Tag) der die Farbe auf Dunkelgrau setzt
- Verwendet einen ID Selektor "viereck" welcher ein div 300px breit und 300px hoch macht und ihm einen #b2d3e6 Hintergrund gibt.
- Verwende einen Class Selektor "monogross" der mindestens von zwei Elementen genutzt wird und die Schriftart zu monospace ändert und die Schriftgröße auf 150% setzt.
- Besitzt einen Link, der den Nutzer auf Google weiterleitet und beim Hovern die Farbe Rot annimmt.

Aufgabe 4:

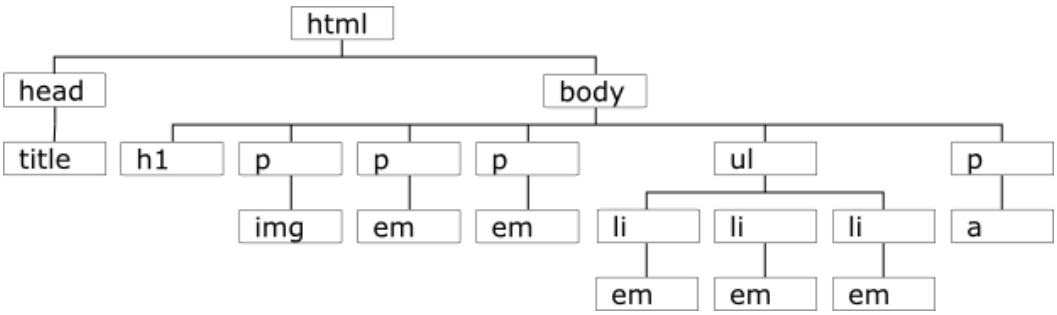
Formatieren Sie die untenstehende HTML-Datei mit Hilfe von CSS.

body

h1

Aufgabe 5:

Gegeben ist folgende DOM-Struktur



Markieren Sie sich die Stellen, die durch den jeweiligen CSS-Selektor ausgewählt werden.

```
h1, em
{
 font-weight: bold;
 font-style: normal;
}
```

```
Em
{
  color: blue;
  font-weight: bold;
}
```

```
ul em
{
 color: red;
}
```

```
Body
{‘
 background-color: orange;
 color: black;
 font-size: small;
 font-family: Georgia, "Trebuchet MS", Verdana, sans-serif;
}
```

Wie wird sich folgende CSS-Anweisung auf den DOM auswirken

```
body
{
  background-color: orange;
  color: black;
  font-size: small;
  font-family: Georgia, "Trebuchet MS", Verdana, sans-serif;
}
em
{
  color: blue;
}
```

Was müssen Sie im HTML-Dokument ändern, damit folgende Regel wirken kann.

```
p.wichtig
{
  background-color: white;
}
```

Aufgabe 5:

Lösen Sie das CSS-Rätsel unter

https://steinam.rigel.uberspace.de/css-diner/