

Hypertext Markup Language

Studiengang: Informatik Bachelor
Name: Hoang Minh, Phan
Matrikelnummer: 1811771

Inhalt

1	Einführung.....	4
2	Was ist HTML.....	4
2.1	Aufbau eines HTML Befehl.....	5
2.2	Aufbau eines HTML Dokument.....	5
2.2.1	Der HTML-Tag <html>	6
2.2.2	Der HEAD-Block <head>	6
2.2.3	Der BODY-Block <body>	6
2.3	Anlegen eines HTML Dokument	6
	Zusammenfassung	6
	Aufgaben zur Selbstüberprüfung.....	7
	7
3	Grundlegende HTML Tag im HEAD Block einer HTML Datei	8
3.1	Der <title> Tag – Der Name einer Webseite	8
3.2	Der <meta> Tag – Unsichtbare Daten für den Webbrowser/die Suchmaschine	8
3.2.1	Charset – Die Zeichenkodierung einer Webseite	9
3.2.2	Name/Content – Wertepaare im Meta Tag.....	9
3.3	refresh – Seite neu laden oder Weiterleiten.....	9
3.4	set-cookie – Cookie ohne JavaScript.....	9
3.5	Der <link> Tag – zur Einbindung externe Dateien	9
3.6	Der <style> Tag – Globale CSS Eigenschaften	9
	Zusammenfassung	10
	Aufgaben zur Selbstüberprüfung.....	11
4	HTML Tag zur Textformatierung und Strukturierung	12
4.1	Grundlegende HTML Tags	12
4.2	Textformatierung in HTML	12
4.2.1	Heading: <h></h> Tag – Überschriften	12
4.2.2	Paragraph: <p></p> Tag – Textabsätze und Break: – Zeilenumbruch.....	12
4.2.3	Horizontal Ruler: <hr> Tag – Thematische Brüche	13
4.2.4	Weitere HTML Tags zur Textformatierung	13
4.3	Anpassung von Inline-Elemente mittels Tag.....	14
4.4	Anchor: <a></> Tag - Verweise	14
4.4.1	Aufbau eines Verweises	15
4.4.2	Verweisanker anlegen	15
4.5	Image: Tag - Bilder.....	15
4.6	Listen.....	16
4.6.1	Ordered List: Tag – Geordnete Liste.....	16

4.6.2	Unordered List: Tag – Ungeordnete Liste.....	17
4.6.3	Description List: <dl></dl> Tag – Beschreibungsliste	17
4.7	Tabelle.....	18
4.8	Division: <div></div> Tag – Bereich	19
	Zusammenfassung	19
	Aufgaben zur Selbstüberprüfung.....	20
5	HTML5.....	21
5.1	Naiver Video- und Audio Wiedergabe.....	21
5.2	Inhaltabhängige Gliederungselementen	21
5.3	Formular Elemente	22
5.4	Canvas Element.....	22
5.5	Offline Datenspeicherung.....	22
5.6	Standortabfragen	23
	Zusammenfassung	23
	Aufgaben zur Selbstüberprüfung.....	24
A	Lösungen der Aufgaben zur Selbstüberprüfung.....	25
B	Literaturverzeichnis.....	27
C	Verwendete Tools.....	28
D	Abbildungsverzeichnis	28

1 Einführung

HTML ist die Abkürzung von Hypertext Markup Language, eine Auszeichnungssprache, die die Grundlage für das sichtbare World Wide Web liefert. Fast alle Webseite werden mittels HTML dargestellt, von Social Media Plattformen wie Facebook bis zu Videoplattformen wie YouTube, dabei ist HTML ursprünglich von Sir Tim Berners-Lee dazu entwickelt worden um wissenschaftliche Informationen strukturiert darzustellen und zu teilen.

Die Anfänge von HTML geht auf das Jahr 1989 zurück, wo die erste HTML Version von Tim Berners-Lee am CERN in Genf vorgestellt wurde. Der erste Webbrowser Prototyp "NeXT" stammte ebenfalls von Tim Berners-Lee und wurde 1990 veröffentlicht. vgl. [DRJLIAMK98, Kapitel 2]

Am 28.10.2018 veröffentlichte das World Wide Web Consortium (W3C) die aktuellste Version von HTML, HTML5. Dieses löste HTML4 sowie XHTML ab.

In dieser Ausarbeitung werden wir uns mit den Grundlagen von HTML sowie dessen Anwendung zur Erstellung einer einfachen Webseite beschäftigen. Zudem werden wir uns einige wichtige Neuerungen betrachten die mit HTML5 eingeführt wurde.

2 Was ist HTML

Dieses Kapitel beschäftigt sich mit den Grundlagen von HTML Dokumenten. Wir lernen hier was HTML ist sowie die Grundlegende Struktur jeder HTML Datei.

Sie sollten nach dem Durcharbeiten dieses Kapitel in der Lage sein eine einfache HTML Datei anzulegen.

HTML steht wie bereits angesprochen für Hypertext Markup Language und ist eine Auszeichnungssprache.

Standard HTML Dateien haben die Dateiendung .htm oder .html, es gibt jedoch auch einigen Ausnahmen von dieser Konvention. HTML Dateien die PHP Abschnitte haben zum Beispiel die Endung .php

Wichtig sind diese Endungen für den Server, der anhand der Endungen den Webbrowser mitteilen kann, um welches MINE-Type (Multipurpose Internet Mail Extension) es sich bei der übermittelten Datei handelt. Der Webbrowser kann dann anhand des übermittelten MINE-Type die übermittelte HTML Datei korrekt verarbeiten.

„In Dokumenten werden in der Regel gleichartige Elemente wie zum Beispiel Titel, Überschriften, Absätze, Listen, Abbildungen, Tabellen und Verzeichnisse verwendet. Diese Elemente sollen konsequent über mehrere Dokumente hinweg in einheitlicher Form dargestellt werden.“ [FE10]

Die HTML Datei selbst enthält den Inhalt sowie die Struktur eines Dokumentes, die Darstellung des Inhaltes kann zwar auch innerhalb der HTML Datei selber realisiert werden, häufiger wird es aber in einer separaten CSS-Datei ausgelagert.

Damit gibt es eine Trennung zwischen dem Inhalt selbst und die Darstellung, dies ist notwendig da bei größeren Webseiten mehrere HTML Dateien auf dieselbe CSS Datei für die Darstellung zurückgreifen kann. Das Design der Webseite kann somit auch zentral geändert und angepasst werden, dies ist nicht möglich, wenn jeder HTML Datei selber für die Darstellung zuständig ist.

In einer Webseite existieren neben der HTML und CSS Dokumentes auch noch weitere wichtige Dokumente, die für die Darstellung einer modernen Webseite unverzichtbar sind. Ein solches Dokument ist das Dokument, das die Skripten der Webseite beinhaltet, sofern diese nicht in der HTML Datei selbst vorhanden ist. Die Skripten, meist als JavaScript Dateien vorhanden, sind für die Interaktion mit dem Nutzer der Webseite zuständig.

Es gibt auch dynamischen Webseiten die mittels PHP, kurz für Hypertext Preprocessor, dargestellt werden. Unteranwendung einer Datenbank, zum Beispiel MySQL, ist es dem Server damit möglich je nach Aufrufer eine individuelle Webseite zu erstellen.

Wichtig ist hierbei jedoch, auch wenn es bei Verwendung von PHP keine eigentlich HTML-Datei mit einem Quellcode auf dem Server vorhanden ist, der Server muss dennoch eine individuelle HTML Datei generieren und es dem Nutzer zusenden, ganz auf HTML kann man also auch hier nicht verzichten.

Diese sind jedoch nicht Gegenstand dieser Ausarbeitung und werden daher nicht näher betrachtet.

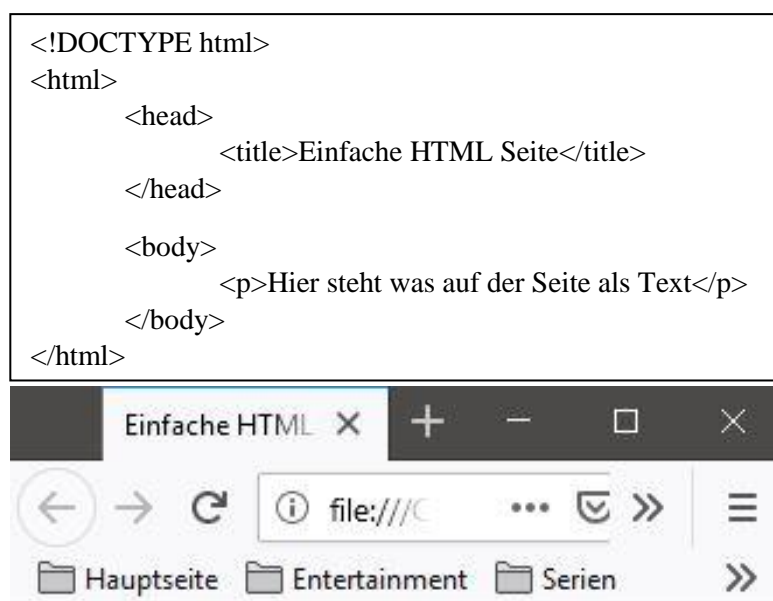
2.1 Aufbau eines HTML Befehl

HTML spezifiziert verschiedene Befehle, die von Webbrowser interpretiert werden kann. Diese Befehle werden auch HTML-Tags genannt und dienen dazu die Struktur des überlieferten Dokuments zu formatieren.

HTML Tag beginnen stets mit einen "<" und enden mit einen ">", zwischen den beiden Zeichen steht der auszuführende Befehl. Die meisten HTML Befehlen müssen durch einen abschließenden HTML-End-Tag abgeschlossen werden, diese beginnt mit "</" und endet mit ">".

Es existieren jedoch auch HTML-Tag die nicht abgeschlossen werden müssen wie z.B.
, <hr>, usw. Diese HTML-Tag werden auch leere Elemente genannt, in Englischen Empty HTML Elements, da sie keine Inhalte beinhalten.

2.2 Aufbau eines HTML Dokument



Hier steht was auf der Seite als Text

Abb. 1 Einfache HTML Dokument und dessen Darstellung im Browser

Die Abb. 1 zeigt wie der Code einer einfachen HTML Datei aussehen kann. Das Dokument beginnt stets mit <!DOCTYPE html>, dieser teilt den verwendeten Webbrowser mit was für Befehle er zu verarbeiten hat.

DOCTYPE legt hier die Document Type Definition (DTD) das Dokument fest, diese definiert die Grammatik sowie der Syntax des Dokuments. Seit HTML5 wurde diese vereinheitlicht zu "html".

Zuvor musste man für beispielweise HTML 4.0.1 Strict den DOCTYPE Tag wie folgt definieren:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN" "http://www.w3.org/TR/html4/strict.dtd">
```

Der Tag DOCTYPE ist hierbei jedoch kein HTML Tag, diese muss daher auch nicht geschlossen werden.

2.2.1 Der HTML-Tag <html>

Nach der DOCTYPE Deklaration wird der eigentlich HTML Code angegeben. Diese beginnt mit dem Tag <html> und endet mit dem Tag </html>. Diese HTML Tag umschließen zwei Blöcke des HTML Dokumentes.

2.2.2 Der HEAD-Block <head>

Im HEAD Block werden Metainformationen über die Webseite gespeichert, diese Informationen gehören nicht zur eigentlichen Webseite und dienen lediglich dem Zweck Informationen anzugeben, die für den Betrachter der Webseite nicht direkt einzusehen sind.

Der Tag <title> bildet hier die Ausnahme, da diese in den meisten Webbrowser angezeigt werden. Dieser ist auch der einzige Tag, der verpflichtend in einen HTML Dokument vorkommen muss.

Im HEAD Block können auch Informationen für Suchmaschinen oder die verwendete Zeichenkodierung angegeben werden. Häufig wird auch die verwendete CSS-Datei für die Darstellung der Webseite im HEAD Block eines HTML Dokuments verlinkt.

2.2.3 Der BODY-Block <body>

Im BODY Block wird der eigentliche Inhalt der Webseite sowie dessen Strukturierung gespeichert. Alles was in diesem Bereich erscheint, wird durch den Webbrowser gerendert und für uns sichtbar dargestellt.

2.3 Anlegen eines HTML Dokument

Da HTML eine Auszeichnungssprache ist, ist die Erstellung eines HTML Dokument durch jeden Text-Editor, unter Windows kann man hier auch einfach den Standard Editor von Windows verwenden.

Zum Anlegen eines einfachen HTML Dokument kann der Quellcode in der Abb. 1 kopiert und in den Editor von Windows hinzugefügt werden. Anschließend muss man noch die Datei als allgemeine Datei mit der Endung .htm oder .html speichern, die neu angelegte Datei kann von einem Webbrowser geöffnet und interpretiert werden.

Durch diese Einfachheit Webseite anlegen zu können, hat sich das HTML Format in den frühen Internet Zeiten als Standard im World Wide Web durchgesetzt.

Natürlich gibt es auch spezielle Tool für die Entwicklung einer Webseite, durch frühzeitige Erkennung von Syntaxfehler, automatisches hinzufügen von End-Tag oder durch Anzeigen von Verschachtelungsebenen ermöglicht ein solches Tool umfangreichere HTML Dokumente schneller zu durchblicken.

Ein sehr empfehlenswertes Programm ist hier Brackets von der Firma Adobe, dieses Tool bietet unter andere eine Livevorschau der Webseite noch während der Entwicklung des HTML Dokument, dies ermöglicht es dem Entwickler seine spätere Webseite noch während der Entwicklungsphase zu betrachten.

Zusammenfassung

In diesem Kapitel haben wir den Grundlegende Aufbau eines HTML Dokument angesehen sowie wie man eine einfache HTML Datei anlegen kann. Hier haben wir definiert das HTML eine Auszeichnungssprache ist, die dazu verwendet werden kann um Inhalte Struktur zu verleihen. E

in HTML Dokument besteht aus einen HEAD sowie ein BODY Block. Im HEAD Block werden Metainformationen über das Dokument gespeichert und im BODY Block wird der eigentliche Inhalt einer Webseite und dessen Struktur gespeichert.

Aufgaben zur Selbstüberprüfung

2.1:

Welche Verwendung hat der `<!DOCTYPE>` Tag am Anfang einer HTML Datei?

2.2:

Antworten Sie kurz warum HTML keine Programmiersprache ist!

2.3:

Geben Sie einen möglich kurzen Quellcode zu einer HTML Seite an, die nur die HTML Konventionen erfüllen muss!

3 Grundlegende HTML Tag im HEAD Block einer HTML Datei

In diesem Kapitel betrachten wir HTML Tags die im HEAD Block einer HTML Datei hinterlegt sind, diese Tags gibt Informationen zu der Webseite an die für den Nutzer selbst meist nicht von Bedeutung ist, da diese Informationen meist durch den Webbrowser nicht angezeigt werden, mit Ausnahme des Titel der Webseite.

3.1 Der <title> Tag – Der Name einer Webseite

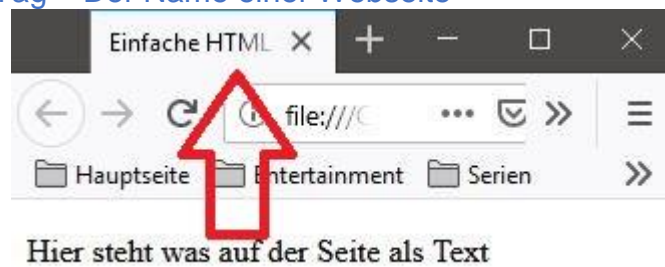


Abb. 2 Den Titel eines HTML Dokument im Browser Kopf

Der <title></title> gibt der Name der Webseite an, die Aufgabe dieses HTML Tag ist es der Name der Seite anzugeben, wenn der Browser die Webseite aufruft, den Name der Webseite wird hierbei im Browser Kopf dargestellt.

Wenn Sie eine Webseite als Favorit speichern möchte, ist der Name der im <title></title> enthalten ist, auch als Standard Name vorgeschlagen.

Eine wesentliche Aufgabe des <title></title> Tag ist es auch für Suchmaschinen wie Google den Titel eine Seite anzuzeigen, dieser Titel wird dann meistens auch als Titel der Webseite in den Suchergebnissen zu den betroffenen Seiten angezeigt.

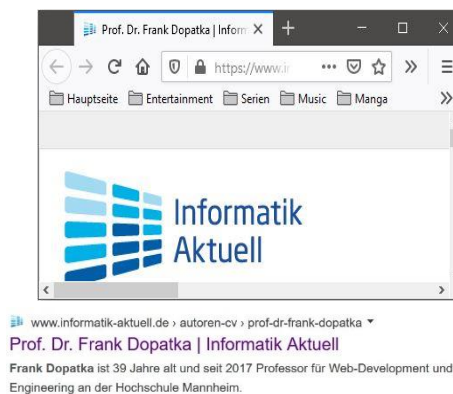


Abb. 3 Die Webseite <https://www.informatik-aktuell.de/autoren-cv/prof-dr-frank-dopatka.html>
Mit dem Titel "Prof. Dr. Frank Dopatka | Informatik Aktuell"
und dessen Darstellung in der Suchmaschine Google

3.2 Der <meta> Tag – Unsichtbare Daten für den Webbrowser/die Suchmaschine

Der Metainformationen sind Daten über Daten, die Informationen, die im leeren HTML Meta-Tag enthalten sind beschreibt das aktuelle HTML Dokument. Diese Informationen werden selber nicht auf der Webseite angezeigt, Sie sind aber für den Webbrowser oder Suchmaschinen lesbar. Die im Meta-Tag enthaltenen Informationen beschreiben oft die Sprache, den Autor oder die letzte Änderung einer Webseite.

Wir betrachten hier einige wichtige Meta-Tag eines HTML Dokumentes.

3.2.1 Charset – Die Zeichenkodierung einer Webseite

Die Angabe der Zeichenkodierung mit zum Beispiel `<meta charset= "UTF-8" />` stellt gibt den Webbrowser die Anweisung die Seite in UTF-8 Zeichen zu kodieren. Dies ist notwendig, wenn der verwendete Server, auf dem die Webseite läuft standardmäßig mit einer anderen Zeichenkodierung arbeitet. Hiermit stellt man sicher das die Darstellung der Webseite in allen Bedingungen in richtigem Format erfolgt.

3.2.2 Name/Content – Wertepaare im Meta Tag

Der Name/Content Tag im Meta-Tag ist ein Wertepaar, wo bei ersteres der Name angibt und letzteres den Inhalt. Diese Informationen legen immer als Paar vor und dienen meisten dazu Suchmaschinen anzusprechen. Hier können nähere Informationen zu einer Webseite hinterlegt werden wie, Sprache, Auto usw.

```
<meta name="Beschreibung" content="Wert">

<meta name="content-language" content="de">
- legt die Sprache des Inhalts auf Deutsch fest

<meta name="author" content="Phan Hoang Minh">
- legt der Name des Autors als Phan Hoang Minh fest

<meta name="keywords" content="Web, html">
- definiert Wörter, unter der die Seite über eine
Suchmaschine zu finden ist
```

Abb. 4 Code Beispiele zu dem Namen/Content Meta Tag

3.3 refresh – Seite neu laden oder Weiterleiten

Mit dem Meta Tag `<meta http-equiv="refresh" content="10;url=https://www.google.de/">` kann eine automatische Weiterleitung auf die angegebene URL nach 10 Sekunden erfolgen. Wenn man auf die Angabe der URL verzichtet wird die Webseite automatisch neugeladen nach 10 Sekunden.

3.4 set-cookie – Cookie ohne JavaScript

Es ist möglich Cookie zu setzen ganz ohne JavaScript anwenden zu müssen.

Der Beispiel Code dafür sieht wie folgt aus:

```
<meta http-equiv="set-cookie" content="Name=Hoang Minh; expires=Thursday, 05-December-19">
```

Das Verwalten von Cookies wird dennoch meisten von JavaScript übernommen, da diese flexiblere Einsatzmöglichkeiten für Cookies bieten.

3.5 Der <link> Tag – zur Einbindung externe Dateien

Der leere HTML `<link>` Tag erzeugt einen Link, der das aktuelle HTML Dokument mit einer externen Datei verknüpft. Hiermit können beispielweise externe CSS und JavaScript Dateien in das aktuellen HTML Dokument eingebunden und verwendet werden.

Ein Beispiel wie es verwendet wird, wird in dem Code Beispiel in der Abb.3.6.1 gezeigt. Es ist Konvention das die CSS Datei möglich früh geladen werden sollte, daher steht der `<link>` Tag mit der externe CSS Datei normalerweise am Anfang des Header Block.

Natürlich kann CSS Dateien auch am Ende des BODY Block eingebunden werden, dies wird häufiger verwendet um das Stylesheet für den Drucker zu laden, damit diese das Aufbauen der Seite nicht unnötiger weise verlangsamt.

3.6 Der <style> Tag – Globale CSS Eigenschaften

Innerhalb des `<style></style>` Tag kann globale CSS Eigenschaften für die Darstellung des HTML Dokument festgelegt werden. Diese hat eine höhere Priorität-Stufe als extern eingebundenen CSS

Dateien, daher können hier Darstellungen spezifisch für diese HTML Datei festgelegt werden, die von den Darstellungen in der eingebundenen CSS Dateien abweichen.

Mit Hilfe des folgenden Codes können alle Textblöcke blau eingefärbt werden, auch wenn diese in einer externe CSS Datei mit grün hinterlegt ist.

Wichtig ist, das lokale CSS Formatierung eine noch höhere Priorität-Stufe bei der Darstellung besitzen als Globale CSS Definitionen, in dem genanntes Beispiel in der Abb. 5 wird der Text "Ich bin rot" in roter Schrift dargestellt, obwohl Texte in der Globale Style Definition als blau definiert ist.

```
<!DOCTYPE html>
<html>
<head>
  <link rel="stylesheet" href="stylesheet_TEXTSCHRIFT_green.css" type="text/css">
  <style>
    p { color: blue}
  </style>
</head>

<body>
  <p>Ich bin blau</p>
  <p style="color:red;">Ich bin rot</p>
</body>
</html>
```

Code in externe **stylesheet_TEXTSCHRIFT_green.css** Datei:

```
p {
  color: green;
}
```

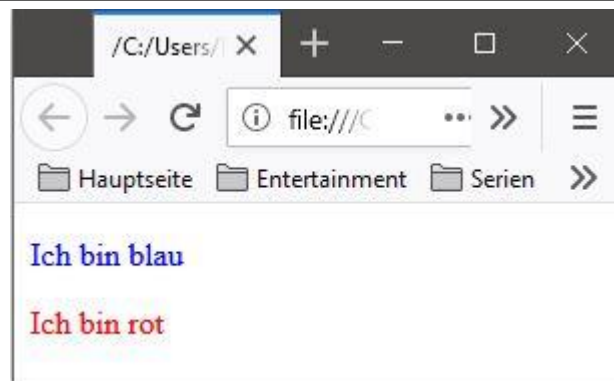


Abb. 5 Darstellung eines HTML Code in Webbrowser, wo die eingebundene CSS Datei überschrieben wird

Zusammenfassung

In diesem Kapitel haben wir uns angesehen wie man Metainformationen im HEAD Block eines HTML Dokument speichern kann. Zusätzlich haben wir uns angesehen wie CSS-Styleguide im Head Block eines HTML Dokument angelegt werden können, sowie die Möglichkeit des Einbindens von externen CSS und JavaScript Dateien.

Aufgaben zur Selbstüberprüfung

3.1

Nennen Sie die Prioritätenstufen in der Umsetzung von CSS Formatierungen!

3.2

Begründen Sie warum es sinnvoll ist eine externe CSS Datei für die Formatierung einer Webseite zu benutzen!

3.3

Geben Sie an, welche Informationen sinnvollerweise in HEAD-Block eines HTML Dokument gespeichert werden sollte!

3.4

Geben Sie eine weitere Möglichkeit an dem Meta Tag `<meta http-equiv="refresh">` zu verwenden außer um die Seite neuzuladen!

4 HTML Tag zur Textformatierung und Strukturierung

In diesem Kapitel wenden wir uns den wichtigsten HTML Tag zu, hier lernen Sie wie Sie wie Sie Ihre Webseite Struktur verleihen kann. Auch lernen Sie hier wie man Texte formatieren kann.

Nach dem Abarbeiten dieses Kapitels sollten Sie in der Lage sein, komplexeren Webseiten mit Texten, Tabellen, Listen und Grafiken anzulegen.

Es gibt über 100 verschiedenen Tag in HTML bzw. HTML5, wir werden uns hier nur mit dem wichtigsten HTML Tags befassen, die zur Strukturierung einer Webseite notwendig sind.

4.1 Grundlegende HTML Tags

Alle HTML Tags, die in diesem Kapitel vorgestellt werden, befinden sich im BODY Block des HTML Dokument, da dieser Block für den Inhalt sowie die Struktur einer Webseite verantwortlich ist.

4.2 Textformatierung in HTML

Wir betrachten hier HTML Tag die es uns ermöglichen Texte zu formatieren, insbesondere werden wir hier Überschriften und Paragraphen betrachten, da dieser ein wesentlichen Bestandteil eines jeden Textes ist.

4.2.1 Heading: <h></h> Tag – Überschriften

Überschriften sind ein wichtiger Bestandteil um Texte in HTML Dokumenten zu strukturieren, HTML kennt 6 Überschriftenebenen, von <h1></h1> bis zum <h6></h6>. Diese werden durch den Webbrowser in jeweils andere Schriftgrößen angegeben. Wenn nicht anders formatiert, wird <h1></h1> am größten und <h6></h6> am kleinsten dargestellt.

Überschriften in einen Text kann man hier wie Kapitel Namen in einen Buch Betrachten, diese dienen dazu einen neuen Textabschnitt einzuläuten.

4.2.2 Paragraph: <p></p> Tag – Textabsätze und Break:
 – Zeilenumbruch

Absätze dienen der Gliederung des Textes, der <p></p> Tag umfasst genau einen Absatz, er steht nicht für den Zwischen Raum zwischen zwei Absätzen. Hier verwendet man normalerweise einen neuen Absatz Block um den nächsten Absatz anzulegen.

Der Webbrowser erzeugt bei der Darstellung der Webseite automatisch einen Zwischenraum zwischen den beiden Absatz Blöcke.

Der leere HTML Tag
 dient dazu einen Zeilenbruch zu erzeugen, es wird meistens verwendet um ein Gedicht oder eine Adresse zu notieren, da Zeilenumbrüche hier semantisch von Bedeutung sind.

Der
 Tag sollte nicht dazu verwendet werden um Lücken in einen Absatz zu erzeugen oder diesen zu vergrößern.

```
<h1>Adresse:</h1>
<p>Phan Hoang Minh<br>
    Am Mühlentor 11<br>
    19417 Warin
</p>
```

Abb. 6 Angaben einer Adresse in einen Absatz mit Hilfe von Zeilenumbrüchen

4.2.3 Horizontal Ruler: <hr> Tag – Thematische Brüche

Der leere <hr> Tag leitet einen Themenwechsel ein in einen HTML Dokument und wird durch eine horizontale Linie dargestellt. Diese sichtbaren Trennlinien dienen auch dazu um visuell eine Abgrenzung von Absatz Blöcke anzuzeigen.

Der <hr> Tag fügt eine Trennlinie ein und erzeugt damit selber einen eigenen Absatz.

4.2.4 Weitere HTML Tags zur Textformatierung

HTML bietet den Entwickler auch weitere Möglichkeiten an, um einen Text zu formatieren. Dies ermöglicht dem Entwickler, nur mit Hilfe von HTML Tag, einen Text, wie in einen Text-Editor, anzupassen.

Im Folgenden werden einige HTML Tag vorgestellt die bei der Formatierung eines Textes eine Rolle spielen kann. Wir betrachten nur einige wichtige Tag die für die Formatierung eines Textes benötigt werden.

Am besten kann man ein solches Beispiel anhand von einer direkten Gegenüberstellung eines Textabsatzes in HTML und dessen Darstellung im Browser zeigen.

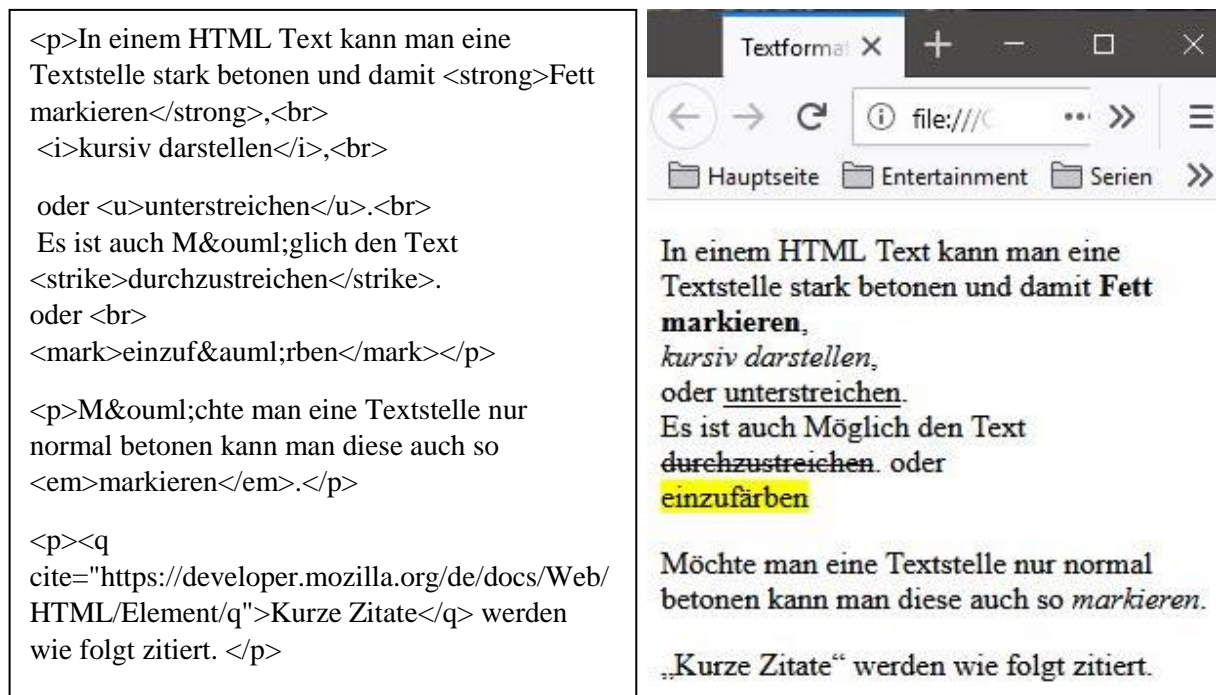


Abb. 7 Quellcode mit HTML Textformatierung Elementen zeigen

Wie im Beispiel in der Abb. 4.2.4.1 gezeigt kann man einen Text mit Hilfe des als Stark betont markieren, dieser Text wird, in den meisten Webbrowser, als Fett markiert dargestellt. Wenn man den Text jedoch kontextlos, ohne Betonungen Fett markieren möchte, kann man hierfür auch den Tag benutzen.

Mit kann man einen Text normal betonen, der Webbrowser stellt diesen Text meist kursiv da.

HTML sieht vor, bei explizit kursiv darzustellender Text, den Tag <i></i> zu verwenden. Um Texte hervorzuheben ist eine Nutzung des HTML Tag <mark></mark> möglich. Diese färbt den Text Hintergrund gelb ein, wie bei einem Textmarker, wenn nicht anderes zuvor mittels CSS definiert wurde.

Sollte der Text unterstrichen werden, verwendet man den Tag <u></u>. Beim durchstreichen des Textes kann der Tag verwendet werden.

Mit Hilfe von der `<q></q>` Tag kann man Zitate in dem HTML Dokument anzeigen, in den `<q>` Tag selber, kann man je nach Bedarf mit `<q cite="url">` eine Online Quelle, die man hier zitiert hat, angeben. Anzumerken ist, dass der `<q></q>` Tag nur für Kurze Zitate vorgesehen ist, längere Zitate sollte man mit dem `<blockquote></blockquote>` Tag zitieren.

Alle diese HTML Tag können auch miteinander kombiniert werden, so erhält man mit `<i><u>Text</u></i>` einen Stark Betonten Text (fett markiert) der zusätzlich dazu auch noch kursiv und unterstrichen ist.

Wichtig ist hierbei das man die HTML Tags auch an der richtigen Stelle schließen muss, der Anfang Tag muss zuletzt geschlossen werden, der zweiter als Tag als vorletztes usw.

4.3 Anpassung von Inline-Elemente mittels `` Tag

Anmerkung: „Inline-Elemente nehmen nur den Platz in Anspruch, der auch vom Inhalt benötigt wird. Typische Inline-Elemente sind beispielsweise die Hervorhebungen wie `` und `<i>` bzw. Links wie `<a>`“ [HTMLSEMINAR]

Betrachten wir hier einige Möglichkeiten einen Text oder andere Inline Elemente zu formatieren. Eine Möglichkeit dazu bietet der HTML Tag ``. Der `` selber hat keinen semantischen Inhalt, und bewirkt von sich aus nichts. Er dient lediglich dazu Inline Elemente zu umfassen und diese mittels CSS zu formatieren.

```
<head>
  <style>
    span { color: red; }
  </style>
</head>
<body>
  <p>Die <span>ueberspannte Stelle</span> ist rot</p>
</body>
```

Abb. 8 Der Code bewirkt, dass "ueberspannte Stelle" rot eingefärbt wird

Natürlich kann auch der `` Tag lokal eingesetzt werden um Texte lokal priorisiert zu formatieren.

```
<p><span style="color: white; background-color: red; font-size: 32px; font-family: Time New Roman;">
Roter Hintergrund, weisse Schriftfarbe und Times New Roman als Schriftart
</span></p>
```

Abb. 9 Der Code zeigt wie in `` Tag selber ein Text mittels CSS formatiert werden kann

4.4 Anchor: `<a>` Tag - Verweise

Der `<a>` Tag ist im HTML eines der zentralen Elemente zum Aufbau von komplexen Webseiten. Verweise können zu anderen Webseiten verlinkt werden. Zu eine andere HTML oder beliebiges Dokument/Datei im eigenen Projektordner. Diese Verweise können aber auch auf einen Verweisziel im aktuellen Dokument zeigen.

Mit Hilfe von Verweisen können mehrere HTML Seiten miteinander verlinkt werden, eine statische Homepage wird so realisiert.

4.4.1 Aufbau eines Verweises

```
<a href="Verweisziel" title="Verweisziel" target="_blank"
download="beispiel.zip">Verweislink</a>
```

Abb. 10 Beispiel Code für einen Verweis

Verweisziel ist eine verpflichtende Angabe beim Konstruieren eines Verweises, hier wird das Ziel angegeben, welches dem Benutzer nach dem Anklicken auf dem Verweislink hinführen soll.

Der Wert im Title wird angezeigt, wenn der Mauszeiger über dem Download Link schwebt, diese Angabe ist optimal.

Mit der Ergänzung target="_blank" kann man angeben das der Link in einem neuen Fenster oder Webbrowser Tab geöffnet werden soll.

Mit der Ergänzung des Verweises um das Attribut "download" kann man eine Datei zum Download anbieten. Wenn der Benutzer auf dem Link klickt, wird Ihn die Datei von Verweisziel zum Download angeboten.

Durch die Angabe eines Wertes für "download" wird ein Name, für den zum Herunterladen freigegebene Datei, für den Benutzer vorgeschlagen.

4.4.2 Verweisanker anlegen

Man kann in ein HTML Dokument Verweisanker definieren, auf diese Anker kann mit Hilfe eines Verweises verlinkt werden, so springt der Nutzer direkt zu der Anker Stelle, wenn er die Seite mit Hilfe eines solchen Verweises aufruft.

Einen unsichtbaren Verweisanker in einen HTML Dokument kann man, wie der Verweis selber, mithilfe des <a> Tag anlegen, um den Ankerpunkt aufzurufen, muss man hier ebenfalls den <a> Tag benutzen.

Ankerpunkt:	
Verweis zu Ankerpunkt:	Verweistext

Abb. 11 Beispiel wie man einen Verweisanker anlegt und auf diesen verweisen kann

Man kann auch zu jedem beliebigen Punkt in eine HTML Datei verweisen, in dem man auf ein bestimmtes id-Attribut verweist, welches in einen HTML-Element zusätzlich angegeben ist.

Ankerpunkt:	<h1 id="marker1">Kapitel 1</h2>
Verweis zu "marker1"	zum Kapitel 1

Abb. 12 Beispiel für einen Verweisanker mit Hilfe des id-Attribut

4.5 Image: Tag - Bilder

Mit dem leeren HTML Tag kann man Bilder und Grafiken in eine HTML Datei einbinden. Das Tag wird wie folgt aufgebaut:

Die beiden Attribute "src" und "alt" müssen immer angegeben werden, hier wird der Speicherort des Bildes oder die Grafik hinterlegt. Das Attribut "alt" gibt einen alternativen Text an, der angezeigt wird, wenn das Bild oder die Grafik nicht geladen werden können.

```

```

Abb. 13 Beispiel Code zum Einbinden eines Bildes

Die beiden anderen Attribute "width" und "height" geben optimal die Bildbreite und Bild Höhe an, diese Angaben sind nicht verpflichtend. Man sollte diese dennoch stets mitangeben, so kann der Browser bereits bevor er die Bilder geladen hat, leere Bildflächen für die spätere Darstellung anzeigen.

Bilder können auch als Link fungieren, mit folgendem Code kann man ein Bild als Verweis nutzen.

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

Abb. 14 Beispiel Code um ein Bild als Linkverweis zu nutzen

4.6 Listen

In HTML gibt es 3 verschiedenen Arten von Listen, die geordnete List, die ungeordnete Liste und die Beschreibende Liste. Diese Listen dienen jeweils einen anderen speziellen Zweck, eine kurze Einführung zu Listen finden Sie jetzt hiernach.

4.6.1 Ordered List: Tag – Geordnete Liste

```
<ol start="17">
  <li>Beginnt mit 17</li>
  <li>Lucky 18</li>
  <li value="4">Wieder bei 4</li>
  <li>Weiter mit 5</li>
</ol>
```

17. Beginnt mit 17
18. Lucky 18
4. Wieder bei 4
5. Weiter mit 5

Abb. 15 Code Beispiel für eine geordnete Liste und dessen Darstellung im Webbrowser

Der HTML Tag gibt eine nummerierte geordnete Liste an, die Reihenfolge der Elemente sind hier semantisch von Bedeutung. Jedes Listen-Element wird von dem Tag eingeschlossen.

In eine geordnete Liste kann man mit dem "start" Attribut angeben, ab welcher Zahl in der Liste gezählt wird. Man kann zudem den Zählwert, an jede beliebige Listenelement, neufestlegen mit dem "value" Attribut.

```
<ol start="17">
  <li>Beginnt mit 17</li>
  <ol start="17">
    <li>Beginnt mit 17</li>
    <li>Lucky 18</li>
    <li value="4">Wieder bei 4</li>
    <li>Weiter mit 5</li>
  </ol>
  <li>Lucky 18</li>
  <li value="4">Wieder bei 4</li>
  <li>Weiter mit 5</li>
</ol>
```

17. Beginnt mit 17
17. Beginnt mit 17
18. Lucky 18
4. Wieder bei 4
5. Weiter mit 5
18. Lucky 18
4. Wieder bei 4
5. Weiter mit 5

Abb. 16 Beispielcode für eine verschachtelte geordnete Liste, sowie dessen Darstellung im Browser

Eine Verschachtelung der geordneten Liste ist ebenfalls möglich, hierbei ist jedoch zu beachten, das mit HTML allein eine „Nummerierungshierarchien wie 1, 1.1, 1.1.1“ [FD19, S.15] nicht möglich ist.

4.6.2 Unordered List: Tag – Ungeordnete Liste

```
<ul>
  <li>Kaffee ist gut</li>
  <li>Redbull ist gut</li>
  <li>Alkohol ist weniger gut</li>
  <li>Computerspiele ist gut</li>
</ul>
```

- Kaffee ist gut
- Redbull ist gut
- Alkohol ist weniger gut
- Computerspiele ist gut

Abb. 17 Code Beispiel für ungeordnete Liste und dessen Darstellung im Webbrowser

Sollten die geordnete Reihenfolge der Listenelemente semantisch keine Rolle spielen, dann kann man diese Listen-Elementen auch in eine ungeordnete Liste hinterlegen. Diese wird erzeugt durch dem HTML-Tag , auch hier werden die Listenelemente in den Tag eingeschlossen.

Die meisten Webbrowser stellen eine ungeordnete Liste als eine Aufzählungsliste mit Aufzählungszeichen (Bullet) dar. Auch hier ist es selbstverständlich möglich eine Liste in eine andere zu Verschachteln.

Moderne Webbrowser können hier, für die Unterliste, andere Aufzählungszeichen verwenden als für die Oberliste.

```
<ul>
  <li>Kaffee ist gut</li>
  <li>Redbull ist gut</li>
  <ul>
    <li>Kaffee ist gut</li>
    <li>Redbull ist gut</li>
    <li>Alkohol ist weniger gut</li>
    <li>Computerspiele ist gut</li>
  </ul>
  <li>Alkohol ist weniger gut</li>
  <li>Computerspiele ist gut</li>
</ul>
```

- Kaffee ist gut
- Redbull ist gut
 - Kaffee ist gut
 - Redbull ist gut
 - Alkohol ist weniger gut
 - Computerspiele ist gut
- Alkohol ist weniger gut
- Computerspiele ist gut

Abb. 18 Codebeispiel einer verschachtelten ungeordneten Liste sowie dessen Barstellung im Webbrowser

4.6.3 Description List: <dl></dl> Tag – Beschreibungsliste

Eine Description List, zu Deutsch Definitionslisten, ist eine Listenart, die seit HTML5 dazu benutzt wird, um eine beschreibende Liste darzustellen. Solche Listen kann man benutzen um Glossare darzustellen.

„Ein Glossar besteht meist aus einer Liste von Einträgen, die wiederum aus mindestens einem zu erklärenden Sachverhalt nebst mindestens einer Erklärung besteht.“ [FD19, S.17]

Die Beschreibungsliste wird durch den Tag <dl></dl> definiert. Innerhalb der Liste wird der zu erläuternde Ausdruck in den Tag <dt></dt> eingeschlossen, die Beschreibung des zu erläuternden Ausdrucks erfolgt innerhalb des Tags <dd></dd> direkt hinter dem <dt></dt> Tag.

Man könnte hier auch sagen, dass die Beschreibungsliste Wertepaare speichern.

```

<dl>
  <dt>HTML</dt>
  <dd>Für Struktur in einer Webseite</dd>
  <dt>CSS</dt>
  <dd>Für Darstellung in einer Webseite</dd>
  <dt>Javascript</dt>
  <dd>Für Interaktion in einer Webseite</dd>
</dl>

```

HTML

Für Struktur der Webseite

CSS

Für Darstellung der Webseite

Javascript

Für Interaktion der Webseite

Abb. 19 Codebeispiel einer Description List, sowie dessen Darstellung in einem Webbrowser

4.7 Tabelle

Mit dem Tag `<table></table>` kann man in einen HTML Dokument eine Tabelle anlegen.

Der `<table></table>` Tag wird hierbei mit dem verschiedenen Tag wie `<thead></thead>`, `<tbody></tbody>`, `<tr></tr>`, `<th></th>` und dem `<td></td>` Tag kombiniert um eine Tabelle darzustellen.

```

<table border="1">
  <tbody>
    <tr>
      <th>Spalte 1</th>
      <th>Spalte 2</th>
      <th>Spalte 3</th>
    <tr>
      <td>Wert 1 Zeile 1</td>
      <td>Wert 2 Zeile 1</td>
      <td>Wert 3 Zeile 1</td>
    <tr>
      <td>Wert 1 Zeile 2</td>
      <td>Wert 2 Zeile 2</td>
      <td>Wert 3 Zeile 2</td>
    </tbody>
</table>

```

Spalte 1	Spalte 2	Spalte 3
Wert 1 Zeile 1	Wert 2 zeile 1	Wert 3 Zeile 1
Wert 1 Zeile 2	Wert 2 Zeile 2	Wert 3 Zeile 2

Abb. 20 Codebeispiel für eine Tabelle und dessen Darstellung im Browser

Der `<tbody></tbody>` Tag markiert hierbei den Beginn des Inhaltes in einer Tabelle, mit dem `<tr></tr>` Tag werden dann jeweils eine Zeile angelegt.

Mit dem `<th></th>` Tag definiert man jeweils den Kopf einer Spalte (Spaltenname) und mit dem `<td></td>` Tag werden die Werte in einer Zeile befüllt.

Wenn gewünscht kann auch die `<th></th>` Tag Zeile für den Tabellenkopf in eine spezielle dafür vorgesehene `<thead></thead>` bereich definiert werden.

Es gibt auch noch dem `<tfoot></tfoot>` Tag, hier werden meist die Ergebnisse der vorherigen Zeilen zusammengefasst und angegeben.

Wie in Abb.20 gezeigt, benötigt man die beiden Tags `<thead></thead>` und `<tfoot></tfoot>` nicht unbedingt, um eine Tabelle darzustellen. Diese dienen lediglich der besseren Strukturierung einer Tabelle, da Sie Tabellenelementen Explizit definieren.

Mit HTML ist es auch möglich eine Tabelle weiter anzupassen, mit den Attributen `<td colspan="x">` kann man eine Zelle in eine Zeile auf **x** Zellen verbreiten, die Zelle nimmt damit den Platz für x Zellen ein.

Auch ist es möglich eine Zelle in einer Spalte auf mehreren Zellen auszubreiten, mit `<td rowspan="y">` wird einer Zelle in einer Spalte um **y** Zellen erweitert.

```
<table border="1">
  <tbody>
    <tr>
      <th>Spalte 1</th>
      <th>Spalte 2</th>
      <th>Spalte 3</th>
    </tr>
    <tr>
      <td rowspan="2" colspan="2">Wert 1 Zeile 1</td>
      <td>Wert 3 Zeile 1</td>
    </tr>
    <tr>
      <td colspan="2">Wert 3 Zeile 2</td>
    </tr>
  </tbody>
</table>
```

Spalte 1	Spalte 2	Spalte 3
Wert 1 Zeile 1		Wert 3 Zeile 1
Wert 3 Zeile 2		Wert 3 Zeile 2

Abb. 21 Codebeispiel für die Formatierung einer Tabelle mit `rowspan` und `colspan` sowie dessen Darstellung im Webbrowser

Wie wir bereits kennengelernt haben, ist in HTML auch eine Kombination der beiden Attribute möglich mit `<td rowspan="2" colspan="2">`.

Diese Tabelle Zelle würde insgesamt 4 Zellen Platz in Anspruch nehmen, **2** vertikale und **2** horizontale.

4.8 Division: `<div></div>` Tag – Bereich

Der `<div></div>` Tag definiert einen Bereich oder Sektion in einen HTML Dokument. Das `div`-Element Element ist meist dazu gedacht, mehrere Elemente eines HTML Dokument wie Text, Listen, Tabellen usw. in einen Bereich zu gruppieren, damit diese mittels CSS formatiert werden kann.

Zusammenfassung

In diesem Kapitel haben wir mehrere elementare Elemente eines HTML Dokumentes kennengelernt. Wir haben kennengelernt wie man Texte formatiert, Verweise anlegt und Bilder einbindet. Wir haben zudem gelernt wie wir Listen und Tabellen anlegen können, sowie den Tag `<div></div>` kurz kennengelernt. Diese wird uns sehr helfen bei der Formatierung von größeren Webseiten mittels CSS, aber das ist nicht Gegenstand dieser Ausarbeitung, daher werden wir darauf nicht näher eingehen.

Sie sollten nach dem Abarbeiten dieses Kapitel bereits in der Lage sein, eine komplexere Webseite zu entwickeln.

Aufgaben zur Selbstüberprüfung

4.1

Beschreiben Sie den Unterschied zwischen den `` und `<div>` Tag!

4.2

Geben Sie die vorhandenen Listen Arten in HTML an!

4.3

Geben Sie vier mögliche Formatierung arten zu einem Text in HTML an!

4.4

Was bewirken die beiden Attribute `rowspan` und `colspan` in einer Tabelle?

4.5

Warum müssen Sie stets das Attribut `alt` angeben, wenn Sie ein Bild in Ihrer HTML Dokument verlinken wollen?

4.6

Wozu verwendet man einen Verweisanker in einen HTML Dokument?

4.7

Recherchieren Sie eigenständig nach, wie Sonderzeichen in der deutschen Sprache in HTML Dokument dargestellt werden.

5 HTML5

HTML5 ist der Momentan aktuellster HTML Standard, dieser ist der Nachfolger von HTML4 und XHTML. Wichtig ist anzumerken, dass HTML5 nahezu zu 100 Prozent abwärtskompatibel ist, das heißt, sollte Ihrer Webseite bisher aller Standard und Konventionen eingehalten haben, ist dieser weiterhin durch einen Webbrowser darstellbar der nur HTML5 unterstützt.

HTML5 bringt neben einige Komfort Änderungen, wie das ermöglichen des weglassen von HEAD und BODY Blöcke in einen HTML Dokument, einige wichtige Neuerungen, die wir in diesem Kapitel kennen lernen werden.

5.1 Naiver Video- und Audio Wiedergabe

```
<video width="320" height="240" controls>
  <source src="beispielVideo.mp4" type="video/mp4">
  <source src="beispielVideo.ogv" type="video/ogg">
  Dein Webbrowser unterstützt HTML5 leider nicht.
</video>

oder

<audio controls="control">
  <source src="beispielAudio.wav" type="audio/wav">
  Dein Webbrowser unterstützt HTML5 leider nicht.
</audio>
```

Abb. 22 Beispiel Code zum Einbinden eines Videos oder Audiodatei in HTML5

HTML bietet es nativ über das HTML Element `<video>` bzw. `<audio>` Video- und Audiodateien einzubinden, und diese mittels CSS zu formatieren und über JavaScript zu steuern. Damit kann auf zusätzlich Plug-In wie dem Adobe Flash Player verzichtet werden.

Man kann auch mehrere Dateien in unterschiedlichen Formaten hinterlegen, der Webbrowser sucht sich in diesem Fall das passende Format aus, den er abspielen kann.

5.2 Inhaltabhängige Gliederungselementen

HTML5 führt einige neuen Elementen ein, die dazu verwendet werden um eine Webseite noch besser zu gliedern. Die Verwendung des `<div></div>` Tag zur Gliederung des HTML Dokumentes in mehreren semantischen Abschnitten, haben wir bereits in Kapitel 4.8 betrachtet.

In HTML5 wird der `<div></div>` Tag durch `<section></section>` ersetzt, damit löst man das Problem, das man durch den `<div>` Tag nicht genau erkennen kann, ob es sich hierbei um einen semantischen Abschnitt oder ob es sich hierbei lediglich um einen Block handelt, der mittels CSS formatiert werden soll.

HTML5 führt ein spezifischer HTML `<nav></nav>` Element ein, der dazu genutzt wird, um eine Navigationsleiste zu erstellen, hier werden Linkssammlungen verwaltet, die die Homepage mit seinen Unterseiten verlinken.

Mit `<header></header>` und `<footer></footer>` Tag führt HTML5 auch neben `<section>` zwei spezifischer Segmente in einer Webseite, beim `<header>` Abschnitt handelt es sich um den Kopfbereich einer Webseite, dieser soll einleitende Inhalte der Webseite gliedern, hier werden häufig die horizontale Navigationsleiste eingegliedert oder das Banner der Seite wird hier angezeigt.

Beim `<header>` Abschnitt handelt es sich um das Ende eine Seite, Element die hier gegliedert sind sollen erst am Ende einer Webseite zu sehen sein. Meist werden hier die Links zur Impressum oder Datenschutzerklärungen angezeigt.

5.3 Formular Elemente

Mit HTML5 wurden neuen Input Typen eingeführt. Die unter HTML4 vorhandenen Input Typen text, hidden und password wurde um weiteren Typen erweitert, darunter sind date, email, tel usw.

Damit ist es möglich Informationen noch vor dem Absenden zu validieren auf Ihre Richtigkeit, dies erspart viel Programmierarbeit und erhöht zudem die Benutzerfreundlichkeit.

```
<form action="beispielurl">
  Name:
  <br>
  <input type="text" name="name">
  <br>
  <input type="submit">
</form>
```

Name:

Abb. 23 Codebeispiel für ein Formular zum Abfragen des Namens und dessen Darstellung im Webbrowser

Das Attribut action beinhaltet als Wert einen URL, wo die Daten hingesendet werden, wenn auf "Senden" geklickt wird. Beim Klick auf senden wird eine http Request ausgelöst die Seitens der Server verarbeitet wird. Handelt es sich bei der URL um eine ScriptDatei, dann wird diese den Request serverseitig bearbeiten.

5.4 Canvas Element

Der <canvas></canvas> Tag wird dazu verwendet um mittels JavaScript Grafiken auf eine Webseite darzustellen. Das <canvas></canvas> Element ist dabei jedoch nur ein Container für die Grafiken, man muss JavaScript verwenden um diese Grafiken zu erstellen.

Mit dem Canvas Element, ist es möglich Grafiken für die Darstellung in eine Webseite dynamisch zu erstellen ohne das diese Grafiken vorher auf dem Server abgelegt werden muss.

```
<!DOCTYPE html>
<html>
  <script type="text/javascript">
    if(window.addEventListener){
      addEventListener("load", drawCanvas, false);
    }
    else{
      attachEvent("onload", drawCanvas);
    }
    function drawCanvas(){
      var canvas = document.getElementById('testcanvas1');
      if(canvas.getContext){
        var context = canvas.getContext('2d');
        context.fillStyle = "rgb(255, 0, 255)";
        context.fillRect(0, 0, canvas.width, canvas.height);
      }
    }
  </script>
  <body>
    <h1>Canvas</h1>
    <canvas id="testcanvas1" width="300" height="200">
      Dein Browser kann diese Grafik nicht darstellen.
    </canvas>
  </body>
</html>
```

Canvas



Abb. 24 Code Beispiel um mit Canvas Eine Fläche zu Zeichnen [PK19]

5.5 Offline Datenspeicherung

HTML5 ermöglicht es WebApplication vollständig, ohne eine stetige Internetverbindung, zu funktionieren. Unter Verwendung des „HTML5 Application Cache“ [W3LAG14, S.7] und DOM Storage werden Ressourcen vorgeladen im Onlinebetrieb und im Offlinebetrieb werden Informationen in einer clientseitigen Datenbank angelegt, um einen Späteren Versand dieser Informationen zu ermöglichen. Wenn dieser beiden Prozesse gut genug miteinander agieren, dann wird der Nutzer nicht mitbekommen das er Offline ist.

Dieser Möglichkeit der Offline Nutzung ist notwendig, da beispielweise Smartphone oft Ihre Internetverbindung kurzfristig verlieren. Damit sollte dieser Technologie eine angenehmere Nutzererfahrung ermöglichen. Da der Nutzer durch die lokale Speicherung von einige Daten trotz Verbindungsabbrüche die Webseite weiter benutzen kann.

5.6 Standortabfragen

Die Geolocation API ermöglicht es dem Nutzer, einer Webapplication seinen aktuellen Standort mitzuteilen.

Die Standortangaben sind für die Anwendung mit einem Smartphone am präzisesten, da diese über einen eingebauten GPS Empfänger verfügt.

Der Standort kann von einer Navigation WebApp verwendet werden, um den Nutzer seinen Standort auf eine Karte mitzuteilen, oder auch einfach um das Wetter am Standort des Nutzers anzuzeigen.

In Verbindung mit der Suchmaschine Google wird uns hiermit auch oft die nächstgelegene Arztpraxis oder Restaurant angezeigt, wenn wir eine solche Suchanfrage starten und Google Zugriff auf unseren Standortinformationen erlauben.

Ein möglicher Code für eine solche Standortabfrage würde wie folgt aussehen.

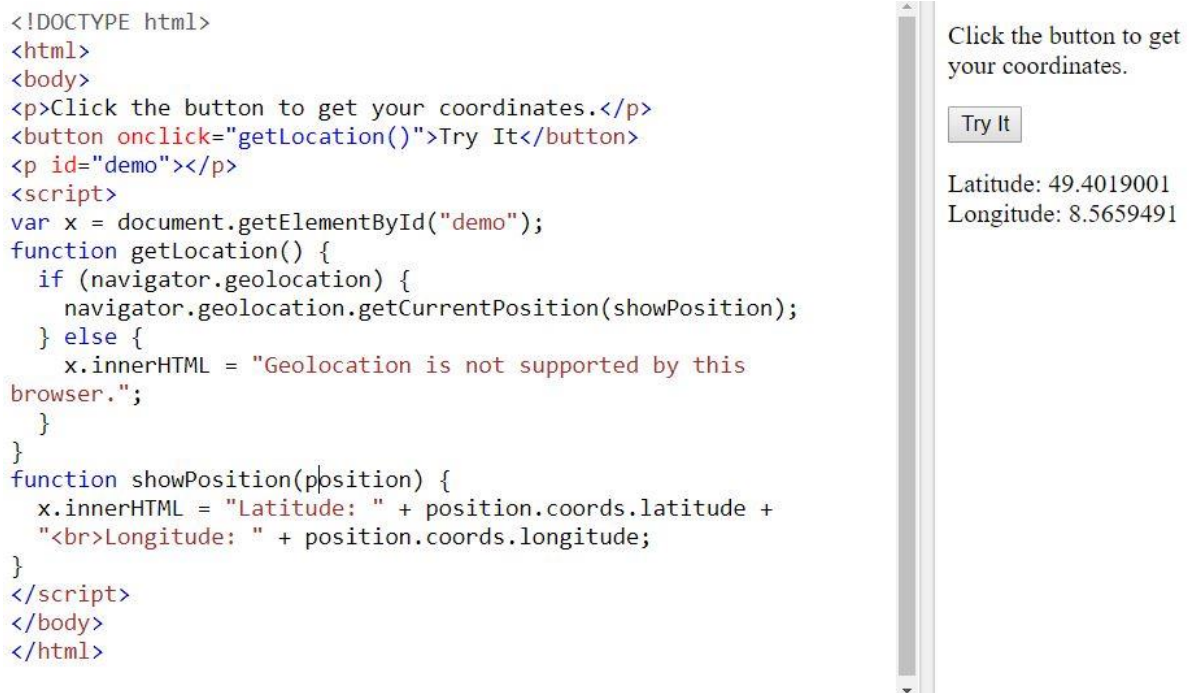


Abb. 25 Codebeispiel für eine Standortabfrage [HTML5GEO19]

Zusammenfassung

Wir haben in diesem Kapitel die Neuerungen angesehen, die in HTML5 eingeführt werden. Neben einige Änderungen zur Erhöhung der Benutzerfreundlichkeit wie die native Videowiedergabe durch den Webbrowser.

HTML5 auch neue Möglichkeiten für die Benutzung von Smartphone, hier wurden Technologien eingeführt, die auf die Nutzung von Smartphone ausgelegt sind, wie Offline Storage und Standortabfragen.

Neben den obengenannten Elementen bringt HTML5 auch die Möglichkeit Grafiken mittels JavaScript zu erstellen mittels Canvas Element, sowie eine bessere Möglichkeit zur Gliederung von Webseiten mit inhaltsabhängige Gliederungselementen.

Aufgaben zur Selbstüberprüfung

5.1

Was ist Ihrer Meinung nach, die wichtigste Neuerung in HTML5, begründen Sie Ihrer Antwort!

5.2

Nennen Sie 3 mögliche Anwendungsszenarien für die Nutzung Ihres Standortes.

5.3

Warum ist die Offline Fähigkeit von HTML5 einen gutes Feature für die Anwendung am Smartphone, begründen Sie Ihre Antwort.

5.4

Was ist der Nutzen von den neu eingeführten Input Typen in HTML5?

A Lösungen der Aufgaben zur Selbstüberprüfung

2.1

`<!DOCTYPE>` Am Anfang einer HTML Datei legt fest um was es sich für ein Dokumententyp es sich hierbei handelt, hier wird zudem in HTML Fall speziell die HTML Version angegeben, damit der Browser diese entsprechend interpretieren kann.

2.2

HTML ist eine Auszeichnungssprache, ihr fehlt es wesentliche Bestandteile einer Sprache damit man damit programmieren kann. Die beiden wichtigsten Bestandteile eine Programmiersprache, Schleifen und Bedingungen sind in HTML nicht vorhanden, daher ist diese auch keine Programmiersprache.

2.3

```
<!DOCTYPE html>
<html>
  <title>HTML</title>
</html>
```

Nach der Neuerung von HTML5 muss kein HEAD und BODY Block mehr angegeben werden, daher gerügt dieser Code vollständig als eine HTML Datei.

3.1

Als erstes werden die Lokale, danach die Globale und erst anschließend die externen CSS Regeln ausgeführt.

3.2

Bei umfangreichere Webpräsenzen mit mehrere HTML Dokumenten die alle das gleiche Design haben sollte, ist es sinnvoller das Design in eine externe Datei zu habe damit man bei Änderungen am Design diese zentral vornehmen kann.

3.3

Im HEAD Block sollte unbedingt der Titel der Seite angegeben werden, auch die verwendete Zeichenkodierung ist sinnvoll, falls der Server eine andere Zeichenkodierung standartmäßig benutzt.

3.4

Der Meta Tag Refresh kann statt der aktuellen Seite neuzuladen auch eine andere Seite aufrufen, diese funktioniert dann wie eine automatische Weiterleitung.

```
<meta http-equiv="refresh" content="10;url=https://www.google.de/">
```

4.1

Das span Element wird dazu verwendet um Inline Elemente zu umschließen um diese anschließend mit CSS zu formatieren. Das div Element hingegen wird dazu genutzt um mehrere Elemente zu Gruppieren in einen semantischen Abschnitt.

4.2

Ordered List, hier spielen die Reihenfolge der Listenelemente semantisch eine Rolle

Unordered List, hier spielen die Reihenfolge der Listenelemente semantisch keine Rolle

Description List, Liste enthält Wertepaare. Eines der Elemente ist der zu Beschreibende Ausdruck und das andere Element ist die Beschreibung dazu.

4.3

Man kann zum Beispiel mit `` einen Text Fett markieren, mit `<strike>` diese durchstreichen, mit `<u>` den Text unterstreichen und mit `<i>` diese kursiv schreiben.

4.4

Das `rowspan` Attribut erweitert die Zelle einer Tabelle in vertikaler Richtung während das `colspan` Attribute die Zelle der Tabelle in horizontaler Richtung erweitert.

4.5

Der Wert in das Attribut `alt` wird angezeigt, wenn die Bilddatei nicht geladen werden kann.

4.6

Der Verweisanker wird innerhalb eines HTML Dokument dazu verwendet um einen Besucher direkt zu dieser Stelle zu führen, wenn er die Webseite mit dem entsprechenden URL aufruft. Ein solcher URL enthält ein `...#fragment` teil der direkt zu der Verweisanker führt.

4.7

Ä	<code>&Auml;</code>	Großbuchstabe für den deutschen Umlaut A
Ö	<code>&Ouml;</code>	Großbuchstabe für den deutschen Umlaut O
Ü	<code>&Uuml;</code>	Großbuchstabe für den deutschen Umlaut U
ä	<code>&auml;</code>	Kleinbuchstabe für den deutschen Umlaut a
ö	<code>&ouml;</code>	Kleinbuchstabe für den deutschen Umlaut o
ü	<code>&uuml;</code>	Kleinbuchstabe für den deutschen Umlaut u
ß	<code>&szlig;</code>	Scharfes S / sz

5.1

Subjektive Antwort, meiner Meinung nach ist die Möglichkeit Video- und Audiodateien ohne Plug-In abspielen zu können eines der besten Feature in HTML5. Für den Nutzer selbst bedeutet weniger Einrichtung Aufwand und höhere Sicherheit durch den Verzicht externen Media-Player.

5.2

Eine Webseite bestimmt mein Standort um die nächstgelegene Filiale anzeigen zu können. Die Wetter vorhersage zu bestimmen. Ortspezifische Nachrichten anzeigen.

5.3

Smartphone haben sehr oft Verbindungsprobleme, besonders wenn das Highspeed Volumen eines Monates aufgebraucht ist. Dann ist diese Funktion sinnvoll, um weiterhin angenehm Webseiten betrachten zu können.

5.4

Die neuen Input Typen sind sinnvoll um die Eingabe des Nutzers, noch vor dem Versenden, validieren zu können, ob es sich dabei auch um korrekte Eingabe handelt.

B Literaturverzeichnis

[DRJLIAMK98] Dave Raggett, Jenny Lam, Ian Alexander and Michael Kmieciak: Raggett on HTML 4; 2. Auflage; Addison-Wesley Longman Publishing Co., Inc; Boston, MA, USA ©1998; ISBN:0-201-17805-2

[SELFHTML19] SELFHTML e.V.; SELFHTML-Wiki
<<https://wiki.selfhtml.org/>>

[W3SCHOOL9] w3schools.com; W3Schools Online Web Tutorials
<<https://www.w3schools.com/>>

[MC19] Mozilla Corporation: HTML: HyperText Markup Language
<<https://developer.mozilla.org/de/docs/Web/HTML> >

[FE10] Dipl.-Math Frank Elsner: Einführung in HTML und CSS; Osnabrück 2010
< <http://www.home.uni-osnabrueck.de/elsner/Skripte/html.pdf>> (30.11.2018)

[HTMLSEMINAR] HTML-Seminar.de: HTML: Block- und Inline-Elemente verstehen
< <https://www.html-seminar.de/block-inline-elemente-verstehen.htm>>(30.11.2018)

[FD19] Prof. Dr. Frank Dopatka: Webbasierte Systeme (WEB) 02a – Frontend-Technologien im Detail – HTML; Mannheim 2019;
<https://moodle.hs-mannheim.de/pluginfile.php/184333/mod_resource/content/0/WEB%20-%20V02a%20-%20Frontend-Technologien%20im%20Detail%20-%20HTML.pdf>

[HTML5GEO19] W3Schools.com: HTML5 Geolocation
<https://www.w3schools.com/html/html5_geolocation.asp>

[W3LAG14] W3L AG: Offline Apps mit HTML5
<https://www.w3l.de/de/fileadmin/user_upload/Offline_Apps_mit_HTML5_2014.pdf>

[PK19] Peter Kröner: Eine kleine Canvas-Einführung
<<https://www.peterkroener.de/eine-kleine-canvas-einfuehrung/> >

C Verwendete Tools

[TEv36] Tryit Editor v3.6 – W3Schools

< https://www.w3schools.com/html/tryit.asp?filename=tryhtml_basic >

D Abbildungsverzeichnis

Abb. 1 Eine einfache HTML-Dokument und dessen Darstellung im Browser	5
Abb. 2 Den Titel eines HTML-Dokument im Browser-Kopf.....	8
Abb. 3 Die Webseite https://www.informatik-aktuell.de/autoren-cv/prof-dr-frank-dopatka.html Mit dem Titel "Prof. Dr. Frank Dopatka Informatik Aktuell" und dessen Darstellung in der Suchmaschine Google	8
Abb. 4 Code-Beispiele zu dem Namen/Content-Meta-Tag.....	9
Abb. 5 Darstellung eines HTML-Codes in Webbrowser, wo die eingebundene CSS-Datei überschrieben wird.....	10
Abb. 6 Angaben einer Adresse in einen Absatz mit Hilfe von Zeilenumbrüchen	12
Abb. 7 Quellcode mit HTML-Textformatierung Elementen zeigen	13
Abb. 8 Der Code bewirkt, dass "ueberspannte Stelle" rot eingefärbt wird.....	14
Abb. 9 Der Code zeigt wie in <code></code> Tag selber ein Text mittels CSS formatiert werden kann	14
Abb. 10 Beispiel-Code für einen Verweis	15
Abb. 11 Beispiel wie man einen Verweisanker anlegt und auf diesen verweisen kann.....	15
Abb. 12 Beispiel für einen Verweisanker mit Hilfe des id-Attribut.....	15
Abb. 13 Beispiel-Code zum Einbinden eines Bildes	15
Abb. 14 Beispiel-Code um ein Bild als Linkverweis zu nutzen	16
Abb. 15 Code-Beispiel für eine geordnete Liste und dessen Darstellung im Webbrowser	16
Abb. 16 Beispielcode für eine verschachtelte geordnete Liste, sowie dessen Darstellung im Browser	16
Abb. 17 Code-Beispiel für ungeordnete Liste und dessen Darstellung im Webbrowser	17
Abb. 18 Codebeispiel einer verschachtelten ungeordneten Liste sowie dessen Darstellung im Webbrowser.....	17
Abb. 19 Codebeispiel einer Description List, sowie dessen Darstellung in einem Webbrowser.....	18
Abb. 20 Codebeispiel für eine Tabelle und dessen Darstellung im Browser	18
Abb. 21 Codebeispiel für die Formatierung einer Tabelle mit rowspan und colspan sowie dessen Darstellung im Webbrowser.....	19
Abb. 22 Beispiel-Code zum Einbinden eines Videos oder Audiodatei in HTML5.....	21
Abb. 23 Codebeispiel für ein Formular zum Abfragen des Namens und dessen Darstellung im Webbrowser.....	22
Abb. 24 Code-Beispiel um mit Canvas eine Fläche zu zeichnen [PK19]	22
Abb. 25 Codebeispiel für eine Standortabfrage [HTML5GEO19]	23