

# Modul 133

## Block 01

### HTML 5 & HTTP(S)

## 1 HTML

### 1.1 Historie und Grundwissen

Die Hypertext Markup Language, abgekürzt HTML, ist eine textbasierte Auszeichnungssprache zur Strukturierung digitaler Dokumente wie Texte mit Hyperlinks; Bildern und anderen Inhalten. HTML wurde 1989 von Tim Berners-Lee erfunden um es Forschern am CERN (Genf) möglich zu machen zwischen auf verschiedenen Computern verteilten Dokumenten zu navigieren.

HTML wird vom World Wide Web Consortium (W3C \*), dem Gremium zur Standardisierung der Techniken im World Wide Web und der Web Hypertext Application Technology Group (WHATWG) weiterentwickelt. Die aktuelle Version ist seit dem 28. Oktober 2014 HTML5, die bereits von vielen aktuellen Webbrowsern und anderen Layout-Engines unterstützt wird.

HTML, wurde mittels SGML (Standard Generalized Markup Language) definiert. SGML ist eine Metasprache, mit deren Hilfe man Markup Sprachen für Dokumente definieren kann. Um ein gültiges Dokument in SGML zu beschreiben, wird die Dokumenttypdefinition (DTD) verwendet, die den strukturellen Aufbau von Dokumenten beschreibt.

Die Regeln für HTML sind also mit Hilfe von SGML formuliert. Nach den Regeln einer SGML basierten Auszeichnungssprache ist eine HTML-Datei erst dann eine gültige (valide) Datei, wenn sie einen bestimmten Dokumenttyp angibt und sich dann innerhalb des restlichen Quelltextes genau an die Regeln hält, die für diesen Dokumenttyp definiert sind. Denn hinter jeder Dokumenttyp-Deklaration stecken so genannte Dokumenttyp-Definitionen (DTD). In den für HTML entwickelten Typen-Definitionen ist geregelt, welche Elemente ein Dokument vom Typ HTML enthalten darf, welche Elemente innerhalb von welchen anderen vorkommen dürfen, welche Attribute zu einem Element gehören, ob die Angabe dieser Attribute Pflicht oder freiwillig ist usw. Konkret heisst das, dass in einer DTD die Struktur des Dokuments festgelegt ist.

## 1.2 Bis HTML 4.x

Eine gewöhnliche HTML-4.x Datei besteht grundsätzlich aus folgenden Teilen:

- Dokumenttyp-Deklaration (Angabe zur verwendeten HTML-Version)
- Header (Kopfdaten. z.B. Angaben zu Titel u.ä.)
- Body (Körper - anzuzeigender Inhalt, also Text mit Überschriften, Verweisen, Grafikreferenzen usw.)

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
  <head>
    <title>Beschreibung der Seite</title>
  </head>
  <body>
    <p><b>Hallo Welt!</b></p>
  </body>
</html>
```

Bei HTML 4.x wird die Dokumenttyp-Deklaration am Anfang der HTML-Datei vor dem einleitenden `<html>`-Tag in Großbuchstaben notiert. Hinter der startenden spitzen Klammer folgt ein Ausrufezeichen. Dahinter folgt die Angabe `DOCTYPE HTML PUBLIC`. Das bedeutet, dass auf die öffentliche verfügbare HTML-DTD verwiesen wird.

Die folgende Angabe, die in Anführungszeichen steht, ist wie folgt zu verstehen:

W3C ist der Herausgeber der DTD. Eine Angabe wie `DTD HTML 4.01 Transitional` bedeutet, dass Sie in der Datei den Dokumenttyp "HTML" verwenden, und zwar in der Sprachversion 4.01 und deren Variante Transitional. Das `EN` ist ein Sprachenkürzel, welches angibt, in welcher Sprache der Inhalt des Dokuments verfasst wurde.

Ferner enthält die Dokumenttyp-Deklaration die Web-Adresse der Dokumenttyp-Definition (DTD). Die Angabe dieser Adresse ist nicht zwingend nötig, da die Browser diesen Standard-Pfad bereits kennen. Jedoch könnte eine alternative URL mit einer anderen Definitionsdatei angegeben werden über die alternative HTML-Regeln beschrieben würden.

**\*) Nebst HTML standardisiert das W3C Technologien wie, XHTML, XML, RDF, OWL, CSS, SVG und WCAG.**

## 1.3 XHTML

Der W3C-Standard **Extensible HyperText Markup Language**, kurz XHTML, ist ebenfalls eine textbasierte Auszeichnungssprache, welche nun aber erweiterbar ist. Es ist eine Neuformulierung von HTML 4.01 in XML: Im Gegensatz zu HTML, welche mittels SGML (Standard Generalized Markup Language) definiert wurde, verwendet XHTML die *strengere* und einfacher zu parsende SGML-Teilmenge XML als Sprachgrundlage. XHTML-Dokumente genügen also den Syntaxregeln von XML.

Noch vor der Dokumenttyp-Deklaration sollte bei XHTML-Dateien der Bezug zu XML hergestellt werden. Dazu dient die erste Zeile mit den Fragezeichen hinter der öffnenden spitzen Klammer und vor der schliessenden spitzen Klammer.

```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE HTML PUBLIC "-//W3C//DTD XHTML 1.1//EN" "http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
  <head>
    <title>Beschreibung der Seite</title>
  </head>
  <body>
    <p><strong>Hallo Welt!</strong></p>
  </body>
</html>
```

Die vom World Wide Web Consortium (**W3C**) herausgegebene XML-Spezifikation definiert eine Metasprache, auf deren Basis durch strukturelle und inhaltliche Einschränkungen anwendungsspezifische Sprachen definiert werden. Diese Einschränkungen werden durch Schemasprachen wie **DTD** oder XML Schema ausgedrückt. Beispiele für XML-Sprachen sind: RSS, MathML, GraphML; XHTML, XAML, Scalable Vector Graphics (SVG), etc.

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<verzeichnis>
  <titel>Wikipedia Städteverzeichnis</titel>
  <eintrag>
    <stichwort>Genf</stichwort>
    <eintragstext>Genf ist der Sitz von ...</eintragstext>
  </eintrag>
  <eintrag>
    <stichwort>Köln</stichwort>
    <eintragstext>Köln ist eine Stadt, die ...</eintragstext>
  </eintrag>
</verzeichnis>
```

## 1.4 HTML 5

Das W3C hat im Oktober 2014 die fertige HTML5-Spezifikation vorgelegt (HTML5 wird jedoch trotzdem stetig weiterentwickelt, was sich z.B. darin zeigt, dass im März 2018 die HTML 5.2 Spezifikation verabschiedet wurde).

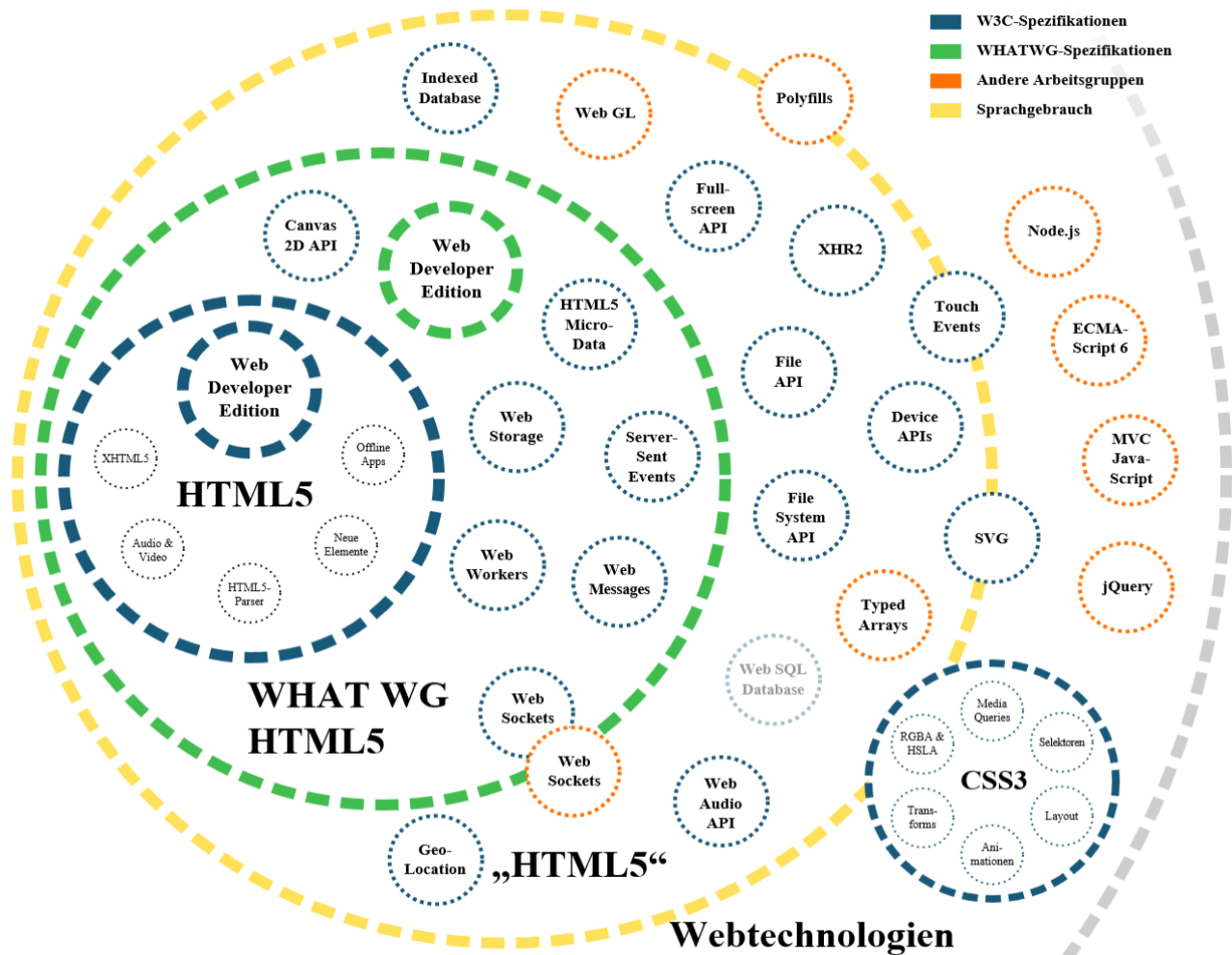
Seit 2014 dient also HTML5 als Nachfolger von HTML4 als die Kernsprache des Webs. Es ersetzt u.a. die Standards HTML 4.01, XHTML 1.0. Unter anderem bietet es neue Funktionen wie Video, Audio, lokalen Speicher und dynamische 2D- und 3D-Grafiken, die von HTML4 nicht direkt unterstützt wurden und sich ohne HTML5 nur mit zusätzlichen Plugins (z. B. Adobe Flash) umsetzen liesse.

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>MyTitle</title>
  </head>
  <body>
    <p>Hallo Welt!</p>
  </body>
</html>
```

```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE HTML PUBLIC "-//W3C//DTD XHTML 1.1//EN"
"http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
  <head>
    <title>MyTitle</title>
  </head>
  <body>
    <p>Hallo Welt! </p>
  </body>
</html>
```

Sie merken, dass der Grundaufbau einer HTML5 Seite einfacher und kürzer geworden ist. Eine HTML5 Grundstruktur besteht aus: **doctype**, **charset**, **head** und **body**

Es gibt nicht den Webstandard HTML5. HTML5 ist ein Technologie-Ökosystem, das aus versch. Technologien besteht, welches von unterschiedlichen Gremien spezifiziert wird.



Die Elemente im grünen Kreis stellen die HTML5.0 Spezifikation der WHAT WG dar. Der blaue Kreis ist die Spezifikation der W3C, die auf der Basis der WHAT WG Spezifikation entwickelt wurde. Das w3c hat die WHAT WG Spezifikation nicht einfach nur übernommen. Da einige Elemente nichts mit HTML zu tun haben (z.B. Canvas 2D API ist eine JavaScript Schnittstelle zur Grafikprogrammierung oder WebWorkers ist eine Technologie für JavaScript Multithreading) wurden diese Teile ausgelagert. Das W3c hat deshalb nur die Elemente in die Spezifikation übernommen, die etwas mit HTML zu tun haben. Anhand der blaupunktierten Kreise sieht man aber, dass auch die W3C Elemente wie WebWorkers spezifiziert hat, aber in eigenen Dokumente. Die HTML5.0 Spezifikationen kann man sich vorstellen wie Anleitungen um einen Webbrowser zu programmieren. Kein Web Browser hat absolut alle Standards implementiert, aber wenn man genügend Features implementiert hat, wird der Web Browser als vollwertig betrachtet (letztlich sind es jedoch die User, welche darüber entscheiden). In der Praxis braucht man Technologien aus versch. Bereichen (gelb) um Webapplikationen auf der Basis von HTML5 zu erstellen.

### 1.4.1 Validierung

Mittels des «Markup Validation Service» von W3C können sie die Qualität von HTML überprüfen.  
<https://validator.w3.org/> .

### Aufgabe 01 Partnerarbeit, 5'

Überprüfen Sie exemplarisch die Schulwebsite auf Fehler. Was stellen Sie fest?

Auch der Inhalt von CSS kann überprüft werden: <http://jigsaw.w3.org/css-validator/>

### 1.4.2 Browserunterstützung

Auch wenn HTML5 nun schon eine ganze Weile da ist, so entscheidet das W3C noch immer über einzelne Spezifikationen und auch die Browserhersteller haben noch nicht alles implementiert.

Um sich einen Überblick darüber verschaffen zu können, welche Features in welchen Browsern in welcher Version bereits eingebaut sind, können Sie die Website [www.caniuse.com](http://www.caniuse.com) verwenden.

## 1.5 Semantic Web

Eines der Ziele von HTML5 war es das Web semantischer zu machen. Dies bedeutet, dass die einzelnen Inhalte einer Website in thematisch und inhaltlich korrekten HTML-Elementen verpackt werden sollte, damit Lesbarkeit sowohl für Mensch wie auch Maschine erhöht werden kann. Maschinen dient die verbesserte Lesbarkeit um beispielsweise für Suchmaschinen besser kenntlich zu machen, welcher Teil der Website der tatsächliche Inhalt und was Werbung ist. Bei Menschen ist eine bessere Lesbarkeit selbstverständlich immer ein Vorteil, da wir als Entwickler sehr viel mehr Zeit für das Lesen als für das Schreiben von Code (und auch Markup) verwenden.

### Aufgabe 02 Partnerarbeit, 10'

Suchen Sie im Internet nach mind. 5 unterschiedlichen schematischen HTML-Elementen und beschreiben Sie, wofür sie eingesetzt werden sollten.

## 2 HTTP

Bis zu diesem Zeitpunkt haben Sie bereits regen Gebrauch vom Hypertext Transfer Protocol gemacht. Wie Sie wissen, wird HTTP insbesondere dafür gebraucht, damit ein Web Browser mit einem Web Server über einen grundsätzlich frei wählbaren Port kommunizieren kann.

### Aufgabe 03 Partnerarbeit, 10' (Selbstständige Suche im Internet)

- a) Notieren Sie sich den Standard-HTTP-Port und den am meisten verwendeten Backup-Port.

80 & 8080



- b) Wie lautet der Standard-Port für HTTPS?

443

- c) Was ist der Vorteil, wenn Sie den HTTP-Standard-Port anstelle eines von Ihnen frei definierten Ports verwenden?

Keine Überschreibung & Port muss nicht eingegeben werden (usability)

- d) Es gibt viele sog. Well-Known Ports. Zählen Sie die für Sie wichtigsten auf und notieren Sie ab welchem Port die Nummern nicht mehr reserviert, aber durch die IANA «vergeben» sind.

FTP = 20 / 21  
SSH = 22  
SMTP = 25  
143 = IMAP, 220 = IMAP3  
3306 = MySQL  
53 = DNS  
118 = SQL  
[https://de.wikipedia.org/wiki/Liste\\_der\\_standardisierten\\_Ports](https://de.wikipedia.org/wiki/Liste_der_standardisierten_Ports)

## 2.1 Request

Ein Grossteil der Kommunikation zwischen Browser und Server besteht darin, dass der Browser eine Ressource des Servers anfordert und der Server diese zurückgibt.

### Aufgabe 04 Partnerarbeit, 5'

Erstellen Sie in VS Code ein HTML-Dokument und nutzen Sie **Live-Server** (VS Code Extension), um Ihre HTML-Datei in einem vereinfachten http-Server zu hosten.

Erstellen Sie nun eine .http Datei und nutzen Sie die VS Code Extension **REST Client**, um Ihr HTML-Dokument auf dem Server aufzurufen.

Wie sieht ein HTTP-**GET**-Request aus? Welche Bestandteile können Sie ausmachen?

```
GET /Block1/test.html HTTP/1.1
Host: 127.0.0.1:5500
```

### Aufgabe 05 Partnerarbeit, 5'

Das Protokoll sieht jedoch noch weitere Request-Methoden vor, welche HTTP-Request Methoden kennen Sie und wofür werden diese verwendet? (Notieren Sie sich mind. 4 Methoden und ihren Verwendungszweck)

GET = Mit der GET-Methode können Sie eine Ressource (zum Beispiel eine Datei) vom Server anfordern.  
POST = Mit der POST-Methode können Sie große Datenmengen (wie Bilder oder HTML-Formular-Daten) zur weiteren Verarbeitung zum Server senden.  
HEAD = Die HEAD-Methode weist den Server an, die gleichen HTTP-Header wie bei GET, nicht jedoch den Nachrichtentrumpf mit dem eigentlichen Dokumentinhalt zu senden.  
So kann zum Beispiel schnell die Gültigkeit einer Datei im Browser-Cache geprüft werden, indem die Größe verglichen wird.  
PUT = Die PUT-Methode dient dazu eine Ressource (zum Beispiel eine Datei) unter Angabe des Ziel-URLs auf einen Webserver hochzuladen.  
Dabei muss sie nicht wie bei POST durch ein Skript verarbeitet werden, sondern wird an der in der ersten Zeile angegebenen Stelle platziert.  
DELETE = Die DELETE-Methode löscht die angegebene Ressource auf dem Server.

## 2.2 Response

Nachdem ein HTTP-Request vom Browser an den Server gesendet wurde, muss der Server HTTP konform eine Antwort zurückschicken.

### Aufgabe 06 Partnerarbeit, 5'

Nutzen Sie wiederum **REST Client**, um die HTTP-Antwort Ihres Servers zu analysieren. Notieren Sie sich den Aufbau der Antwort. Welche Bestandteile müssen zwingend vorhanden sein?

Als erster kommt der Header der Response, Nachher die Seite als HTML Text.

## 3 HTTPS

Wie Sie wissen, steht das S in HTTPS für Secure. HTTPS ist also die sichere Variante von HTTP. Dabei ist HTTPS syntaktisch identisch mit dem HTTP-Schema. Bis auf den Verschlüsselungslayer funktioniert HTTP und HTTPS also exakt gleich.

Die zusätzliche Verschlüsselung, welche für eine höhere Sicherheit sorgt, geschieht mittels SSL/TLS: Unter Verwendung des SSL-Handshake-Protokolls findet zunächst eine geschützte Identifikation und Authentifizierung der Kommunikationspartner statt. Anschliessend wird mit Hilfe asymmetrischer Verschlüsselung oder des Diffie-Hellman-Schlüsselaustauschs ein gemeinsamer symmetrischer Sitzungsschlüssel ausgetauscht. Dieser wird schliesslich zur Verschlüsselung der Nutzdaten verwendet. Die verschlüsselte Kommunikation dient in erster Linie dazu Man-in-the-Middle Angriffe zu unterbinden, also die Kommunikation abhörsicher zu gestalten.

Wichtig ist in diesem Zusammenhang zu wissen, welche Teile der Kommunikation denn nun verschlüsselt und welche unverschlüsselt stattfinden: Grundsätzlich wird bei HTTPS alles verschlüsselt übertragen, mit einer einzigen Ausnahme: Die Domain. Ein Angreifer kann also sehen, welche Domain Sie aufgerufen haben, jedoch nicht welche Ressource:

```
https://de.wikipedia.org/w/index.php?search=https&title=Spezial
```

Domain

Ressource

Parameter

Sowohl die Ressource, wie auch GET- und POST-Parameter, Formular-Inhalte und auch Cookies werden also sicher übertrage. Die Domain wird unsicher übertragen, weil zuerst ein DNS-Server angefragt werden muss, welche IP-Adresse denn nun hinter der Domain verborgen ist. Und da der DNS-Server die verschlüsselte Kommunikation nicht lesen darf, wird die Domain in Klartext übertragen, sodass der DNS-Server seine Arbeit machen kann.